5-2018

Analysis and Implementation of E-ticketing with Live Chat

Laxmi Varthi
St. Cloud State University, llvarthi@stcloudstate.edu

Analysis and Implementation of E-ticketing with Live Chat

by

Laxmi Lavanya Varthi

A Starred Paper

Submitted to the Graduate Faculty of

St. Cloud State University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in Information Assurance

May, 2018

Starred Paper Committee:
Susantha Herath, Chairperson
Mark Schmidt
Sneh Kali

Abstract

In this paper, idea of developing a system - online event ticketing- is presented along with practical solutions to some of the biggest issue - security. This is a web based application with main aim to simplify the user's dependability on ticket sellers or outlets or kiosks to buy the event tickets and reduce the fraud of online transactions through modern and secured payment gateways allowing the user to use credit/debit card to complete their payments. User can access the system through website at any time through mobile/laptop or any smart device instead of going to the campus to buy the event tickets. This application would also allow the user to buy tickets individually or buy tickets for group by registering the group through a secured payment gateway – 'PayPal'. Many events hosted in the college will have different pricing for students, staff, faculty, outsiders. For suppose, a student planning to buy tickets for a cultural event with different pricing levels for him and his parents can create a group with three of them and pay for one student and two outsiders by selecting the payment levels requested during the group registration. These features of the application are incorporated in the application to address the problems of group tickets with different pricing levels and secured payments which is considered as the major challenge for online transactions systems.

Keyword: E-Ticketing

Table of Contents

Page

Chapter Page

Chapter Page

List of Tables

Table Page

List of Figures

Chapter 1: Introduction

E-Ticketing is a website event management service. It is an easy way to organize the event, sell tickets and manage registration for all kinds of events. Many organizations are experiencing a high demand to an option of purchasing tickets online as manual ticketing process is too much time consuming for staff manage tickets for different events. This is especially challenging for organizations whose budgets and staff are small. Another challenge organizations are facing is the need to track ticket sales, prevent the misuse of tickets. For example, a same ticket being used twice and to provide a greater accountability and reporting to facility.

So, with all these challenges what are the benefits offered by utilizing the online ticketing? Organization’s ability to sell tickets online from your organization’s website and providing the customers customized ticketing experience 24/7 access to the tickets can also increase the ticket sales potential. It provides a single solution for variety of different events and simplifies the ticket management procedure by automating much of your ticking process including seat selection, payment processing, tracking and reporting. This allows the organization staff to focus more time on the event and less time on the paperwork. Finally, it allows the organization to build a strong relationship with customers by providing a convenient flexible ticketing experience.

As part of the customized ticketing application, the organization will be able to sell tickets directly from dedicated webpage. Customers will access the ticketing application through the organization’s website giving the customer a streamlined ticketing experience.

All tickets are not distributed equally, the organization may wish to sell tickets for variety of events. Online ticketing system allows you to set up reserved seating and general admission events all within the same system. Organization can also sell tickets to general admission by section or a mixed venue of reserved seating and general admission within the same event. Events may be performed at various venues or utilize unique venue setups. The ticketing software allows the organization to setup multiple venues and select which venues will be used for each event. Venue graphic can also be included to provide the user with more visual experience.

Each event may have a unique ticket design to fit the event. Tickets may also be designed to include number of helpful elements such as ticket ID, seat assignment, performance information and other helpful information. Organization's logo can be included on the ticket. For security, a unique bar code may also be included on the ticket. This will assist in tracking ticket and preventing the ticket duplication.

Finally, to give the customers more customized experience o ticketing, is by giving them the option to select how they wish to receive their tickets. Online ticketing provides your organization with following four convenient delivery methods for each event.

1. E-Ticket to print at home
2. Mail the tickets
3. Pick the tickets a will call office
4. Print them at the venue

Organization may select which of the above optional methods is best for each event. While online ticketing encourages customers to buy tickets in advance, staff will need options for assisting customers at the venue.

Tracking the ticket sale will be critical to your event success. The ticketing software provides real-time reports that allow you to track your ticketing events by ticket order, event performance and price level.

E-Commerce is a study of electronic transactions which tends to have high demand and development. One can understand that; online event ticketing is one sort of e-commerce enterprise that will develop quickly and keep on thriving in the market although introduced years prior. Expanding technology however brought a set of new difficulties and dangers for each organization in business. Some of such difficulties incorporate security enforced payment methods, replacing the old technologies with new ones to satisfy the user needs and sustain in the market. Different strategies were required to handle such challenging situations. Choosing the right way to solve these problems is the very important factor for success in this competitive market. The opportunities were vast which provided the ticket sellers a good path to expand their business, efficient ways to cope with the emerging challenges.

A long time before the begin of web based ticketing to even now, numerous colleges or the venues altogether rely on upon their outlet or booth built up at different spots, as the main methods for permitting customers to buy tickets at same time. These outlets require trained staff, hardware resources which in turn incur extra cost, and are not open all the time i.e., they are limited by their hours of operation. Traditional ticketing agents or outlets for selling tickets in this advanced world are more complicated as they involve more labor, money and time. Manual ticketing faces with the problem of distributed pricing for different individuals when a single individual is trying to buy tickets for entire group. For example, student trying to buy a ticket for an outsider, will usually ask for two tickets but it is difficult for the ticket seller to figure out whether the second ticket is for student or outsider to charge accordingly.   Internet has changed the entire process of ticketing, by enabling virtual ticketing available at each door step enabling the access through laptop or even a mobile phone making ticketing system more comfortable, easier, flexible with less or no cost.

As we all know, ticket sales are directly proportional to the revenue generated towards any event. There are many events like cultural nights, karaoke nights, sports for entertainment, workshops for practical knowledge development, job fairs, classes, conferences for the student career development, fundraising events etc., are being organized by every university.

With the advancement of technology, with new software and new payment methods, the online event ticketing systems are subject to new challenges of adjusting and overcoming all these difficulties to retain their main asset - customer satisfaction - in this competitive market. Internet ticketing system was named suitably, focusing the way that it is the quickest, less expensive and most helpful method for purchasing tickets, in the meantime this system can be environment friendly and more secure way.

Need of Study

With all the advantages understood from a little research on the e-ticketing provokes:

1. To study and become familiar with the process, implementation and technology used in E-Ticketing.
2. To study the requirement to develop e-ticketing web application with live chat for the university of saint cloud state.
3. To analyze the architecture of e-ticketing and present the output results.

Problem Statement

Although there are many online ticketing websites known as ticket vendors to overcome the problem of manual ticketing, none of these websites provide customers with immediate help. This is a major problem because customer when in doubt or did not find any event related information, will hardly think of contacting you for questions as we all know how long it takes the customer to reach the customer support. Rather they try to get the tickets through other ticketing vendors these websites charge the organizers or the customers. Additionally, these vendors charge fixed amount on any kind of event whereas other vendors charges vary upon events, celebrities attending the events and days before the event when you want sell tickets for your events through their website. College events can be sold to maximum only if the charge ticket price is reasonable. Opting to sell tickets through a ticket vendor would increase the price of each ticket.

Significance of the Problem

Ticketing through vendors, when used as means to sell tickets requires people to invest more on tickets which in turn will ultimately reduce the number of attendees to the event, reducing the revenue generated for the event. Considering all these factors, it is very important to choose a right ticket vendor to sell the tickets. Customers when in doubt or could not find any event or event details while searching will hardly think of contacting you for questions as we all know how long it takes the customer to reach the customer support resulting in reduction of number of customers.

This main aim is to provide a live chat service making the search easy and reduce time consumed to buy tickets with user friendly and secured service that can be run on any Internet browser enabling the user to buy single or multiple tickets with most secured payment gateway, while helping the university sell tickets with no extra charge, manage the events, event information, attendees, pricing levels for various attendees and revenue generated from these event ticket sales.

Research Questions

1. Why is this ticketing system different than the other ticketing software?
2. Does this application have any additional feature compared to other ticketing sites?

3. What is live chat? How is it useful to the user using this application?
4. Who all can use this application?
5. Is the system limited to University students?
6. Which technology is best to implement this application?
7. What are the different levels of pricing?
8. Can the system handle different pricing levels assigned to various attendees? How?
9. What are the various payment gateways?
10. Why is PayPal payment gateway preferred to other payment gateways?

Definition of Terms

Table 1. Definition of Terms

UML	Unified Modeling Language
ASP	Active Server Pages
ANSI	American National Standards Institute
ISO	International Organization for Standardization
SQL	Standard Query Language
T-SQL	Transact – Standard Query Language
IDE	Integrated Development Environment
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
ADO.NET	.NET Database Connectivity
CLR	Common Language Runtime
XML	Extensible Markup Language
GUI	Graphical User Interface
SSL	Secure Socket Layer
SET	Secure Electronic Transaction

IIS	Internet Information Services
SDK	Software Development Kit
www	World Wide Web
TCP/IP	Transmission Control Protocol/ Internet Protocol
HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Secure
XSS	Cross Site Scripting
RAD	Rapid Application Development
UAT	User Acceptance Testing
DML	Data Manipulation Language
DDL	Data Definition Language
DCL	Data Control Language
URL	Unified Resource Locator

Research Methodology

The information required for this research is gathered from two sources:

Primary sources. World Wide Web (WWW)

These days if we have the internet connection using the web is easy. I’ve visited many webpages regarding existing e-ticketing systems, e-commerce, online finance systems, payment gateways, live chat, and many other webpages related to the technologies used in this application to understand the working, enhance the knowledge and implement the acquired knowledge in developing this application.

Secondary sources. IEEE papers and related articles.

I’ve utilized the secondary sources to gain a deep insight on e-ticketing, e-commerce, why PayPal is most used payment gateway. This way of gaining knowledge is bit time consuming as it requires to read the whole paper or the article to better understand the topic. However, the secondary sources helped to enhance my knowledge gained from primary sources to another extent.

Summary

This chapter introduced Online Ticketing system along with the brief overview of the problem with manual ticketing system, how important the problem was and main objective of conducting this study. Next chapter will discuss the background details and literature review of problem and proposed system.

Chapter 2: Literature Review and Background

Literature Review of the Problem

Of a concert, play, tour or a sports event can be predicted based on the ticket sales for those events. Choosing an event ticketing platform has a direct impact on when and how your revenue is collected and how tickets are delivered. Most importantly it impacts the price of tickets and the effect that it has on sales and ultimately on consumer satisfaction. Selecting the right platform or company for a specific venue starts with an understanding of the five event ticketing types available says (Tickets, 2016). This article also describes the diverse types of vendors charge for event ticketing are as follows.

Monthly Service Model

This is a new model which offers a monthly service with zero per-ticket fees. The lack of convenience charges results in lower ticket prices. This means that all ticket proceeds are deposited nightly directly to the bank. Ticket delivery is provided via email, and if on-site thermal ticket printers are available, integrated printing allows for hard ticket delivery by regular mail. The results of using this type of platform are lower ticket costs that translate into higher sales. Additionally, there is a substantial financial benefit of having cash on hand through nightly deposits. In the case of venues choosing to add their own fee, they can retain 100% of all fees collected. When it comes to music venues, this type of revenue is not shared with the band so it represents an additional revenue stream.

Fee added to ticket price. This ticketing model results in fees upwards of 25% or more of the total ticket cost. Tickets are delivered via email as printable ticket, or if consumers select a hard ticket it is sent at a higher cost by regular mail. The convenience fees for ordering online are added onto the ticket cost even when tickets are sold at the door. This oldest online model is the most profitable for the ticketing company. The downside of the higher ticket price is the negative effect it has on sales for small venues. This model is slowly plummeting on the market considering the excessive cost and low customer response to the events ticketed by this type of service.

Fee rolled into ticket price. The fees are the same as those mentioned in the previous model, the difference being that they are rolled into the ticket price itself and thus hidden from the customers. Opting for this type can result in a sudden jump of ticket price in online sales which in return causes significant loss in the number of regular customers. This model has the same disadvantages as the previous one - higher prices resulting in lower sales number. It can lead to mistrust between the vendor and the customer due to hidden costs. The venue itself receives the proceedings after the event takes place.

Fee sharing. A new twist on the old model. The ticket companies provide a share of the fees collected. Translated into numbers it means that if the added ticket fee is $5.00 and the venue’s share is 10%, the venue gets 50 cents towards its ticket sales revenue. In this case 90% of the profit would go to the ticket company. For venues, the problems are the same as with the previous two models. The fee is paid after the event takes place and the drop-in sales is significant due to increased ticket price. With this model ticket delivery is done only via email.

Percentage based ticketing. This model is fairer to venues, considering that fees are not added to the ticket cost. Instead, a percentage is shared with the ticketing company. For example, 10% of the total ticket price. This means that a $10.00 ticket means a $9.00 profit for the venue, and proceeds are paid to the venue after the event takes place. The downside of this model is the fact that venues are responsible for all aspects of ticket delivery upon receiving emailed notification of sales made. This seems like a great option considering that a $25 ticket costs the venue $2.50. The additional costs come from the design, printing, mailing, and administration of the ticket delivery. Choosing an event ticketing platform or company starts with an understanding of how a service affects sales. Sales are driven by more than the event itself: fair ticket prices, low or no fees, and consumer confidence all weigh in on your customer's decision to buy and, in the end, dictate the success of your venue. To date, the most pointed example of potential Internet misuse by the third-party ticket sellers occurred in 2007 was discussed briefly in (Kirkman, 2009).

Payment process. As this application involves e-commerce transactions, use of a payment gateway to send card and order details to the merchant either through merchant processor or through payment gateway will be an efficient and reliable means to make payment for the purchased tickets. Merchant then forwards them to the respective card's bank to check if the provided details were of an authorized user. On successful authorization, the merchant payment processor receives a confirmation, the confirmation is sent to merchant via the payment gateway, then the payment transaction will be completed by the merchant. For the most part, an organization that needs to acknowledge e-transaction requires both, a payment gateway and merchant processor. In a few occurrences, both functions are handled by one organization handling the tasks and responsibilities of both. Examples of such organizations are as Authorize.net, Stripe, Beanstream, and PayPal.

Literature Review of the Proposed System

According to the research conducted by (TicketScript, 2013), digital ticketing over traditional means are evidently having an impact on buyers' preferences: people want to make purchases that are more convenient, practical, environmentally friendly, simple and secure. Digital tickets deliver on these advantages and consumers are taking note.

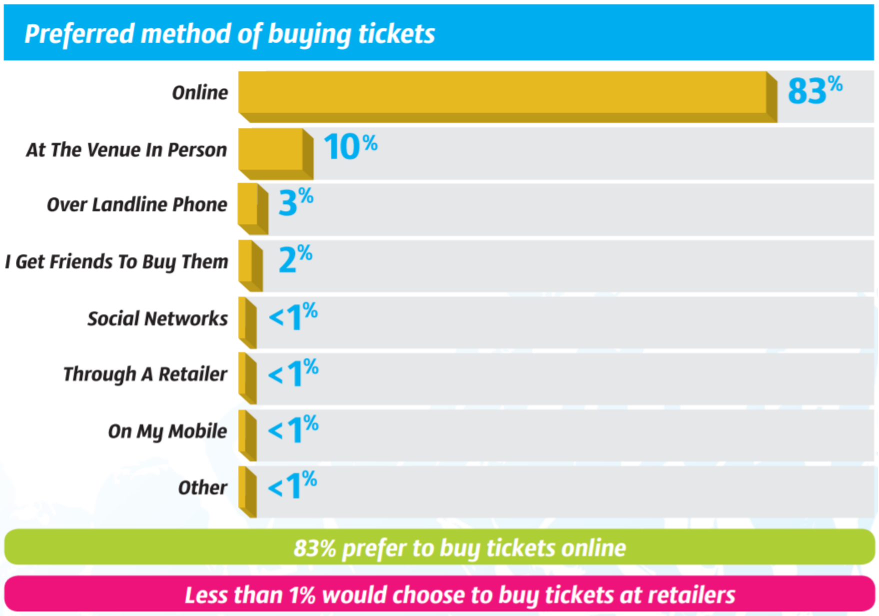


Figure 2.1. Survey on Preferred Method of Buying Tickets

Of eight options given, (TicketScript, 2013) research showed that online purchasing was the overwhelming first choice, with 83% of people selecting it instead of methods such as in-person at the venue or over the phone. In the 18-34 age bracket this number ramps up to 9 out of 10 people.

Standard security protocol. Thinking about developing an E-Commerce website? One should more likely know how important is the security. Security is important for three big reasons. Organization can be held financially liable for stolen credit cards, if there is ever a breach, fine can be imposed, customers lose their trust in case of breach. When the website is insecure, any data sent from customer’s browser to your site can be intercepted. Many people browse and shop at coffee shops or over other public Wi-Fi networks, and when you’re on a network like that, anyone can intercept the data.

Anytime you enter a sensitive information, like credit cards or passwords, you should make sure that the page is secure. As an example, when a user login to Facebook it uses HTTPS to protect your password. HTTPS encrypts traffic, preventing third parties from intercepting your information. HTTPS encrypts website data using the SSL protocol. To use SSL on your website, typing in https is not just sufficient, SSL certificate is required for the website.

If any organization's website is collecting the credit card numbers, one must make sure that the page is secure. Many payment gateways such as PayPal lets the users go off-site for transaction and then they redirect back to the original site to complete the order. In this case because the organization is not taking the credit card numbers, the organization don't necessarily need to make its site secure. While accepting the credit cards directly on you site, following the PCI Compliance is highly demanding for a small business. It's much easier and safer to start with an off-site payment gateway.

According to (H.M. Deitel), SSL and SET technologies are used for data security where data is encrypted and digitally signed before transmission over the Internet. Secure Socket Layer (SSL) and Secure Electronic Transaction (SET) are the standard security protocols that protect the integrity of these online transactions. SSL was developed by Netscape Communications, a nonproprietary protocol used to secure communication in the Internet and the Web. This SSL is built into many web browsers including Netscape Communicator, Microsoft Internet Explorer and numerous other software products. SSL uses public-key technology and digital certificates to authenticate the server in a transaction and to protect the confidential information and transmit the data over the Internet with integrity. However, in the case of online credit card transactions for example, there are more to make than just encrypting the credit card information upon transmission to the merchant – such as checking the validity of the card, the authorization of the card, etc. SET is a cryptographic protocol which was developed by Visa International and MasterCard International was designed specifically to protect and handle the complete online transactions both for the customers and the merchants. SET uses the digital certificates to authenticate each party in a transaction. Additionally, it requires special SET software to process transactions. With this protocol, the merchant never sees the customer's information like the credit card information as it is not stored on his server which reduces the risk of fraud.

Secured payment gateway – PayPal. Everyday all around the world PayPal connects millions of people spending, sending, receiving billions of dollars. How does a PayPal work? To get started with PayPal, an individual is required to link his/her credit card, debit card and bank accounts to the account. Once they are successfully linked, you would not be required to enter them or shipping information again. That is all, now you are ready to use your PayPal account to spend or send and receive money from your friend. You can send money to anyone using their email address or mobile number. Whether they are right next to you or half way across the world, the money goes right to their PayPal account in seconds. When it comes to online shopping, check for the PayPal button at checkout, use your PayPal password and skip entering your credit/debit card, bank account and shipping information. With one touch, you have an option to checkout even faster. Most importantly, your eligible purchases are protected by PayPal purchase protection. If you have a business, you can start accepting cards and other payments through PayPal even from customers who do not have PayPal account. PayPal has another benefiting feature of providing access to capital giving an opportunity to build your business with PayPal working capital. PayPal has also come forward with helping hands by accepting donations. So, you can now donate and accept donations for charity using PayPal. No matter what you do or where you live, money is part of everyone's life. Helping people to find better ways to pay, to get paid and manage money is the main working motivate of this organization. Summing up to the whole description according to (PayPal Advantages, n.d.)

PayPal is secure (Whether you are paying bills, sending money or making a purchase, PayPal works hard to keep your financial information secure), fast (No need to enter all your financial information every time you shop. Speed through checkout with PayPal), No Fees (Not only is signing up for PayPal free, but you can also shop and send money to family and friends at no extra cost.

"Arising from the popularity of eBay online auctions, PayPal (www.paypal.com) has quickly become dominant in online transaction processing" according to (E-Business, 2008). Many people still think of PayPal primarily as the service to use to pay for items they buy on eBay. PayPal originally started as a peer-to-peer money transfer system for eBay auctions, but has also expanded as a third payment processor for any website. Two of their main gateway products that they offer are Payflow Pro and Payflow Link. Retrieved from ((PayPal, n.d.)).

According to (Niranjanamurthy, 2014) PayPal is now also an all-in-one online payment solution. Their Website Payments regular program lets you accept Visa, MasterCard, Discover, and American Express credit card payments as well as bank transfer and offer PayPal as well – with no monthly cost, setup or termination fees. PayPal charges you a fee of 1.9 to 2.9 percent of transaction plus 30 cents per order, depending on your company's sales volume.

Credit card security. Access to user accounts is one of the most valuable targets of hackers. The only one that's larger than that is credit cards. It should be obvious why credit cards are sought after, especially by professional hackers. Stolen cards can be used like money to make purchases more often, hackers sell the, to other people on the black market. If the organization's website is accepting credit card payments then you must take special precautions and many of these precautions are not optional.

They’re required by the credit card company and the payment processors. PCI compliance is shorthand for following these data security standards. If the organization doesn’t do its part for following these data security standards then they don’t have to ensure your losses or shield you from any damages. They can sometimes simply let the organization stop accepting the credit card payments. The data security standards are not just one simple set of rules. They do have different requirements levels depending on the business.

The first thing is that the organization should transmit all payment information over SSL. The main reason is that it allows us to encrypt the information so that it’s not visible while it’s in transit. It also has added benefit of making sure that we’re sending the information to the person that we really think we’re sending it to, it verifies the authenticity of the receiver.

Another interesting idea that comes up in the last few years are credit card vaults. This is where the processor stores the card information for you, so you don’t have to keep the card number. You send it to them and they will store the card number and send you back a token. Whenever you need your card you can use the token as your reference.

When you sign up for a PayPal account, you type in the credit cards and bank accounts you wish to use for purchases one time on the PayPal secure system. Once that is accomplished, you never again should reveal your credit card information online according to (Niranjanamurthy, 2014), By paying with PayPal, you keep your credit card information confidential, meaning that you can shop without worrying about a hacker downloading your account data from your online merchant.

Summary

This chapter discussed the background details and literature review of problem and proposed system. Next chapter will describe the methodology to design the proposed system.

Chapter 3: System Analysis

The world runs on software. Applications control almost every aspect of our lives, ranging from the software that flies airplanes to applications that dispense money through ATMs. Software developers works hard every day to bring automation and integration to many aspects of our lives. Our increasing dependence on software makes it increasingly important that we use software that is known to be secure and reliable. Recent news items have underscored the importance of software security.

The details of software security vary quite a bit depending upon how organizations acquire their software. In many cases we purchase software from known vendors like Microsoft, adobe and oracle. In other cases, we develop our own software, customized to meet our specialized business needs. We have security responsibilities in either case. Application hardening is one of the core principles of software security. Cyber security experts must carefully test software to ensure that it is locked down as much as possible and safe against attacks.

Some of the key principles of application hardening are ensuring that applications use proper authentication to validate the identity of users, that applications encrypt any sensitive data so that attackers can’t read it by accessing the underlying storage directly ensuring that applications validate any user input to ensure that it does not contain dangerous code that might harm the security of the software or underlying computing infrastructure and ensuring that applications are not vulnerable to any known exploits.

And when vulnerabilities are discovered that they are promptly corrected. One of the ways that organizations correct software vulnerabilities is promptly applying security patches after they are released by software vendors. Developers of major applications frequently receive reports of security issues in their software and issue corrective patches designed to prevent future attackers from exploiting the vulnerability, once knowledge of vulnerability becomes public, organizations that till run the unpatched code are especially vulnerable to attack because the news is out there and attackers may actively seek out organizations that are slow to correct security problems.

Analysis helps ensure the stakeholders see and understand the complete and high-quality requirements. When we analyze requirements, we find the important things that were unknow or unsaid during stakeholder conversations. Analysis identifies connections, gaps, impacts and questions that have not yet been discussed. Without analysis teams end up with many misunderstandings which leads to solutions with lots of defects.

Analysis can happen with elicitation and almost always leads to more elicitation. Analysis techniques can also vary by level of formality from elaborate techniques with a variety of interactive steps to informal conversations. The techniques can be supported by anything from digital models to white board drawings. Stakeholders see the solution differently than engineers and developers do. Analysis should help them connect the dots by using models and diagrams to understand the interconnected branches.

Analysis can be a combination of intense thinking alone, or it can be collaborative helping the group of stakeholders analyze and inspire each other find gaps and impacts.

Existing System

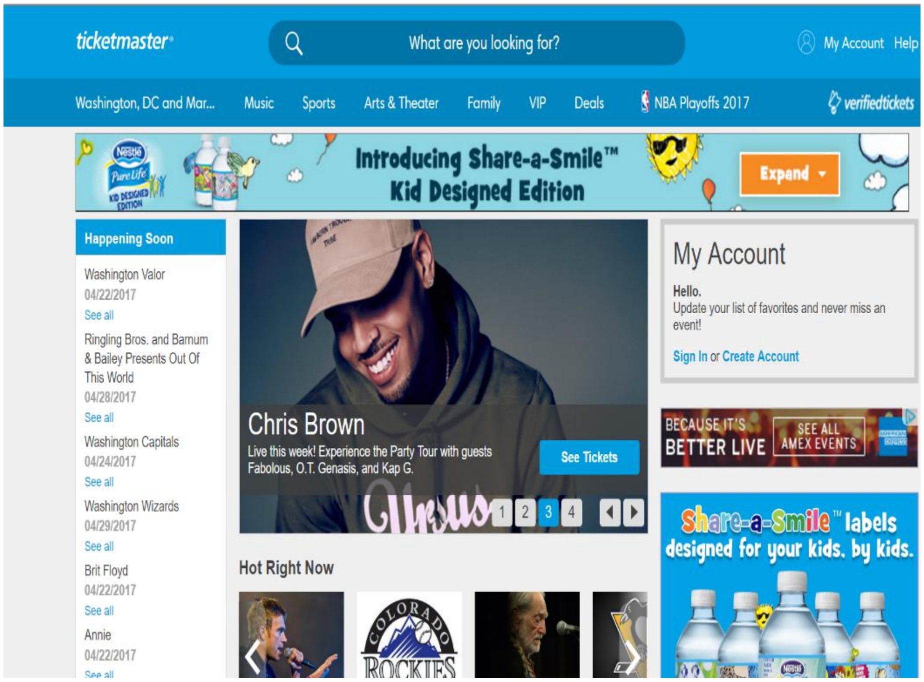
There are currently several different websites offering online tickets sales among which below mentioned three are the most well-known and are used for this research.

1. Ticket Master (www.ticketmaster.com), Figure 2.
2. Aloud (www.aloud.com), Figure 3.

3. Ticket Web (m.ticketweb.com), Figure 4.

Although these three are most popular and most used ticketing systems, each of these sites offers a very similar service to their customers while each have their unique way of navigating the clients and helping them search for required events in the site.

Ticket master and aloud keeps its customers updated with the links pages selling latest tickets on their home page to what they describe as ‘hot tickets’. region. The Advanced search allows user to search by region a date and by key words. Below are the images of homepages all these three websites and a brief description on how they navigate their users through the website to buy tickets.

