# Project Assessment System For ABET

Graduation Project Report

CMSE 406

Team members

Muhammad Ryan Farooqi, 15701793

Mustafa Murataga, 15330092

Ali Küçüker, 16450022

Supervisor: Assoc.Prof.Dr.Adnan Acan

Computer Engineering Department

Eastern Mediterranean University

Term: Fall 2020-2021

## ABSTRACT

The aim of this project is  to create a web based system to assess Projects for ABET accreditation. **ABET** requires that programs show student achievement and certain course outcomes. Hence this system that has been developed is a **Node.js** based application which was created in helps of various tools and commands including latest forms and variations of database handling techniques using **MongoDB** and its clutch peripheral **Mongoose** in it where the correlation between client, server and the database are withheld within the series of intricate designing and sequencing between merging multiple Data languages **i.e : Node.js, HTML, CSS, JavaScripting, React.js, MongoDB** in addition to recommended IDE platforms such as **Atom** and Debian Based Terminal modulation using **Hyper Terminal.** Also for Front-End setup **Express.js** was preferred.

In essence this framework is a firsthand test prototype that has in convenience to previous alterations of its predecessors that has been the most until recent times as one of the most User-friendly and efficient model building tools in **web-development**.

**Keywords:** ABET, COURSE OUTCOMES, WEB BASED SYSTEM, PROJECT AND PLANNING, DATABASE, REQUIREMENTS ANALYSIS, DESIGN

## Table of Contents

LIST OF FIGURES

LIST OF TABLES

# 1. INTRODUCTION

ABET authorize colleges fulfill certain guidelines. Thus the ABET organization requires an appraisal report to guarantee that these principles are being met. Nonetheless, the age of the reports is no simple accomplishment, a segment in the report to decide a bunch of conceivable course results. The aim of creating this framework is for Admins[Teachers] assign a particular curriculum vise based Project assignments that can be viewed by non-root users[Students] who can view the files and return them with the Project solutions, This in terms of DataBase Managements systems created and response and trace backing from the Front-end and Back-end Peripherals which have been logistically analyzed and lay out has been made as decipherable as possible for 3rd parties to acknowledge and verify the integrity of the system.

This web-based application was made using the conventional tools needed for developing a web application, But in furthering the process of the creativity the tools were also complied with the latest backend development techniques including the frontend, The goal is to create a framework that can help the Teachers and Students bridge a formidable flow to Data Base circulation of Project files that can be with ease accessed and evaluation based on prior specified criteria by the Administration to continue logging all the necessary academic approaches in accessibility.

# 2. PROJECT PLANNING AND MANAGEMENT

### 2.1 Project Definition:

We are making an electronic application that handles ABET evaluation of Projects offered in an academic session. The application should be able to create a Data Base for users where the Admin and Non-Admin roles can be assigned and also the Admin can Add or Remove other users as well as create tasks for the existing users to view.

### 2.2 Aim of Project:

Time efficiency in coordinating an organized Project assigning platform.

### 2.3 Target Users:

Our objective clients are Professors, Asst. Educators, Lecturers, Students and if all works out in a good way, different Universities.

### 2.4 Scope of Project:

1. <u>Feasibility and Pre-research</u>**:** In this stage, we complete examination, recognizing the prerequisites, partners, comparable applications,tools to utilize, restrictions, requirements, and so on We assemble all data that will give us better understanding concerning how to approach building up the application. In the event that there existing frameworks that are like our rendition, we dissect it and perceive how to grow it to accommodate our own.

2. <u>System Design</u>**:** At this stage, we settle on choices concerning the structure of the framework. We are utilizing the Iterative and Incremental advancement techniques. This would permit us to separate the advancement cycle into more modest assignments and actualize and test consistently. We picked this strategy since it creates working programming quicker and is more adaptable and less expensive to change necessities. Additionally we search for suitable strategies and calculations, information stream outlines, information base ER plan and so on to utilize.

3. <u>Software development</u>**:** We will start to create (compose code) the framework at this stage. The front-end will undoubtedly be executed in HTML, CSS, and JavaScript dialects utilizing Angular Framework Along with Node.js. The back-end will be composed utilizing Mongoose Framework. The MongoDB information base where information can be put away and controlled by the framework continuously would likewise be executed in this stage.

4. <u>Prototype implementation and testing work</u>**:** By each indent of the solution for the framework we further our time-execution by testing each package based on individual functionality of the code snippets.

5. <u>Maintenance</u>**:** We will go over the program and utilize the outcomes from

the usage stage to fix any current issues. There would likewise be intensive documentation which would make troubleshooting simpler.

**2.5 Required Resources:**

| | Instrument/Equipment/<br><br>Software | Purpose | Unit Price(TL) |
|---|---|---|---|
| 1 | Laptop | To write the program | 2500 |
| 2 | Internet connection | Research, download files, etc. | No Cost |
| 3 | App Creately. | Creating Use-Case Diagram | No Cost |
| 4 | Microsoft Office | Documentation | No Cost |
| 5 | ZenFlow Chart | Organizational Assignments | No Cost |
| 6 | Mongoose and MongoDB | DataBase Programming and Management | |
| 7 | Atom | IDE | No Cost |
| 8 | Hyper Terminal | For server and run time analysis | No Cost |
| | | | |
| 9 | Go.gliffy.com<br><br>Canva.com | Creating high-level system architecture diagram<br><br>Gannt Chart | No Cost |

|  |  |  |  |  |
|---|---|---|---|---|
| 10 | DBSchema | Relation Diagram | No Cost |  |
| 11 |  |  | No Cost |  |

Table 1 - Required resources

### 2.6 Gantt chart:

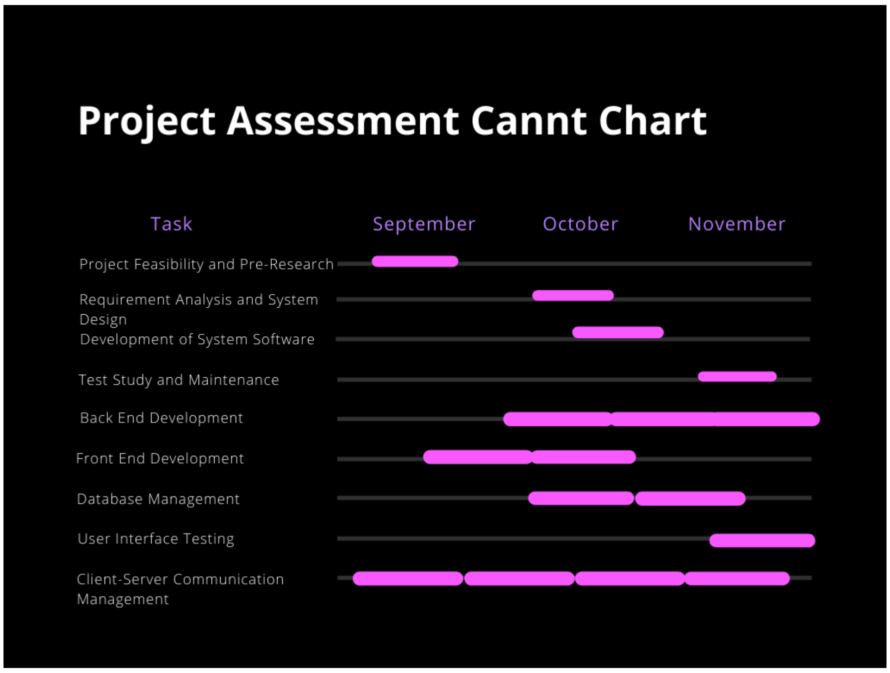


Figure 1. Gantt chart

### 2.7 Project Work Packages:

| | |
|---|---|
| **Work Package No** | 1 |
| **Work Package Name** | **Project Feasibility and Pre-Research** |
| **Start-End Date and Time** | 1th Sept – 20nd Sept, 2020 |

**1- Activities of work package:**

**1.1 Project Process and Economic Feasibility:**

- Analyze scope of the system

- Accumulate resources

- Strategize in Development risks & Deployment

- Examine creation costs (to keep a monetarily reasonable level all through cycle)

**1.2 Technological Feasibility:**

- Characterize System Operational Environment and Operating System Constraints (Cross-Compatibility).

**2- Methods and parameters that will be used for work package:**

- Examination of system creation methodology and factors that bound the system during demonstration will be carried out by the developer where the front-end and back-end frame work will be simultaneously be run from the perspective of a Teacher and Student and negligence of Admin control shall at all costs not be underestimated

**3- Experiments, tests and analysis in the work package:**

- Research using Internet Facilities
- Evaluate similar or pre-existing systems framwork
- Cost analyses
- Resources and Time-Constraints

**4- Output of work package and its success criteria:**

**Outputs:**

- Data and involvement with the activities of existing Project appraisal frameworks
- Deciding how doable the undertaking is, and anticipating the dangers and expenses
- Obligations division and work portion set up

**Success Criteria:**

- Better handle on the timetable, time limitations and spending imperatives
- Overall standard picture of the general viewpoint of the framework- Proposal Acceptance

**5- Relation of output with other work packages:**

This work bundle is set up to be the "establishment" on which most of the other work bundles would be mounted on. Data assembled in this bundle would be defended in the realization of other work bundles and the venture outlook.

Table 2 - Work Package 1

|  |  |
|---|---|
| **Work Package No** | 2 |
| **Work Package Name** | **Requirement Analysis and System Design** |
| **Start-End Date and Time** | 13th Oct – 16th Oct, 2020 |
| **1- Activities of work package:** |  |
| - Frame Requirements Examination<br>- Requirement Document Production<br>- Allocation of Resources<br>- Architecture of Software(UI, Modeling, Metadata, Backend) |  |
| **2- Methods and parameters that will be used for work package:** |  |
| - Chart Implementation<br>- Understand of Merging and routing front-end & backend communication<br>- Client-Server programming and time management<br>- Solo effort to give the developer a clear view and control over the system |  |
| **3- Experiments, tests and analysis in the work package:** |  |
| - Make necessities records reflecting consequences of data gathering measure |  |

- Prototype production
- Design and Function correlation
- Understand and Manage Resources
- Portrayal of reliance of framework capacities and abilities.

**4- Output of work package and its success criteria:**

**Outputs:**

- Requirements Analysis document
- Framework Scope extended to System Structure
- Prototype fidelity [low]
- Architecture and Model

**Success Criteria:**

- Reduction in Costs
- Structure validation

**5- Relation of output with other work packages:**

Broadening the effective fulfillment and progression from the first Work Package, this bundle affirms of the degree of nature of the pre-research measure indicating how the gathered data can be introduced in an implementable arrangement. The particulars and necessities data would be utilized to guarantee adherence and backing progress towards

| |
|---|
| the ideal usefulness of the completed framework. |

Table 3 - Work Package 2

| | |
|---|---|
| **Work Package No** | 3 |
| **Work Package Name** | **Development of System Software** |
| **Start-End Date and Time** | 20th Oct – 30th Oct, 2020 |
| **1- Activities of work package:** | |
| - UI Implementation<br>- Implementation of Database<br>- Framework functionalities implementation | |
| **2-Methods and parameters that will be used for work package:** | |
| - Relations between Databases using ER-Diagram<br>- Node.js for backend programming<br>- Frontend creation using HTML, CSS with the help of material-UI<br>- Model for client-server communication<br>- Authentication phase[API] | |

| **3- Experiments, tests and analysis in the work package:** |
|---|
| - Review of Functional instances<br>- Identification of modular design<br>- Programming of Code and Database<br>- Run-Time testing prototype |
| **4- Output of work package and its success criteria:** |
| **Outputs:**<br>- Sample application execution<br>- Code run-time analyses completion<br><br>**Success Criteria:**<br>- Least possibility outcome of errors<br>- Efficient and Effective design |
| **5- Relation of output with other work packages:** |
| This bundle finishes up the coordination of framework functionalities with the interface. The principal model would be created at this stage. Testing of the application can start |

| |
|---|
| not long after the accomplishment of this work bundle. |

Table 4 - Work Package 3

| | |
|---|---|
| **Work Package No** | 4 |
| **Work Package Name** | **Test Study and Maintenance** |
| **Start-End Date and Time** | 5th Nov – 16th Nov, 2020 |
| **1- Activities of work package:** | |
| - Part testing<br>- Integration<br>- User Acceptance validation | |
| **2-Methods and parameters that will be used for work package:** | |
| - Data Validation<br>- Ease of Use testing<br>- Integration & Testing of the frame work<br>- Validation and Performance check | |

| |
|---|
| - Test run for Maintenance |
| **3- Experiments, tests and analysis in the work package:** |
| Unit test will be finished and revising problems in modules and individual capacities. Mix test will test the correspondence between modules. Client Acceptance test will be applied to make the UI more natural and the entire projects more reliable. |
| **4- Output of work package and its success criteria:** |
| **Outputs:** - Report for Tests - Documentation of release **Success Criteria:** - The frame work is up and running and ready for launch |
| **5- Relation of output with other work packages:** |
| This work bundle is the last one. Thus, if this work bundle's means are totally finished, our venture will be done and fit to be delivered. |

