A Project ETE Report

on

STUDENT GUIDE PORTAL

Submitted in partial fulfillment of
the requirement for the award of the
degree of

Bachelor in Technology-Computer Science and Engineering

GALGOTIAS UNIVERSITY
(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Name of Supervisor: Dr. Kavita
Designation: Associate Professor

Submitted By

Tanya Uppal- 18SCSE1130001

Saumitya Srivastava- 18SCSE1070009

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA
December, 2021

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the project, entitled **"STUDENT GUIDE PORTAL"** in partial fulfillment of the requirements for the award of the Batchelor in Technology submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of September, 2021 to December 2021, under the supervision of Dr. Kavita, Associate Professor, Department of Computer Science and Engineering, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the project has not been submitted by me for the award of any other degree of this or any other places.

Tanya Uppal, 18SCSE1130001

Saumitya Srivastava, 18SCSE1070009

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Dr. Kavita Saini

Associate Professor, SCSE, GU

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of Tanya Uppal:18SCSE1130001; Saumitya Srivastava: 18SCSE1070009, has been held on __________________ and his/her work is recommended for the award of Bachelor of Technology in Computer Science and Engineering:

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: December, 2021

Place: Greater Noida

Acknowledgement

Abstract

Today advancement in technology is changing and challenging everyone’s life and making it easier for humans to fulfil their needs, and to expect a great evolution in expansion for web development. Now when one hears about online systems, one seems to be well known to this term, whereas not long time ago, everyone was quite hesitant towards it and considered it a taboo and unnecessary science revolution. But with this constant mobile and web application development, one need to be updated and follow all the trends to become a professional as well as a developer.

The most basic and first step for development is to decide the right frontend framework. There is a wide variety of options emerging every year as solutions for the problems developers face every day with browser-based applications taking over the world. We tried to come up with a proposed solution for the recognition of the best framework for the developer and also tried to use the framework best suited for our project to create a website for project evaluation of major and minor project in colleges using web development.

Node.js gave rise to the Full Stack Developers who are now able to manage server and client side by their own. Node.js is fast and reliable for heavy files and heavy network load applications due to its event driven, non-blocking, and asynchronous approaches, where developers can also maintain a complete projects in single pages (SPA) and can use for IOT. The result of study concludes from a survey and from literature review the implementation areas and challenges of the Node.js. Lastly will provide suggestion on how to improve to overcome the challenges.

Responsive design allows software developers to build a Web page that can dynamically adapt to the size of the devices. This development philosophy enables the rendering of Web pages in a fast and optimized way, ensuring a good user experience on mobile devices, tablet and desktop. In the scope of this study, we intend to explore the main advantages and limitations associated with responsive Web design. We adopted a quantitative approach based on the industry that allowed us to identify the reasons that lead software developers to the adoption of the responsive design and also address the limitations felt by them. The results obtained indicate that offering a good user experience and increasing accessibility stands out as being

the most important advantages. On the other hand, the main limitations include the compatibility with older Web browsers, the higher loading time and the difficulties in optimizing user experience. Finally, it was found that the perception of the advantages and limitations of responsive design is distinct for professionals with more professional experience in the field and for freelancer developers.

For the development for our website, we used the NodeJS and ReactJS framework and library respectively. The website is using MongoDB for the database configuration with different schemas for different logins.

The rapid emergence of multimedia based (including Web-based) information systems in recent years presents a serious challenge for the development skills of information systems professionals. Until recently, many such systems were simple, stand-alone applications like electronic brochures. Some have now become extensive, complex, e-commerce applications. As Web-based systems integrate with organizational activities such as customer support, sales and marketing, and distribution and technical support, Web system developers will encounter similar development issues as conventional and multimedia systems development. Therefore, we propose a more sophisticated and disciplined approach toward systems development, which includes resolving various problems that accompany large scale systems development. Recently, structured methods for large-scale systems development projects and visually oriented or object-oriented methods for interface design have dominated the systems development field.

As a result, we created a website to handle tasks as defined by teachers to the students through the portal accessed by the admin. This can help students to reach out their guide/mentors and ask their guidance as soon as possible.

In future, we can improve the user interface by taking recommendations from students as well as faculty. We can also extend the website by providing it to other departments as well as colleges. Different algorithms could be used to handle the large datasets and improve the functioning of the portal.

Table of Contents

List of Figures

Acronyms

JS	JavaScript
DOM	Document Object Model
GUI	Graphical User Interface
IoT	Internet Of Things
MVC	Model View Controller
CLI	Command-Line Interface
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
ASP	Active Server Pages
AJAX	Asynchronous JavaScript and XML
PHP	Hypertext Preprocessor
URL	Uniform Resource Locator
API	Application Program Interface
CBA	Component Based Architecture

CHAPTER 1: INTRODUCTION

Web Development refers to the building of an application from scratch that works with internet. It is mainly of two types- Frontend and Backend Development. Frontend is the component with which has direct contact with client. It includes HTML (Hyper Text Markup Language), CSS (Cascading Style Sheets) and JavaScript. Backend is the not interactive part for user which works only with the server side of any site. It is mainly used to store and organize data.

JavaScript (JS) is a computer technology used by people, who build the components of a website that we as a user see on our screens when we access any browser or web application, to carry through and run active content. This content consists of graphics, mutual forms, pictures, etc. It also includes opened pull requests which denotes the quantity of code being put. It started with working along with HTML for writing pages of web and authenticate at user’s side. Progressing with AJAX, with gives the developer an option to update a content of the web page without uploading the web page as a whole from the server, making the GUI and response time better. But still there was a need to use other scripting languages like Python, PHP and other technologies like JSP, ASP. Then came frameworks like NodeJS and ReactJS, these gave developer the privilege to create a fully working application with just using JavaScript. With all these pros, one will be eager to know which framework or library to choose accordingly.

JavaScript library is a collection or group of already written code fragment that is being reused by other developers to execute common functions of JavaScript. This can be filled into the remaining of one’s code on an initial basis. Most commonly known JavaScript libraries are ReactJS and jQuery. Taking common example of libraries, there are number of mathematics libraries that let the developer call the functions without knowing how the algorithm works. It works on limited scope and does not have any fixed constraints, that’s why their APIs requires few numbers of dependencies.

JavaScript framework is a frontend tool that helps in shaping and ordering a web application where the development defines its features on its own. Library is to produce the functions to help but framework also provides the structure to work on. Some well-known frameworks are Angular, NodeJS.

FORMULATION OF PROBLEM

- As we mentioned we want to improve the web development approach by analyzing the various frameworks and libraries.
- We want to follow the very basic comparison to do so.
- We will first explore the different frameworks present in the market.
- We will then study about NodeJS so that we can use it to make the application.
- Now we will engage in feature engineering which is followed by evaluating data.
- Now the visualization part kicks in where we make best UI/UX using suitable libraries.
- We will now design different web pages and try to link them using import.
- Then we will deploy our own web application which will in turn be efficient.
- We will combine all this into a web application for better understanding and making user interaction descent.

TOOLS AND TECHNOLOGIES USED

- HTML

  The HyperText Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

  Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

- Node.js

  NodeJS is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server-side and client-side scripts.

- Visual Studio Code

  It is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

- React

  It is a free and open-source front-end JavaScript library for building user interfaces based on UI components. It is maintained by Meta (formerly Facebook) and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications. However, React is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.

- CSS

  Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

CHAPTER 2: LITERATURE SURVEY

The Internet is an essential tool for citizens, businesses and public entities. Since its appearance, a number of significant changes have arisen, from the most basic paradigm of Web 1.0 to Web 4.0. The first phase of the Web, known as Web 1.0, is characterized by static content publishing, in which users had the power to only consume the content placed by companies. At that time, there was no bidirectional communication between the client and a company, and email emerged as the primary form of digital contact between these two entities. Web 1.0 is characterized by its low interactivity. Companies used the Internet as an alternative channel to publish content and news, without worrying much about user's opinions, who behaved like a passive viewer of information.

Website design is a crucial point in a website development process. It involves the arrangement of content into graphical models that can be used as a basis for coding a site. The increasing number of Internet users and mobile devices, such as smartphones and tablets, has caused a need to adapt the content displayed on each device. The idea of designing multiple versions of the same site to attend each size of screen and resolution would not be possible, because it would originate impractical costs and a high maintenance effort.

Web based applications are increasing its popularity as they become easier to develop, maintain and secure. Also, they are easily reachable to the clients and does not require additional installations in most cases and are quickly customizable. Web application is derived from web-based system, which have additional functionality to execute business logic of an organization. These applications are totally web based instead of requiring to install a separate application on the operating system. Google Docs, Web based retail stores, Google Maps, and the web-based email applications are kind of Web applications.

The Web development industry will find two kinds of developers. i.e. Front-end developers and Back-end developers. Front-end developers require to have knowledge of HTML, CSS, and a programming language to add effects and more to the front – end i.e. JavaScript. They build the web sites display and effects which are shown to the clients by converting the designer's design. Back-end Developers build the business logic behind any web application. The actions for instance adding and retrieving news highlights to and from a web applications, or sending email from a web based forms, or authenticating a visitors or clients credentials are all part of back-end developers. A back-end developers need to know languages like PHP, .NET, Java, and others. Back-end developers should also have knowledge of databases like My SQL, Oracle, and SQL Server, or should hire or entrust a database administrator to work with the flow. A database administrator will take care of database server and ensure its smooth performance.

A review of currently available frameworks was necessary to get a clear idea of where to use and which framework to be used. Most research into these has focused on usage of frameworks, instead of providing a survey of all well-known frameworks to have a good sense of comparison. We tried to give a better sense of pros and cons with comparative study to help the developers to choose and proceed in their work without any delay.

1) NodeJS is more of an open-source, runtime environment rather than a framework that runs on the V8 engine developed by Ryan Dahl in year 2009. The main reason behind its efficiency across distributed system is that it uses event-based input/output model for development of server-side and grid applications. No buffering could be ever encountered in this framework as it uses a single threaded code which further can provide to a large number of requests, that is, stack is used for requests that are incoming, without holding back for responses making it a great in concurrent management of threads.

   Node.js is built from ground for the purpose of handling asynchronous I/O as it is built of JavaScript and JavaScript is built as event loop. Like the on click event for a button in client side JavaScript is and event loop. While other environments do have this feature, they have it with using third party libraries or are not built from ground for the same purpose like the Node.js and hence they are often slow, or lags and does not belongs as a standard feature to them. Some of the examples are Event Machine – built for Ruby, Twisted – licensed under open source MIT License, it is introduced for Python and is available since Python 2 onwards, and network framework library for Apache named as Apache MINA which is also called "Networking Socket Library" and is another example of providing event-driven and asynchronous limited to APIs only. Similarly Apache AsyncWeb is built using Apache MINA and Perl's Any Event. Similarly an edge of Node.js over others will be that it will be capable of handling multiple request while it will act like a client towards the third party services by executing only a single thread. Other languages in this regard will block the processing until the remote server responds first for their initial request as a result they will be requiring multiple threading for executions. Comparatively in Node, all what you will use is asynchronous as it will become quite hard if you are to write non-asynchronous code in it. Also Node.js will never force to buffer data before outputting while the others like Event Machine, forces buffering in many cases to buffer the data.

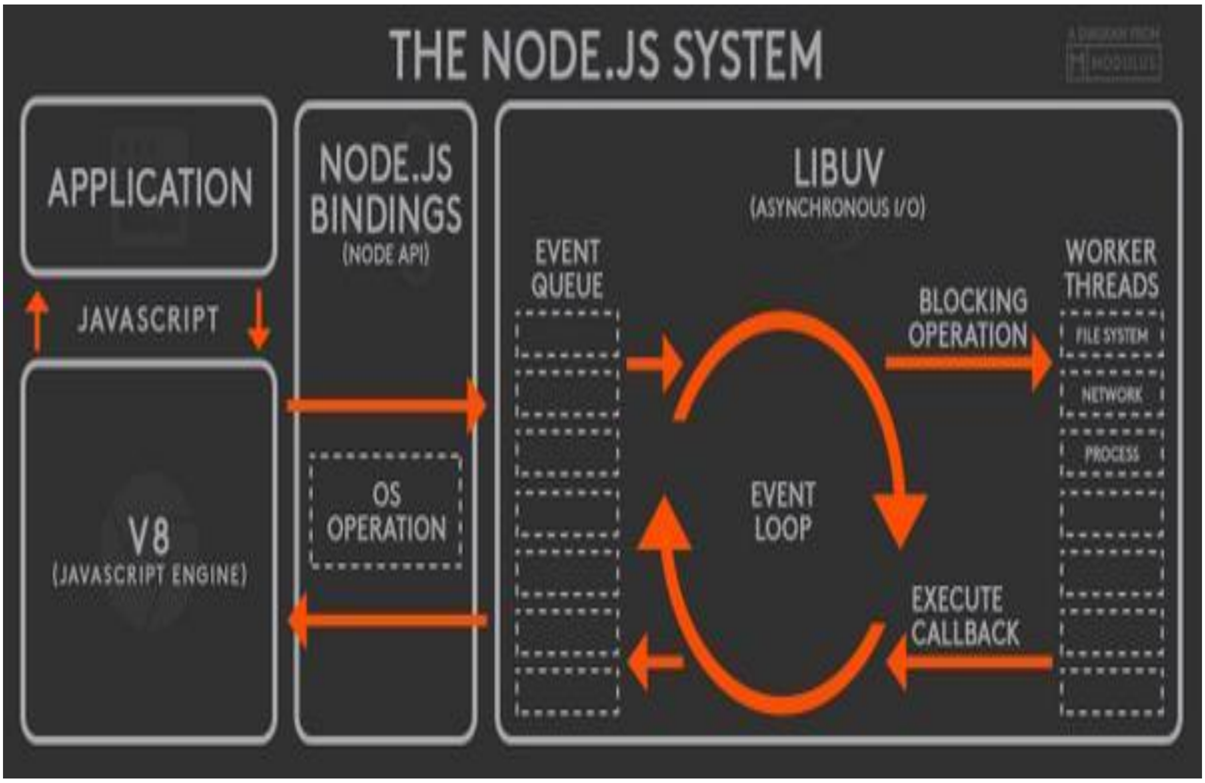


Figure 1: Architectural Diagram of NodeJS

2) Vue.js is an open-source model-view-view model front end JavaScript progressive framework for building UI, developed by Evan You. View Model comprises of DOM listeners and binding of data. It mainly works on declarative offering, directives which binds the HTML value to the data of development and content formation with its library only related to view layer. It offers an interface to make changes on the displayed content of the webpage according to current URL path.
Vue.js is a progressive framework for JavaScript used to build web interfaces and one-page applications. Not just for web interfaces, Vue.js is also used both for desktop and mobile app development with Electron framework. The HTML extension and the JS base quickly made Vue a favored front-end tool, evidenced by adoption by such giants as Adobe, Behance, Alibaba, Gitlab, and Xiaomi. In an interview with developers, he

reveals that initially Vue.js was an attempt to take the best of Angular and build a custom tool, but a lighter weight one: “For me, Angular offered something cool which is data binding and a data driven way of dealing with a DOM, so you don’t have to touch the DOM yourself.”

