

Governors State University

OPUS Open Portal to University Scholarship

All Capstone Projects

Student Capstone Projects

Spring 2023

FundRaiser

Sireesha Chitla

Follow this and additional works at: <https://opus.govst.edu/capstones>

For more information about the academic degree, extended learning, and certificate programs of Governors State University, go to http://www.govst.edu/Academics/Degree_Programs_and_Certifications/

Visit the [Governors State Computer Science Department](#)

This Capstone Project is brought to you for free and open access by the Student Capstone Projects at OPUS Open Portal to University Scholarship. It has been accepted for inclusion in All Capstone Projects by an authorized administrator of OPUS Open Portal to University Scholarship. For more information, please contact opus@govst.edu.

FUNDRAISER

By

Sireesha Chitla

B. Tech, JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD, 2019

GRADUATE CAPSTONE SEMINAR PROJECT

Submitted in partial fulfillment of the requirements.

For the Degree of Master of Science,

With a Major in Computer Science



Governors State University
University Park, IL 60484

2023

ABSTRACT

Fund Raiser is a website where users can start a campaign seeking public and private funding for a person, a project, or a cause. Users can use tools of this website to publish the campaign web pages and to market the fund-raising event. In addition, the website allows anyone (who is willing) to donate to any fundraising event.

There are different ways to seek funds for different activities today. Fund Raiser is a Java-based web application to get funds for events, floods, opening new offices, constructing school buildings, etc. There are different websites available in the market to perform this operation. Our Fund Raiser website can accept donations from other countries. A centralized database is maintained to store all the operations. A separate user interface is developed for the admin, donor, and user. Every actor must register with the system. After successful login, only they can perform various operations.

Table of Content

1	<i>Project Description</i>	1
1.1	Competitive Information	1
1.2	Relationship to Other Applications/Projects	1
1.3	Assumptions and Dependencies	1
1.4	Future Enhancements.....	1
1.5	Definitions and Acronyms	1
2	<i>Project Technical Description</i>	2
2.1	Application Architecture	2
2.2	Technical Architecture.....	3
2.3	Interactions with other applications	3
2.4	Capabilities	3
2.5	Risk Assessment and Management.....	3
3	<i>Project Requirements</i>	4
3.1	Identification of Requirements	4
3.2	Operations, Administration, Maintenance, and Provisioning (OAM&P).....	4
3.3	Security and Fraud Prevention.....	5
3.4	Release and Transition Plan.....	5
4	<i>Project Design Description</i>	5
5	<i>Internal/external Interface Impacts and Specification</i>	6
6	<i>Design Units Impacts</i>	11
6.1	Functional Area A/Design Unit A	12
6.1.1	<i>Functional Overview</i>	12
6.1.2	<i>Impacts</i>	12
7	<i>Open Issues</i>	13
8	<i>Acknowledgements</i>	13
9	<i>References</i>	13

1 Project Description

This website is designed to get funds from different donors. This application has three leading roles. The first is an administrator who adds fundraising events and maintains users. He can view or delete users' fundraising events. In addition, the admin can view details of any donations made. The second role is the donor. A donor is the one who donates money. Every donor has a unique id; after successful login, he can view different fund-raising events and pay an amount. The third role is the user. Every user must register with the system, and after successful login, he can add different fund-raising events and h and nation details too.

1.1 Competitive Information

Many sites in the market allow people to post their fund-raising events and get donations from different donors. Some of them are GoFundMe, Donorbox, and Donately. Our application is different from the existing ones. In our application, multiple users can post various fundraising events. So that donors can search for fundraising events based on location and category. Donors can complete the payment process. A centralized database is maintained to store all these operations securely.

1.2 Relationship to Other Applications/Projects

This project is related to other online fundraising projects available over the network, which helps to start these campaigns and accept donations from different locations. We developed this web page using open-source technologies like HTML, CSS, JavaScript, JSP, and JDBC.

1.3 Assumptions and Dependencies

Only registered users can access the system. Users and donors can register with the system. Multiple users can access it simultaneously. The administrator owns this application; he has all the privileges like adding/removing user fundraising events. We developed this web application using Java, which is platform-independent and server independent.

1.4 Future Enhancements

The project can be updated soon if any new requirement arises because it is flexible regarding expansion. In the future, we can add security algorithms for storing information related to funds. We can implement different search patterns to search for fundraising events. We can also implement online payment along with credit/debit card payments.

1.5 Definitions and Acronyms

Admin: He is the owner of this web application. Admin can add/view/delete fundraising events. He can view donation details also.

Donor: The one who can search for fund-raising events and donate money.

User: The one who adds fundraising events to the system and can view donations.

Fundraising events: Events added by the user and admin.

Donations: Amount sent by the donor.

2 Project Technical Description

The project contains three roles. Each actor is assigned operations specified in the proposal document. The roles are admin, donor, and user. The system design and technical requirements are discussed in this chapter.

2.1 Application Architecture

The proposed system is a 3-tier application. Specifically, our web application has three roles: admin, donor, and user. The following diagram shows the architecture diagram of our web application.

