Implementation Document

for

ClubZen IITK

Version 2.0

Prepared by

Group 12 Group Name: Tiny Coders

|  |  |  |
| --- | --- | --- |
| Akash Biswas | 200074 | [abiswas20@iitk.ac.in](mailto:abiswas20@iitk.ac.in) |
| Avinash Prased | 200231 | [avinashp20@iitk.ac.in](mailto:avinashp20@iitk.ac.in) |
| Chandekar Vidish Vijay | 200291 | [cvvijay20@iitk.ac.in](mailto:cvvijay20@iitk.ac.in) |
| Harshit Gupta | 200429 | [guptah20@iitk.ac.in](mailto:guptah20@iitk.ac.in) |
| Jaya Meena | 200472 | [jayameena20@iitk.ac.in](mailto:jayameena20@iitk.ac.in) |
| Nikhil Verma | 200637 | [nikhilv20@iitk.ac.in](mailto:nikhilv20@iitk.ac.in) |
| Priyal Agrawal | 200730 | [priyalag20@iitk.ac.in](mailto:priyalag20@iitk.ac.in) |
| Priyanka Meena | 200731 | [priyankam20@iitk.ac.in](mailto:priyankam20@iitk.ac.in) |
| Shreyasi Mandal | 200956 | [shreyansi20@iitk.ac.in](mailto:shreyansi20@iitk.ac.in) |
| Vartika | 201089 | [vartikag20@iitk.ac.in](mailto:vartikag20@iitk.ac.in) |

|  |  |
| --- | --- |
| Course: | CS253 |
| Mentor TA: | Aishwarya Gupta |
| Date: | 15.02.2022 |

Content

Contents ii

Revisions ii

1 IMPLEMENTATION DETAILS 1

2 CODEBASE 3

3 COMPLETENESS 5

Appendix A - Group Log 6

Revisions

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| 1.0 | Tiny Coders | Implementation of the software | 20/03/22 |
| 2.0 | Tiny Coders | Updation | 28/03/22 |

# Implementation Details

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Back end** | **Benefits** | **Front end** | **Benefits** |
| **Programming language** | Java | Compared to C++, Java offers much more flexibility by providing a wide range of frameworks. | HTML | Every browser supports HTML and it is very easy to use. |
| CSS | CSS saves time as we can write CSS once and then reuse the same sheet in multiple HTML pages.It is also easy to maintain as to make a global change, simply change the style, and all elements in all the web pages will be updated automatically. |
| JS | JavaScript is very fast because it can be run immediately within the client-side browser. Unless outside resources are required, JavaScript is unhindered by network calls to a backend server. Also it is easy to implement. |
| **Frameworks** | Spring boot | It offers an effortless way to create spring-based applications using Java.It reduces the time and efforts of developers by reducing all the manual work of writing annotations,  boilerplate codes and XML configurations.  It also provides a lot of plugins which aid in development and testing. | Node.js | Applications written in Node.js require fewer files and less code compared to those with different languages for front-end and back-end. |
| **Libraries** | Lombok | Java library to reduce boilerplate code significantly and easily**.** | ReactJS | ReactJS allows developers to utilize individual components of an application on both client-side and the  server-side. It is also easier to update and manage due to its modular structure. |
| **Spring boot dependencies** | Spring Security | Configuration support for Java.   Comprehensive support for tasks like authentication and authorization.   Servlet API integration. | NA | |
| Spring Data MongoDB | As MongoDB is used as the database system, Spring boot offers this dependency to integrate the backend and database. |
| Spring Web | The single spring-boot-starter-web dependency transitively pulls in all dependencies related to web development. |
| **DBMS** | MongoDB | MongoDB is faster than MySQL due to its ability to handle large amounts of unstructured data when it comes to speed. | NA | |
| **Build System** | Maven | It simplifies the process of  project building as Maven can add all the dependencies required for the project automatically by reading the pom file. | Node.js and npm | NPM (a node package manager), Node’s package ecosystem, is the largest and the fastest growing software registry in the world. It provides numerous libraries and reusable templates. |

# Codebase

**GitHub Repository** - <https://github.com/Shreyasi2002/CS253_Project>

## About the Web App

As of now, the front-end and the backend are separate units.

For Back-end Part,

RESTful APIs are deployed on Heroku using GitHub and can be accessed anywhere worldwide.

For the Front-end Part,

The web app source code can be run using **npm** and hosts the main page, login page, clubs page, councils page, festivals page, and calendar page.

