

San Jose State University SJSU ScholarWorks

Master's Projects

Master's Theses and Graduate Research

Spring 2011

Video Chat Application for Facebook

Shirali Chokshi
San Jose State University

Follow this and additional works at: https://scholarworks.sjsu.edu/etd_projects

Part of the [Other Computer Sciences Commons](#), and the [Software Engineering Commons](#)

Recommended Citation

Chokshi, Shirali, "Video Chat Application for Facebook" (2011). *Master's Projects*. 186.
DOI: <https://doi.org/10.31979/etd.cr2h-svuk>
https://scholarworks.sjsu.edu/etd_projects/186

This Master's Project is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Projects by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

Video Chat Application for Facebook

A Writing Project

Presented to

The Faculty of the Department of Computer Science
San José State University

In Partial Fulfillment of the

of the Requirements for the Degree
Master of Science

By

Shirali Chokshi

May 2011

© 2011

Shirali Chokshi

ALL RIGHTS RESERVED

SAN JOSÉ STATE UNIVERSITY

The Undersigned Project Committee Approves the Project Titled

Video Chat Application for facebook

by

Shirali Chokshi

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

_____	_____	_____
Dr. Soon Tee Teoh	Department of Computer Science	Date

_____	_____	_____
Dr. Robert Chun	Department of Computer Science	Date

_____	_____	_____
Mr. Chirag Patel	Quality Assurance Software Engineer	Date

ABSTRACT

Video Chat Application for facebook

By

Shirali Chokshi

This project is mainly written for the facebook users. In today's world, there are many social networking sites available. Among those social networking web sites, facebook is widely used web site. Like all other social networking web sites, Facebook also provides many features to attract more and more users. But it lacks in providing the most important feature of social networking, i.e. video chat. I explore the different options and requirements needed to build the video chat application. I have also described the integration of the application with the facebook.

Table of Contents

1. INTRODUCTION	1
1.1 Background Information	1
1.1.1 Why Video Chat for Facebook?.....	1
2. PROJECT WORKFLOW	3
2.1 Login Flow	4
2.1.1 Challenges in Implementing Login Functionality	6
2.2 Video Chat.....	6
2.2.1 Challenges in Implementing Video Chat	7
3. STREAMING SERVER.....	8
3.1 Internal Architecture of Red5	8
3.1.1 Web Container - Jetty	8
3.1.2 Web Container - Tomcat.....	9
3.1.3 Jetty or Tomcat?	10
3.1.2 Apache Mina	12
3.1.3 Spring Framework.....	14
3.2 Why Red5?	15
4. FACEBOOK INTERACTION.....	17
4.1 Creating Application in Facebook	17
4.2 Facebook Registration (Authorization and Authentication)	22
4.2.1 Authentication	22
4.2.1.1 Server side flow.....	23
4.2.1.2 Client side flow.....	26
4.2.2 Authorization	26

5. User Experience	28
5.1 User Feedback.....	28
5.2 Visualization of Application.....	32
6. Conclusion	38
References:	39
Appendix A: Additional images of Facebook interaction.....	41

List of Figures

[FIGURE 1: VIDEO CHAT APPLICATION DIAGRAM]	2
[FIGURE 2: SIMPLE PROJECT FLOW]	4
[FIGURE 3: LOGIN FLOW OF APPLICATION]	5
[FIGURE 4: APACHE TOMCAT ARCHITECTURE]	10
[FIGURE 5: WHERE MINA COMES INTO PICTURE]	12
[FIGURE 6: INTERNAL WORKFLOW OF APACHE MINA]	13
[FIGURE 7: SPRING IOC CONTAINER]	15
[FIGURE 8: SPRING FRAMEWORK]	15
[FIGURE 9: HOME PAGE OF FACEBOOK ACCOUNT]	17
[FIGURE 10: DEVELOPER'S PAGE IN FACEBOOK]	17
[FIGURE 11 : SECURITY CHECK TO CREATE NEW APPLICATION IN FACEBOOK]	18
[FIGURE 12: BASIC INFORMATION OF THE APPLICATION]	19
[FIGURE 13: FACEBOOK INTEGRATION INFORMATION PAGE OF AN APPLICATION]	20
[FIGURE 14: PAGE AFTER APPLICATION IS CREATED]	21
[FIGURE 15: APPLICATION HOME PAGE]	22
[FIGURE 16: FACEBOOK LOGIN PAGE]	24
[FIGURE 17: REQUEST FOR PERMISSION FOR BASIC INFORMATION PAGE]	24
[FIGURE 18: REQUEST FOR PERMISSION MORE THAN BASIC INFORMATION PAGE]	25
[FIGURE 19: PREVIOUS HOME PAGE]	28
[FIGURE 20 : CURRENT HOME PAGE]	29
[FIGURE 21: THE SAME SIZE OF WINDOW]	30
[FIGURE 22: CURRENT SIZE OF WINDOWS]	31
[FIGURE 23 : WELCOME PAGE]	32
[FIGURE 24 : FACEBOOK LOGIN PAGE]	33
[FIGURE 25 : USER PERMISSION]	34
[FIGURE 26 : VIDEO CHAT WINDOW]	35
[FIGURE 27 : PERMISSION TO START CAMERA]	36
[FIGURE 28 : CAPTURING VIDEO]	37

List of Figures in Appendix

[FIGURE_APX 1: CREATE NEW APPLICATION PAGE IN FACEBOOK]	41
[FIGURE_APX 2 : CANVAS PAGE AND CANVAS URL OF AN APPLICATION]	42

List of Tables

[TABLE 1: TECHNOLOGY DESCRIPTION].....	3
[TABLE 2: TECHNOLOGIES USED BY STREAMING SERVER]	8
[TABLE 3 : COMPARISON BETWEEN JETTY AND TOMCAT]	12
[TABLE 4 : COMPARISON BETWEEN RED5 AND FLASH MEDIA SERVER]	16

1. INTRODUCTION

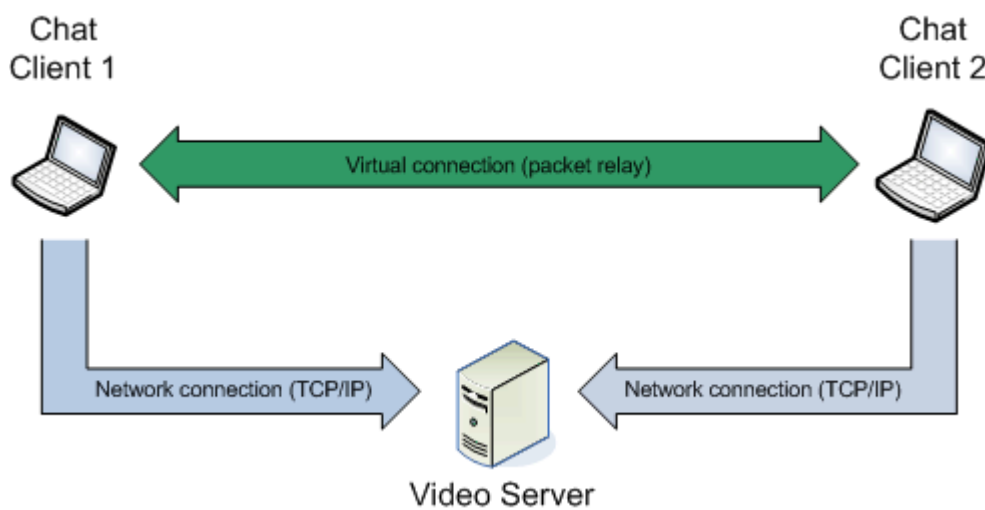
1.1 Background Information

Today we are so dependent on internet that we cannot imagine our life without internet. Now a day people want everything to be done on internet. They want everything live. People want to do their business online, they want to be in touch with everyone and also want to talk with them live, just like they are sitting in front of each other. Online video chat is one of the smartest ways to be in touch with all your friends and relatives. The good thing about this is that it is not just about entertainment or online get-togethers with friends/relatives. This form of chatting has also become one of the easiest and safest ways of doing business meetings and handling a wide array of issues in general, with the help of the Internet. The most common manner in which we use webcam video chatting is mainly recreational and entertaining, such as online get-togethers with friends in distant places. It has always been fun hanging around, chatting with friends online. Video chatting adds a little more flavor, as you are able to see whom you are talking to, making it twice the fun. This is great, particularly when you want to get in touch with someone close to you who lives abroad or at least a significant distance away, saving a lot of money and trouble. Free video chat offers a great alternative as you are able to stay logged on for as long as you like and there is no extra cost involved. The only investment you will have to do concerns directly the acquisition of a webcam, which is not so expensive any longer. Another immense benefit of webcam video chat is the application in various business fields, as many entrepreneurs nowadays prefer video conferences to settle in order affairs, instead of long and expensive journeys. Video chat offers a great opportunity for people to socialize and develop relationships on internet.

1.1.1 Why Video Chat for Facebook?

Now a day people are very addicted to social networking website; for example, Facebook. Today almost everyone has an account on Facebook. Facebook provides a great way to be in touch with people. It provides a great way to be socialized. Facebook has many features to give people cent percent entertainment. People also do their business on such websites. It offers different games and applications using which people get full recreation. It also offers chatting functionality. But Facebook is lacking in providing today's most exciting and smartest

way of chatting, i.e. VIDEO CHAT. If Facebook has Video chat functionality then it will become very helpful to people. Using this, people will get everything at the same place. People don't need to go to any other place to do video chat with their friends, relatives, and beloved ones. They will find everything at the same place. Video chat has actually become one of the most popular ways for people to get in touch and, as we have seen, its applications and benefits are countless as it cuts costs, saves money and energy and also provides users with a handy, quick and easy method of communication. Figure 1 shows how the basic video chat application works.



[Figure 1: Video chat application diagram]

2. PROJECT WORKFLOW

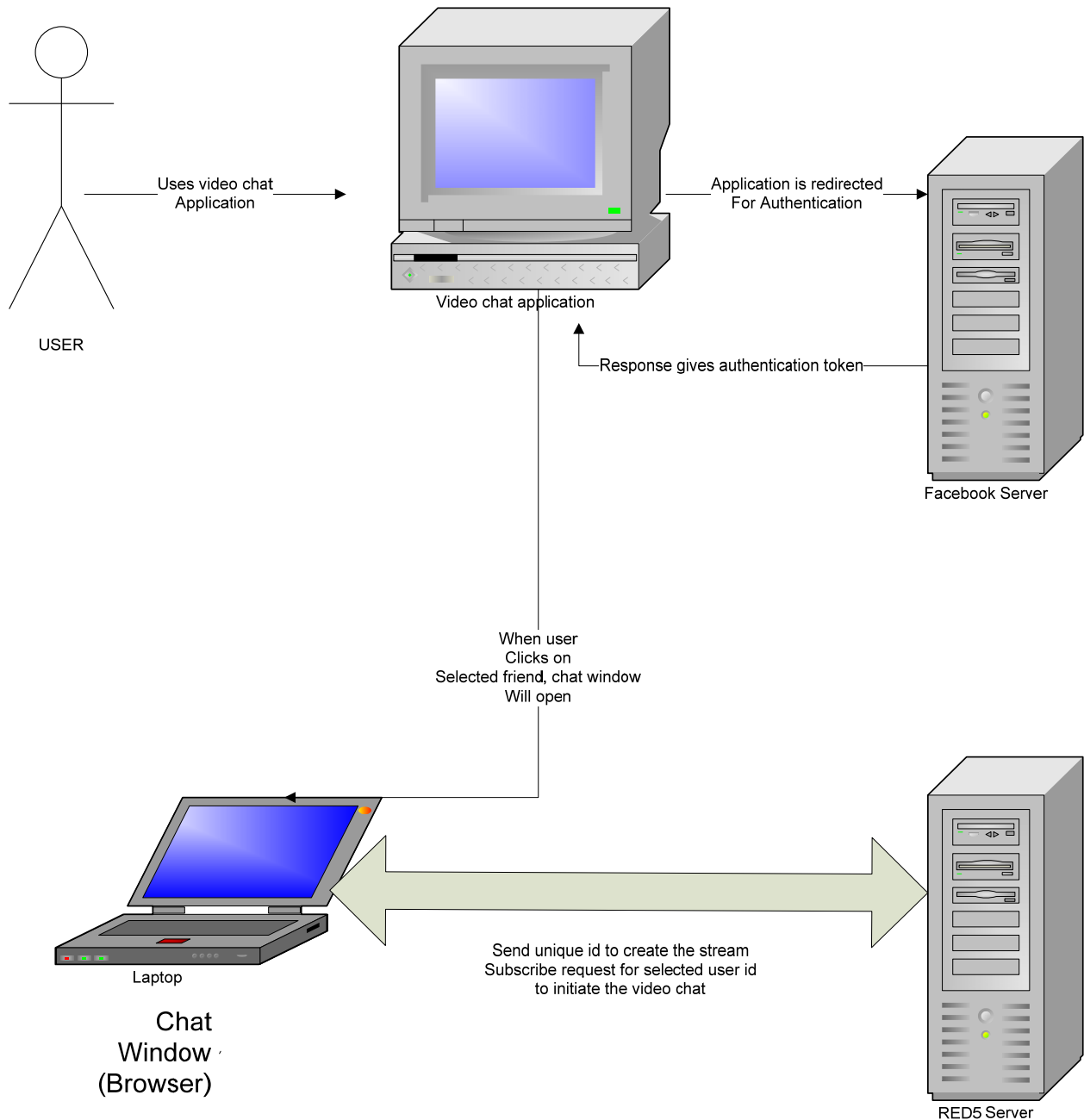
I have used the following technologies in my Video chat application.

Sr. No	Component	Technology
1.	Client Side	Adobe Flex 4.0
2.	Server Side	Servlet – JSP Technology (J2EE)
		Restful webservice using JSON format.
3.	Application Server	Red5

[Table 1: Technology Description]

As shown in figure 2, it is the basic flow of my application. To explain the project flow in easy manner, I have divided the whole project in two modules.