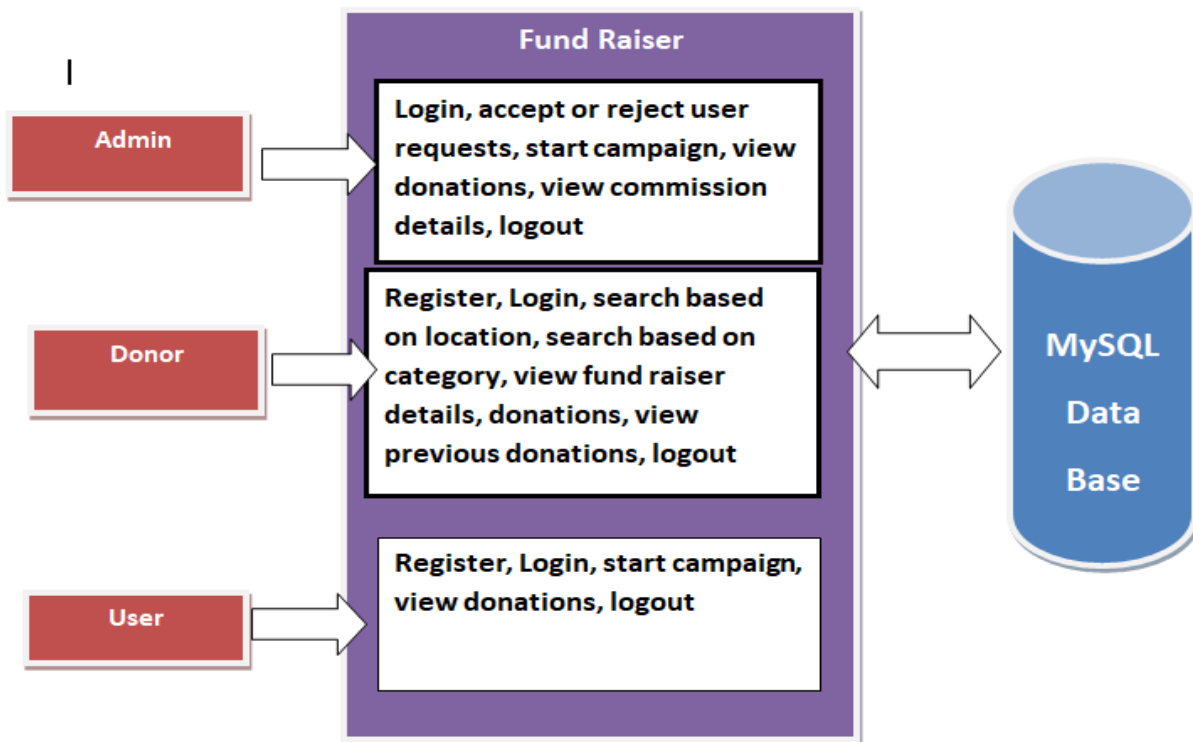


Figure 1: Architecture Diagram of Fund Raiser

2.2 Technical Architecture

The current application uses 3-tier architecture [1] as a prototype. In this model, the web browser acts as a client; the Tomcat server handles requesting an appointment and checking the visitor's status. Finally, a separate tier SQL server runs the database functions like storing and viewing details. Figure 2 is the technical architecture of our website.



Figure 2: Technical Architecture

2.3 Interactions with Other Applications

It can interact with other fundraising websites like GoFundMe, where donors can search for different fundraising events based on location and category. In addition, donors can view previous donation details, target amount, and target achieved.

2.4 Capabilities

This web application is developed using free source software. HTML, CSS, and Bootstrap are used to build application UI. MySQL is used as a backend database. All users can perform insert, delete, and update operations on a database. For example, whenever a user registers with the system, we perform an insert operation to store details. Whenever the admin deletes user fundraising events, we perform a delete operation on the database. To approve user fundraising events, we perform update operations. Finally, we perform a particular function on the database to display fundraising events or donation details.

2.5 Risk Assessment and Management

If a risk materializes, our team will quickly spot, evaluate, and monitor it through the project's completion. To lessen the impact on the system, they will recognize risks early and take steps to mitigate them. They also keep a risk management [2] log for future reference. There are three categories of risks: high, medium, and low. Risks are managed in order of importance, starting with the highest priority.

3 Project Requirements

3.1 Identification of Requirements

There are different websites available in the market for fundraising. In the existing system, everything is done manually. There is no database to store the details of donors and users. There needs to be a mechanism to track previous donations. The needs of donors, users, and administrators must be adequately served by the existing fundraising websites, which were created using technologies like PHP and ASP. They also did not offer practical procedures, such as deleting unnecessary information for user acceptance. The proposed system is a website used to provide better interaction between donors and fundraisers. This web application manages updates for both donors and users. It allows donors to search for different fundraising options. Donors can search based on category, location, etc. A rich UI is designed to perform these operations. Users registered on the website have the facility to use these services. Being an authorized user, he can publish fundraising details, and he can view donation details also.

3.2 Operations, Administration, Maintenance, and Provisioning (OAM&P)

The Donor can search for the fundraising campaigns based on location and category. The donor is the leading actor in this application. The primary goal is to perform search operations and view fundraising campaigns. Following are the functions performed by the donor.

- Register
- Login
- Search fundraising events.
- View campaign details.
- Send donations.
- View previous donation details.
- Logout

The other important role in the application is the user. He is responsible for adding new events. Following are the operations performed by the user.

- Login
- Add new fundraising events.
- View donations.
- Logout

Another role in the application is admin, who is responsible for the entire project. He will monitor fundraising events and users. He can accept donations, and he can view donation details. The following are the operations performed by the admin.

- Login
- Add Fundraising events.
- Accept/delete user fundraising events.
- View donation details.
- View registered users.
- View registered donors.
- Logout

3.3 Security and Fraud Prevention

Sensitive information is frequently posted on fundraising events, which could be exploited to steal an applicant's identity or commit financial fraud. We need to maintain a proper security mechanism to store data. Otherwise, data may be exposed to exploitation. We can detect any fraud operations with the following two steps.

- 1) Keep an eye out for the number of fund-raising events that the user posts.
- 2) Look at how many donations are being sent by a particular donor
- 3) If the target amount is higher, the admin can examine and, if necessary, delete that event.