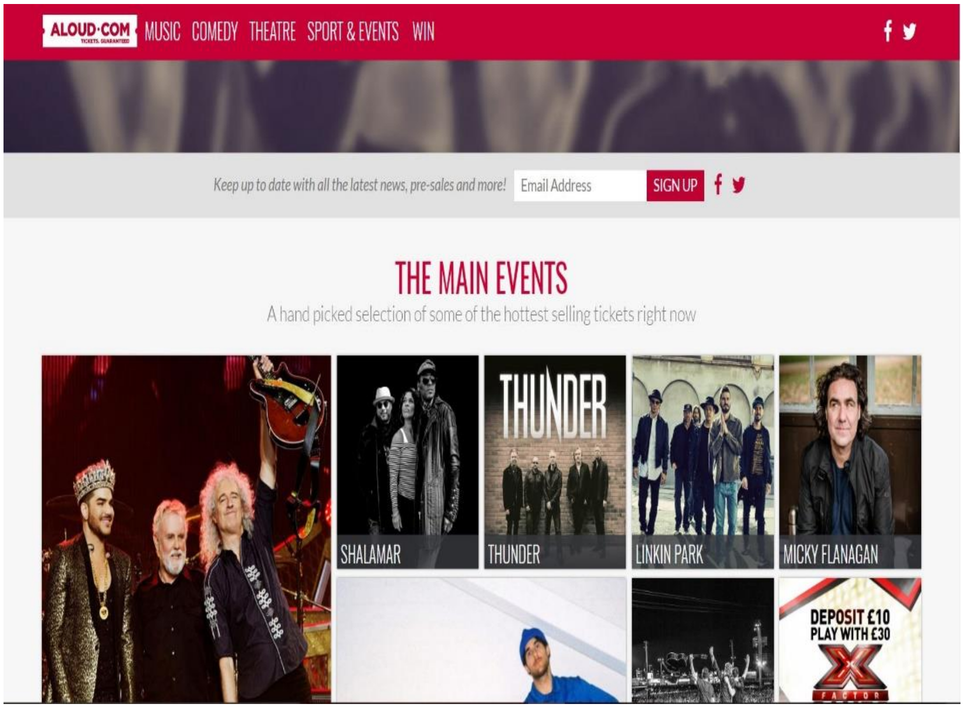


Live this week! Experience the Party Tour with guests Fabolous, O.T. Genasis, and Kap G.

Update your list of favorites and never miss an event!

Figure 3.1. Ticket Master Home Page (Ticket Master, n.d.)

To the left of the home page of Ticket master is a section ‘Happening soon’ to keep its customers updated with the upcoming events nearby their location. Filtering the events based on their location is one good feature ticket master provides as it makes the user task easy. The ticket master website offers accounts to customers so that their details are stored and the process of purchasing tickets is quicker and customers do not have to fill in forms. Customers who have accounts receive and added a list of favorites will receive emails periodically from the site to promote the upcoming events. The search on Ticket Master allows users to search by artist, team or venue in a chosen location. The advanced search allows users to search by artist or event, a set of dates, a venue name and a town.



Keep up to date with all the latest news, pre-sales and more!

THE MAIN EVENTS

A hand picked selection of some of the hottest selling tickets right now

Figure 3.2. Aloud.com Home Page (Aloud, n.d.)

Above image is the home page of aloud displaying the upcoming events and links to buy tickets. Aloud keeps its customers updated with the links pages selling latest tickets on their home page to what they describe as ‘hot tickets’. However, the Aloud website has a basic search comprising of artist/band or town. The aloud website offers customers an email list facility where users can enter their email address to be kept up to date with upcoming events according to (Bradley, 2006). Aloud also has a feature of sending out emails to customers who sign up periodically from the site, promoting upcoming events.

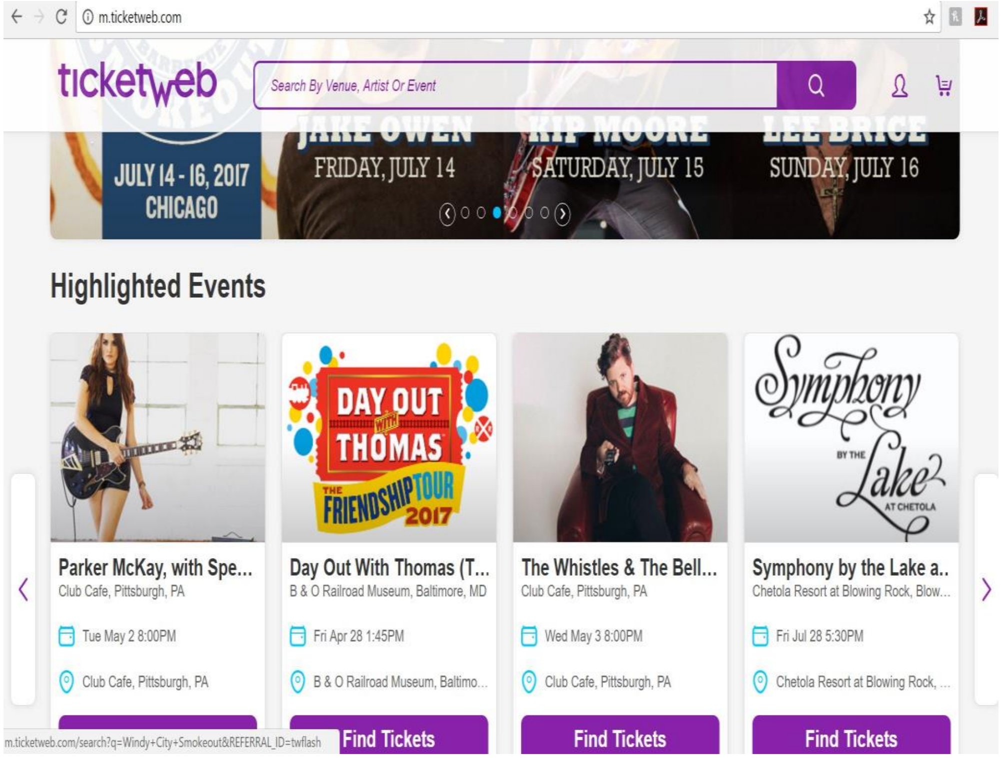

m.ticketweb.com
ticketweb
Search By Venue, Artist Or Event
JULY 14 - 16, 2017
CHICAGO
JAKE OWEN
FRIDAY, JULY 14
KIP MOORE
SATURDAY, JULY 15
LEE BRICE
SUNDAY, JULY 16
Highlighted Events
DAY OUT WITH THOMAS
THE FRIENDSHIP TOUR 2017
Symphony BY THE Lake AT CHETOLA
Parker McKay, with Spe...
Club Cafe, Pittsburgh, PA
Tue May 2 8:00PM
Club Cafe, Pittsburgh, PA
Day Out With Thomas (T...
B & O Railroad Museum, Baltimore, MD
Fri Apr 28 1:45PM
B & O Railroad Museum, Baltimo...
The Whistles & The Bell...
Club Cafe, Pittsburgh, PA
Wed May 3 8:00PM
Club Cafe, Pittsburgh, PA
Symphony by the Lake a..
Chetola Resort at Blowing Rock, Blow...
Fri Jul 28 5:30PM
Chetola Resort at Blowing Rock, ...
Find Tickets
Find Tickets
Find Tickets
m.ticketweb.com/search?q=Windy+City+Smokeout&REFERRAL_ID=twflash

Figure 3.3. Ticket Web Home Page (Ticketweb, n.d.)

The Ticket Web site home page requires the user to select an area or search for an artist or event. This is bit complex as in time taking as the user will be encountered with many events when attempted to search with respect to place if the user does not know the exact name of the event. The search facilities available on each of the sites are similar. The ticket web search basic search allows users to search for artists or events or browse events. User when in doubt will not have any immediate assistance so the next option will be to check another ticketing software.

To avoid the problem of losing customers who are unable to find required information, live chat can be an option to guide the customer with required information and it also assures them that there is someone available immediately in case they need any information, providing a more easier way to book tickets. Neither of the above-mentioned systems have an option to chat on live with their customers.

Proposed System

After analyzing existing system to see which of their features were useful and how they could be improved, there was also a small amount of research done into finding out what user of the existing systems felt could improve the service offered to them. From this research, the following requirements for this project were gathered to modify the existing system to some extent.

It was decided when the website was produced for this project will allow client to sign up to the site enabling them to purchase tickets quicker by limiting the amount of forms and further if the customer is in doubt or could not find information about event can utilize the option to chat live with one of the representatives from the organization instead of long awaited customer service.

The application was basically designed to create an event, allow the user to register, include a security driven payment system and send the ticket confirmation. Every industry including mobile operators and internet companies have adapted this easy-to-use e-commerce payment systems. This development has provided consumers with more effective, convenient, and secure ways of purchasing and expanding variety of products, including digital goods and services.

Design of Study

The methodology that will be used throughout the development process will be a variation on the waterfall life cycle. As the requirements for the project are unlikely to change

dramatically this methodology will fit the project. If the project looked like the requirements would be changing often a more agile methodology would have been chosen. The waterfall lifecycle works by following a defined path through the development process not moving on to the next stage until the previous
stage has been completed. The various stages of this project were:

Stage 1 of the project involved researching into existing ticketing systems, user expectations and then drawing up the requirements of the project. Once the requirements were gathered, technologies to be used to develop the system had been researched and decided upon those technologies that are familiar and easy to implement.

With all the gathered requirements and technologies being used to develop these requirements in to a working model will start at stage 2 i.e., designing the front end followed by the backend can been completed. Once the design stage is completed Stage 3, implementation can begin, although there will be no formal test-driven development for this project when new features are added or code is edited the system will be tested to ensure that no bugs have been introduced and the program runs as per the requirement.

After the implementation has been completed the entire system development will move forward to stage 4 where the entire application will be tested thoroughly to check if it was working correctly or else modify to make it work as required.

New users are required to register into system at the registration page. Existing users can login directly from the login page. After login user will be directed to the home where in he/she could select from various categories of events like

1. Athletics
2. Cultural

3. Entertainment
4. Recreation
5. Conferences and Workshops

User when selects one from these categories will be directed to the page displaying list of upcoming events under that category. Upon choosing one event, user would be asked if he/she would like to purchase a single ticket or ticket for a group. If single ticket, directs to the payment page where the user must select a level to which he/she belongs to from various levels. Various levels for the college events can be

1. Student
2. Staff/Faculty
3. Outsider

If purchasing for a group, user should enter the name of each person from group and the level to which that person belongs to.

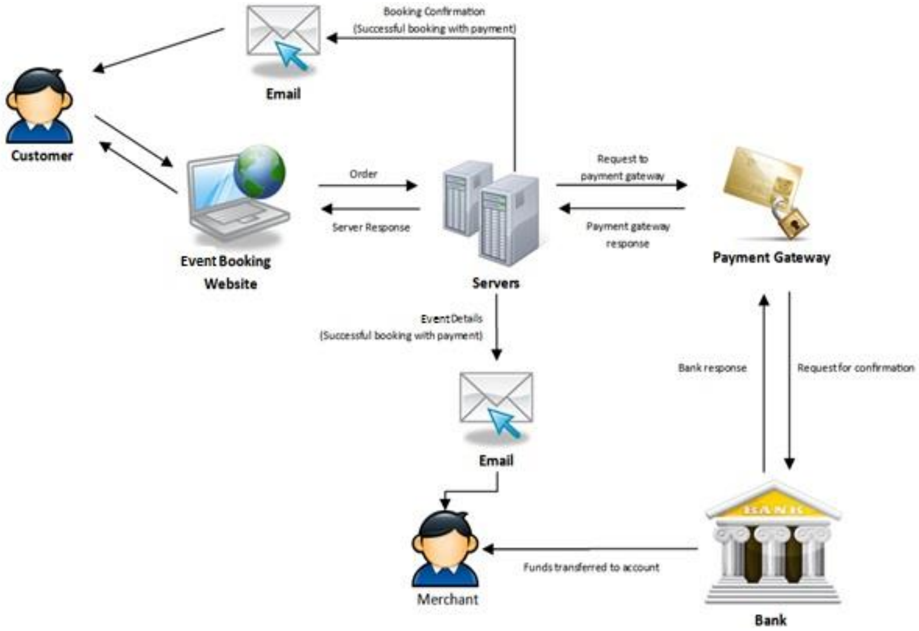


Figure 3.4. System Architecture

A computerized system that stores and distributes customer information for booking a seat in the event is called a reservation system. This reservation system can be either be web-based or non-web-based depending upon the business needs. This is a web- based application.

The above diagram (Figure 5) shows that customer accesses the event booking website checks for the interested events and adds the number of attendees and then when buy now button is clicked, all the user information is sent over the server. The server makes request to the PayPal payment gateway. The PayPal contacts the bank for confirmation and upon the bank's response, PayPal sends a confirmation reply to customer through the servers. After the verification, bank transfers the funds from the user account to the merchant. Server can be set up to send a confirmation email to the merchant. This application also allows the user to login to the system, and view the status of their booking.

The site is needed to split into two separate parts, the first for the clients to see. This part of the website will allow clients to sign up for the web site, search for concerts and purchase tickets for the concert. The other part of the website for administrators, will allow admin staff to add, edit and delete content. The two sites could be hosted on completely different servers as long as they can both connect to the same database. As it is, the admin part of the site is held in a sub directory within the client side of the site.

Starting with administrators. An admin could login, create an event, (with a name, event date, event time, place where the event is taking place, etc.). Each event would have several 'levels' upon which the ticket price varies. For suppose, if the college is hosting an event which would be limited to college students allowing admission only of college students, then there would be only single level called 'students' and eventually the price would also same for every student. At the same time, there may be some other events which can include participants from college students/faculty/staff/others. In this case, there would be various levels like ‘student’, ‘faculty’, ‘staff’, ‘others’ allowing the admin to assign different prices for various levels.

Users. User can login to the system, view profile details, update profile and change password frequently for more secured experience. Every user should first register in to the portal. Registered user can login to system and choose interested event from list of upcoming events hosted by the college. So, if a registrant is a student, he/she would buy the ticket from the student level, pay $XX which is assigned by the admin to this event. Other levels could be like an outsider would pay $YY and faculty/staff pay $ZZ. An event could have unlimited “levels” as I would call them.

Additionally, this application will allow an individual to book and buy tickets for entire group who would want to attend the event. There will be a group leader who could login to the system, create a group for XX event, put in how many participants at which levels, save it all, get the total amount to be paid, pay that amount online using credit/debit cards in a secured way. When the payment is successful, the confirmation of tickets purchased would be sent to the group leader with the names of the individuals in the group on it. This application not only allows the users to buy tickets online for entire group but also makes it easy to manage the number of attendees for each level.

Online finance system. This system needs to handle the financial transactions when customers buy tickets. For this, research was needed into online banking and payment systems and how the online ticket sales system could interact with the different systems available.

PayPal was the most recommended e-commerce online payment method was said by (Niranjanamurthy, 2014) was used for payment transactions in this system.

Unlike the other ticketing software, this system would not require any fees i.e., it allows the university to host events for free. Application developed for the college is more trustable when compared other third-party ticketing software.

Application patch management is a critical security control. Organizations also often have quite a bit of control over the configuration of application security settings. For example, our organization has a complex and a large number of data regarding the students, in such case the organization often makes configuration choices, such as type and scope of encryption used on disks containing this data, the users who will have access to this data and the authentication techniques they will use to connect, the cope of access authorized for each user who has access, and the security of the databases, servers, networks and other infrastructure supporting the application.

Configuring these settings is a complex undertaking and involves many different configuration parameters. One of the best ways that organizations can manage this difficult problem is using configuration baselines that allow a quick comparison between the current settings and the desired security profile. If the current settings deviate from security's standards baseline, administrators may then act to remediate the vulnerability and restore the application to its secure baseline.

Data Collection

Planning for data collection during the analysis phase is important because the project team knows exactly which hypothesis is to be tested, what data needs to be collected, how much data to collect, from whom the data should be collected? Planning for data collection in the analysis phase starts with the following questions.

1. What is the theory or hypothesis to be tested?

2. How much dats is neede to run the test and draw a valid conclusion?
3. Does the application need one data point or 100 Data points?
4. From which people should the data be collected?
5. For how long should the data be collected? A day, a week or a month?

The benefit in spending time and effort to plan for data collection cannot be explained enough. Data related to the study and results was collected from testing the application on different computers with various users.

Summary

This chapter discussed in detail the methodology to design the proposed system.

Chapter 4: System Design

Software engineers who set off on the development process without carefully-designed requirements are acting like the construction workers who begin building a home without a set of blueprints. The finished product is not likely to match the customer’s vision, and that there is a good chance that it will fall apart. Once they have the set of requirements in hand, developers may begin the process of creating software. Depending upon the organization’s approach and the details of a specific project, they may choose one of the several software-development methodologies.

The classic approach to software development is a methodology known as waterfall approach. This model, developed by Winston Royce back in 1970’s approaches software development as a linear process. It follows a rigid series of steps that begin with developing system requirements, move on to developing software requirements, then produce a preliminary design from those requirements that is used as a basis for detailed design. Once that design is completed, developers begin the coding and debugging process where they create the software.

When they finish coding, the software is tested rigorously, and if it passes those tests, it’s moved to operations and maintenance mode. This approach does allow for a movement back to an earlier step but only one phase at a time. for example, if the software fails the testing process, it moves back into coding and debugging before being submitted into additional testing. This process is very rigid and does not allow for many changes to the software while development is in progress.

For example, if a business unit identifies a desirable new feature halfway through the coding process, there’s no opportunity to modify the design. Because of this there aren’t many modern software-development shops that embrace the waterfall model. In 1980’s a new model was introduced called the spiral model, a software-development model designed to mitigate some of the disadvantages associated with the waterfall model. This model treats the software development as an iterative process that has four phases.

In the first phase developers determine objectives, alternatives and constraints. Then they move on to evaluating alternatives and identifying and resolving risks. From there, they develop and test the code and then they begin the planning phase for future development work. While this may sound like the waterfall model, the major difference is that developers move through this phases in an iterative fashion, following a spiral motion. They begin in the first phase and then they move on to other phases multiple times (or iterations) until they have a satisfactory product.

More recently developers around the world have come to embrace the agile approach to software development. This approach values rapidly moving to the creation of the software, and is quite popular. The creators of the agile approach authored a document called the agile manifesto that discusses their approach in detail. There are four core concepts that the writers of the manifesto come to value. They value individuals and interactions over processes and tools. They value working software over comprehensive documentation.

They value customer satisfaction over contract negotiation and they value responding to change over a fixed plan. They went on to write the 12 principles of the agile software. Agile developers follow these principles: highest priority is to satisfy the customers through early and continuous delivery of valuable software. They welcome changing requirements even late in development. “agile process harness change” for the customers competitive development.

These developers deliver working software frequently, from couple of weeks to a couple of months with a preference to a shorter timescale. Business people and developers must work together daily throughout the project. This methodology believes that the most efficient and effective method of conveying information to and within a development's team is face-to-face conversation. Working software is the primary method of the progress.

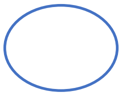
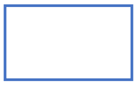
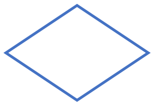
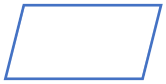
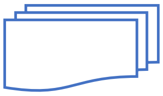
Agile processes promote sustainable development. The sponsors, developers and users should be able to maintain constant price indefinitely. Continuous attention to technical excellence and good design enhances agility. Simplicity is essential. The best architectures, requirements and designs, emerge from self-organizing teams. Finally, at regular intervals the team reflects on how to become more effective then tunes and adjusts its behavior accordingly. Every organization needs to select software-development methodologies that are appropriate to their own environment and comfortable for business leaders and developers.

Unified Modeling Language (UML) Diagram

Flowcharts are a great way to create a visual representation of our program flow. A UML isn't a programming language. This is a graphical notation specifically for drawing diagrams of an object-oriented system.UML Diagrams also provide a visual of the classes we need to create. Creating flowcharts forces us to break the program into pieces. This process is also known as stepwise refinement. Here are the basic symbols used for creating the UML Diagrams of this application

Flow chart symbols.

Table 2. Flow Chart Symbols

Start/End	
Process	
Decision	
Data	
Files	

The Unified Modelling Language permits the software developer to express a pictorial representation utilizing the standard modeling representations that has set syntax and rules. A UML framework is spoken to utilizing five unique perspectives that portray the framework from a point of view. Each view is characterized by a diagram.

There are web-based programming tools that can also help build UML diagrams like gliffy.com and creately.com but if one is looking for bit more than just a drawing tool, can also have tools that are more programming tools that will even help you generate code from your UML diagram. These diagrams are much more prevalent in the Java, C# and VB.NET worlds. Some programming IDEs like Visual Studio have UML tools built-in to some additions and with other IDEs like eclipse they are plugins available.

Use-Case Diagram

A use case diagram in the Unified Modeling Language (UML) is a sort of behavioral graph characterized by and made from a Use-case investigation. Its motivation is to display a graphical outline of the usefulness gave by a system as far as on-screen characters, their objectives (described as utilization cases), and any conditions between those utilization cases. The principle motivation behind utilization case graph is to show what system capacities are performed for which on-screen character. Parts of the performers in the system can be delineated.

Use cases can be written at several levels of formality, but at the most essential; they need three things: a little that describes the goal we're looking for, the person who wants that goal referred to as the actor, and the steps needed to accomplish the goal because it is not a description of an isolated feature but a description of interaction in full context, a complete step-by-step encounter which could have multiple possible results.

Examples here might be register a new member, transfer funds, purchase tickets and so on. These could all be separate use cases, separate but distinct goals of application. And then we have the actor. Now the reason I say actor rather than generic user is because we often need to identify exactly who is having this interaction. It could be a user but also a customer, member or administrator.

This application has users in different roles and each user would have different goals to describe for each of these roles and it doesn't have to be a human being. It could be another computer system interacting with the application. So any external entity that acts on our system is an actor.

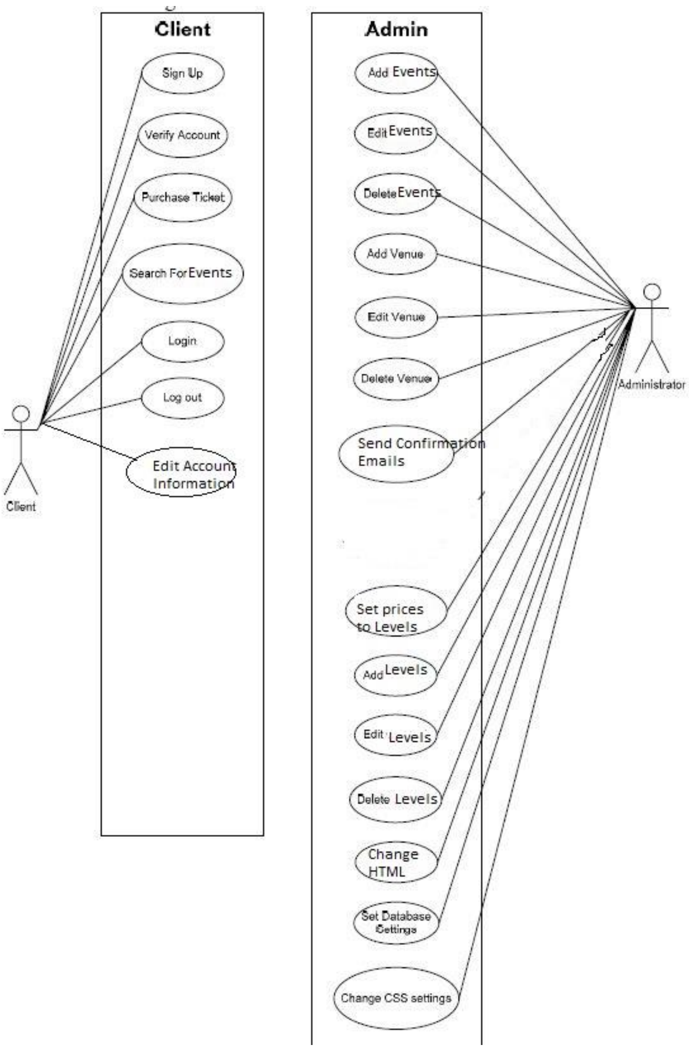


Figure 4.1. Client and Admin Use-case Diagram

The above use-case diagram shows the different functions client/admin can perform on this application. Client can

1. Signup
2. Login
3. Search tickets
4. Purchase tickets

5. Edit account information

Admin.

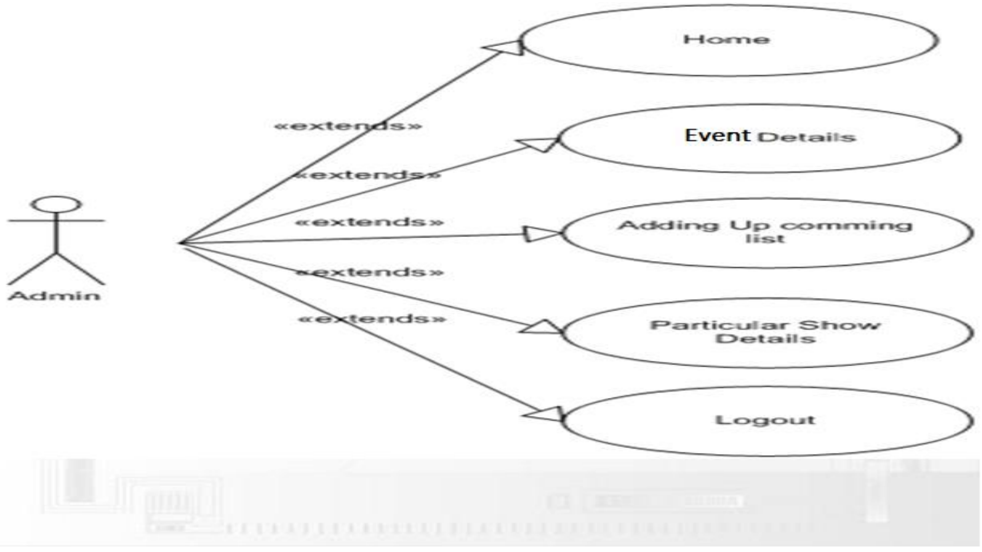


Figure 4.2. Admin Use-case Diagram

Responsibilities of admin include:

1. Add/Edit/delete events
2. Add/edit/delete venue
3. Add/edit/delete price levels

Typically, in a use case diagram, we put the primary actors on the left-hand side, these are the one who initiate any of the use cases and the secondary actors on the right-hand side, and these actors take more of a reactive role.

Now this diagram is not referencing to sequence, there is no necessary to order it. Sometimes people write the use cases in order from the top down if the application naturally flows that way, but vary often the use cases themselves don't have a built-in sequence. This use case diagram is a simple overview of multiple use cases and multiple actors at the same time without the details of each particular written use case. This use case diagram can sometimes be used by the business users to figure out if something is missing from the picture.

Data-Flow Diagram

We often encounter with a common problem of entering the address three times when shopping online i.e., we enter the address on the screen when we create our profile or start a purchase and this address doesn't show up later in the process for shipping,billing,confirmation page and receipt. Usually this is referred to as data flow in the requirements

Data flow analysis is a critical tool for generating high quality requirements. Without this analysis, customer satisfaction drops and defects rise. Users get frustrated when they don't see the data they expect, and are asked to re-enter data they've already provided

Data flow diagram is a graphical representation of the data flow in the applicaion. This was first step followed to get an overview of the system. Developing this diagram was extremely helpful as it gave a clear understanding of the functionality and the how the data travel from one module to another. Data flow diagrams show the inside of a system and data flow is the primary focus.