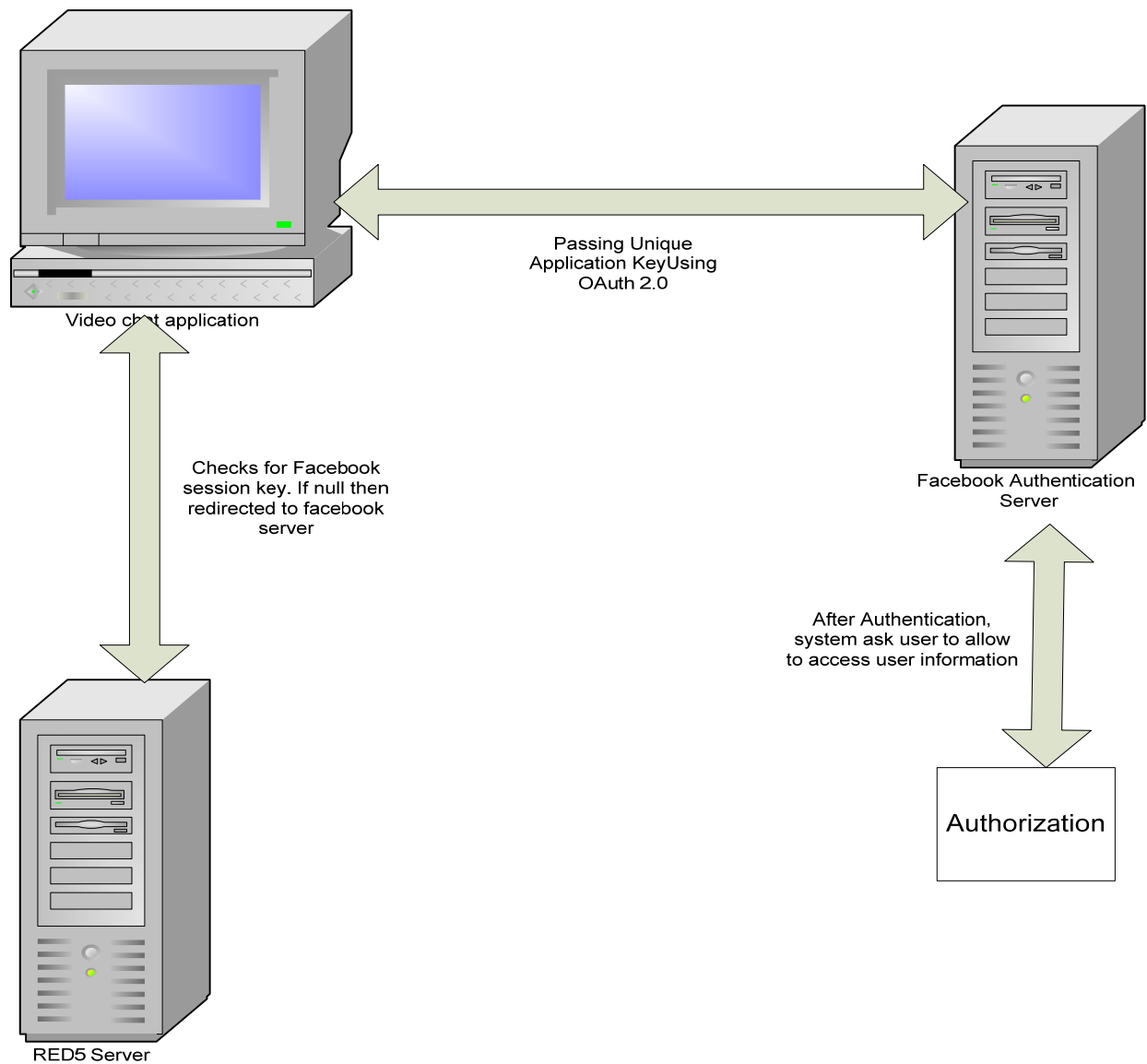
- The Login Flow
- The video chat flow



[Figure 2: Simple Project Flow]

2.1 Login Flow

I had decided to make video chat application for facebook. So, I had chosen facebook API to create the user authentication. In facebook, the user authentication can be created by two ways.



[Figure 3: Login flow of Application]

- By using restful web service.
- By using Spring security API.

Here, I had chosen to go with the restful web service. I had decided to use Facebook login to authenticate my users. I had used facebook OAuth 2.0 API. I had initiated the service using restful webservice with JSON format. In any application, authentication can be done by two ways.

- Authenticating users
- Authorizing application

Facebook provides API for authenticating users and authorizing application. In my application, it works as follow.

When users run the application, VideoChatServlet is called. In this servlet, I had used FacebookClient class. In this class, there are two classes. One is FacebookCallbackServlet and another is FacebookLoginServlet. This class checks whether facebook session key is null or not. If it is null that means user is not logged into his/her account and FacebookLoginServlet will navigate user to the facebook login page. If session key is not null then this means that user is logged into his/her facebook account and system process further request. Now, if user is already logged into his/her facebook account then he/she is navigated to the page where users can find other friends who are online in this application. For the navigation, my application uses the canvas URL.

If user is not logged into his/her facebook account then FacebookLoginServlet is called. Here user is navigated to the facebook login page. After user is authenticated successfully, facebook application will pass authentication token. Based on the authentication token provided by facebook, I have called `createSessionKey` function of facebook application which will give me unique session key.

2.1.1 Challenges in Implementing Login Functionality

Initially I had decided to use Spring security for authentication with facebook. I had successfully completed the configuration for that. Now at the client side for authentication it uses facebook floatbox control. This creates problems in browsers. Some Javascripts are not initialized properly. So later on I had decided to use the restful webservice authentication using JSON format.

2.2 Video Chat

For the video chat, I had decided to use the streaming media server. There are many servers but most of them are paid servers. So, I had chosen to go with the RED5 server as this is an open source server and it fulfills all my requirements.

To initiate the video chat, I need to connect the client to the server. For the media and flex connection, I had used NetConnection and NetStream classes. The use of these classes is as follow:

NetConnection:

The NetConnection class creates a bidirectional connection between a Flash Player or AIR an AIR application and a Flash Media Server application or between a Flash Player or AIR an AIR application and an application server running Flash Remoting. A NetConnection object is like a pipe between the client and the server.

NetStream:

It is used to send the object streams through the pipe.

Baesd on the NetConnection class I had created the connection with red5 using RTMP (Real Time Messaging Protocol) protocol. After successful connection with red5, to initiate the chat with the friends, I had subscribed my friend's streaming request to red5 server using RTMP. Also I had published my own stream to red5 so that other users can also request red5 to see it. Next step was to create the unique stream for each user. I had used "user id" field as my primary choice. I had passed my facebook "user id" to red5 server and this way I had created the unique stream.

2.2.1 Challenges in Implementing Video Chat

In the tutorial of the red5, they had hard coded stream name in video display spark control. My challenge was to make the stream dynamic. For that I had removed stream name from video display spark control and passed it dynamically to video display control in action script. I had faced similar challenge with the host name. I did same exercise to solve the problem.

3. STREAMING SERVER

I had used Red5 as a streaming server for my Video chat application. Red5 is an open source server. It is a flash server written in JAVA. The Red5 media server uses following technologies.

Web container	Network Application Framework	Framework	Clustering Technology
Jetty	Mina	Spring	Terracotta
Tomcat			

[Table 2: Technologies used by streaming server]

Above all, for the clustering technology “Terracotta”, the research work is still going on.

3.1 Internal Architecture of Red5

Red5 uses both jetty and tomcat as web container.

3.1.1 Web Container - Jetty

It is an open source web container. It provides HTTP server, HTTP client and servlet container. Jetty can be embedded in devices, frameworks, and application servers. Jetty can also be used with the clustering technologies. It has following features:

- It is based on the standards.
- It is an open source.
- Jetty is very flexible and extensible.
- It can be embedded.
- It is asynchronous.
- It can be used with clustering technologies and cloud computing.

The Red5 uses following main classes of jetty server.

1. JettyApplicationContext:

This class implements IApplicationContext interface. It wraps the jetty webapp context.

2. JettyApplicationLoader:

This class implements IApplicationLoader interface. It loads new applications in jetty.

3. JettyLoader:

This class implements LoaderMBean and ApplicationContextAware interfaces. It loads the red5 applications in jetty. The ApplicationContextAware is a spring framework interface. This interface is implemented when an object needs access to a set of beans. It is also implemented when an object needs access to file resources.

4. Red5WebPropertiesConfiguration:

This class implements Serializable, EventListener, and Configuration interfaces. It is used to read the web properties configuration.

3.1.2 Web Container - Tomcat

Tomcat is also an open source web container. It is implementation of java servlet and java server pages (JSP). Figure 4 shows the architecture of Tomcat.

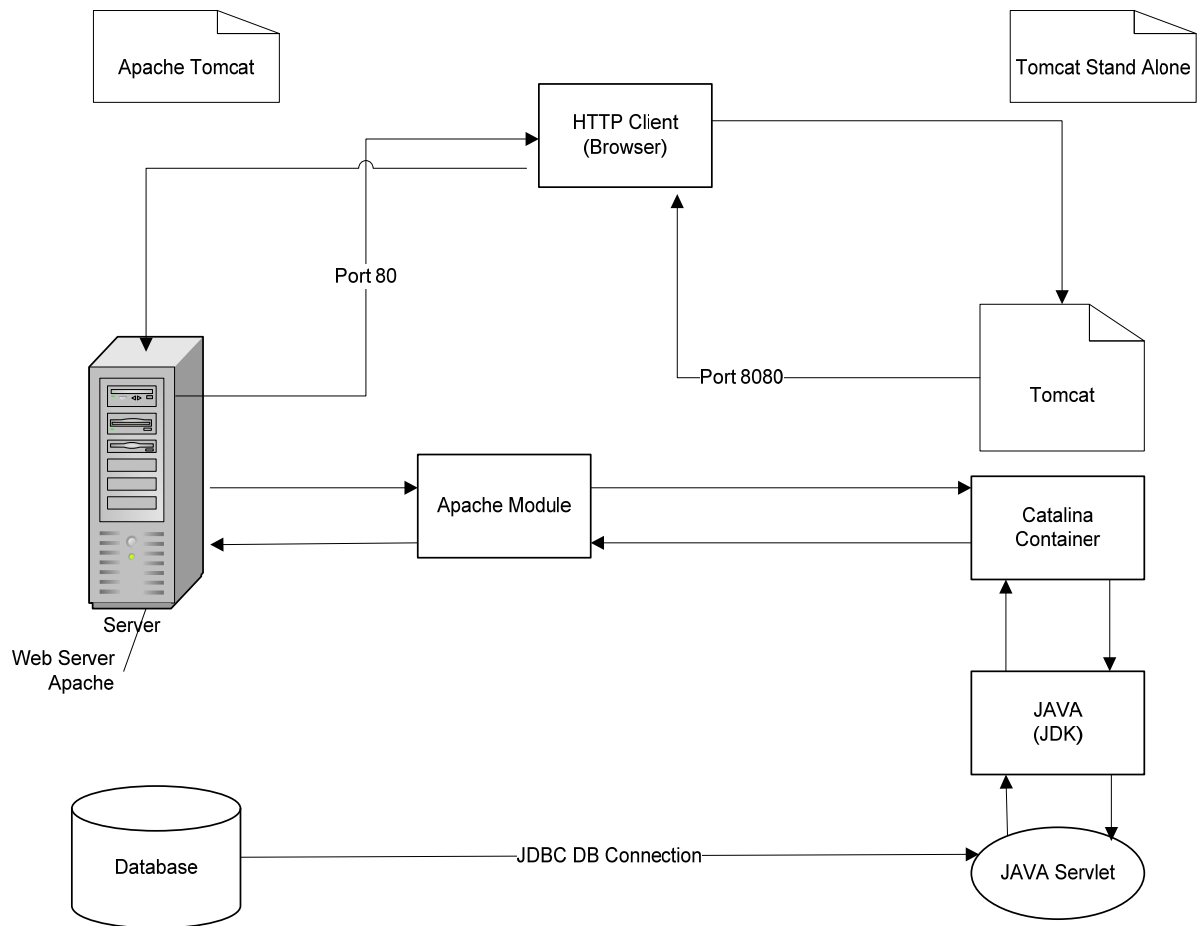
Using tomcat, it is easy to build the web applications. It provides robust capabilities and extensive customization features. Benefits of using tomcat as a web container are:

- It is flexible

It will allow you to run the apache on one server and JSP/servlet on another machine. This can also be used to provide the security.

- Tomcat server is very stable

If there is some problem that cause the tomcat server to fail completely then also it would not render your entire service unusable. Only the JSP page and servlet will be affected.



[Figure 4: Apache Tomcat Architecture]

3.1.3 Jetty or Tomcat?

For my project, I can use either jetty or tomcat embedded version of red5. I have compared both the web container and decided to go with Tomcat embedded red5 server.

Sr. No	Comparison Type	Jetty	Tomcat
1.	Architecture	<ul style="list-style-type: none"> • Jetty is a set of software components. It provides HTTP and servlet services. • It can be installed as an 	<ul style="list-style-type: none"> • Tomcat is an application server. It can be installed as software on system and from which one can install applications

		<p>application server. It can also be embedded in an application or framework</p> <ul style="list-style-type: none"> • Jetty's flexible architecture allows it to be deployed and integrated in a various range of instances. For example, from cell phones to large servers. • It allows the deployment of various services without any complexities and inefficiencies. 	<ul style="list-style-type: none"> • It can be embedded or built up as JEE server.
2.	Performance	<ul style="list-style-type: none"> • Jetty provides better scalability when there are many connections with idle time • In jetty, less memory and less CPU cache is consumed by servlet container. So, more cache is available to speed up the execution. 	<ul style="list-style-type: none"> • Tomcat gives little bit better performance when there are busy connections.
3.	Features	<ul style="list-style-type: none"> • Jetty provides an easy migration path for full EE towards Geronimo, Jboss, JOnAS, and 	<ul style="list-style-type: none"> • Tomcat provides an easy migration path for full EE towards Jboss and Geronimo.

