WEB APPLICATION WITH ROLE BASED ACCESS

**SUBMITTED BY:**

1. KRISHNA S.
2. PRIYA PRASAD.
3. V ROHITH.
4. ROHIT K.

**COLLEGE:** GOVERNMENT ENGINEERING COLLEGE, THRISSUR

**DEPARTMENT:** COMPUTER SCIENCE AND ENGINEERING

**SEMESTER:** S7

INTRODUCTION

The purpose of this project was to develop a web application with role based access. We developed this application under the guidance of Ms. Gadha P S, Scientist/Engineer SF at **Enterprise application software division of Vikram Sarabhai Space Center**. The Vikram Sarabhai Space Centre (VSSC) is a major space research centre of the [Indian Space Research Organization](https://en.wikipedia.org/wiki/Indian_Space_Research_Organisation) (ISRO), focusing on rocket and space vehicles for India's [satellite](https://en.wikipedia.org/wiki/Satellite) programme.  It is located in [Thiruvananthapuram](https://en.wikipedia.org/wiki/Thiruvananthapuram), in the Indian state of [Kerala](https://en.wikipedia.org/wiki/Kerala). It was renamed in honour of Dr. [Vikram Sarabhai](https://en.wikipedia.org/wiki/Vikram_Sarabhai), often regarded as the father of the Indian space program. The Vikram Sarabhai Space Centre is one of the main research and development establishments within ISRO. VSSC is an entirely indigenous facility working on the development of sounding rockets.

We were a group of four members, Krishna S, Priya Prasad, V Rohith, Rohit K, pursuing Bachelor of technology in Computer Science and Engineering , seventh semester, under APJ Abdul Kalam Technological University at Government Engineering College, Thrissur.

The duration of this project was from 01.07.2019 – 15.08.2019. Throughout this project, we learned in depth about developing a web application from scratch, under the guidance of our mentor. We learned more about how to integrate various components of a web application, database creation and management, java servlet components and JSP during the development of this project.

TABLE OF CONTENT

|  |  |  |
| --- | --- | --- |
| SERIAL NO | CONTENT | PAGE NO |
| 1 | Executive Summary | 3 |
| 2 | Technologies   * MVC architecture * Java servlet * Java Server Pages * Apache Tomcat | 4  4  5  6  7 |
| 3 | Design description   * Workflow * Login page * User home page * Editor home page * Admin home page * Database * Role based access | 8  8  9  10  11 |
| 4 | Conclusion | 12 |
| 5 | References | 13 |

EXECUTIVE SUMMARY

In this project, we developed a web application with role based access. There are three kinds of users, namely, Designers, Editors and Admin. Every user can log into the application with their username and password. Every entered username and password is checked against values already stored in the database for their validity. Any invalid credentials will prevent the user from logging in to the application. If the username and password is valid, then we retrieve the role associated with this user from database and accordingly they were taken to respective home pages. If the role associated with a user is of Designer, then the user will be taken to Designer home page, where the designer will have provision to upload files and to enter details about files like filename and version. This detail will be stored in a table named data in the database. Designer can only upload zip files, otherwise it will show an alert message. Trying to upload files without entering filename will also give alert message about empty fields. If a user is of role Editor, after logging in, the user will be taken to Editor home page. Editor home page will list all the uploaded files along with their entered details and also have the provision to download and verify files. All the verified files will be taken to a folder named verified. Admin home page will have a list of verified files and a list of pending files to be verified.

Eclipse is the integrated development environment used for developing the application. It contains a base workspace and an extensible plug-in system for customizing the environment. Application resources used for developing this application are java servlet, java server pages, tomcat server and MySQL for database.

**TECHNOLOGIES**

MVC ARCHITECTURE

We followed MVC architecture to build this application. The MVC is an architectural pattern that separates an application into three main logical components

1. **M**odel
2. **V**iew
3. **C**ontroller

Hence the abbreviation MVC. Each component is built to handle specific development aspect of an application.

**MODEL:** The model component stores data and its related logic. This represents data that is being transferred between controller components or any other related business logic. For example, a Controller object will retrieve the customer info from the database. It manipulates data and send back to the database or use it to render the same data.It responds to the request from the views and also responds to instructions from the controller to update itself. It is also the lowest level of the pattern which is responsible for maintaining data.

**VIEW**: A View is that part of the application that represents the presentation of data. Views are created by the data collected from the model data. A view requests the model to give information so that it resents the output presentation to the user. The view also represents the data from charts, diagrams, and table. For example, any customer view will include all the UI components like text boxes, drop downs, etc.