Table 5 - Work Package 4

## 2.8 Estimations

**Basic COCOMO Calculations**

***Using LOCS* = 1,103**

**Using:**

| BASIC | a | b | c | d |
|---|---|---|---|---|
| Organic | 2.4 | 1.05 | 2.5 | 0.38 |
| Semi-detached | 3.0 | 1.12 | 2.5 | 0.35 |
| Embedded | 3.6 | 1.20 | 2.5 | 0.32 |

***semi-detached***

**Effort, $E = a * (KLOC)^{b}$**

**$E = 4.0 * (1.103)^{1.12}$**

**$= 4.46$**

-

**Duration, $D = c * (E)^{d}$**

**$D = 2.5 * (4.46)^{0.35}$**

**$= 4.22$**

-

**Person = E/D**

**Person = 4.46/4.22**

**= 1.06 person**

## 2.9 PROGRAM EVALUATION AND REVIEW TECHNIQUE (PERT):

| Task ID | Task Name | Duration (days) | Dependency |
|---|---|---|---|
| **A** | Project Initialization/ Feasibility Studies | 7 | |
| **B** | Requirements Analysis and Development | 9 | A |
| **C** | Resources Procurement/Allocation | 6 | B |
| **D** | System Design and Modeling | 10 | B |
| **E** | Coding and Implementation of Functionalities | 35 | C, D |
| **F** | Testing Activities and Modifications | 27 | E |
| **G** | Project Closure | 1 | F |

Table 6 - Activity Table

| Paths | Duration |
|---|---|
| **A B D E F G** | **89** |
| A B C E F G | 85 |

Table 7 - Path table

Critical Path: ABDEFG

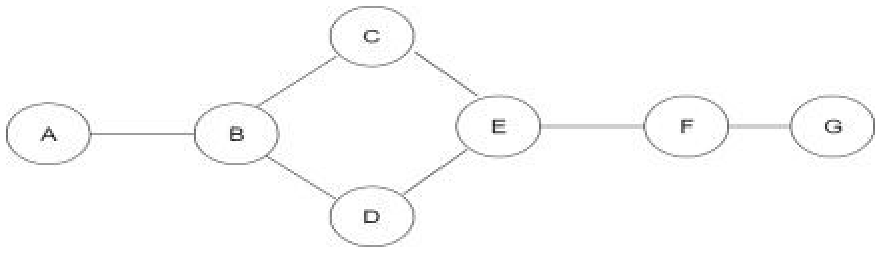


Figure 2 - Network Diagram

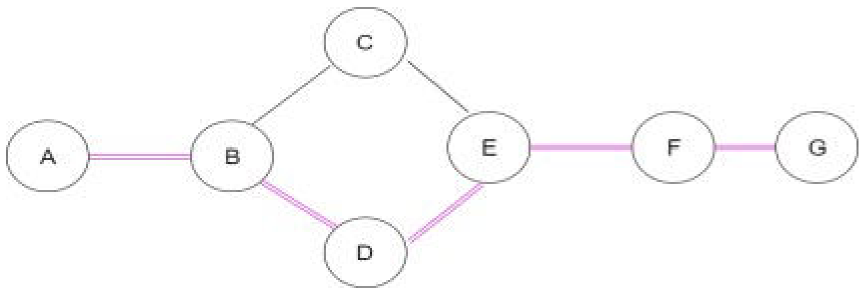


Figure 3 - Network Diagram with Critical Path

## 2.10 Organization Scheme:

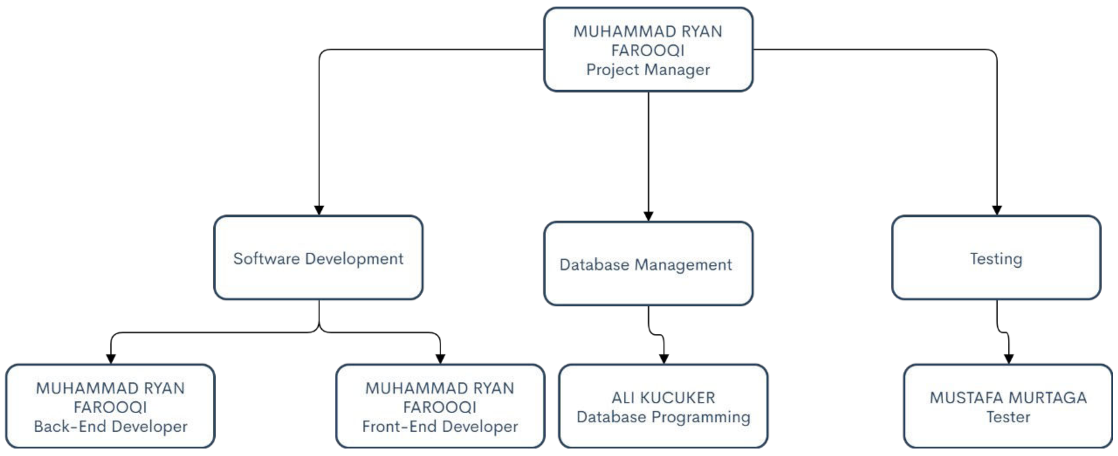

MUHAMMAD RYAN FAROOQI Project Manager
Software Development
Database Management
Testing
MUHAMMAD RYAN FAROOQI Back-End Developer
MUHAMMAD RYAN FAROOQI Front-End Developer
ALI KUCUKER Database Programming
MUSTAFA MURTAGA Tester

Figure 4 - Organizational Role Flowchart

## 2.11 Risk Analysis Table:

| **Risk** | **Probability** | **Effects** | **Strategy** |
|---|---|---|---|
| **Lack of resources** | low | tolerable | System should be back checked once in a while |
| **Developer may not be skilled in some parts of the procedure** | High | Serious | Spend more time from foreign resources to learn new or required set of skills |
| **Poor communication** | Low | Tolerable | Ease of widespread of information |
| **Requirements for the project might change** | Moderate | Tolerable | Check how the progressions influence the spending plan and timetable of the undertaking and roll out fitting improvements to the spending plan and cutoff time of the venture. |
| **Technological components aren't interoperable** | Moderate | Tolerable | Use multiple snippets to ensure smooth run-time comparisons |
| **Key member(s) leaves in the middle of project** | High | Serious | Re-organize strategic output facilities to enable the developer to execute steps based on a strict time frame |
| **The size of software is underestimated** | Moderate | tolerable | Use methods of file compression techniques. (.rar,7z) |
| **The database used in the** | low | tolerable | Double check input |

| | | | |
|---|---|---|---|
| **system cannot process as many transactions per second as expected.** | | | schematics |

Table 8 - Risk Analysis Table

## 3. REQUIREMENTS ANALYSIS

### 3.1 Functional Requirements

Our framework will be utilized by the instructors and students  of the institutions that have ABET accreditation.

1. Login feature availability
2. Frame work should allow Teacher/Admin to create and assign students to Projects
3. Ability to grade and review student projects
4. The system should be able to distinguish roles between Teacher and Students.
5. Option for Uploading and Download document files
6. Simple and clean UI interface

**Use-Case diagram:**

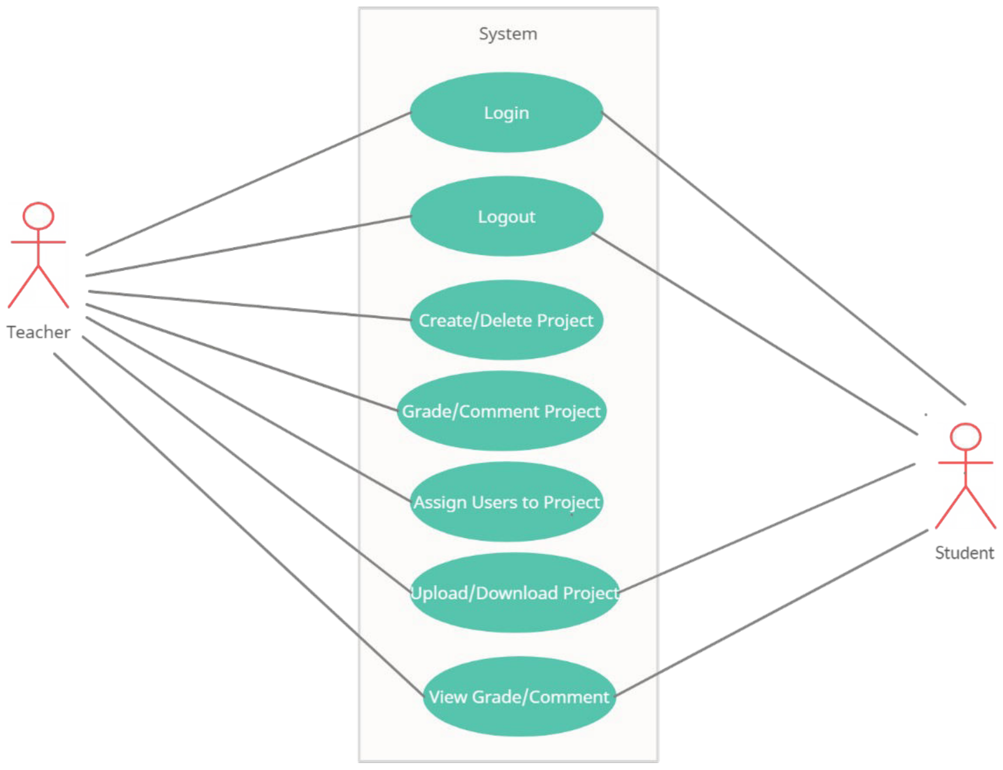


Figure 5 - Use-Case diagram

### 3.2 Non-Functional Requirements

The non-practical necessities are additionally called quality ascribes of the product being worked on. They depict how the framework should function. It is additionally isolated into unwavering quality, accessibility, security, practicality, compactness and so on as the attributes of the product.

**Reliability**

It implies the degree to which program performs with required accuracy. The site created ought to be very dependable and secure so data about any inquiries and so on isn't leaked. The framework will not be down multiple occasions in a year.

**Availability**

The product will be accessible just to approved clients like understudies to see their enlisted course, and administrator to add an update/erase understudy subtleties. Watching that the framework consistently has something to work and in every case spring up error messages in the event of segment disapproval. All things considered the blunder messages show up when something turns out badly to win accessibility issues.

**Security**

The security prerequisites manage principally security. The product ought to be dealt with simply by the chairman and approved clients. Just the manager has the option to appoint authorizations like making new records and creating passwords.

**Maintainability**

The application is to be planned so it is effectively kept up. Likewise it ought to permit fusing new necessities in any module of the framework. Reinforcements for information bases are accessible.

**Portability**

The product is an electronic application and is implied with MongoDB and Mongoose. So it is stage free and is autonomous of OS. The application will be effectively versatile on any window based framework.

## Other Non-Functional Requirements

1. System up-time should be 24/7.
2. Compatibility among major and numerous browser plugins.
3. Simple UI.
4. Secure and Safe.
5. Portability and Ease of OS migration
6. Protection of data, the fare of confined advancements, licensed innovation rights, and so on ought to be evaluated.

## 3.3 Realistic constraints

The page will require a PC with web access. It will be allowed to utilize. It won't need any additional instrument. The highlights to be resolved in the framework will be completely as per ABET accreditation. The codes will be composed by us. There will be help areas and will discuss them and discussion about the creator in the references segment to maintain a strategic distance from moral limitations and copyright issues.

### 3.4 Ethical issues

In Accordance with ABET requirements there won’t be any ethical/legal issues.