A user may be engaging in fraud if they submit the same content several times, and their behavior is further evidenced if they are required to pay any additional fees. For example, it may be deemed fraudulent if a donor repeatedly gives the same amount to one user.

3.4 Release and Transition Plan

A three-tier architecture has been designed for this project. Our web application is server and platform-independent. It can therefore be installed on any server or operating system. All the necessary software is installed and prepared for testing to run the application. Future releases are planned according to the user's needs. Every three months following a successful launch, maintenance and upgrades take place.

4 Project Design Description

There are three main UIs developed in our application. The administrative UI is responsible for maintaining the entire system and database. This UI helps perform insert, delete, and update operations on the database. The user UI allows the users to store and retrieve data whenever required. We can perform various DML operations using this interface. The third UI is designed to perform donor operations. Donors can search and can view multiple fundraising events. After careful examination, the system is developed with three modules.

- 1) Admin
- 2) Donor
- 3) User

Admin module will perform the following operations.

- Login
- Add fundraising events.
- Accept/Reject user fundraising events.
- View user fundraising campaigns.
- View donations
- View registered users.
- View registered donors.
- Logout

The donor will perform the following operations.

- Register
- Login
- View fundraising events and can donate.
- View previous donations.
- Logout

Users will perform the following operations.

- Register
- Login
- Start fundraising events.
- View donations.
- Logout

5 Internal/external Interface Impacts and Specification

To design our UI, we have used different technologies to make user-friendly interfaces, like HTML, CSS, JavaScript, JSP, JDBC, and MySQL. We used HTML [3] to develop the user interface, CSS to apply styling, JavaScript to handle client-side validations, and Bootstrap to improve navigation. The back-end database is MySQL. We can easily carry out several SQL operations with this straightforward, open-source program. Java Server Pages (JSP) and Java Database Connectivity (JDBC) provide Middleware. Tomcat server is used for project deployment, an open-source web server designed to run Java-based web applications. We

developed our application using the waterfall paradigm [4], which is appropriate for brief, time-sensitive applications. The waterfall model utilized in this application is depicted in the diagram below.