The name of the framework – Vue – is the same phonetically in English as *view*, and it corresponds to the traditional Model-View-Controller (MVC) architecture. Simply put, view is a UI of an application/website, and the core library of Vue.js focuses the view layer by default. But, MVC doesn’t mean that Vue.js can’t be used with a different architectural approach like the Component Based Architecture (CBA) used in React.

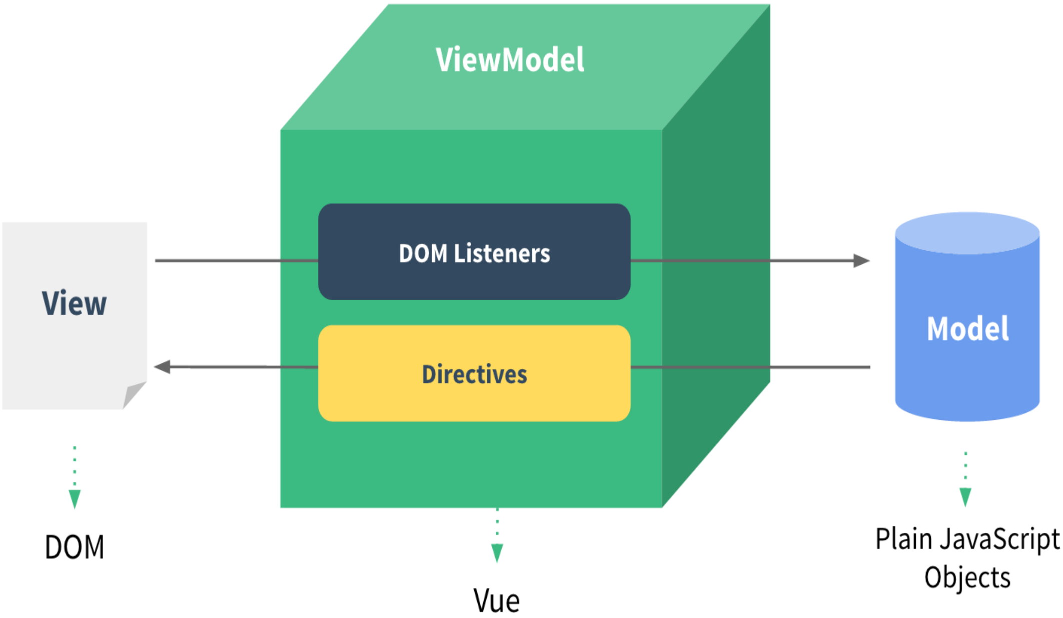


Figure 2: Architectural Diagram of VueJS

3) AngularJS is a framework that is used in an HTML page with a script tag. It is also an open-source front-end framework, maintained by Google mainly used to help both the testing and creating of web applications. This framework is an extension of traditional HTML that allows self-regulating synchronization to show current active content by two-way binding of data. Following the limitations of regular Angular, the usage of it became very low, which paved the way to the Angular 2 which is based on TypeScript instead of JavaScript.

The AngularJS framework works by first reading the Hypertext Markup Language (HTML) page, which has additional custom HTML attributes embedded into it. Angular interprets those attributes as directives to bind input or output parts of the page to a model that is represented by standard JavaScript variables. The values of those JavaScript variables can be manually set within the code, or retrieved from static or dynamic JSON resources.

AngularJS is built on the belief that declarative programming should be used to create user interfaces and connect software components, while imperative programming is better suited to defining an application's business logic. The framework adapts and extends traditional HTML to present dynamic content through two-way data-binding that allows for the automatic synchronization of models and views. As a result, AngularJS de-emphasizes explicit Document Object Model (DOM) manipulation with the goal of improving testability and performance.

AngularJS's design goals include:

- to decouple DOM manipulation from application logic. The difficulty of this is dramatically affected by the way the code is structured.
- to decouple the client side of an application from the server-side. This allows development work to progress in parallel and allows for reuse of both sides.
- to provide structure for the journey of building an application: from designing the UI, through writing the business logic, to testing.

AngularJS implements the MVC pattern to separate presentation, data, and logic components. Using dependency injection, Angular brings traditionally server-side services, such as view-dependent controllers, to client-side web applications. Consequently, much of the burden on the server can be reduced.

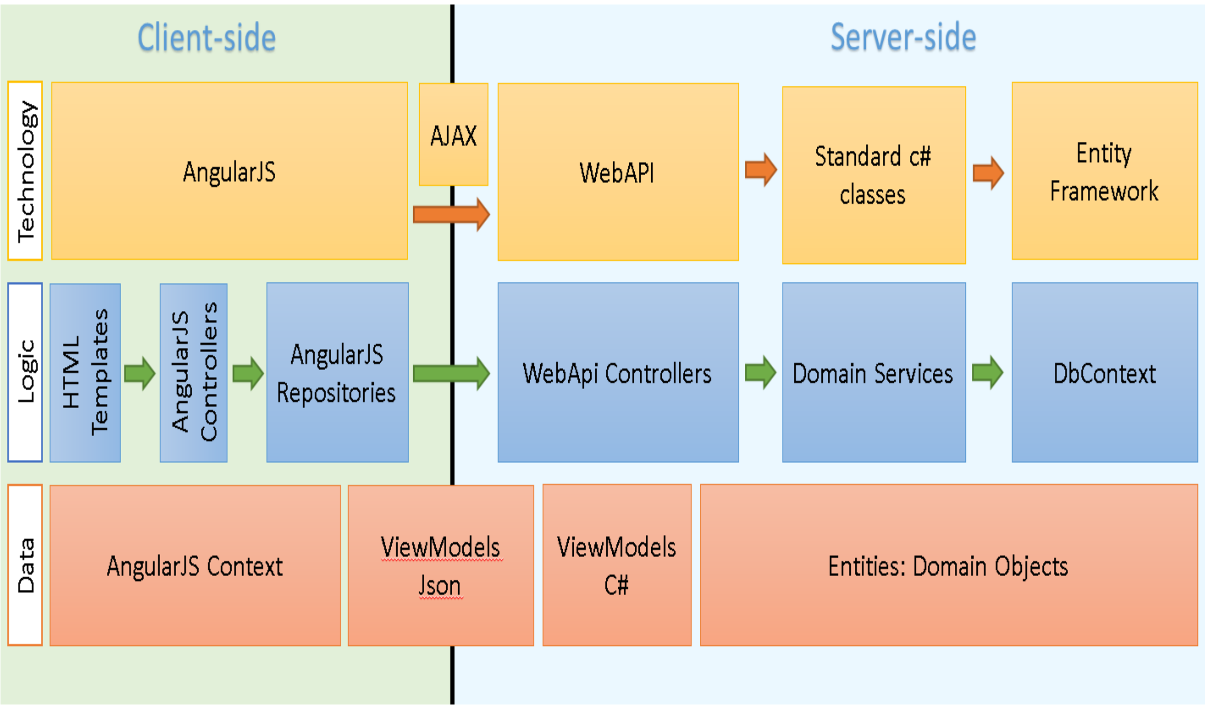


Figure 3: Architectural Diagram of AngularJS

4) Ember.js is an open-ended source, free client side for developing any web application, developed by Yehuda Katz. By using HTMLBars template, it offers the new binding structure supporting binding of data for the creation of a link between two attributes that depends on each other for the variation in one's value according to the second one. It has his own tools- Ember Inspector used for debugging its applications and Ember CLI which is a command line tool used for fast building of the structure of a web application.

Ember consists of five key concepts:

- Routes

  In Ember, the state of an application is represented by a URL. Each URL has a corresponding route object that controls what is visible to the user.

- Models

  Every route has an associated model, containing the data associated with the current state of the application. While one can use window, fetch to load JSON objects from a server and use those objects as models, most applications use a model library such as Ember Data to handle this.

- Templates

  Templates are used to build the application's HTML and are written with the HTMLBars templating language. (HTMLBars is a variation of Handlebars that builds DOM elements rather than a String.)

- Components

  A component is a custom HTML tag. Behavior is implemented using JavaScript and its appearance is defined using HTMLBars templates. Components "own" their data. They can also be nested and can communicate with their parent components through actions (events). Other component libraries such as Polymer can also be used with Ember.

- Services

  Services are just singleton objects to hold long-lived data such as user sessions. Ember also provides dependency injection, declarative one-way data-flow, tracked properties, and automatically updating templates.

FUTURE DEVELOPMENT

Project status can be tracked via the core team meeting minutes. However major changes to Ember go through the Request For Comment process. This gives the Ember community a chance to give feedback on new proposals. Notable RFCs include:

- Engines

  Engines allow multiple logical applications to be composed together into a single application from the user's perspective. Currently released as an experimental addon

- Release cycle improvements

  Among other things it proposes changes to Ember CLI to support "svelte builds", which will strip out deprecated and unused features.

- Outlet Focusing

  Making Ember accessible by default. This RFC aims to improves the user experience for people using screen readers.

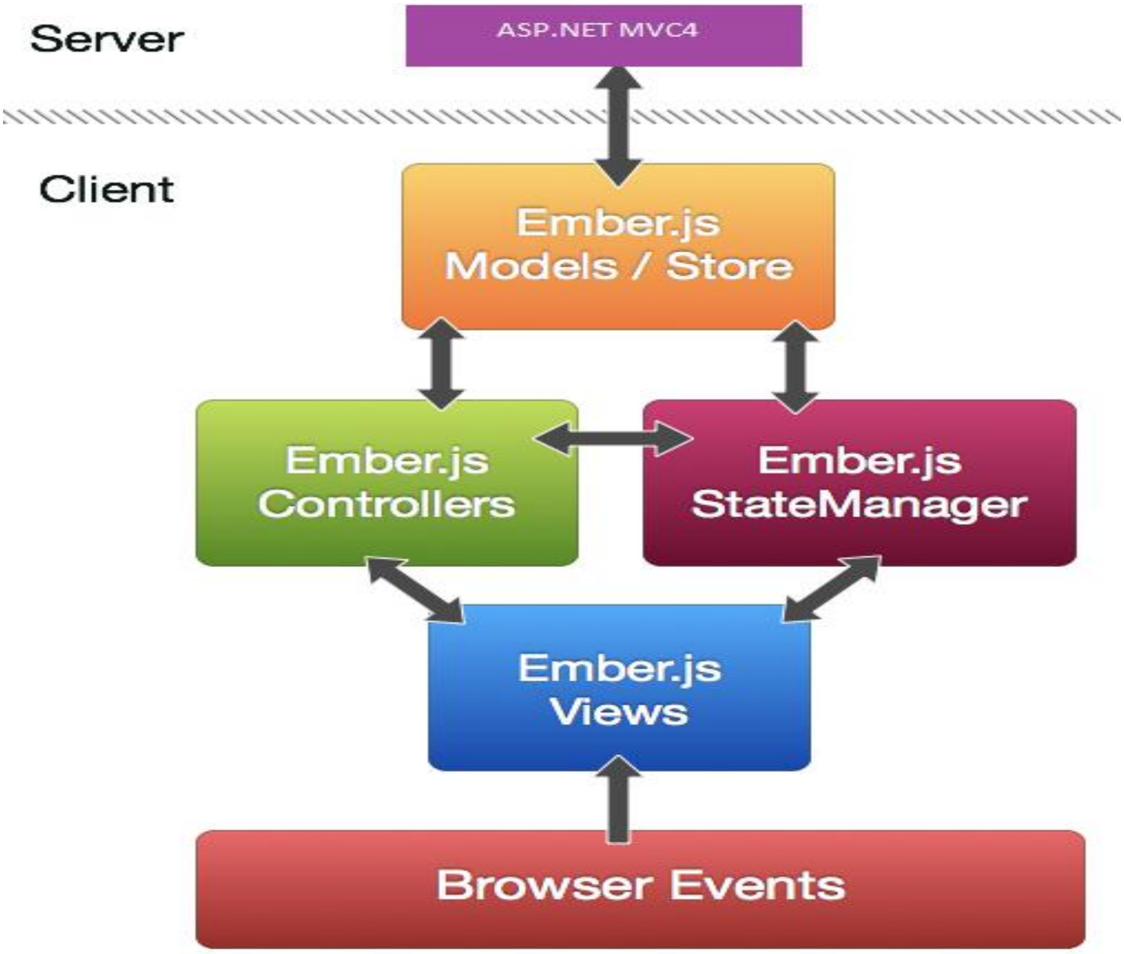


Figure 4: Architectural Diagram of VueJS

5) React is basically a library to build reusable components of User Interface offering a rich interface and not a MVC framework. Initially it was developed by Facebook for their personal use but later on it got recognition and valued by large companies like Netflix, Uber, Airbnb for their development. In it, each part manages their state on their own denoting a single view and organizes them to the UI, keeping it out of the DOM state. React is easily readable, fast because of its unidirectional flow and data binding feature. Developers opt for this library largely because it does not need CSS for animation but provides its own built APIs for the animations. It is widely used in applications which require a large set of data as well as users.

React versions

The initial version, 0.3.0 of React is released on May, 2013 and the latest version, 17.0.1 is released on October, 2020. The major version introduces breaking changes and the minor version introduces new feature without breaking the existing functionality. Bug fixes are released as and when necessary. React follows the Sematic Versioning (semver) principle.