## 4. DESIGN

### 4.1 High level design (architectural)

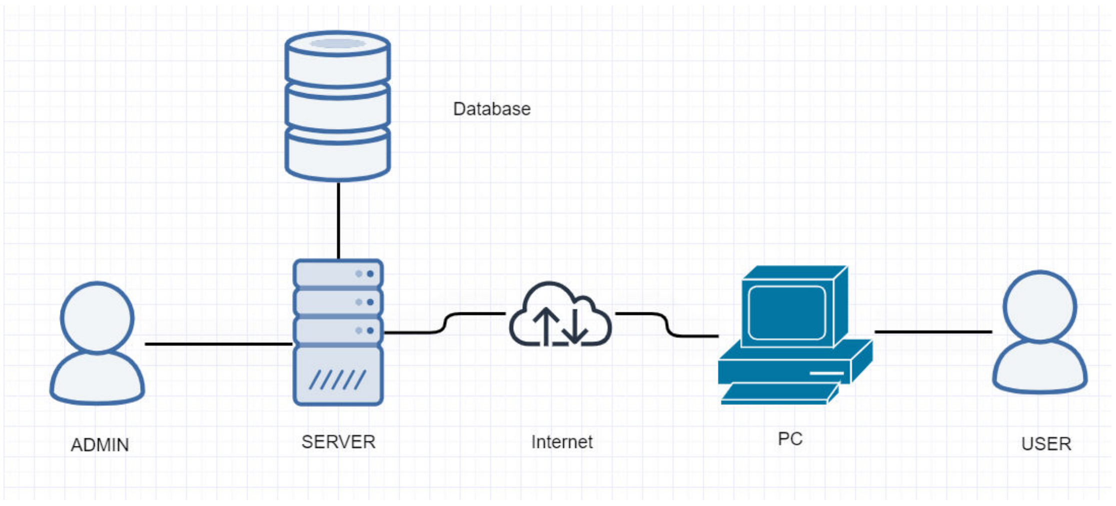


Figure 6: High Level Architecture of the System

### 4.2 Software design

**Backend(Server)**

**Modular  Hierarchy Diagram:**

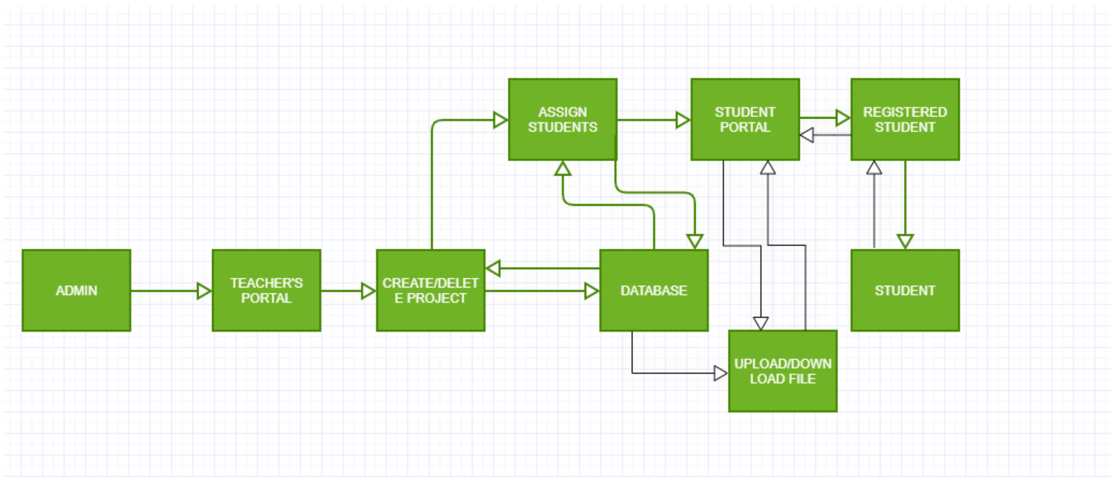


Figure 7: Server Modular Hierarchy Diagram

**DB Module:**

This module manages information base tasks, for example, peruse and compose information. All the information going to information base and all the information coming from the data set should experience this module. The submodules program, course, client and understudy manages information base tasks for explicit role in the framework, for example, understudy or potentially educator.

**Controller:**

This module manages the interior logic outlook for a request to the server. This is the place where the actual logic from the request is laid. For instance, saving a document, mentioning information from DB or ascertaining something about the information recovered from DB. The submodules manage request sent for specific system roles.

**Repository:**

This module is a reflection layer between the information base and the genuine application itself. For instance we have a clients store that has explicit capacities to make our work simpler when we need to peruse/keep in touch with the information base.

**Routing:**

______This module deals with routing the incoming request to the right controller and constructing the API layer. The submodules deal with specific data routing.

**Data Flow Diagram:**

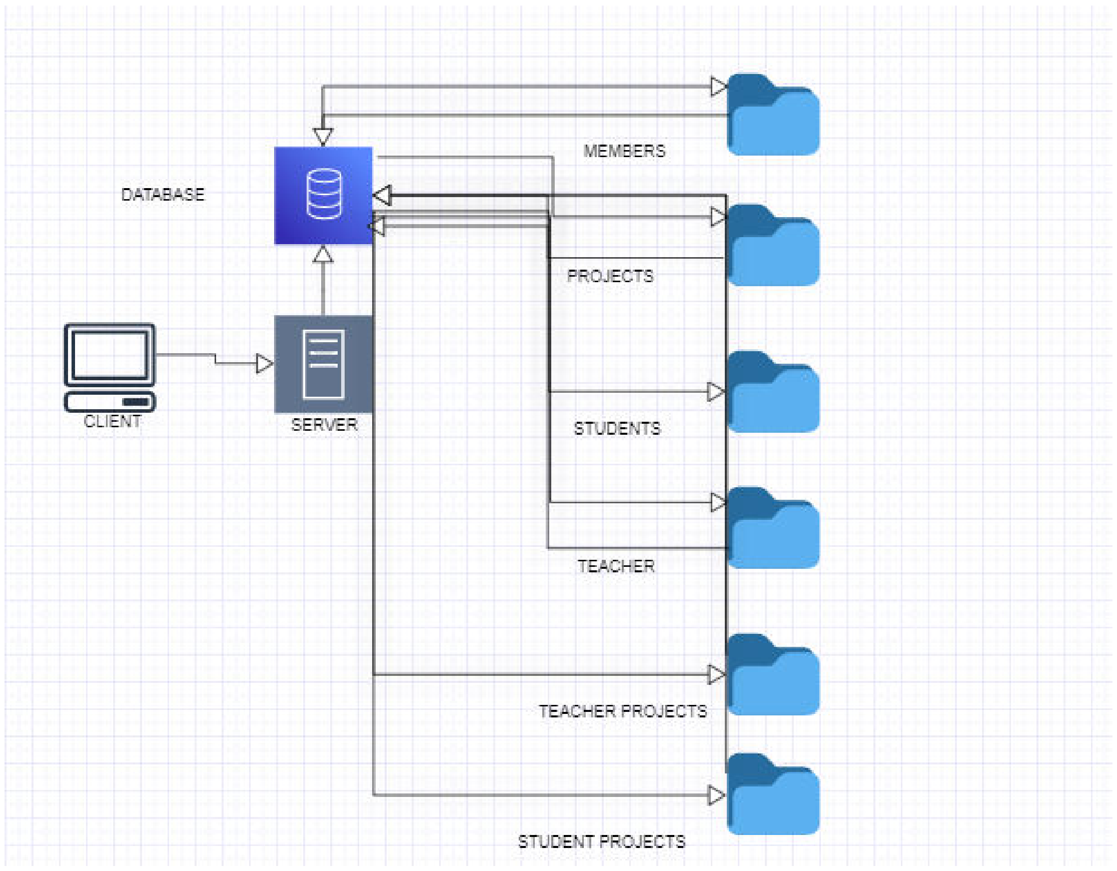


Figure 8: Server Data Flow Diagram

### Client(Frontend)

**Modular  Hierarchy Diagram:**

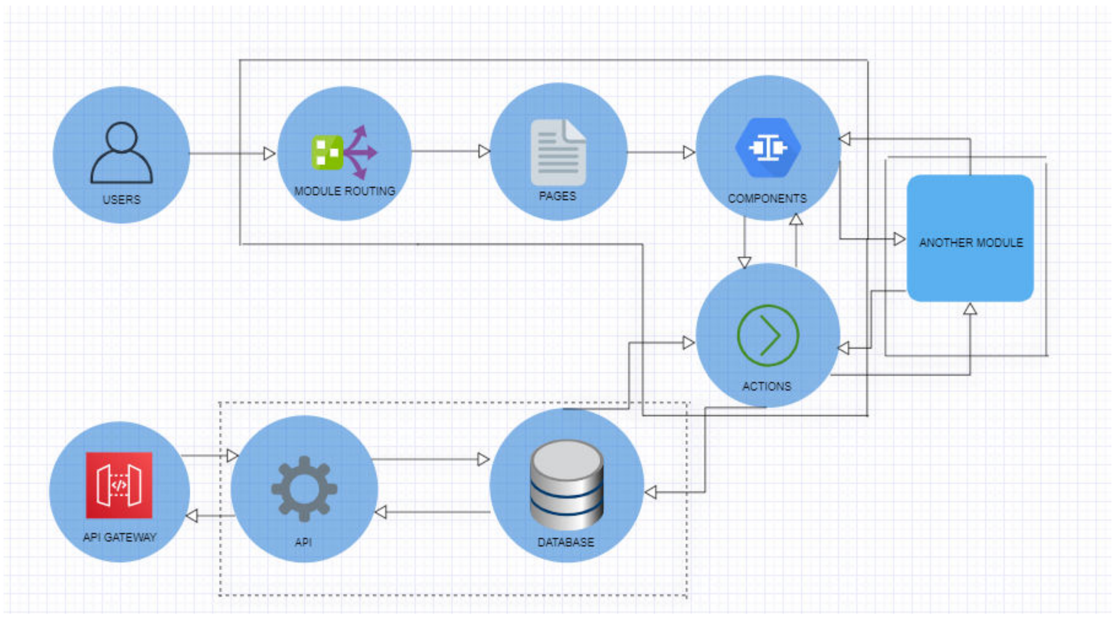


Figure 9: Client Modular hierarchy Diagram

**API Module:**

This module manages directing the approaching path to the correct regulator and building the API layer. The submodules manage data routings.

**Outline:**

The module that manages making a structure to produce a blueprint archive of a course highlight of the framework. a given role in the framework.

**Grade Sheet:**

The module that manages creating a grade (or structure manufacturer) by considering all inquiries posed (with its alternatives if accessible) in a test highlight of the framework.

**Outcomes:**

The module that manages making a structure for showing "the assessment and translation obviously learning results and commitment to understudy results" highlight of the program.

