

**Software Engineering and Web Technologies Laboratory**

**Integrated Project**

**On**

**ACTIVITY MAGAZINE**

**Bachelor of Engineering**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

***Submitted By***

**Team No: 05**

Soumya Katagihalli 01FE20BCS002 102

Neha Patil 01FE20BCS006 106

Srishti Kadam 01FE20BCS010 110

Sneha Pamali 01FE20BCS035 135

**Faculty In charges**

Dr. Padmashree Desai and Prof Indira Bidari

**SCHOOL OF COMPUTER SCIENCE & ENGINEERING**

**HUBLI–580 031 (India)**

**Academic year 2022-23**

|  |  |
| --- | --- |
| **Table of Content** |  |
| **Chapters** | **Page**  **No** |
| **1. Introduction**  1.1 Preamble  1.2 Problem Definition  1.3 Objectives | **3**  **3**  **3**  **3** |
| **2. Software Requirement Specifications**  2.1 Functional Requirements and use case diagram  2.2 Non-Functional Requirements  2.3 Hardware and software requirements  2.4 Test plan and Test cases | **4**  **4**  **6**  **6**  **7** |
| **3. System Design & Implementation**  3.1 Architecture/Flowchart/ Block Diagram  3.2 Database Description   * 1. Modules description   3.4 Model View Control Architecture | **10**  **10**  **11**  **11**  **12** |
| **4. Results and Discussions**  4.1 Results/ Snapshots with description  4.2 Testing Report  4.3 Testing Tool  4.4 Continuous integration and continuous delivery (CI/CD) | **12**  **12**  **15**  **15**  **16** |
| **5. Conclusion & Future scope** | **16** |
| **Appendix**   1. **Roles and Responsibility within team** 2. **photos of meeting with customer** | **17**  **17**  **18** |

**1. INTRODUCTION**

* 1. **. Preamble**

The goal of the project is to use a web-based application to automate the school of computer science and engineering's activity and event management system. The user can login using the same username and password they used at registration, which saves their login information. The opportunity to add or update the event list is offered to the authorized individual. The added event is shown on the calendar as an upcoming event. It is also possible to keep track of the information students put on the registration forms provided to them in order to take part in activities.

**1.2. Problem Definition**

To design and develop a web application for the KLE Technical University's computer science department to keep track of events and activities.

**1.3. Objectives**

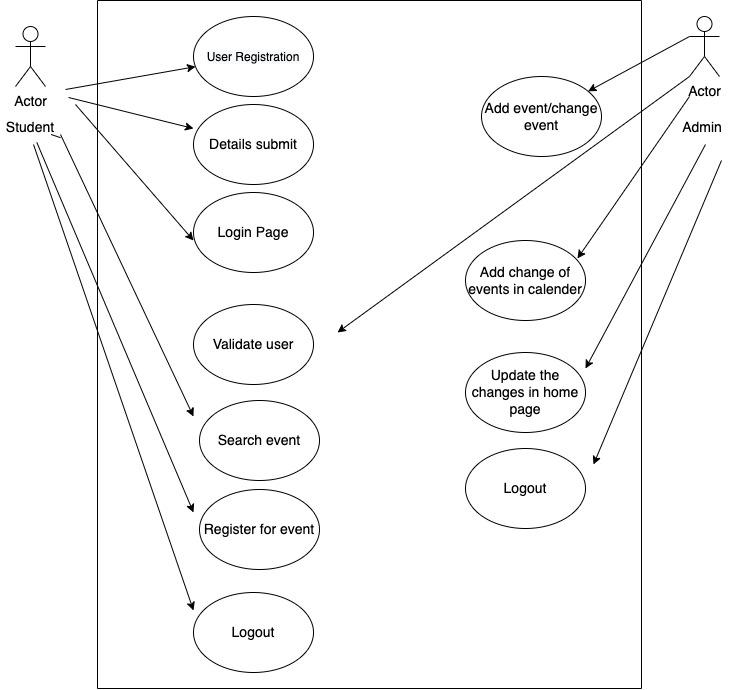
* To validate a student's credentials while login via a registration(signup) form.
* To offer the authorized person the chance to add, remove, or amend the calendar of events.
* To keep track of the data that students put on registration forms in order to take part in events.
* To display forthcoming events on a calendar for users to view.

**2. SOFTWARE REQUIREMENTS SPECIFICATIONS**

**2.1. Functional Requirements**

* The web application shall give the user access to information about recent and forthcoming events and activities held by the department.
* The web application shall have students as the end user and an admin to make changes/update the web page.
* The web application shall provide students with the access to register to any of the co-curricular activities/events being conducted.
* The web application shall store the details of students who have registered for certain events and display only to the admin.
* The web application shall give the visualization of the forthcoming events on the calendar.

**2.1. Use case diagrams**



**Fig 1. Use case diagram of Activity Magazine**

**2.2. Non - Functional Requirements**

* This web application should not have a delay of more than 2 seconds while switching between pages.
* Once registration is done by the student, the system should notify it to the students, every time they login to the web page.

**2.3. Hardware and Software Requirements**

**Hardware requirements:**

Processor: Intel(R) Core (TM) i5.

RAM: 8.00 GB.