Features

The salient features of React library are as follows –

- Solid base architecture
- Extensible architecture
- Component based library
- JSX based design architecture
- Declarative UI library

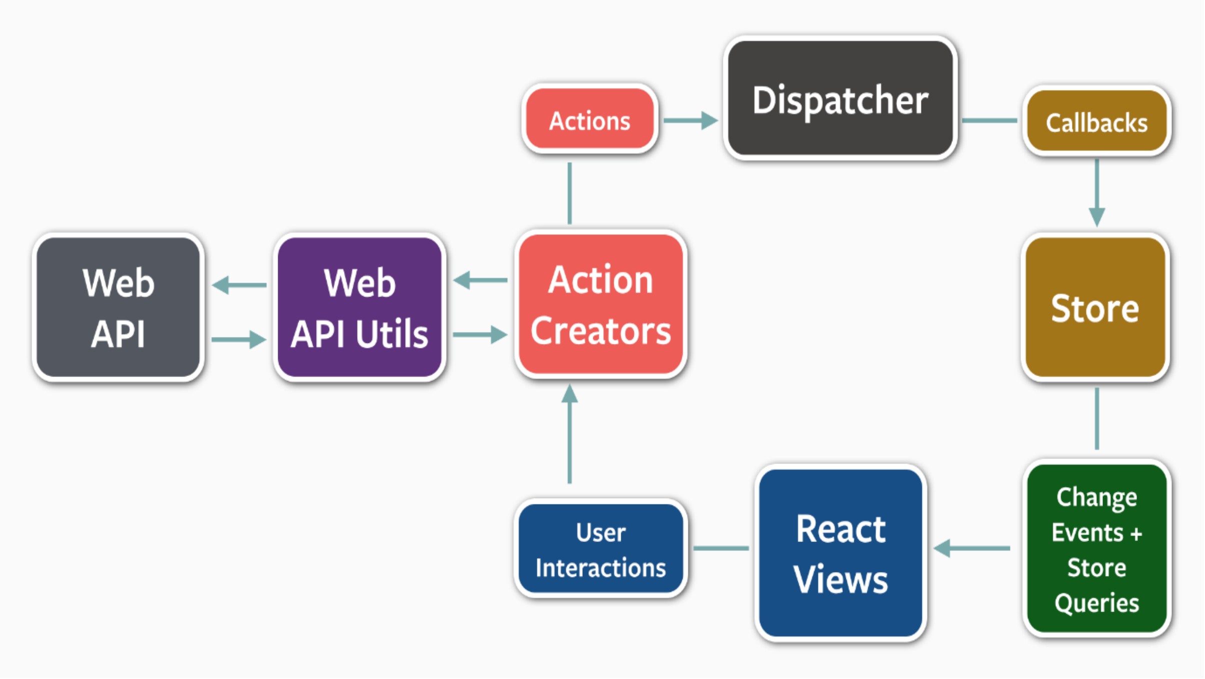


Figure 5: Architectural Diagram of React

For the comparison, we kept in mind different factors and got statistics as follows:

a) A survey done among developers, showing the percentage of people using frameworks as per their requirement

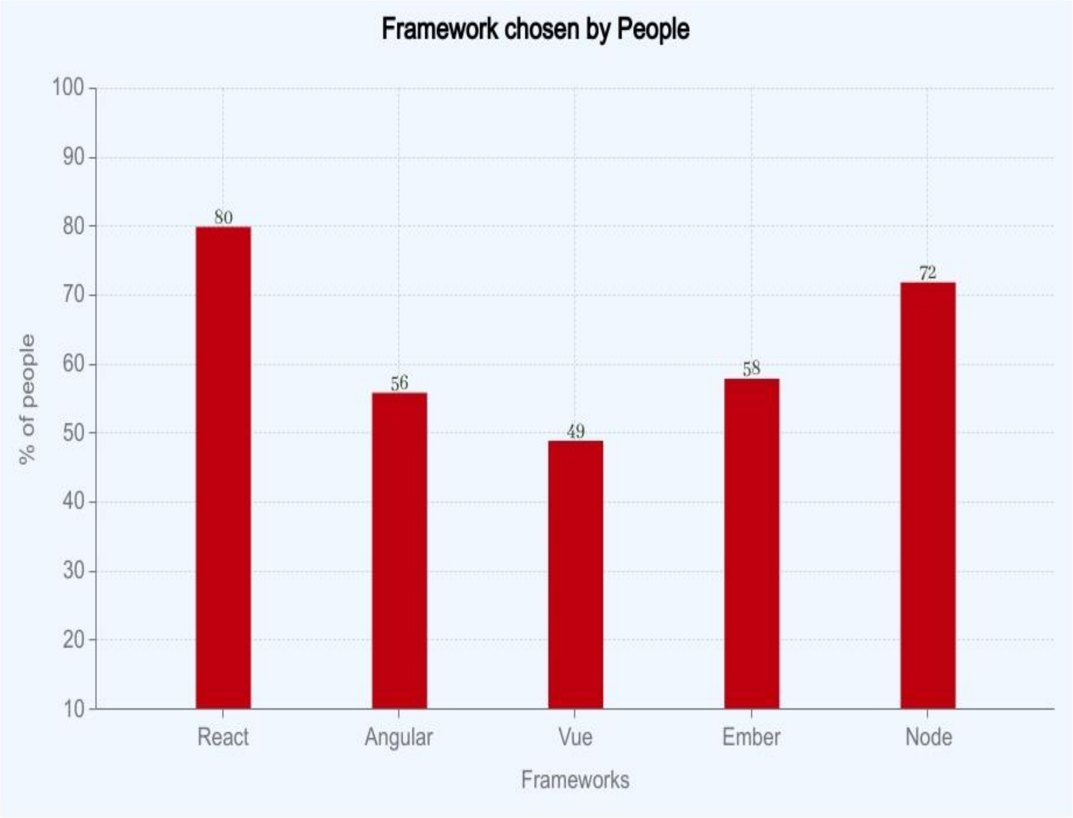


Figure 6: Framework chosen by people

b) Jobs in Framework to show the future in web application development

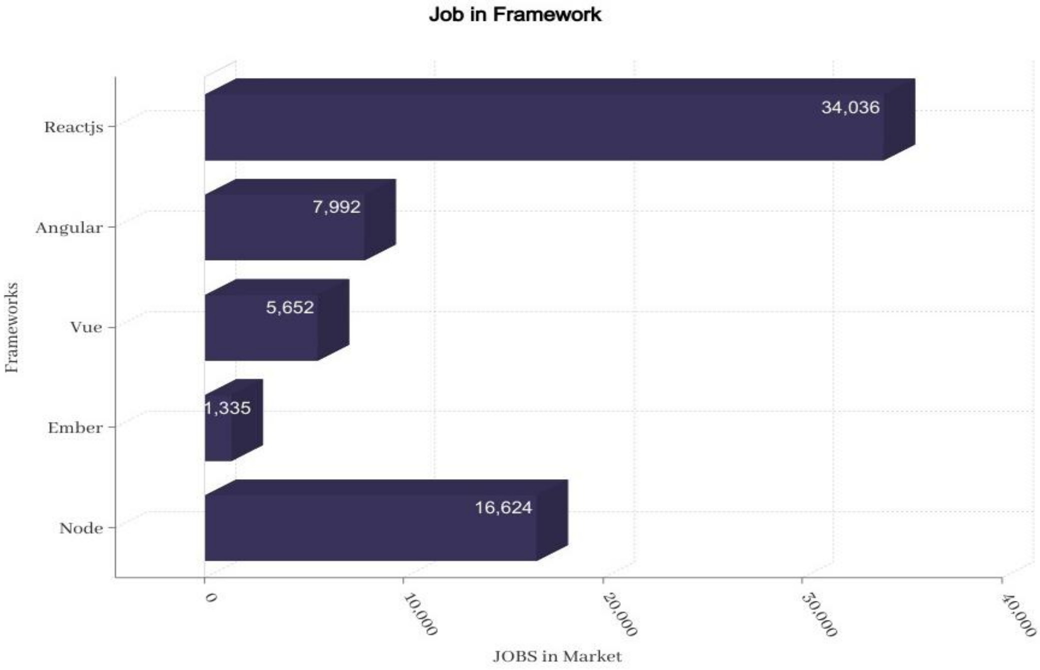


Figure 7: Jobs in Framework

PROJECT DESIGN

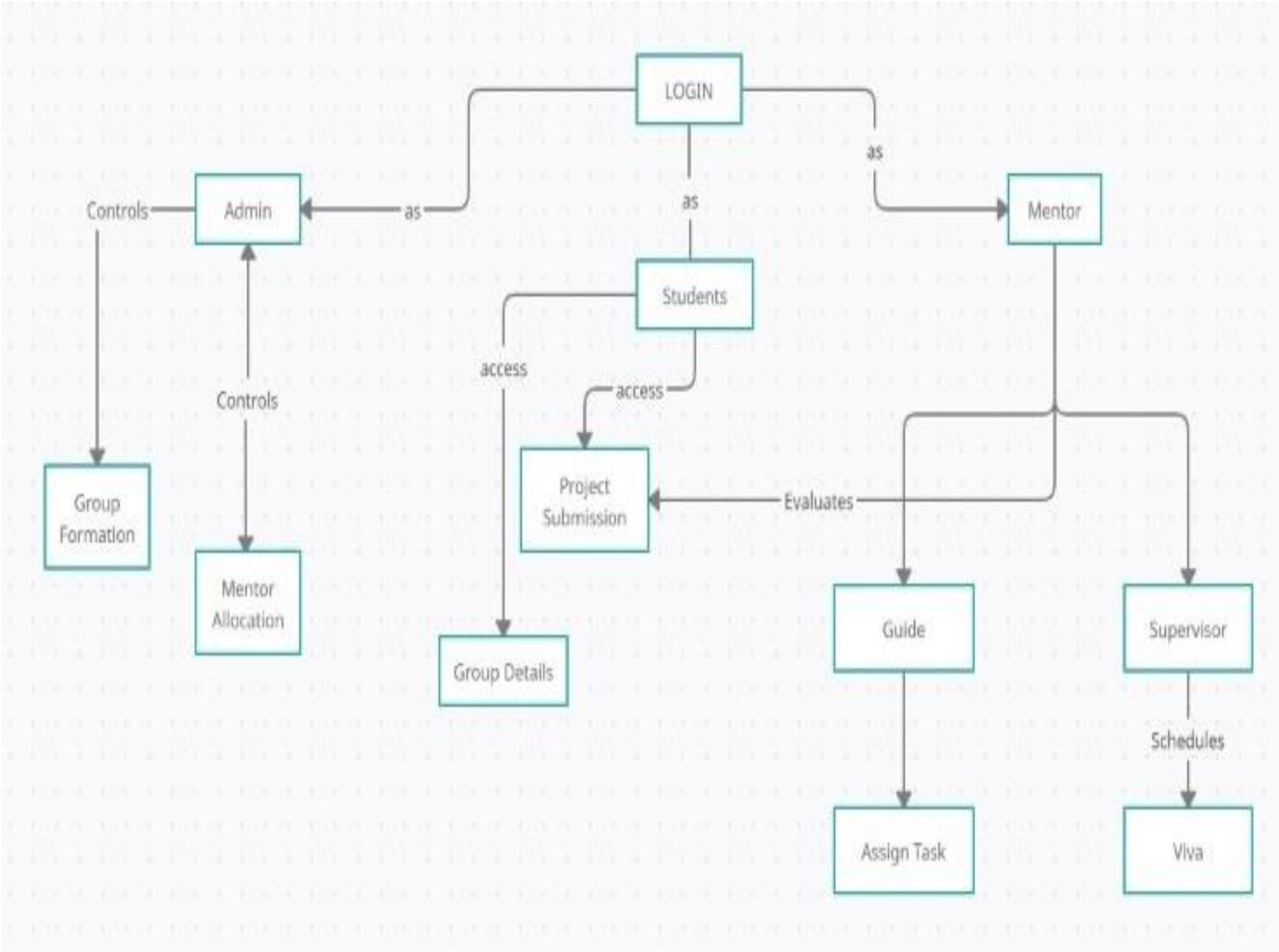


Figure 8: Project Design for the description of project

The website consists of three types of login - Admin, Mentor/Guide and Students where Admin is the supervisor who have all the authenticity and authorization over the web pages. Students can decide the number of students they want in their group, which have to be minimum 3-4 in number, and also the student they want to work with. Guide and reviewer allocation would be done by Admin or whom we call project coordinators in college.

Mentor would guide the students for the project they are working on and also give them tasks with the deadlines to maintain the curriculum. The link will be provided to the students and will be closed after the deadline, not only to maintain the discipline among students but also to help the mentor to evaluate them accordingly.

Students would upload their work and reviewers would be assign to evaluate them by accessing the documents uploaded by the students. Evaluation would be done on the basis of content of the document. Grades provided by reviewers after evaluating will not be visible to the students unless the reviewer wants.

