INTRODUCTION

The major idea behind this application is to allow enterprises such as schools, colleges, vital day-to-day processes such as employee management, faculty management, meeting and schedule management, analytics and reporting. Most colleges keep track of their tasks, manually (books, files). This is a manual process and often takes a lot of time, it is difficult to maintain and is clearly a wastage of paper. Social media or other time management tools provide limited functionality and are not customizable. While there are tools that provides options to log details or activities, they are often difficult to use and are not user friendly. The application consists of two main components. One is the web application and the other being the mobile application. The web application is responsible for authentication, user role management and database management whereas the mobile application is responsible for users’ attendance or presence via scan-to-confirm strategy. It provides customizable options for creating users with role management and secure authentication. The schedule of activities are recorded by the administrator at the beginning of the session. The mobile app is responsible for recording the actual events occurred when the session is held. It records the time, date and location of the incident by a QR Code and sends the data securely to the web application which in turn stores and processes the information. Reports can be generated based on processed information and made available to different users based on their roles. Whenever there is a change in schedule, it will be notified to the faculty through the app.

The project is built keeping scalability in mind. The selection of tools and technologies for the project helps us in serving millions of users, including real-time data without any delays. The application is containerized into microservices where a load balancer is attached that handles a heavy amount of traffic. Also, continuous integration and continuous delivery techniques of deploying and testing the project has been implemented for better developer operations. The project is currently under the development phase where it is being tested for accuracy and reliability. It will help every enterprise in making their day to day processes easier and accountable. Further plan is to deploy the product in any enterprise organization which strive for quality.

ANALYSIS

Literature survey:

There are n number of applications which are both free and proprietary. The process of building this application had references to the following applications.

Replicon - is a configurable time and attendance platform enables your organization to manage dispersed workforce. Employees can provide inputs from the field using our advanced mobile app. Supervisors gain complete control over employee availability, time off and scheduling.

Jibble - Manage time & attendance for team. Employees can clock in, punch in or as we say, jibble in and out using the iPad Kiosk, Web or Mobile (iOS & Android). Work hours are accurately captured including activities and notes. Generate automated timesheets, activity/project tracking, client billing and powerful reporting for your team. Whether you are running a tech startup, consulting firm, or running the local restaurant, Jibble helps you with payroll, billing or team productivity

TrackerPal - Time and Attendance Software with Geofencing. Easily manage remote and field employee attendance. Employees punch in and out using their mobile phone. Optional selfie. You can specify a location for attendance. Supports leave request and approval.

TimeCamp - In its mobile version, TimeCamp helps users to track time automatically to specific projects, whether new or existing ones, and change the time entries manually. This attendance tracking software can also be set up on Android and iPhone and other mobile devices which run the iOS platform.

Many of these applications are built focusing employee management and not solely around attendance management. This applications focus is to make the attendance management and generate a report which can be used to improve the efficiency of the organization.

Functional requirements:

* Authorization: Only the authorized admin should be able to login to the application where he/she can use the application.
* Role management: Admin can have one more account as a user/faculty. He/she should be assigned proper role.
* Schedule management: Admin should be able to make changes in the existing time table if there is a need.
* Admin should be able to create departments, sessions, add faculty, and add subjects
* Report: Admin should be able to generate and analyze the weekly, monthly and semester wise report

Nonfunctional requirements:

* Application is containerized for traffic distribution and scalability
* Distributed version control and task management
* Continuous integration and continuous delivery for seamless deployment
* Works across devices or any screen via a browser
* Platform independence:  The application should run on both the popular operating systems Windows, Ubuntu etc.
* Performance: The application should be able to run on all versions of operating systems after a limit and should consume less power. The application should not crash at any condition.
* Permissions and Authentication: The application should ask for user’s permissions to access internet, send notifications etc. Anyone apart from the authenticated user should not be able to log in. If someone tries to penetrate the application, the tokens generated are verified using libraries like JWT.

Software requirements:

* Backend:
  + Node.js v10.x, Express.js v8.x
* Frontend:
  + React.js v16.x, Redux v16.x
* Database:
  + MongoDB v4.x
* Testing:
  + Mocha, Postman
* DevOps:
  + Github, Travis CI, AWS, Heroku
* Other Tools:
  + Draw.io, Balsamiq

Tools and Technologies:

Application Development technologies:

This application is built using MERN stack i.e. MongoDB, Express, React Native, NodeJS. Redux and Firebase are also used to build this application.