Basic components of the data flow diagram are:

1. External entities represented by : ▭
2. Data flow represented by: ⇨
3. Data Process is represented by : ○
4. Multiple actions are represented by : ◇

Below are the data flow diagrams of user and admin.

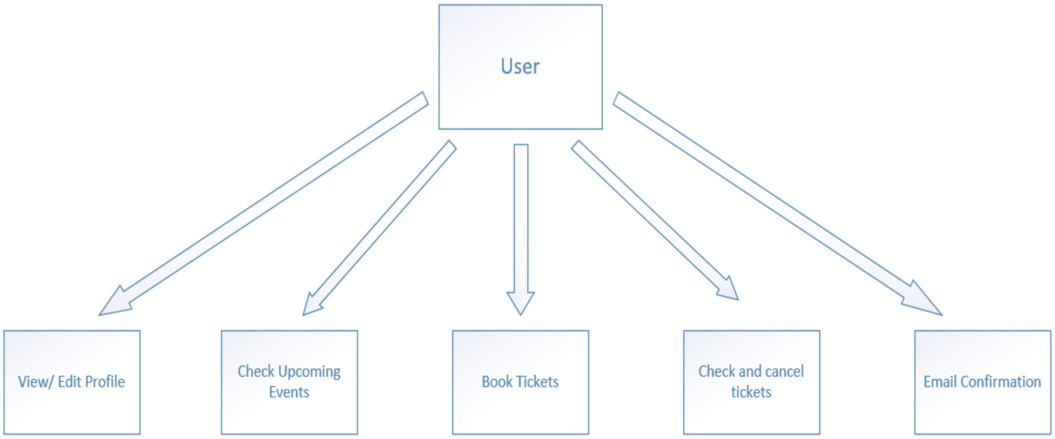


Figure 4.3. Client Data Flow Diagram

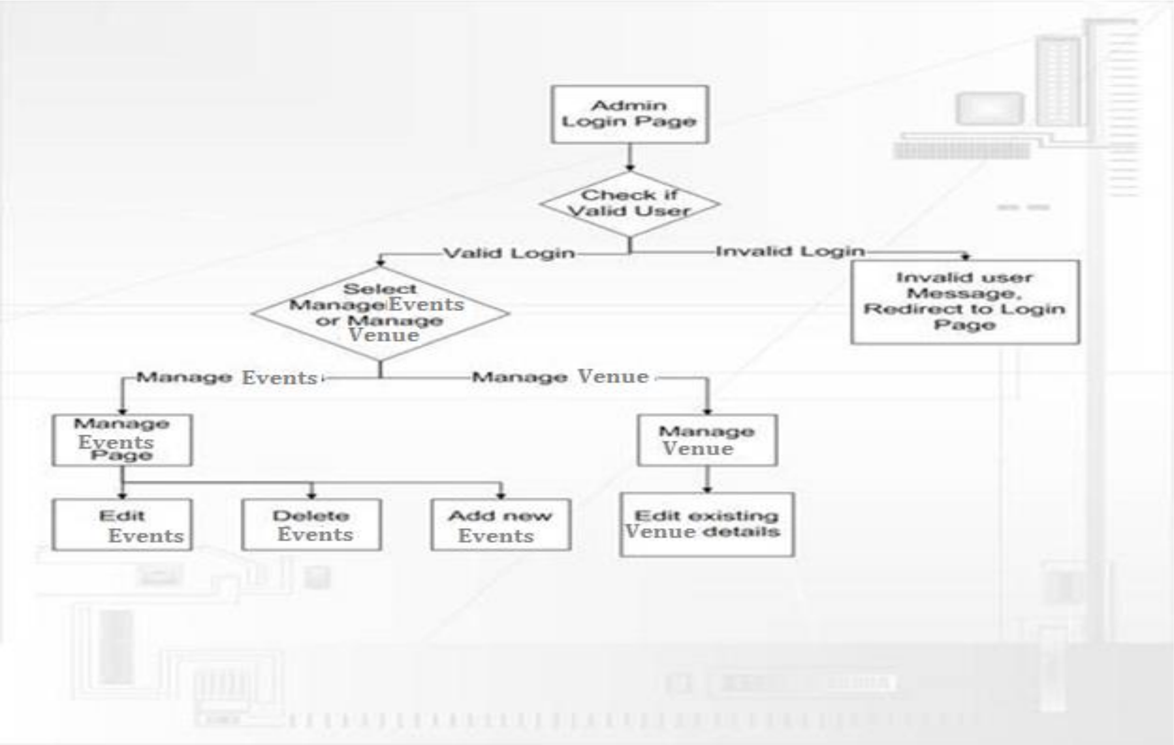


Figure 4.4. Admin Data Flow Diagram

Sequence Diagram

These are what are considered structural or static diagrams. They are great representing the overview of the classes in your application and seeing inheritance and composition or the actors in a system. But they are not so great at representing, say, the lifetime of an object or how objects interact with one another. So there are also Behavioral or Dynamic diagrams in UML. And these can describe how different objects change and how they communicate with each other.

And the most common one is the sequence diagram. Now a sequence diagram does not describe the entire system just one part of it, one particular interaction between a few objects in one scenario. We start a sequence diagram with some boxes at the top that represent the objects, the participants in this sequence could be two or three or even more. Sequence diagram is used to understand the interaction between objects and shows the scope or the lifetime of an object. The message flow is shown vertically in a waterfall manner i.e., from top to bottom and dashed lines represent the duration for which the object will live. Messages sent from the object are represented by a dark arrow and dark arrow head.

Sequence diagrams offer an overview of the important parts of this process not to try and model every if statement or while loop. Now one of the benefits of sequence diagram is that at this level of interaction you should be able to sit down with a business user, someone who is not a developer and explain this general process and they hopefully give you some ideas whether that would be correct or whether something else is missing out of this. This is a thinking tool helps the developer think through the process.

One can create several sequence diagrams to help you understand the specific scenarios, but should be aware that there is no need to try and diagram every single part of your application.

Sequence diagrams are for sketching out a situation that’s not completely clear and this diagram do a great job in making the situation clear and often result in realizing that a new class needs to be created somewhere.

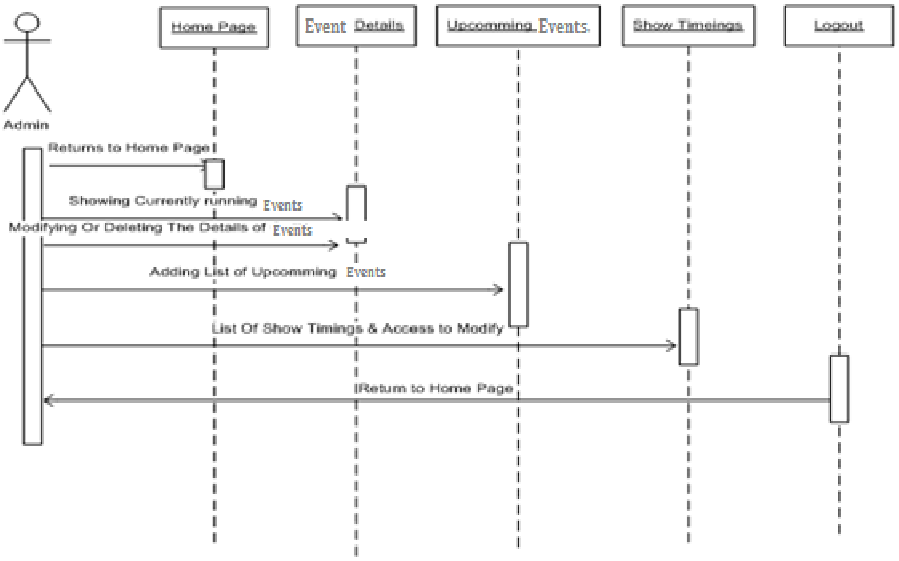

Admin
Home Page
Event Details
Upcomming Events
Show Timeings
Logout
Returns to Home Page
Showing Currently running Events
Modifying Or Deleting The Details of Events
Adding List of Upcomming Events
List Of Show Timings & Access to Modify
Return to Home Page

Figure 4.5. Sequence Diagram

Chapter 5: Software Requirement

Developers work together with the customers to outline the specific purpose of the software and the details of business goals it will achieve. This process is known as requirements definition and is crucial in developing software that meets the organization's needs. After developing the business requirements, developers then move on and translate those requirements into technical design. This is where the technical experts layout the map for software development and determine the interfaces between software components that will ensure everything fits together properly in the end. Software development is a sophisticated engineering process that is every bit is as complex as a major construction project.

.NET Environment

Before any programming could be done decisions needed to be made on which technologies should be used to program the project with. The following is a critical evaluation of the different programming languages that could be used to program such an online application.

This system was developed in .NET framework using C#, ASP.NET, ADO.NET, SQL Server. N-Tier architecture to ensure the availability, integrity and confidentiality of the data and PayPal – a secured payment gateway to assure the users that their credit/debit card will be confidential and flexibility Visa/MasterCard/Discover/American Express cards for their payments were used.

.NET framework. The .NET Framework has two main components: the CLR and the .NET Framework class library. The common language runtime is the foundation of the .NET Framework. One can think of the runtime as an agent that manages code at execution time, providing core services such as memory management, thread management, and remoting, while also enforcing strict type safety and other forms of code accuracy that ensure security and robustness. Managing the code is a basic functionality of the runtime. Code that objectives the runtime is known as managed code, while code that does not focus on the runtime is known as unmanaged code. According to (IT Pro Today, n.d.) the class library, the other fundamental segment of the .NET Framework, is an extensive, object-oriented collection of reusable sorts that you can use to create applications running from command line or graphical UI (GUI) applications to applications like the most recent developments gave by ASP.NET, for example, Web Forms and XML Web administrations.

Features of the CLR. The common language runtime manages memory, thread execution, code execution, code safety verification, compilation, and other system services. These features are intrinsic to the managed code that runs on the common language runtime says (My Code Lines, n.d.))

With regards to security, managed components are awarded varying degrees of trust, depending on a few factors that include their origin (such as the Internet, enterprise network, or local computer). This means that a managed component might or might not be able to perform file-access operations, registry-access operations, or other sensitive functions, even if it is being used in the same active application.

The runtime enforces code access security. For example, users can trust that an executable embedded in a Web page can play an animation on screen or sing a song, but cannot access their personal data, file system, or network. The security features of the runtime thus enable legitimate Internet-deployed software to be exceptionally feature rich. Finally, the runtime can be hosted by high-performance, server-side applications, such as Microsoft SQL Server™ and Internet Information Services (IIS) (retrieved from: (microsoft, n.d.)

Application development. Web Application is a piece of software that can be accessed from a browser. Web application can be called as a web server. Although there is a difference that web server can host multiple web applications. Web server is a network application running on some machine tistening to a port. The browser is called a web client or more formally we call the browser as the user. This application will be developed in .NET Framework using N-Tier Architecture and the browser used for accessing this web application is Google Chrome.

Cascading Style Sheets

Cascading Style Sheets (CSS) gives the diverse ways it could be used to control both the layout and style of the site. Experiments were conducted with using tags and CSS to control layout, as well as using tables to control layout with some CSS to format things like alignment and color. Both approaches give web developers a lot of control of site layout. However pure CSS layout allows the designer to separate style and layout from content, whereas table based layout does not allow for complete separation of the two. This means that if a site uses pure CSS layout it could look completely different as elements of the site can be moved to any part of the page unlike tables that have a more solid construction and do not allow for this.

N-Tier Architecture

In any software development, it is important that the developer can bring all the designed software componenets to working together. Firstly, what is the difference between tier and a layer? Tier usually means a physical deployment computer whereas layer means logical software component or logical separation of the code which may not run in the same process or same machine. Applications built on N-Tier Architecture can easily implement the concepts of Distributed Application Design and Architecture. The N-Tier Applications provide strategic benefits to Enterprise Solutions (retrieved from (N tier , n.d.)).

1. Improved Security because of Data Abstraction
2. System Availability
3. Makes the Application Management easy
4. Code Maintenance is not complex

Simply stated, an N-Tier application helps us distribute the overall functionality into various tiers or layers:

1. Presentation Layer
2. Business Rules Layer
3. Data Access Layer
4. Database/Data Store

Each layer can be developed independent of the other if it adheres to the standards and communicates with the other layers as per the specifications.

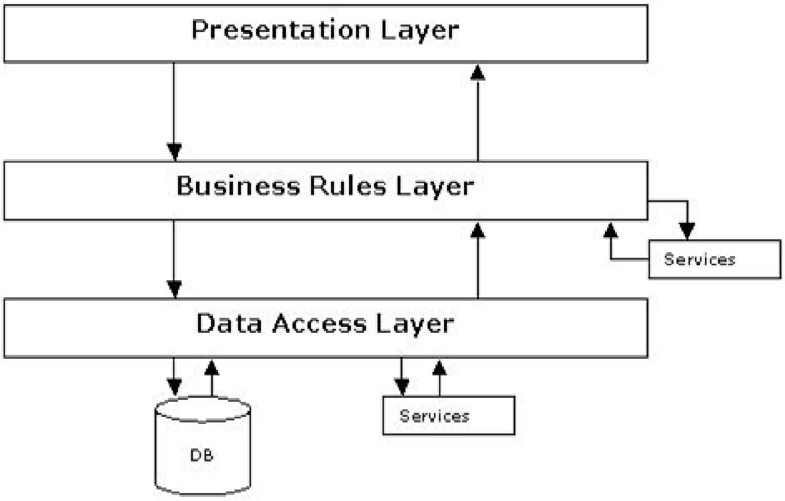


Figure 5.1. N-Tier Architecture

ADO.NET

ADO.NET uses a multilayered architecture that revolves around a few key concepts, such as Connection, Command, and Dataset objects. However, the ADO.NET architecture is quite a bit different from classic ADO.

ADO.NET data providers. A data provider is a set of ADO.NET classes that allows you to access a specific database, execute SQL commands, and retrieve data. Essentially, a data provider is a bridge between your application and a data source (retrieved from: (microsoft, n.d.)).

The classes that make up a data provider include the following:

Connection: object was used to establish a connection to a data source.

Command: was used to execute SQL commands and stored procedures.

Data Reader: This object was used to read the data from the data source, and forwards access to the data retrieved from a query.

Data Adapter: This object was used when two tasks, first to fill a Dataset (a disconnected collection of tables and relationships) with information extracted from a data source. Second, to apply changes to a data source, according to the modifications you've made in a Dataset were required to perform.

ADO.NET doesn't include generic data provider objects. Instead, it includes different data providers specifically designed for different types of data sources. Each data provider has a specific implementation of the Connection, Command, DataReader and DataAdapter classes that's optimized for a specific RDBMS (Relational database management system). This application is designed and developed on SQL Server database, therefore a connection class named SQL Connection was used.

SQL

In 1970, Dr. E.F. Codd published a paper called A relational model of data for Large, Shared Databanks. This has become the standard model for relational database management systems. Thereafter, IBM developed the structured Query languade, originally called SEQUEL, later shortened to SQL. Over the years the standards have been updated and the standard names reflect that with the most recent SQL 2011. T-SQL is the dialect that Microsoft has implemented for SQL Server product.

SQL is a language for relational database. SQL is a non-procedural i.e., when we use SQL we specify the task that should be done instead of the way of pursuing the task.

Basic SQL Commands

Data Definition Language commands (DDL)

Data Manipulation Language commands (DML)

Transaction Control Language commands (TCL) and Data Control Language commands (DCL) were used to query the data from SQL Server Database.

Hardware and Software Environment

Table 3. Hardware Requirements

Hard Disk	80 GB
Floppy Drive	1.44 Mb
Monitor	13’ Colour Monitor
Mouse	Optical Mouse
RAM	512 Mb

Table 4. Software Requirements

Operating system	Windows 10
Application server	ASP.NET
Web server	IIS
Coding Language	C# .NET
IDE	Visual Studio 2015
Web technologies	JavaScript, CSS, HTML, Bootstrap
Data Base	My SQL Server 2012
Payment Gateway	PayPal

Below mentioned are few other application servers that can be used for developing dynamic web applications

1. Microsoft Active server pages - This is an older version of Microsoft's active server technology.
2. Adobe ColdFusion – across operating system product. This works on many operating systems including windows, MAC, Linux and then uses its own proprietary markup language called ColdFusion Markup language.
3. PHP – completely free and an open source product available on many operating systems
4. Java based servers (J2EE) – there are many servers for the Java developers including
    i. IBM WebSphere
    ii. ORACLE WebLogic (formerly BEA WebLogic)
    iii. JBOSS – a completely free J2EE

Why ASP.NET? ASP.NET is one of the members of the family packages known as application servers. An application server is a piece of software that is installed on a web server to create a dynamic application platform. The world wide web includes many clients and servers connected over the internet. Clients and servers communicate with each other using a protocol TCP/IP. Computers communicate with each other through HTTP. This type of communication is used for both static web pages – web pages that transmit only fixed web pages – and dynamic web applications such as those that are supported by ASP.NET.

A client starts a process by sending HTTP request for web resource which may be a web form, image file, JavaScript, CSS file. Server responds by sending back the resource that has been requested regardless of static or a dynamic web application. Client uses a software called the web browser. Personal computers such as those running on windows, MAC use web browsers google chrome, Firefox, Microsoft edge, internet explorer, safari. There are many other clients including mobile phones, tablets and many other devices that runs the web server. Web server is also a part of software that runs the entire software on HTTP server. There are two major server products on the market.

1. Microsoft's IIS (Windows only)
2. Apache – free HTTP server product developed and founded by Apache foundation runs on all operating systems.

This application was developed using ASP.NET is deployed on the IIS because only IIS does a good job of integrating the ASP.NET application server.

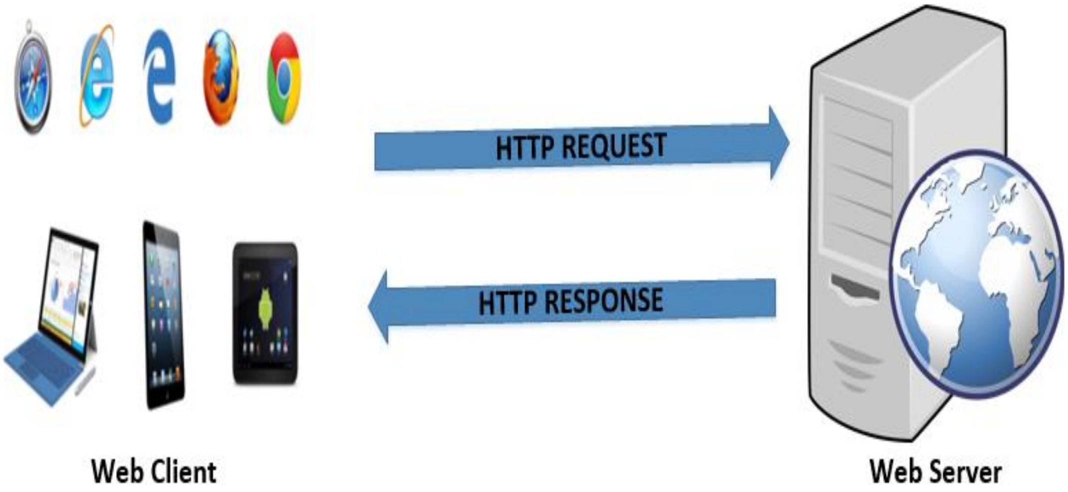


Figure 5.2. Static Web Page

This diagram shows how a static web page works. The web client makes a request for a resource (the resource here is a web page) by sending an HTTP formatted request over the internet to the web server. Server responds by locating the requested file stored on its hard disk. This is known as the static web page with a file extension of .htm or .html. The web server responds by that content back to the client and the client which is the web brower on the clients computer renders and displays the web page visually.

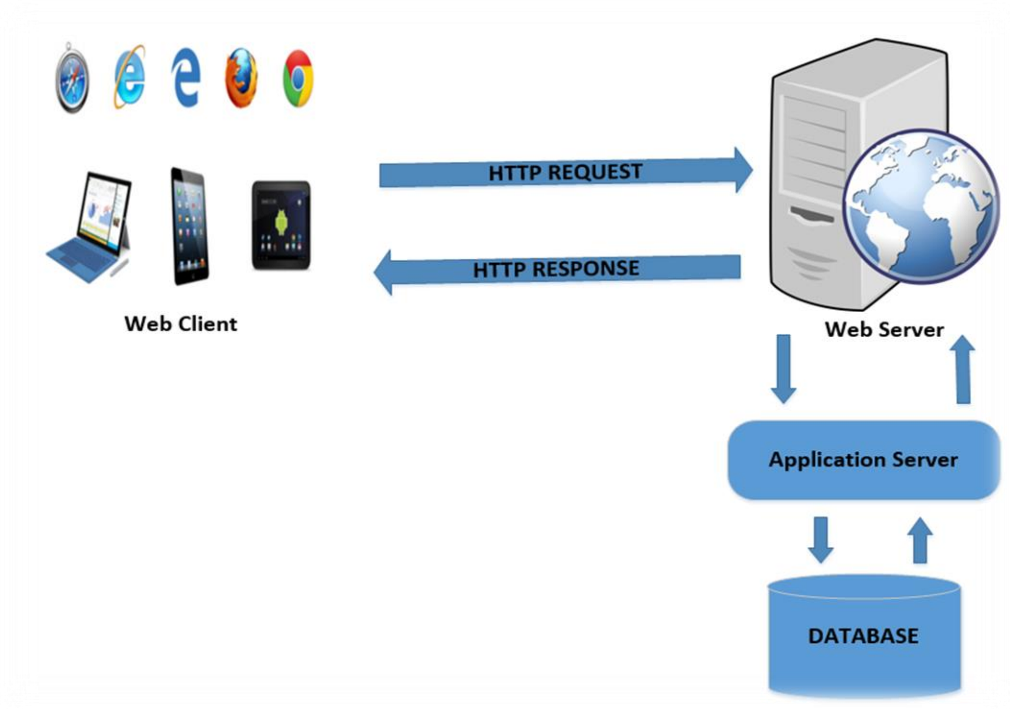


Figure 5.3. Dynamic Web Page

Figure 13 shows the working of a dynamic web page. In the dynamic web application, the application server is installed on the same computer as the web server. For example, the ASP.NET application and the IIS. The application server in this case is the ASP.NET can communicate with the database and with the other server resources. Now when the web client makes the HTTP formatted request to web server, the web server talks with application server, application server talks with database and other resources and then its upto the web server to construct a html response to send back to the web client. The interesting thing here is, the web browser does not know that there is an application server at work and therefore doesn’t need any additional software. The web browser makes the request in HTML page and receives the response in HTML page and also it doesn’t know the difference between the static web page retrieved from hard disk and a dynamic web page generated based on the request. There are many softwares one can select from as a application server. ASP.NET is choosen to develop this application because of the following advantages

1. Flexibility to look at in any browser
2. Reduces the lines of code in large applications
3. Merely a server side technology which means that the code is executed on the server side before it is sent to the web browser
4. High security because the code or the business logic is not visible on the browser
5. Supports dynamically adding/editing content of an online page

ASP.NET is an object-oriented platform that allows a developer to select from a multiple programming languages to create a dynamic web-based application where the client makes an HTTP formatted request and server responds by constructing HTML dynamically and sending that content back to the client. The dynamic web pages can include Javascript code, CSS, image files and everything else that could be included in a static web page.

Why T-SQL?

T-SQL is known as a set based and declarative language. This is contrast to programming languages that are considered to be procedural. In other words, T-SQL describes what you want, not how to get it. The database engine handles the necessary processing on the back end to determine where to pull the data from. T-SQL uses discreet groupd of commands. They are known as DML, DDL, DCL.

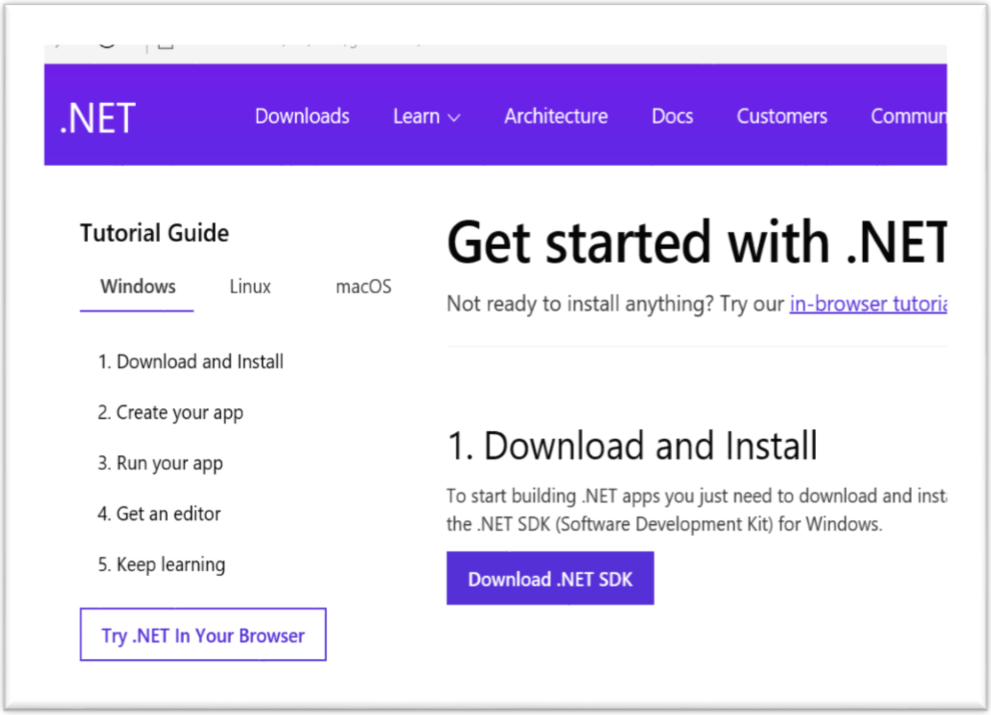
The DML components are known as the Data Manipulation Language and they are used for querying and modifying data. DML statements are used in this research to query the data from SQL Server. DDL components are used for creating and modifying the database objects such as tables, views and other compnents. Finally, DCL is used to implement the security on the database such as adding users, granting and revoking permissions. T-SQL used for developing the dataase of this application is derived from and extends ANSI and ISO SQL standards.

Chapter 6: Implementation

This chapter shows the installation of ASP.NET and other tools required for the developing this application. To start with let’s see what are the elements of a web development stack. Below are the libraries and functions that provide the base on which this application is built.

1. Application Framework
2. Web Server
3. Database
4. Development Tools

First require element is application framework and the one used in this application is ASP.NET. Web server which is the second element is a software that runs the application and delivers the HTML to the end user’s browser. In this application, IIS is the web server. Next element is the database which is used to store all the information and data that this application will use. SQL Server is used as the database in this application. Finally, the development tools that are used to writ, debug, test and deploy the application. The IDE used in this application is the visual studio. All these tools are obtained at once with something called the software development kit (SDK). SDK and the whole web development stack is downloaded from the ASP.NET website. Below figure 14 shows the asp.net website.



Get started with .NET

Not ready to install anything? Try our in-browser tutori

1. Download and Install

To start building .NET apps you just need to download and inst the .NET SDK (Software Development Kit) for Windows.