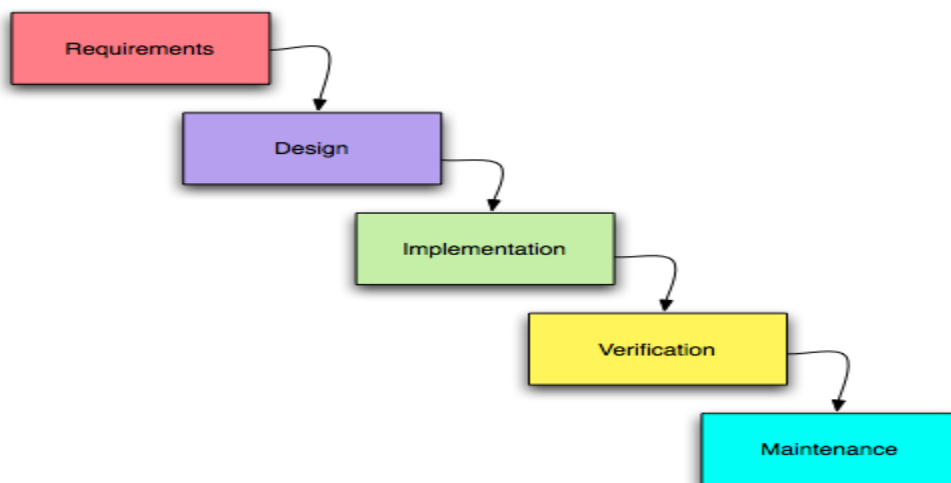


Figure 3: SDLC model for Fund Raiser

Screenshots

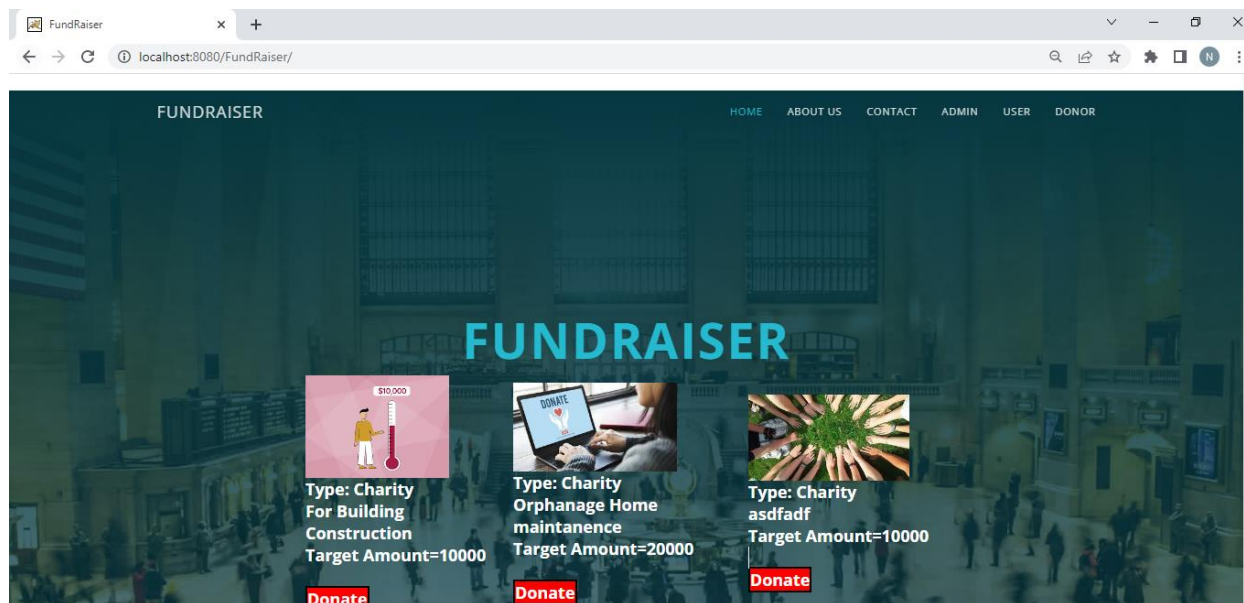


Figure 4: Home Page

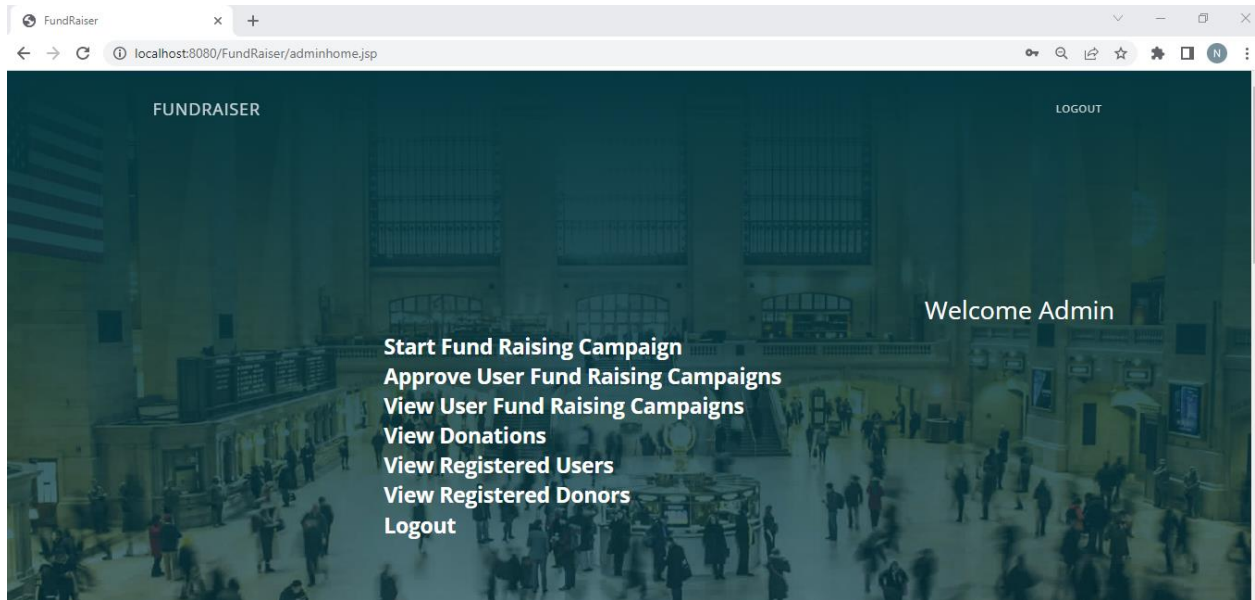


Figure 5: Admin Home Page