**<u>Comment:</u>**

_______The module that deals with the data input form for coment feature of the program.

**Data Flow Diagram:**

Figure 10: Client Data Flow Diagram

### Database

#### E-R Diagram

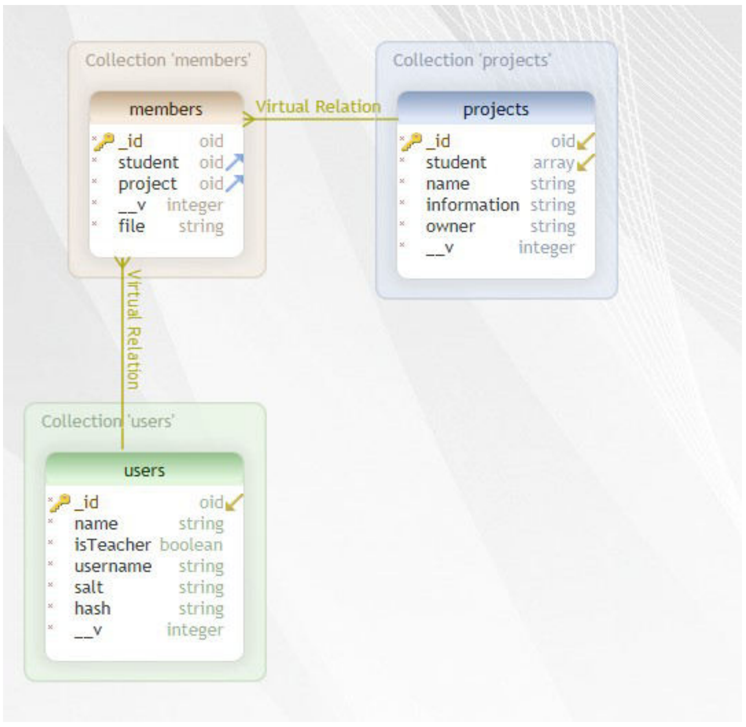


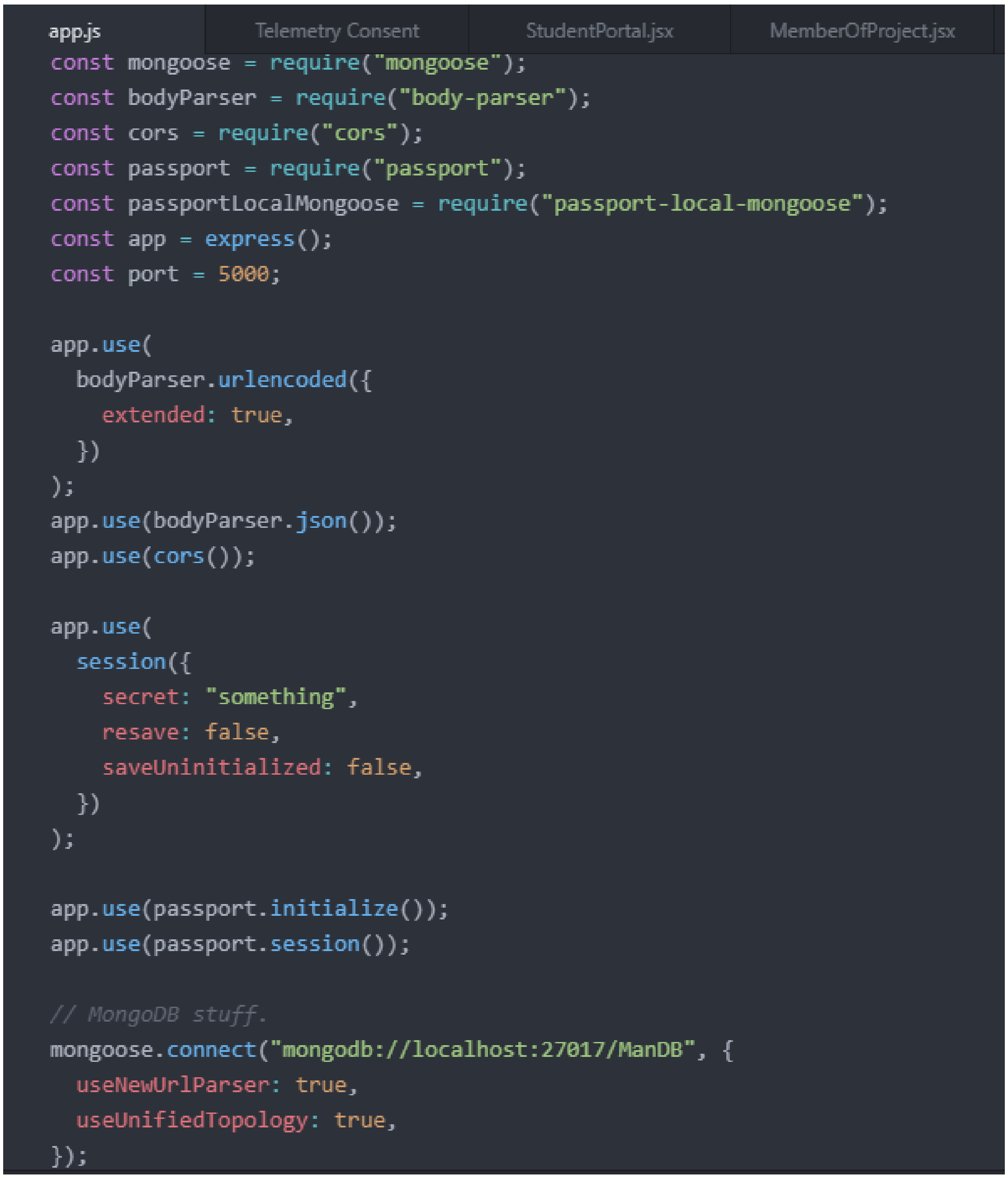
Figure 11 - E-R Diagram

# 5. IMPLEMENTATION

As we can find in the Figure 6(High Level Architecture of the System), we have utilized client-server engineering and fully isolated the client and server to decrease the complex nature of the framework and simplicity of the board of the codebase.

## Server Implementation:

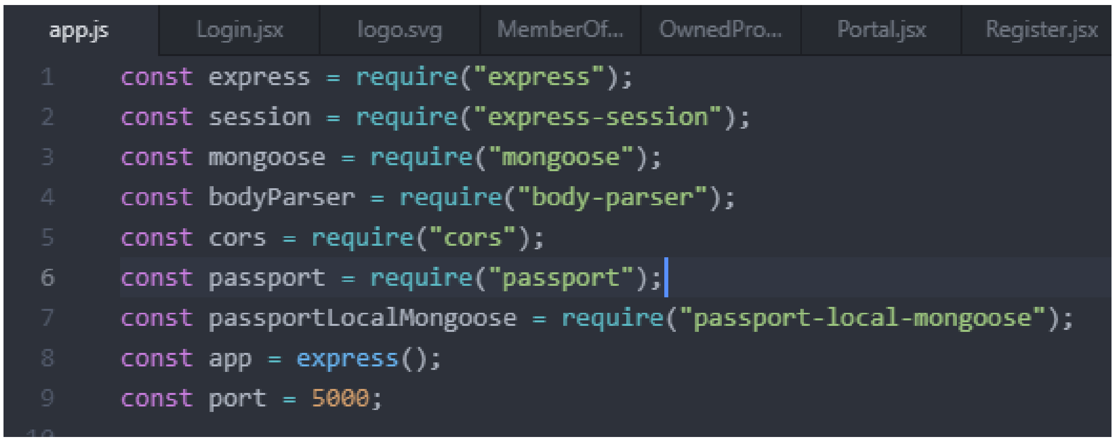
Server is an application that runs always and tunes in to the approaching data and does the fundamental jobs(updating of information base,  important counts and so on) for that demand and sends a reaction back to the client.. In our usage of the server, we have utilized NodeJS Framework and we have utilized MongoDB for our information base framework. Joining the two, we can make a truly straightforward, solid and light-weigth server application. Here is our app.js principle document that associates with the information base and starts the server as shown in the Figure 12 underneath.



```
const mongoose = require("mongoose");
const bodyParser = require("body-parser");
const cors = require("cors");
const passport = require("passport");
const passportLocalMongoose = require("passport-local-mongoose");
const app = express();
const port = 5000;

app.use(
  bodyParser.urlencoded({
    extended: true,
  })
);
app.use(bodyParser.json());
app.use(cors());

app.use(
  session({
    secret: "something",
    resave: false,
    saveUninitialized: false,
  })
);

app.use(passport.initialize());
app.use(passport.session());

// MongoDB stuff.
mongoose.connect("mongodb://localhost:27017/ManDB", {
  useNewUrlParser: true,
  useUnifiedTopology: true,
});
```

Figure 12: app.js

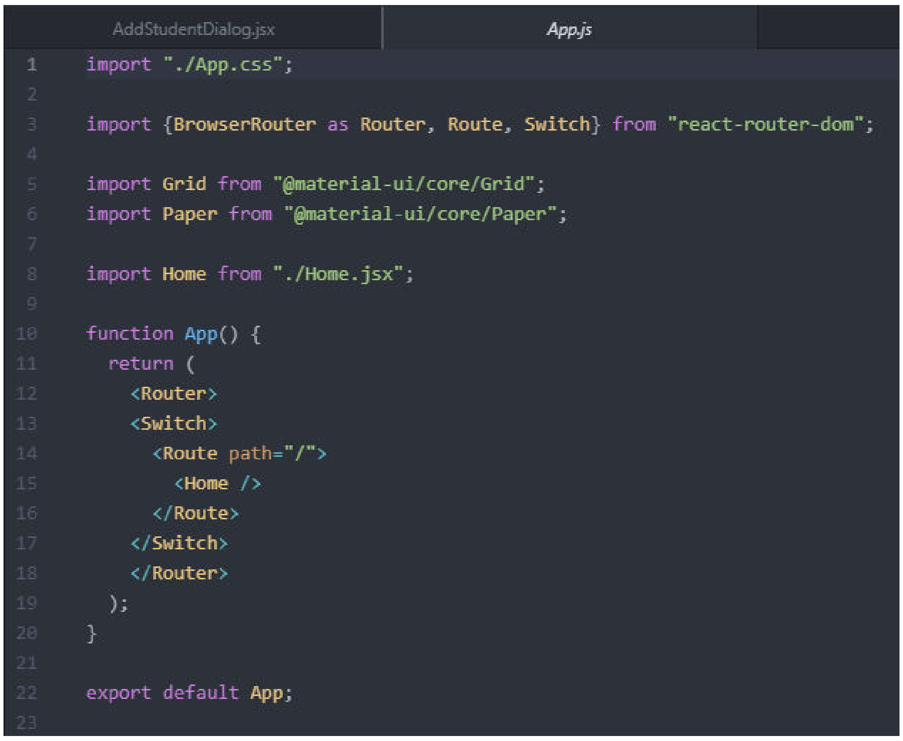
In Figure 12, we have our routes(API endpoints) that we acknowledge approaching data from. We can take a look at the  record in the route organizer in the Figure 13 beneath.



```javascript
const express = require("express");
const session = require("express-session");
const mongoose = require("mongoose");
const bodyParser = require("body-parser");
const cors = require("cors");
const passport = require("passport");
const passportLocalMongoose = require("passport-local-mongoose");
const app = express();
const port = 5000;
```

Figure 13: app.js

In figure 14 below we can undermine that upon a certain action performance that route the client from one page functionality class to another by a click.

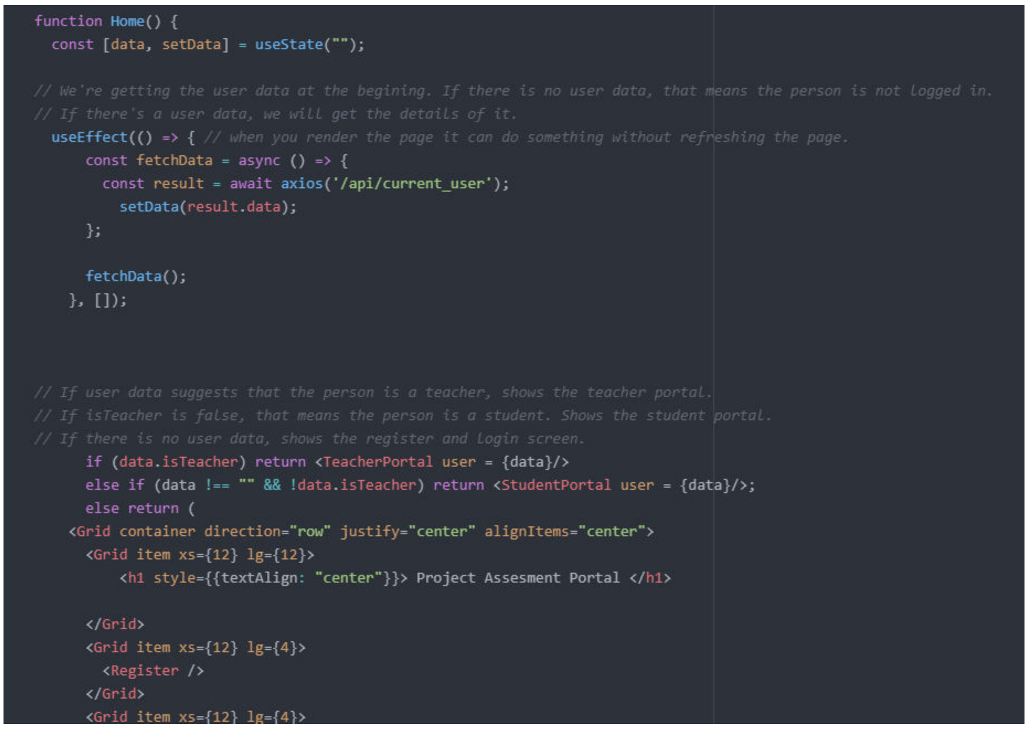


```javascript
import "./App.css";

import {BrowserRouter as Router, Route, Switch} from "react-router-dom";

import Grid from "@material-ui/core/Grid";
import Paper from "@material-ui/core/Paper";

import Home from "./Home.jsx";

function App() {
  return (
    <Router>
    <Switch>
      <Route path="/">
        <Home />
      </Route>
    </Switch>
    </Router>
  );
}

export default App;
```

Figure 14: App.js

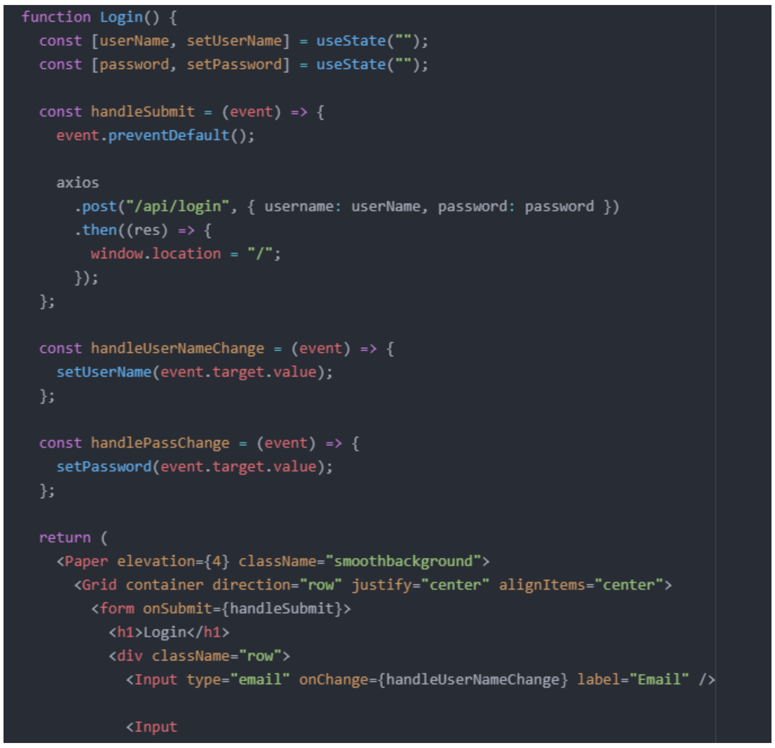
The plan of the application is comparative all over the place. As we have login facility we likewise have assign , upload, grade and comment and score base. Every one of them have their own view and relating part class and executing the fundamental information from the programming interface support and do the important activities inside the segment class and show the yield by controlling the view in like manner. Full source code of the application will as of now be given as an Attachment.