Figure 6.1. .NET Website

Click on the download button. All the required software tools is installed and prompts to run the application. Provide the admin username and password when asked to make changes to the system. Also, a couple of other dialogue boxes that I had to click allow to provide permission to the .NET SDK to run all the required tools. While installing a SQL Server, a window pops up asking which mode it has be installed. The options available are windows integrated authentication and the mixed mode authentication. SQL Server for this application was installed on Mixed mode authentication which requires user to type in a password and this password should be remembered as we require to enter the password in whenever the SQL server is opened. Clicking on continue will install all the applications required for developing this application.

Installing Visual Studio Express 2013

This section of the chapter explains how to download and install a free edition of Visual Studio. But before that what an IDE is? An integrated development environment like visual studio, aims to provide to us code as developers, a platform to develop, edit, run the code. Most likely, the bulk of the time worked in Visual Studio, was spent editing code. ASP.NET development could also be done with note pad or a command line compiler. But the lack of color coded text editor I think will be more helpful for identifying the errors and increasing the productivity.

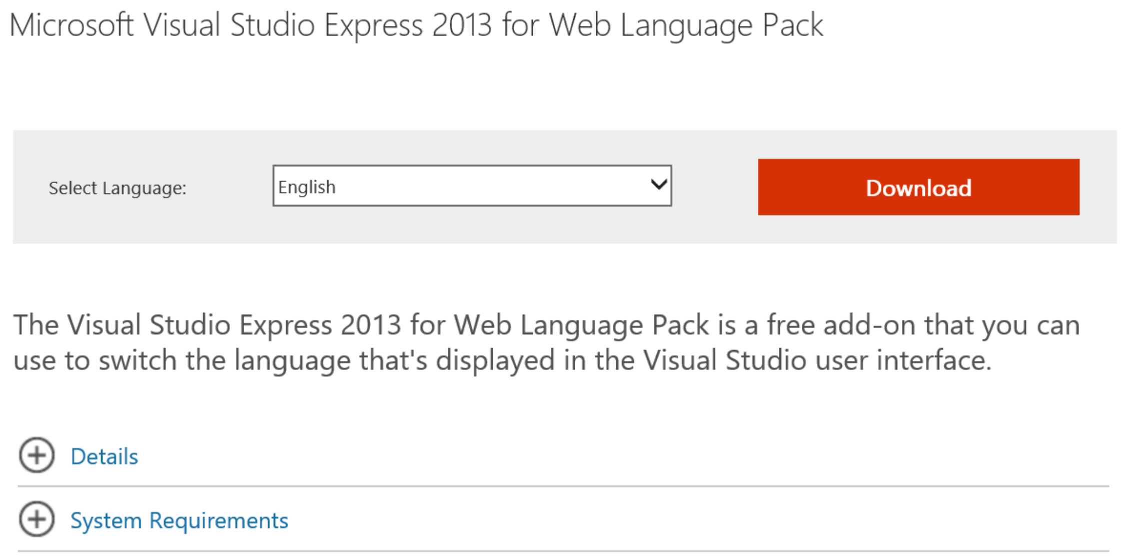
In addition to the text editor, a good IDE should also help navigate code and assist in completing statements and identifying typos and syntax errors. It’s also nice if compilers and build tools are bundled into the IDE. Allowing to make changes to the code, check to see if they compile and even publish projects all from within the same application. Good debugging tools are essential to developer productivity.

Sadly, a lot of developers still work without them today. For a long time, they didn’t even exist for things like JavaScript and CSS. But if debugging your application line by line, watching the values of the variables makes the work of detecting where the error is, a lot easier. To develop this application one of the express versions of visual studio was used. All of which are completely free and has great features. One of the drawbacks to the express editions is that they don’t support third party plugins or extensions.

A lot of developers complain that they really can’t live without some of these plugins. Another difference between the express editions of visual studio and the paid for versions is that the express versions each target a single platform only. So, one can’t use one application to develop for the web, desktop and the phone. This feature can be achieved when you purchase a different premium version of Visual Studio.

Finally, there are many other advanced features for debugging, testing and analyzing the application that aren't present in the express versions. But which many developers and small teams, seem to get by just fine without.

To install the visual studio, search Visual Studio Express 2013 for web. Below figure 15 shows how the web page looks.



Microsoft Visual Studio Express 2013 for Web Language Pack

Select Language: English

The Visual Studio Express 2013 for Web Language Pack is a free add-on that you can use to switch the language that's displayed in the Visual Studio user interface.

Figure 6.2. Visual Studio Express Download Website

Click on the download button and then I select the vns_full.exe because then it doesn't unpack or mount an image file, and then click on next. And then I chose the run option and agree to the licensing terms and privacy policy and then click install. It takes approximately 15 to 20 minutes to install the required software.

Including among other items, this version of .net framework, a development version of SQL Server and a light weight development web server that is used is called the IIS express.

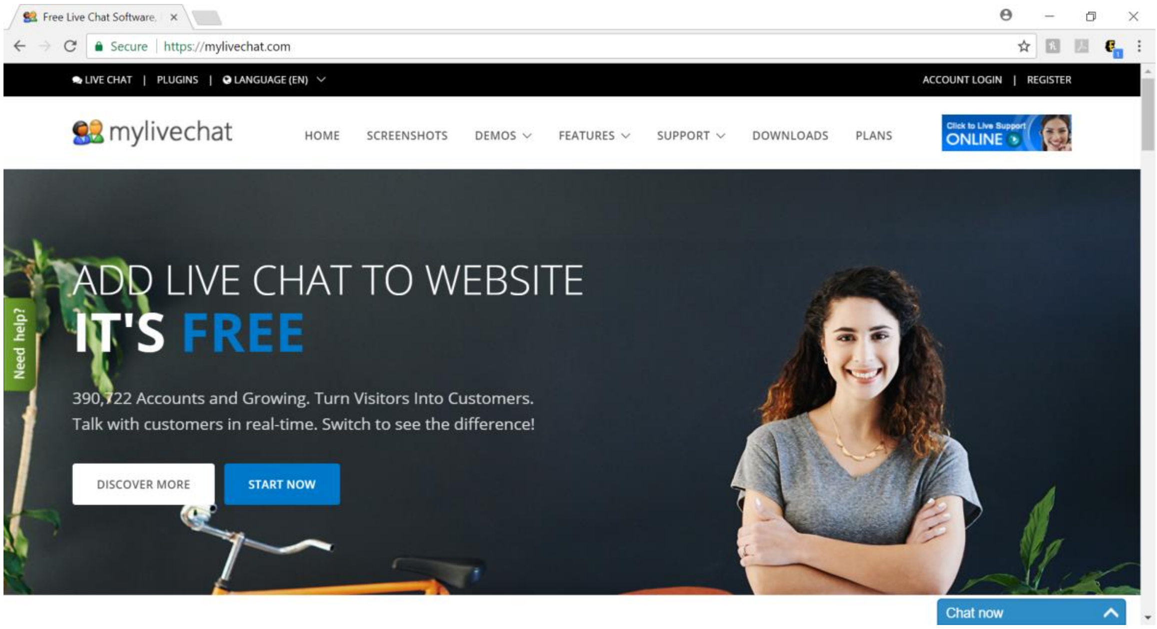
When the setup completes go ahead and click launch. And that's all one needs to do, to get ready to start building the ASP.NET applications.

Live Chat

This application has an additional feature of live chat in the form of a live help tool for the visitors to explore the SCSU tickets website. MyliveChat is a free live chat software used in this website.

MyliveChat is a customer service tool used while developing this application and this enables visitors to instantaneously communicate with the college personnel. Visitors to your website simply click a chat now button on the bottom right corner on the contact us page of the application and then immediate available college admin will respond. Admin representative can assist the visitor by providing information, links, graphics, or even guiding them through the SCSU ticket web site.

As per the (My Live Chat, n.d.) My Live Chat solution is provided on an Application Service Provider (ASP) basis. This means there is no hardware to install and you can get started instantly.

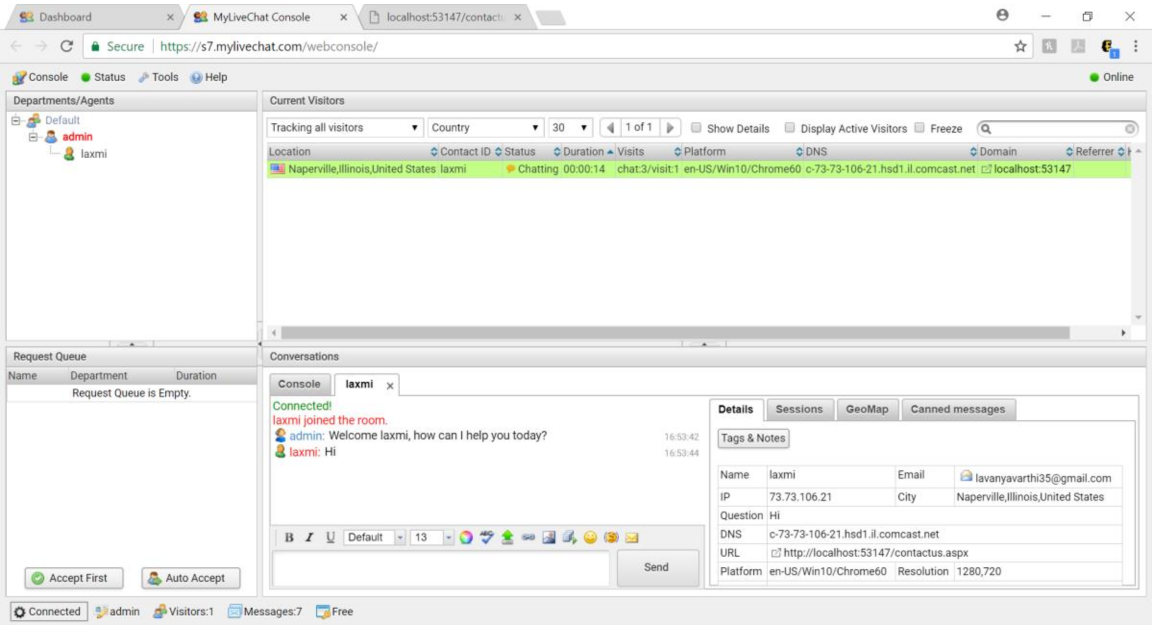


ADD LIVE CHAT TO WEBSITE
IT'S FREE

390,722 Accounts and Growing. Turn Visitors Into Customers.
Talk with customers in real-time. Switch to see the difference!

Figure 6.3. Live Chat Website

MyliveChat is the application integrated with this project to provide users an option to chat live with admin.



Location	Contact ID	Status	Duration	Visits	Platform	DNS	Domain	Referrer
Naperville,Illinois,United States	laxmi	Chatting	00:00:14	chat:3/visit:1	en-US/Win10/Chrome60	c-73-73-106-21.hsd1.il.comcast.net	localhost:53147	

Name	Department	Duration
	Request Queue is Empty.	

Connected!
laxmi joined the room.
admin: Welcome laxmi, how can I help you today? 16:53:42
laxmi: Hi 16:53:44

Name	laxmi	Email	lavanyavarthi35@gmail.com
IP	73.73.106.21	City	Naperville,Illinois,United States
Question	Hi		
DNS	c-73-73-106-21.hsd1.il.comcast.net		
URL	http://localhost:53147/contactus.aspx		
Platform	en-US/Win10/Chrome60	Resolution	1280,720

Figure 6.4. Admin Chat Page

Payment Gateway – PayPal

PayPal is a convenient money transfer service that’s faster than cash, check and money orders. It’s more secure than using debit and credit cards alone. PayPal accounts come in different types and one can fund PayPal balance from the bank account, a credit card or a debit card.

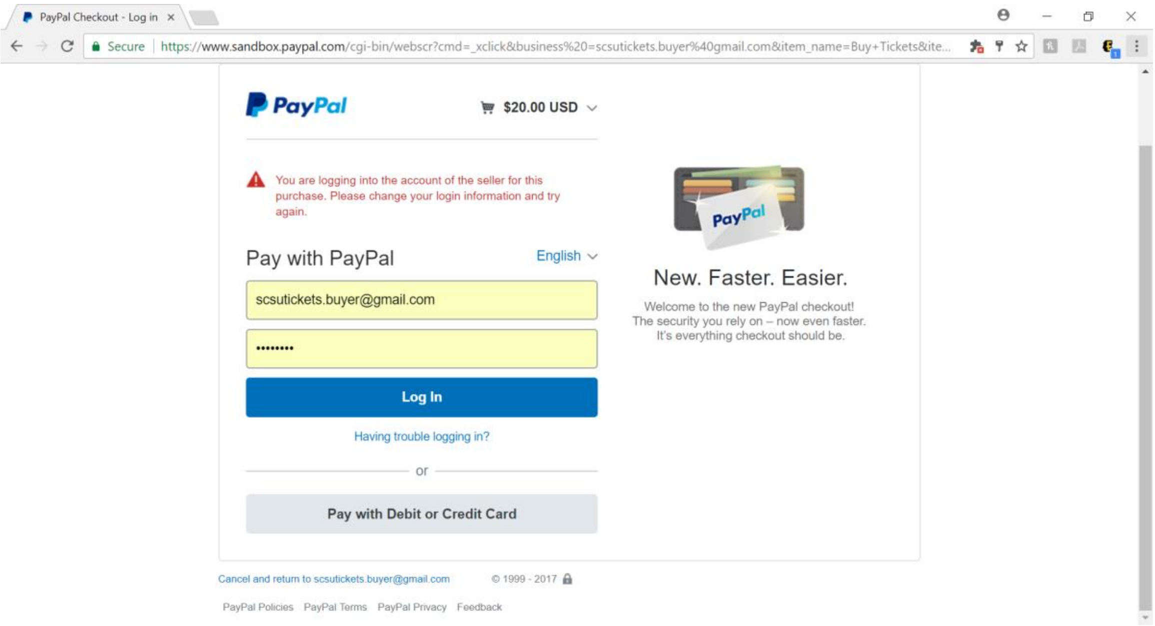
PayPal users can send and receive money for virtually any reason. The main focus in this application is commerce, so I’ve set up a PayPal business account with an emphasis of selling college event tickets. Personal accounts look slightly simplified as compared to a business account. If anyone shopping online, intend to send and receive money between friends and family members, a personal account will do. However, this application intends for a online business, so a business account has been set up. In order to do business with the PayPal account, credit and debit card payments should be accepted by the merchant.

This is very convenient because there’s no cost for using the debit card, and it has a MasterCard logo. Business accounts are recommended for all the merchant types. This account will display SCSU name to the customers, accept debit cards, credit cards and bank account payments. Business accounts permit to set up subscriptions and recurring payments with the customers. This account will also allow to send bulk payments to hundreds of people at one time.

To get started, click on the signup link on the top right-hand side of PayPal site. Here I was given two options and I selected business account option. Additionally, it asks if the account trying to open is a free one which offers standard PayPal payments or a PayPal payments Pro account which allows you to accept cards over the phone, by fax, or through the mail and to customize the checkout look and feel for your customers. For now, I’ve selected the free account for standard payments. Next step is to enter the Email address, complete the captcha and then click next. Then click agree and continue after reading the agreements and policies. Next step is to explain the nature of the business for statistical purposes.

The business information section contains your essential contact information. It's important to keep this current. The MyMoney link is where you manage bank accounts, cards on file, and pre- approved payments. Under my settings, we can change settings and set up notifications. To ensure the security of the account, we can add some security questions under my business info tab.

PayPal now is closely associated with vast array of E-commerce websites that are designed to accept PayPal payments with just a few configuration steps. After selecting the number of tickets and when the user clicks on the buy now button, the website redirects the user to PayPal website where the user can pay for the tickets by putting in the PayPal account details instead of the credit or debit card details. However, if the user doesn't have an account with PayPal, it is an easy process to get an account and the best thing is that the credit or debit card information has to be entered only once when you create your account for the first time. Later, whenever user wants to make payments with PayPal must just enter the PayPal login information and PayPal pulls the credit card information from your account and it is not displayed on the webpage.



https://www.sandbox.paypal.com/cgi-bin/webscr?cmd=_xclick&business%20=scsutickets.buyer%40gmail.com&item_name=Buy+Tickets&ite...

You are logging into the account of the seller for this purchase. Please change your login information and try again.

Pay with PayPal

scsutickets.buyer@gmail.com

••••••••

New. Faster. Easier.

Welcome to the new PayPal checkout!
The security you rely on – now even faster.
It's everything checkout should be.

Cancel and return to scsutickets.buyer@gmail.com

© 1999 - 2017

Figure 6.5. PayPal Page. “Buy Now” button when clicked directs to the PayPal Page. User is just required to log in to the PayPal site with the credentials and PayPal will take care of the payment without disposing the credit card details.

Chapter 7: System Testing

Many organizations find themselves developing software, whether for their own or for their internal use. Security concerns should be taken into account at every phase of the software development to ensure that the end result is safe, secure code that meets the requirements. Every software product should begin with a set of requirements. Software Testing is a part of secure code development process. Tests verify that the codes function properly and that it does not introduce any security issues. Throughout the software development process, developers and testers must engage in the software testing process to ensure that the finished product will function properly and meet the requirements. There are two main activities that occur during the testing. Model validation and verification. Software model validation ensures that software produced by a development effort is meeting the original business requirements. Basically, software model validation answers the question, are we building the right software? Software verification occurs throughout the development process and consists of tests that verify that the software functions properly.

Software verification answers the question, are we building the software right? As a developer, it is important to move beyond the small-scale tests done during development when the code is ready to move to production to ensure that the code works in the real-world production loads. During the load tests, developers use automated scripts in case automated testing or the testing team writes manual scripts in case of manual testing to stimulate real-world activity on the application.

These tests should verify that the system is able to handle maximum expected load that it will experience in the real-world. The tests also often continue to increase the load until the system fails to determine system’s maximum capacity. User acceptance testing is usually the final phase in the software testing. Once the developers are confident, they move the code to production and turn it over end-users for their final evaluation. The goal of user-acceptance testing (UAT) is to focus on usability and ensure that the system is easy to use

The purpose of testing is to identify errors in the process of checking the system work flow. It is designed to check the functionality of components in a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner.

In software development. There’s a lot of different approaches to testing an application. For example, there’s acceptance testing, black box testing, compatability testing, white box testing. One of the most popular is unit testing which plays an important role in the test-driven development.

Unit testing is the process in which tiny testable parts of a program, referred to as units, are independently tested for expected functionality. It could be automated or manually done. The testing in this application is done manually. Test-driven development is a process that uses these unit tests to drive the design of software. It provides several advantages such as, your intentions are clear and get to know what is wanted and then creating it makes the process easy.

When the code is tightly coupled, if one method is updated, it might break another and that forces to update the other methods too. This leads to a complex process. So, it is easier to catch bugs at the early stages.

Coding can be considered an art and when one developer is interested in such art and ready to accept that, he/she should also be ready to accept that testing is also are because the choices one makes to test the code have a creative element to it. Choices made to test the code may differ from one developer to another. Usually, there are two testing approaches that render the same result, but the first might be shorter and more time efficient for a deadline, but at the same time, the second approach might be more complete. As a developer, it is important not only code well but also defend why that approach have been choosen.

Why System Testing?

When testing the application check if the following questions are answered. These are few commonly raised questions from a testing team

1. Does the software meet the requirements? Completely? Partially?
2. Are all the features working correctly?
3. Does the application have any defect?
4. How is the quality of the software?
5. What is the risk of failure if application is moved to production?

.NET Applications Testing

There are number of different ways of testing .NET applications. Apart from the normal manual testing, however a lot of the methods work better during development rather than final testing of the site. Some of the .NET Integrated Development Environments (IDEs) available have some form of debugging and testing tools built into them. Visual Studio is such IDE used for the development of this web application and it has testing and debugging tools integrated. The testing of this program will consist of manual testing. The site will be test for its level of accessibility to ensure all pages meet the requirements.

User testing was done to comprehend the usability, accessibility and helps to attain a better understanding into how users will use the application and where they find problems rather than relying on the development and manual testing. This testing will include getting a host of different users to complete some simple task on the website than collect data on how quickly they are able to complete the tasks and their comments on what they thought of the site. This test should be happening through development to ensure that the site remains simple to use as development may make changes to the site that users do not like or find difficult to use.

This application often stores, processes and transmits sensitive information. This application serves the public, so it is better that the firewalls and other security devices are configured to allow access to the from the internet. If this application isn't written with sound security practices in mind, they can present a major vulnerability to the organization.

Cross-site scripting attack (XSS) is one such vulnerability where the attacker places a malicious script on a site that contains instructions directing a web browser to access a second site. Then the attacker waits. When the victim visits the site, the victim's browser unknowingly downloads and runs the code that attempts to access the second site. If the victim is already logged into the second site, the attacker's code can perform actions on the victim's behalf on that second site. If the victim isn't already logged in to the second site, the attack will fail.

But if the victim is logged into the other site, it just might succeed. Cross-site scripting, or XSS attacks, work under two conditions. First, the website must have reflected input. That means that there must be an opportunity where the attackers can provide text to the site that will then be displayed to other users. Second, that input must be invalidated. The attacker must be able to embed script commands into the input that will execute when the next user comes along and views that page.

We can execute our own scripts on the victim's machine. What if, for example, we instructed user X browser to visit a bank's website and transfer cash to user Y? if X wasn't a customer of that bank, or wasn't logged into that site, the attack would fail. But if X was logged into bank's website in another tab, it might succeed. XSS attackers try this thousands of times, waiting until they hit the jackpot once. They don’t care even they have 999 failures before a single success. Some other attacks include SQL Injection, Privilege escalation, directory traversal, overflow attacks, cookies, session hijacking, malicious add-ons, code execution attacks.

Problems noticed in current e-ticketing system. At present, our college ticketing system has a tiredness of ticket reservation from kiosks because of long queues and time restrictions. The three ticketing applications

1. Does not provide different payment levels
2. No Live chat

Chapter 8: Conclusion

The main motive of my research is to provide the user with a safer and secure system by using a few guidelines and rules and taking a few precautions. The whole purpose of the research is to ensure that the user of the application can trust the application and save his information on it without the worry of information theft or fraud. The purpose which is stated is more about the quality of the product than that of the quantity as to how many users are using it. Hence a qualitative study is more apt for it than the quantitative. The research done according to the qualitative study.

In this article, I have tried to investigate three of the existing e-ticketing applications to research and incorporate more user-interactive features in the existing system to sell our university events tickets to larger audience.

Futurework

The entire application can be developed in present advanced framework ASP.NET MVC because of lack of knowledge and working experience with MVC, this application was developed in ASP.NET 4.5. JQUERY can be used to make the user interface more interactive and attractive.

Additional features that can be incorporated with this application includes showing the users the number of tickets available, the number of tickets sold, allow the customers to cancel their tickets and hence the process a refund back to the users PayPal account.

Due to the time constraint, this system has been developed as web application. We all know that mobile phones are most widely used now-a-days and to provide easier user access to this application, a mobile application can be developed serving the same purpose.

Acheivements

This paper addresses the problems with the current e-ticketing applications metioned in chapter 7.

- ✓ No tiredness of standing in long queue for ticket reservation
- ✓ Addresses different levels Payments
- ✓ Provides Live chat
- ✓ Secured payment gateway – 'PayPal'

Advantages of the Application

1. Cross platform: Application can easily be ported virtually to any platform with a web browser
2. Can be distributed offline. Utilities make the porting easy across various platforms
3. Not required to use a proprietary software that locks the access of the platform

Limitations

ASP.NET webforms development environment works on a process called RAD. As soon as we drag and drop components in the designer mode, visual studio creates an object at the back end and when we double click on the component it creates an event. The RAD programming is great. However, it leads to some other issues. One of such issues is with the behind code. Automation has its own benefits and issues. The biggest advantage of automation is that the projects are delivered faster but the disadvantage with this automation is that it has led to a very inappropriate architecture.

For example, when button is clicked on the UI, only the button_click event should be executed, but first the page life cycle events are executed no matter whichever event is called. This issue has been overcome in the MVC.

Other limitations of this research include

1. The research limits in analyzing and implementing the different Payment gateways in E-Ticketing
2. The study mainly focuses on implementation of live chat incorporating online live chat application in web application
3. The data collected for this research and practical programming may not be accurate

Bibliography

"Aloud" (n.d.). Retrieved Jan 2017, from Aloud: http://www.aloud.com/

"An article on PayPal" (n.d.). Retrieved July 2017, from https://www.paypal.com/us/cgi-bin/webscr?cmd=_payflow-gatewayoverview-outside

"An overview of .NET framework. microsoft" (n.d.). Retrieved Jan 2017, from https://msdn.microsoft.com/en-us/library/zw4w595w(v=vs.110).aspx

Bradley, M. (2006). Online ticket booking system, Dissertation CS39030 online ticket sales system. Link: https://pdfs.semanticscholar.org/8179/a366378246b3454966f1a6cdb4ab5868eb9f.pdf

Deitel H.M. (2001). E-business and E-commerce for managers. Prentice hall publishing, New Jersey. pages 92, 94, 95, 97, 193

E-Business., Western, Pan. (2008). Internet payment processing http://www.treasuryalliance.com/assets/publications/payments/Fundamentals_of_Payment_Systems.pdf

Efraim, Turban., David, King., Jae, Lee., & Dennis, Viehland. (2008). Online material: Electronic commerce – A managerial perspective. http://wps.prenhall.com/wps/media/objects/5073/5195381/pdf/Turban_Online_W12.pdf

E-ticketing system development team, (May 2002). E-ticketing system user manual: System concept. University of Macau.

Gu, Yongfei. (2001). Multi-platform portable E-ticketing systems. Master thesis, University of Macau.

Guo, Z. S., Siu, W. I., Wong, C. T. (2009). Full-scale online event ticketing system the design and implementation. Faculty of science and technology. University of Macau.

Jacob, Suher. (2009). Forecasting event ticket sales. University of Pensylvania

Kirkman, Clark P. (2009) Who needs tickets? Examining problems in the growing online ticket resale industry. Federal communications law journal: Vol. 61: Iss. 3, Article 8. http://www.repository.law.indiana.edu/fclj/vol61/iss3/8

"My Code Lines" (n.d.). Retrived on Aug 2017, from https://mycodelines.wordpress.com/2009/01/22/features-of-the-common-language-runtime/

"My Live Chat" (n.d.). Retrieved June 2017, from https://mylivechat.com/default.aspx

Niranjanamurthy, M. (July 2004). E-commerce recommended online payment method – PayPal. International journal of computer science and mobile computing, Vol.3 Issue.7, pg. 669-679

"N-tier" (n.d.). Retrieved on May 2017, from http://n-tier.com/articles/csovervw.html

Tickets, Barry's. (January 2016). Digital challenges for online event ticketing industry.

"Ticket master" (n.d.). Retrieved Jan 2017, from Ticket master: https://www.ticketmaster.com/

"Ticketscript" (2013). Analysing the trend towards direct and online industry. The state of UK ticketing industry. Redshift research crowdology survey.