**Software Requirements:**

Visual Studio with:

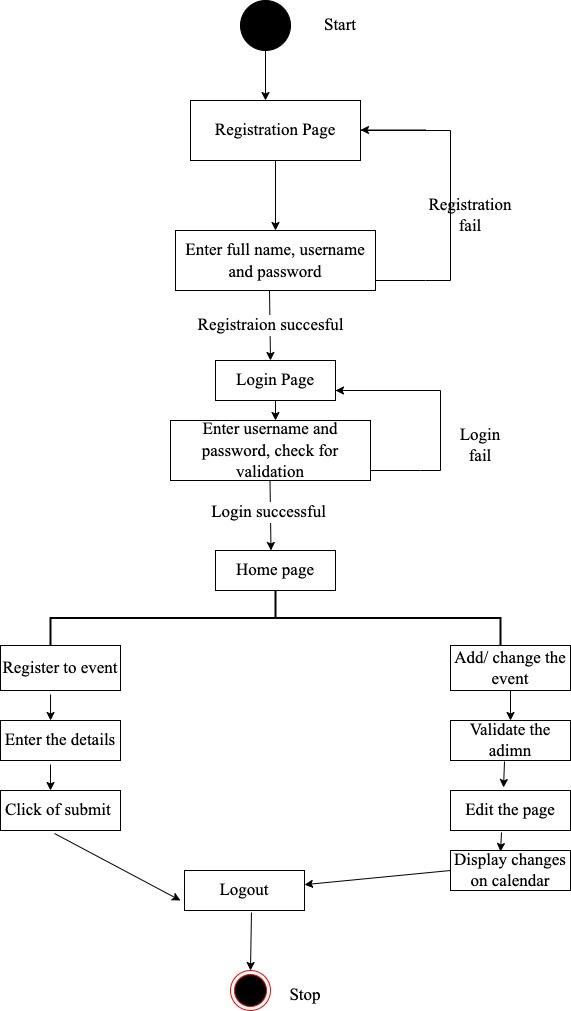
* NODE JS installation
* MONGODB

**2.4 Test plans and Test cases**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID | Test case id | Test case description | Test Steps | Test input Data | Expected Results | Actual Results | Pass/Fail |
| RE1 | T01 | Register the student with user name, full name and password, | Enter the username, password, full name in the signup page and click on signup. | Username, password, Full name. | Should alert the user with a successful registration message. | Alert message is given as “successful registration” | Pass |
| RE 02 | T02 | Login with the username and password as entered in registration. | Enter the username and password and click on the login button. | Validated username and password. | Should alert the user with a successful login message. And should go to the home page, the name of the user has to be displayed at the corner of the home page. | The user is directed to the home page and the name of the user is displayed at the corner of the home page. | Pass |
| RE 03 | T03 | Edit or upload the events to home page | The authorized user is given an email and password , they must login with a button. | Option though the button | Should direct to the page where editing takes place, where a person can change and amend the events. | The editing option is given to the user. | Pass |
| RE 04 | T04 | Edit the events. | After validating the authorized person, they can edit the events where names, location, dates are changed if needed or added. | Page and options. | The changes of events are made if needed. | According to the wish of the authorized person, changes are made. | Pass |
| RE 05 | T05 | View the added or amended events on the calendar. | After changes or addition of the events the option is given to view the name of the event on the calendar. | Option | The event name has to be displayed on the date of the event forthcoming. | The visualization is achieved via calendar display. | Pass |
| RE 06 | T06 | Register students to events. | The registration page is given to the user where they can enter the details and register themselves to the event. | Page | The registration successful message has to be sent to the user. | The registration successful message is sent. | Pass |
| RE 07 | T07 | Log out the user | click on the name of the user at the corner of the page where the logout option is given. | Option | The user should be redirected to the login page. | The user is redirected to the login page and logged out. | Pass |

**3. SYSTEM DESIGN AND IMPLEMENTATIONS**

**3.1 Flow Chart**

****

**Fig 2. Flow chart for Implementation**

**3.2 Database Description**

Non-relational document database MongoDB supports storage that is similar to JSON. The MongoDB database has full indexing support, replication, and a flexible data schema that makes it possible to store unstructured data. It also features comprehensive and user-friendly APIs.

Whenever the user registers with name, username and password the details entered are stored in the database. When this is directed to the login page, validation is done by comparing the credentials stored in the database while registration.

When the admin has made changes with events or added a new event, this information is stored in the database.

Our key role is to maintain the details entered by the students for registration, these are again stored in the dataset.

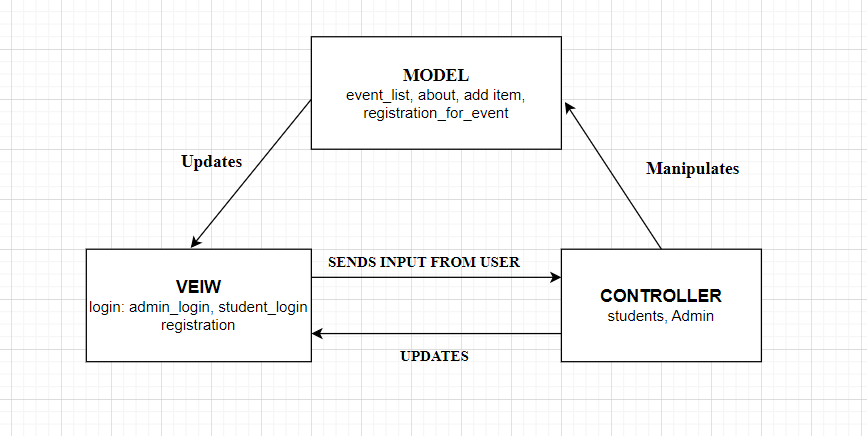
Further enhancements can be done to fetch the details that are display in our web application.