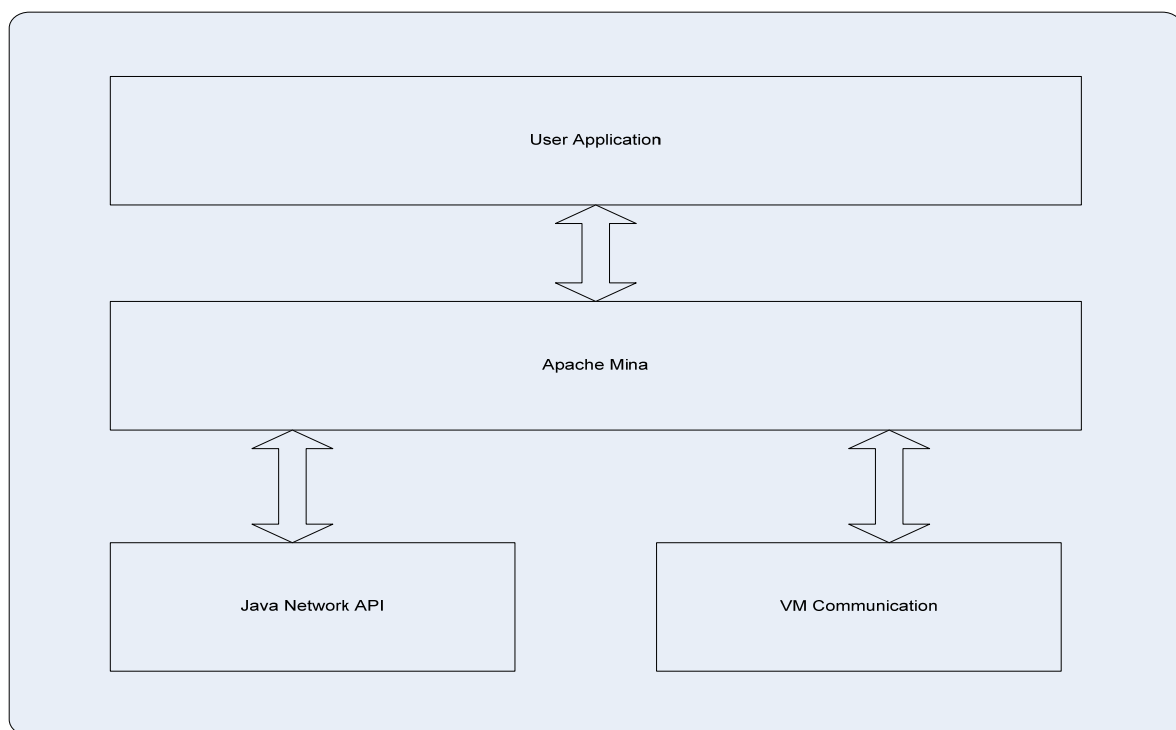
		Sybase EAServer.	
--	--	------------------	--

[Table 3 : Comparison between Jetty and Tomcat]

At the end, I can say that both Tomcat and Jetty serves their purpose but today people use Tomcat more than Jetty.

3.1.2 Apache Mina

Red5 uses Apache mina as a network application framework. This helps users to create high performance and high scalable network applications.

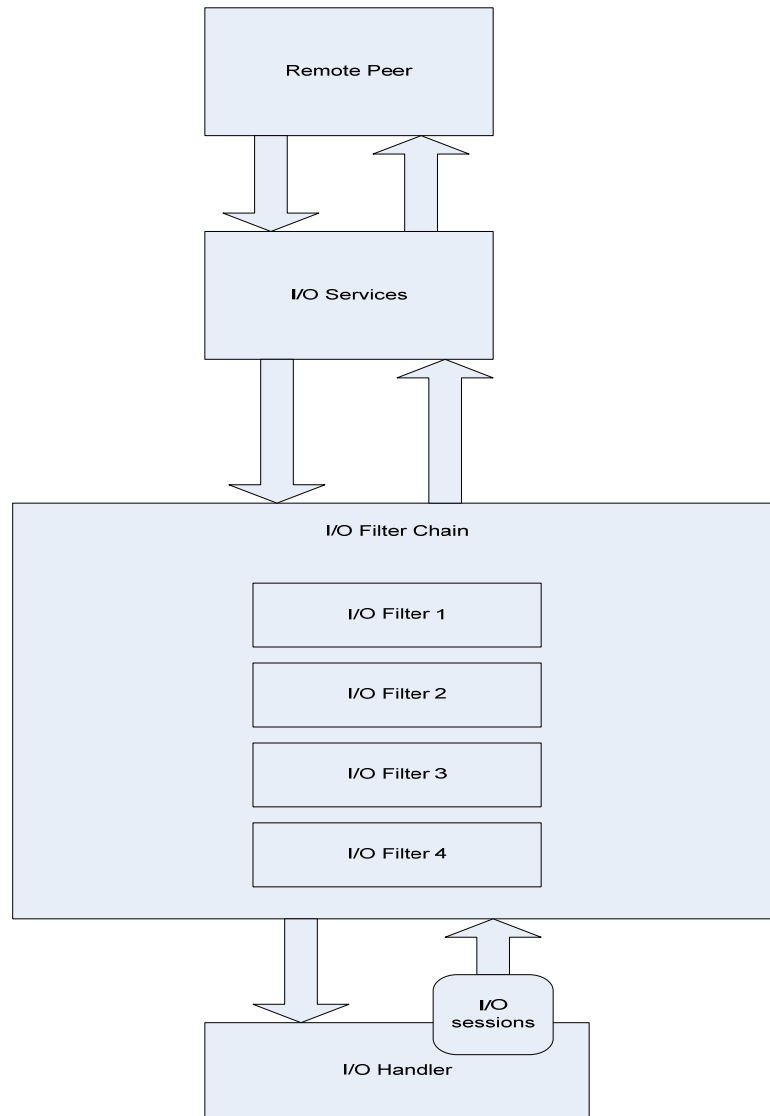


[Figure 5: Where Mina comes into picture]

Apache Mina provides an abstract, event driven and asynchronous API. It has many features which are as follow:

- It provides API for different transport types like,
 - TCP/IP and UDP/IP via JAVA NIO
 - Serial communication
 - Pipe communication
- It provides filter interface. For example, as an extension point

- It also provides filter interface same as servlet.



[Figure 6: Internal Workflow of Apache Mina]

- It can work with both low-level and high-level API
 - For low-level, it uses ByteBuffers
 - For high-level, it uses user defined objects.
- It can do overload shielding and traffic throttling.
- It can be managed with JMX.
- It supports stream based I/O support.

- It can be easily integrated with other containers like Spring.

Figure 6 shows Apache Mina's internal work flow diagram. In this diagram, there are mainly 3 components.

1. I/O Services
2. I/O Filters
3. I/O Handler

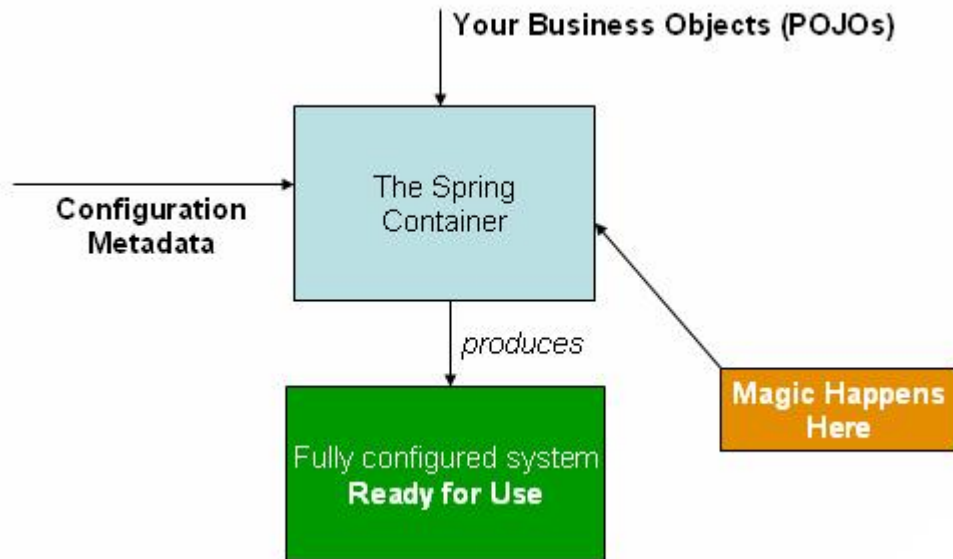
I/O service performs actual I/O operations. Apache mina can have more than one filter. The main function of filter is to filter the events and requests. I/O handler is responsible for logic. It contains the logic of your program.

3.1.3 Spring Framework

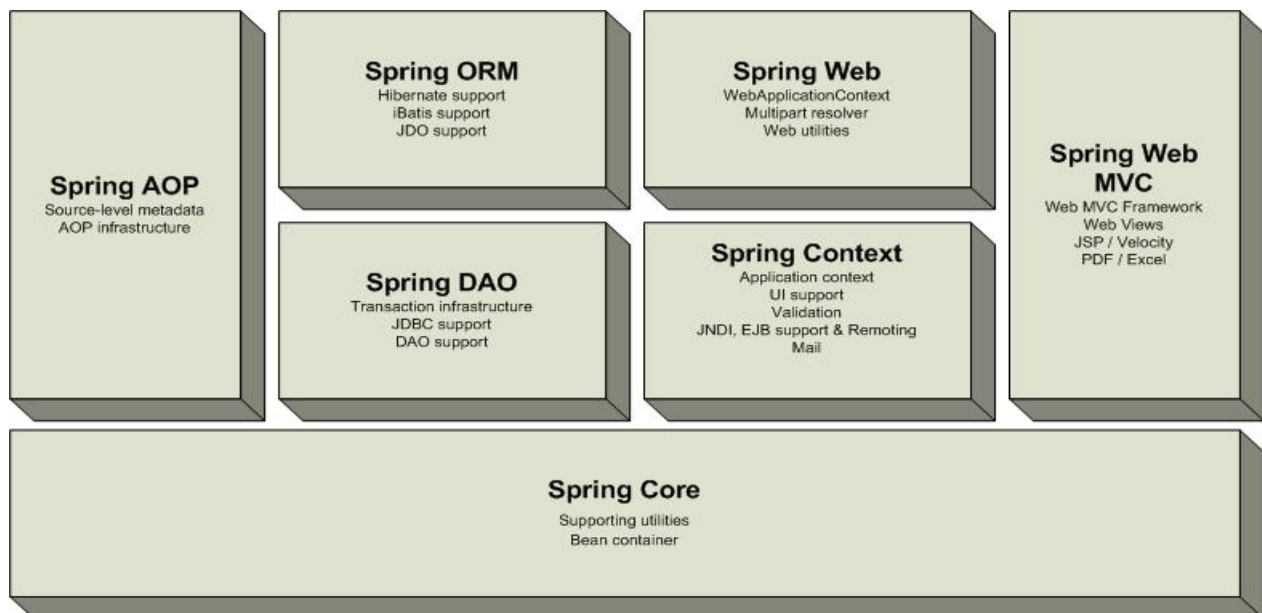
Spring framework is light weighted and comprehensive framework for building java EE applications. Red5 is written in Spring framework. Red5 uses Spring to perform dependency injection (DI) with the help of Spring's IoC (Inversion of Control) container. It reduces the dependencies of components and allows developers to configure Red5 without modifying the core Red5 API. Also, spring provides the functionality and features that anyone can use for his/her server side logic. One of the reasons red5 server is so flexible and extensible is – its “Spring framework”.

Spring framework provides many key features, which are as follow:

- It provides a core bean factory which can be used globally.
- It provides separate abstraction layer for database transactions.
- It has in built strategies for Java Transaction API (JTA). So that there will be no need of Java EE environment dependency.
- It can be easily integrated with many other frameworks like Hibernate, JDO, and iBATIS.
- It also supports MVC based web applications.
- Spring framework is Extensive aspect-oriented programming (AOP) framework. That means it is designed to improve the modularity of systems those are created using spring framework.