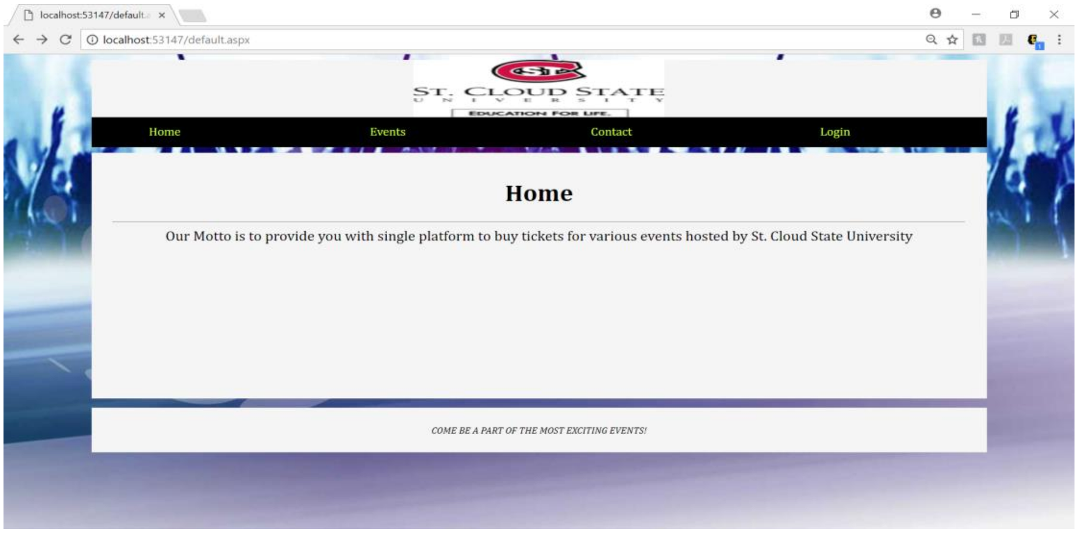
"Ticketweb" (n.d.). Retrieved Jan 2017, from Ticket web: https://www.ticketweb.com/

Weng, In Siu., Guo, Zhen Sheng. (Nov 2001). Application to electronic online trading with smart card technology. The 6th INFORMS conference on information systems & technology (CIST-2001).

“What are the benefits of using PayPal?” (n.d.). https://www.paypal.com/us/selfhelp/article/whatare-the-benefits-of-using-paypal-faq3704

Appendix

Screenshots: CLIENT MODULE

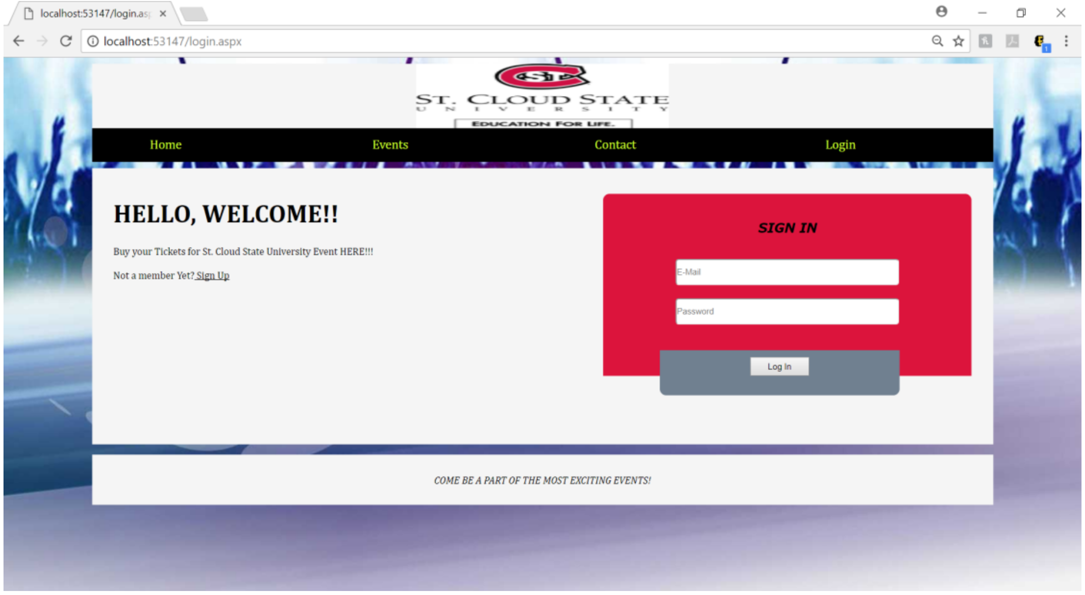


Home

Our Motto is to provide you with single platform to buy tickets for various events hosted by St. Cloud State University

COME BE A PART OF THE MOST EXCITING EVENTS!

Figure 8.1. Web Application Home Page



HELLO, WELCOME!!

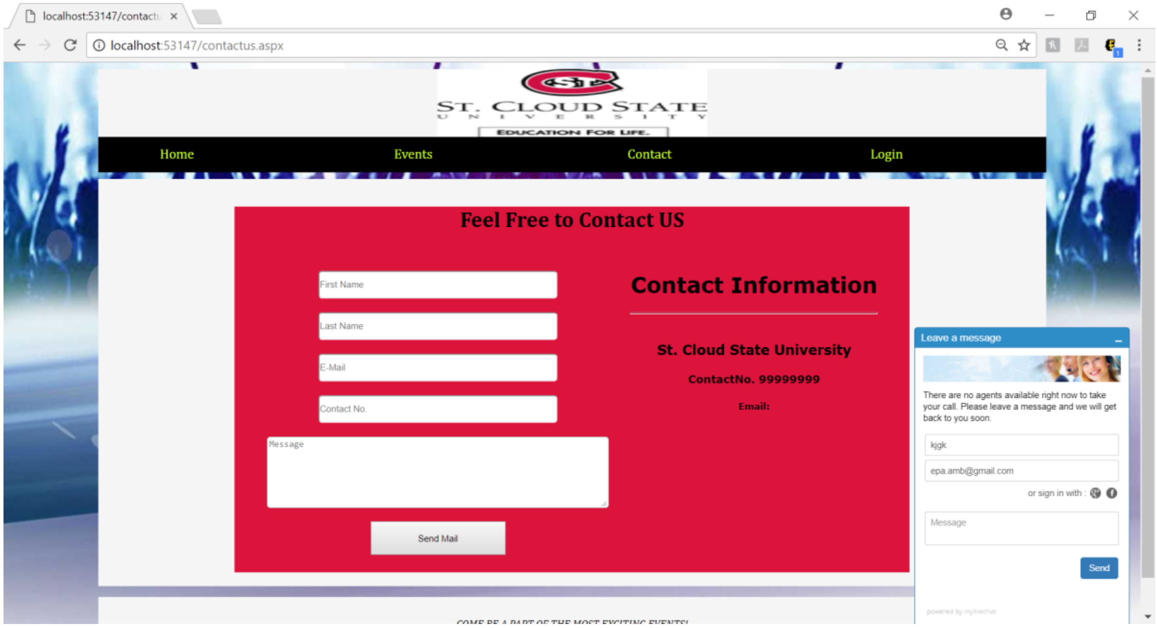
Buy your Tickets for St. Cloud State University Event HERE!!!

Not a member Yet? Sign Up

COME BE A PART OF THE MOST EXCITING EVENTS!

Figure 8.2. Web Application Login Page

Requires user email and password to login. Also has a signup option



Feel Free to Contact US

First Name

Last Name

E-Mail

Contact No.

Message

Contact Information

St. Cloud State University

ContactNo. 99999999

Email:

Leave a message

There are no agents available right now to take your call. Please leave a message and we will get back to you soon.

kjgk

epa.amb@gmail.com

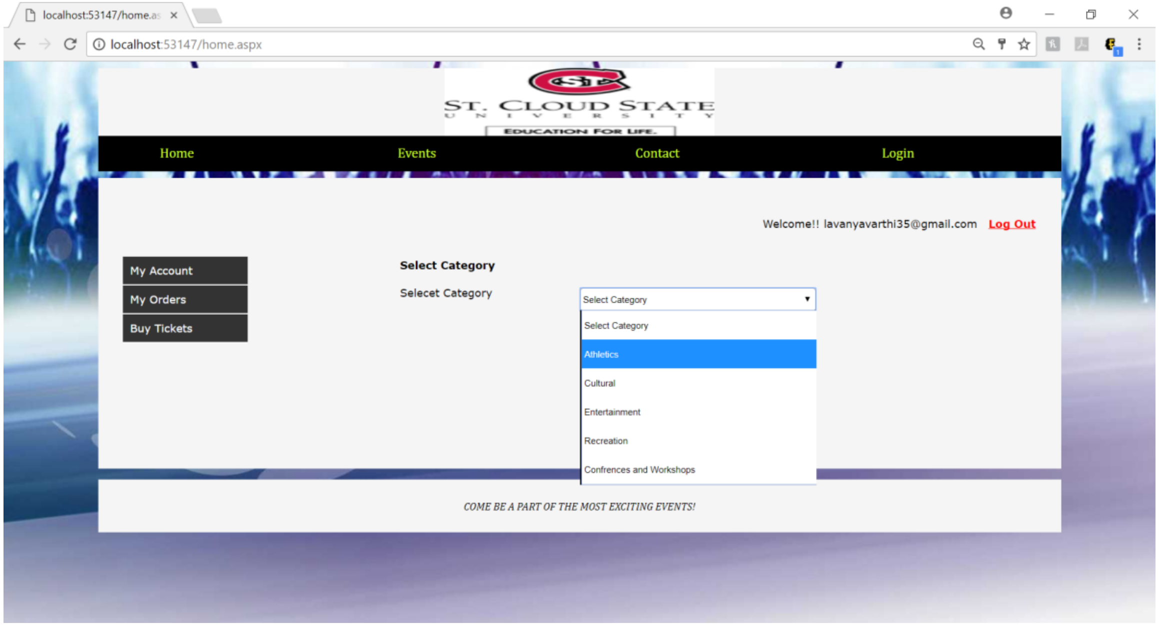
or sign in with :

Message

powered by mylivechat

Figure 8.3. Contact Page with Live Chat

User can send an email or chat live with admin



Welcome!! lavanyavarthi35@gmail.com

Select Category

Selecet Category

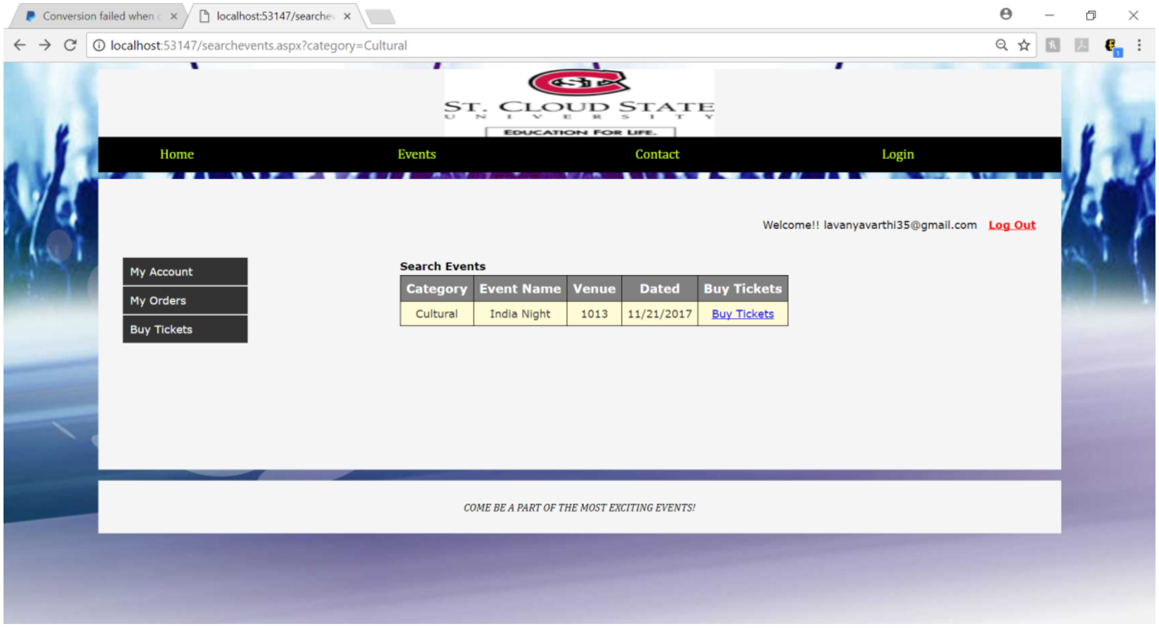
Select Category

- Select Category
- Athletics
- Cultural
- Entertainment
- Recreation
- Confrences and Workshops

COME BE A PART OF THE MOST EXCITING EVENTS!

Figure 8.4. Buy Tickets Page

List of categories of events at SCSU



localhost:53147/searchevents.aspx?category=Cultural

Welcome!! lavanyavarthi35@gmail.com

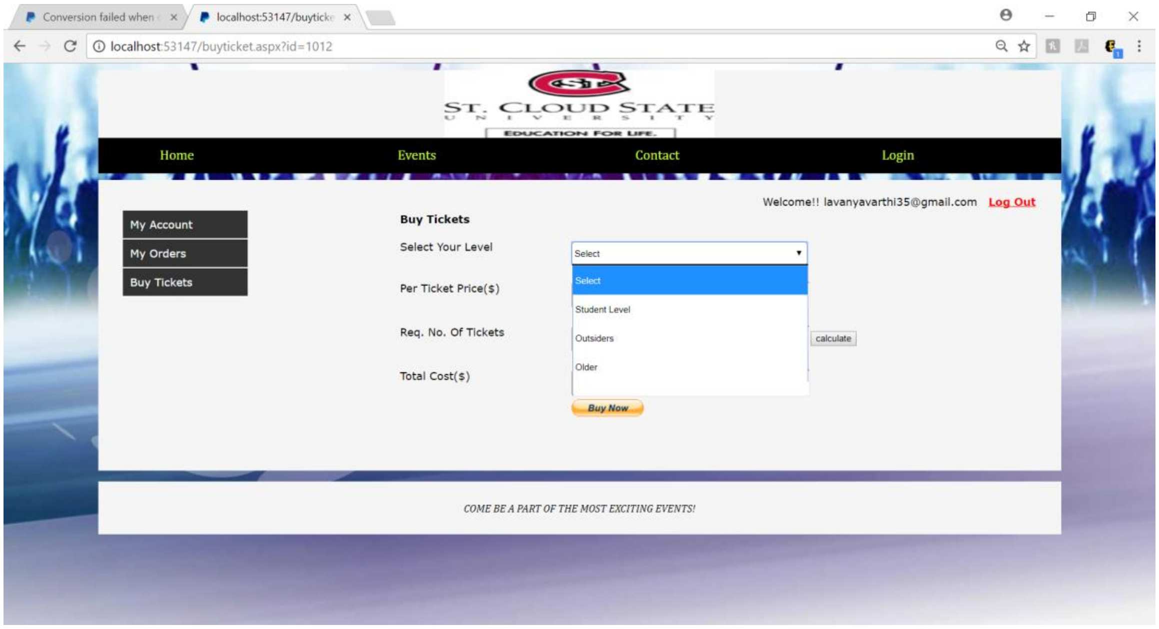
Search Events

Category	Event Name	Venue	Dated	Buy Tickets
Cultural	India Night	1013	11/21/2017	Buy Tickets

COME BE A PART OF THE MOST EXCITING EVENTS!

Figure 8.5. List of Events under Cultural Category

Shows all the events under that particular category



localhost:53147/buyticket.aspx?id=1012

Welcome!! lavanyavarthi35@gmail.com

Buy Tickets

Select Your Level

Per Ticket Price($)

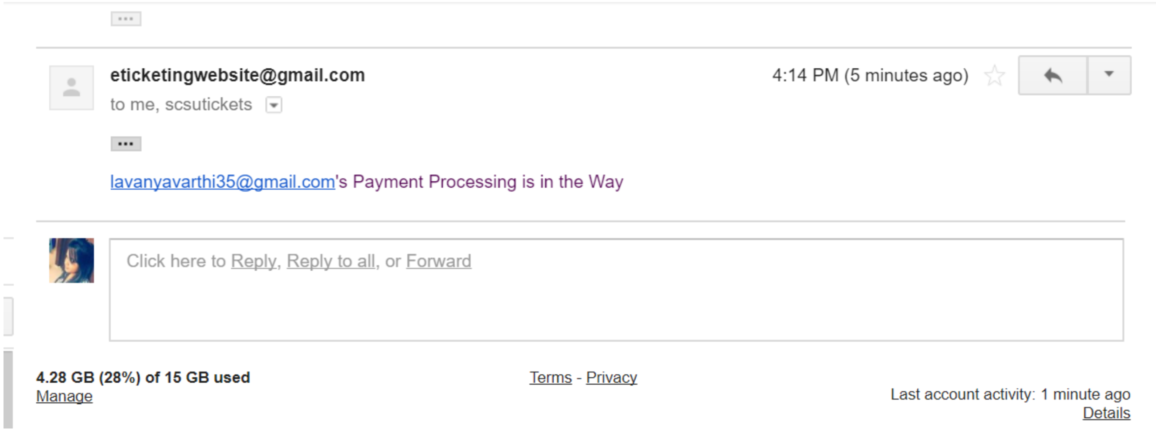
Req. No. Of Tickets

Total Cost($)

COME BE A PART OF THE MOST EXCITING EVENTS!

Figure 8.6. Various Levels of Attendees and Price Set Page

User is required to select the level and the price per ticket populates accordingly



eticketingwebsite@gmail.com

to me, scsutickets

4:14 PM (5 minutes ago)

lavanyavarthi35@gmail.com's Payment Processing is in the Way

Click here to Reply, Reply to all, or Forward

4.28 GB (28%) of 15 GB used
Manage

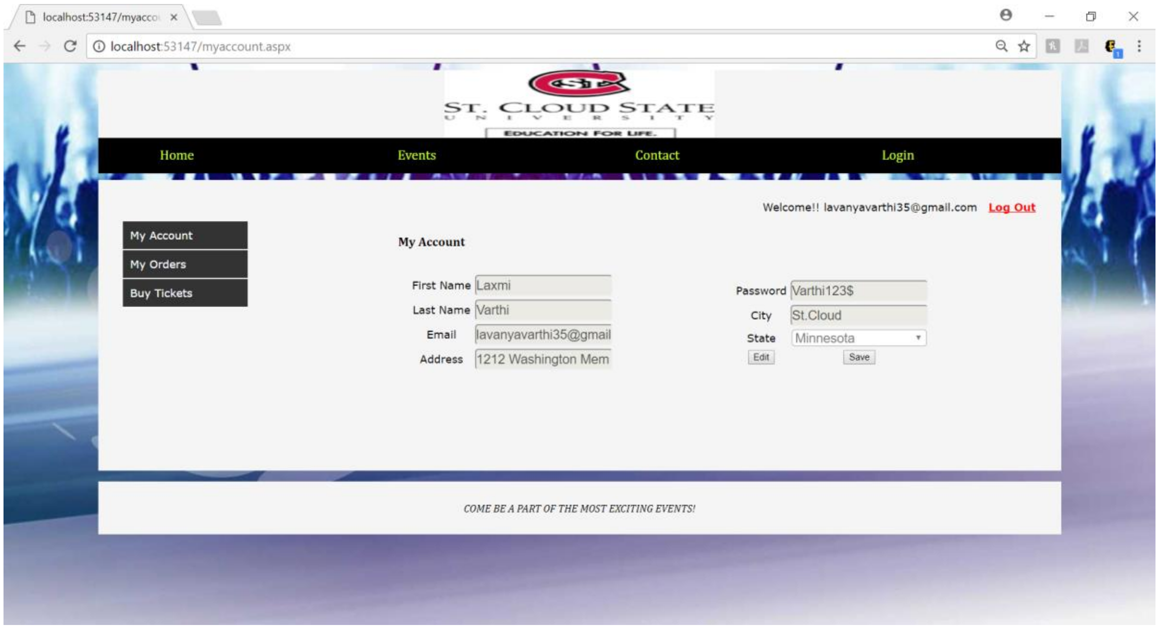
Terms - Privacy

Last account activity: 1 minute ago
Details

Figure

8.7. Email of Payment Confirmation Page

Email Confirmation will be sent to both user and the merchant. Once the Merchant receives the payment, ticket confirmation will then be sent to the user.



localhost:53147/myaccount.aspx

Welcome!! lavanyavarthi35@gmail.com Log Out

My Account

First Name: Laxmi

Last Name: Varthi

Email: lavanyavarthi35@gmail

Address: 1212 Washington Mem

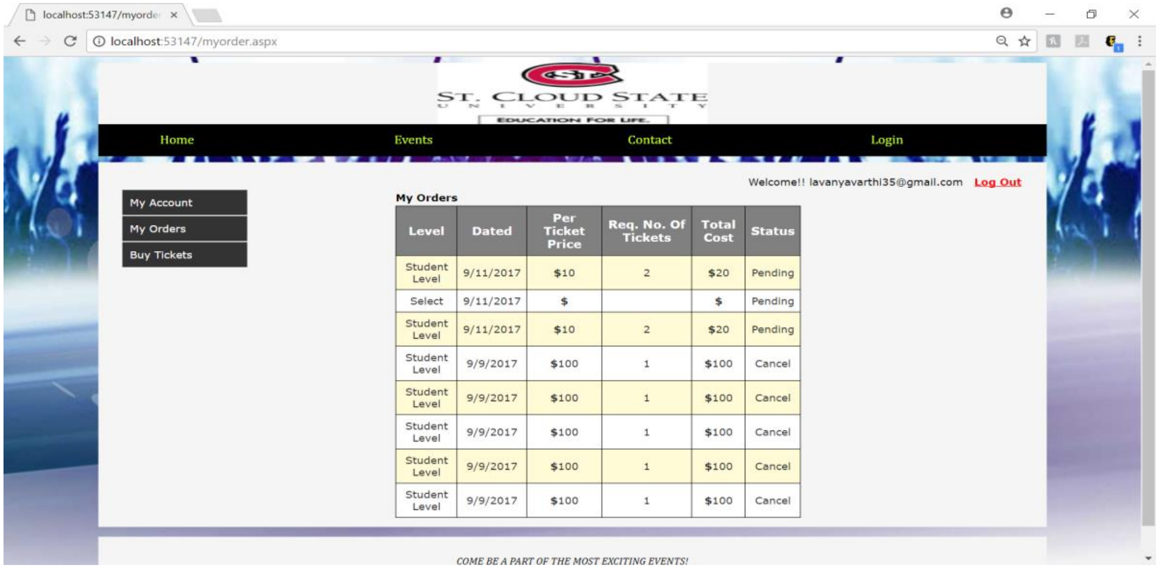
Password: Varthi123$

City: St.Cloud

State: Minnesota

Figure 8.8. My Account Page

User can reset account details



Welcome!! lavanyavarthi35@gmail.com

My Orders

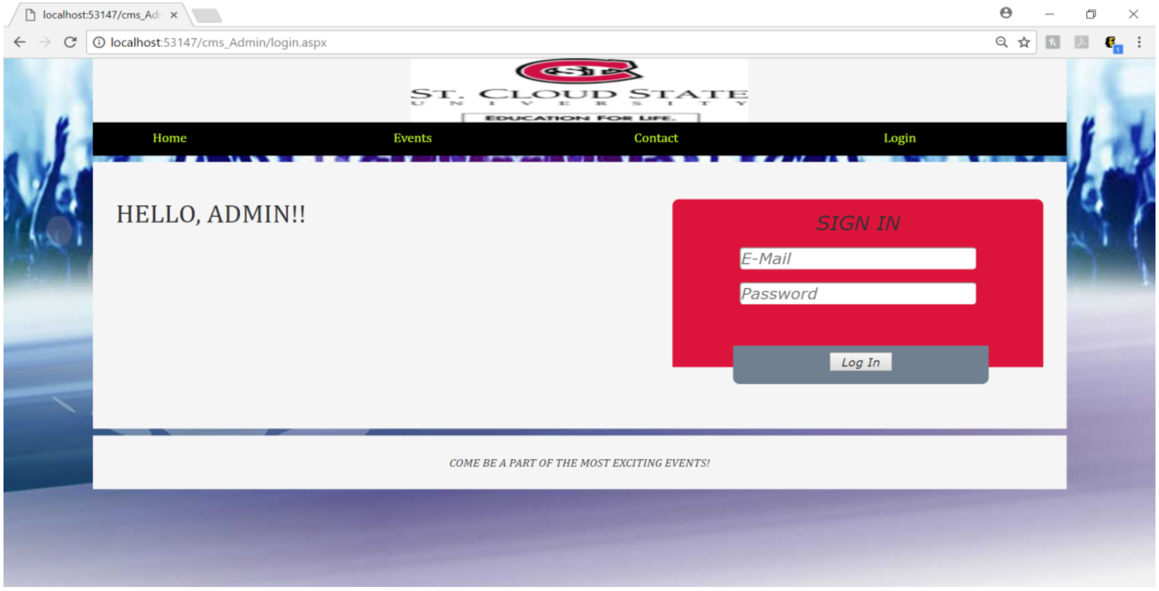
Level	Dated	Per Ticket Price	Req. No. Of Tickets	Total Cost	Status
Student Level	9/11/2017	$10	2	$20	Pending
Select	9/11/2017	$		$	Pending
Student Level	9/11/2017	$10	2	$20	Pending
Student Level	9/9/2017	$100	1	$100	Cancel
Student Level	9/9/2017	$100	1	$100	Cancel
Student Level	9/9/2017	$100	1	$100	Cancel
Student Level	9/9/2017	$100	1	$100	Cancel
Student Level	9/9/2017	$100	1	$100	Cancel

COME BE A PART OF THE MOST EXCITING EVENTS!

Figure 8.9. My Orders Page

List and status of the order.

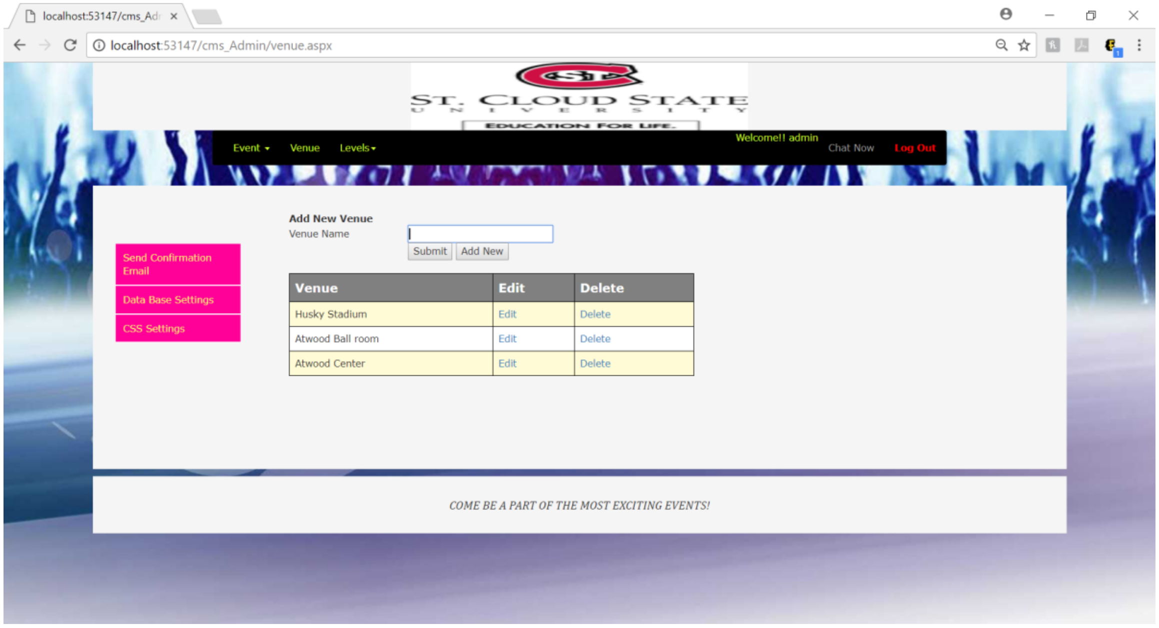
Screenshots: ADMIN MODULE



HELLO, ADMIN!!