**CONTROLLER:** The Controller is that part of the application that handles the user interaction. The controller interprets the mouse and keyboard inputs from the user, informing model and the view to change as appropriate. A controller send's commands to the model to update its state (E.g., Saving a specific document). The controller also sends commands to its associated view to change the view's presentation (For example scrolling a particular document).

**JAVA SERVLET**

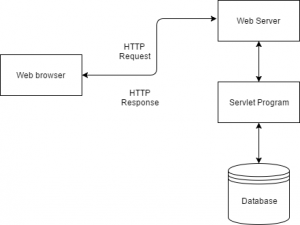
Servlets are the Java programs that runs on the Java-enabled web server or application server. They are used to handle the request obtained from the web server, process the request, produce the response, and then send response back to the web server.  
**Properties of Servlets:**

* Servlets work on the server-side.
* Servlets capable of handling complex request obtained from web server.

**Execution of Servlets:**

Execution of Servlets involves the six basic steps:

1. The clients send the request to the web server.
2. The web server receives the request.
3. The web server passes the request to the corresponding servlet.
4. The servlet processes the request and generate the response in the form of output.
5. The servlet send the response back to the web server.
6. The web server sends the response back to the client and the client browser displays it on the screen.
7. The following diagram shows servlet architecture:



**JAVA SERVER PAGES**

Java Server Pages (JSP) is a technology for developing Webpages that supports dynamic content. This helps developers insert java code in HTML pages by making use of special JSP tags, most of which start with <% and end with %>.

A Java Server Pages component is a type of Java servlet that is designed to fulfill the role of a user interface for a Java web application. Web developers write JSPs as text files that combine HTML or XHTML code, XML elements, and embedded JSP actions and commands.

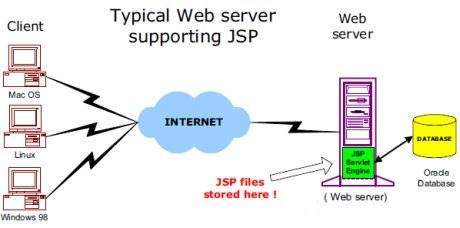
Using JSP, you can collect input from users through Webpage forms, present records from a database or another source, and create Webpages dynamically.

JSP tags can be used for a variety of purposes, such as retrieving information from a database or registering user preferences, accessing JavaBeans components, passing control between pages, and sharing information between requests, pages etc.

The web server needs a JSP engine, i.e., a container to process JSP pages. The JSP container is responsible for intercepting requests for JSP pages.

A JSP container works with the Web server to provide the runtime environment and other services a JSP needs. It knows how to understand the special elements that are part of JSPs.

Following diagram shows the position of JSP container and JSP files in a Web application.



APACHE TOMCAT SERVER

Apache Tomcat, is an open source web server and servlet container developed by the Apache Software Foundation. Basically, it implements the Java Servlet and the Java Server Pages (JSP) specifications from Sun Microsystem, and provides a "pure Java" HTTP web server environment for Java code to run in. In the simplest configuration Tomcat runs in a single operating system process. The process runs a Java virtual machine (JVM) and every single HTTP request from a browser to Tomcat is processed in the Tomcat process in a separate thread.

Apache Tomcat includes tools for configuration and management, but can also be configured by editing XML configuration files. Even with Java EE certification, Tomcat is an incredibly lightweight application. If offers only the most basic functionality necessary to run a server, meaning it provides relatively quick load and redeploy times compared to many of its peers, which are bogged down with far too many bells and whistles. This lightweight nature also allows it to enjoy a significantly faster development cycle.

Tomcat is quite flexible. You can run it in virtually any fashion you choose, and it’ll still work as intended. The fact that it’s open-source helps as well, since you can tweak it to fit your needs, provided you’ve the knowledge to do so.

Tomcat is an extremely stable platform to build on – and using it to run your applications will contribute to your server’s stability, as well. This is because Tomcat runs independently of your Apache installation – even if a significant failure in Tomcat caused it to stop working, the rest of your server would run just fine.