[Figure 7: Spring IoC Container]



[Figure 8: Spring Framework]

3.2 Why Red5?

As I mentioned before, red5 is a flash media server. Why I had chosen Red5 over the Adobe flash media server has many reasons. Few of the reasons are as follow:

Sr. No	Comparison Type	Red5 Media Server	Adobe Flash Media Server
1.	General	<ul style="list-style-type: none"> • It is an open source. • It is free server. 	<ul style="list-style-type: none"> • It is created by Macromedia/Adobe System. • It costs around \$4500.
2.	Protocol support	<ul style="list-style-type: none"> • RTMP • RTMPT • RTMPS • RTMPE • RTMPTE 	<ul style="list-style-type: none"> • RTMP • RTMPT • RTMPS • RTMPE • RTMPTE • RTMFP
3.	OS Support	<ul style="list-style-type: none"> • Windows • Mac OS • Linux • Solaris • Other Unix 	<ul style="list-style-type: none"> • Windows • Linux
4.	Container format support (Audio/video Streaming)	<ul style="list-style-type: none"> • MP4 • FLV • MP3 	<ul style="list-style-type: none"> • MP4 • FLV • MP3
5.	Server side	Java	AS2
6.	Client	Flash (RTMP)	Flash (RTMP)

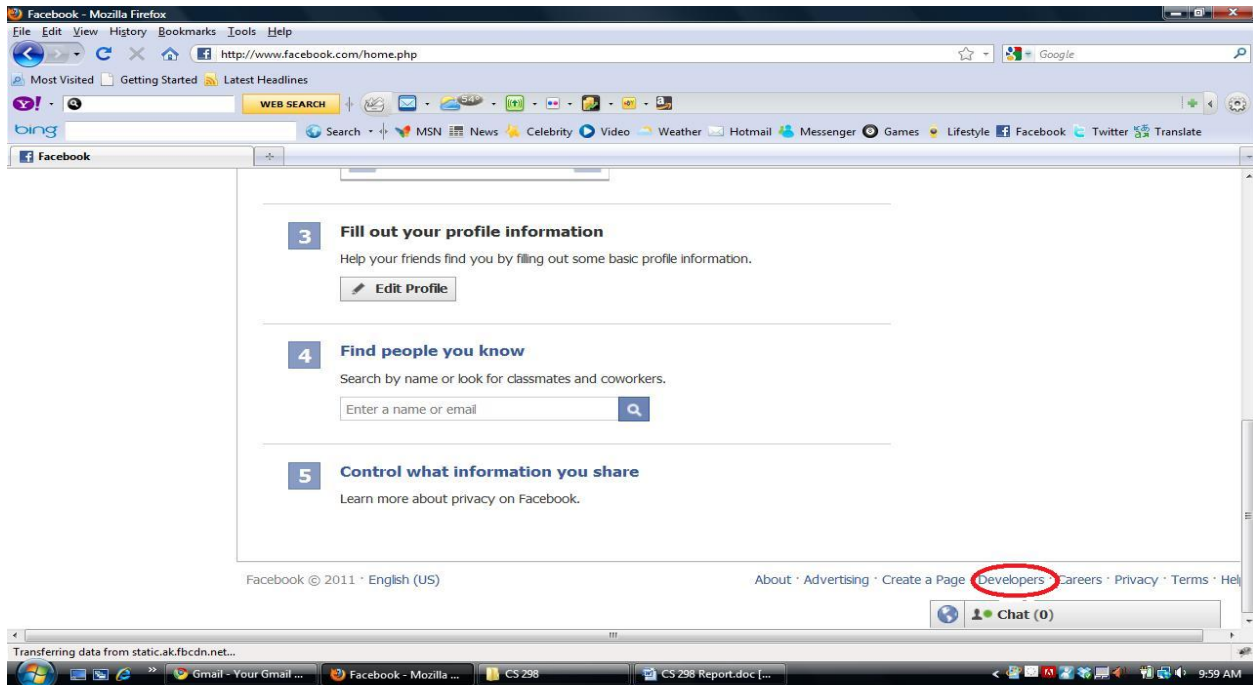
[Table 4 : Comparison between Red5 and Flash Media Server]

4. FACEBOOK INTERACTION

4.1 Creating Application in Facebook

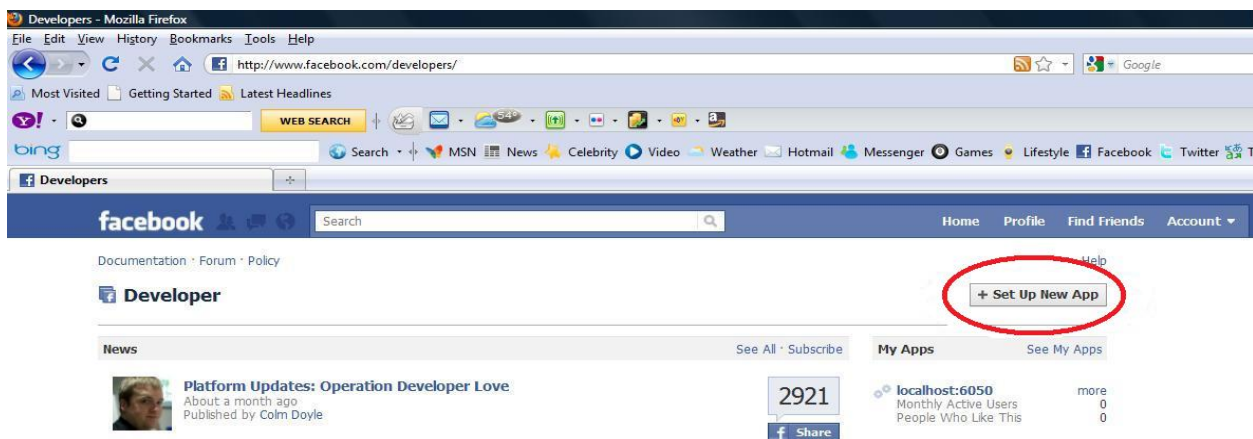
To create an application in facebook, I went through following steps.

First of all I had created an account with facebook. Once, I logged into my account, I found developers link at the bottom right corner of my facebook homepage. This is shown in figure 9.



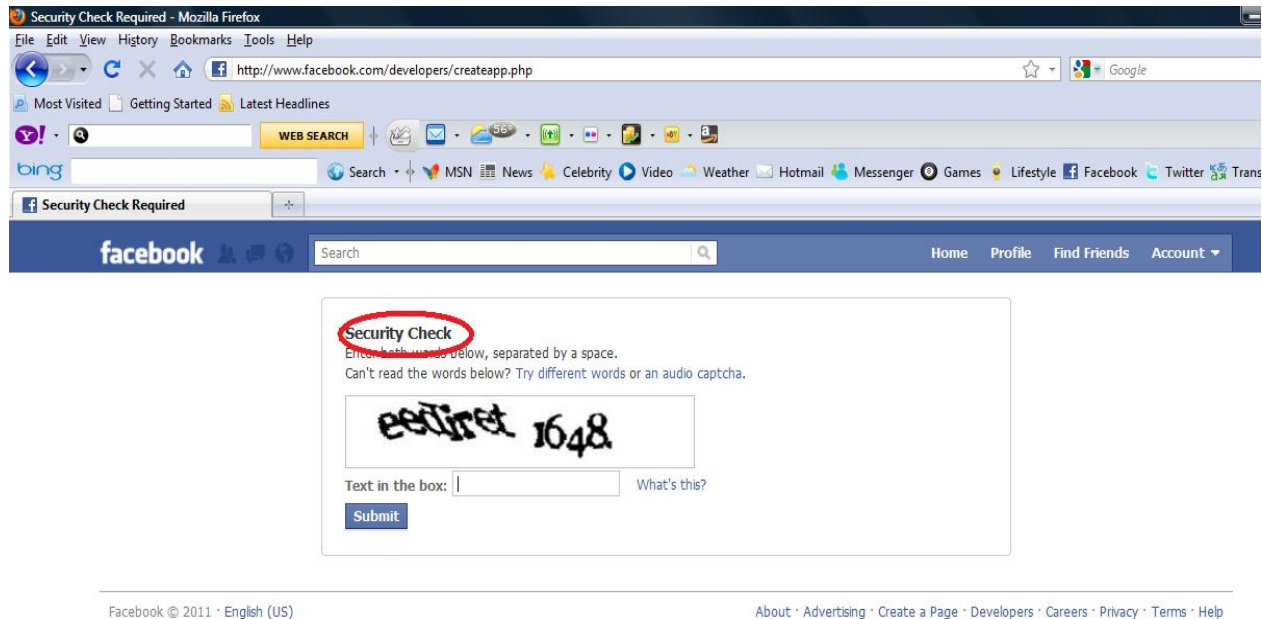
[Figure 9: Home page of facebook account]

Then I went to the developer's page. After that I went to the myApp page. In this page, on top right corner as shown in figure 10, I saw the button named "Set up new App".



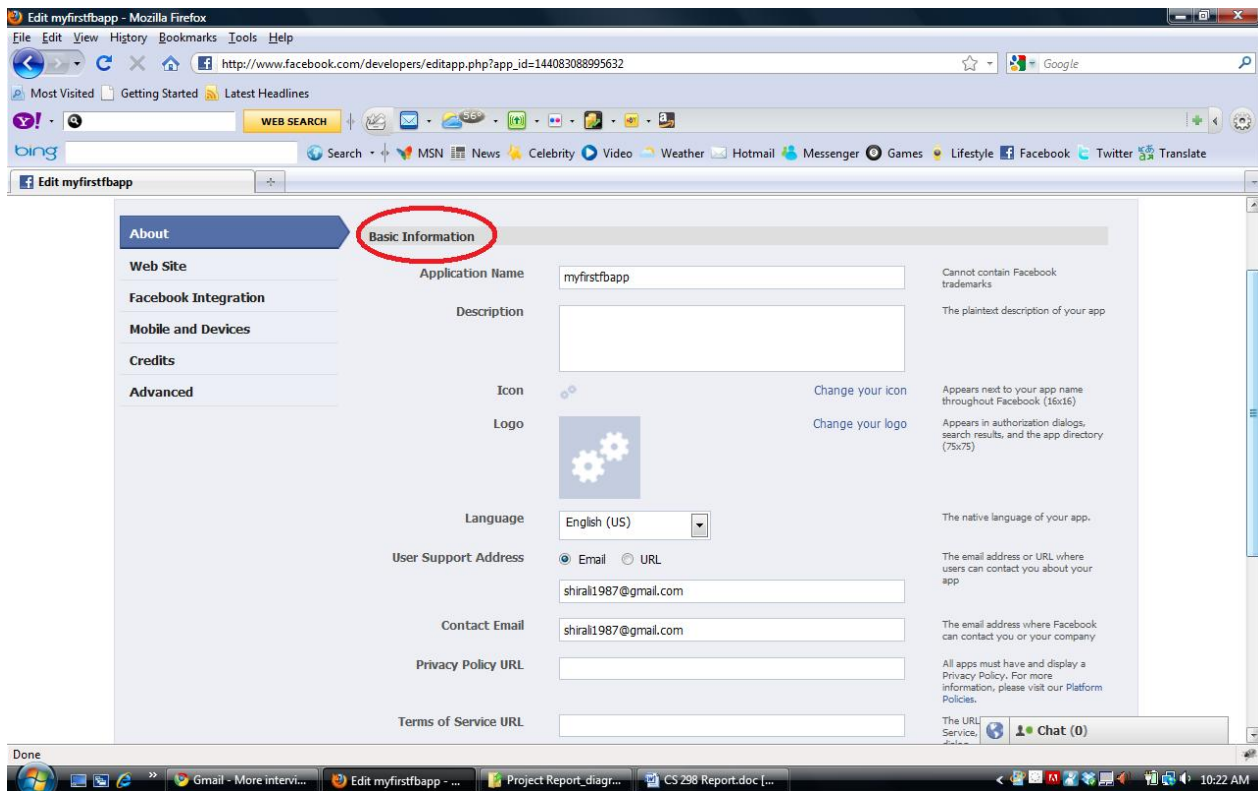
[Figure 10: Developer's Page in Facebook]

This button navigated me to the “create application” page. In this page, I entered the application name. Then by selecting agree, I pressed “Create App” button. By doing this, facebook navigated me to the security check page. In this page, I had to clear the captcha test as shown in figure 11.



[Figure 11 : Security check to create new application in facebook]

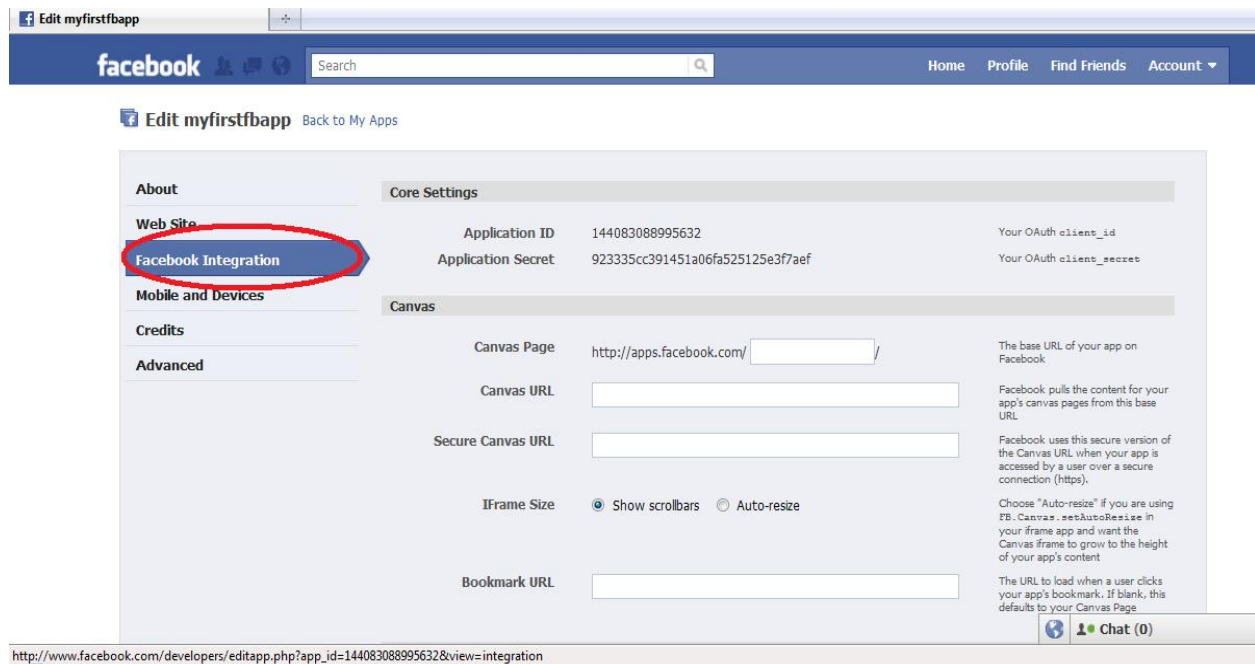
Once I had cleared the security test, I was directed to the application basic information page as depicted in figure 12. In this page, I was able to edit the application name, was able to add logo/icon for my application, was able to provide email address, privacy policy URL, Terms of services URL; I was also able to add more persons as administrator and also edit the current person.