COME BE A PART OF THE MOST EXCITING EVENTS!

Figure 8.10. Admin Login Page



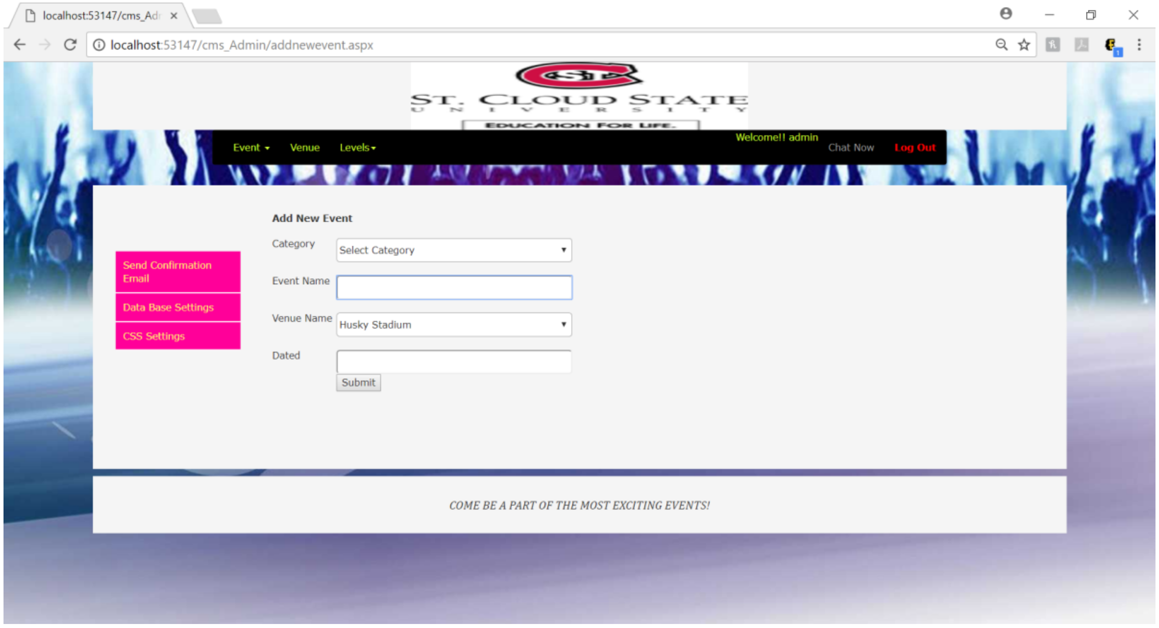
Add New Venue

Venue Name

Venue	Edit	Delete
Husky Stadium	Edit	Delete
Atwood Ball room	Edit	Delete
Atwood Center	Edit	Delete

Figure 8.11. Add Venue Page

Admin can add/delete/edit venue.



Add New Event

Category: Select Category ▾

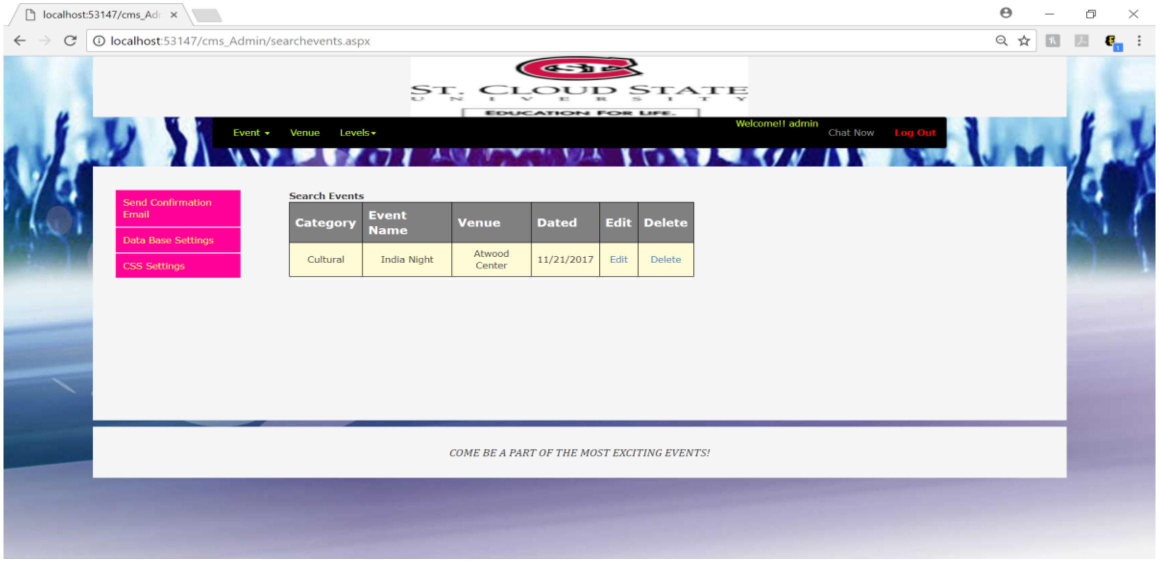
Event Name

Venue Name: Husky Stadium ▾

Dated

Figure 8.12. Add Event Page

Allows admin to create new event, event venue and event date.

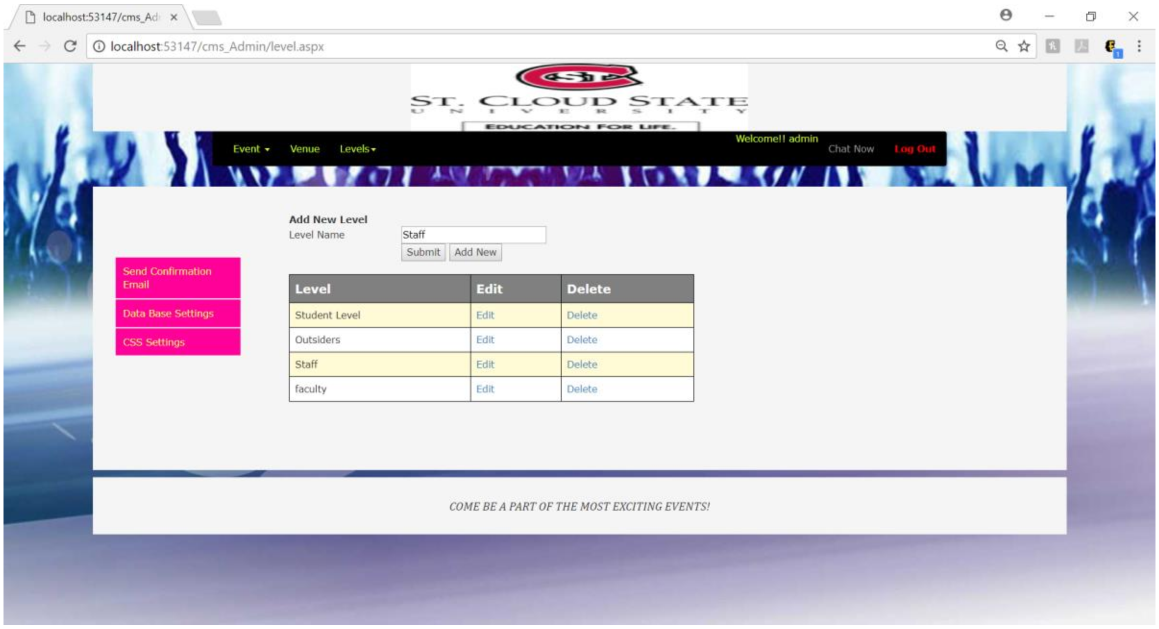


Search Events

Category	Event Name	Venue	Dated	Edit	Delete
Cultural	India Night	Atwood Center	11/21/2017	Edit	Delete

Figure 8.13. Search/Delete Event Page

Admin can edit/delete events.



Add New Level

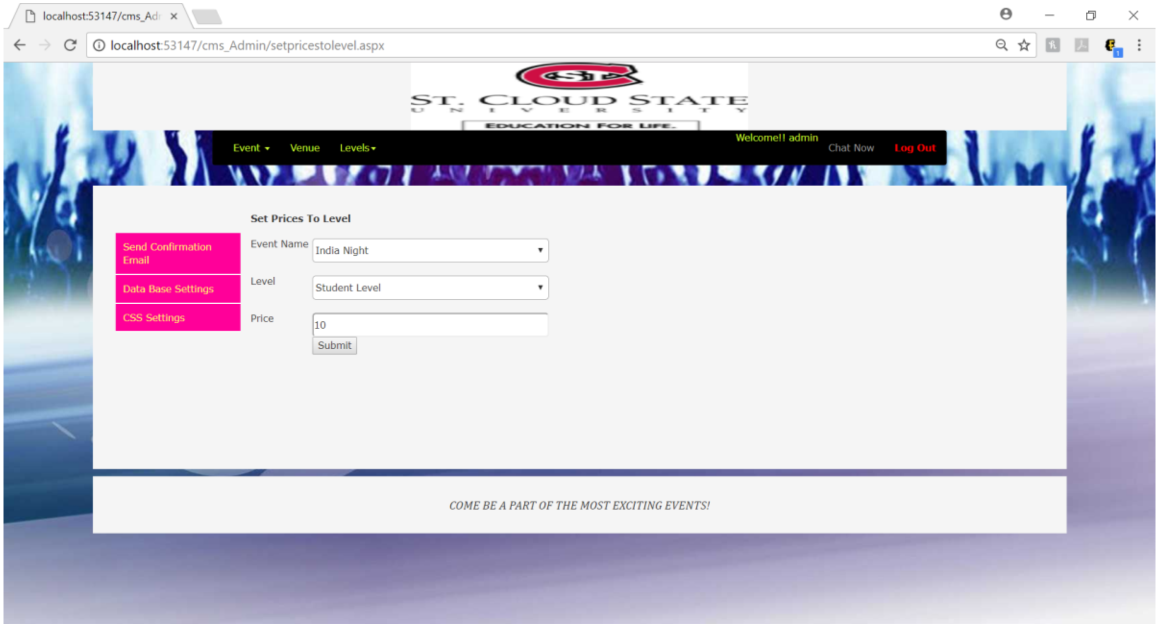
Level Name: Staff

Level	Edit	Delete
Student Level	Edit	Delete
Outsiders	Edit	Delete
Staff	Edit	Delete
faculty	Edit	Delete

Figure 8.14. Add/Delete Levels Page.

Different levels noticed till date are student, staff, faculty, and others.

This page allows admin to add/delete levels.



Set Prices To Level

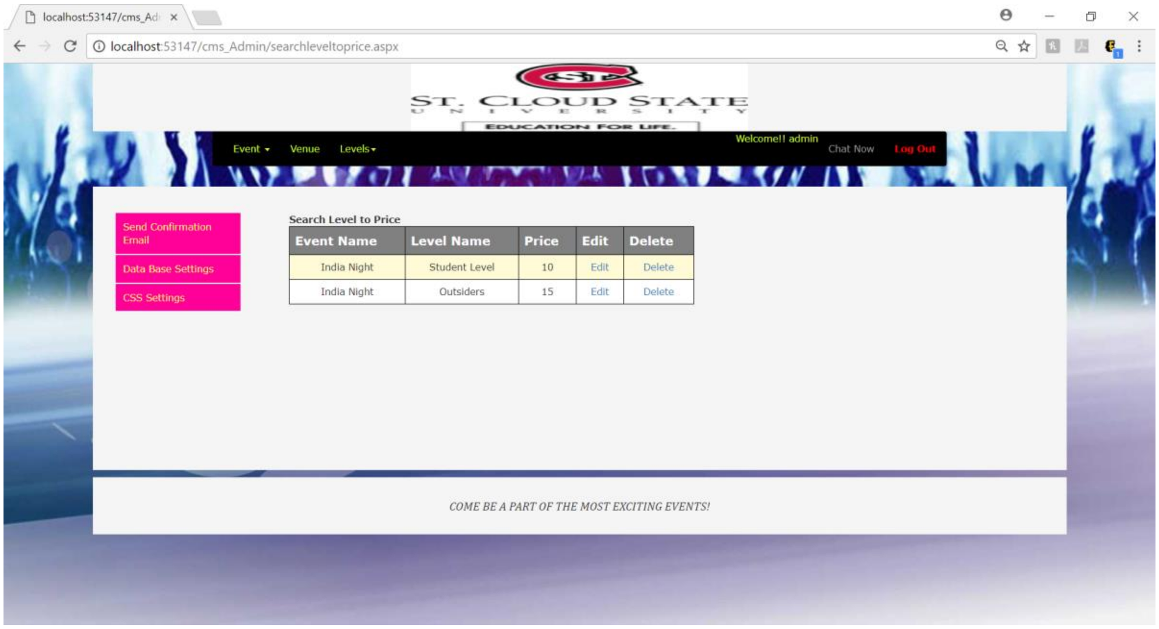
Event Name: India Night

Level: Student Level

Price: 10

Figure 8.15. Set Price Levels Page

To test, prices are set for student -$10, faculty-$15, Staff -$8.

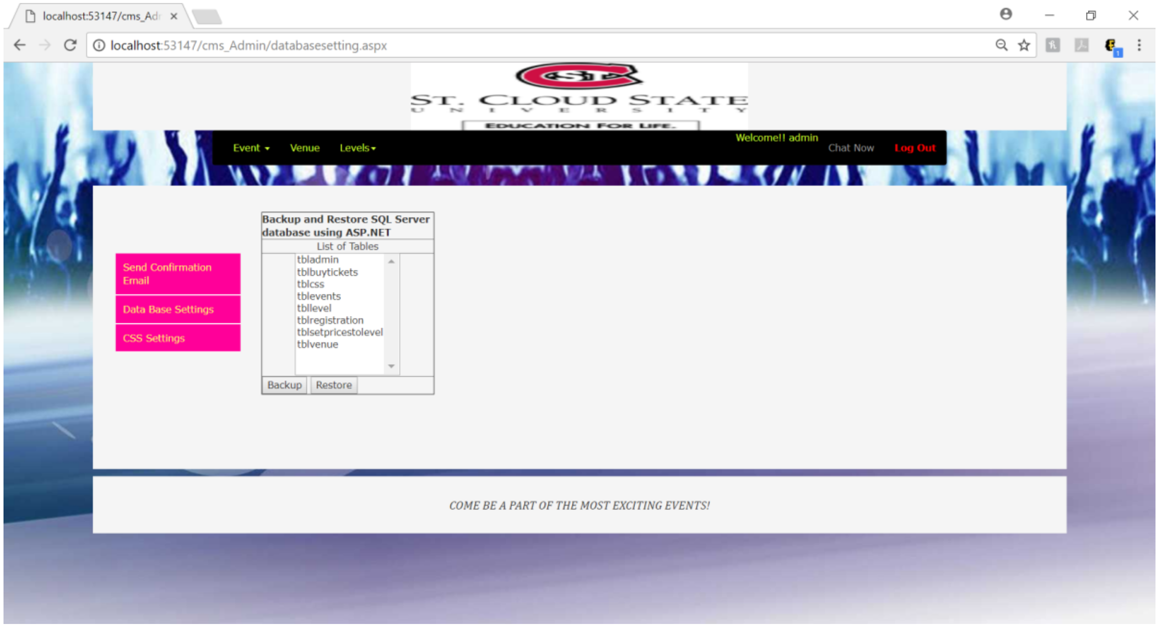


Search Level to Price

Event Name	Level Name	Price	Edit	Delete
India Night	Student Level	10	Edit	Delete
India Night	Outsiders	15	Edit	Delete

Figure 8.16. Search/Delete Price Levels Page

Prices set can be edited on this page by admin.



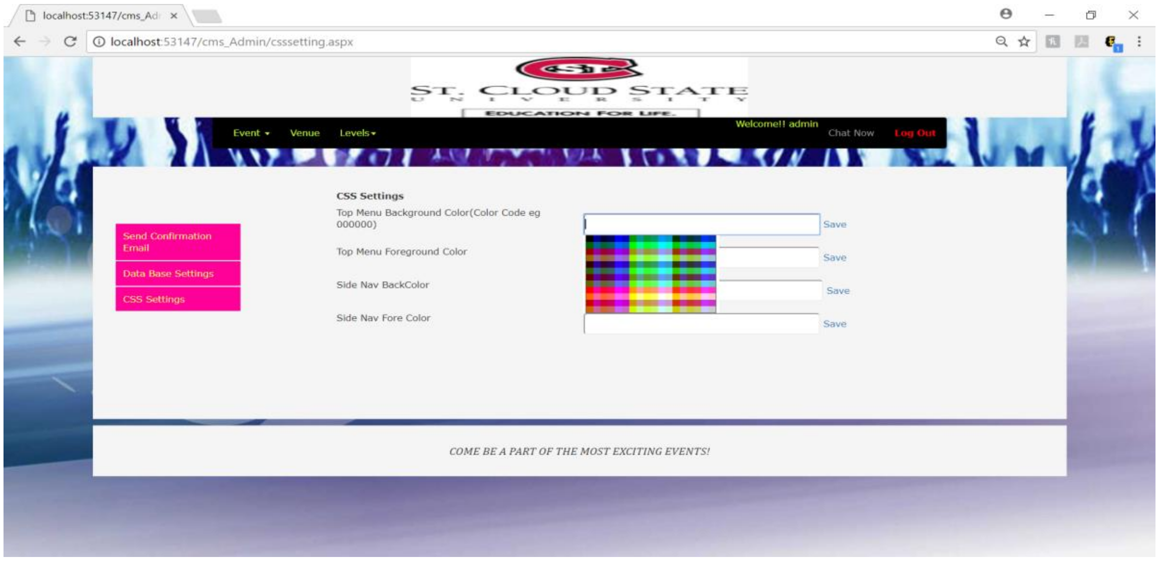
Backup and Restore SQL Server database using ASP.NET

List of Tables

tbladmin
tblbuytickets
tblcss
tblevents
tbllevel
tblregistration
tblsetpricestolevel
tblvenue

Figure 8.17. Database Settings Page

Allows admin to backup and restore database from the application.



CSS Settings

Top Menu Background Color(Color Code eg 000000)

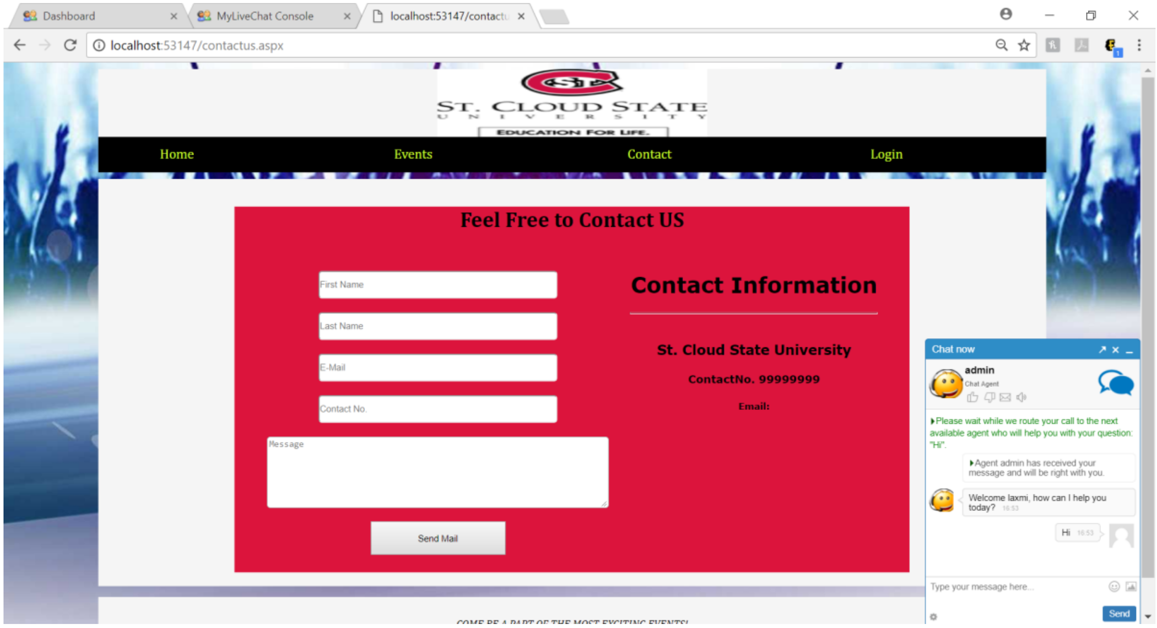
Top Menu Foreground Color

Side Nav BackColor

Side Nav Fore Color

Figure 8.18. CSS Settings Page

Admin can change colors of UI from this page.



Feel Free to Contact US

Contact Information

St. Cloud State University

ContactNo. 99999999

Email:

admin
Chat Agent

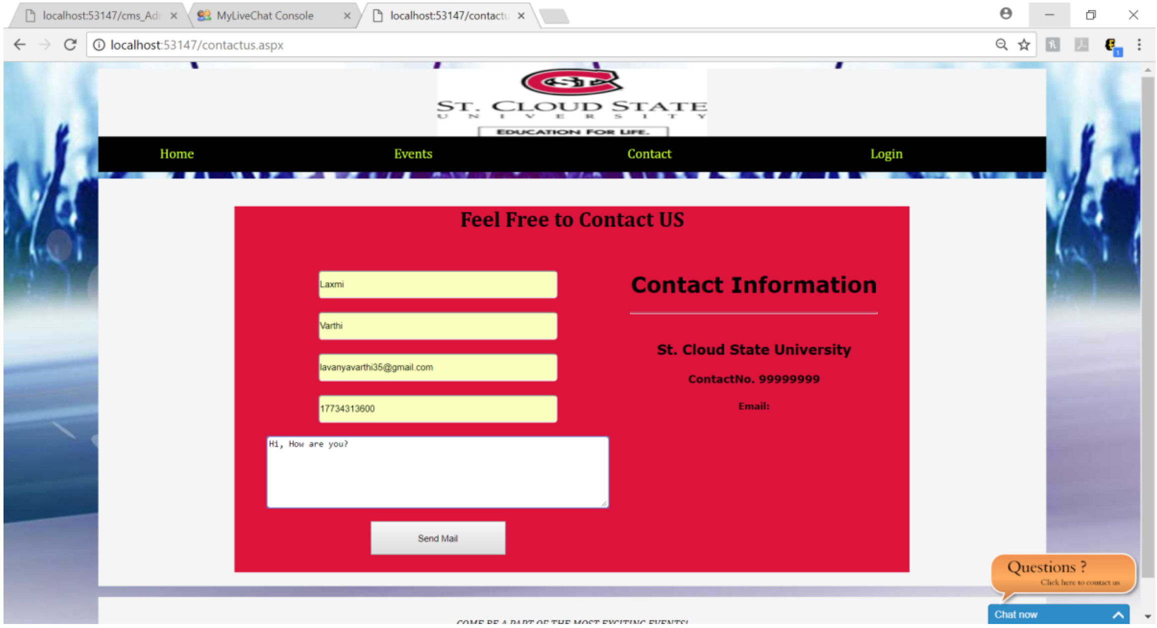
Please wait while we route your call to the next available agent who will help you with your question: "Hi".

Agent admin has received your message and will be right with you.

Welcome laxmi, how can I help you today? 16:53

Hi 16:53

Figure 8.19. User Chat Page



Feel Free to Contact US

Laxmi

Varthi

lavanyavarthi35@gmail.com

17734313600

Hi, How are you?

Contact Information

St. Cloud State University

ContactNo. 99999999

Email:

Figure 8.20. Contact Email Page

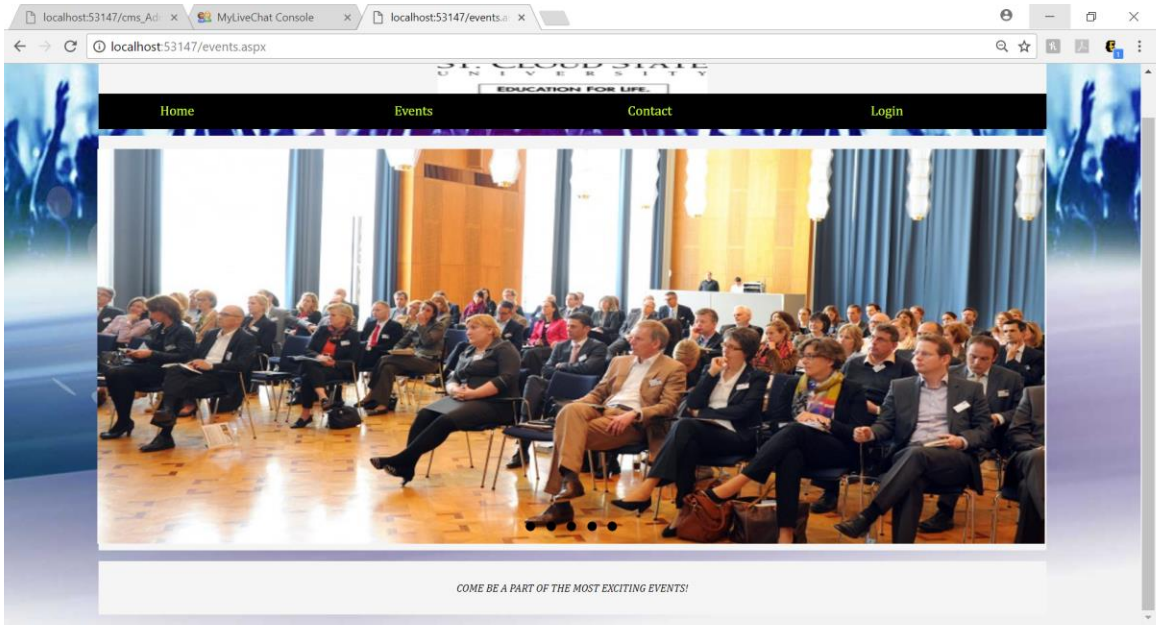
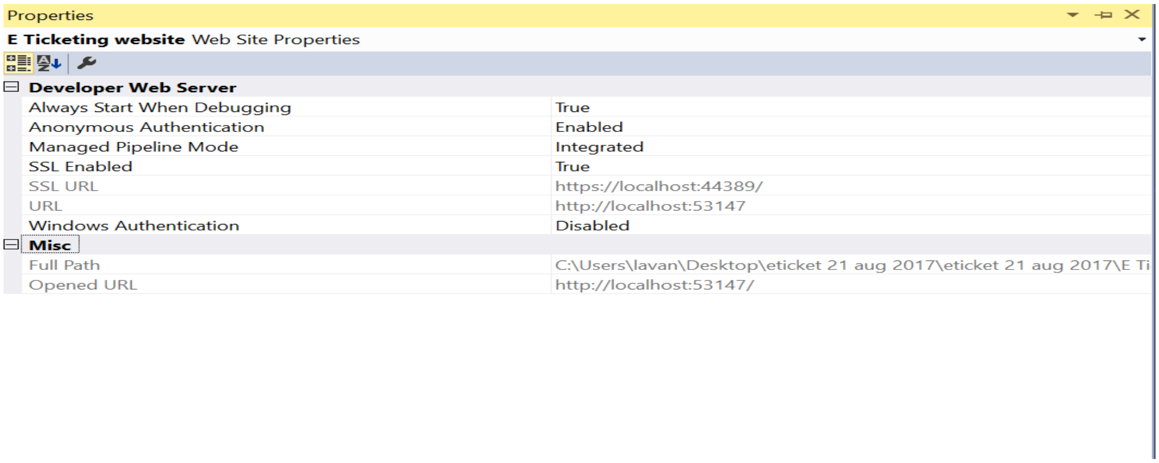


Figure 8.21. Events Page

Slideshow of upcoming events.