- **Home.jsx**



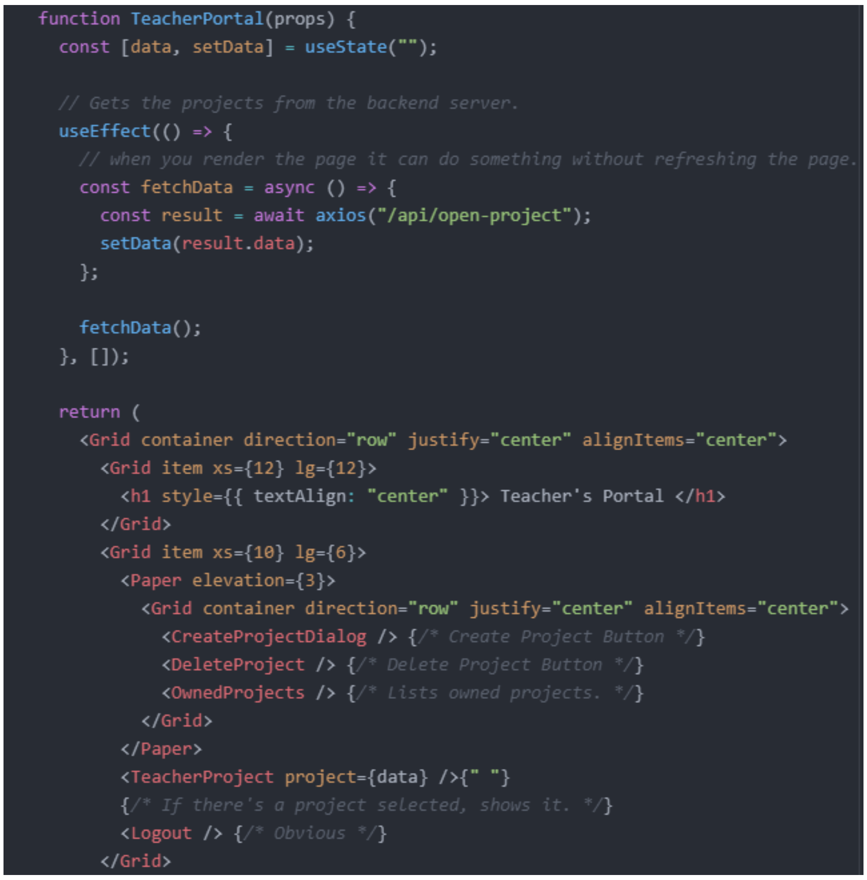
```
function Home() {
  const [data, setData] = useState("");

// We're getting the user data at the begining. If there is no user data, that means the person is not logged in.
// If there's a user data, we will get the details of it.
  useEffect(() => { // when you render the page it can do something without refreshing the page.
      const fetchData = async () => {
        const result = await axios('/api/current_user');
          setData(result.data);
      };

      fetchData();
    }, []);



// If user data suggests that the person is a teacher, shows the teacher portal.
// If isTeacher is false, that means the person is a student. Shows the student portal.
// If there is no user data, shows the register and login screen.
      if (data.isTeacher) return <TeacherPortal user = {data}/>
      else if (data !== "" && !data.isTeacher) return <StudentPortal user = {data}/>;
      else return (
    <Grid container direction="row" justify="center" alignItems="center">
      <Grid item xs={12} lg={12}>
          <h1 style={{textAlign: "center"}}> Project Assesment Portal </h1>

      </Grid>
      <Grid item xs={12} lg={4}>
        <Register />
      </Grid>
      <Grid item xs={12} lg={4}>
```

- **Login.jsx**



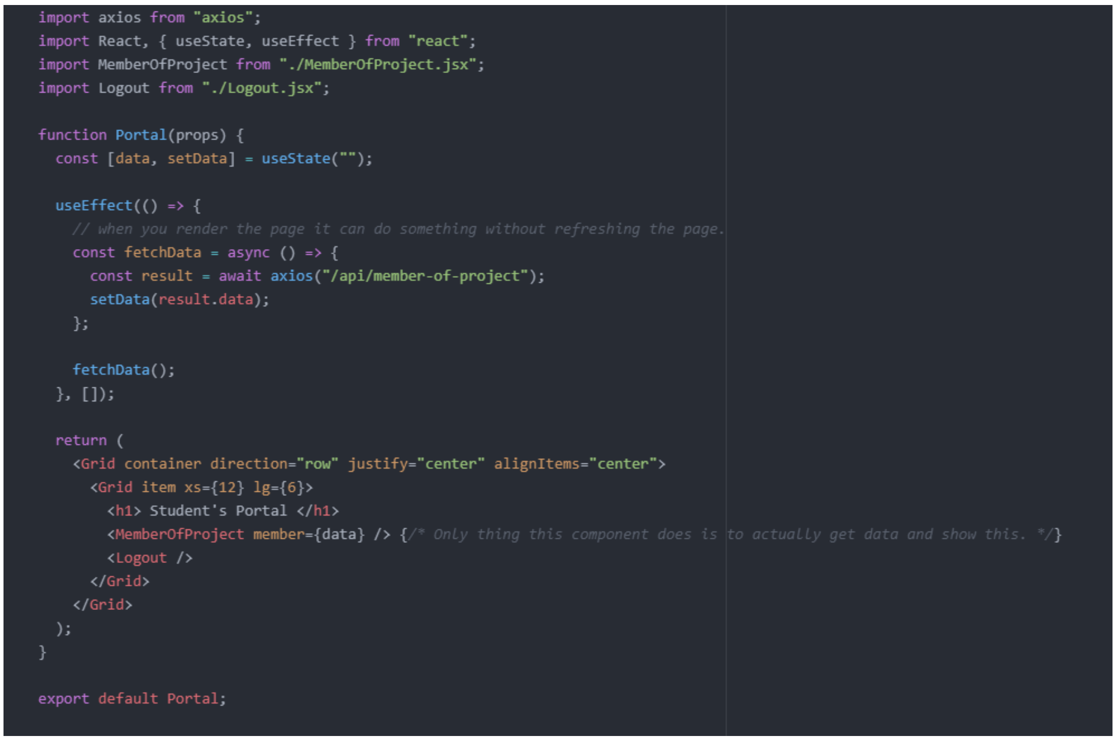
```
function Login() {
  const [userName, setUserName] = useState("");
  const [password, setPassword] = useState("");

  const handleSubmit = (event) => {
    event.preventDefault();

    axios
      .post("/api/login", { username: userName, password: password })
      .then((res) => {
        window.location = "/";
      });
  };

  const handleUserNameChange = (event) => {
    setUserName(event.target.value);
  };

  const handlePassChange = (event) => {
    setPassword(event.target.value);
  };

  return (
    <Paper elevation={4} className="smoothbackground">
      <Grid container direction="row" justify="center" alignItems="center">
        <form onSubmit={handleSubmit}>
          <h1>Login</h1>
          <div className="row">
            <Input type="email" onChange={handleUserNameChange} label="Email" />

            <Input
```

- **TeacherPortal.jsx**



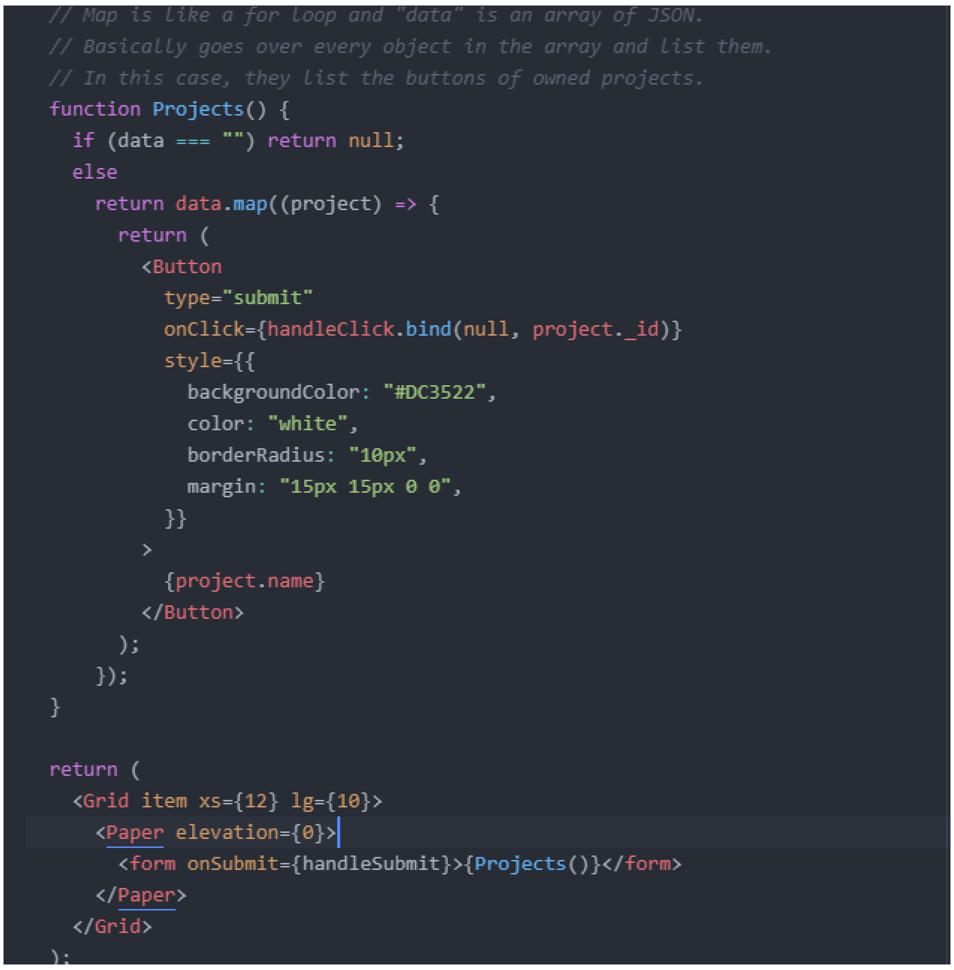
```jsx
function TeacherPortal(props) {
  const [data, setData] = useState("");

  // Gets the projects from the backend server.
  useEffect(() => {
    // when you render the page it can do something without refreshing the page.
    const fetchData = async () => {
      const result = await axios("/api/open-project");
      setData(result.data);
    };

    fetchData();
  }, []);

  return (
    <Grid container direction="row" justify="center" alignItems="center">
      <Grid item xs={12} lg={12}>
        <h1 style={{ textAlign: "center" }}> Teacher's Portal </h1>
      </Grid>
      <Grid item xs={10} lg={6}>
        <Paper elevation={3}>
          <Grid container direction="row" justify="center" alignItems="center">
            <CreateProjectDialog /> {/* Create Project Button */}
            <DeleteProject /> {/* Delete Project Button */}
            <OwnedProjects /> {/* Lists owned projects. */}
          </Grid>
        </Paper>
        <TeacherProject project={data} />{" "}
        {/* If there's a project selected, shows it. */}
        <Logout /> {/* Obvious */}
      </Grid>
```