[Figure 12: Basic Information of the application]

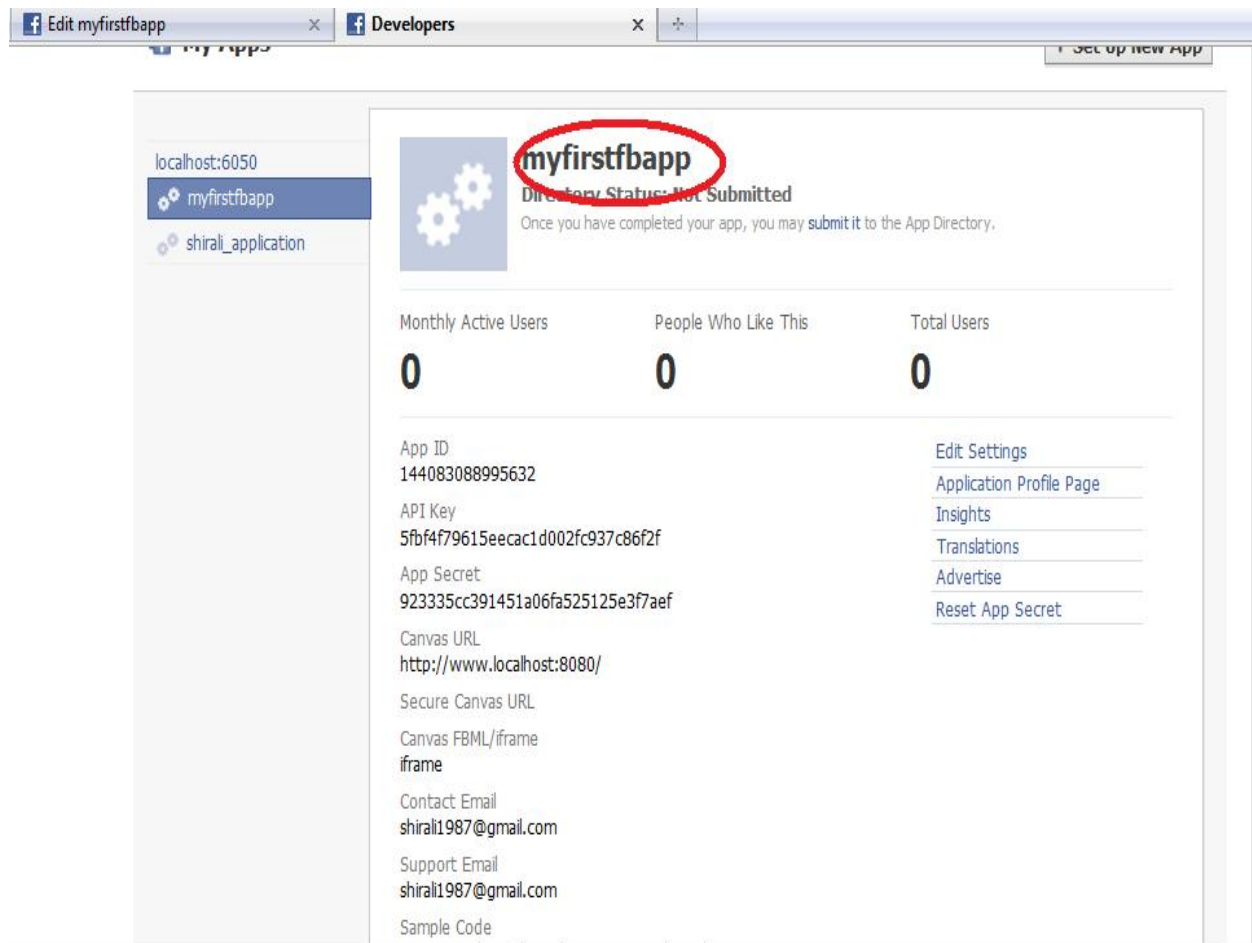
From the left side menu bar, I went to the facebook integration page where I found the most important information that were required in future to integrate my application with facebook. Here as shown in figure 13, I found following information:

- Application Id
- Application Secret
- Canvas page URL
- Frame Size



[Figure 13: Facebook Integration information page of an application]

After that I entered the application name at the canvas page text box. Then I made the changes as per my requirement and saved all the changes.

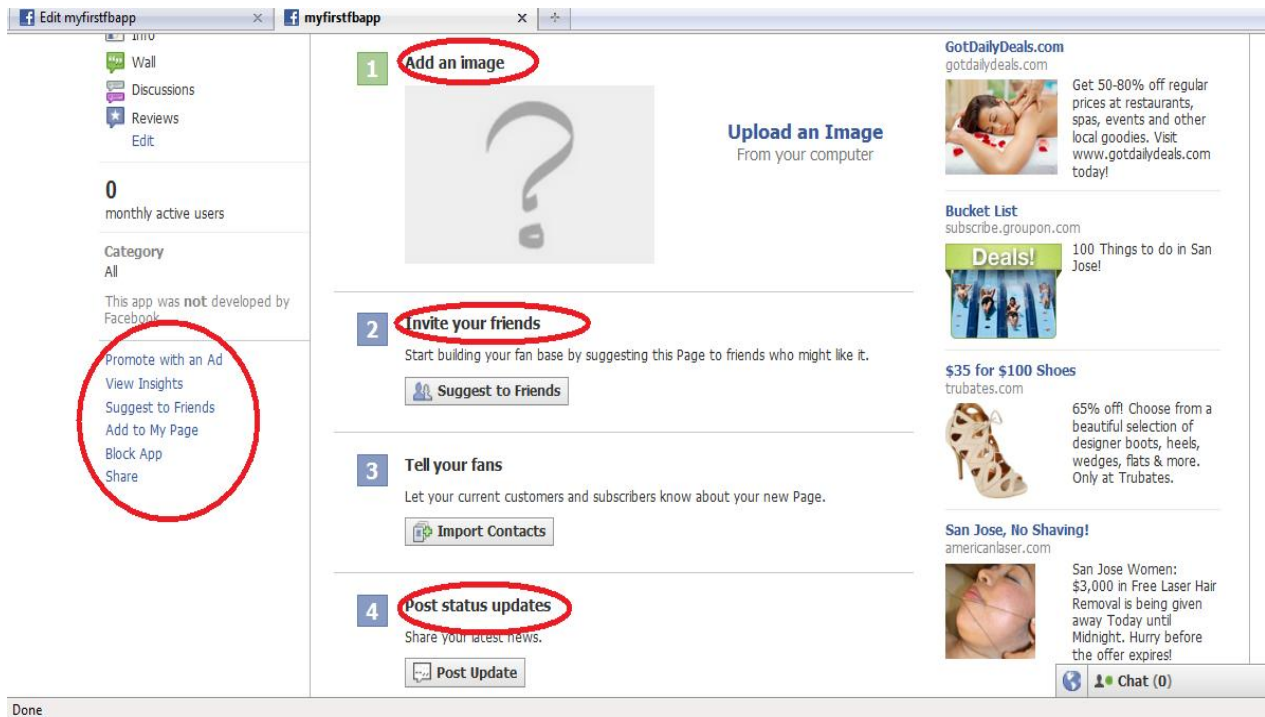


[Figure 14: Page after application is created]

After my application was created, I went to the application profile page. In this page, I found different options like,

- I can add image
- I can invite my friends
- I can post the status messages
- I can promote the page to my website
- I can also set up the mobile phone

This is shown in figure 15.



[Figure 15: Application home page]

4.2 Facebook Registration (Authorization and Authentication)

Facebook allows users to register or sign in functionality by providing its login system. I had also decided to use this feature of facebook. Using facebook's login system, I didn't need to create login functionality for my application. As long as user is signed into Facebook, he/she will automatically sing into my application. Also the basic information of the user, i.e. the information available publicly on facebook, will be available to the application. For users to log into system, facebook performs two main tasks.

- Authentication
- Authorization

4.2.1 Authentication

Authentication can be done by two ways:

- By using OAuth 2.0 protocol.
- By using open source JavaScript API.

In my project, I had used OAuth API. Facebook platform supports two different flows for user login.

- Server side flow:

Server side flow is also known as authentication code flow. Using this Graph API can be called from the web server.

- Client side flow:

Client side flow is also known as implicit flow. Using this, Graph API calls can be made from a client. This client can be some JavaScript running on a web browser or it can be any mobile or desktop application.

To implement the OAuth protocol, I had to fulfill following steps:

- User authentication

This ensures that users are who they say they are.

- Application authorization

This is used to make sure that users know what data and capabilities they are providing to the application.

- Application authentication

This is used to guarantee that users are giving their information to the application they are interacting with and not to some other application.

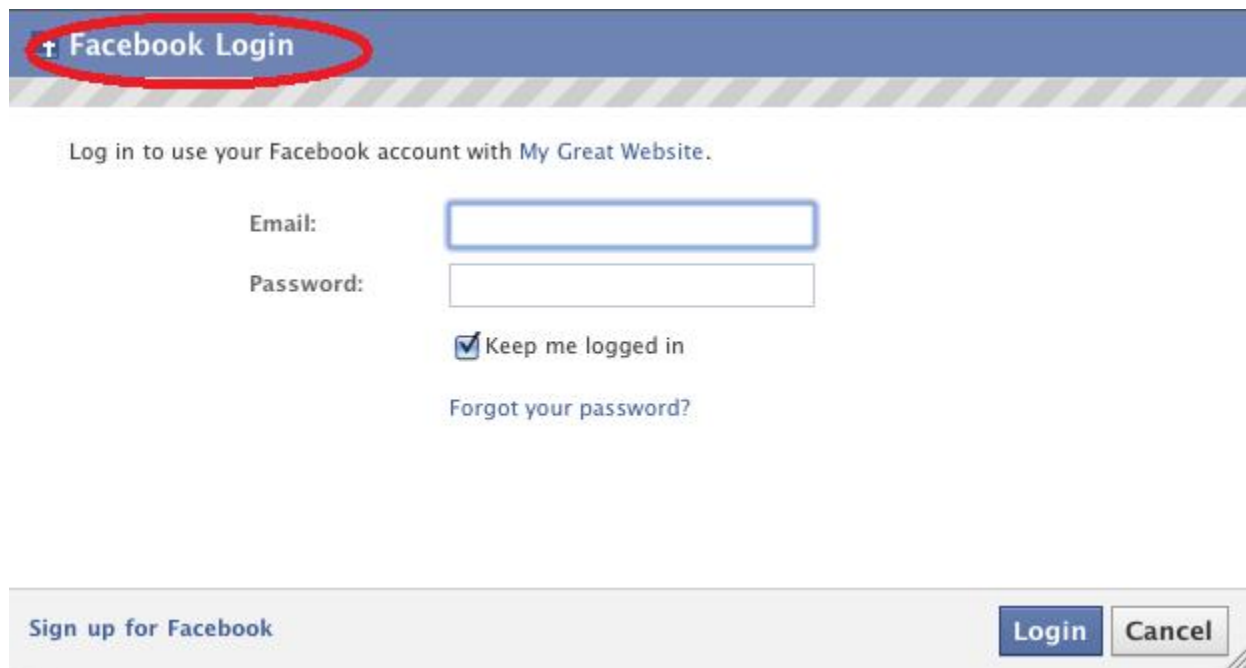
Once these steps were approved, my application was issued user access token. Using this token, my application got the permission to access user's information.

4.2.1.1 Server side flow

In facebook, User authentication and application authorization are handled by forwarding the user to the OAuth dialog. At the time I invoked this dialog, I had passed two parameters, which are:

- An “application id” – this was generated when I had created an application.
- An “URL” where my browser will be redirected back, once application authorization completes. My redirected URL must be within the same domain as the site URL that was specified in website tab of my application.

Here if the user is already logged in then system validates the login cookie that has been stored in user's browser and this way it authenticates the user. And if the user is not logged in then he/she has to enter his/her email id and password to log into the system. This is shown in figure 16.



Facebook Login

Log in to use your Facebook account with My Great Website.

Email:

Password:

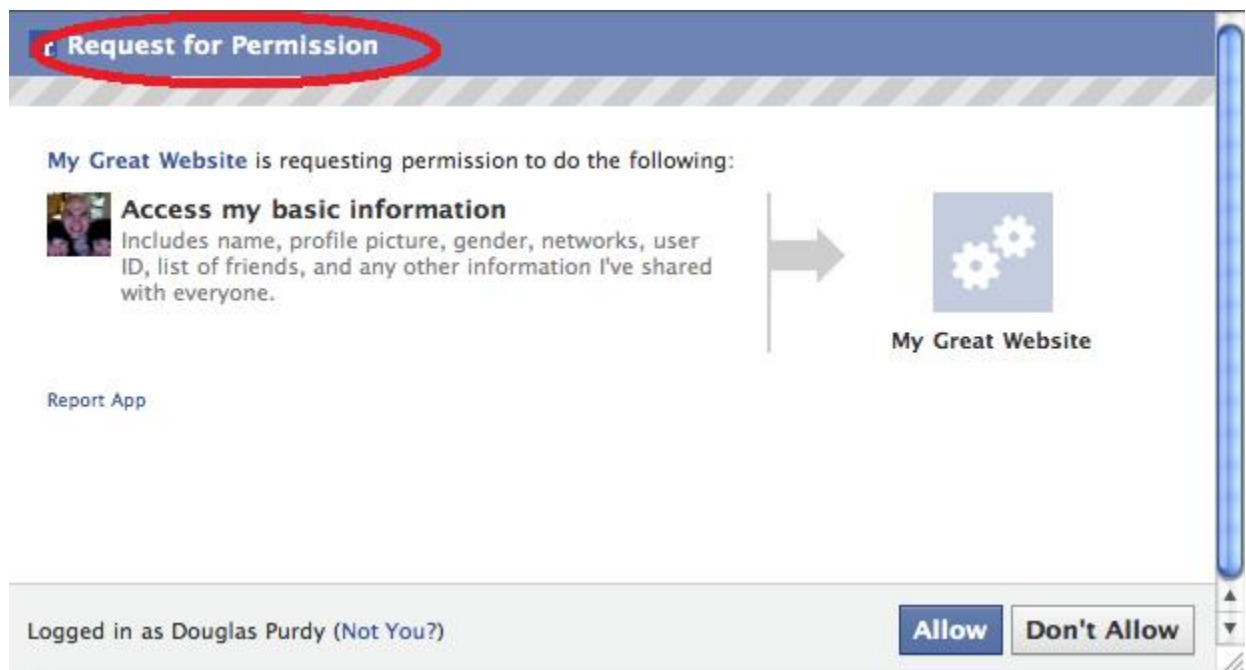
☒ Keep me logged in

[Forgot your password?](#)

[Sign up for Facebook](#) [Login](#) [Cancel](#)


[Figure 16: Facebook Login Page]


Once the user is authenticated successfully then the OAuth dialog ask user to authorize the application. This is shown into figure 17.