E Ticketing website Web Site Properties

Developer Web Server	
Always Start When Debugging	True
Anonymous Authentication	Enabled
Managed Pipeline Mode	Integrated
SSL Enabled	True
SSL URL	https://localhost:44389/
URL	http://localhost:53147
Windows Authentication	Disabled
Misc	
Full Path	C:\Users\lavan\Desktop\eticket 21 aug 2017\eticket 21 aug 2017\E Ti
Opened URL	http://localhost:53147/

Figure 8.22. Standard Security Protocol (SSL) Enabled

SSL enabled application when runs on server transmits the encrypted information.

URL: http:/ SSL URL: https:/

userMaster.master

```
    <%@ Master Language="C#" MasterPageFile="~/masterpage.master" AutoEventWireup="true"
CodeFile="userMaster.master.cs" Inherits="userMaster" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">



    <table style="width:100%; height:300px">
    <tr>
    <td colspan="2" style="text-align:right">
        <asp:Label ID="lblemail" runat="server" Text=""></asp:Label>  &nbsp;
        <asp:LinkButton ID="LinkButton4" runat="server" CssClass="btnSignUp"
            style="color:red" onclick="LinkButton4_Click"><b>Log
Out</b></asp:LinkButton></td>

    </tr>
    <tr>
    <td style="width:30%; vertical-align: top;">
     <div class="menu_simple">

<ul>
  <li><a href="myaccount.aspx"> My Account</a></li>
  <li><a href="myorder.aspx"> My Orders</a></li>
        <li> <a href="home.aspx"> Buy Tickets</a></li>

    </td>
     <td style="width:70%;  vertical-align: top;">
         <asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server">
         </asp:ContentPlaceHolder>
    </td>
    </tr>
    </table>
</asp:Content>
```

userMaster.master.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class userMaster : System.Web.UI.MasterPage
{
    protected void Page_Load(object sender, EventArgs e)
    {
        lblemail.Text ="Welcome!! "+ Session["email"].ToString();
    }
    protected void LinkButton4_Click(object sender, EventArgs e)
    {
         Session["email"] = null;
         Response.Redirect("login.aspx");
    }
}
```

Events.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/masterpage.master"
AutoEventWireup="true" CodeFile="events.aspx.cs" Inherits="events" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
  <div id="jssor_1" style="position:relative;margin-left:-2.6%;
top:0px;left:0px;width:1200px;height:500px;overflow:hidden;visibility:hidden;">
        <!-- Loading Screen -->
        <div data-u="loading" class="jssorl-004-double-tail-spin"
style="position:absolute;top:0px;left:0px;width:100%;height:100%;text-
align:center;background-color:rgba(0,0,0,0.7);">
            <img style="margin-top:-
19px;position:relative;top:50%;width:38px;height:38px;" src="img/double-tail-spin.svg" />
        </div>
        <div data-u="slides"
style="cursor:default;position:relative;top:0px;left:0px;width:1300px;height:500px;overfl
ow:hidden;">
            <div>
                <img data-u="image" src="images/1.jpg" />

            </div>
            <div>
                <img data-u="image" src="images/2.bmp" />

            </div>
            <div>
                <img data-u="image" src="images/3.jpg" />

            </div>
            <div>
                <img data-u="image" src="images/4.jpg" />

            </div>


            <div>
                <img data-u="image" src="images/5.jpg" />

            </div>

            <a data-u="any" href="https://www.jssor.com" style="display:none">bootstrap
carousel</a>
        </div>
        <!-- Bullet Navigator -->
        <div data-u="navigator" class="jssorb051"
style="position:absolute;bottom:14px;right:12px;" data-autocenter="1" data-scale="0.5"
data-scale-bottom="0.75">
            <div data-u="prototype" class="i" style="width:18px;height:18px;">
                <svg viewbox="0 0 16000 16000"
style="position:absolute;top:0;left:0;width:100%;height:100%;">
                    <circle class="b" cx="8000" cy="8000" r="5800"></circle>
        </svg>
      </div>
    </div>
    <!-- Arrow Navigator -->

  </div>


        <script src="js/jssor.slider-25.2.1.min.js"
type="text/javascript"></script>
  <script type="text/javascript">
    jssor_1_slider_init = function () {

      var jssor_1_SlideoTransitions = [
       [{ b: 900, d: 2000, x: -379, e: { x: 7}}],
       [{ b: 900, d: 2000, x: -379, e: { x: 7}}],
       [{ b: -1, d: 1, o: -1, sX: 2, sY: 2 }, { b: 0, d: 900, x: -171, y: -341, o:
1, sX: -2, sY: -2, e: { x: 3, y: 3, sX: 3, sY: 3} }, { b: 900, d: 1600, x: -283, o: -1,
e: { x: 16}}]
      ];

      var jssor_1_options = {
        $AutoPlay: 1,
        $SlideDuration: 800,
        $SlideEasing: $Jease$.$OutQuint,
        $CaptionSliderOptions: {
          $Class: $JssorCaptionSlideo$,
          $Transitions: jssor_1_SlideoTransitions
        },
        $ArrowNavigatorOptions: {
          $Class: $JssorArrowNavigator$
        },
        $BulletNavigatorOptions: {
          $Class: $JssorBulletNavigator$
        }
      };

      var jssor_1_slider = new $JssorSlider$("jssor_1", jssor_1_options);

      /*#region responsive code begin*/
      function ScaleSlider() {
        var containerElement = jssor_1_slider.$Elmt.parentNode;
        var containerWidth = containerElement.clientWidth;
        if (containerWidth) {
          var MAX_WIDTH = 3000;

          var expectedWidth = containerWidth;

          if (MAX_WIDTH) {
            expectedWidth = Math.min(MAX_WIDTH, expectedWidth);
          }

          jssor_1_slider.$ScaleWidth(expectedWidth);
        }
        else {
          window.setTimeout(ScaleSlider, 30);
        }
```

```
            }

            ScaleSlider();
            $Jssor$.$AddEvent(window, "load", ScaleSlider);
            $Jssor$.$AddEvent(window, "resize", ScaleSlider);
            $Jssor$.$AddEvent(window, "orientationchange", ScaleSlider);
            /*#endregion responsive code end*/
        };
    </script>
     <script type="text/javascript">         jssor_1_slider_init();</script>
                </ContentTemplate>
            </asp:UpdatePanel>
        </div>
    </div>
</asp:Content>
```

Login.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/masterpage.master"
AutoEventWireup="true" CodeFile="login.aspx.cs" Inherits="login" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

    <div id="divText">
 <p style="font-family:Cambria; "  >
 <h1 STYLE="font-family:Cambria; font-size:40px">HELLO, WELCOME!!</h1>
<p style="font-family:Cambria; "  >Buy your Tickets for St. Cloud State University Event
HERE!!!<br /><br />
 Not a member Yet?<a href="registerationpage.aspx" class="btnSignUp"> Sign Up</a></p>
 </div>
  <div class="divLoginContents" >

     <h2>SIGN IN</h2>
          <asp:TextBox ID="txtemail" CssClass="txtEmailPswd" runat="server"
placeholder="E-Mail"></asp:TextBox><br />
     <asp:TextBox ID="txtpassword" CssClass="txtEmailPswd" runat="server"
          placeholder="Password" TextMode="Password"></asp:TextBox><br />

        <div class="divbtnLogin" >
     <asp:Button ID="Button1" runat="server" Text="Log In"
         onclick="Button1_Click" Height="30px" Width="93px" /></div>
 </div>
</asp:Content>
```

Login.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Configuration;
public partial class login : System.Web.UI.Page
```

```
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
    SqlConnection con = new
SqlConnection(ConfigurationManager.ConnectionStrings["cn"].ConnectionString);
    protected void Button1_Click(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("select * from tblregistration where email='" +
txtemail.Text + "' and password='" + txtpassword.Text + "'", con);
        SqlDataReader dr = cmd.ExecuteReader();
        if (dr.Read())
        {
            Session["email"] = txtemail.Text;
            Session["userid"] = dr["id"].ToString();
            Response.Redirect("home.aspx");
        }
        else
        {
            ScriptManager.RegisterClientScriptBlock(this, this.GetType(), "alertMessage",
"alert('Wrong Credentials, Please Try Again')", true);
        }
        con.Close();
    }
}
```

registrationPage.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/masterpage.master"
AutoEventWireup="true" CodeFile="registerationpage.aspx.cs" Inherits="Default2" %>

<%-- Add content controls here --%>
<asp:Content ID="Content1" runat="server"
    contentplaceholderid="ContentPlaceHolder1">
 <script type="text/javascript">




     function test() {


         var txt = document.getElementById('<%=txtpassword.ClientID%>');
         var rtxt = document.getElementById('<%=txtretypepassword.ClientID%>');
         //it var filter = /^(?=.*\d)(?=.*[!@#$%^&*])(?=.*[a-z])(?=.*[A-Z]).{8,}$/;
         var filter = "^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[$@$!%*?&])[A-Za-
z\d$@$!%*?&]{8,}"

         var test_bool = filter.test(txt.value);

         if (txt.value == "") {

             alert('Please enter valid Password ');
             txt.focus();
             return false;
         }
         else if (test_bool == false) {

             alert('Please enter valid Password with Alphabets and Special Characters
Combination');
              txt.focus();
             return false;
         }
         else if (txt.value != rtxt.value) {
             alert('Please Match The Password');
              txt.focus();
             return false;
         }
     }


    </script>

 <center>
 <div style="background-color:crimson; margin-top:10px; margin-bottom:10px; padding-
bottom:10px;width:1000px ">
     <h1 style="text-align:center; color:black; font-family:cambria">Create New
Account</h1>
    <center> <table style="text-align:center" cellpadding="15">
     <tr>
     <td>
          <asp:TextBox ID="txtfirstname" CssClass="txtEmailPswd" runat="server"
placeholder="First Name"></asp:TextBox><br />
     <asp:TextBox ID="txtlastname" CssClass="txtEmailPswd" runat="server"
placeholder="Last Name"></asp:TextBox><br />

          <asp:TextBox ID="txtemail" CssClass="txtEmailPswd" runat="server"
placeholder="E-Mail"></asp:TextBox><br />
           <asp:TextBox ID="txtaddress" CssClass="txtEmailPswd" runat="server"
placeholder="Address"></asp:TextBox><br />

     </td>
     <td>
      <asp:TextBox ID="txtpassword" CssClass="txtEmailPswd" runat="server"
             placeholder="Passwor(Min. 8 Char., 1 Uppercase,1 Symbol)"
TextMode="Password"></asp:TextBox><br />

          <asp:TextBox ID="txtretypepassword" CssClass="txtEmailPswd" runat="server"
             placeholder="Retype Password" TextMode="Password"></asp:TextBox><br />

          <asp:TextBox ID="txtcity" CssClass="txtEmailPswd" runat="server"
placeholder="City"></asp:TextBox><br />
     <asp:DropDownList ID="ddlstate" CssClass="txtEmailPswd" runat="server">
            <asp:ListItem Text = "Select State" Value = "-1"></asp:ListItem>
        <asp:ListItem Text = "Virginia" Value = "1"></asp:ListItem>
        <asp:ListItem Text = "Minnesota" Value = "2"></asp:ListItem>
        <asp:ListItem Text = "NewJersey" Value = "3"></asp:ListItem>
        <asp:ListItem Text = "NewYork" Value = "4"></asp:ListItem>
        <asp:ListItem Text = "Boston" Value = "5"></asp:ListItem>
        <asp:ListItem Text = "Illinois" Value = "6"></asp:ListItem>
        <asp:ListItem Text = "Texas" Value = "7"></asp:ListItem>
        <asp:ListItem Text = "Washington" Value = "8"></asp:ListItem>
        <asp:ListItem Text = "MaryLand" Value = "9"></asp:ListItem>
        <asp:ListItem Text = "California" Value = "10"></asp:ListItem>
     </asp:DropDownList><br />

     </td>
     </tr>
     </table>
      <asp:Button ID="Button1" runat="server" Text="Create Account" OnClientClick="return
test();" style="width:200px; height:50px"
         onclick="Button1_Click" /><br /><br />

      Existing Member?<a href="login.aspx" class="btnSignUp"> Sign in</a></center>

 </div></center>

</asp:Content>
```

registrationPage.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Configuration;
public partial class Default2 : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
    SqlConnection con = new
SqlConnection(ConfigurationManager.ConnectionStrings["cn"].ConnectionString);
    protected void Button1_Click(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("insert into
tblregistration(firstname,lastname,email,password,address,city,state) values('" +
txtfirstname.Text + "','" + txtlastname.Text + "','" + txtemail.Text + "','" +
txtpassword.Text + "','" + txtaddress.Text + "','" + txtcity.Text + "','" +
ddlstate.SelectedItem.ToString() + "')", con);
        cmd.ExecuteNonQuery();
        con.Close();
        Response.Redirect("regsucess.aspx");
    }
}
```

myAccount.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/userMaster.master"
AutoEventWireup="true" CodeFile="myaccount.aspx.cs" Inherits="myaccount" %>
```

```
<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

     <h3 style="text-align:left; color:black; font-family:cambria">My Account</h3>
    <center> <table style="text-align:center; width:100%" cellpadding="15">
     <tr>
     <td>
     <table cellpadding="2" >
     <tr>
     <td>
     First Name
     </td>
     <td><asp:TextBox ID="txtfirstname" CssClass="txtEmailPswd2" runat="server"
             placeholder="First Name" Enabled="False"></asp:TextBox>
     </td>
     </tr>

     <tr>
     <td>Last Name</td>
     <td><asp:TextBox ID="txtlastname" CssClass="txtEmailPswd2" runat="server"
             placeholder="Last Name" Enabled="False"></asp:TextBox></td>
     </tr>

     <tr>
     <td>  Email</td>
     <td> <asp:TextBox ID="txtemail" CssClass="txtEmailPswd2" runat="server"
             ReadOnly="true" placeholder="E-Mail" Enabled="False" ></asp:TextBox></td>
     </tr>

     <tr>
     <td>Address</td>
     <td>    <asp:TextBox ID="txtaddress" CssClass="txtEmailPswd2" runat="server"
             placeholder="Address" Enabled="False"></asp:TextBox>
     </td>
     </tr>

     </table>

     </td>
     <td>

     <table cellpadding="2" >
     <tr>
     <td>Password</td>
     <td>
         <asp:TextBox ID="txtpassword" CssClass="txtEmailPswd2" runat="server"
             placeholder="Password" Enabled="False"></asp:TextBox></td>

     </tr>
     <tr>
     <td>City</td>
     <td>
      <asp:TextBox ID="txtcity" CssClass="txtEmailPswd2" runat="server"
             placeholder="City" Enabled="False"></asp:TextBox>
     </td>
     </tr>

     <tr>
     <td>State</td>
     <td>
     <asp:DropDownList ID="ddlstate" CssClass="txtEmailPswd2" runat="server"
             Enabled="False">
             <asp:ListItem Text = "Select State" Value = "-1"></asp:ListItem>
        <asp:ListItem Text = "Virginia" Value = "1"></asp:ListItem>
        <asp:ListItem Text = "Minnesota" Value = "2"></asp:ListItem>
        <asp:ListItem Text = "NewJersey" Value = "3"></asp:ListItem>
        <asp:ListItem Text = "NewYork" Value = "4"></asp:ListItem>
        <asp:ListItem Text = "Boston" Value = "5"></asp:ListItem>
        <asp:ListItem Text = "Illinois" Value = "6"></asp:ListItem>
        <asp:ListItem Text = "Texas" Value = "7"></asp:ListItem>
        <asp:ListItem Text = "Washington" Value = "8"></asp:ListItem>
        <asp:ListItem Text = "MaryLand" Value = "9"></asp:ListItem>
        <asp:ListItem Text = "California" Value = "10"></asp:ListItem>
     </asp:DropDownList>
     </td>
     </tr>
     <tr>
     <td>
        <asp:Button ID="Button1" runat="server" Text="Edit"
         onclick="Button1_Click" />
     </td>
     <td>
        <asp:Button ID="Button2" runat="server" Text="Save"
         onclick="Button2_Click" />
     </td>
     </tr>
     </table>


     </td>
     </tr>
     </table>


</center>
</asp:Content>
```

myAccount.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Configuration;
using System.Data;
public partial class myaccount : System.Web.UI.Page
{
    SqlConnection con = new
SqlConnection(ConfigurationManager.ConnectionStrings["cn"].ConnectionString);
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {

            con.Open();
            SqlCommand cmd = new SqlCommand("select * from tblregistration where id='"+
Session["userid"]+"'", con);
            SqlDataReader dr = cmd.ExecuteReader();
            while (dr.Read())
            {
              txtfirstname.Text=dr["firstname"].ToString();
                txtlastname.Text=dr["lastname"].ToString();
                txtemail.Text=dr["email"].ToString();
                txtpassword.Text=dr["password"].ToString();
                txtaddress.Text=dr["address"].ToString();
               txtcity.Text=dr["city"].ToString();
                ddlstate.SelectedValue=dr["state"].ToString();//) values('" +
txtfirstname.Text + "','" + txtlastname.Text + "','" + txtemail.Text + "','" +
txtpassword.Text + "','" + txtaddress.Text + "','" + txtcity.Text + "','" +
ddlstate.SelectedItem.ToString() + "')", con);

            }
            con.Close();

        }
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        txtaddress.Enabled = true;
        txtcity.Enabled = true;
        ddlstate.Enabled = true;
        txtfirstname.Enabled = true;
        txtlastname.Enabled = true;
        txtpassword.Enabled = true;

    }
    protected void Button2_Click(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("update tblregistration set firstname='" +
txtfirstname.Text + "',lastname='" + txtlastname.Text + "',email='" + txtemail.Text +
"',password='" + txtpassword.Text + "',address='" + txtaddress.Text + "',city='" +
txtcity.Text + "',state='" + ddlstate.SelectedValue.ToString() + "' where id='" +
Session["userid"].ToString() + "'", con);
        cmd.ExecuteNonQuery();
        con.Close();
        txtaddress.Enabled = false;
        txtcity.Enabled = false;
        ddlstate.Enabled = false;
        txtfirstname.Enabled = false;
        txtlastname.Enabled = false;
        txtpassword.Enabled = false;
    }
}
```

Buyticket.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/userMaster.master"
AutoEventWireup="true" CodeFile="buyticket.aspx.cs" Inherits="buyticket" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
    <script type="text/javascript">
        function calculate() {

            var perticketprice =
document.getElementById('<%=txtperticketprice.ClientID%>').value;

            var requirednumberoftickets =
document.getElementById('<%=txtrequirednumberoftickets.ClientID%>').value;

            var cost = document.getElementById('<%=txtcost.ClientID%>').value;

            cost = perticketprice * requirednumberoftickets;
            document.getElementById('<%=txtcost.ClientID%>').value = cost;

        }
    </script>
    <table width="100%">


<tr>
<td colspan="2" style="text-align: left">
    <strong>Buy Tickets</strong></td>
</tr>

<tr>
<td>
Select Your Level
</td>
<td>
    <asp:DropDownList ID="ddllevel" runat="server" CssClass="txtEmailPswd"
        AutoPostBack="True" onselectedindexchanged="ddllevel_SelectedIndexChanged">

    </asp:DropDownList>
</td>
</tr>
<tr>
<td>
Per Ticket Price($)
</td>
<td>
    <asp:TextBox ID="txtperticketprice" runat="server"
CssClass="txtEmailPswd"></asp:TextBox>
</td>
</tr>
<tr>
<td>
Req. No. Of Tickets
</td>
<td>
```

```
   <asp:TextBox ID="txtrequirednumberoftickets" runat="server" CssClass="txtEmailPswd" 
onchange="calculate();"></asp:TextBox><input type="button" value="calculate" 
onclick="calculate();" />
</td>
</tr>

<tr>
<td>
Total Cost($)
</td>
<td>
   <asp:TextBox ID="txtcost" runat="server" CssClass="txtEmailPswd"></asp:TextBox>
</td>
</tr>



<tr>
<td>

</td>
<td>
    <asp:ImageButton
    ID="PayPalBtn"
    runat="server"
    ImageUrl="https://www.paypalobjects.com/en_GB/i/btn/btn_buynow_LG.gif"
    onclick="PayPalBtn_Click" />
</td>
</tr>
</table>
</asp:Content>
```

Buyticket.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Configuration;
using System.Text;
using System.Data;
using System.Net.Mail;
public partial class buyticket : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            con.Open();
            SqlCommand cmd = new SqlCommand("select * from tbllevel join 
tblsetpricestolevel on tbllevel.id=tblsetpricestolevel.levelid and eventid='" + 
Request.QueryString["id"] + "'", con);
```

```
            SqlDataAdapter adp = new SqlDataAdapter(cmd);
            DataSet ds = new DataSet();
            adp.Fill(ds);
            ddllevel.DataSource = ds;
            ddllevel.DataTextField = "levelname";
            ddllevel.DataValueField = "id";
            ddllevel.DataBind();
            con.Close();
            ddllevel.Items.Insert(0, "Select");
        }
    }
    SqlConnection con = new
SqlConnection(ConfigurationManager.ConnectionStrings["cn"].ConnectionString);
    protected void Button1_Click(object sender, EventArgs e)
    {

        ///  txtdated.Text = "";
        //  txteventname.Text = "";
        //  ddlvenue.SelectedIndex = 0;
        // ddlCATEGORY.SelectedIndex = 0;
        //ScriptManager.RegisterClientScriptBlock(this, this.GetType(), "alertMessage",
"alert('Event Added Successfully')", true);
    }
    protected void ddllevel_SelectedIndexChanged(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("select * from tbllevel join tblsetpricestolevel
on tbllevel.id=tblsetpricestolevel.levelid and levelid='" +
ddllevel.SelectedValue.ToString() + "' and eventid='" + Request.QueryString["id"] + "'",
con);
        SqlDataReader dr = cmd.ExecuteReader();
        while (dr.Read())
        {
            txtperticketprice.Text = dr["price"].ToString();
        }
        con.Close();
    }
    protected void PayPalBtn_Click(object sender, ImageClickEventArgs e)
    {
        Session["eventid"] = Request.QueryString["id"];
        {
            con.Open();
            SqlCommand cmd = new SqlCommand("insert into
tblbuytickets(level1,userid,eventid,dated,ticketprice,requirednumberoftickets,status,tota
lcost) values('" + ddllevel.SelectedItem.ToString() + "','" +
Session["userid"].ToString() + "','" + Request.QueryString["id"] + "','" +
System.DateTime.Now.ToShortDateString() + "','" + txtperticketprice.Text + "','" +
txtrequirednumberoftickets.Text + "','Pending','" + txtcost.Text + "')", con);
            cmd.ExecuteNonQuery();
            con.Close();
        }
        {
            using (MailMessage mm = new MailMessage("eticketingwebsite@gmail.com",
Session["email"].ToString()))
            {

                mm.CC.Add(new MailAddress("scsutickets@gmail.com"));
                mm.Subject = Session["email"] + "'s Payment Processing is in the Way";
                mm.Body = Session["email"] + "'s Payment Processing is in the Way";

                mm.IsBodyHtml = false;
                SmtpClient smtp = new SmtpClient();
                smtp.Host = "smtp.gmail.com";
                smtp.EnableSsl = true;
                System.Net.NetworkCredential NetworkCred = new
System.Net.NetworkCredential("eticketingwebsite@gmail.com", "eticket@123");
                smtp.UseDefaultCredentials = true;
                smtp.Credentials = NetworkCred;
                smtp.Port = 587;
                smtp.Send(mm);
                ClientScript.RegisterStartupScript(GetType(), "alert", "alert('Email
sent.');", true);
            }
        }

        string business = "eticketing@gmail.com";
        string itemName = "Buy Tickets";
        double itemAmount = Convert.ToDouble(txtcost.Text);
        string currencyCode = "USD";

        //StringBuilder ppHref = new StringBuilder();

        //ppHref.Append("https://www.paypal.com/cgi-bin/webscr?cmd=_xclick");
        //ppHref.Append("&business=" + business);
        //ppHref.Append("&item_name=" + itemName);
        //ppHref.Append("&amount=" + itemAmount.ToString("#.00"));
        //ppHref.Append("&currency_code=" + currencyCode);

        //Response.Redirect(ppHref.ToString(), true);
        // Response.Redirect(RedirecttoPaypal.getItemNameAndCost(itemName,
itemAmount.ToString("#.00"), "", "", "", ""));


        decimal price = Convert.ToDecimal(txtperticketprice.Text);
        string returnURL = "";
        // returnURL += "https://www.paypal.com/xclick/business=" +
HttpUtility.UrlEncode(email);
        returnURL += "https://www.sandbox.paypal.com/cgi-bin/webscr?cmd=_xclick&business
=" + HttpUtility.UrlEncode("scsutickets.buyer@gmail.com");
        //Passing Item Name as dynamic
        returnURL += "&item_name=" + HttpUtility.UrlEncode(itemName);
        returnURL += "&item_number=" +
HttpUtility.UrlEncode(txtrequirednumberoftickets.Text);
        returnURL += "&amount=" + HttpUtility.UrlEncode(txtcost.Text);
        returnURL += "&currency=" + HttpUtility.UrlEncode("USD");
        returnURL += "&return=http://localhost:53147/success.aspx";
        returnURL += "&cancel_return=http://localhost:53147/canc.aspx";
        returnURL += "&data-env=sandbox";
        Response.Redirect(returnURL);
    }
}
```

ADMIN MODULE CODE:

Login.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="masterpage.master" AutoEventWireup="true"
CodeFile="login.aspx.cs" Inherits="login" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

    <div id="divText">
 <p style="font-family:Cambria; "  >
 <h1 STYLE="font-family:Cambria; font-size:40px">HELLO, ADMIN!!</h1>

 </div>
  <div class="divLoginContents" >

     <h2>SIGN IN</h2>
          <asp:TextBox ID="txtemail" CssClass="txtEmailPswd" runat="server"
placeholder="E-Mail"></asp:TextBox><br />
     <asp:TextBox ID="txtpassword" CssClass="txtEmailPswd" runat="server"
          placeholder="Password" TextMode="Password"></asp:TextBox><br />

        <div class="divbtnLogin" >
     <asp:Button ID="Button1" runat="server" Text="Log In"
         onclick="Button1_Click" Height="30px" Width="93px" /></div>
 </div>
</asp:Content>

<asp:Content ID="Content2" runat="server"
    contentplaceholderid="ContentPlaceHolder2">