**3.2 Module Description**

Our modules are organized into several components such as registration page, login page, home page, about page.

Registration page will provide the user an option to register with their complete name, username and password. The login page will verify the user name and password before redirecting to the home website, which has information about upcoming events and a about page with connections to the official social media pages. Home page also allows the students to register to events by putting details in supplied registration link. This permits admin or authorized individual to add a new activity’s details such as activity’s name, date, location. This website also allows the user to log out whenever needed.

* 1. **Model View Controller Architecture**

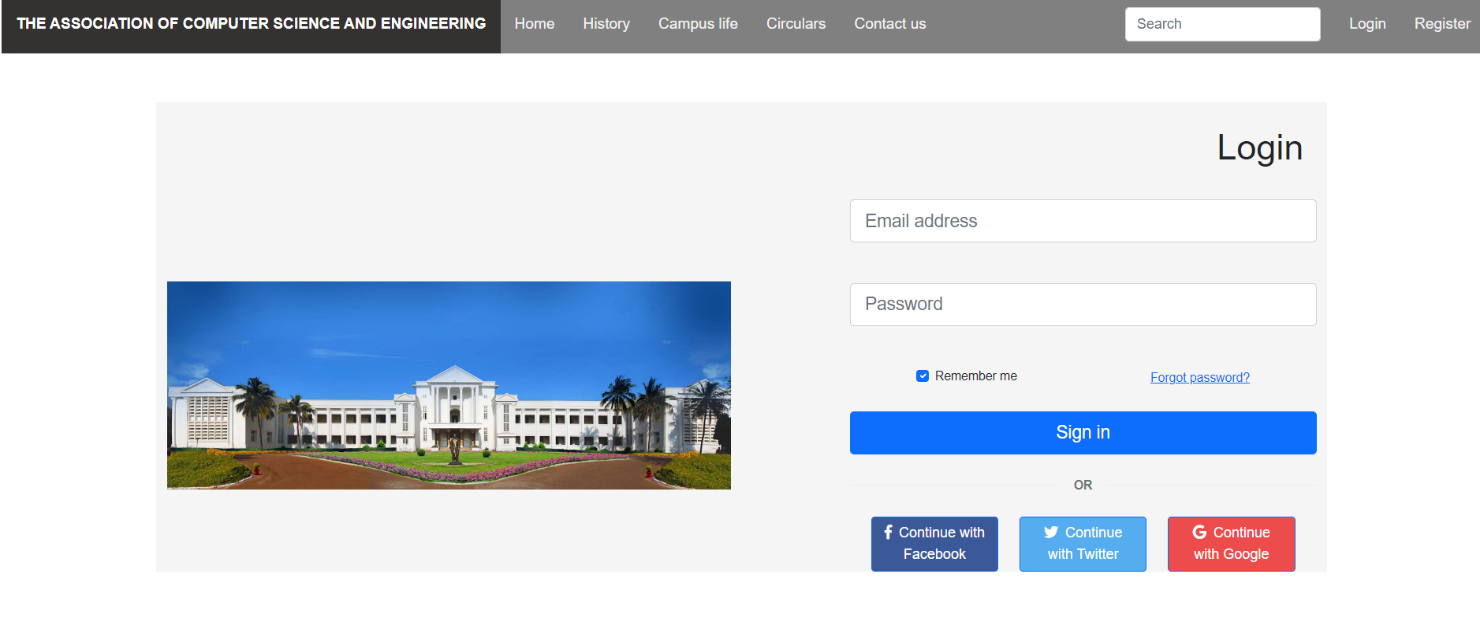
****

**Fig 3. MVC of Activity Magazine System**

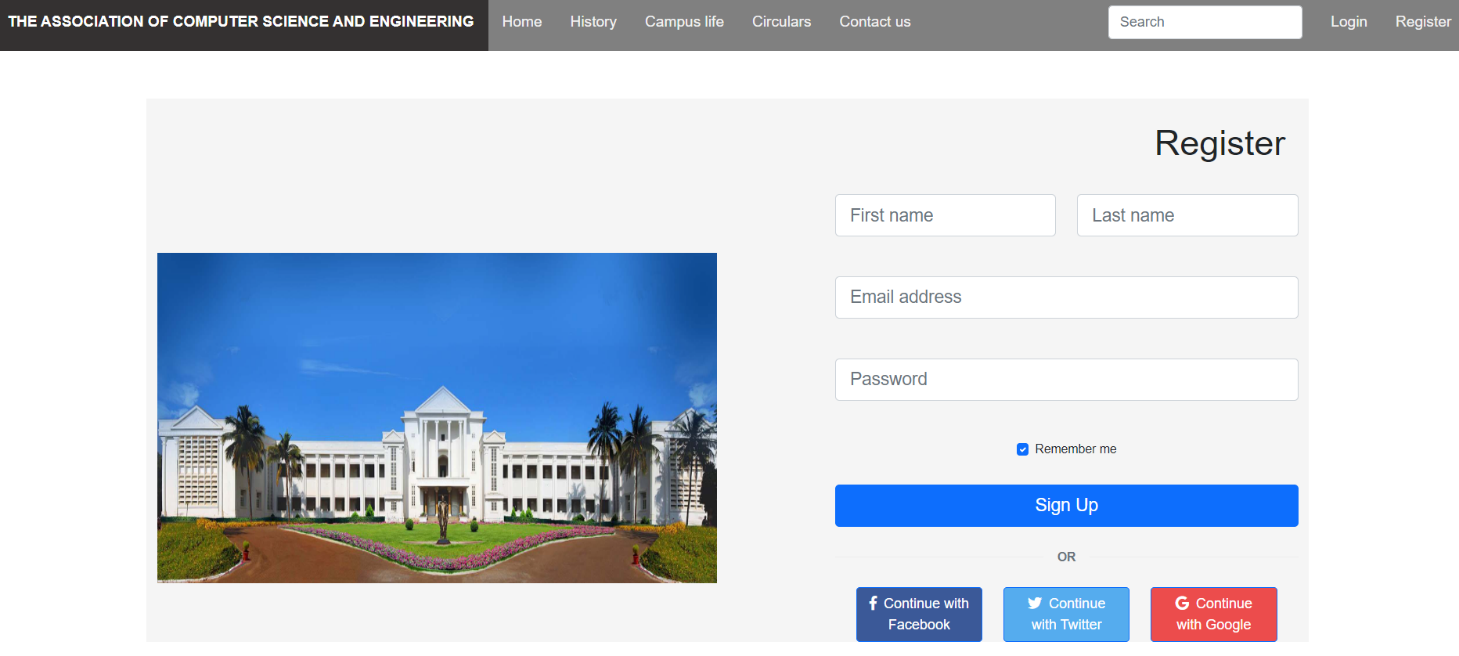
Model-View-Controller (MVC) as shown in Figure 3 is a software architectural pattern that separates an application into three main components: the model, the view, and the controller. The model represents the data and business logic of the application. It responds to requests for information and updates the data as needed. The view is the user interface of the application. It displays the data provided by the model and allows the user to interact with it. The controller receives user input and translates it into actions to be performed by the model and view. It mediates the interaction between the two and ensures that they remain independent of each other. By separating the application into these components, MVC makes it easier to develop, maintain, and test the individual parts of the application. This improves the overall design and organization of the code, leading to more modular and reusable code.