## Codebase Navigation

**FrontEnd**

The Front-end code is present in “/Interface Design/clubzen-iitk/”. The “src” folder contains all the React source code files. The “public” folder includes whatever is rendered at the web browser.

Graphical user interface

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**BackEnd**

The Back-end code is present in “/Backend Units”. The individual folders contain the source codes of the individual units. The “clubzen” unit is the integration of all the units into one single unit. The “src” folder in every unit folder contains the source code files individually. However, the “sample data” folder contains some sample data fetched from the MongoDB database to be used for demonstration purposes.

The src folders’ basic structure is divided into models, repositories, exceptions, controllers, listeners, configurations, etc. The below image makes it more clear.

Graphical user interface

Description automatically generated with low confidence

## Installation Requirements

1. Web Browser
2. Node.js
3. npm
4. Maven
5. Tomcat
6. Spring Framework
7. Spring Boot
8. JDK (Java Development Kit)
9. Any Java or Spring IDE (for e.g., IntelliJ, Eclipse, Spring Tools Suite)

## Installation

For Frontend, i.e., website,

1. Navigate to "/Interface Design/clubzen-iitk/".
2. Open Command Line Terminal in this folder.
3. Run the following commands one by one -
4. npm install
5. npm start

The webpage will automatically open in localhost (<http://localhost:3000/>).

If it doesn't automatically open, the user can open it manually.

For the Back-end part,

1. Navigate to "/Backend Units/{calendar or loginsignup or newsandevents}/.”
2. Open Command Line Terminal in this folder.
3. Run the following command one by one (assuming a Tomcat Local server has already been built)
4. mvn clean install
5. mvn spring-boot:run

Now the user can test the APIs on localhost (<http://localhost:8080/>).

## Test Run

The webpage can be tested using npm as stated in the "Installation" part.

The Backend Units can be tested using POSTMAN or any other API Client.

The links to access the different APIS of different units are -

1. loginsignup - [LoginSignup](https://cs253-signin-signup.herokuapp.com/)
2. calendar - [Calendar](https://cs253-calendar.herokuapp.com/)
3. newsandevents - [NewsAndEvents](https://cs253-news-and-events.herokuapp.com/)

# Completeness

|  |  |  |
| --- | --- | --- |
| **SRS features** | **Status** | **Future development plan** |
| Profile login/registration | Completed | Give IITK users login based access to different sections not available to guest users based on their roles. |
| Calendar/News | Completed | Give some publishing rights to managers of the Clubs, so that they can directly post news related to their Clubs. |
| News and Events | Completed | We plan on further optimizing the design of the news and events section. |
| PoR Holders’ directory | Completed | At present, modification of this data by an admin user is not allowed. We plan on providing this option based on roles. |
| Posting and Reviewing | In progress | We plan to give publishing rights to users, due to which they would be able to post in the respective Clubs section. Also, it would be necessary that the content goes under review first. |
| Forums | In progress | We are developing a forums section, wherein all the users can post their queries and also comment on other user’s queries. The Backend implementation for this has been completed. |
| Customized notifications | In progress | Through this functionality, a user can select some particular events or clubs of which the user will receive notifications. |
| Achievements and Badges | In progress | This functionality requires all the above functionalities in place to work, so this is in the last priority of the development. |

Appendix A - Group Log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl. no.** | **Date** | **Timings** | **Venue** | **Description** |
| 1 | 04/03/22 | 4 pm - 5 pm | RM building | Discussed how we will carry out the implementation part. |
| 2 | 08/03/2022 | 11 am- 12 pm | Google meet | Meeting with the TA. She discussed the frameworks and resources that we can use for implementation. |
| 3 | 10/03/2022 | 4 pm - 6 pm | Google meet | Planned the work distribution. We divided ourselves into two groups- the frontend and the backend. |
| 4 | 11/03/2022 | 6 pm - 8 pm | Google meet | Gathered the resources for learning various frameworks and languages required for the implementation part |
| 5 | 12/03/22- 14/03/22 | — | Online meetings | Worked among the groups separately on frontend and backend |
| 6 | 15/03/2022 | 5 pm - 7 pm | Google Meet | The entire group meet in which we discussed the progress |
| 7 | 16/03/2022-18/03/22 | — | Online meetings | Completed the remaining implementation work |
| 8 | 19/03/22 | 4 pm - 6 pm | Google Meet | Completed the implementation document |
| 9 | 20/03/2022 | 5 pm - 8 pm | RM building | Final review of the implementation document and other tasks |