- **<u>StudentPortal.jsx</u>**



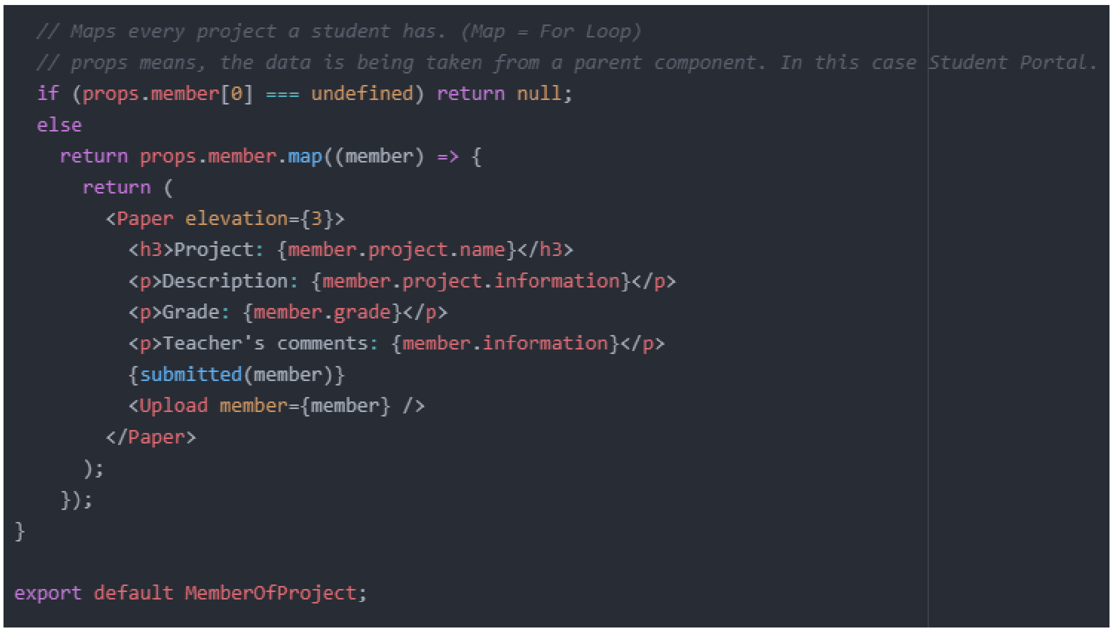
```jsx
import axios from "axios";
import React, { useState, useEffect } from "react";
import MemberOfProject from "./MemberOfProject.jsx";
import Logout from "./Logout.jsx";

function Portal(props) {
  const [data, setData] = useState("");

  useEffect(() => {
    // when you render the page it can do something without refreshing the page.
    const fetchData = async () => {
      const result = await axios("/api/member-of-project");
      setData(result.data);
    };

    fetchData();
  }, []);

  return (
    <Grid container direction="row" justify="center" alignItems="center">
      <Grid item xs={12} lg={6}>
        <h1> Student's Portal </h1>
        <MemberOfProject member={data} /> {/* Only thing this component does is to actually get data and show this. */}
        <Logout />
      </Grid>
    </Grid>
  );
}

export default Portal;
```

- **<u>OwnedProjects.jsx(Introducing Map Function)</u>**



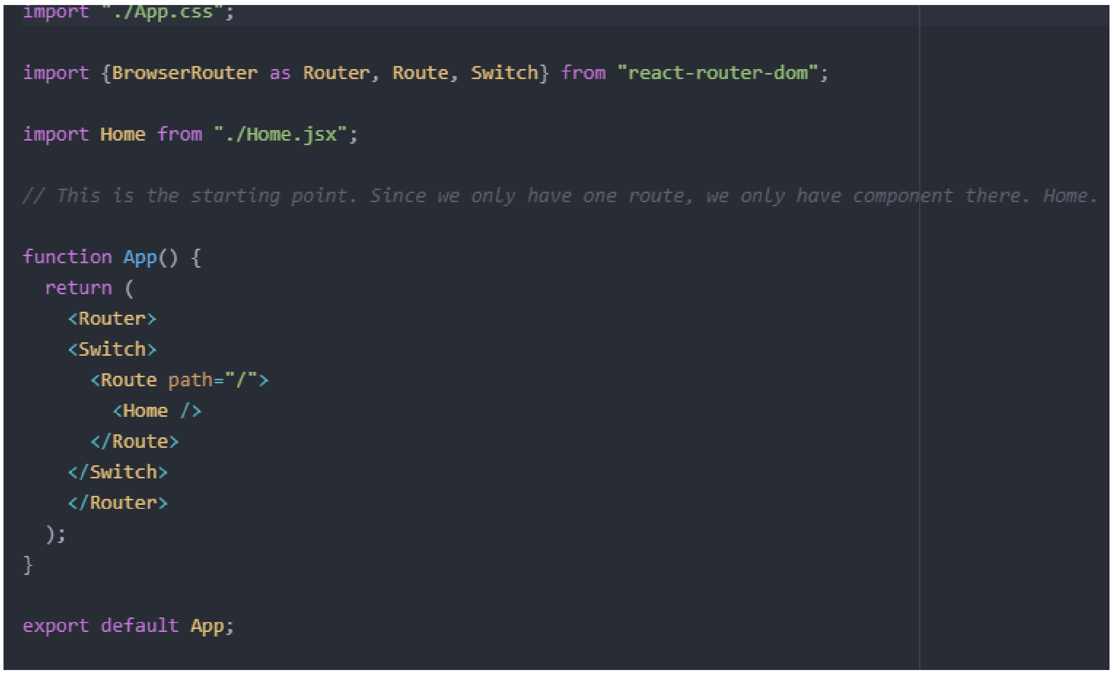
```
// Map is like a for loop and "data" is an array of JSON.
// Basically goes over every object in the array and list them.
// In this case, they list the buttons of owned projects.
function Projects() {
  if (data === "") return null;
  else
    return data.map((project) => {
      return (
        <Button
          type="submit"
          onClick={handleClick.bind(null, project._id)}
          style={{
            backgroundColor: "#DC3522",
            color: "white",
            borderRadius: "10px",
            margin: "15px 15px 0 0",
          }}
        >
          {project.name}
        </Button>
      );
    });
}

return (
  <Grid item xs={12} lg={10}>
    <Paper elevation={0}>
      <form onSubmit={handleSubmit}>{Projects()}</form>
    </Paper>
  </Grid>
);
```

- **<u>MemberOfProject.jsx</u>**



```
  // Maps every project a student has. (Map = For Loop)
  // props means, the data is being taken from a parent component. In this case Student Portal.
  if (props.member[0] === undefined) return null;
  else
    return props.member.map((member) => {
      return (
        <Paper elevation={3}>
          <h3>Project: {member.project.name}</h3>
          <p>Description: {member.project.information}</p>
          <p>Grade: {member.grade}</p>
          <p>Teacher's comments: {member.information}</p>
          {submitted(member)}
          <Upload member={member} />
        </Paper>
      );
    });
}

export default MemberOfProject;
```

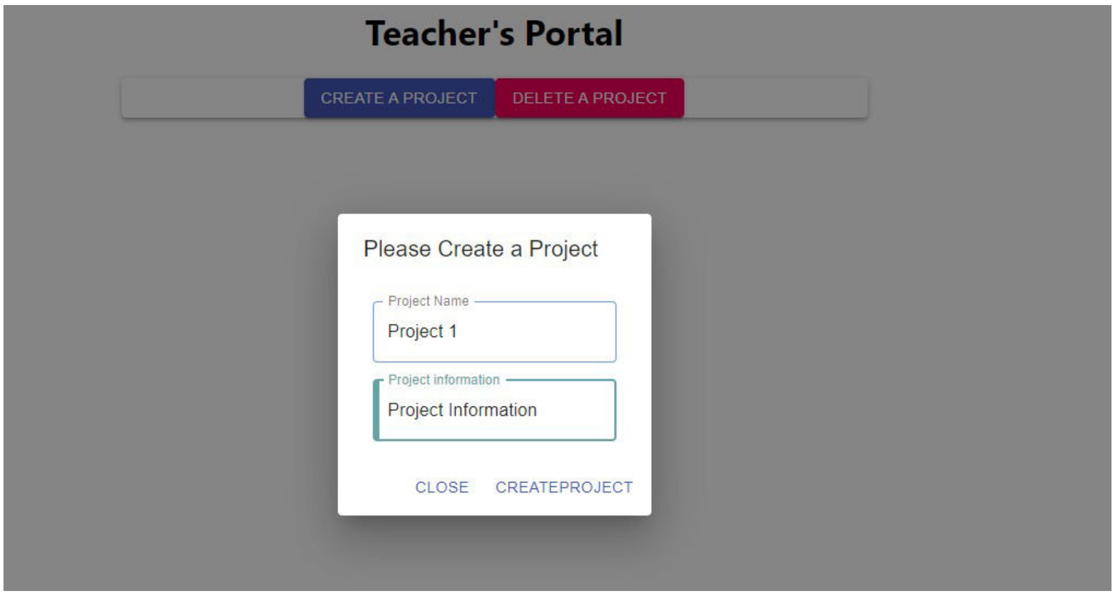
- **<u>App.js</u>**



```
import "./App.css";

import {BrowserRouter as Router, Route, Switch} from "react-router-dom";

import Home from "./Home.jsx";

// This is the starting point. Since we only have one route, we only have component there. Home.

function App() {
  return (
    <Router>
    <Switch>
      <Route path="/">
        <Home />
      </Route>
    </Switch>
    </Router>
  );
}

export default App;
```

## 6. TESTING

- **Create Project**



Teacher's Portal

CREATE A PROJECT | DELETE A PROJECT

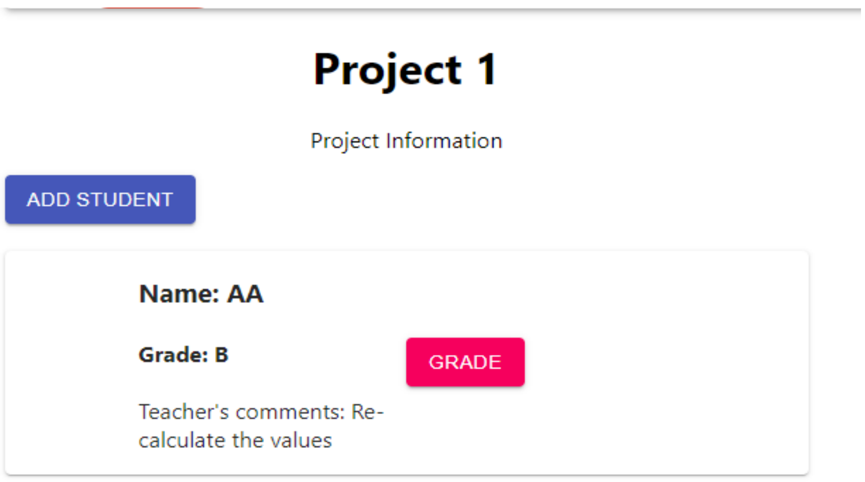
Please Create a Project

Project Name: Project 1

Project information: Project Information

CLOSE | CREATEPROJECT

- **Grade and Comment**


Project 1

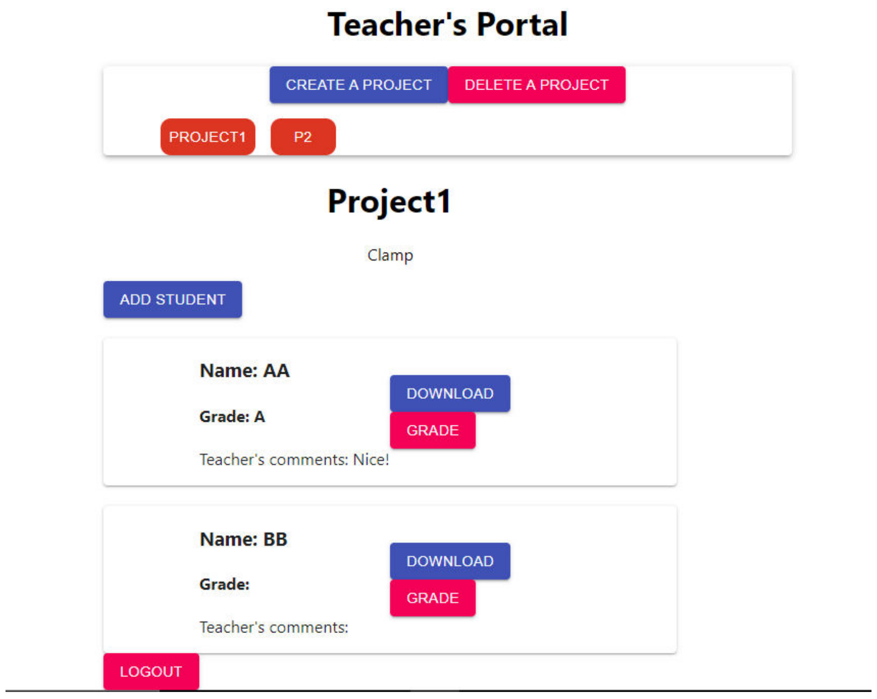
Project Information

Name: AA

Grade: B

Teacher's comments: Re-calculate the values

- **Download Submitted Work**


Teacher's Portal

Project1

Clamp

Name: AA

Grade: A

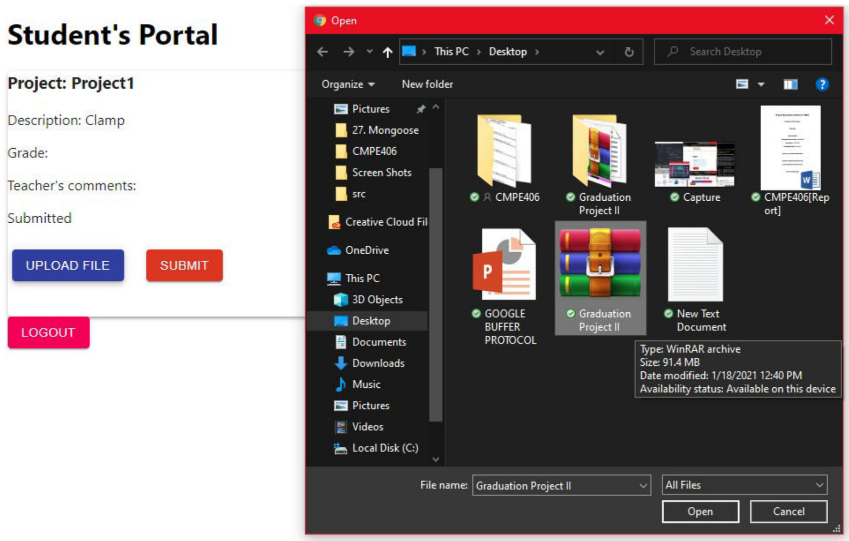
Teacher's comments: Nice!