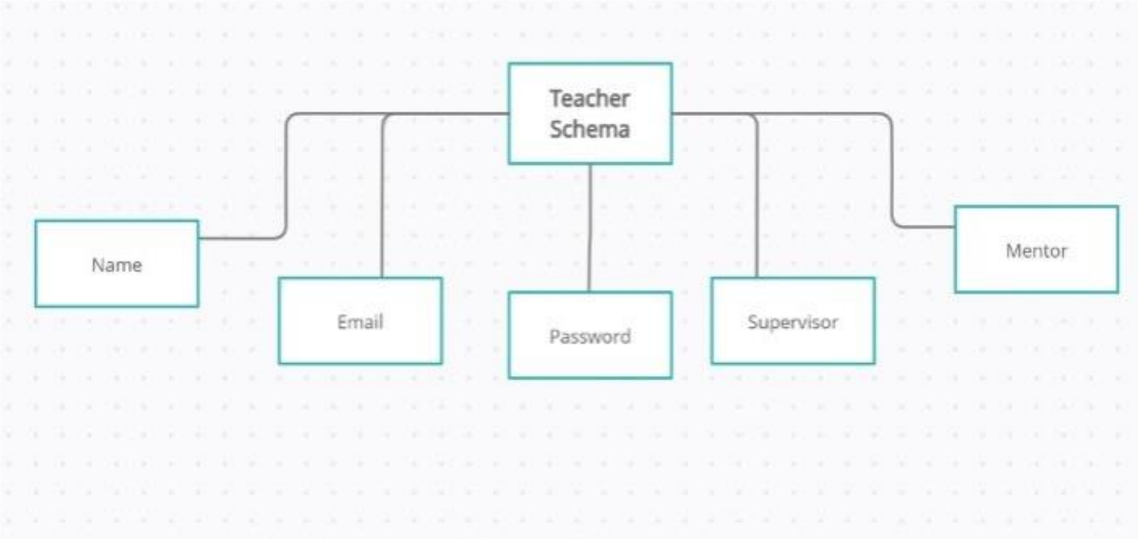


Figure 9: Teacher Schema for database designing

The schema consists of 5 components:

- Name
- E-mail
- Password
- Supervisor
- Mentor

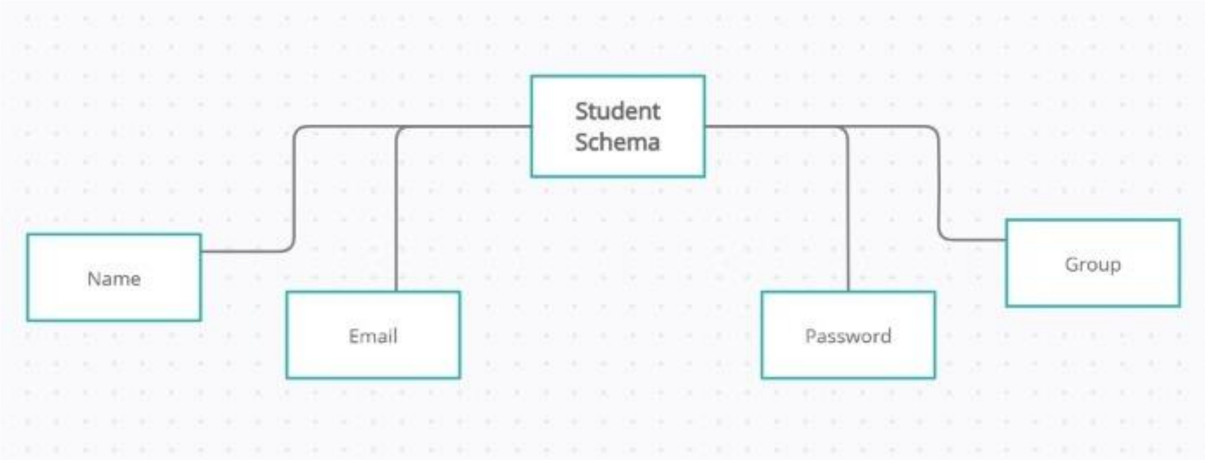


Figure 10: Student Schema for database designing

The schema consists of 5 components:

- Name
- E-mail
- Password
- Group ID allocated

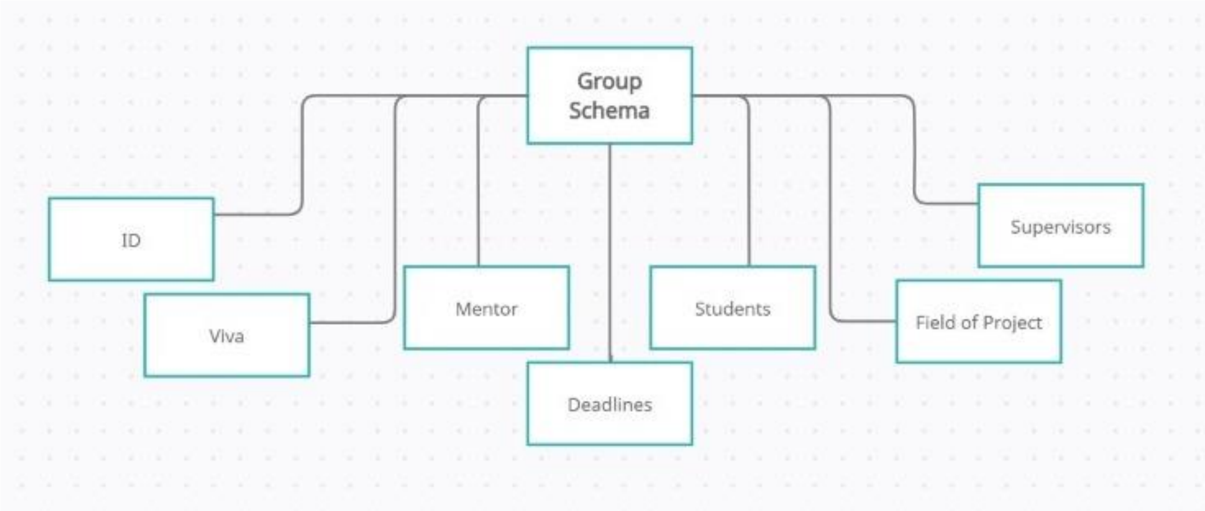
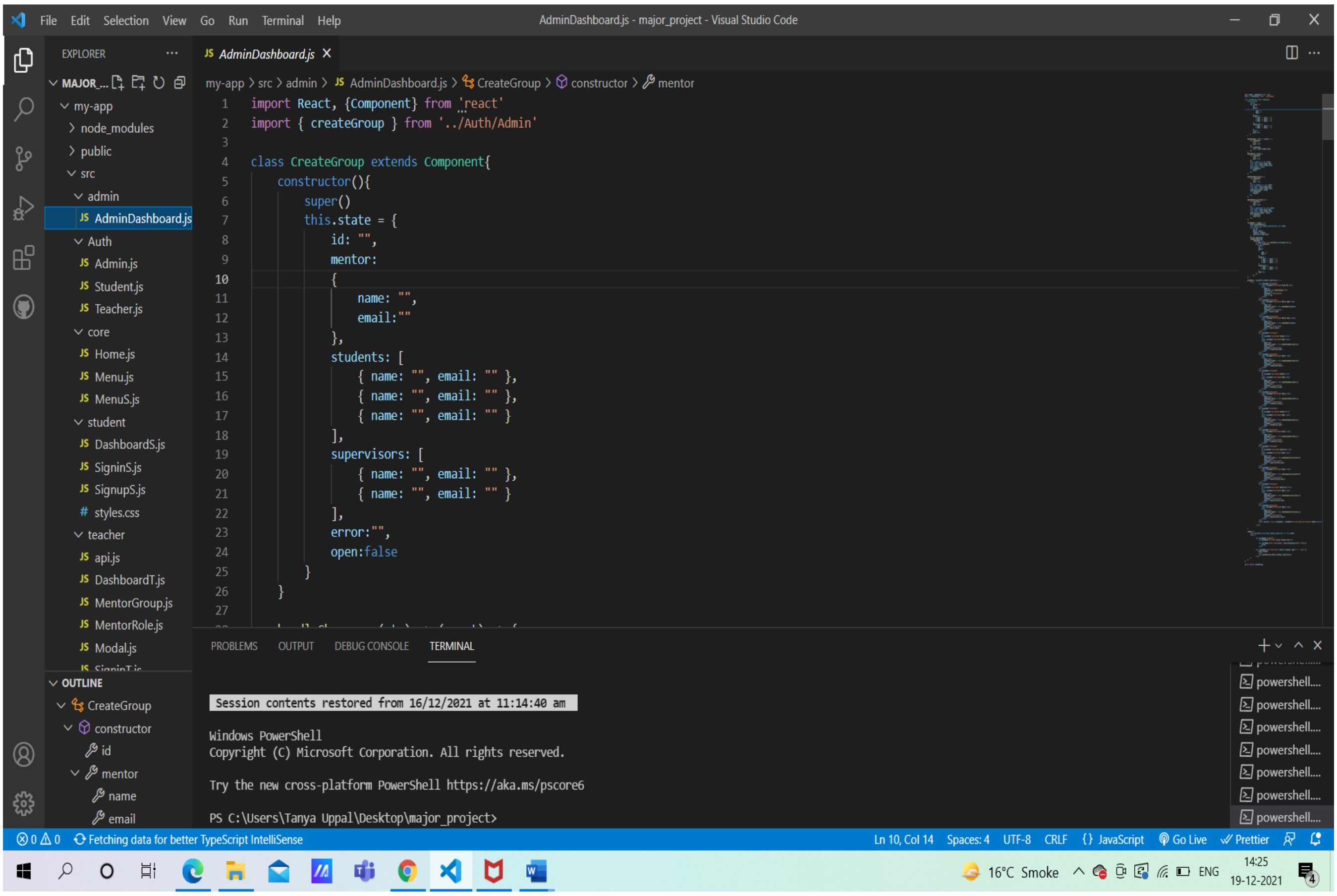


Figure 11: Group Schema for database designing

The schema consists of 5 components:

- ID
- Viva schedule
- Mentor
- Deadlines
- Students under them
- Project Domain
- Supervisor

CHAPTER 3: WORKING OF A PROJECT


File Edit Selection View Go Run Terminal Help
AdminDashboard.js - major_project - Visual Studio Code
EXPLORER
AdminDashboard.js
my-app > src > admin > AdminDashboard.js > CreateGroup > constructor > mentor
MAJOR_...
my-app
node_modules
public
src
admin
AdminDashboard.js
Auth
Admin.js
Student.js
Teacher.js
core
Home.js
Menu.js
MenuS.js
student
DashboardS.js
SigninS.js
SignupS.js
styles.css
teacher
api.js
DashboardT.js
MentorGroup.js
MentorRole.js
Modal.js
OUTLINE
CreateGroup
constructor
id
mentor
name
email
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
powershell...
Ln 10, Col 14 Spaces: 4 UTF-8 CRLF JavaScript Go Live Prettier
Fetching data for better TypeScript IntelliSense
16°C Smoke ENG 14:25 19-12-2021

```
import React, {Component} from 'react'
import { createGroup } from '../Auth/Admin'

class CreateGroup extends Component{
    constructor(){
        super()
        this.state = {
            id: "",
            mentor:
            {
                name: "",
                email:""
            },
            students: [
                { name: "", email: "" },
                { name: "", email: "" },
                { name: "", email: "" }
            ],
            supervisors: [
                { name: "", email: "" },
                { name: "", email: "" }
            ],
            error:"",
            open:false
        }
    }
```

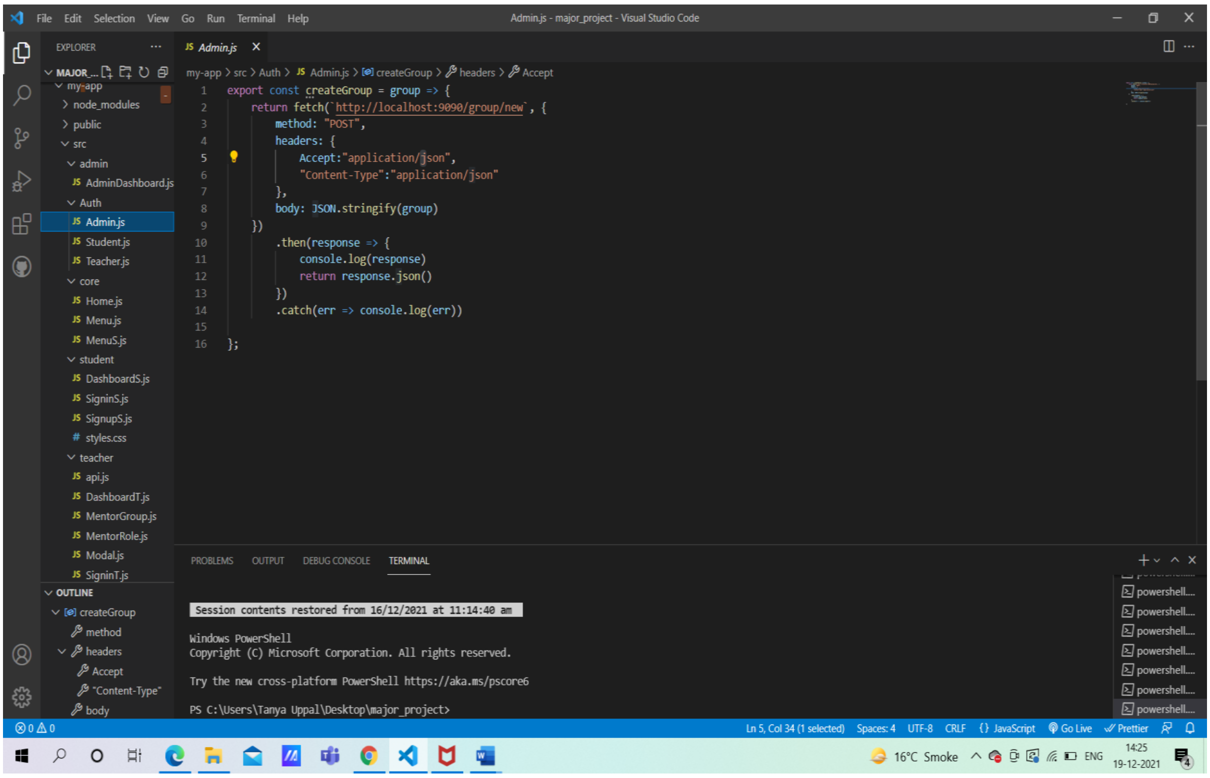
```
Session contents restored from 16/12/2021 at 11:14:40 am

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Tanya Uppal\Desktop\major_project>
```

Figure 12: JavaScript Code for Admin dashboard

The following figure shows a single snapshot of coding done for Admin dashboard. It imports from Authentication's Admin to connect it through links. Then it consists of different ID functions like for Mentor, Students and Supervisors. All of them take two entries for name and for e-mail.

Then it sets the state to different conditions for error and to revert back accordingly. It has similarly for handle event for Mentor, Students and Supervisors.