DESIGN DESCRIPTION

WORKFLOW

VIEW

VERIFY

UPLOAD

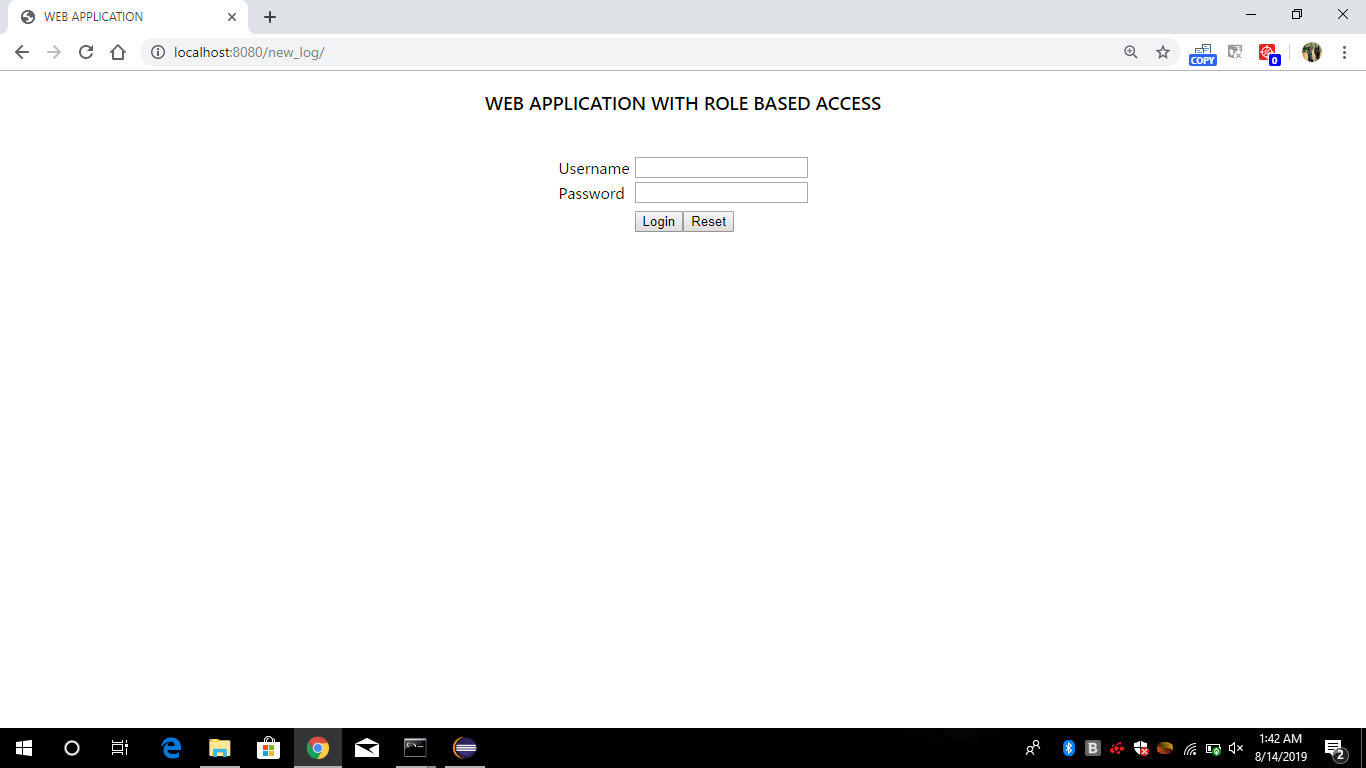
DESIGNER EDITOR ADMIN

There are three kind of users as shown in the workflow diagram, designer, editor and admin. Each user will perform different kind of function. Designer can upload files and can also enter file related information into database. This entered detail will be stored in a separate table in the database. Designer can only upload zip files. Alert message will be shown if we try to upload with empty field or non-zip files. Editor will get a list of all uploaded files. Editor is the one who verifies the file uploaded by designer. Editor can also download the files if needed. Admin is like a central authority who keeps an eye on every activity. Admin home page will list all the verified files and files which are waiting to get verified.