Request for Permission

My Great Website is requesting permission to do the following:

 **Access my basic information**
Includes name, profile picture, gender, networks, user ID, list of friends, and any other information I've shared with everyone.

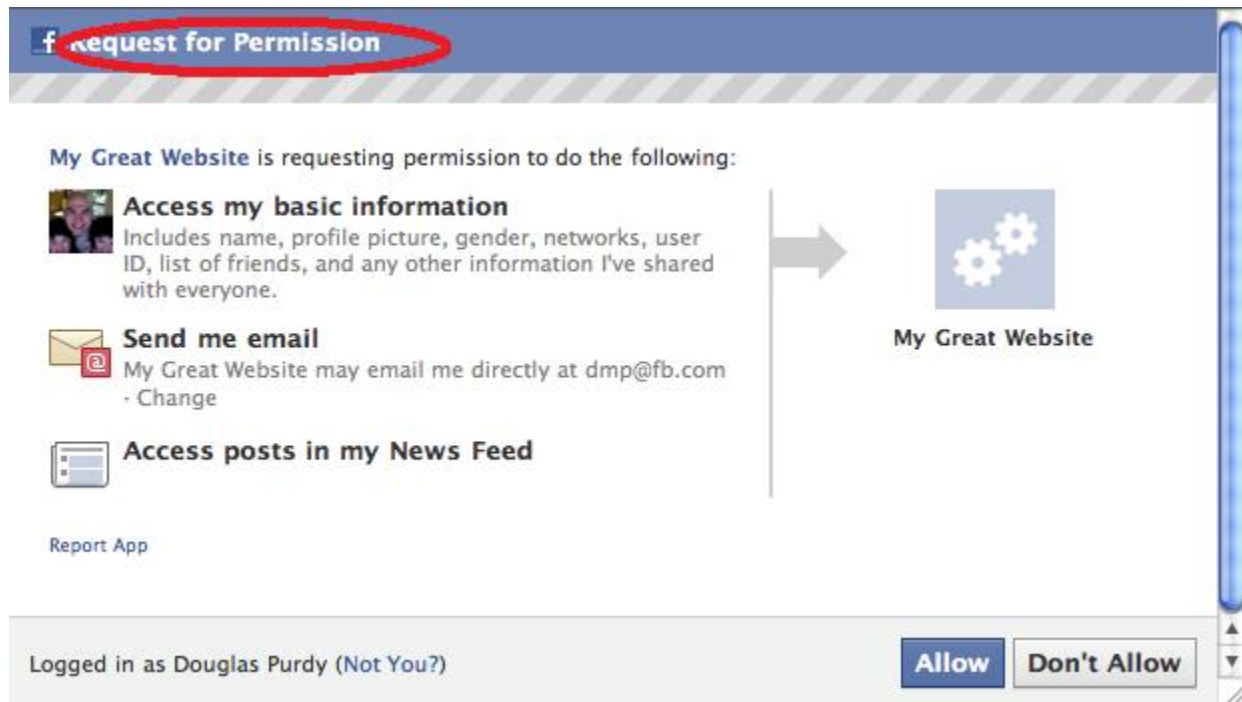

My Great Website

[Report App](#)

Logged in as Douglas Purdy (Not You?) [Allow](#) [Don't Allow](#)

[Figure 17: request for permission for basic information page]

If in my application, I want to access the basic information of user that is available publically or on facebook then first user has to authorize my application. If in my application, I need more than the basic information to perform the particular function then I must request the specific permissions by adding a scope parameter to the OAuth dialog as depicted in figure 18.



[Figure 18: Request for permission more than basic information page]

The greater the number of permissions is asked, the lesser the number of users that grant them. So I only requested the permissions those were actually needed in my application.

If any user selects “Don’t Allow” button when the OAuth dialog asks them to authorize the application, then my application will not be authorized and the OAuth Dialog will redirect the user's browser to the URL that I had passed as a redirected URL parameter. Here it will redirect the user to that URL and also show the following error with that:

[http://your_URL?error_reason=user_denied&](http://your_URL?error_reason=user_denied&error=access_denied&error_description=The+User+denied+your+request)

`error=access_denied&error_description=The+User+denied+your+request`.

Only if the users select the “Allow” button then my application will be authorized and then the OAuth Dialog will redirect the user's browser to the URL that I had passed as a redirected URL parameter. Here with that it also sends the authorization code.

[http://your_URL?code=A Code provided By Server](http://your_URL?code=A_Code_provided_By_Server)

After this was done, the next step was to do the application authentication by getting the access token. To gain the access token, I interacted with the OAuth API.

For the application authentication, I passed the authorization code and the application secret to the Graph API token end point. The application secret was generated at the time I had created the application. If my application is successfully authenticated and the authorization code is valid then the authorization server returns the access token. If there is a problem in authenticating my application then the authorization server issues an HTTP 400 error and returns this error as a response.

```
{
  "error": {
    "type": "OAuthException",
    "message": "Error validating verification code."
  }
}
```

4.2.1.2 Client side flow

At the client side, I again used the OAuth Dialog for user authentication and application authorization. The only difference between the server side flow and client side flow is that in client side flow, I mentioned the response type with the token value. Same as the server side flow, I could have requested the additional permissions using the scope parameter.

Now here system checks for facebook session key. If it's not available then application is redirected to the facebook login page. After successful login, facebook API calls canvas URL which redirects users to my application. When facebook redirects users to my application, it gives authentication token id and facebook session key as described in previous chapter.

4.2.2 Authorization

To have the feelings of good personalized user experience, facebook sends user the application information about the user. I had passed this information to the Canvas URL by using the HTTP POST method. In this method, I had passed a signed request parameter which had the encoded JSON object. In facebook, to get the access of all user information, first I need to authorize the application. Once my application is authorized, the user's information will be

available for my application. For this, I had used the OAuth Dialog. To invoke this dialog, I had redirected the user's browser to the following URL.

`https://www.facebook.com/dialog/oauth?`

`client_id=MY_APP_ID&redirect_uri=MY_CANVAS_PAGE`

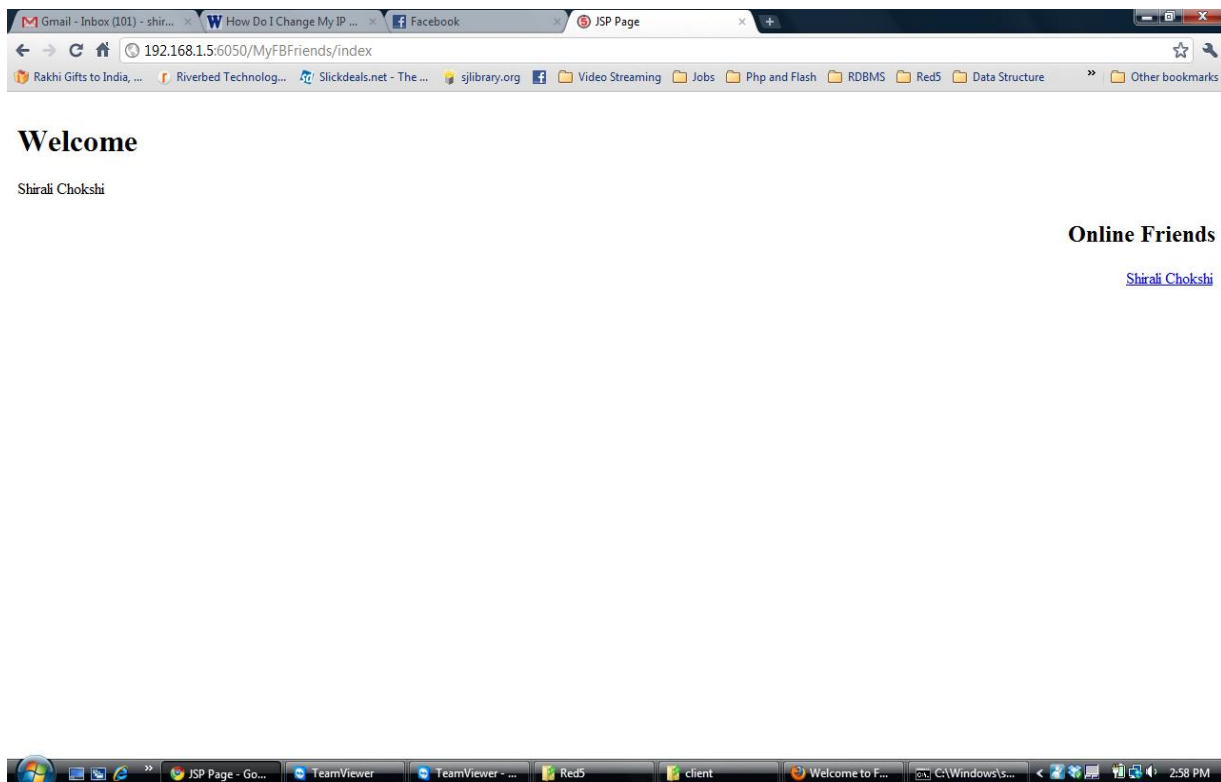
The application is authorize by user only if user selects **ALLOW** button of the request permission page.

5. User Experience

5.1 User Feedback

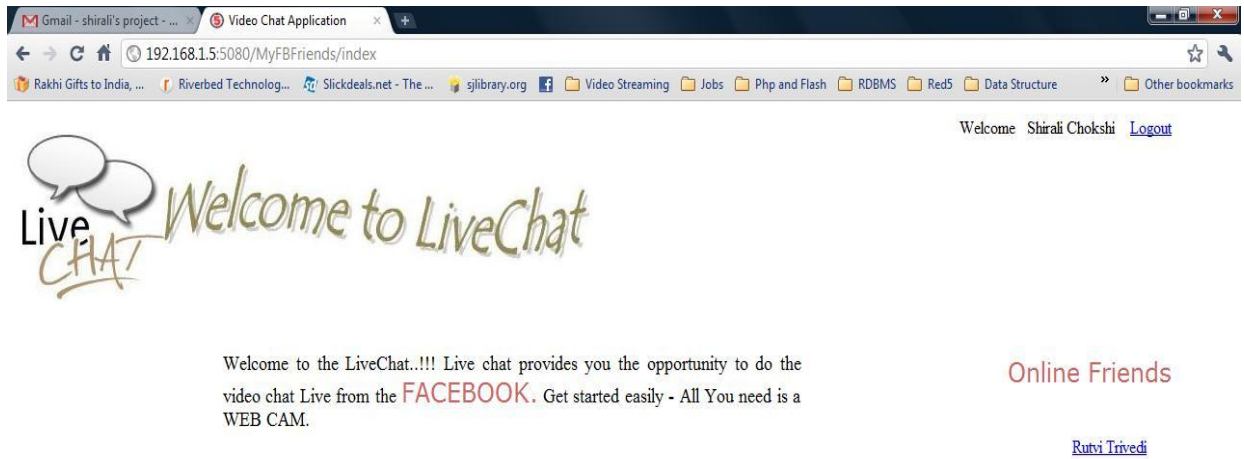
After implementing the application, I had asked many users to try out this application. By doing this I came to know if there were any bugs in the application. I also got different suggestions from the users to make the design of my application more users friendly. Some of the suggestions are as follow:

- Initially as shown in figure 19, the home page of my application was not attractive and user friendly. It only contained the list of online friends in a right side of a page. New users were not able to understand what this application is used for and what it does.



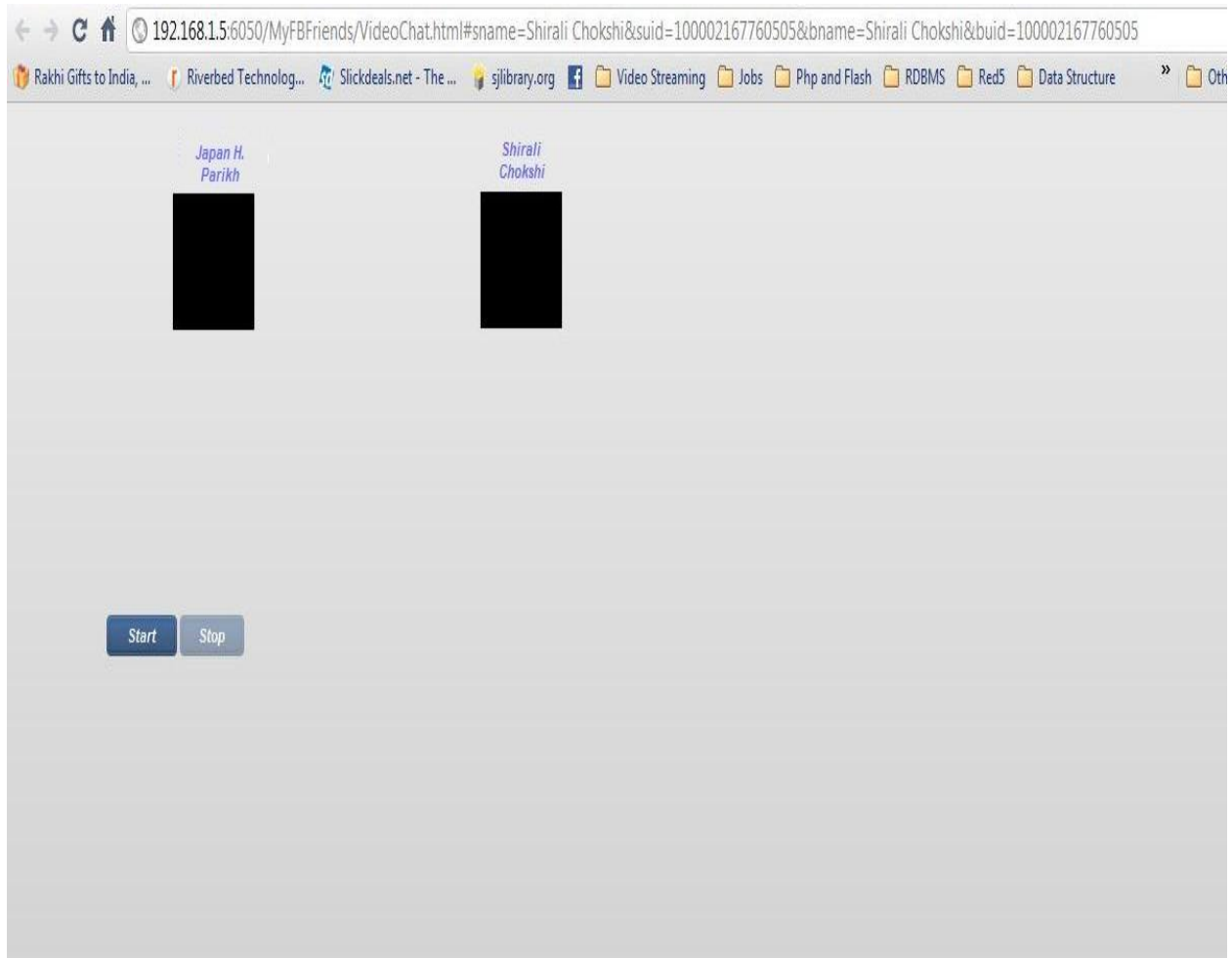
[Figure 19: Previous home page]

Later on from the feedback of users, I had created the welcome page with the description of the application. This helps a lot to new users. It gives a basic overview of what this application does and how it works. This is shown in figure 20.