Name: BB

Grade:

Teacher's comments:

- **Student Portal**



**Student's Portal**

Project: Project1

Description: Clamp

Grade:

Teacher's comments:

Submitted

UPLOAD FILE SUBMIT

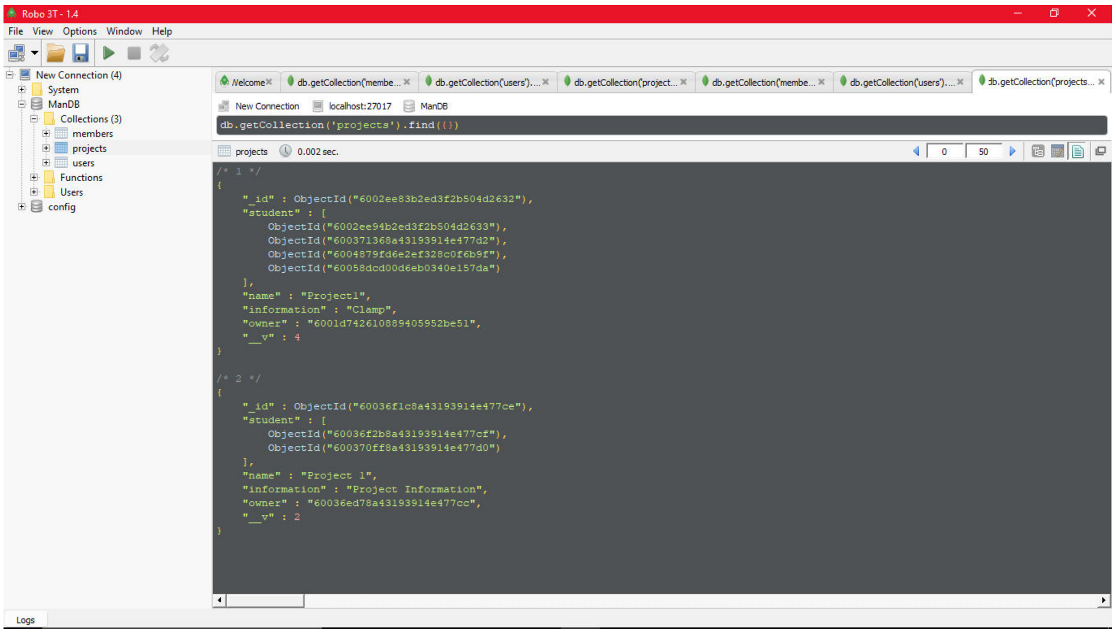
LOGOUT

Type: WinRAR archive
Size: 91.4 MB
Date modified: 1/18/2021 12:40 PM
Availability status: Available on this device

File name: Graduation Project II

All Files

- **Robo3T Connect with MongoDB**

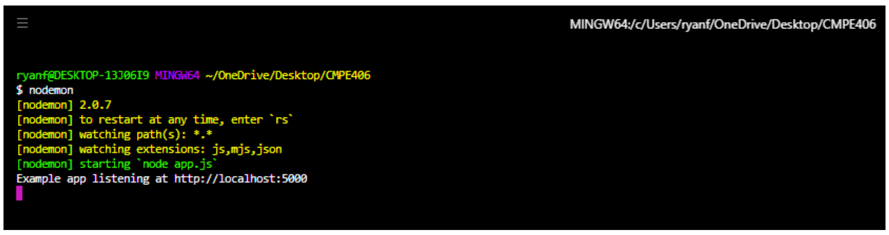


Robo 3T - 1.4
File View Options Window Help
New Connection (4)
System
ManDB
Collections (3)
members
projects
users
Functions
Users
config
Welcome
db.getCollection('membe...
db.getCollection('users')....
db.getCollection('project...
db.getCollection('membe...
db.getCollection('users')....
db.getCollection('projects...
New Connection
localhost:27017
ManDB
projects
0.002 sec.
0
50
Logs

```
db.getCollection('projects').find({})
```

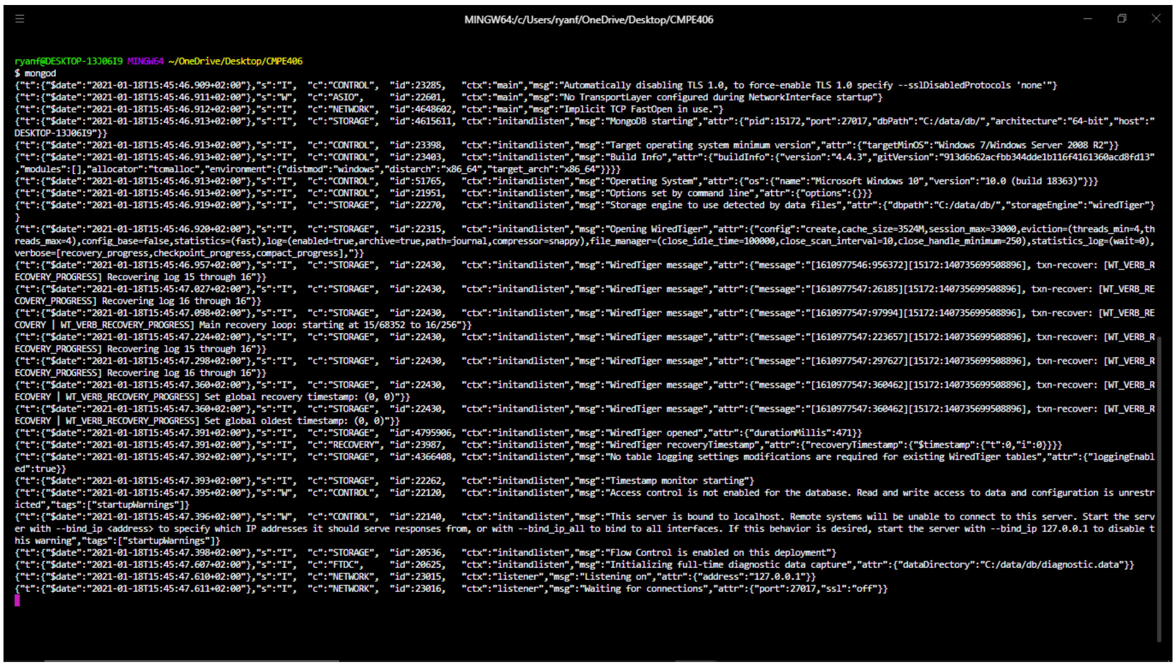
```
/* 1 */
{
    "_id" : ObjectId("6002ee83b2ed3f2b504d2632"),
    "student" : [
        ObjectId("6002ee94b2ed3f2b504d2633"),
        ObjectId("600371368a43193914e477d2"),
        ObjectId("6004879fd6e2ef328c0f6b9f"),
        ObjectId("60058dcd00d6eb0340e157da")
    ],
    "name" : "Project1",
    "information" : "Clamp",
    "owner" : "6001d742610889405952be51",
    "__v" : 4
}

/* 2 */
{
    "_id" : ObjectId("60036f1c8a43193914e477ce"),
    "student" : [
        ObjectId("60036f2b8a43193914e477cf"),
        ObjectId("600370ff8a43193914e477d0")
    ],
    "name" : "Project 1",
    "information" : "Project Information",
    "owner" : "60036ed78a43193914e477cc",
    "__v" : 2
}
```

- **HyperTerminal Running ‘nodemon’ for Server[Front End]**



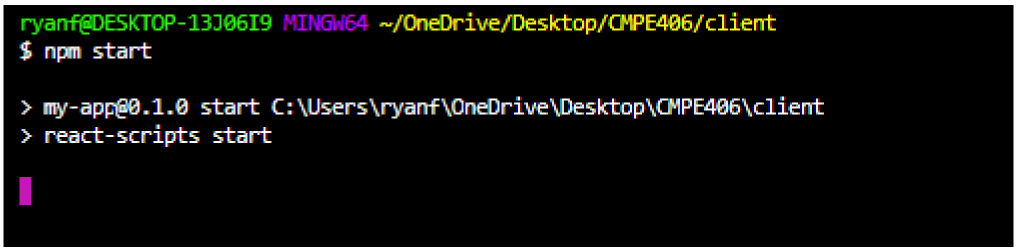
```
MINGW64:/c/Users/ryanf/OneDrive/Desktop/CMPE406

ryanf@DESKTOP-13J06I9 MINGW64 ~/OneDrive/Desktop/CMPE406
$ nodemon
[nodemon] 2.0.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app.js`
Example app listening at http://localhost:5000
```