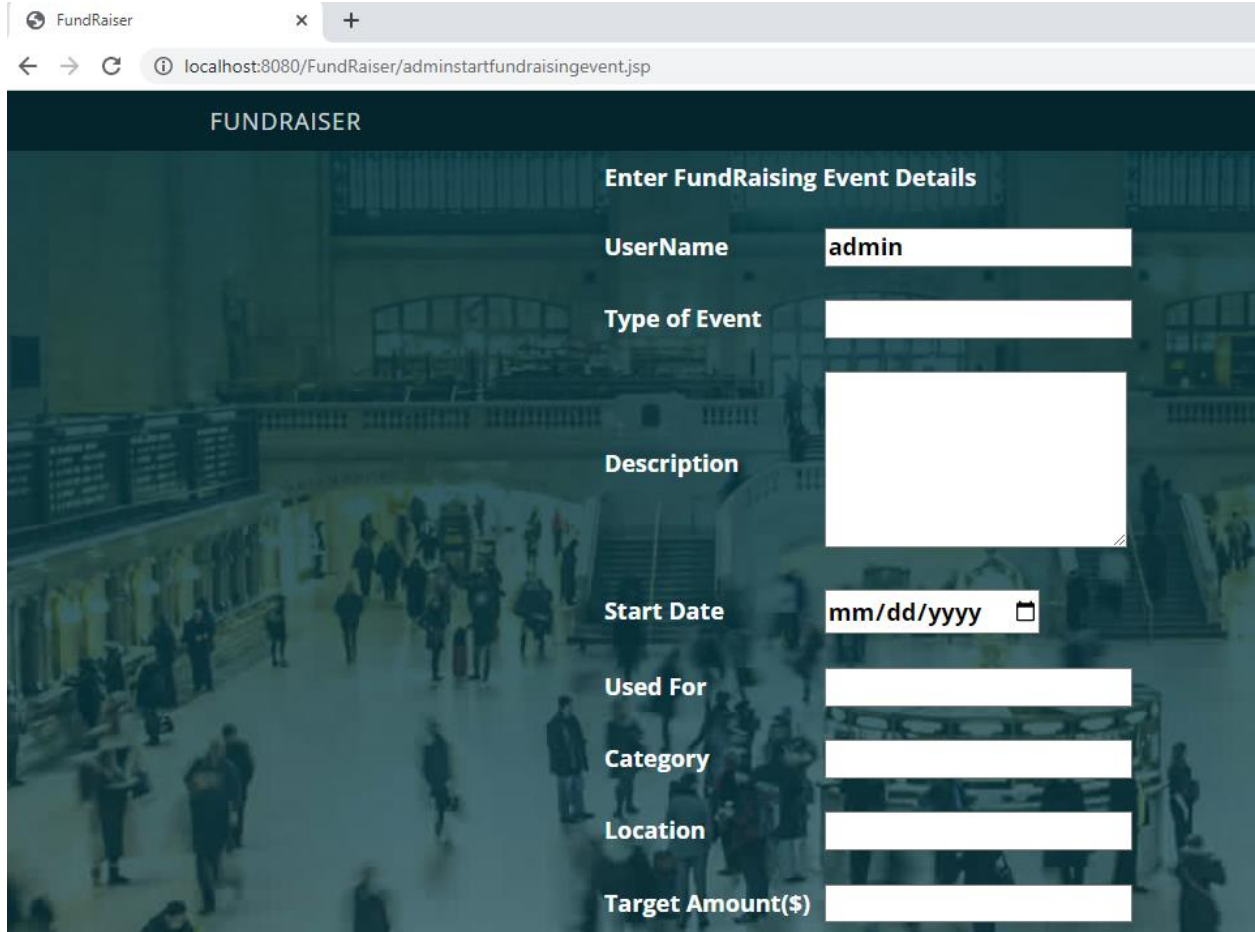


Figure 6: Admin adding fundraising events.

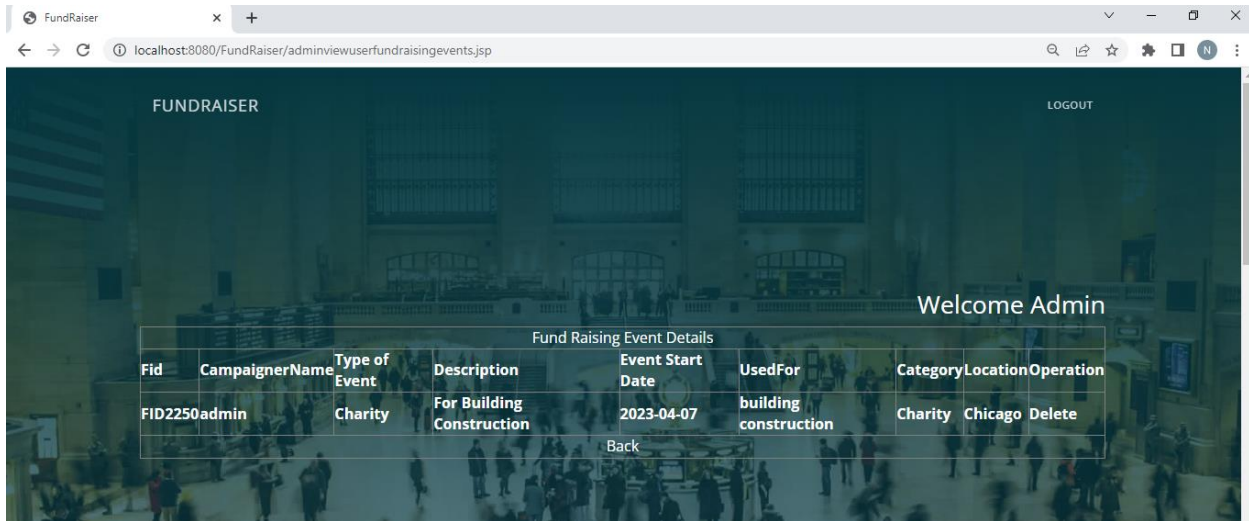


Figure 7: Admin view Fund raising events.

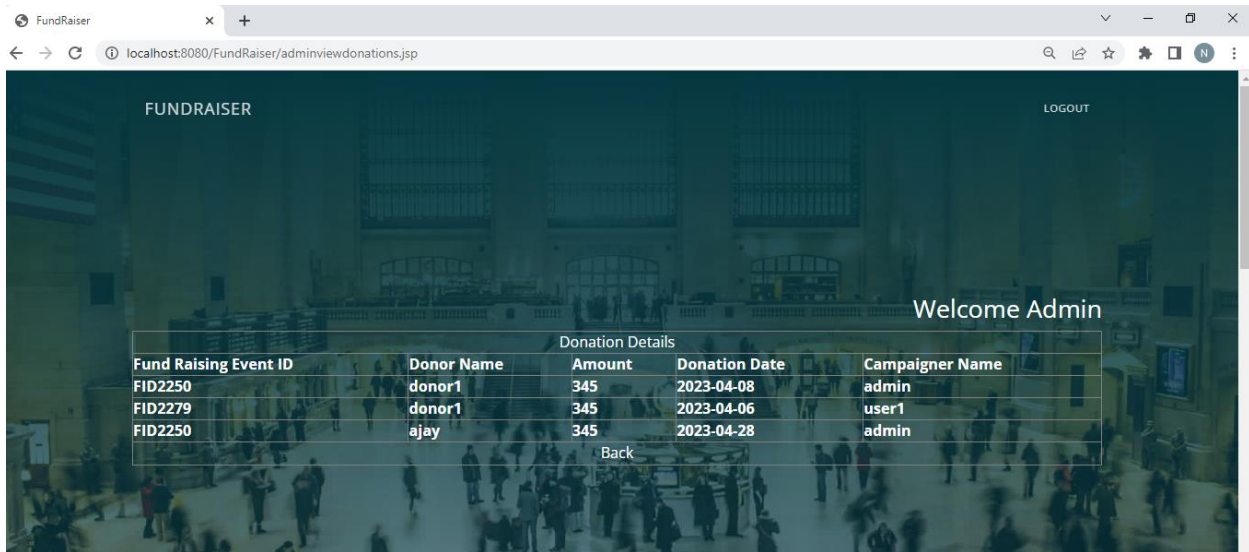


Figure 8: Admin view donations.