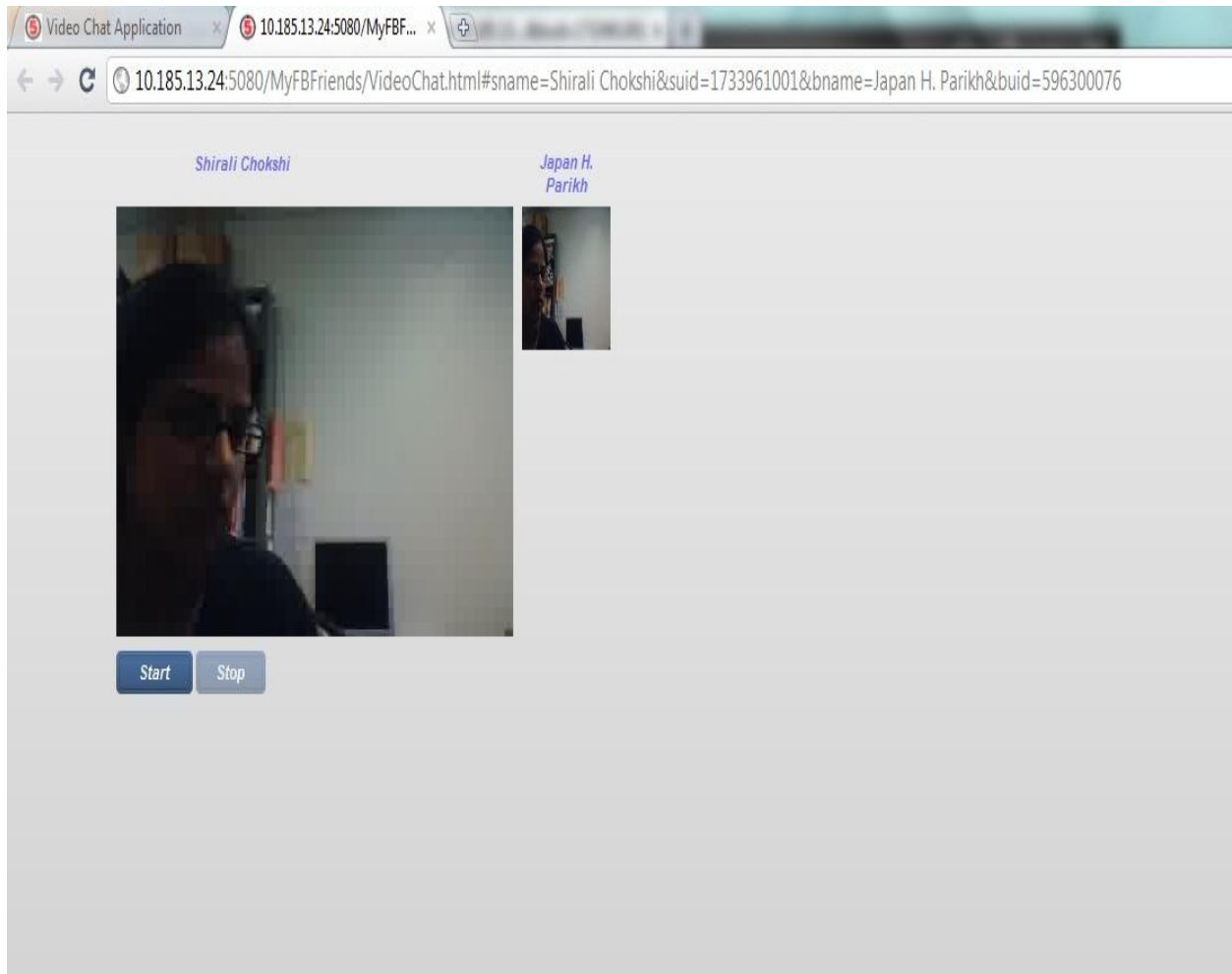
[Figure 20 : Current Home Page]

- The users had suggested me to change the size of the video chat window. Initially I was showing the small size of video chat window. The sizes of both the person's windows were same. This is shown in figure 21.



[Figure 21: The same size of window]

Later on, based on the feedback, I had changed the sizes of the windows. I had followed the look and size of the windows as provided by Google. I had kept the larger size of the window for the person with whom user talks and the user can see him/her in a smaller size of window. This is shown in figure 22.



[Figure 22: Current Size of Windows]

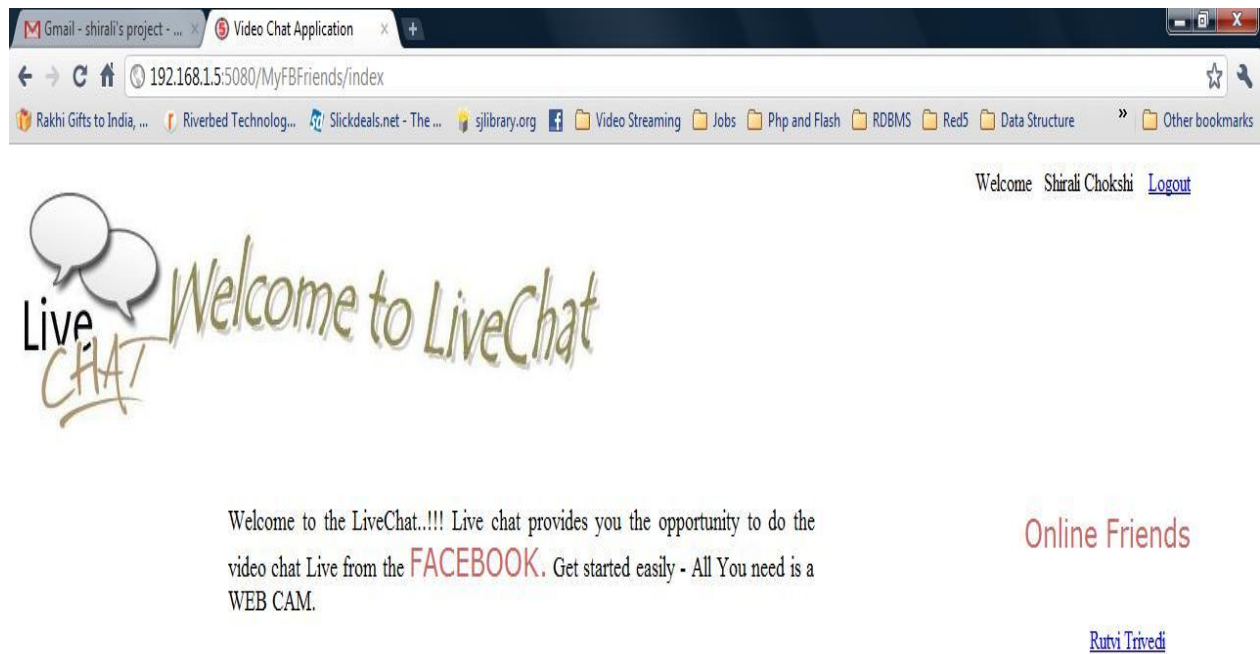
Once I had made changes based on the feedback provided by users, I asked them to try it again. I also asked them to give their feedback on my improved version.

Users tried my application with the improved version. Now users are satisfied with the improvements and they are happily using this application.

5.2 Visualization of Application

1. Welcome Page

On a welcome page, user can see a welcome note and list of online friends.



[Figure 23 : Welcome Page]

2. Facebook Login page

If user is not logged into facebook then before displaying welcome page, application navigates user to the facebook login page. Once user enters user id and password, then user will be navigated to welcome page of the application.

facebook

[Sign Up](#) Facebook helps you connect and share with the people in your life.

Facebook Login

Log in to use your Facebook account with MyFBFriends.

Email:

Password:

☐ Keep me logged in

[Login](#) or [Sign up for Facebook](#)

[Forgot your password?](#)

[English \(US\)](#) [Español](#) [Português \(Brasil\)](#) [Français \(France\)](#) [Deutsch](#) [Italiano](#) [العربية](#) [हिन्दी](#) [中文\(简体\)](#) [日本語](#) »

Facebook © 2011 [Mobile](#) [Find Friends](#) [Badges](#) [People](#) [Pages](#) [About](#) [Advertising](#) [Create a Page](#) [Developers](#) [Careers](#) [Privacy](#) [Terms](#) [Help](#)

[Figure 24 : Facebook Login Page]

3. Permission page

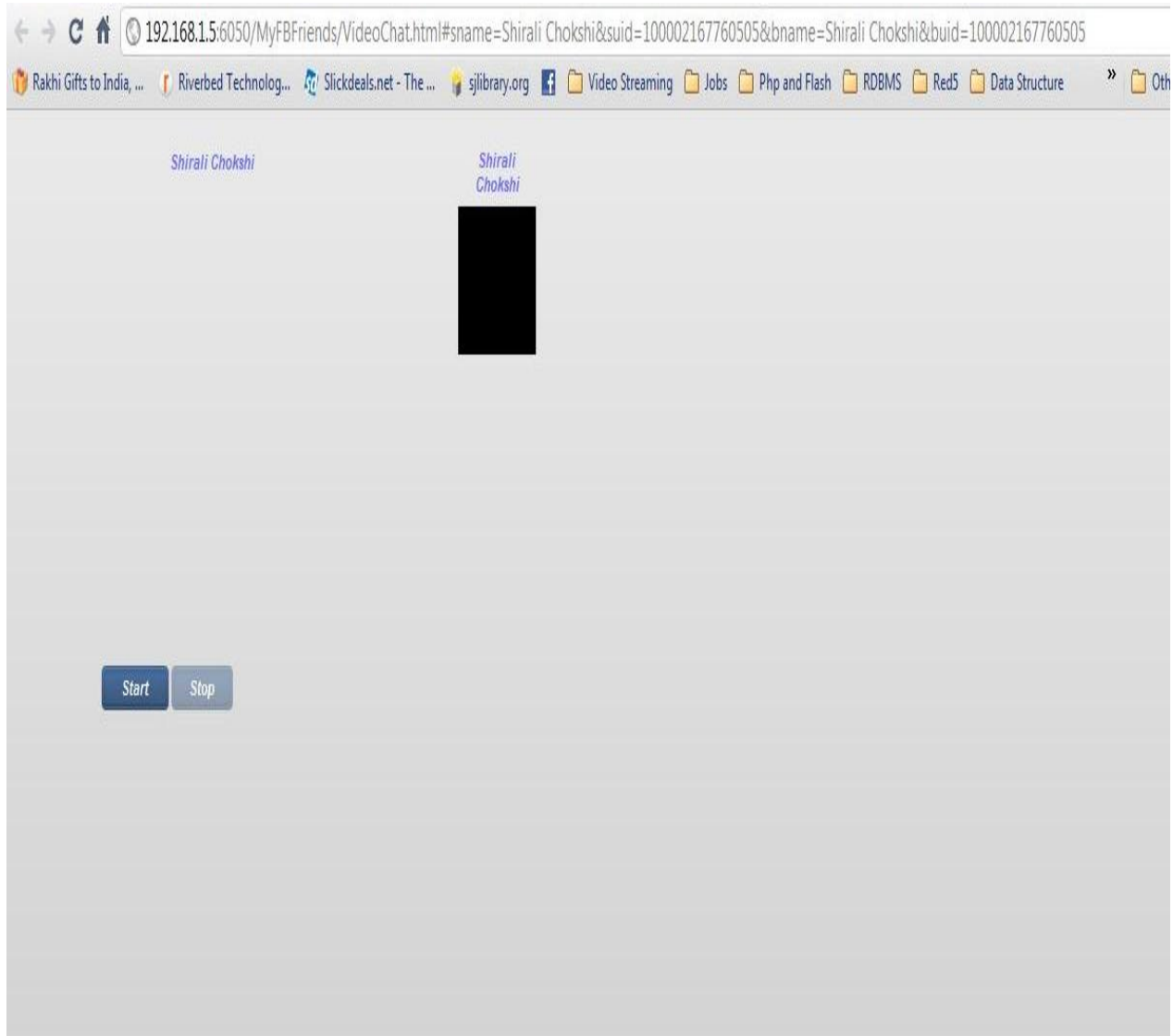
If users use this application for the first time then application asks users for the permission to use their basic information from their facebook account.