MongoDB is an open-source database software which is NoSQL in architecture. It stores data as JSON documents. It is fast, reliable and partition tolerant.

Express is a  web application framework for Node.js. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js

React native is a JavaScript framework for writing real, natively rendering mobile applications for iOS and Android.

Node.js is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code outside of a browser.

Redux is an open-source JavaScript library for managing application state. It is most commonly used with libraries such as React or Angular for building user interfaces. Similar to Facebook's Flux architecture

Firebase is a mobile and web application development platform

All these technologies are open-source and has wide community support. MERN stack is very popular, growing is usage and developer friendly.

**Integration tools:**

**Travis-CI:**

Travis CI is a hosted, distributed continuous integration service used to build and test software projects hosted at GitHub. Open source projects may be tested at no charge via travis-ci.org. Private projects may be tested at travis-ci.com on a fee basis.

**Git and GitHub :**

Git is a distributed version-control system for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files. Its goals include speed, data integrity, and support for distributed, non-linear workflows.

GitHub is a web-based hosting service for version control using Git. It is mostly used for computer code. It offers all of the distributed version control and source code management functionality of Git as well as adding its own features.

Since ‘Vibhaag mobile app’ is a companion of web app, it is developed on the ‘development branch’ and then integrated to master branch where it is tested.

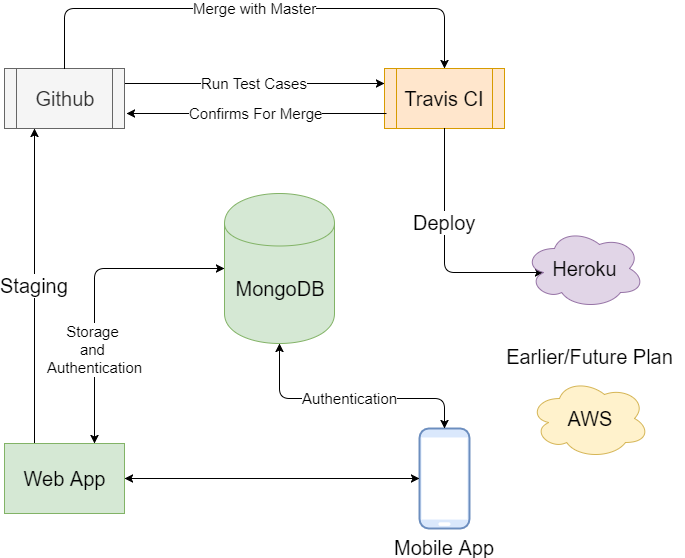
**Visual Studio Code :**

Visual Studio Code is a source-code editor developed by Microsoft for Windows, Linux and macOS. It includes support for debugging, embedded Git control, syntax highlighting, intelligent code completion, snippets, and code refactoring.

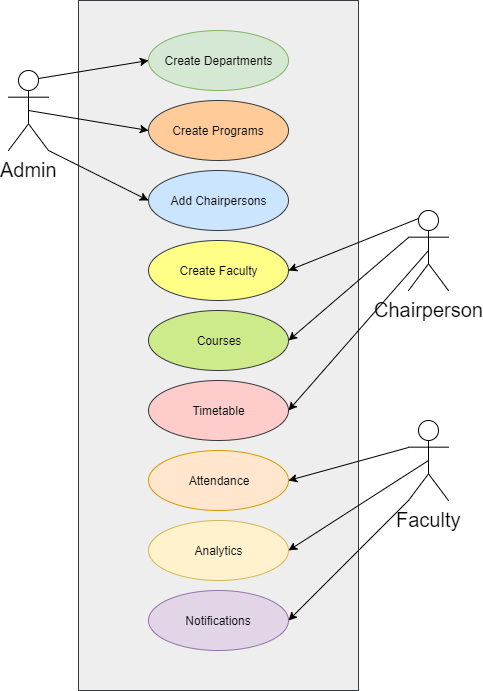
Source code is generated on VS Code and compiled in Android Studio. It is also tested on the android and device while developing