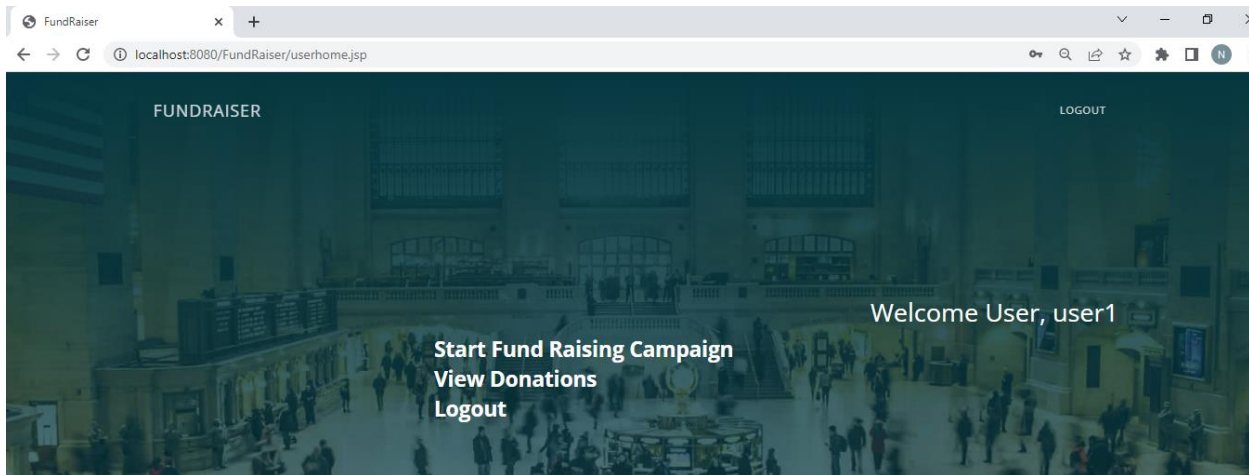


Figure 9: User Home page

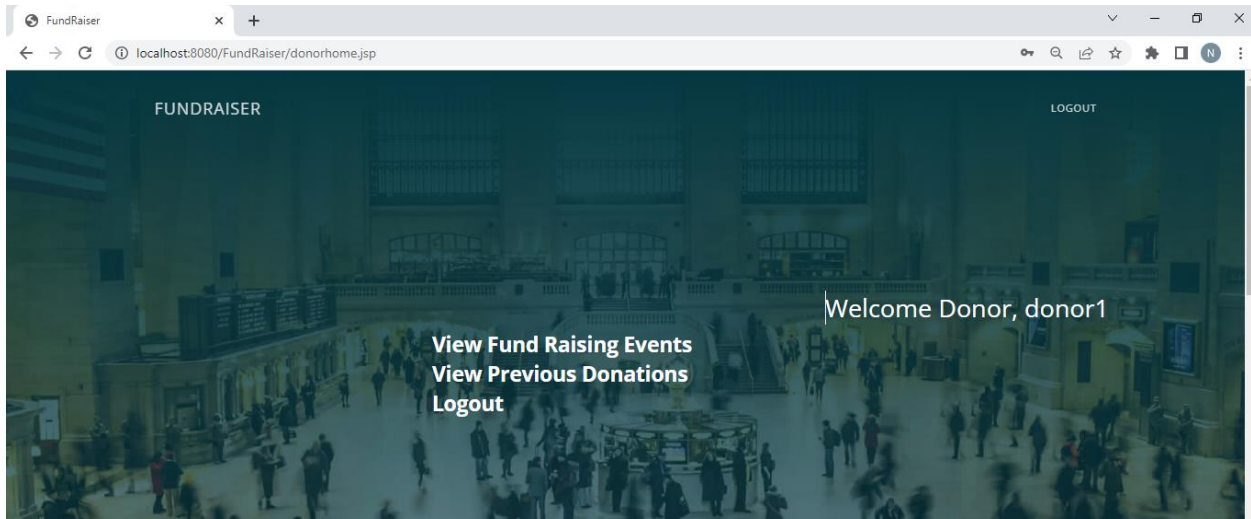


Figure 10: Donor Home page

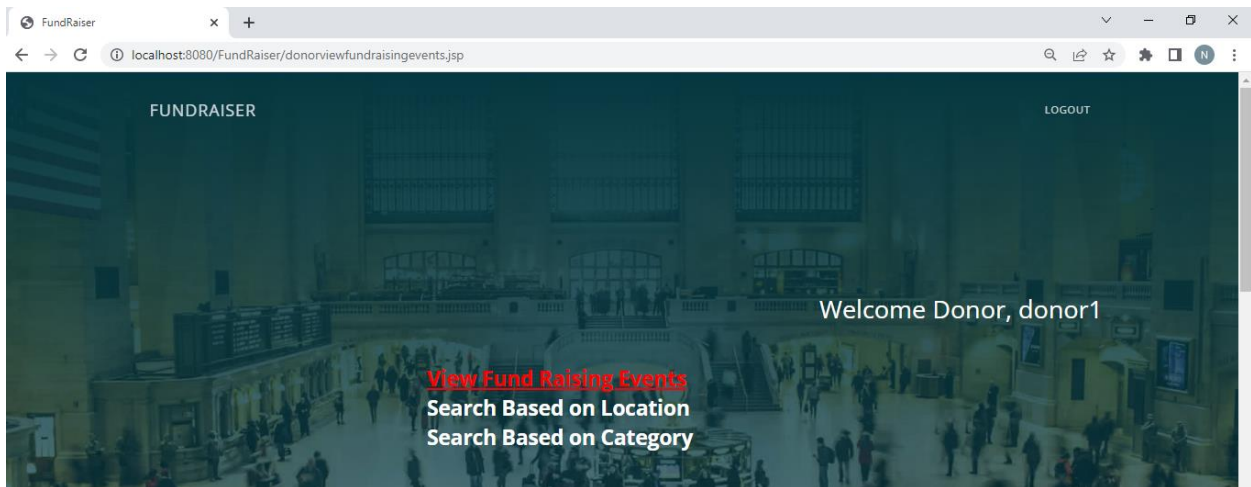


Figure 11: Donor search for Fundraising events.