It also consists the coding for Submit button with its event handlers for HTML coding for formation of groups and to allocate the groups. These groups are then allocated with respective guides and then to the supervisor as well as reviewer allocation is done. These works are all done by the Admin dashboard login page.



```javascript
export const createGroup = group => {
    return fetch(`http://localhost:9090/group/new`, {
        method: "POST",
        headers: {
            Accept:"application/json",
            "Content-Type":"application/json"
        },
        body: JSON.stringify(group)
    })
        .then(response => {
            console.log(response)
            return response.json()
        })
        .catch(err => console.log(err))

};
```

```
Session contents restored from 16/12/2021 at 11:14:40 am

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Tanya Uppal\Desktop\major_project>
```

Admin.js - major_project - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
my-app > src > Auth > Admin.js > createGroup > headers > Accept
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 5, Col 34 (1 selected) Spaces: 4 UTF-8 CRLF {} JavaScript Go Live Prettier
16°C Smoke ENG 14:25 19-12-2021

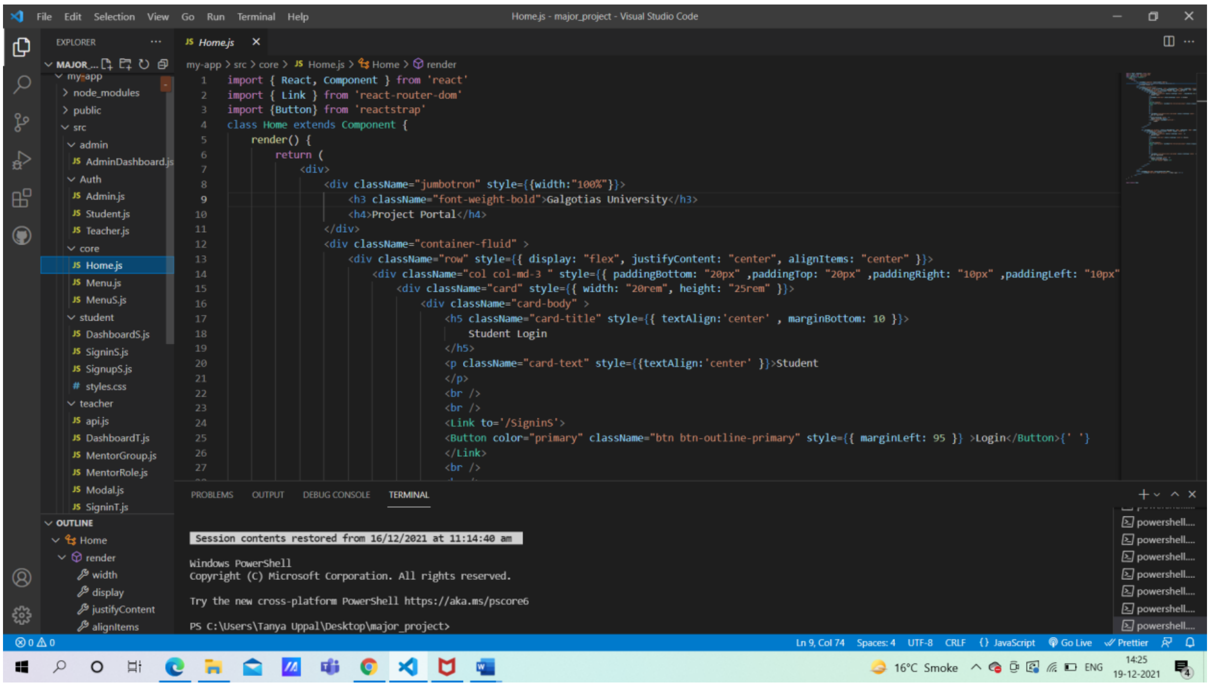
Figure 13: JavaScript Code for Authorisation dashboard

The following figure shows a single snapshot of coding done for Authorisation dashboard which further consists of Admin, Student and Teacher. It imports from different signups to connect it through links. Then it consists of different ID functions like for Mentor, Students and Supervisors. All of them take two entries for name and for e-mail and then its sign-out, sign-in options.

Then it sets the state to different conditions for error and to revert back accordingly. It has similarly for handle event for Mentor, Students, Admin and Supervisors.

It also consists the coding for Submit button with its event handlers for HTML coding for signing the individuals according to their designation. For example, Student's JS file will take them to the Student Dashboard, Teacher's JS file will take them to the Teacher Dashboard and then same goes for the admin dashboard.

This helps in signing as well as signup functions.



File Edit Selection View Go Run Terminal Help
Home.js - major_project - Visual Studio Code
EXPLORER
Home.js
my-app > src > core > Home.js > Home > render
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 9, Col 74 Spaces: 4 UTF-8 CRLF JavaScript Go Live Prettier

```
import { React, Component } from 'react'
import { Link } from 'react-router-dom'
import {Button} from 'reactstrap'
class Home extends Component {
    render() {
        return (
            <div>
                <div className="jumbotron" style={{width:"100%"}}>
                    <h3 className="font-weight-bold">Galgotias University</h3>
                    <h4>Project Portal</h4>
                </div>
                <div className="container-fluid" >
                    <div className="row" style={{ display: "flex", justifyContent: "center", alignItems: "center" }}>
                        <div className="col col-md-3 " style={{ paddingBottom: "20px" ,paddingTop: "20px" ,paddingRight: "10px" ,paddingLeft: "10px"
                            <div className="card" style={{ width: "20rem", height: "25rem" }}>
                                <div className="card-body" >
                                    <h5 className="card-title" style={{ textAlign:'center' , marginBottom: 10 }}>
                                        Student Login
                                    </h5>
                                    <p className="card-text" style={{textAlign:'center' }}>Student
                                    </p>
                                    <br />
                                    <br />
                                    <Link to='/SigninS'>
                                    <Button color="primary" className="btn btn-outline-primary" style={{ marginLeft: 95 }} >Login</Button>{' '}
                                    </Link>
                                    <br />
```

```
Session contents restored from 16/12/2021 at 11:14:40 am

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Tanya Uppal\Desktop\major_project>
```

Figure 14: JavaScript Code for Home Page dashboard

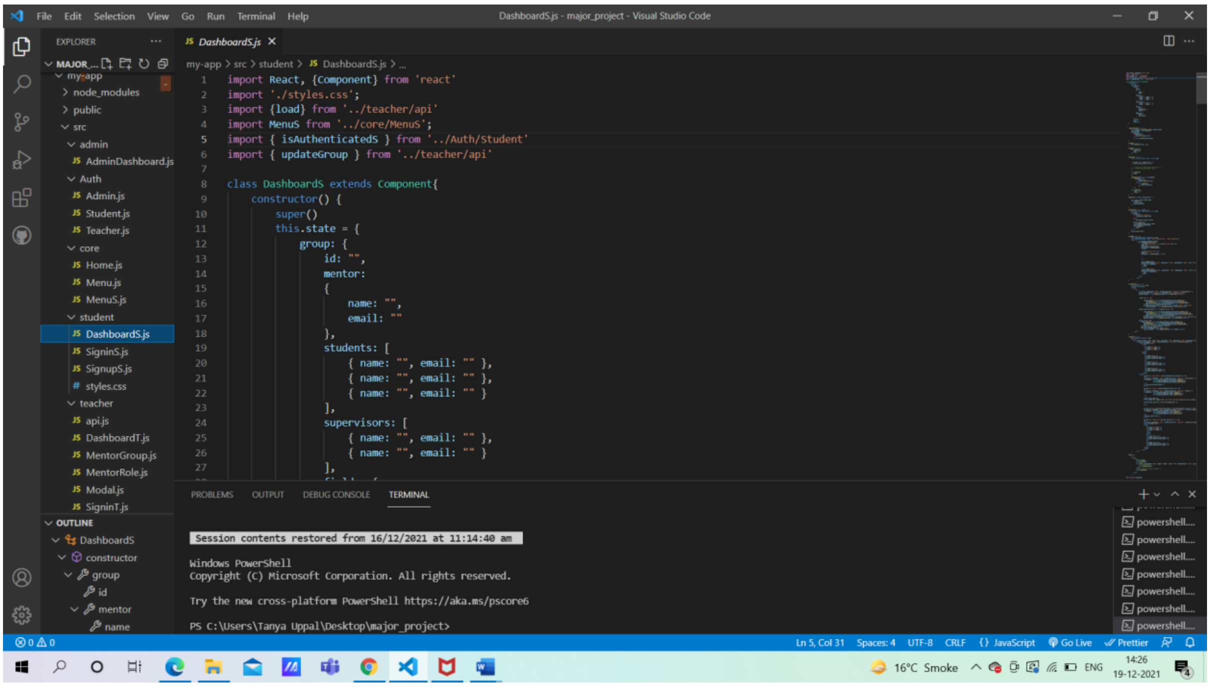
The following figure shows a single snapshot of coding done for Core dashboard. It imports from Link’s and Button’s click to connect it through links. Then it consists of different ID functions like for Mentor, Students and Supervisors. It consists of all types of clicks and links operations.

Then it sets the state to different conditions for error and to revert back accordingly. It has similarly for handle event for Mentor, Students and Supervisors.

It also consists the coding for Submit button with its event handlers for coding of signing out the student, teacher or mentor. It also shows the active status.

Menu Bar is present in it which shows functions like course enrolled and further.

MenuS is for students. It consists of Menu option like for checking the timeline of their work and how it will further effect the growth of their project.



```javascript
import React, {Component} from 'react'
import './styles.css';
import {load} from '../teacher/api'
import MenuS from '../core/MenuS';
import { isAuthenticatedS } from '../Auth/Student'
import { updateGroup } from '../teacher/api'

class DashboardS extends Component{
    constructor() {
        super()
        this.state = {
            group: {
                id: "",
                mentor:
                {
                    name: "",
                    email: ""
                },
                students: [
                    { name: "", email: "" },
                    { name: "", email: "" },
                    { name: "", email: "" }
                ],
                supervisors: [
                    { name: "", email: "" },
                    { name: "", email: "" }
                ],
```

```
Session contents restored from 16/12/2021 at 11:14:40 am

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Tanya Uppal\Desktop\major_project>
```

Figure 15: JavaScript Code for Student dashboard

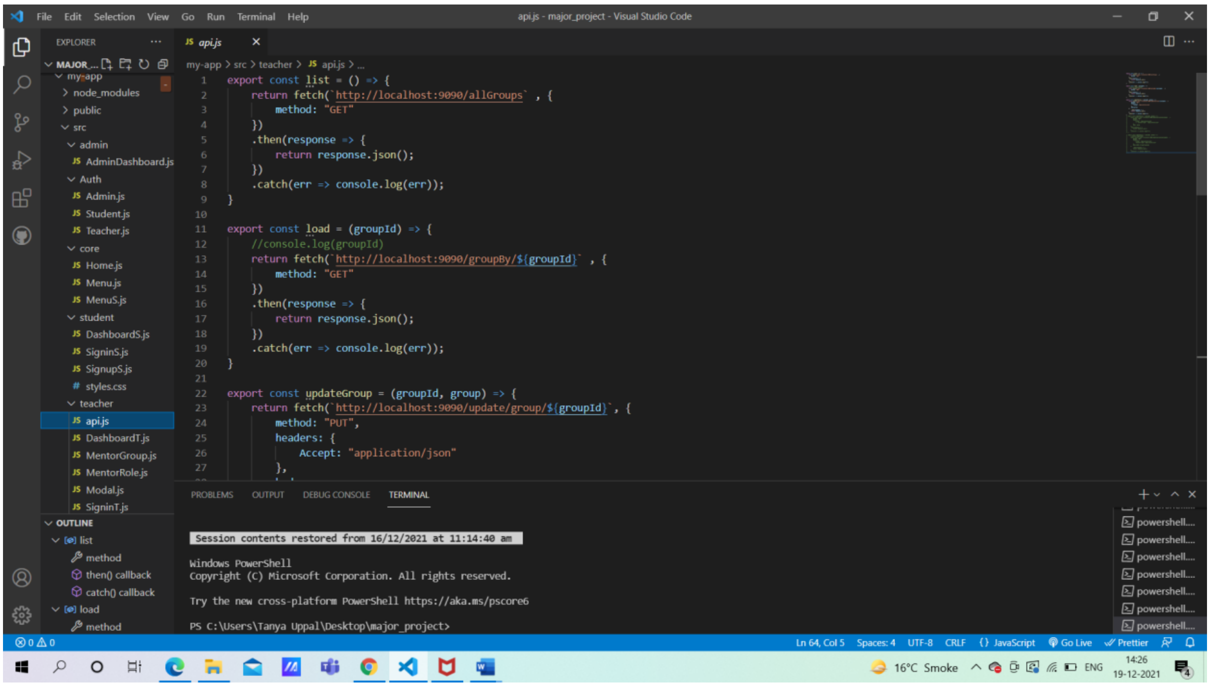
The following figure shows a single snapshot of coding done for Student dashboard. It imports from styles file for CSS, teacher's API, Student's authentication and Core's Menu to connect it through links. Then it consists of different ID functions like for mentor, student, supervisor, fields and deadlines. All given functions consist of different modules like mentor, students and supervisors consists of name and e-mail and fields and deadlines consists of title and descriptions.

Then it sets the state to different conditions for error and to revert back accordingly. It has similarly for handle event for Mentor, Students and Supervisors.

All these functions allow Students to get their assignment and to know the deadline of their project. They also become aware of the guide allocated to them as well as supervisor supervising them. All these things require the authentication of admin to check and update the students about. The group ID is also allotted to the students through this dashboard.

Logging-in and logging-out are also features of this page. Students can also see the deadline for the submission which comes in Date, Month and year.