DESIGN

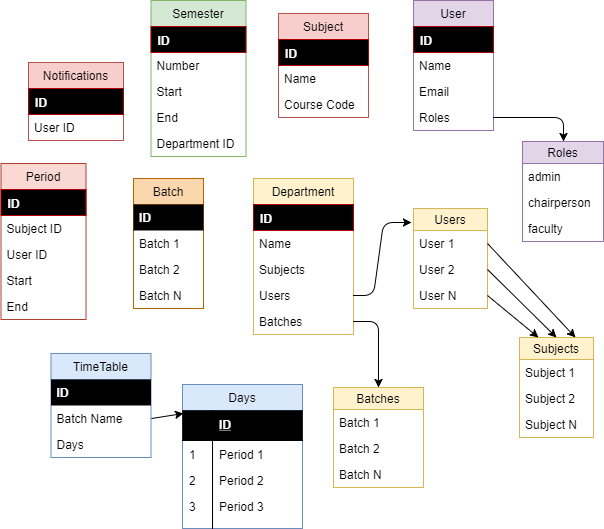
Data flow:



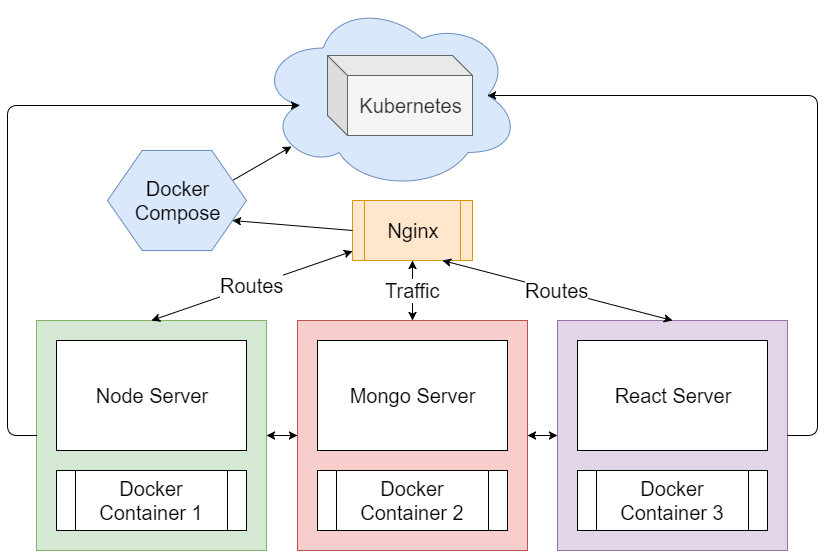
Use cases:



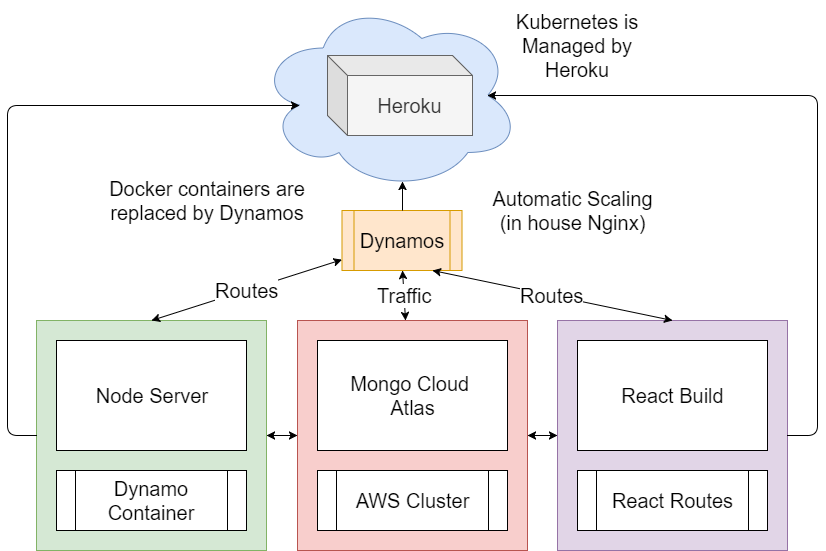
Database structure:



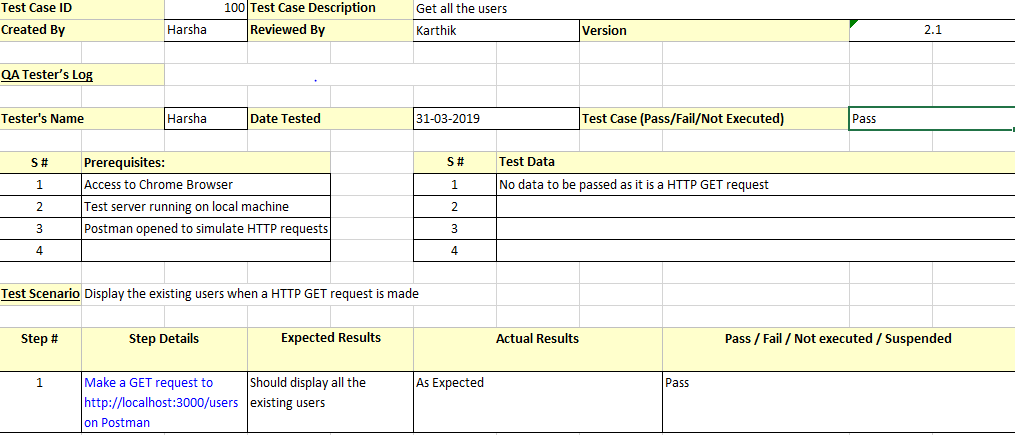
Architecture:

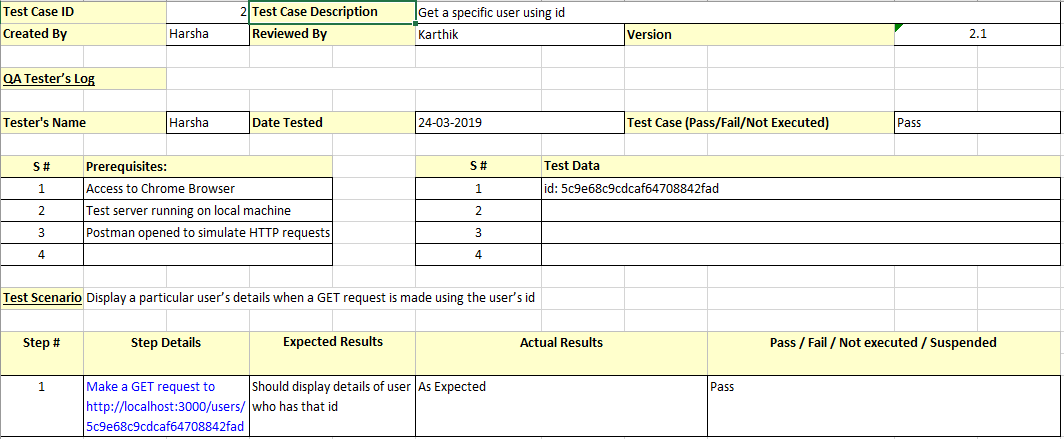


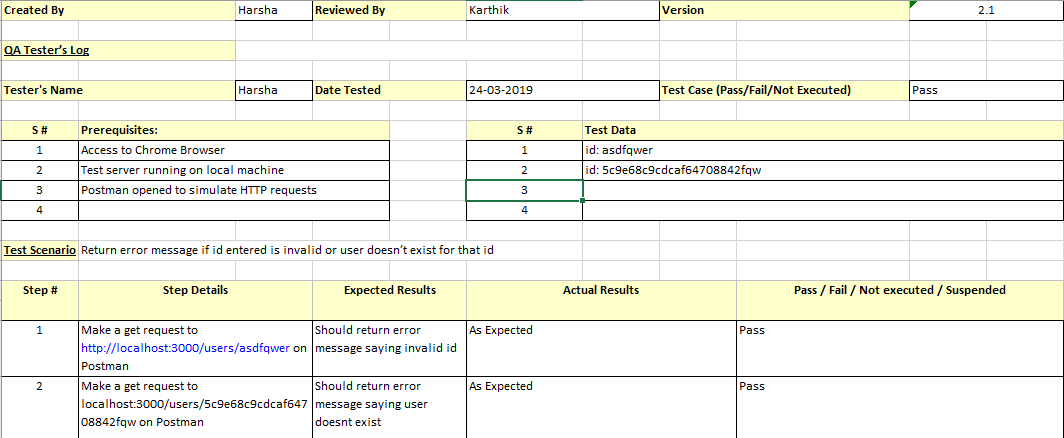
Implemented architecture:

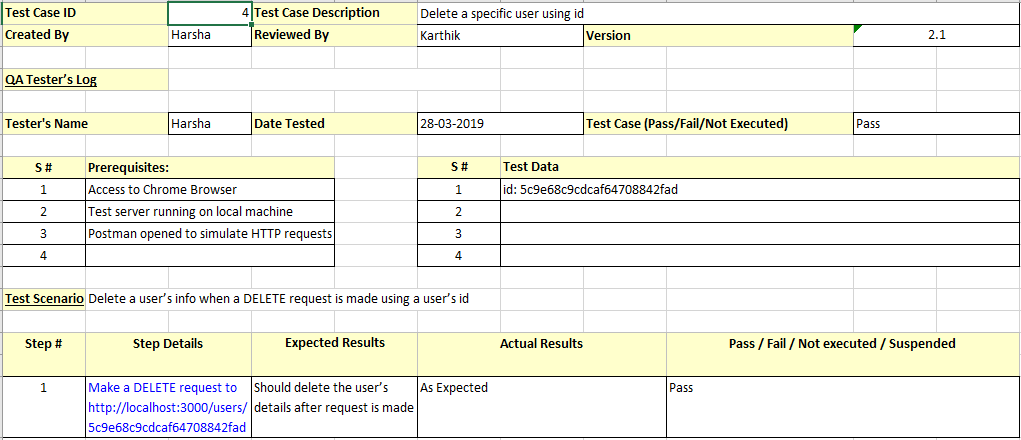


TESTING



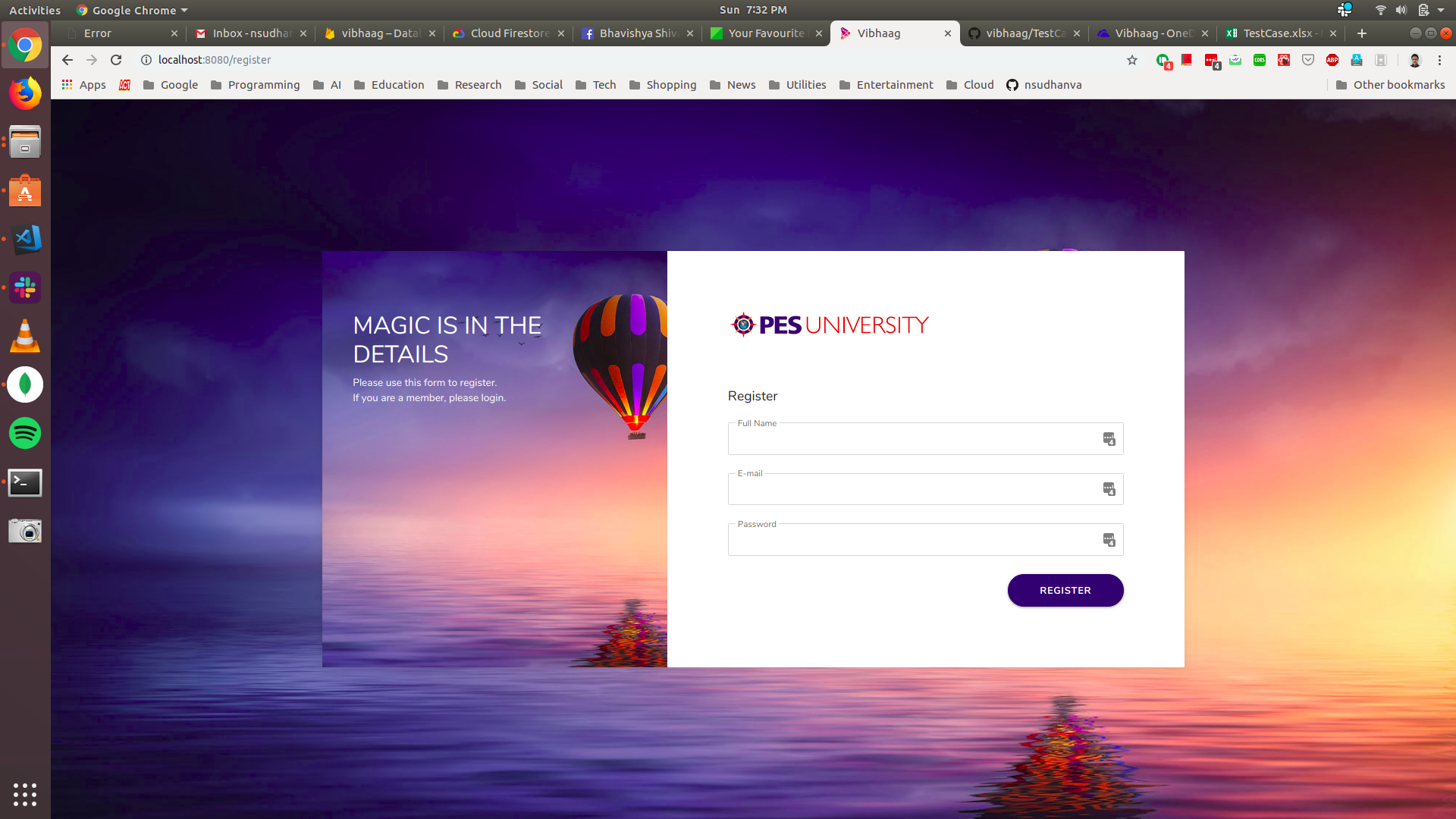