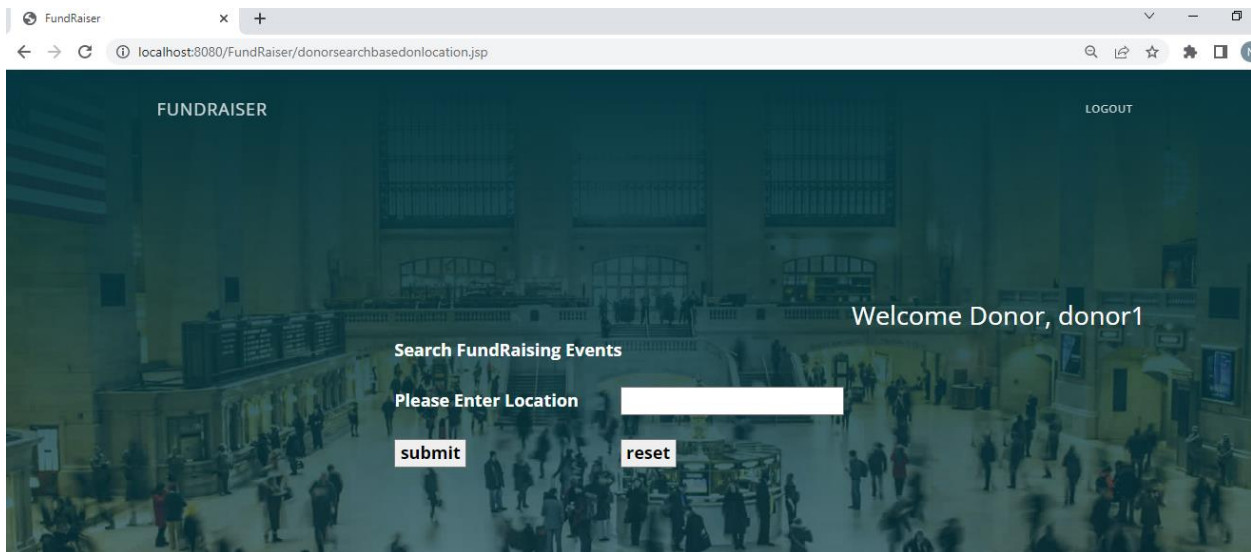


Figure 12: Donor search based on location.

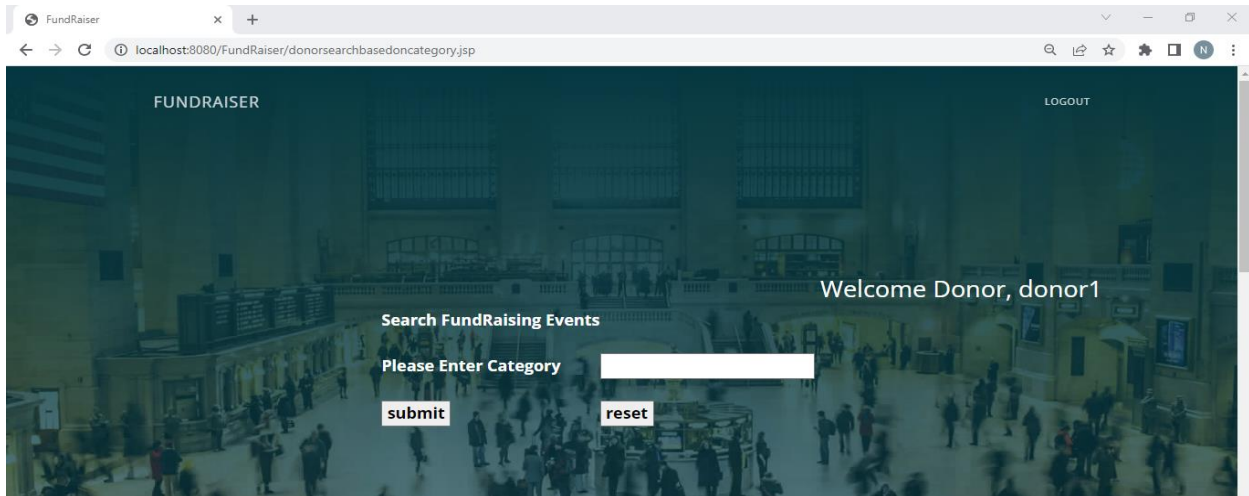


Figure 13: Donor search based on category.