Students do the submission through these modules according to the desired input document type.



```
export const list = () => {
    return fetch(`http://localhost:9090/allGroups` , {
        method: "GET"
    })
    .then(response => {
        return response.json();
    })
    .catch(err => console.log(err));
}

export const load = (groupId) => {
    //console.log(groupId)
    return fetch(`http://localhost:9090/groupBy/${groupId}` , {
        method: "GET"
    })
    .then(response => {
        return response.json();
    })
    .catch(err => console.log(err));
}

export const updateGroup = (groupId, group) => {
    return fetch(`http://localhost:9090/update/group/${groupId}`, {
        method: "PUT",
        headers: {
            Accept: "application/json"
        },
```

```
 Session contents restored from 16/12/2021 at 11:14:40 am 

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Tanya Uppal\Desktop\major_project>
```

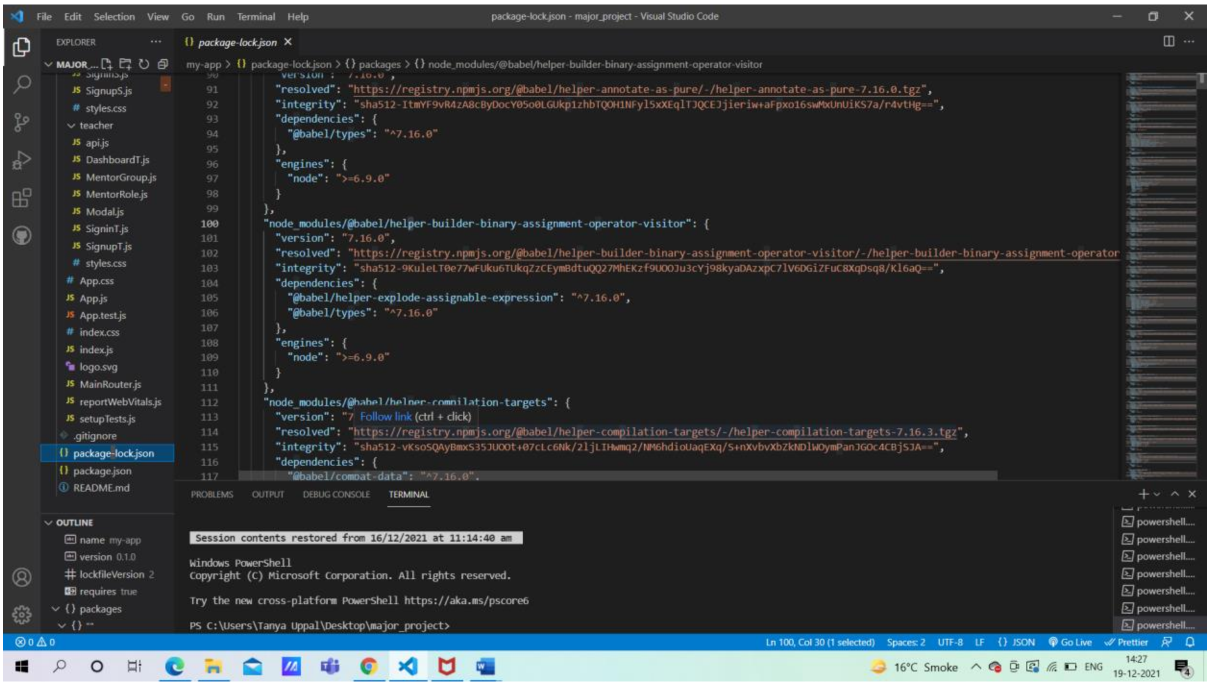
Figure 16: JavaScript Code for Teacher dashboard

The following figure shows a single snapshot of coding done for Teacher dashboard. It imports from styles file for CSS, Packet's API, Teacher's authentication and Core's Menu to connect it through links. Then it consists of different ID functions like for mentor, student, supervisor, fields and deadlines. All given functions consist of different modules like mentor, students and supervisors consists of name and e-mail and fields and deadlines consists of title and descriptions.

Then it sets the state to different conditions for error and to revert back accordingly. It has similarly for handle event for Mentor, Students and Supervisors.

All these functions allow Teachers to set their assignment and to update the deadline of the project under them. They also become aware of the students allocated to them as well as supervisor supervising the students allocated. All these things require the authentication of admin to check and update the students about. The group ID is also allotted to the teachers through this dashboard.

Logging-in and logging-out are also features of this page. Teachers can also see the submission students made and also the delay in the submission made.


File Edit Selection View Go Run Terminal Help
package-lock.json - major_project - Visual Studio Code
package-lock.json
my-app > package-lock.json > packages > node_modules/@babel/helper-builder-binary-assignment-operator-visitor
Follow link (ctrl + click)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 100, Col 30 (1 selected) Spaces: 2 UTF-8 LF JSON Go Live Prettier
16°C Smoke ENG 14:27 19-12-2021

```
      "version": "7.16.0",
      "resolved": "https://registry.npmjs.org/@babel/helper-annotate-as-pure/-/helper-annotate-as-pure-7.16.0.tgz",
      "integrity": "sha512-ItmYF9vR4zA8cByDocY05o0LGUkp1zhbTQOH1NFyl5xXEqlTJQCEJjieriw+aFpxo16swMxUnUiKS7a/r4vtHg==",
      "dependencies": {
        "@babel/types": "^7.16.0"
      },
      "engines": {
        "node": ">=6.9.0"
      }
    },
    "node_modules/@babel/helper-builder-binary-assignment-operator-visitor": {
      "version": "7.16.0",
      "resolved": "https://registry.npmjs.org/@babel/helper-builder-binary-assignment-operator-visitor/-/helper-builder-binary-assignment-operator
      "integrity": "sha512-9KuleLT0e77wFUku6TUkqZzCEymBdtuQQ27MhEKzf9UOOJu3cYj98kyaDAzxpC7lV6DGiZFuC8XqDsq8/Kl6aQ==",
      "dependencies": {
        "@babel/helper-explode-assignable-expression": "^7.16.0",
        "@babel/types": "^7.16.0"
      },
      "engines": {
        "node": ">=6.9.0"
      }
    },
    "node_modules/@babel/helper-compilation-targets": {
      "version": "7
      "resolved": "https://registry.npmjs.org/@babel/helper-compilation-targets/-/helper-compilation-targets-7.16.3.tgz",
      "integrity": "sha512-vKsoSQAyBmxS35JUOOt+07cLc6Nk/2ljLIHwmq2/NM6hdioUaqEXq/S+nXvbvXbZkNDlWOymPanJGOc4CBjSJA==",
      "dependencies": {
        "@babel/compat-data": "^7.16.0",
```

```
 Session contents restored from 16/12/2021 at 11:14:40 am 

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Tanya Uppal\Desktop\major_project>
```

Figure 17: JSON file for package loading

package.json is used to store the metadata associated with the project as well as to store the list of dependency packages. In order to add dependency packages to your projects, you need to create package.json file. The file makes it easy for others to manage and install the packages associated with the project.

A package.json file:

- lists the packages your project depends on
- specifies versions of a package that your project can use.
- makes your build reproducible, and therefore easier to share with other developers.

This file outlines all the settings for the React app.

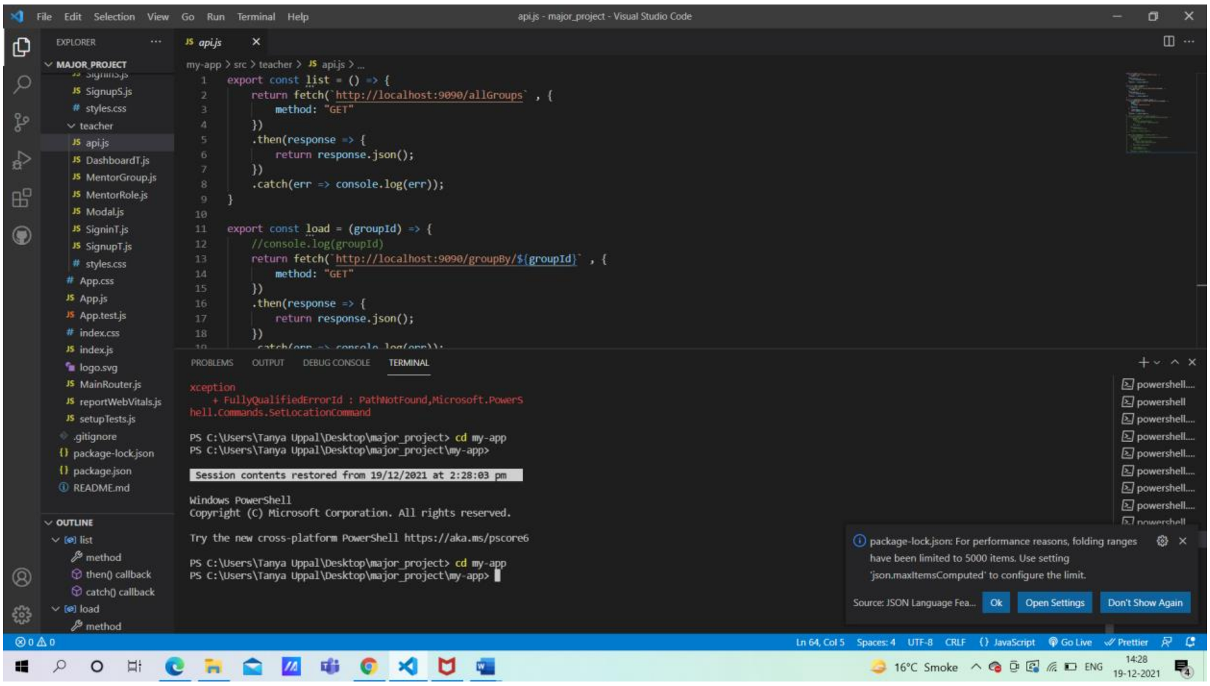
Each of the attributes in the file has its importance in some way or the other. Where some of these are basic and clear from their names, some are not. This article is aimed at simplifying and developing a better understanding of the same.

- name is the name of your app, which you give while executing create-react-app<name-of-application> . You can give any name of your choice to the app, the only condition being is that it should be in lowercase. It may also contain hyphens and underscores.

- version is the current version of your app. The version field must be of the form x.x.x. By default, create-react-app initializes it as 0.1.0
- "private": true is one of the most crucial attributes. The use is that if you set private as true in your package.json, then npm will refuse to publish it within npm ecosystem. This is a way to prevent the accidental publication of private repositories.
- dependencies contains all the required node modules and versions required for the application in production. In the snapshot above, it contains three dependencies, which allows us to use react , react-domand react-scripts in our JavaScript. react-scripts provide a set of useful development scripts for working with React.

The file will be created in the respective directory and it looks like the below screenshot.

- name default is set to author name unless in a git directory, in which case, it will be the name of the repository
- version always 0.1.0
- description default is set to empty string
- main always index.js
- scripts by default, creates an empty test script
- author default is set to empty string
- license ISC



```
export const list = () => {
    return fetch(`http://localhost:9090/allGroups` , {
        method: "GET"
    })
    .then(response => {
        return response.json();
    })
    .catch(err => console.log(err));
}

export const load = (groupId) => {
    //console.log(groupId)
    return fetch(`http://localhost:9090/groupBy/${groupId}` , {
        method: "GET"
    })
    .then(response => {
        return response.json();
    })
    .catch(err => console.log(err));
```

```
xception
    + FullyQualifiedErrorId : PathNotFound,Microsoft.PowerS
hell.Commands.SetLocationCommand

PS C:\Users\Tanya Uppal\Desktop\major_project> cd my-app
PS C:\Users\Tanya Uppal\Desktop\major_project\my-app>

 Session contents restored from 19/12/2021 at 2:28:03 pm

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Tanya Uppal\Desktop\major_project> cd my-app
PS C:\Users\Tanya Uppal\Desktop\major_project\my-app>
```

package-lock.json: For performance reasons, folding ranges have been limited to 5000 items. Use setting 'json.maxItemsComputed' to configure the limit.

Source: JSON Language Fea... Ok Open Settings Don't Show Again

Figure 18: Terminal script running command

This runs an arbitrary command from a package's "scripts" object. If no "command" is provided, it will list the available scripts.

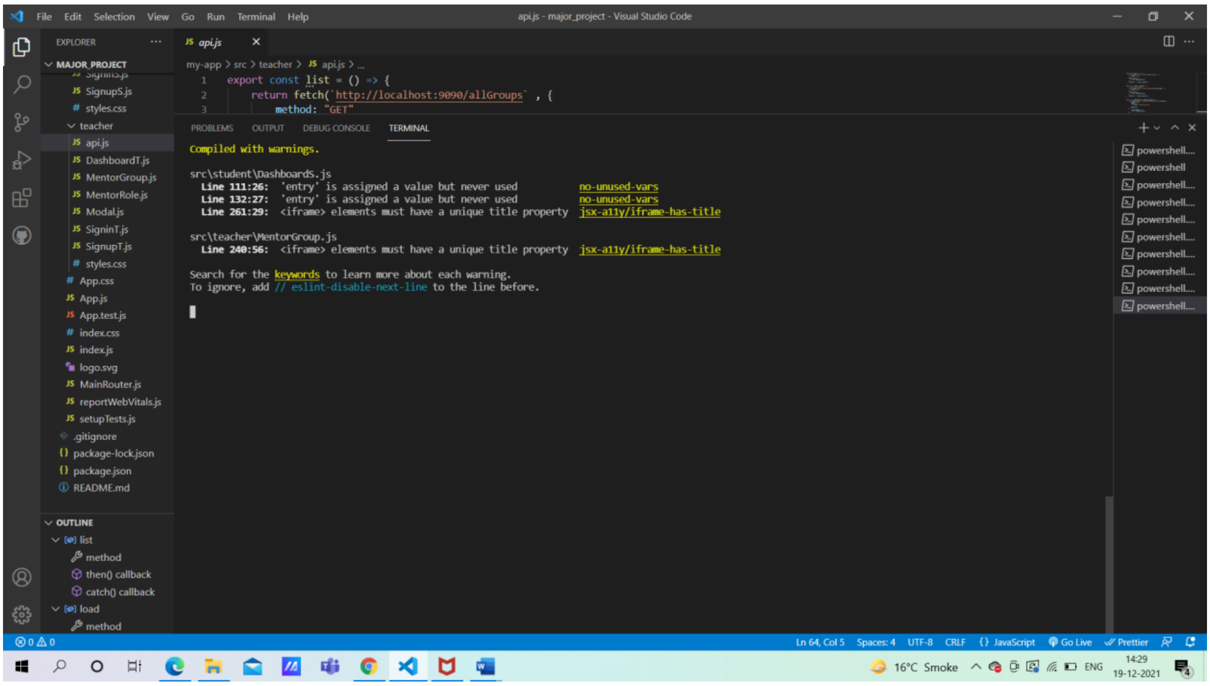
run[-script] is used by the test, start, restart, and stop commands, but can be called directly, as well. When the scripts in the package are printed out, they're separated into lifecycle (test, start, restart) and directly-run scripts.

Any positional arguments are passed to the specified script. Use -- to pass --prefixed flags and options which would otherwise be parsed by npm.

The arguments will only be passed to the script specified after npm run and not to any pre or post script.

The env script is a special built-in command that can be used to list environment variables that will be available to the script at runtime. If an "env" command is defined in your package, it will take precedence over the built-in.