**LOGIN PAGE**

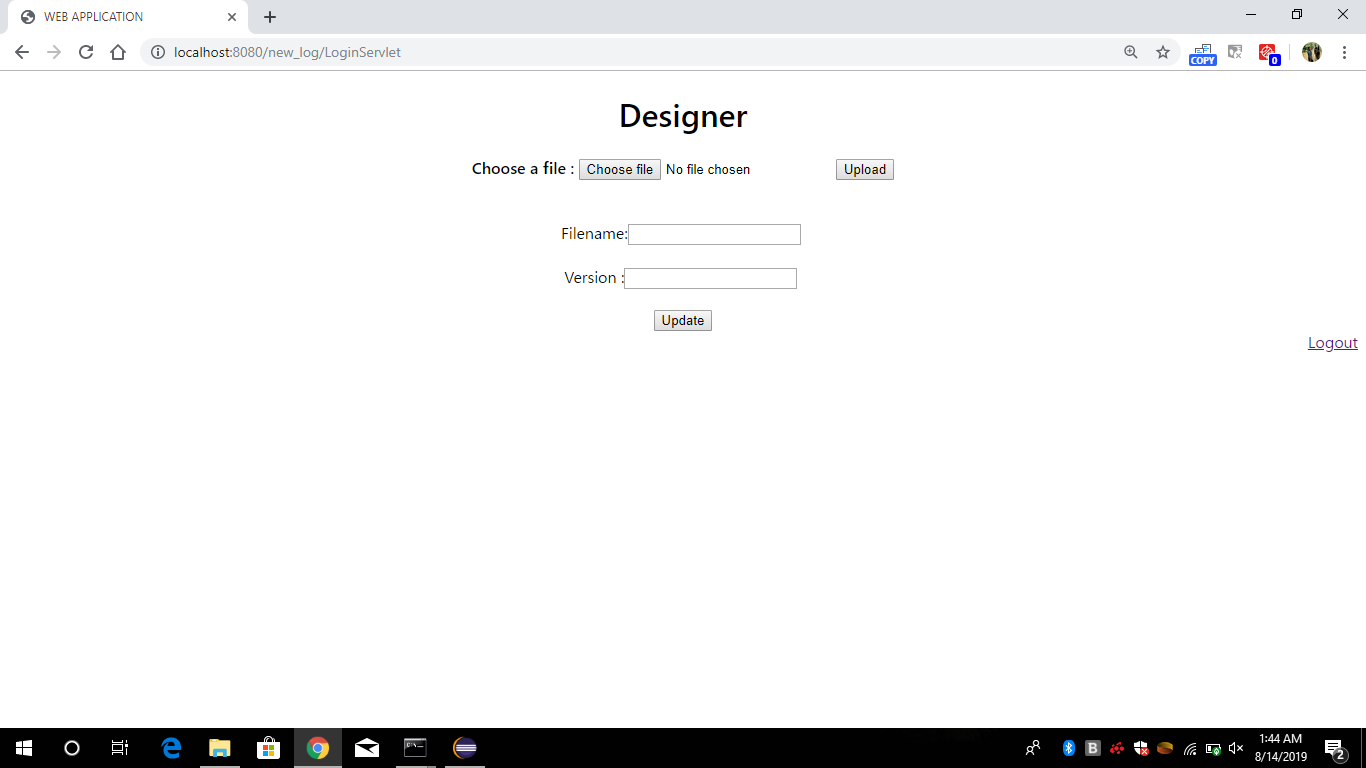
Every user will have their own username and password with which they can login to the application. When a user enters username and password, entered details is checked against the details already stored in the database. If the username and password is valid, then it checks the kind of role associated with this user in the database and accordingly they were taken to respective home pages. For example, if the username matches with the role of a designer, then the user will be taken to designer homepage, same for Editor Role and Admin Role. Any invalid credentials will prevent the user from logging in to the application.

User login page will be of the form:



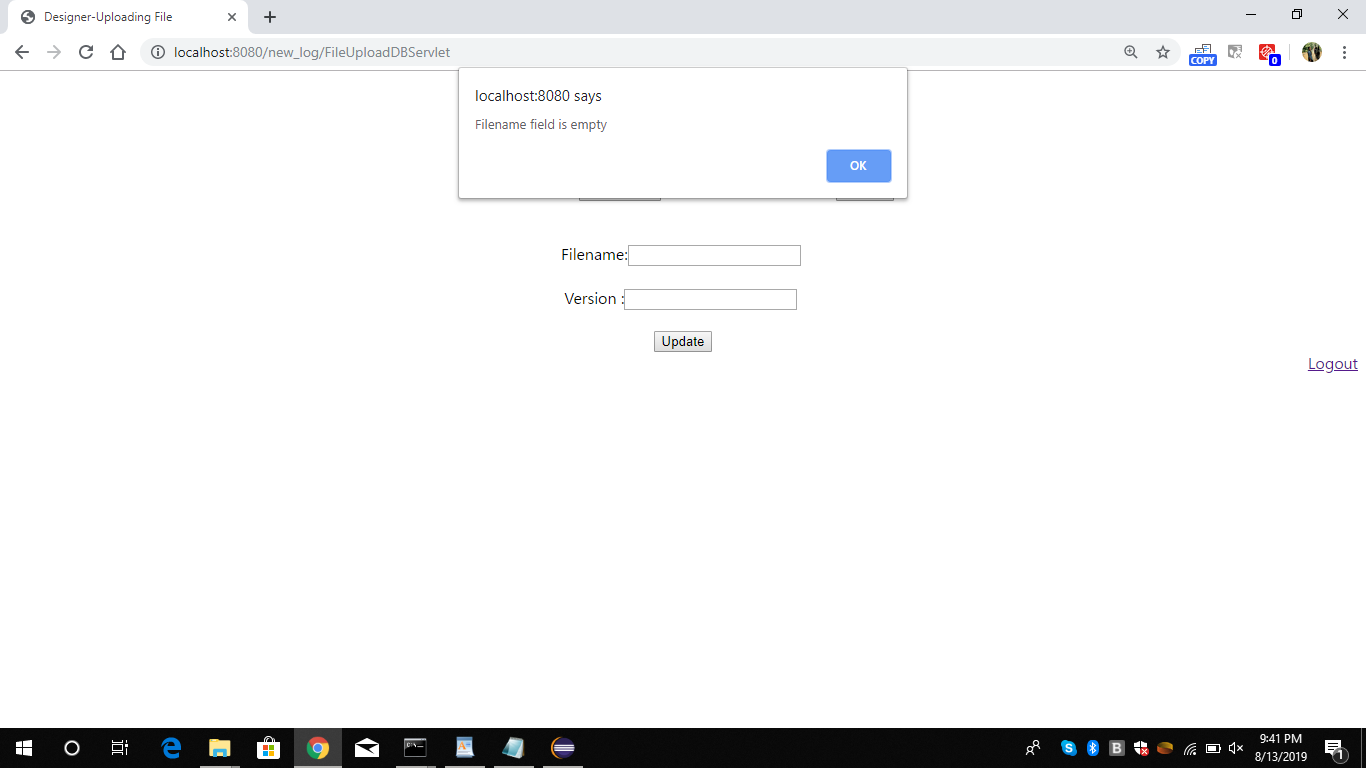
**DESIGNER HOME PAGE**

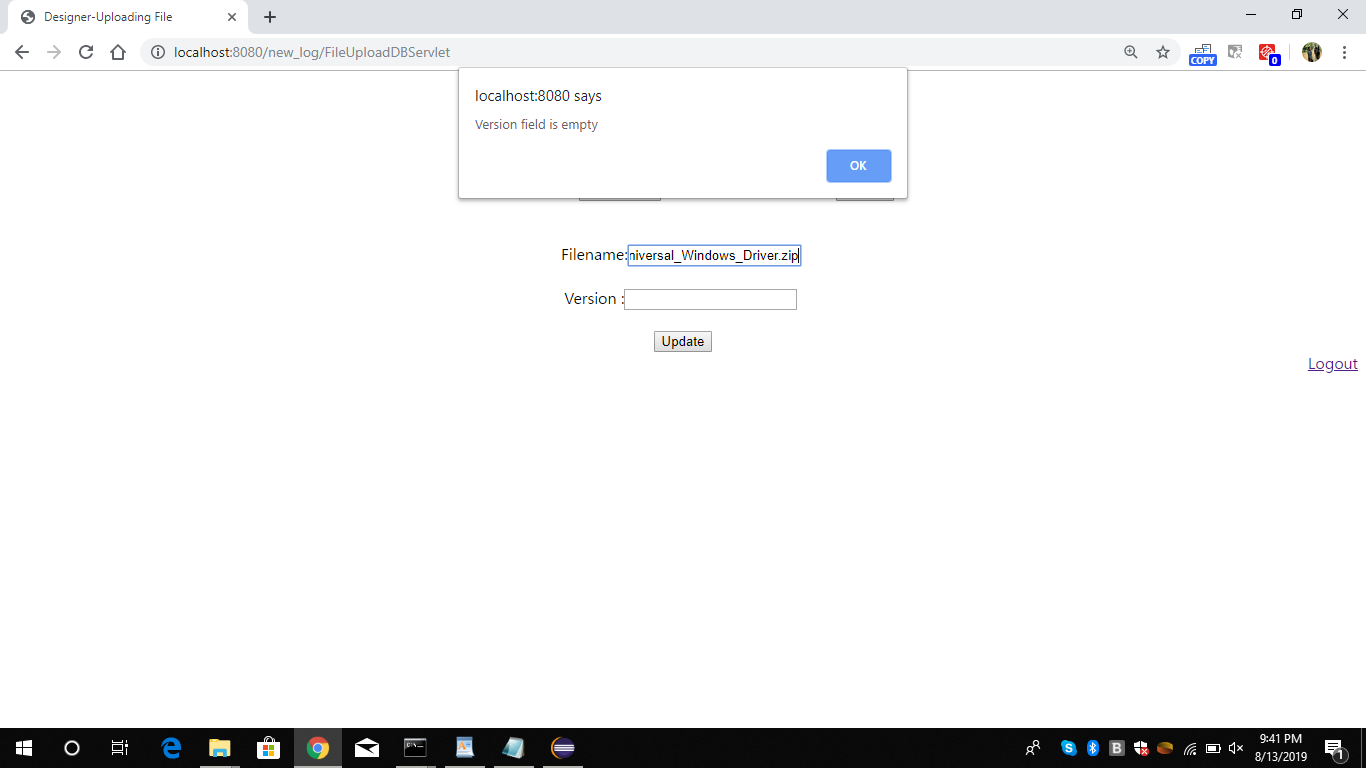
If the role of a user matches with designer, user will be taken to designer home page. Designer home page will have the provision to upload zip files. Any non-zip files, if uploaded will give alert message. Designer home page will also have the fields to enter data related to uploaded file like filename and version. Keeping the fields empty when we are uploading, will also give alert message.