- **HyperTerminal Running ‘mongod’ for Communication between Server and DataBase**



```
MINGW64:/c/Users/ryanf/OneDrive/Desktop/CMPE406

ryanf@DESKTOP-13J06I9 MINGW64 ~/OneDrive/Desktop/CMPE406
$ mongod
{"t":{"$date":"2021-01-18T15:45:46.909+02:00"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx":"main","msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2021-01-18T15:45:46.911+02:00"},"s":"W",  "c":"ASIO",     "id":22601,   "ctx":"main","msg":"No TransportLayer configured during NetworkInterface startup"}
{"t":{"$date":"2021-01-18T15:45:46.912+02:00"},"s":"I",  "c":"NETWORK",  "id":4648602, "ctx":"main","msg":"Implicit TCP FastOpen in use."}
{"t":{"$date":"2021-01-18T15:45:46.913+02:00"},"s":"I",  "c":"STORAGE",  "id":4615611, "ctx":"initandlisten","msg":"MongoDB starting","attr":{"pid":15172,"port":27017,"dbPath":"C:/data/db/","architecture":"64-bit","host":"
DESKTOP-13J06I9"}}
{"t":{"$date":"2021-01-18T15:45:46.913+02:00"},"s":"I",  "c":"CONTROL",  "id":23398,   "ctx":"initandlisten","msg":"Target operating system minimum version","attr":{"targetMinOS":"Windows 7/Windows Server 2008 R2"}}
{"t":{"$date":"2021-01-18T15:45:46.913+02:00"},"s":"I",  "c":"CONTROL",  "id":23403,   "ctx":"initandlisten","msg":"Build Info","attr":{"buildInfo":{"version":"4.4.3","gitVersion":"913d6b62acfbb344dde1b116f4161360acd8fd13"
,"modules":[],"allocator":"tcmalloc","environment":{"distmod":"windows","distarch":"x86_64","target_arch":"x86_64"}}}}
{"t":{"$date":"2021-01-18T15:45:46.913+02:00"},"s":"I",  "c":"CONTROL",  "id":51765,   "ctx":"initandlisten","msg":"Operating System","attr":{"os":{"name":"Microsoft Windows 10","version":"10.0 (build 18363)"}}}
{"t":{"$date":"2021-01-18T15:45:46.913+02:00"},"s":"I",  "c":"CONTROL",  "id":21951,   "ctx":"initandlisten","msg":"Options set by command line","attr":{"options":{}}}
{"t":{"$date":"2021-01-18T15:45:46.919+02:00"},"s":"I",  "c":"STORAGE",  "id":22270,   "ctx":"initandlisten","msg":"Storage engine to use detected by data files","attr":{"dbpath":"C:/data/db/","storageEngine":"wiredTiger"}
}
{"t":{"$date":"2021-01-18T15:45:46.920+02:00"},"s":"I",  "c":"STORAGE",  "id":22315,   "ctx":"initandlisten","msg":"Opening WiredTiger","attr":{"config":"create,cache_size=3524M,session_max=33000,eviction=(threads_min=4,th
reads_max=4),config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=(close_idle_time=100000,close_scan_interval=10,close_handle_minimum=250),statistics_log=(wait=0),
verbose=[recovery_progress,checkpoint_progress,compact_progress],"}}
{"t":{"$date":"2021-01-18T15:45:46.957+02:00"},"s":"I",  "c":"STORAGE",  "id":22430,   "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":"[1610977546:956372][15172:140735699508896], txn-recover: [WT_VERB_R
ECOVERY_PROGRESS] Recovering log 15 through 16"}}
{"t":{"$date":"2021-01-18T15:45:47.027+02:00"},"s":"I",  "c":"STORAGE",  "id":22430,   "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":"[1610977547:26185][15172:140735699508896], txn-recover: [WT_VERB_RE
COVERY_PROGRESS] Recovering log 16 through 16"}}
{"t":{"$date":"2021-01-18T15:45:47.098+02:00"},"s":"I",  "c":"STORAGE",  "id":22430,   "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":"[1610977547:97994][15172:140735699508896], txn-recover: [WT_VERB_RE
COVERY | WT_VERB_RECOVERY_PROGRESS] Main recovery loop: starting at 15/68352 to 16/256"}}
{"t":{"$date":"2021-01-18T15:45:47.224+02:00"},"s":"I",  "c":"STORAGE",  "id":22430,   "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":"[1610977547:223657][15172:140735699508896], txn-recover: [WT_VERB_R
ECOVERY_PROGRESS] Recovering log 15 through 16"}}
{"t":{"$date":"2021-01-18T15:45:47.298+02:00"},"s":"I",  "c":"STORAGE",  "id":22430,   "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":"[1610977547:297627][15172:140735699508896], txn-recover: [WT_VERB_R
ECOVERY_PROGRESS] Recovering log 16 through 16"}}
{"t":{"$date":"2021-01-18T15:45:47.360+02:00"},"s":"I",  "c":"STORAGE",  "id":22430,   "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":"[1610977547:360462][15172:140735699508896], txn-recover: [WT_VERB_R
ECOVERY | WT_VERB_RECOVERY_PROGRESS] Set global recovery timestamp: (0, 0)"}}
{"t":{"$date":"2021-01-18T15:45:47.360+02:00"},"s":"I",  "c":"STORAGE",  "id":22430,   "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":"[1610977547:360462][15172:140735699508896], txn-recover: [WT_VERB_R
ECOVERY | WT_VERB_RECOVERY_PROGRESS] Set global oldest timestamp: (0, 0)"}}
{"t":{"$date":"2021-01-18T15:45:47.391+02:00"},"s":"I",  "c":"STORAGE",  "id":4795906, "ctx":"initandlisten","msg":"WiredTiger opened","attr":{"durationMillis":471}}
{"t":{"$date":"2021-01-18T15:45:47.391+02:00"},"s":"I",  "c":"RECOVERY", "id":23987,   "ctx":"initandlisten","msg":"WiredTiger recoveryTimestamp","attr":{"recoveryTimestamp":{"$timestamp":{"t":0,"i":0}}}}
{"t":{"$date":"2021-01-18T15:45:47.392+02:00"},"s":"I",  "c":"STORAGE",  "id":4366408, "ctx":"initandlisten","msg":"No table logging settings modifications are required for existing WiredTiger tables","attr":{"loggingEnabl
ed":true}}
{"t":{"$date":"2021-01-18T15:45:47.393+02:00"},"s":"I",  "c":"STORAGE",  "id":22262,   "ctx":"initandlisten","msg":"Timestamp monitor starting"}
{"t":{"$date":"2021-01-18T15:45:47.395+02:00"},"s":"W",  "c":"CONTROL",  "id":22120,   "ctx":"initandlisten","msg":"Access control is not enabled for the database. Read and write access to data and configuration is unrestr
icted","tags":["startupWarnings"]}
{"t":{"$date":"2021-01-18T15:45:47.396+02:00"},"s":"W",  "c":"CONTROL",  "id":22140,   "ctx":"initandlisten","msg":"This server is bound to localhost. Remote systems will be unable to connect to this server. Start the serv
er with --bind_ip <address> to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable t
his warning","tags":["startupWarnings"]}
{"t":{"$date":"2021-01-18T15:45:47.398+02:00"},"s":"I",  "c":"STORAGE",  "id":20536,   "ctx":"initandlisten","msg":"Flow Control is enabled on this deployment"}
{"t":{"$date":"2021-01-18T15:45:47.607+02:00"},"s":"I",  "c":"FTDC",     "id":20625,   "ctx":"initandlisten","msg":"Initializing full-time diagnostic data capture","attr":{"dataDirectory":"C:/data/db/diagnostic.data"}}
{"t":{"$date":"2021-01-18T15:45:47.610+02:00"},"s":"I",  "c":"NETWORK",  "id":23015,   "ctx":"listener","msg":"Listening on","attr":{"address":"127.0.0.1"}}
{"t":{"$date":"2021-01-18T15:45:47.611+02:00"},"s":"I",  "c":"NETWORK",  "id":23016,   "ctx":"listener","msg":"Waiting for connections","attr":{"port":27017,"ssl":"off"}}
```

- **HyperTerminal Running ‘npm start’ for Server [ Back End]**



```
ryanf@DESKTOP-13J06I9 MINGW64 ~/OneDrive/Desktop/CMPE406/client
$ npm start

> my-app@0.1.0 start C:\Users\ryanf\OneDrive\Desktop\CMPE406\client
> react-scripts start
```

## Evaluation and Implementation

### Student Outcomes:

Graduates of the Computer Engineering program should attain

1. an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics

2. an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors

3. an ability to communicate effectively with a range of audiences

4. an ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts

5. an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives

6. an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions

7. an ability to acquire and apply new knowledge as needed, using appropriate learning strategies.

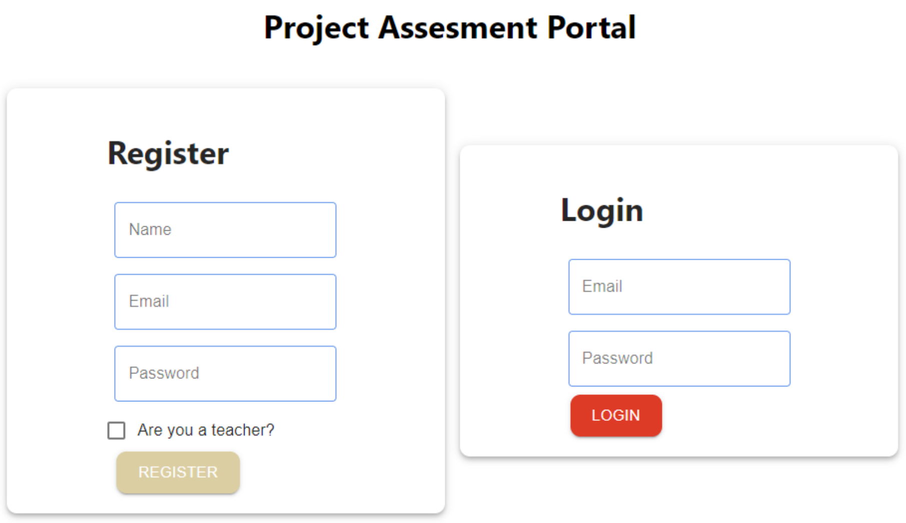
In case the level of achievement cannot be easily judged by rubrics, and a numeric score will be used for assessment of student performance, the following mapping between numeric scores and degree of achievement will be used.

| **Numeric Achievement (NA) = ( score obtained / max score possible)** | **Achievement Level** |
|---|---|
| NA >= 75 % | 4 |
| 50 <= NA < 75 | 3 |
| 25 <= NA < 50 | 2 |
| NA < 25 | 1 |

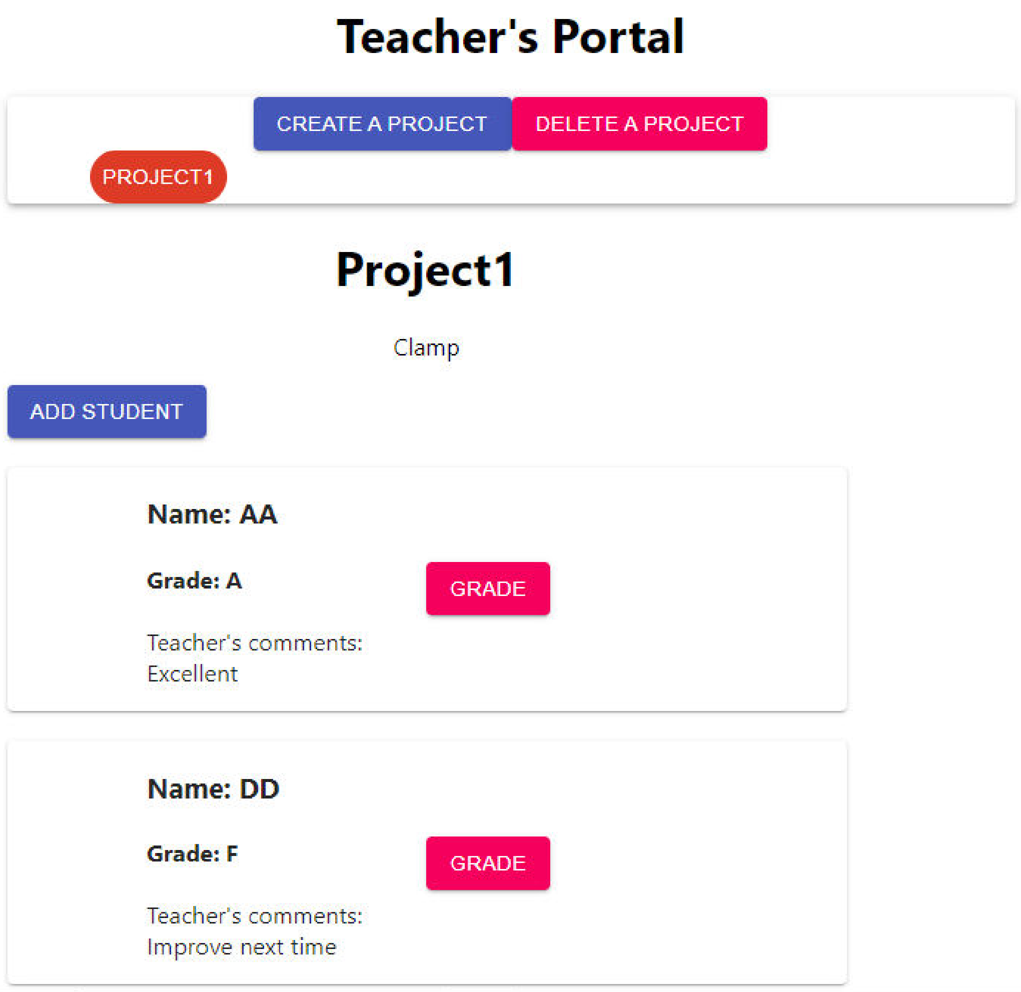
**For direct measurement, achievement levels 3 and 4 are to be called "satisfactory" and achievement levels 1 and 2 are to be called "unsatisfactory". For survey results (which are based on a scale of 1 to 5), achievement levels 4 and 5 are to be called "satisfactory" and achievement levels 1, 2 and 3 are to be called "unsatisfactory".**

## 7. USER GUIDE OF THE SYSTEM

### Login/Register



**Teacher's Portal**



Name: AA

Grade: A

Teacher's comments:
Excellent

Name: DD

Grade: F

Teacher's comments:
Improve next time

## 8. DISCUSSION

Having a digitized methods and means of data requisition in terms of quick and fast response based hierarchy of client-server methodology where a user can simply send and receive data over the intranet where areas such practice of coding and other data retrieval are being adopted worldwide reduction paper production also for a fact that during the times of the pandemic SARS-CoV2, educational and institutionalized sectors have adapted to the communication standards by using online [web-based] applications and facilities to pursue further and normalize the use and circulation of data virtually instead of using paper causing a decrease and demand and reducing the cutting down of trees.

## 9. CONCLUSION

ABET authorize colleges satisfy certain guidelines. Thus the ABET organization requires an appraisal report to guarantee that these principles are being met. In any case, the age of the reports is no simple accomplishment and is troublesome. The point of this report is make an online application that will facilitate the information assortment measure and create a legitimate report. We've made an electronic application that will introduce a configuration to gather information for making plot record of a Project, create Project\Student Assigning learning results archive of the course, acknowledge score sheet include and decipher for evaluation and producing last report for ABET accreditation appraisal.

Throughout building up the venture,We in an incredible arrangement about web advancement – NodeJS(Back-end dev.), Material-UI(front-end dev.), JavaScript, HTML/CSS, MongoDB, front-side/server side engineering and so on.

## 10. REFERENCES

1. https://staff.emu.edu.tr/duygucelik/en/teaching/cmse405/report-formats

2. https://www.zenflowchart.com/organizational-chart/

3. https://jsonworld.com/demo/uploading-files-with-reactjs-and-nodejs

4. http://expressjs.com/en/api.html#res.download

5. https://go.gliffy.com/

6. https://canva.com/

7. https://dbschema.com/