In addition to the shell's pre-existing PATH, npm run adds node_modules/.bin to the PATH provided to scripts. Any binaries provided by locally-installed dependencies can be used without the node_modules/.bin prefix.



```
export const list = () => {
    return fetch(`http://localhost:9090/allGroups` , {
        method: "GET"
```

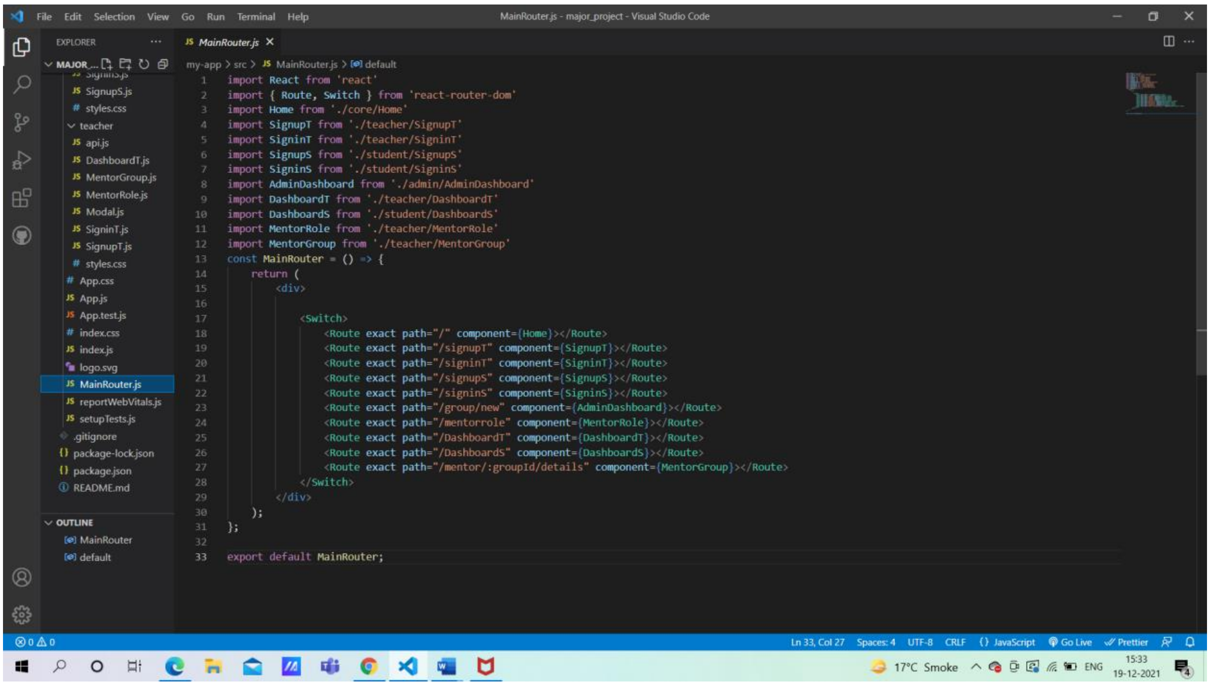
```
Compiled with warnings.

src\student\DashboardS.js
  Line 111:26:  'entry' is assigned a value but never used           no-unused-vars
  Line 132:27:  'entry' is assigned a value but never used           no-unused-vars
  Line 261:29:  <iframe> elements must have a unique title property  jsx-a11y/iframe-has-title

src\teacher\MentorGroup.js
  Line 240:56:  <iframe> elements must have a unique title property  jsx-a11y/iframe-has-title

Search for the keywords to learn more about each warning.
To ignore, add // eslint-disable-next-line to the line before.
```

Figure 19: Terminal script running commands

This would build our application for production to the build directory. The "build" folder would contain all the static files which can be directly used on any web server. Also, the build command transpiles our source code into code which the browser can understand. It uses Babel for this and files are optimized for best performance. All of our JS files are bundled into a single minified file and even HTML/CSS code is minified to significantly reduce the download times on the client's browser.

This would run our application in development mode. We can just navigate to http:localhost:3000 in any browser to preview our app live. The page will automatically reload whenever it detects any code change in the source files. Warnings and errors can also be seen in the console.

Internally, npm start uses webpack dev server to start a dev server so that we can communicate with the same.



```
import React from 'react'
import { Route, Switch } from 'react-router-dom'
import Home from './core/Home'
import SignupT from './teacher/SignupT'
import SigninT from './teacher/SigninT'
import SignupS from './student/SignupS'
import SigninS from './student/SigninS'
import AdminDashboard from './admin/AdminDashboard'
import DashboardT from './teacher/DashboardT'
import DashboardS from './student/DashboardS'
import MentorRole from './teacher/MentorRole'
import MentorGroup from './teacher/MentorGroup'
const MainRouter = () => {
    return (
        <div>

            <Switch>
                <Route exact path="/" component={Home}></Route>
                <Route exact path="/signupT" component={SignupT}></Route>
                <Route exact path="/signinT" component={SigninT}></Route>
                <Route exact path="/signupS" component={SignupS}></Route>
                <Route exact path="/signinS" component={SigninS}></Route>
                <Route exact path="/group/new" component={AdminDashboard}></Route>
                <Route exact path="/mentorrole" component={MentorRole}></Route>
                <Route exact path="/DashboardT" component={DashboardT}></Route>
                <Route exact path="/DashboardS" component={DashboardS}></Route>
                <Route exact path="/mentor/:groupId/details" component={MentorGroup}></Route>
            </Switch>
        </div>
    );
};

export default MainRouter;
```

Figure 20: JavaScript Code for Main Router

Step 1 - Install a React Router

A simple way to install the react-router is to run the following code snippet in the command prompt window.

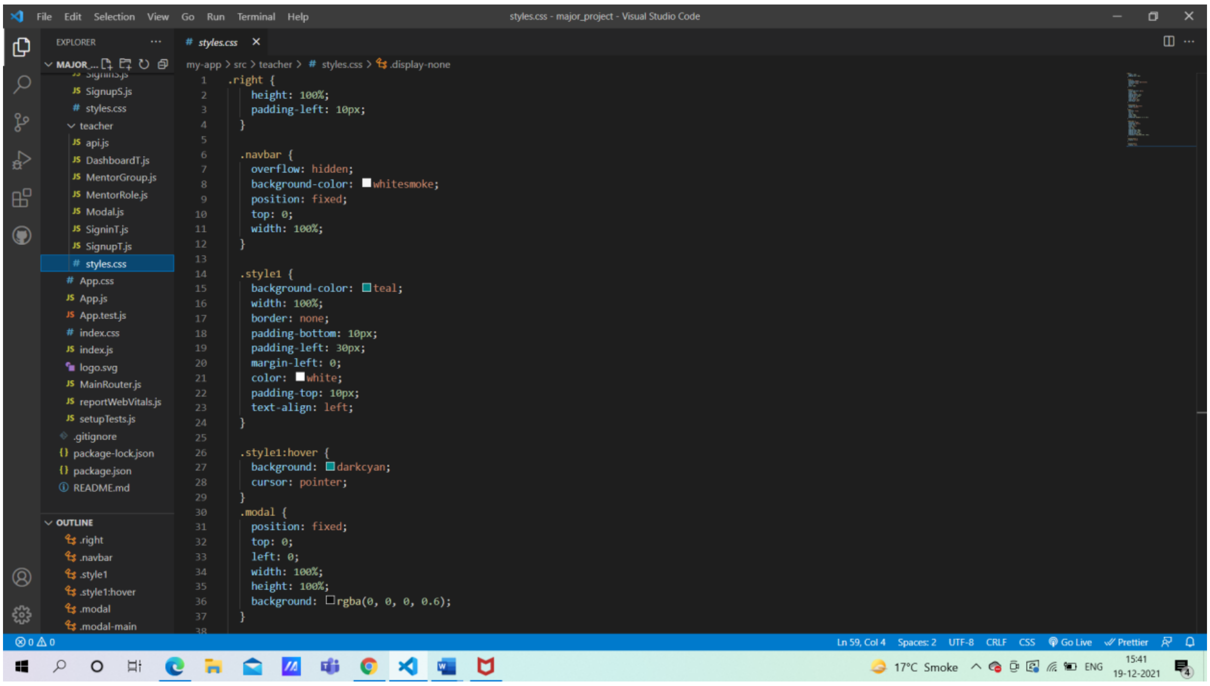
C:\Users\username\Desktop\reactApp>npm install react-router

Step 2 - Create Components

In this step, we will create four components. The App component will be used as a tab menu. The other three components (Home), (About) and (Contact) are rendered once the route has changed.

Step 3 - Add a Router

Now, we will add routes to the app. Instead of rendering App element like in the previous example, this time the Router will be rendered. We will also set components for each route.



```css
.right {
   height: 100%;
   padding-left: 10px;
 }

 .navbar {
   overflow: hidden;
   background-color: whitesmoke;
   position: fixed;
   top: 0;
   width: 100%;
 }

 .style1 {
   background-color: teal;
   width: 100%;
   border: none;
   padding-bottom: 10px;
   padding-left: 30px;
   margin-left: 0;
   color: white;
   padding-top: 10px;
   text-align: left;
 }

 .style1:hover {
   background: darkcyan;
   cursor: pointer;
 }
 .modal {
   position: fixed;
   top: 0;
   left: 0;
   width: 100%;
   height: 100%;
   background: rgba(0, 0, 0, 0.6);
 }
```

Figure 21: CSS file for styling the Teacher’s Dashboard

CSS in React is used to style the React App or Component. The style attribute is the most used attribute for styling in React applications, which adds dynamically-computed styles at render time. It accepts a JavaScript object in camelCased properties rather than a CSS string. There are many ways available to add styling to your React App or Component with CSS. Here, we are going to discuss mainly four ways to style React Components, which are given below:

1. Inline Styling
2. CSS Stylesheet
3. CSS Module
4. Styled Components

1. Inline Styling

The inline styles are specified with a JavaScript object in camelCase version of the style name. Its value is the style’ss value, which we usually take in a string.

2. CSS Stylesheet

You can write styling in a separate file for your React application, and save the file with a .css extension. Now, you can import this file in your application.

3. CSS Module

CSS Module is another way of adding styles to your application. It is a CSS file where all class names and animation names are scoped locally by default. It is available only for the component which imports it, means any styling you add can never be applied to other components without your permission, and you never need to worry about name conflicts. You can create CSS Module with the .module.css extension like a myStyles.module.css name.

4. Styled Components

Styled-components is a library for React. It uses enhance CSS for styling React component systems in your application, which is written with a mixture of JavaScript and CSS.

The styled-components provides:

- Automatic critical CSS
- No class name bugs
- Easier deletion of CSS
- Simple dynamic styling
- Painless maintenance

Installation

The styled-components library takes a single command to install in your React application. which is:

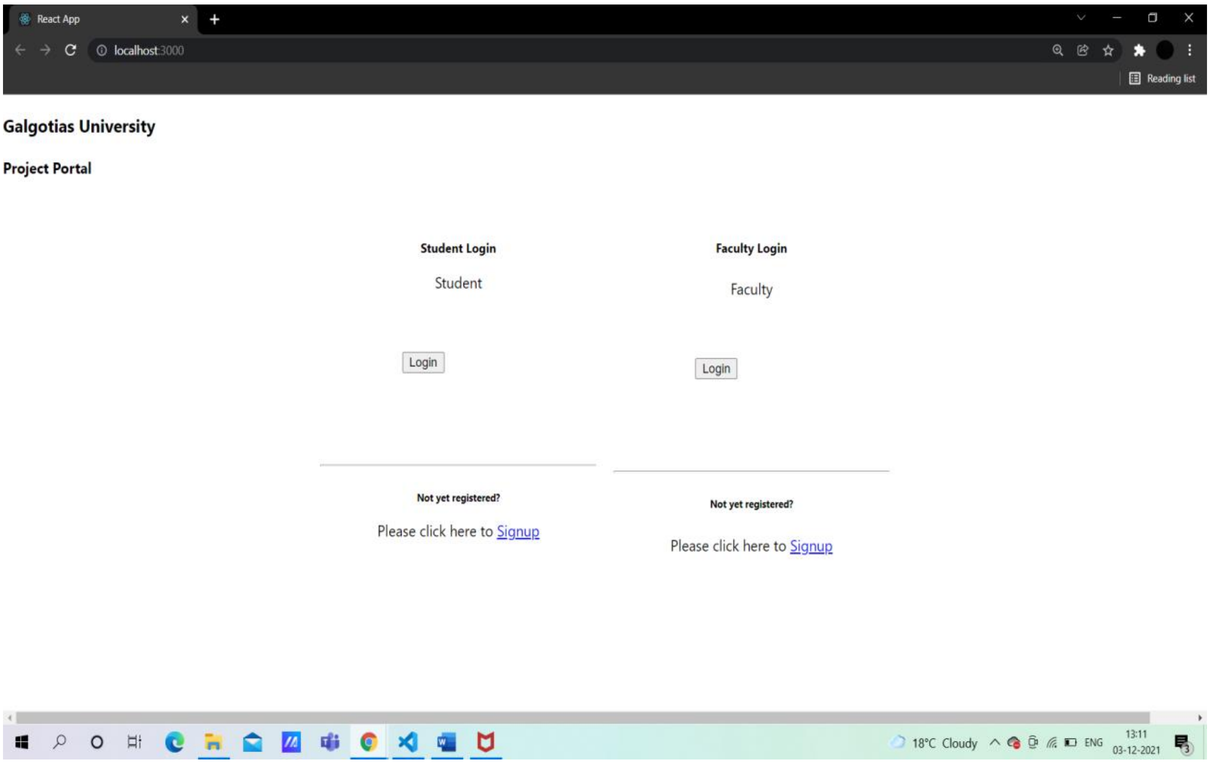
1. $ npm install styled-components --save

CHAPTER 5: RESULT AND DISCUSSION

In this report, we made an attempt to know the concepts of JavaScript frameworks and libraries. We studied the existing frameworks and analyzed them, which gave us a brief knowledge of major frameworks emerging the most in the world. Our literature review of the existing JavaScript led us to spot the major key points that made them remarkable and better than others. This finally leads us to redefine web development as the idea of a functional block of any type of app and how to move forward and in which direction for the development. Keeping these points in mind, we created a website to handle tasks as defined by the teachers to the students through the admin. In this following report, we have made an honest effort in covering various challenges and solutions one can experience while creating a web or mobile application. This manual will define the first step of the structure of the app-development cycle.