VALIDATIONS

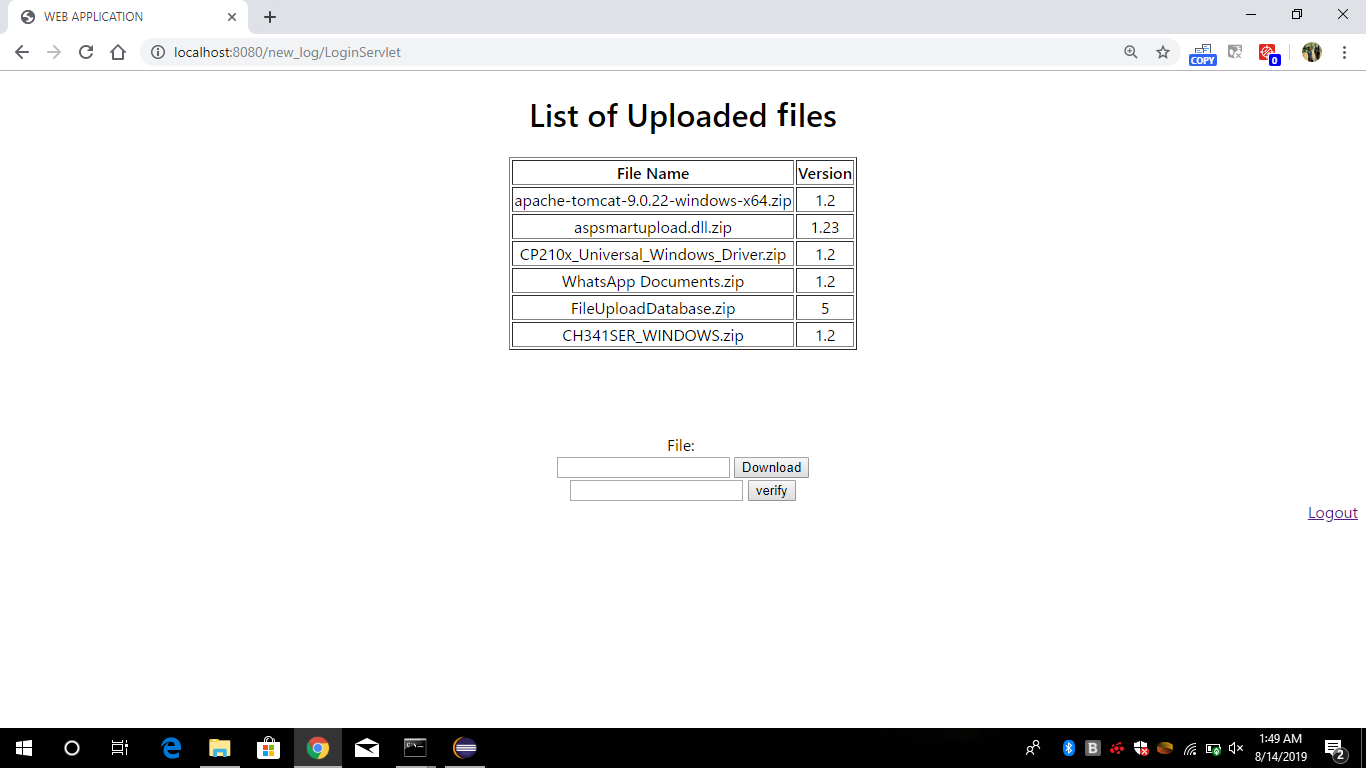
Whenever designer tries to upload any non-zip files, an alert message will be shown as designer an only upload zip files. Submitting, keeping the detail fields empty, also shows alert message about empty fields.