**4. RESULTS AND DISCUSSIONS**

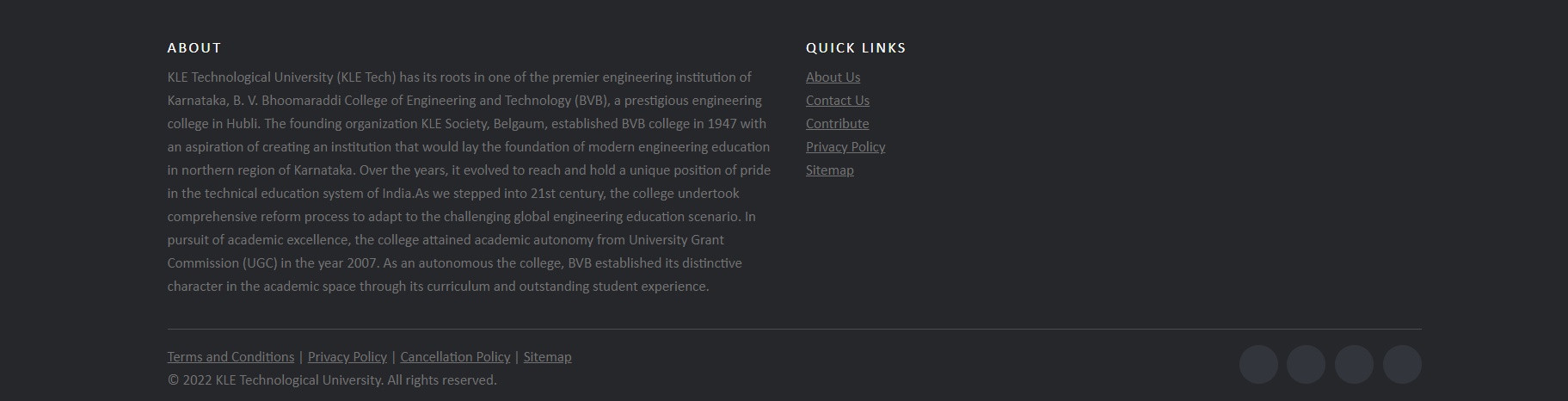
**4.1 LOGIN PAGE**



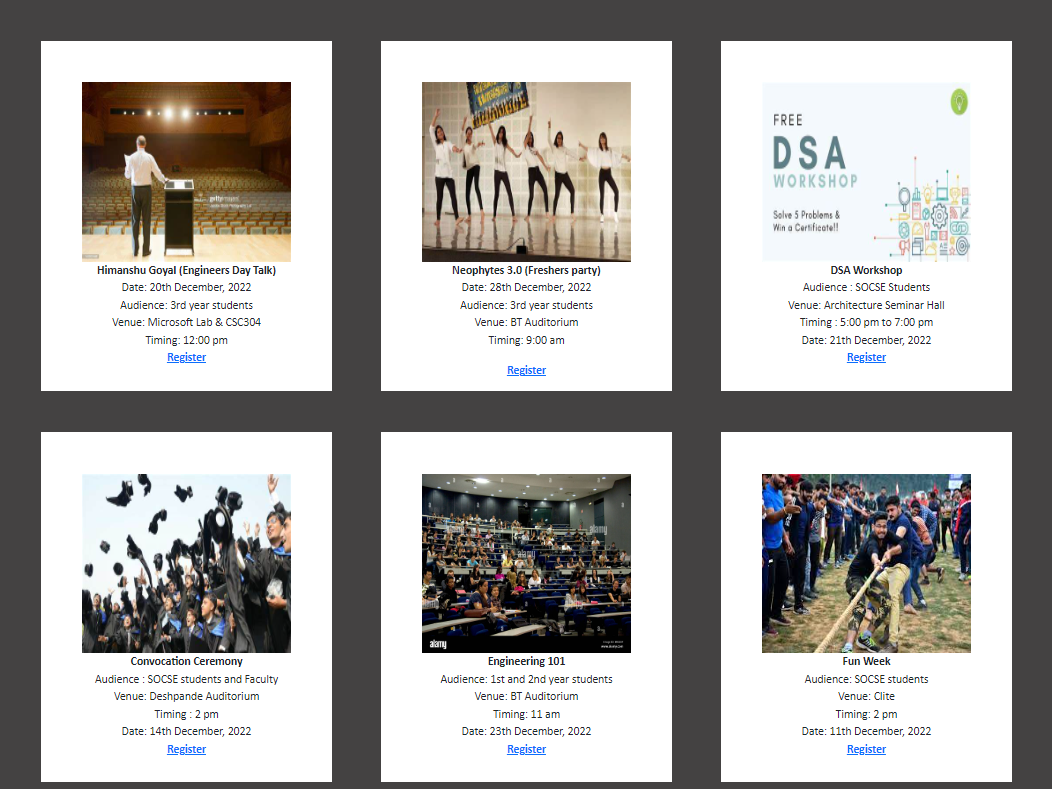
**4.2 REGISTER PAGE**



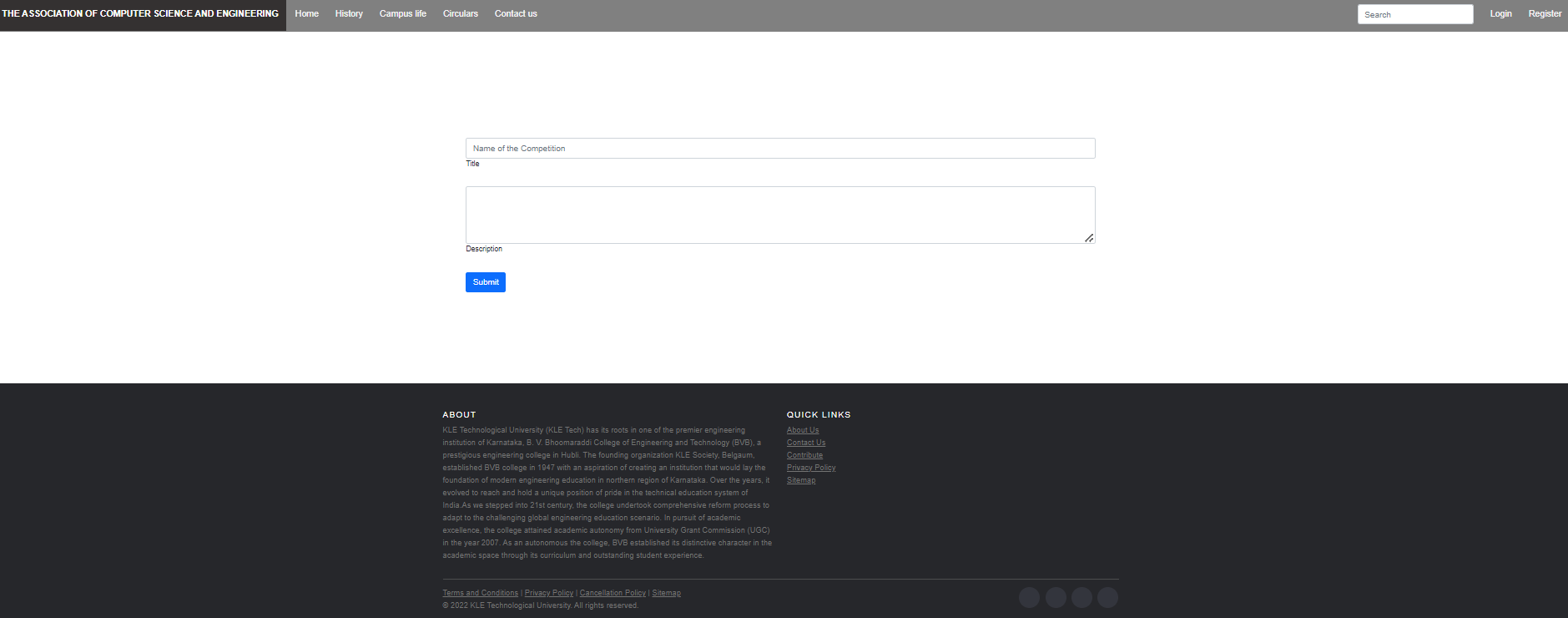
**4.3ABOUT PAGE**



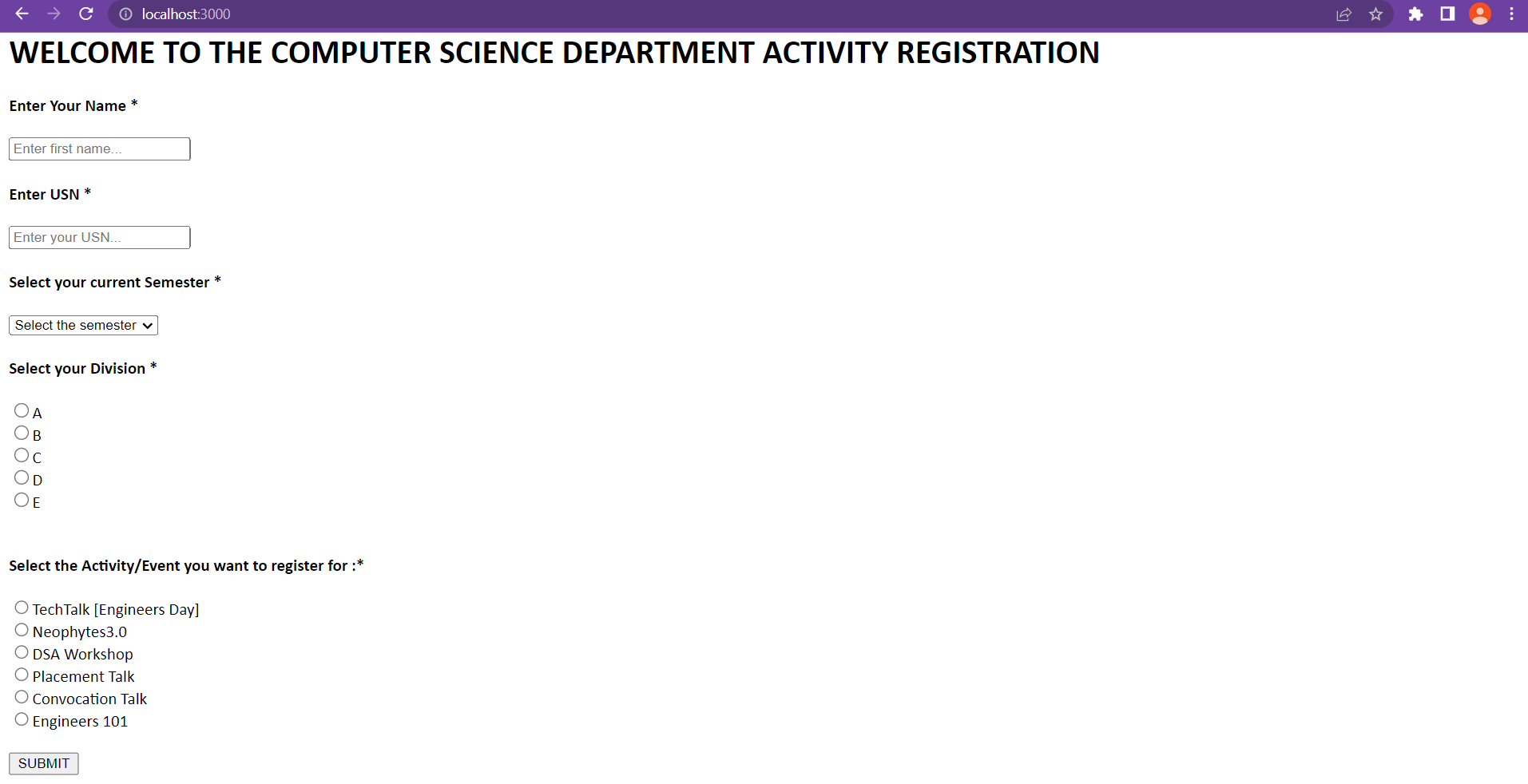
**4.4 EVENT LIST PAGE**



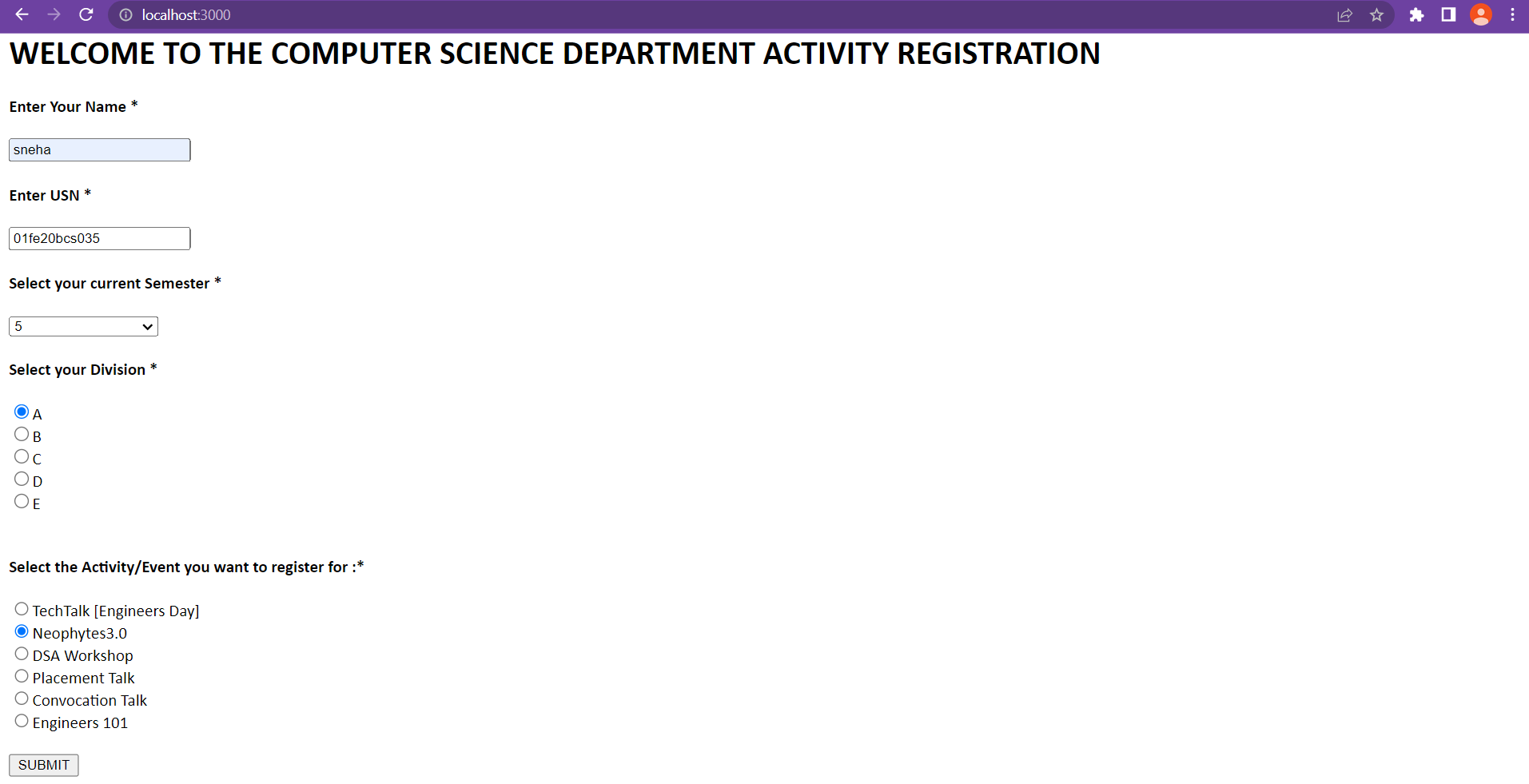
**4.5. ADDING EVENT**



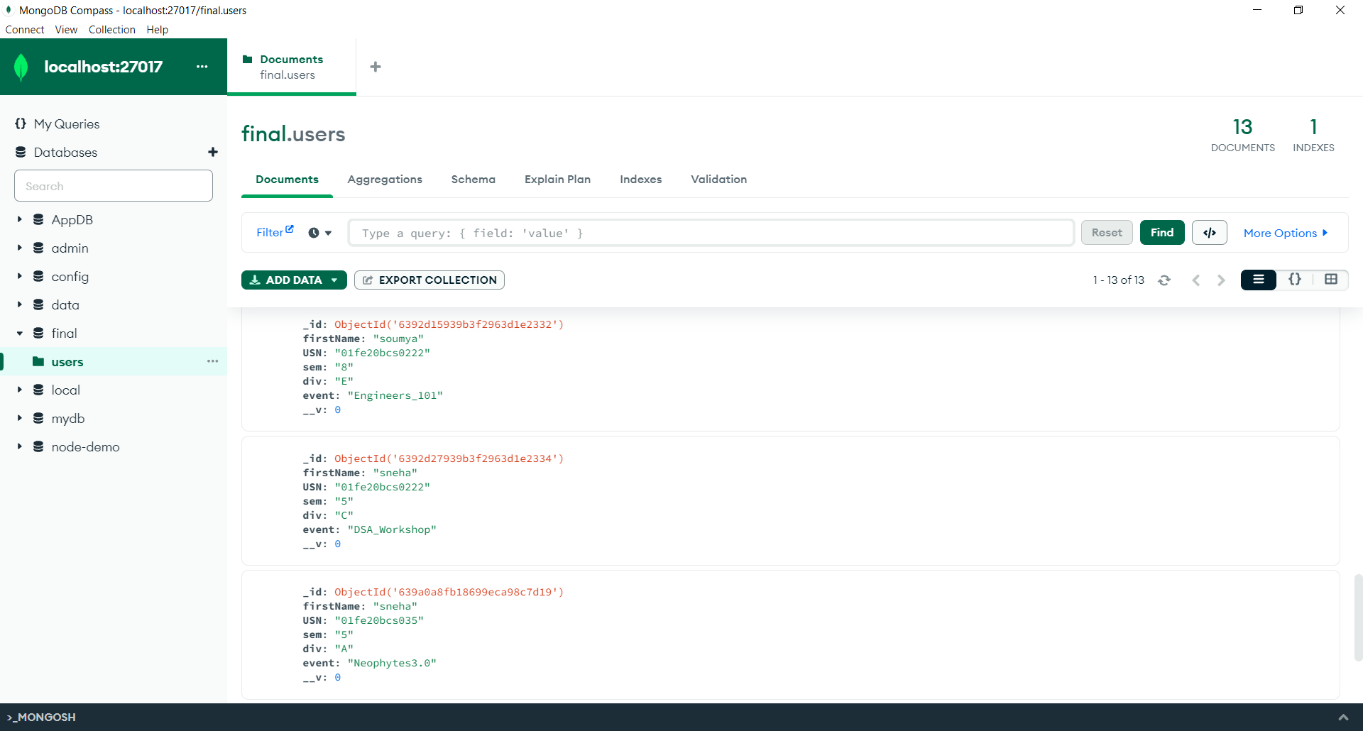
**4.6 Registration form for the event**



**4.7 Enter the student details**



**4.8 Storing the student details in MongoDB Database**



**4.3 Testing Tool**

**Git**

One of the most well-liked version control programmers is called Git. It is one of the systems for distributed version control. It is an open-source project that works with a variety of IDEs and operating systems.