    <table  width="100%" style="font-family:Cambria; font-size:20px; margin-top:10px" >
   <tr>
    <td style="text-align:center; color:white">
        <asp:HyperLink ID="l1" NavigateUrl="~/default.aspx"
runat="server">Home</asp:HyperLink></td>
       <td style="text-align:center; color:white">    <asp:HyperLink ID="l2"
NavigateUrl="~/events.aspx" runat="server">Events</asp:HyperLink></td>
          <td style="text-align:center; color:white">    <asp:HyperLink ID="l3"
NavigateUrl="~/contactus.aspx" runat="server">Contact</asp:HyperLink></td>
             <td style="text-align:center; color:white">    <asp:HyperLink ID="l4"
NavigateUrl="~/login.aspx" runat="server">Login</asp:HyperLink></td>
    </tr>
    </table>
</asp:Content>
```

Login.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Configuration;
public partial class login : System.Web.UI.Page
{
    SqlConnection con = new
SqlConnection(ConfigurationManager.ConnectionStrings["cn"].ConnectionString);
    protected void Page_Load(object sender, EventArgs e)
    {

        {
            con.Open();
            SqlCommand cmd = new SqlCommand("select * from tblcss where id=(select
max(id) from tblcss where cssitem='MenuBackColor')", con);
            SqlDataReader dr = cmd.ExecuteReader();
            while (dr.Read())
            {

                l1.Style.Add("background-color", dr["cssvalue"].ToString());
l2.Style.Add("background-color", dr["cssvalue"].ToString()); l3.Style.Add("background-
color", dr["cssvalue"].ToString()); l4.Style.Add("background-color",
dr["cssvalue"].ToString());
                //   menusimple1.Style.Add("background-color",
dr["cssvalue"].ToString());

                //   menusimple89.Style.Add("background-color",
dr["cssvalue"].ToString());
            }
            con.Close();
        }
        {
            con.Open();
            SqlCommand cmd = new SqlCommand("select * from tblcss where id=(select
max(id) from tblcss where cssitem='MenuForeColor')", con);
            SqlDataReader dr = cmd.ExecuteReader();
            while (dr.Read())
            {

                l1.Style.Add("color", dr["cssvalue"].ToString()); l2.Style.Add("color",
dr["cssvalue"].ToString()); l3.Style.Add("color", dr["cssvalue"].ToString());
l4.Style.Add("color", dr["cssvalue"].ToString());
                //   menusimple1.Style.Add("background-color",
dr["cssvalue"].ToString());

                //   menusimple1.Style.Add("background-color",
dr["cssvalue"].ToString());

                //   menusimple89.Style.Add("background-color",
dr["cssvalue"].ToString());
            }
            con.Close();
        }

    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("select * from tbladmin where username='" + 
txtemail.Text + "' and password='" + txtpassword.Text + "'", con);
        SqlDataReader dr = cmd.ExecuteReader();
        if (dr.Read())
        {
            Session["email"] = txtemail.Text;
            Response.Redirect("home.aspx");
        }
        else
        {
            ScriptManager.RegisterClientScriptBlock(this, this.GetType(), "alertMessage", 
"alert('Wrong Credentials, Please Try Again')", true);
        }
        con.Close();
    }
}
```

Addnewevent.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/cms_Admin/adminMaster2.master" 
AutoEventWireup="true" CodeFile="addnewevent.aspx.cs" Inherits="cms_Admin_addnewevent" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
    <table style="margin-left:30px" width="100%">
<tr >
<td colspan="2" style="text-align: left">
    <strong>Add New Event</strong></td>
</tr>

<tr>
<td>
Category
</td>
<td>
    <asp:DropDownList ID="ddlCATEGORY" runat="server" CssClass="txtEmailPswd">
        <asp:ListItem>Select Category</asp:ListItem>
        <asp:ListItem>Athletics</asp:ListItem>
        <asp:ListItem>Cultural</asp:ListItem>
        <asp:ListItem>Entertainment</asp:ListItem>
        <asp:ListItem>Recreation</asp:ListItem>
        <asp:ListItem>Confrences and Workshops</asp:ListItem>
    </asp:DropDownList>
</td>
</tr>
<tr>
<td>
Event Name
</td>
<td>
    <asp:TextBox ID="txteventname" runat="server" CssClass="txtEmailPswd"></asp:TextBox>
</td>
</tr>
<tr>
<td>
Venue Name
</td>
```

```
<td>
    <asp:DropDownList ID="ddlvenue" runat="server" CssClass="txtEmailPswd">
    </asp:DropDownList>
</td>
</tr>
<tr>
<td>
Dated
</td>
<td>
    <asp:TextBox ID="txtdated" runat="server" CssClass="txtEmailPswd"></asp:TextBox>
</td>
</tr>
<tr>
<td>

</td>
<td>
    <asp:Button ID="Button1" runat="server" Text="Submit" onclick="Button1_Click" />
</td>
</tr>
</table>
</asp:Content>
```

Addnewevent.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Configuration;
using System.Data;
public partial class cms_Admin_addnewevent : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            bind();
        }
    }SqlConnection con = new
SqlConnection(ConfigurationManager.ConnectionStrings["cn"].ConnectionString);
    public void bind()
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("select * from tblvenue", con);
        SqlDataAdapter adp = new SqlDataAdapter(cmd);
        DataSet ds = new DataSet();
        adp.Fill(ds);
        ddlvenue.DataSource = ds;
        ddlvenue.DataTextField = "venuename";
        ddlvenue.DataValueField = "id";
        ddlvenue.DataBind();
        con.Close();
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        {
            con.Open();
            SqlCommand cmd = new SqlCommand("delete from  tblevents where 
id='"+Request.QueryString["id"]+"'", con);
            cmd.ExecuteNonQuery();
            con.Close();
        }


        {
            con.Open();
            SqlCommand cmd = new SqlCommand("insert into 
tblevents(category,eventname,venue,dated) values('" + ddlCATEGORY.SelectedItem.ToString() 
+ "','" + txteventname.Text + "','" + ddlvenue.SelectedValue.ToString() + "','" + 
txtdated.Text + "')", con);
            cmd.ExecuteNonQuery();
            con.Close();
        }
        txtdated.Text = "";
        txteventname.Text = "";
        ddlvenue.SelectedIndex = 0;
        ddlCATEGORY.SelectedIndex = 0;
        ScriptManager.RegisterClientScriptBlock(this, this.GetType(), "alertMessage", 
"alert('Event Added Successfully')", true);
    }
}
```

Csssetting.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/cms_Admin/adminMaster2.master" 
AutoEventWireup="true" CodeFile="csssetting.aspx.cs" Inherits="cms_Admin_csssetting" %>

<%@ Register assembly="AjaxControlToolkit" namespace="AjaxControlToolkit" tagprefix="cc1" 
%>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
    <table width="100%">


<tr>
<td colspan="2" style="text-align: left">
    <strong>CSS Settings</strong><cc1:ToolkitScriptManager
        ID="ToolkitScriptManager1" runat="server">
    </cc1:ToolkitScriptManager>
    </td>
</tr>

<tr>
<td>
Top Menu Background Color(Color Code eg 000000)
</td>
```

```
<td>
    <asp:TextBox ID="txtmenubackcolor" runat="server"
CssClass="txtEmailPswd"></asp:TextBox>
    <asp:LinkButton ID="LinkButton5" runat="server"
onclick="LinkButton5_Click">Save</asp:LinkButton>
    <cc1:ColorPickerExtender ID="TextBox1_ColorPickerExtender" runat="server"
        Enabled="True" TargetControlID="txtmenubackcolor">
    </cc1:ColorPickerExtender>
</td>
</tr>
<tr>
<td>
Top Menu Foreground Color
</td>
<td>
    <asp:TextBox ID="txtforecolor" runat="server" CssClass="txtEmailPswd"></asp:TextBox>
    <cc1:ColorPickerExtender ID="txtforecolor_ColorPickerExtender" runat="server"
        Enabled="True" TargetControlID="txtforecolor">
    </cc1:ColorPickerExtender>
    <cc1:ColorPickerExtender ID="ColorPickerExtender1" runat="server"
        Enabled="True" TargetControlID="txtforecolor">
    </cc1:ColorPickerExtender>
    <asp:LinkButton ID="LinkButton6" runat="server"
onclick="LinkButton6_Click">Save</asp:LinkButton>
</td>
</tr>
<tr>
<td>
Side Nav BackColor
</td>
<td>
   &nbsp;<asp:TextBox ID="txtsidenavbackcolor" runat="server"
        CssClass="txtEmailPswd"></asp:TextBox>
    <cc1:ColorPickerExtender ID="txtsidenavbackcolor_ColorPickerExtender"
        runat="server" Enabled="True" TargetControlID="txtsidenavbackcolor">
    </cc1:ColorPickerExtender>
    <asp:LinkButton ID="LinkButton7" runat="server"
onclick="LinkButton7_Click">Save</asp:LinkButton>
</td>
</tr>

<tr>
<td>
Side Nav Fore Color
</td>
<td>
    <asp:TextBox ID="txtsidenavforecolor" runat="server"
CssClass="txtEmailPswd"></asp:TextBox>
    <cc1:ColorPickerExtender ID="txtsidenavforecolor_ColorPickerExtender"
        runat="server" Enabled="True" TargetControlID="txtsidenavforecolor">
    </cc1:ColorPickerExtender>
    <asp:LinkButton ID="LinkButton8" runat="server"
onclick="LinkButton8_Click">Save</asp:LinkButton>
</td>
</tr>
```

```
<tr>
<td>

</td>
<td>
    &nbsp;</td>
</tr>
</table>
</asp:Content>
```

Csssettings.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Configuration;

public partial class cms_Admin_csssetting : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
    SqlConnection con = new
SqlConnection(ConfigurationManager.ConnectionStrings["cn"].ConnectionString);
    protected void LinkButton5_Click(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("insert into tblcss(cssitem,cssvalue)
values('MenuBackColor','#" + txtmenubackcolor.Text + "')",con);
        cmd.ExecuteNonQuery();
        con.Close();

        ScriptManager.RegisterClientScriptBlock(this, this.GetType(), "alertMessage",
"alert('Added Successfully')", true);
    }
    protected void LinkButton6_Click(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("insert into tblcss(cssitem,cssvalue)
values('MenuForeColor', '#" + txtforecolor.Text+ "')", con);
        cmd.ExecuteNonQuery();
        con.Close();

        ScriptManager.RegisterClientScriptBlock(this, this.GetType(), "alertMessage",
"alert('Added Successfully')", true);
    }
    protected void LinkButton7_Click(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("insert into tblcss(cssitem,cssvalue) 
values('SideMenuBackColor', '#" + txtsidenavbackcolor.Text+ "')", con);
        cmd.ExecuteNonQuery();
        con.Close();

        ScriptManager.RegisterClientScriptBlock(this, this.GetType(), "alertMessage", 
"alert('Added Successfully')", true);
    }
    protected void LinkButton8_Click(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("insert into tblcss(cssitem,cssvalue) 
values('SideMenuForeColor', '#" + txtsidenavforecolor.Text + "')", con);
        cmd.ExecuteNonQuery();
        con.Close();

        ScriptManager.RegisterClientScriptBlock(this, this.GetType(), "alertMessage", 
"alert('Added Successfully')", true);
    }
}
```

Databasesetting.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/cms_Admin/adminMaster2.master" 
AutoEventWireup="true" CodeFile="databasesetting.aspx.cs" 
Inherits="cms_Admin_databasesetting" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
    <div>
<table style="margin-left:30px"  cellpadding=”2” cellspacing=”1” border=”1” 
style=”border-collapse: collapse;”>

<tr>

<th style=”background-color:#e0e0e0;”>

Backup and Restore SQL Server

<br />

database using ASP.NET</th

</tr>

<tr>

<td align="Center">

List of Tables</td>

</tr>

<tr>

<td align="Center">
```

```
<asp:ListBox ID="ListBox1" Rows="10" runat="Server" DataTextField="table_name"
DataValueField="table_name">

</asp:ListBox>

</td>

</tr>

<tr>

<td align="Center”>

<span onclick="return confirm(‘Are you sure to backup selected table?’)">

<asp:Button ID="btnBackup" runat="Server" Text="Backup" OnClick="BackUpNow" />

</span>

<span onclick="return confirm(‘Are you sure to restore selected table?’)">

<asp:Button ID="btnRestore" runat="Server" Text="Restore" OnClick="RestoreNow" />

</span>

</td>

</tr>

</table>



<asp:Label ID="lblMessage" runat="Server" EnableViewState="False"
ForeColor="Blue"></asp:Label>
</div>
</asp:Content>
```

Databasesetting.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Configuration;
//using Microsoft.SqlServer.Management.Smo;
//using Microsoft.SqlServer.Management.Common;


using System.Data;
public partial class cms_Admin_databasesetting : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {

            PopulateDatabaseTables();

        }

    }SqlConnection con = new
SqlConnection(ConfigurationManager.ConnectionStrings["cn"].ConnectionString);
    private void PopulateDatabaseTables()
    {

        string tableName = string.Empty;

        string sql = "SELECT *, name AS table_name FROM sys.tables WHERE Type ='U' ORDER
BY table_name";

        using (SqlConnection conn = new
SqlConnection(ConfigurationManager.ConnectionStrings["cn"].ConnectionString))
        {

            using (DataTable table = new DataTable())
            {

                conn.Open();

                using (SqlDataAdapter dAd = new SqlDataAdapter(sql, conn))
                {

                    dAd.Fill(table);

                }

                ListBox1.DataSource = table;

                ListBox1.DataBind();

            }
        }
    }

protected void BackUpNow(object sender, EventArgs e)

{

string tableName = ListBox1.SelectedValue;

using (DataSet dSetBackup = new DataSet())

{

    using (SqlConnection conn = new
SqlConnection(ConfigurationManager.ConnectionStrings["cn"].ConnectionString))

{
```

```
using (SqlDataAdapter dAd = new SqlDataAdapter("select * from "+ tableName, conn))

{

dAd.Fill(dSetBackup, tableName);

}

}

dSetBackup.WriteXml(Server.MapPath("~/backup/" + tableName + ".xml"));

lblMessage.Text = "Backup for table <b>" + tableName + "</b> successful!";

}

}

    protected void RestoreNow(object sender, EventArgs e)
    {
        string restoreConnStr = 
ConfigurationManager.ConnectionStrings["cn"].ConnectionString;

        string tableName = ListBox1.SelectedValue;

        using (SqlConnection conn = new SqlConnection(restoreConnStr))
        {

            using (DataSet dSetBackup = new DataSet())
            {

                // get the schema of the selected table from the database to restore

                using (SqlDataAdapter dAd = new SqlDataAdapter("select * from " + 
tableName, conn))
                {

                    // Following line will get the schema of the selected table in the 
dSetBackup DataSet

                    dAd.Fill(dSetBackup, tableName);



                    // get the data for selected table from backed up XML file

                    using (DataSet dSet = new DataSet())
                    {

                        dSet.ReadXml(Server.MapPath("~/backup/" + tableName + ".xml"));

                        // Loop through all rows of the selected table and add into 
dSetBackup DataSet

                        foreach (DataRow row in dSet.Tables[0].Rows)
                        {
                            dSetBackup.Tables[0].NewRow();

                            dSetBackup.Tables[0].Rows.Add(row.ItemArray);

                        }


                        // Create a command builder to update dSetBackup DataSet

                        SqlCommandBuilder cmd = new SqlCommandBuilder(dAd);


                        // Following update command will push all added rows of
dSetBackup DataSet into the database

                        dAd.Update(dSetBackup, tableName); // We are done !!!

                    }

                }

                lblMessage.Text = "Restore of table <b>" + tableName + "</b>
successful!";

            }

        }
    }
    protected void btnBackup_Command(object sender, CommandEventArgs e)
    {

    }
}
```

Editevent.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/cms_Admin/adminMaster2.master"
AutoEventWireup="true" CodeFile="editevent.aspx.cs" Inherits="cms_Admin_editevent" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
      <table width="100%">
<tr>
<td colspan="2" style="text-align: left">
    <strong>Edit Event</strong></td>
</tr>

<tr>
<td>
Category
</td>
<td>
    <asp:DropDownList ID="ddlCATEGORY" runat="server" CssClass="txtEmailPswd">
        <asp:ListItem>Select Category</asp:ListItem>
        <asp:ListItem>Athletics</asp:ListItem>
        <asp:ListItem>Cultural</asp:ListItem>
        <asp:ListItem>Entertainment</asp:ListItem>
        <asp:ListItem>Recreation</asp:ListItem>
        <asp:ListItem>Confrences and Workshops</asp:ListItem>
    </asp:DropDownList>
</td>
</tr>
<tr>
<td>
Event Name
</td>
<td>
    <asp:TextBox ID="txteventname" runat="server" CssClass="txtEmailPswd"></asp:TextBox>
</td>
</tr>
<tr>
<td>
Venue Name
</td>
<td>
    <asp:DropDownList ID="ddlvenue" runat="server" CssClass="txtEmailPswd">
    </asp:DropDownList>
</td>
</tr>
<tr>
<td>
Dated
</td>
<td>
    <asp:TextBox ID="txtdated" runat="server" CssClass="txtEmailPswd"></asp:TextBox>
</td>
</tr>
<tr>
<td>

</td>
<td>
    <asp:Button ID="Button1" runat="server" Text="Submit" onclick="Button1_Click" />
</td>
</tr>
</table>
</asp:Content>
```

Editevent.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Configuration;
using System.Data;
public partial class cms_Admin_editevent : System.Web.UI.Page
{
    SqlConnection con = new
SqlConnection(ConfigurationManager.ConnectionStrings["cn"].ConnectionString);
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            bind();
         //   tblevents(category,eventname,venue,dated)
                con.Open();
                SqlCommand cmd = new SqlCommand("select * from tblevents where id='" +
Request.QueryString["id"] + "'", con);
                SqlDataReader dr = cmd.ExecuteReader();
                while (dr.Read())
                {
                    ddlCATEGORY.SelectedValue = dr["category"].ToString();
                    txteventname.Text = dr["eventname"].ToString();
                    txtdated.Text = dr["dated"].ToString();
                    ddlvenue.SelectedValue = dr["venue"].ToString();
                }
                con.Close();



        }
    }

    public void bind()
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("select * from tblvenue", con);
        SqlDataAdapter adp = new SqlDataAdapter(cmd);
        DataSet ds = new DataSet();
        adp.Fill(ds);
        ddlvenue.DataSource = ds;
        ddlvenue.DataTextField = "venuename";
        ddlvenue.DataValueField = "id";
        ddlvenue.DataBind();

        con.Close();
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        {
            con.Open();
            SqlCommand cmd = new SqlCommand("delete from tblevents where
id='"+Request.QueryString["id"]+"'", con);
            cmd.ExecuteNonQuery();
            con.Close();

        }


        {
            con.Open();
            SqlCommand cmd = new SqlCommand("insert into
tblevents(category,eventname,venue,dated) values('" + ddlCATEGORY.SelectedItem.ToString()
+ "','" + txteventname.Text + "','" + ddlvenue.SelectedValue.ToString() + "','" +
txtdated.Text + "')", con);
            cmd.ExecuteNonQuery();
            con.Close();
            txtdated.Text = "";
            txteventname.Text = "";
            ddlvenue.SelectedIndex = 0;
            ddlCATEGORY.SelectedIndex = 0;
            ScriptManager.RegisterClientScriptBlock(this, this.GetType(), "alertMessage",
"alert('Event Updated Successfully')", true);
        }
    }
}
```

Sendconfirmationemail.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/cms_Admin/adminmaster2.master"
AutoEventWireup="true" CodeFile="sendconfirmationmail.aspx.cs"
Inherits="cms_Admin_sendconfirmationmail" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
    <script type="text/javascript">

        function DeleteConfirm()
        {
            var Ans = confirm("Do you want to Delete Tickets Record for particular
user?");
            if (Ans)
            {
                return true;
            }
            else
            {
                return false;
            }
        }
</script>
    <table style="width: 100%">
    <tr>
        <td>
           <asp:Button ID="btnDeleteRecord" runat="server" BackColor="#E7CEB6"
Height="32px" OnClick="btnDeleteRecord_Click" Text="Delete" Width="64px" /> </td>
    </tr>
    <tr>
        <td style="height: 263px">
                  <asp:GridView    onrowcommand="GridView1_RowCommand" width="600px"
         CssClass="gridview" runat="server" ID="GridView1" GridLines="None"
CellPadding="4"
                    ForeColor="#333333" AutoGenerateColumns="False" style="font-size: x-
small"
                      AllowPaging="True"
onpageindexchanging="GridView1_PageIndexChanging"
                      PageSize="5">
                <AlternatingRowStyle BackColor="White" />
                      <Columns>
                           <asp:TemplateField>
                    <ItemTemplate>
                        <asp:CheckBox ID="chkDel" runat="server" />
                    </ItemTemplate>
                </asp:TemplateField>
                          <asp:BoundField DataField="userid" HeaderText="id" />
                          <asp:BoundField DataField="email" HeaderText="Email" />
                          <asp:BoundField DataField="level1" HeaderText="Level" />
                       <asp:TemplateField HeaderText="Dated">
<ItemTemplate>
<%#Convert.ToDateTime( Eval("dated")).ToShortDateString()%>
</ItemTemplate>
</asp:TemplateField>
                          <asp:TemplateField HeaderText="Per Ticket Price">
<ItemTemplate>
<b>$</b><%# Eval("ticketprice") %>
</ItemTemplate>
</asp:TemplateField>
                          <asp:BoundField DataField="requirednumberoftickets"
                              HeaderText="Req. No. Of Tickets" />
                          <asp:TemplateField HeaderText="Total Cost">
<ItemTemplate>
<b>$</b><%# Eval("totalcost") %>
</ItemTemplate>
</asp:TemplateField>
                          <asp:BoundField DataField="status" HeaderText="Status" />
                          <asp:TemplateField HeaderText="Send Email">
                              <ItemTemplate>
                                  <asp:LinkButton ID="LinkButton5" runat="server"
CommandName="sendemail" CommandArgument='<%#Eval("email") %>'>Send Email</asp:LinkButton>
                              </ItemTemplate>
                          </asp:TemplateField>
                      </Columns>
                <FooterStyle BackColor="#990000" Font-Bold="True" ForeColor="White" />
                <HeaderStyle BackColor="#990000" Font-Bold="True" ForeColor="White" />
                <PagerStyle BackColor="#FFCC66" ForeColor="#333333"
HorizontalAlign="Center" />
                <RowStyle BackColor="#FFFBD6" ForeColor="#333333" />
                <SelectedRowStyle BackColor="#FFCC66" Font-Bold="True" ForeColor="Navy"
/>
                <SortedAscendingCellStyle BackColor="#FDF5AC" />
                <SortedAscendingHeaderStyle BackColor="#4D0000" />
                <SortedDescendingCellStyle BackColor="#FCF6C0" />
                <SortedDescendingHeaderStyle BackColor="#820000" />
            </asp:GridView>
        </td>
    </tr>
    <tr>
        <td>
            &nbsp;</td>
    </tr>
</table>
</asp:Content>
```

Sendconfirmationemail.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Data;
using System.Configuration;
public partial class cms_Admin_sendconfirmationmail : System.Web.UI.Page
{
    SqlConnection con = new
SqlConnection(ConfigurationManager.ConnectionStrings["cn"].ConnectionString);
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            bind();
            btnDeleteRecord.Attributes.Add("onclick", "javascript:return
DeleteConfirm()");
        }

    }
    public void bind()
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("select * from tblregistration join tblbuytickets
on tblregistration.id=tblbuytickets.userid", con);
        SqlDataAdapter adp = new SqlDataAdapter(cmd);
        DataSet ds = new DataSet();
        adp.Fill(ds);
        GridView1.DataSource = ds;

        GridView1.DataBind();
        con.Close();
    }
    protected void GridView1_RowCommand(object sender, GridViewCommandEventArgs e)
    {
        if (e.CommandName == "sendemail")
        {
            Response.Redirect("sendemail.aspx?mailid="+e.CommandArgument.ToString()+"");
        }
    }
    protected void GridView1_PageIndexChanging(object sender, GridViewPageEventArgs e)
    {
        GridView1.PageIndex = e.NewPageIndex;
        bind();
    }
    protected void DeleteRecord(int id)
    {

        SqlCommand com = new SqlCommand("delete from tblbuytickets where userid=@ID",
con);
        com.Parameters.AddWithValue("@ID", id);
        con.Open();
        com.ExecuteNonQuery();
        con.Close();
    }
    protected void btnDeleteRecord_Click(object sender, EventArgs e)
    {
        foreach (GridViewRow grow in GridView1.Rows)
        {
            //Searching CheckBox("chkDel") in an individual row of Grid
            CheckBox chkdel = (CheckBox)grow.FindControl("chkDel");
            //If CheckBox is checked than delete the record with particular empid
            if (chkdel.Checked)
            {
                int empid = Convert.ToInt32(grow.Cells[1].Text);
                DeleteRecord(empid);
            }
        }
        //Displaying the Data in GridView
        bind();
    }
}
```

DATABASE DESIGN:

TblAdmin:

```
USE [eticketing]
GO

/****** Object:  Table [dbo].[tbladmin]    Script Date: 9/11/2017 6:40:23 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[tbladmin](
       [id] [int] IDENTITY(1,1) NOT NULL,
       [username] [nvarchar](max) NULL,
       [password] [nvarchar](max) NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
```

tblBuyTickets:

```
USE [eticketing]
GO

/****** Object:  Table [dbo].[tblbuytickets]    Script Date: 9/11/2017 6:47:29 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[tblbuytickets](
       [id] [int] IDENTITY(1,1) NOT NULL,
       [level1] [nvarchar](max) NULL,
       [userid] [int] NULL,
       [eventid] [int] NULL,
       [dated] [datetime] NULL,
       [ticketprice] [nvarchar](max) NULL,
       [requirednumberoftickets] [nvarchar](max) NULL,
       [status] [nvarchar](max) NULL,
       [totalcost] [nvarchar](max) NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
```

Tblcss:

```
USE [eticketing]
GO

/****** Object:  Table [dbo].[tblcss]    Script Date: 9/11/2017 6:48:04 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[tblcss](
       [id] [int] IDENTITY(1,1) NOT NULL,
       [cssitem] [nvarchar](max) NULL,
       [cssvalue] [nvarchar](max) NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
```

tblEvents:

```
USE [eticketing]
GO

/****** Object:  Table [dbo].[tblevents]    Script Date: 9/11/2017 6:48:31 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[tblevents](
       [id] [int] IDENTITY(1,1) NOT NULL,
       [category] [nvarchar](max) NULL,
       [eventname] [nvarchar](max) NULL,
       [venue] [int] NULL,
       [dated] [nvarchar](max) NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
```

tblLevels:

```
USE [eticketing]
GO

/****** Object:  Table [dbo].[tbllevel]    Script Date: 9/11/2017 6:49:19 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[tbllevel](
       [id] [int] IDENTITY(1,1) NOT NULL,
       [levelname] [nvarchar](max) NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
```

tblRegistration:

```
USE [eticketing]
GO

/****** Object:  Table [dbo].[tblregistration]    Script Date: 9/11/2017 6:50:24 PM
******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[tblregistration](
       [id] [int] IDENTITY(1,1) NOT NULL,
       [firstname] [nvarchar](max) NULL,
       [lastname] [nvarchar](max) NULL,
       [email] [nvarchar](max) NULL,
       [password] [nvarchar](max) NULL,
       [address] [nvarchar](max) NULL,
       [city] [nvarchar](max) NULL,
       [state] [nvarchar](max) NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
```

tblsetpricestolevel:

```
USE [eticketing]
GO

/****** Object:  Table [dbo].[tblsetpricestolevel]    Script Date: 9/11/2017 6:51:20 PM
******/
```

```
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[tblsetpricestolevel](
       [id] [int] IDENTITY(1,1) NOT NULL,
       [eventid] [int] NULL,
       [levelid] [int] NULL,
       [price] [nvarchar](max) NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
```

tblvenue:

```
USE [eticketing]
GO

/****** Object:  Table [dbo].[tblvenue]    Script Date: 9/11/2017 6:52:02 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[tblvenue](
       [id] [int] IDENTITY(1,1) NOT NULL,
       [venuename] [text] NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
```