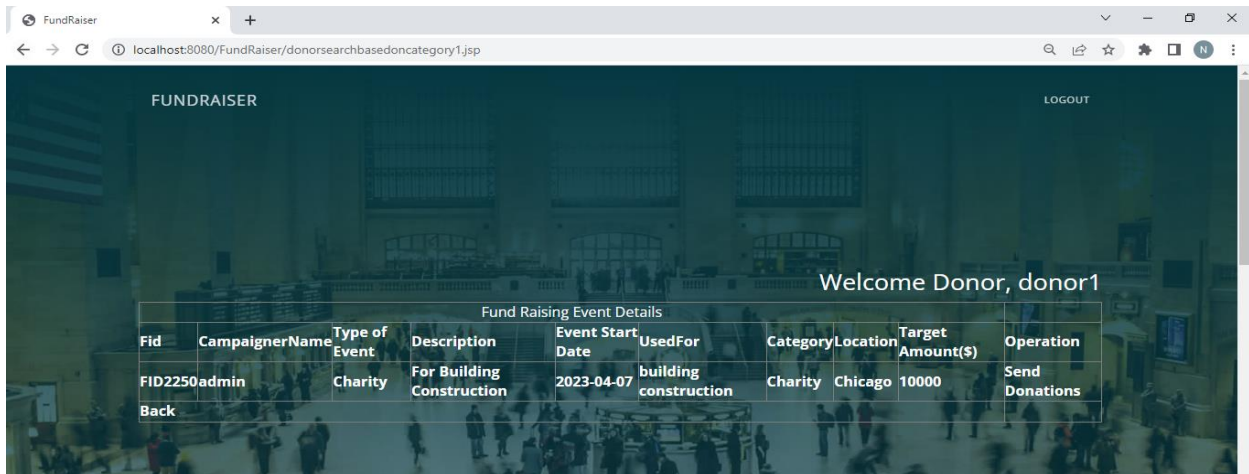


Figure 14: Donor view fundraising details.

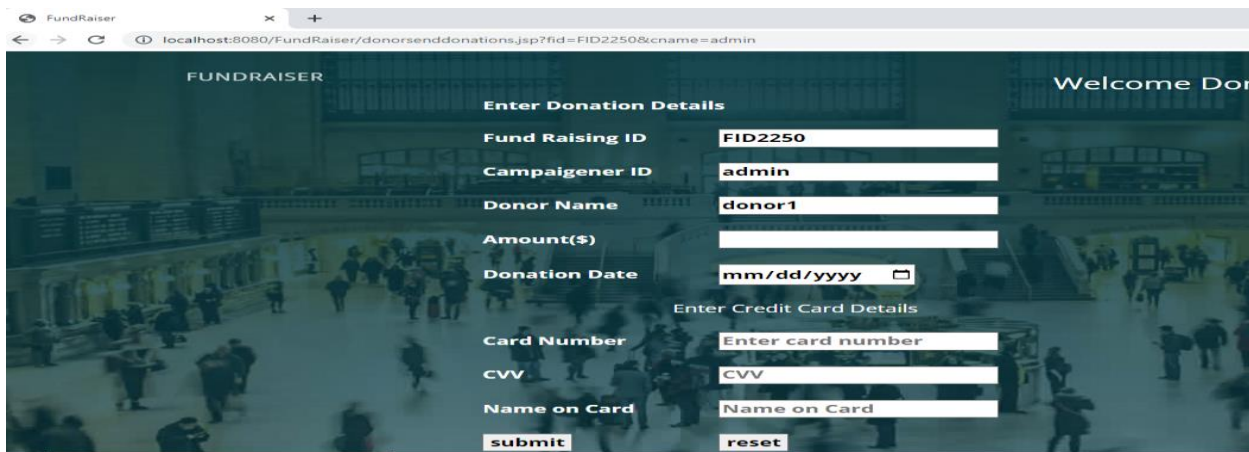


Figure 15: Donor sends donations.

6 Design Units Impacts

Donors can sign up, and users can search for events.

- Users can add fundraising events from any location.
- Users can post multiple events.
- Donors can send multiple donations.
- User campaigns are visible to donors only after the admin accepts them.
- Only registered donors are allowed to view events and send donations.

6.1 Functional Area A/Design Unit A

6.1.1 Functional Overview

In the software development life cycle, requirement specification is a primary criterion. To create this web application, we used a variety of hardware and software. Following are the technologies we used in this application.

Software Requirements

- Web Presentation : HTML, CSS, and Bootstrap
- Client-side Scripting : JavaScript
- Programming Language : Java
- Web based Technologies : Servlets, JSP
- Database Connectivity : JDBC
- Backend Database : MySQL
- Operating System : Windows
- Web Server : Tomcat
- IDE : Eclipse IDE

Deployment Requirements

- **Server:** Apache Tomcat
- **Database:** MySQL Database

6.1.2 Impacts

There is a significant impact in providing the required details to the donor and making their search successful, but in terms of functional consequences, there are no broad area effects on the application; later, the database must be increased to take care of the vast users, so that they can create multiple fund-raising events and manage the connection issues with the MySQL database.

7 *Open Issues*

In this modern era, donors can access this application from anywhere in the world. So, they might face some risks. However, the main risk is to find a genuine fundraiser. Hence, proper risk management must be implemented to reduce risk impact.

8 *Acknowledgements*

We want to express our sincere gratitude to my department for giving me the excellent opportunity to work as a team and on the selected platforms for the completion of the project. Furthermore, we are grateful to our professor, Dr. Yunchuan Liu, for helping us choose and carry out the project. We completed our project successfully under our professor's guidance, and valuable weekly feedback was given. This project plays a vital role in our academic career.

9 *References*

- [1]. Margaret Rouse, "Three-Tier Architecture," <https://www.techopedia.com/definition/24649/three-tier-architecture>.
- [2]. Risk Analysis & Risk Management in Software Engineering, <https://www.guru99.com/risk-analysis-project-management.html>
- [3]. Ahmad Shadeed, "The Process of Implementing A UI Design from Scratch"
- [4]. Mara Calvello, "Waterfall Methodology: How to Use It for Your Next Big Project," <https://www.g2.com/articles/waterfall-methodology>
- [5]. A.R. Andreasen and P. Kotler, "Strategic Marketing for Nonprofit Organizations" in Upper Saddle River, NJ, Prentice Hall, 2003.
- [6]. M. Hager, P. Rooney, and T. Pollak, "How Fund-raising Is Carried Out in US Nonprofit Organizations," *International Journal of Nonprofit and Voluntary Sector Marketing*, vol. 7, no. 4, pp. 311-324, 2002.