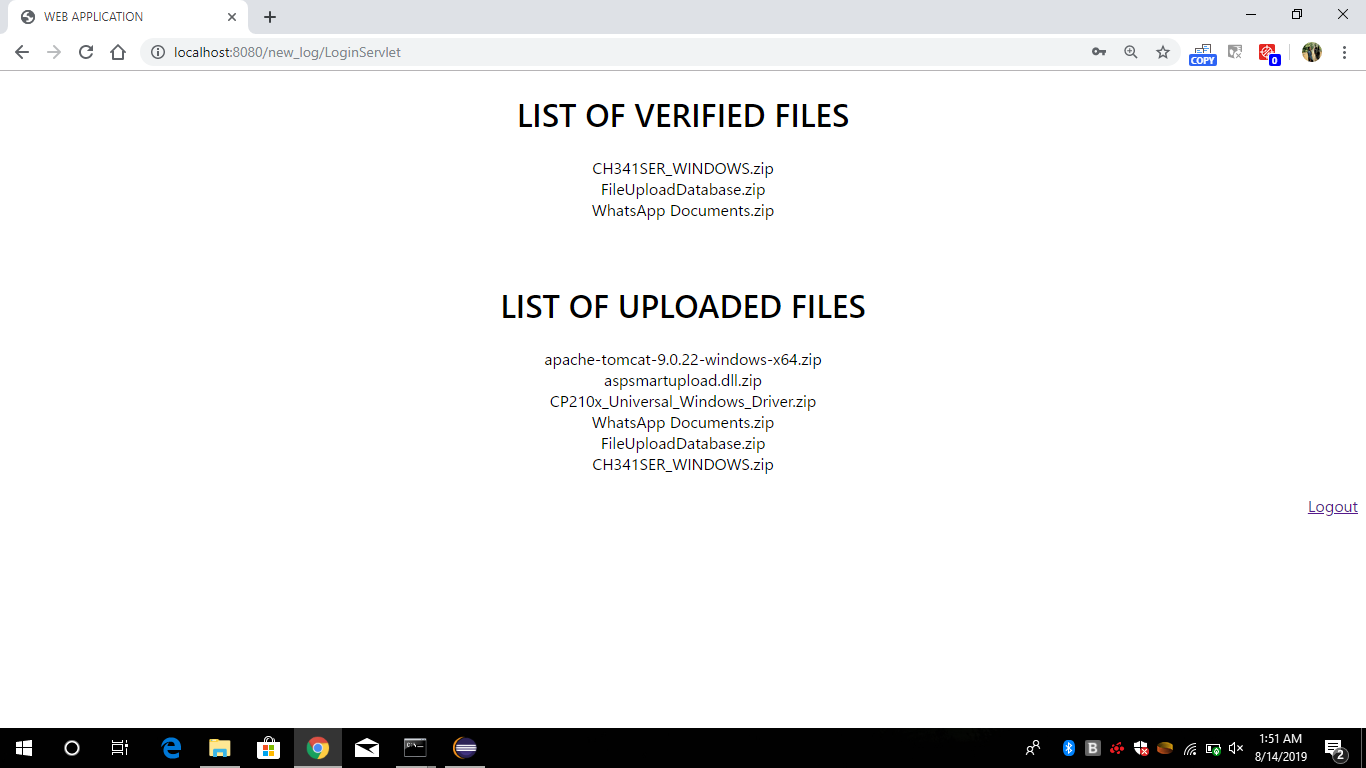
EDITOR HOME PAGE

Editor can view all the uploaded files by user and can verify it. All the designer uploaded files will be stored in a folder named uploaded and Editor will get a list of uploaded files along with the entered details regarding the uploaded files. Editors also have the provision to download any uploaded files .All the verified files by the editor will be taken to a folder named verified.