It enables us to monitor changes made to an application, a folder, or a single file over time by various users on various computers. Every time we make a new commit with Git, all files are captured in a snapshot. Each member of a project's commit team always has a copy of the whole repository on their computer. So, even when offline, we may commit to the Git file system.

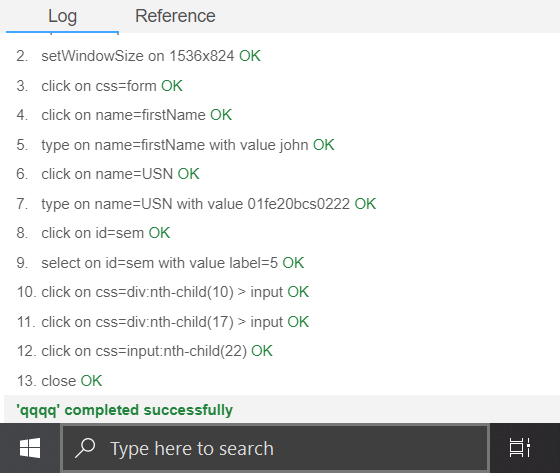
**Selenium Maven**

The most recent build testing tool is Maven. When compared to Ant, it has several new features, such as dependence, etc.

Maven is a tool for managing or building projects. When several test engineers merge their files into one framework, it is utilized to check for compilation problems between the components.