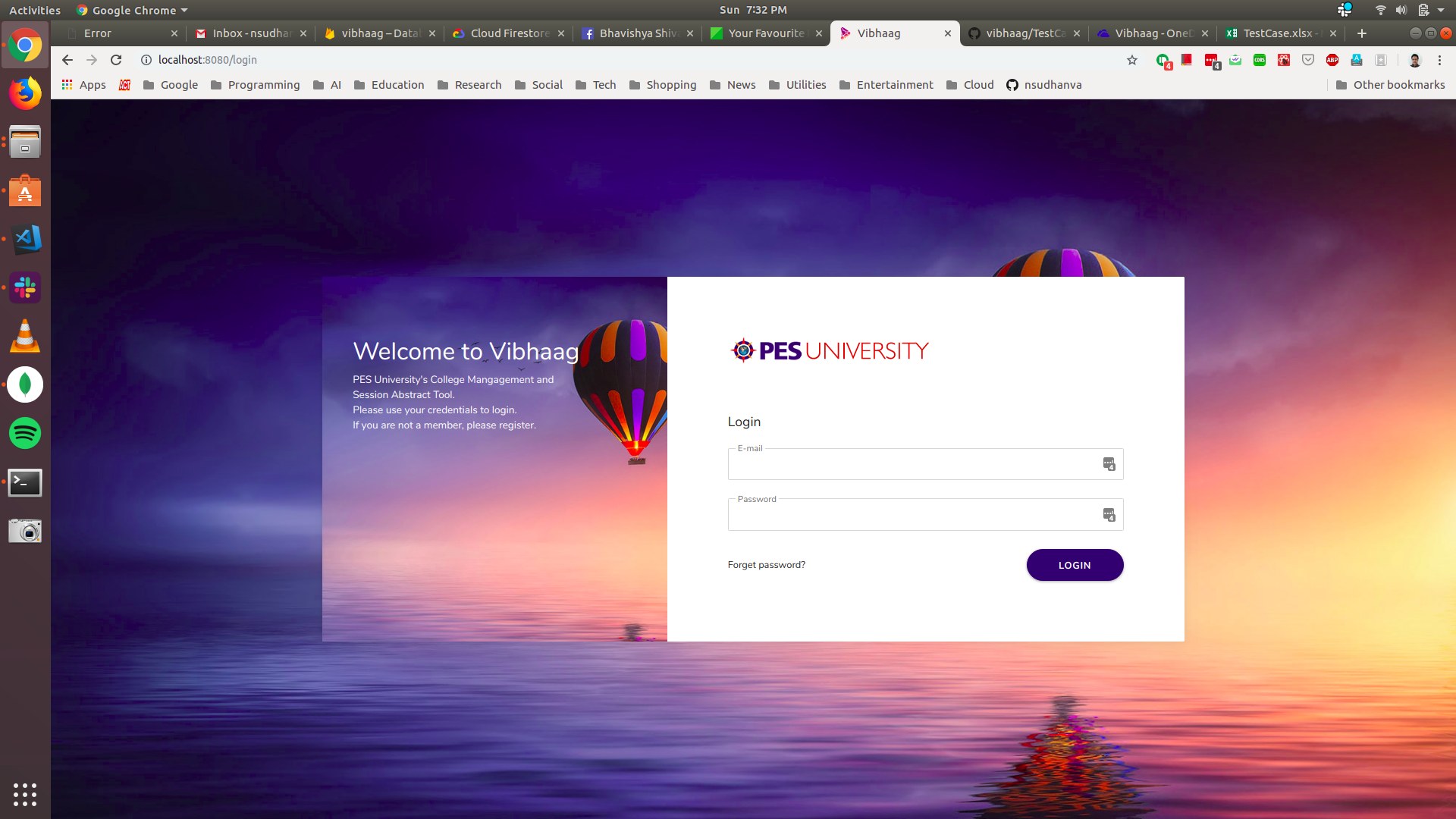


SCREENSHOTS