ADMIN’S HOME PAGE

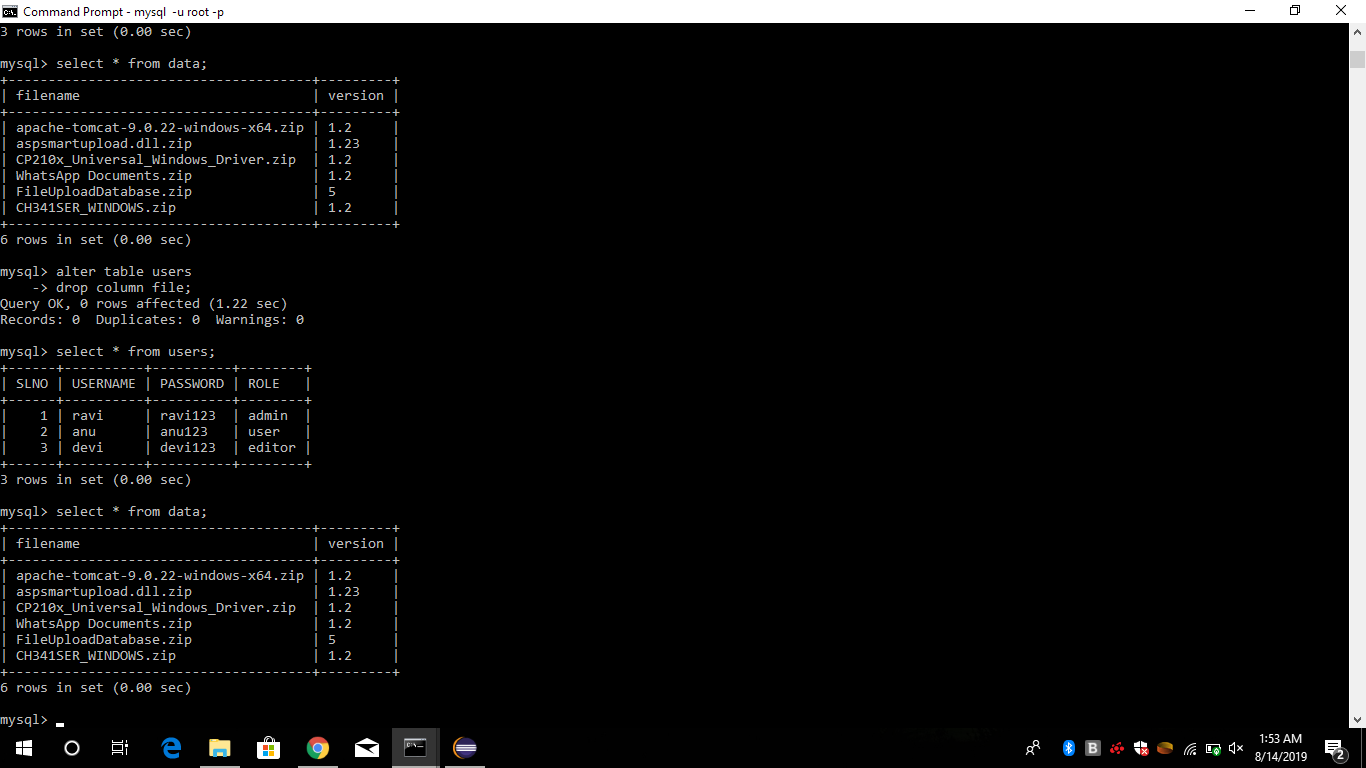
Like every other user, Admin will also login to the application with their valid username and password. After logging in, they were taken to Admin’s home page. Admin home page will list all the verified files by the editor and also give a list of files which are still waiting to get verified.



**DATABASE**

We used MySQL to create database. MySQL is a relational database management system based on SQL – Structured Query Language. The application is used for a wide range of purposes, including data warehousing, e-commerce, and logging applications. The most common use for MySQL however, is for the purpose of a web database.

We have created a table named users which contains attributes USERNAME, PASSWORD and ROLE. When a user enters username and password in login page, entered value is checked against this stored value in the database and the value of role attribute is used to take the user to appropriate home pages. There is also a table named data, where the details entered by designer, about files is stored.



**ROLE BASED ACCESS**

Each user, when logged in are taken to respective homepages according to the role defined for the user in the database. When a user tries to login to the application, after checking the validity of entered username and password, the role associated with the user is retrieved from the database and accordingly they were taken to respective home pages.

**CONCLUSION**

We built a web application which gives each user a specific role and specific kind of access to the web page. Designer can upload files, editors will verify the uploaded files and Admin can view all the details of the uploaded files. We used technologies like Java servlet, JSP, Tomcat server and MySQL to build this project. Integrated development environment used for building this project is Eclipse IDE**.**

Through this project, we learned more about MVC Architecture and how to build an application using java servlet components. MVC stands for Model, View and Controller. MVC separates application into three components - Model, View and Controller. Model represents shape of the data and business logic. View is a user interface and Controller handles the user request. [Servlet](https://www.edureka.co/blog/java-servlets) is a server-side [Java](https://www.edureka.co/blog/what-is-java/) program module that handles client requests and implements the servlet interface. Servlets can respond to any type of request, and they are commonly used to extend the applications hosted by web servers.

We also learned about database creation and management. MySQL, a relational database management system based on SQL – Structured Query Language is used to build database. The application is used for a wide range of purposes, including data warehousing, e-commerce, and logging applications. The most common use for MySQL however, is for the purpose of a web database.

Altogether, we learned about how to build an application from scratch, how to work with an IDE, various java servlet components, JSP, uses of Tomcat server and database creation and management.

**REFERENCES**

* <https://stackoverflow.com/>
* <https://www.geeksforgeeks.org/>
* [https://www.guru99.com](https://www.guru99.com/)
* <https://www.wikipedia.org/>
* <https://www.javatpoint.com/servlet-tutorial>