[Figure 25 : User Permission]

4. Video Chat Window

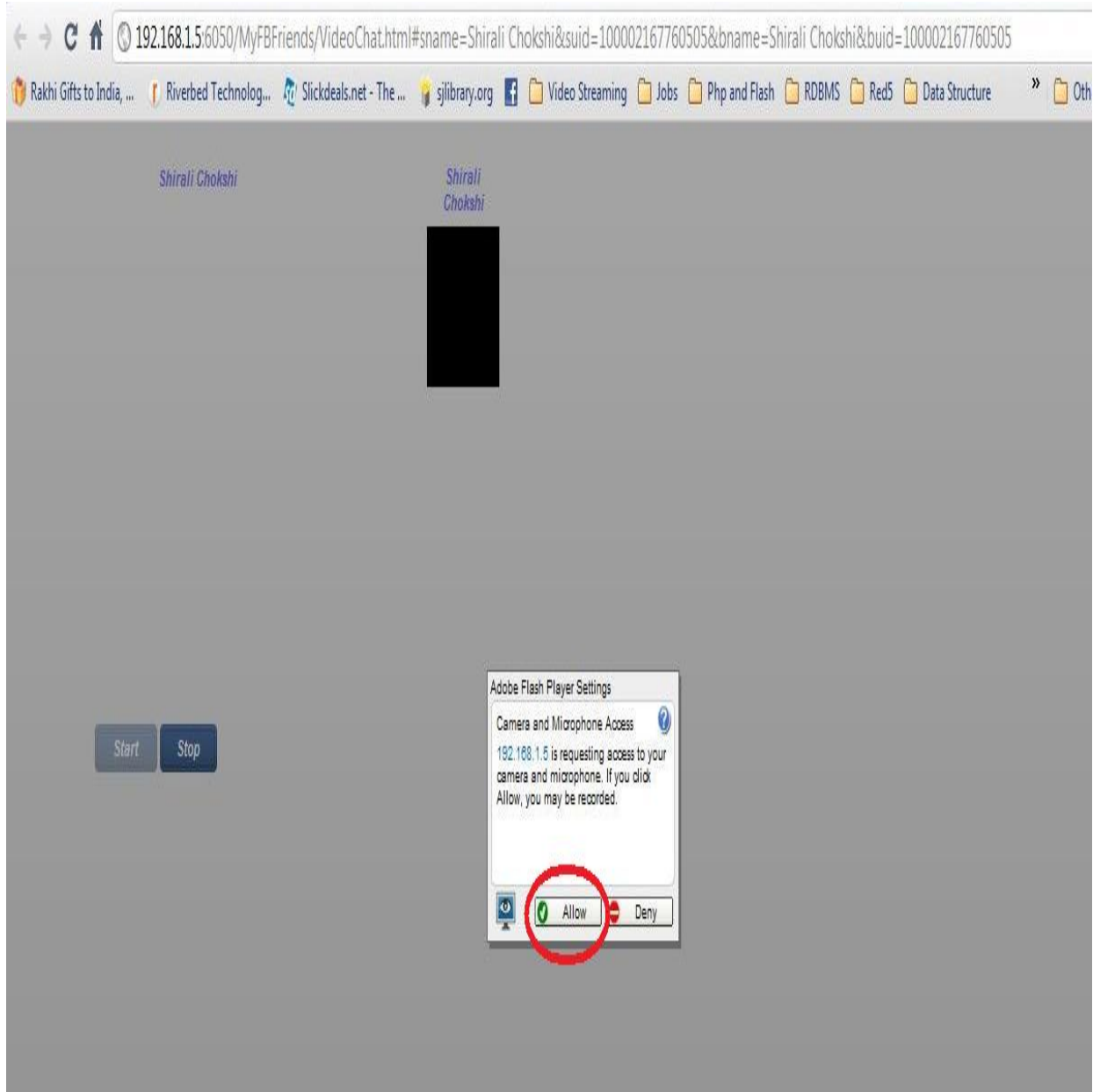
Once user clicks on any of online friends, application navigates user to a new window. This new window displays the screens where user can see the video. Using this, user can do video chat.



[Figure 26 : Video Chat Window]

5. Permission to start Camera

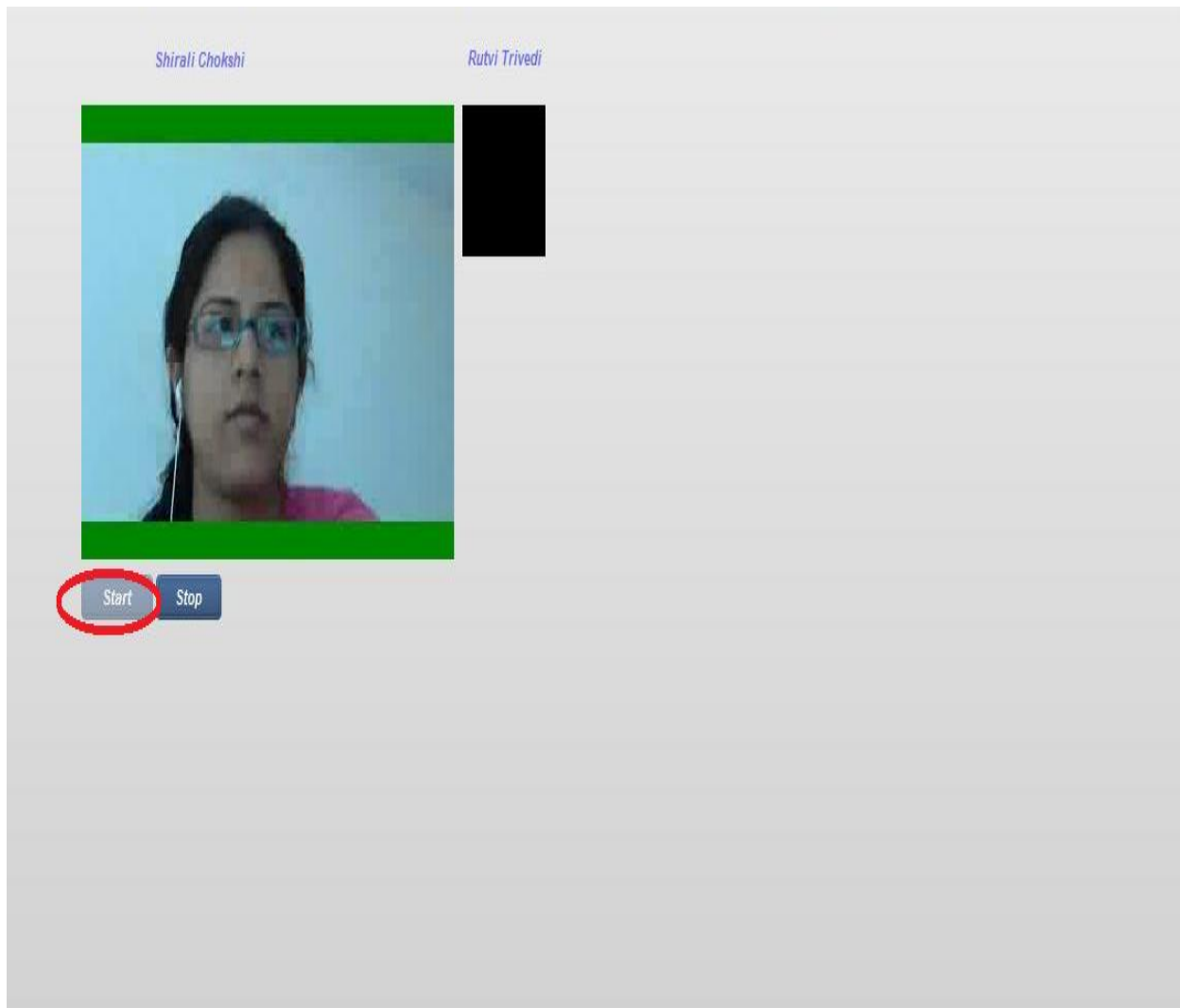
Once user clicks on “start” button on the video chat window, application asks for the permission to start the camera. If user clicks on “Allow” then only camera starts capturing video.



[Figure 27 : Permission to start camera]

6. Capturing video

Once user gives the permission to start the camera, it starts capturing video.



[Figure 28 : capturing video]

6. Conclusion

Today social networking sites are the place where you can find everything to be connected with each other. These sites have become part of our life. Among all social networking sites, facebook is a site that is used by many people. As we all know that facebook is lacking in providing the video chat functionality, I had decided to develop it as my master's project. First I successfully finished the implementation of a simple video chat application and then I had merged it with the facebook.

During the development of this application, I learnt lots of things. I came to know that how to create an application in facebook and how to use such application in real life. I also learnt how to use facebook login feature in any other application and how to create the secure authentication of users. I learnt the Flash to implement my client. I got an experience of using an open source product for my own application.

After successful implementation of my project, it was given to a group of users to test it. Users gave me the feedback and based on the feedback I had made changes to the application. Users also tested the improved version of my application. According to them, now this application works fine and they are happily using this application.

This report has included the requirement of video chat application especially for facebook, the application work flow, in depth information of red5 streaming server, step by step procedure of how to create an application in facebook, how to use the features of facebook for any other application and visual screen shots of the application.

This project presents a software application that can be used by any user in his/her everyday life. The video chat application makes user's life better by providing a good way of communication.

References:

1. Alex Rodriguez, 2008. Restful Web Services: the Basics, <http://www.ibm.com/developerworks/webservices/library/ws-restful/>
2. Greg Wilkins, May 2008. Jetty Vs. Tomcat: A Comparative Analysis, <http://www.webtide.com/choose/jetty.jsp>
3. Wikipedia, http://en.wikipedia.org/wiki/Comparison_of_web_server_software
4. The Apache Software foundation, <http://mina.apache.org/features.html>
5. Tsavo, 2006. Begin Red5 Development, <http://jwamicha.wordpress.com/2006/12/14/red-5-beginners-guide/>
6. Cirtex, 2005. Creating an application with red5 and flash, <http://blog.cirtex.com/2010/06/21/creating-an-application-with-flash-and-red5-is-simple-with-this-tutorial/>.
7. Dominick Accattato, CTO infrared5, <http://www.newviewnetworks.com/nvnhome/blog/client/?p=62>.
8. Red5 streaming tutorial, 2009. <http://lelandcope.com/beginner-red5-streaming-tutorial-%E2%80%93-part-3-of-3-2/>
9. Techno Gumbo: Red5 Media Server Development Setup Tutorial, <http://www.technogumbo.com/tutorials/Red5-Media-Server-Development-Setup-Tutorial/Red5-Media-Server-Development-Setup-Tutorial.php>
10. Red5 tutorials, http://www.red5tutorials.net/index.php/Tutorials:Red5_and_Flex_2_on_Windows_XP.
11. FLX, Training and Solutions for the Flash Platform. <http://www.flashextensions.com/tutorials.php>
12. Red5 and eclipse video tutorial, <http://red5world.com/index.php/red5/red5-and-eclipse-video-tutorial>
13. Stack Overflow, <http://stackoverflow.com/questions/1546479/simple-remote-shared-object-with-red5-flash-server>.
14. Adobe Help resource Center: Creating a shared object, http://livedocs.adobe.com/flex/3/html/help.html?content=Isos_3.html.
15. Tallbrunette22, 2007. <http://www.youtube.com/user/tallbrunette22#p/p>.

16. Facebook developers. <http://developers.facebook.com/>
17. E. Hammer-Lahav, Ed., Yahoo!, D. Recordon, facebook, D. Hardt, Microsoft, January 2011. The OAuth 2.0 Authorization Protocol, <http://tools.ietf.org/pdf/draft-ietf-oauth-v2-12.pdf>.
18. Facebook-Java- api, <http://code.google.com/p/facebook-java-api/>

Appendix A: Additional images of Facebook interaction

facebook  Search  Home Profile Find Friends Account ▾

Create Application [Back to My Apps](#)

Essential Information

App Name Cannot contain Facebook trademarks or have a name that can be confused with an app built by Facebook.

Terms Do you agree to the Facebook Terms?

☐ Agree ☒ Disagree

[Create App](#)

Create an Ad

Every Child Is An Artist



Every Child is An Artist offers so much more than just painting a portrait or drawing some fruit. Learning experts insist that education

[Aiden](#)

[Figure_Apx 1: Create new Application Page in facebook]

Facebook interface showing the "Edit myfirstfbapp" page. The page is divided into sections: About, Web Site, Facebook Integration, Mobile and Devices, Credits, and Advanced. The "Canvas" section is highlighted, showing settings for the application's Canvas page and URL. The "Canvas Page" field is circled in red, indicating the URL: `http://apps.facebook.com/myfirstfbapp/`. The "Canvas URL" field is empty. The "Secure Canvas URL" field is empty. The "IFrame Size" section has two options: "Show scrollbars" (selected) and "Auto-resize". The "Bookmark URL" field is empty. The "Core Settings" section shows the Application ID (144083088995632) and Application Secret (923335cc391451a06fa525125e3f7aef). The "Your OAuth client_id" and "Your OAuth client_secret" fields are also present.

Facebook interface showing the "Edit myfirstfbapp" page. The page is divided into sections: About, Web Site, Facebook Integration, Mobile and Devices, Credits, and Advanced. The "Canvas" section is highlighted, showing settings for the application's Canvas page and URL. The "Canvas Page" field is circled in red, indicating the URL: `http://apps.facebook.com/myfirstfbapp/`. The "Canvas URL" field is empty. The "Secure Canvas URL" field is empty. The "IFrame Size" section has two options: "Show scrollbars" (selected) and "Auto-resize". The "Bookmark URL" field is empty. The "Core Settings" section shows the Application ID (144083088995632) and Application Secret (923335cc391451a06fa525125e3f7aef). The "Your OAuth client_id" and "Your OAuth client_secret" fields are also present.

[Figure_Apx 2 : Canvas Page and Canvas URL of an application]