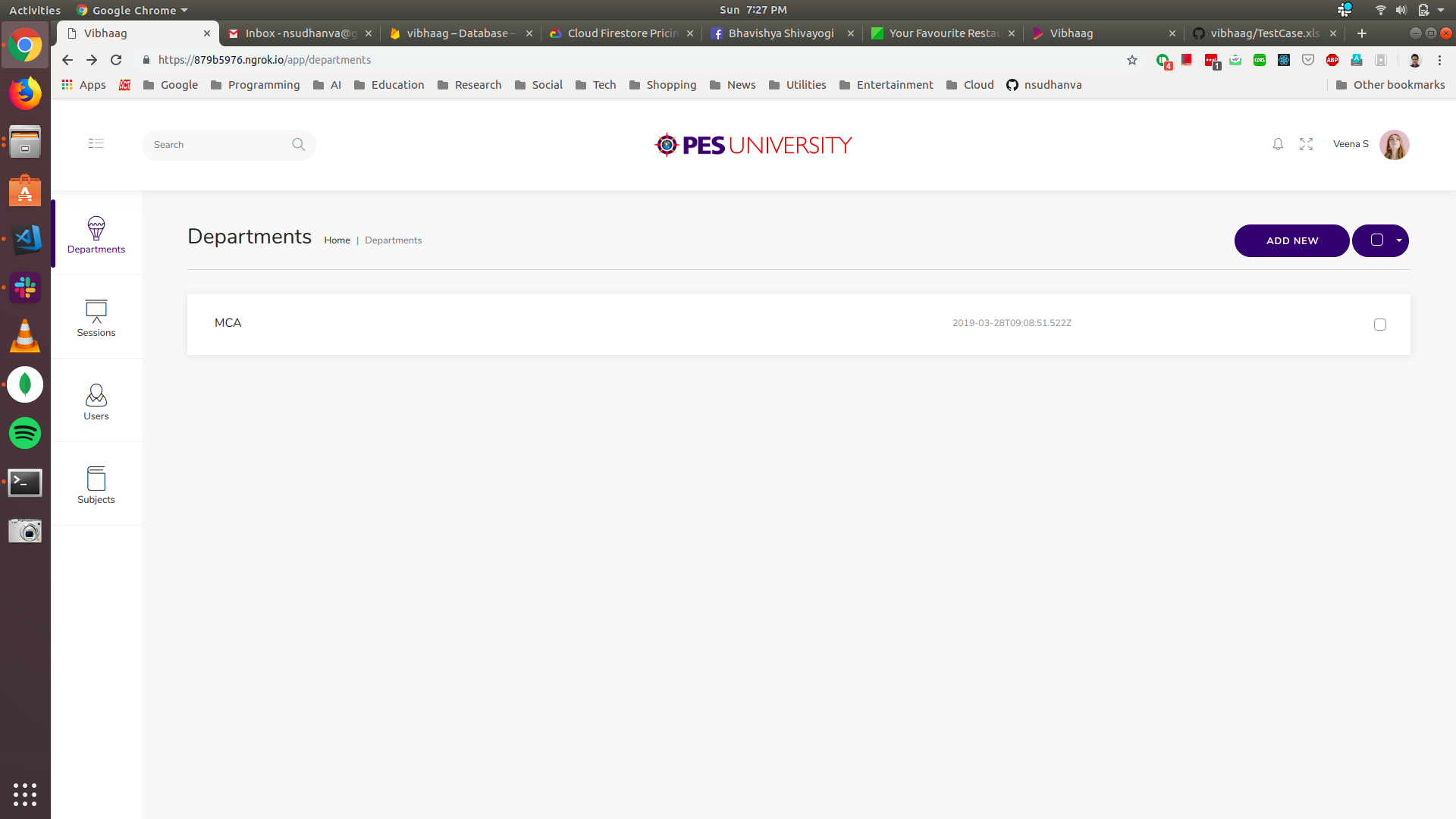
This is the screenshot of UI of user Registration