Every time a change is made to the framework, it offers build status modification and always maintains the monitor, framework components, or build.

It provides a 'build success' message if there are no compilation issues in the framework or else provides a 'build failure' message.



**Test: Registration for event**

**4.4 Continuous integration and continuous delivery (CI/CD)**

**Jenkins**

The leading open-source automation server, Jenkins provides hundreds of plugins to support building, deploying and automating any project.  It is used to implement CI/CD workflows, called pipelines. To operate Jenkins, pipelines are created. A pipeline is a series of steps the Jenkins server will take to perform the required tasks of the CI/CD process. Jenkins runs as a server on a variety of platforms including Windows, MacOS, Unix variants and especially, Linux. It requires a Java 8 VM and above and can be run on the Oracle JRE or OpenJDK. Usually, Jenkins runs as a Java servlet within a Jetty application server. It can be run on other Java application servers such as Apache Tomcat. Complex pipelines, especially, can be difficult to code, debug and maintain.

**5. CONCLUSION AND FUTURE SCOPE**

For the purpose of automating the management of events and activities, we have created a web application for the computer science students at KLE Technical University.

This is user-friendly and offers users the chance to see the events in real time and even sign up for upcoming activities. To add to or change the event list, administrators need their own credentials which are over-seen as well. This web application also maintains the login and registration information.

Further enhancement:

We can further extend our web application to maintain the past history as well, where user can see the previous programs and winners as well.

**APPENDIX:**

1. **Roles and Responsibility within team**

|  |  |  |
| --- | --- | --- |
| **Role (Responsibility)** | **Programmer 1** | **Programmer 2** |
| Analyst (requirement collection) | Soumya | Nehs |
| (HTML+CSS+Javascript) | Sneha | Neha |
| Backend designer (Database) | Srishti | Soumya |
| Developer (Implementation) | Neha | Sneha |
| Tester | Soumya | Srishti |

1. **Photos of meeting with the user**