The general idea of HTML5 and other tools are the formal specification and the establishment of uniform solutions for technologies and functionalities which have already been in use through various hacks and plug-ins proposed by web developers. The majority of modern rich and interactive web designs was based on Adobe Flash technology which was supported by all major browser vendors. The Flash plug-in 280 Jakus, Jekovec, Tomažič, Sodnik offered an excellent support for multimedia content, especially animations and animated interfaces. HTML5 simplifies the implementation of such functionality through native browser support. The new notable trend on the Web today is the introduction of semantics in web documents. The web content is shaped and designed primarily to be read and understood by people; therefore, a computer cannot provide any extensive help by analyzing, searching and processing the data. The introduction of semantics will eventually lead to the third generation of Web, the so-called Semantic Web. New development practices, rich web content and the need for semantics in web documents are already manifesting themselves in practice. Besides some changes in HTML syntax and vocabulary, the most important new features in HTML5 are the introduction of semantics in the form of microdata and ARIA attributes, support for RIA by bringing new form widgets, support for multimedia and dynamic graphic rendering.
Being server-side JavaScript, another admirable edge of Node.js over others is that a developer will be required to only have knowledge and experience of a single language i.e. JavaScript, no matter if he is developing client side scripts or scripts for server side. The developer is not required to swap his brain cycles from for one language at client side and then for another

language at server side. Here the database of JavaScript also store date like a JavaScript Object. Adding to this, it is also worth considering that Node.js is new and thus have benefit of taking precautions against the mistakes which other languages had come across in the past such as the mistake of backward compatibility. About 47% of web surfers wants a website to be loaded within 2 seconds and a 3 second delay drops the customer satisfaction by 16%. Here the Node.js leads as the interpreter of Node.js is smaller and fast compared to other languages like PHP. Here the server side apps are permanently kept ON unlike other languages where every initiation of the application will follow cycles consuming steps of for example loading configurations, followed database connectivity, acquire required information and finally render the markup language. Node.js on other hand reduces these steps by keeping an app permanently ON.



Galgotias University

Project Portal

Student Login

Student

Login

Faculty Login

Faculty

Login

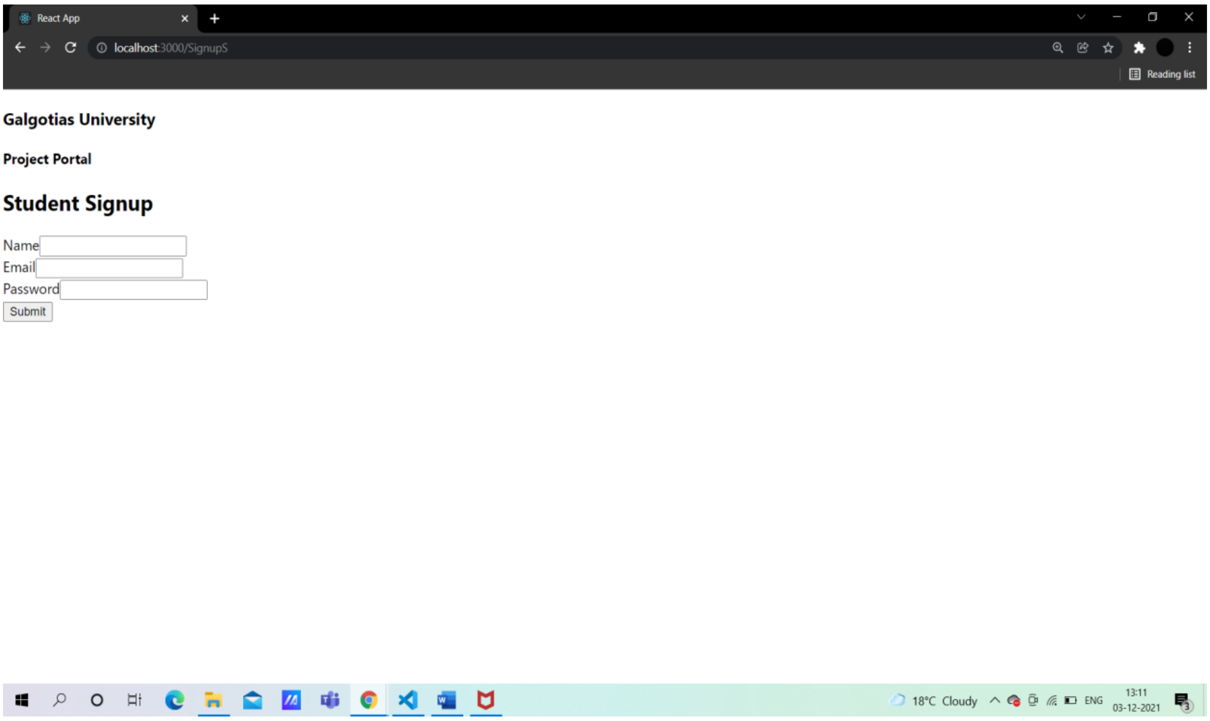
Not yet registered?

Please click here to Signup

Not yet registered?

Please click here to Signup

Figure 22: Home Portal



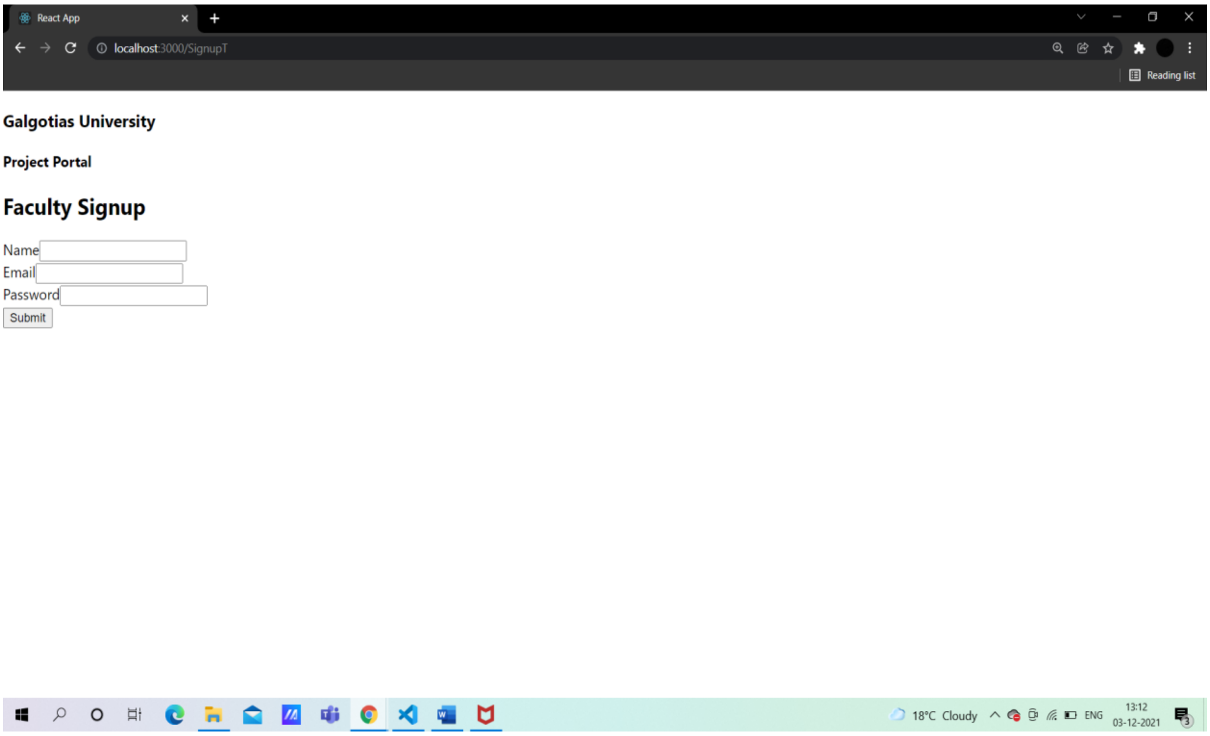
Galgotias University

Project Portal

Student Signup

Name
Email
Password

Figure 23: Student Signup Page



Galgotias University

Project Portal

Faculty Signup

Name
Email
Password

Figure 24: Faculty Signup Page

CHAPTER 6: CONCLUSION AND FUTURE SCOPE

The way people access the Web is changing rapidly. The Web is increasingly used in smartphones, tablets, TVs and other devices. Therefore, it is crucial the adoption of a new approach of creating Web pages and making them able to adapt and respond to any device. Responsive design is characterized by ensuring a dynamic adaptation of the contents to the display size, thus ensuring a good viewing experience regardless of the Web access device. Several advantages are associated with responsive design such as high flexibility, a good user experience, low development and maintenance time & costs, improved SEO, a lower bounce rate, increased productivity, and increased accessibility.

However, responsive design is not exempt from criticism, and also presents a set of limitations, respectively in the adaptation of contents, difficulties in optimizing user experience, higher loading time, not suitable for complex projects, low availability of CSS media queries and issues in browser compatibility with older versions. The performed study turns out that software developers emphasize the guarantee of a good user experience and the increase of accessibility as the two most important advantages of responsive design. Additionally, there is greater disagreement in considering time and maintenance costs as an advantage of responsive design. Regarding the limitations, they emphasize the compatibility with older versions of a Web browser, the higher loading time and the difficulties in optimizing user experience. Finally, it should be mentioned that the perception of the advantages and limitations of responsive design is slightly different for professionals with greater years of experience and for freelancers. The professionals with greater professional experience in the sector and in the field of responsive design give less importance to the benefits in terms of the time and costs of development and maintenance, besides mentioning greater difficulties in the process of adaptation of contents; freelancers have the opposite behavior as they give greater importance to maintenance development costs and times, as well as being less sensitive to limitations in the loading time of a website and in compatibility with older web browsers.

The study got the finding about the implementation of Node.js. The Node.js have made Full Stack Developer’s job a dream come true. In absence of Node.js it was hard for a developer to learn several different languages and environments to manage the complete system at server side and client side. Organizations and developers can now with the invent of Node.js build highly load bearable and faster applications and by using Single Page Applications (SPA) now the server calls are reduced and the applications are more user friendly and faster. Node.js made it easy to achieve high load operations like graphic processing and Internet GIS very quicker, and it can be reliably used in every field where the files sizes are high or the network bandwidth is highly consumed. Node.js will make such operations faster and with less need of bandwidth. Community like its feature that the same language is also being used at server side while JavaScript is always been used at client side for ages. Node.js have some challenges in context of its use in the community as well as its adoption by the developers and organizations over the existing programming technologies. No doubt that Node.js have great benefits, it also have some challenges to the community. One such challenge is the ability of misuse of the widely used language by developers. One enthusiast has made a backdoor software using Node.js on Raspberry Zero. It can create backdoors in the target computer and their network even if the computer is password locked. Although there are solutions from such backdoors but some seems impractical like totally blocking the USB ports, and closing the web browser every time the user leaves the computer. And other options are not implemented by majority and mostly might not be aware of it like using secure layer on ones websites (https), and enabling secure flags on the cookies which common users might not know about it. There is a plus point but as understood from the survey conducted that the community feel it hard to learn JavaScript for Node.js. Also the developers having knowledge of other programming languages have complications in adopting Node.js. Even the setting up of server for their programming work is not an obstacle. This is as concluded from the survey results. Another plus points were event-Driven Programming, Non-Blocking I/O, and asynchronous feature. But according the survey results is that the features like event-driven programming, Non-Blocking I/O, and Asynchronous processing is a hindrance.

As a result of the survey, a challenge comes to front is that most organizations are not ready to adopt the new technologies like Node.js over their existing ones like PHP, .Net, etc. Also, there is a lack of market awareness which is causing a barrier to adopt Node.js for implementation. At a developer level, there is a challenge which is seen from the survey results that they are not feeling it easy to learn the database working and using of the JavaScript environment. And there also seems a lack of enough knowledge among the developers as from the survey results a reasonable respondents to the questions on the important features like event-driven, non-blocking I/O and asynchronous processing is making the decision about this. The community can be taken to the Node.js by little efforts of trainings, and conducting workshops to introduce

Node.js to new and existing programmers and explain to them the benefits of Node.js features like event-driven programming, non-blocking I/O, and asynchronous processing. The community need to be updated about the features of how Node.js is faster in performing network related tasks specially when it come to the challenge of handling large files over the network and handling multiple callbacks in the other languages. Challenges in Implementation like PHP, .NET, etc, causing the server to overload or require more memory. Also, organization should be briefed about the benefits of hiring Full Stack Developers based on Node.js and how they will cut costs related to server bandwidths and developers hiring and speed of the applications built over Node.js.

We are looking forward to build this application on the android platform due to its huge user-base. The major challenge for us would be to gain trust of the both students as well as teachers and get them to join this site for their use. We will target the audience of all students of a particular branch first and then forwarding with the students and teachers of different schools. We will set up the foundations by just adding a few numbers of student of just a class first. The web app would be economically feasible. There are a lot of things to be done in future. Firstly, we can implement algorithms to get a better hold of our users and that can benefit our users as well as they can save their work on the site too and can start from the point where one left. We would also try to add more types of submissions possible. Word of mouth would be the first advertisement option and then followed by social media promotions. Secondly, we can extend the website by providing the structure to other colleges too so they can get benefitted by this. Large database system to store all the details of the project would also be used.

REFERNCE

1. Research paper on “Front-end Frameworks and Libraries in E-Business Development” by YongKang Xing, J iaPeng Huang and YongYAo Lai from Nanfang College of Sun Yat-sen University, China in 11th International Conference on Computer and Automation Engineering, 2019.
2. “ReactJS: An Open Source JavaScript Library for Front-end Development” by Naimul Naim in Metropolia University of Applied Science (2017) IT Thesis.
3. Research Paper on “REACT.JS and Front End Development” by Ritwik C and Anitha Sandeep, Bangalore, in International Research Journal of Engineering and Technology (IJRET), e-ISSN: 2395-0056, p-ISSN: 2395-0072. Volume 7, Issue 4. Ver. I (April 2020)
4. “Comparative Study of ReactJS and NodeJS” by Gayathri G, Maharani’s Science College for Women, Bangalore, in IJRECE e-ISSN: 2348-2281, p-ISSN: 2393-9028. Volume 7, Issue 1. Ver. I (January-March 2019)
5. ReactJS Wikipedia. https://en.wikipedia.org/wiki/React_(JavaScript_library)
6. Angular Wikipedia. https://en.wikipedia.org/wiki/Angular_(web_framework)
7. EmberJS. https://www.tutorialspoint.com/emberjs/emberjs_overview.htm
8. NodeJS Wikipedia. https://en.wikipedia.org/wiki/Node.js
9. VeuJS Wikipedia. https://en.wikipedia.org/wiki/Vue.js
10. “JavaScript Frameworks Vs Libraries-What’s the Difference?”. Difference. https://skillcrush.com/blog/javascript-frameworks-vs-libraries/