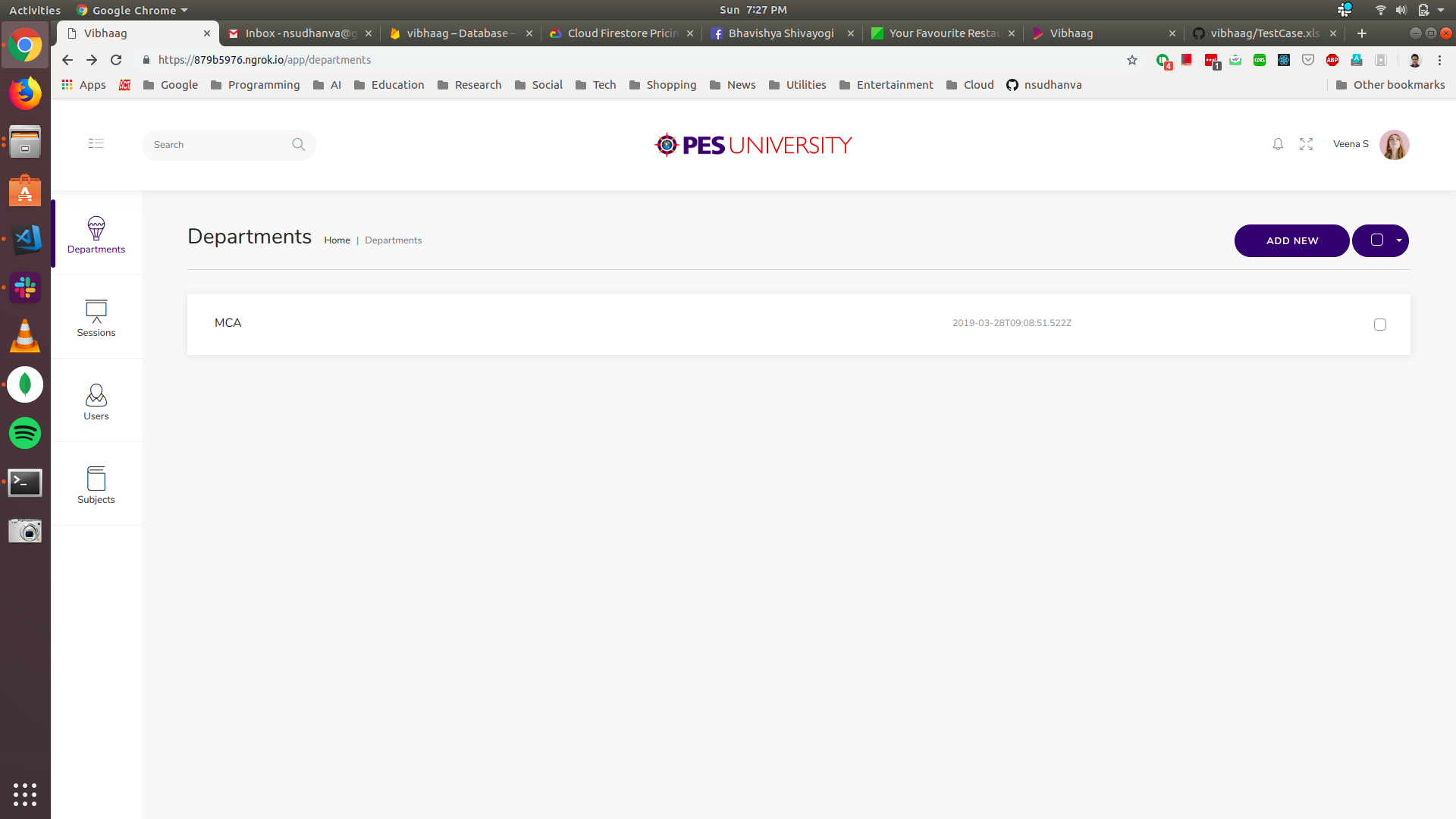
This is the screenshot of UI of user Login



This is the screenshot of UI of Home page



This is the screenshot of UI of department page



CONCLUSION

BIBLIOGRAPHY

1. ReactJS documentation: <https://reactjs.org/>
2. Redux documentation: <https://redux.js.org/>
3. NodeJS documentation: <https://nodejs.org/en/>
4. Firebase documentation: <https://firebase.google.com>
5. Medium and freecodecamp blogs: 1
   1. <https://medium.freecodecamp.org/understanding-redux-the-worlds-easiest-guide-to-beginning-redux-c695f45546f6>
   2. <https://medium.freecodecamp.org/a-beginners-guide-to-redux-9f652cbdc519>
   3. <https://medium.freecodecamp.org/these-are-the-concepts-you-should-know-in-react-js-after-you-learn-the-basics-ee1d2f4b8030>
   4. <https://medium.freecodecamp.org/what-exactly-is-node-js-ae36e97449f5>