

A REPORT ON PROJECT WORK TITLE
ONLINE EXAMINATION SEATING ARRANGEMENT



SUBMITTED BY
PROJECT ASSOCIATES:

S.NO	PROJECT TEAM	PIN NO
1	Ch. Rohini	22010-CM-023
2	B. Molisha	22010-CM-013
3	P. Venu	22635-CM-029
4	Ch. Shanmukha Vamsi	22010-CM-028
5	B.Sailendra Kumar	22010-CM-017
6	K. Anil Kumar	22010-CM-057
7	B. Siddardha	22010-CM-020
8	B.Gideon	22010-CM-021
9	A. Tharun	21202-CM-001

IN PARTIAL FULFILLMENT OF REQUIREMENTS FOR THE
AWARD OF

DIPLOMA IN
COMPUTER ENGINEERING

UNDER THE ESTEEMED GUIDENCE OF

Ms. V.Anasuya, M.Tech

DEPARTMENT OF COMPUTER ENGINEERING
ANDHRA POLYTECHNIC

KAKINADA-533002

2022-2025

DEPARTMENT OF COMPUTER ENGINEERING
CERTIFICATE

This is to certify that the project work entitled “ONLINE EXAMINATION SEATING ARRANGEMENT ” is the bonafide record work done by Mr./Ms. _____ bearing the Pin No. _____ of the final year, submitted in the partial fulfilment of the requirement for the award of DIPLOMA IN COMPUTER ENGINEERING during the academic year 2022-2025.

Signature of project guide

(Smt. V.ANASUYA, M.Tech)

Head of the Department

(Sri. G. Partha Sarathi, M.Tech)

Signature of the External Examiner

ACKNOWLEDGEMENT

We have been tried to our level best to complete this project with group discussion among our batch-mates and taking valuable suggestions from our classmates and made it to work properly and effectively.

We thankful to our Project guide Smt V. Anasuya, M.Tech for her cooperation and valuable suggestions, which motivated us to complete this project.

We also extend our thanks to our Senior Lecturer for their valuable suggestions during the course of Project from time to time.

We also thankful to our Head of Computer Engineering and our project coordinator Sri G. Parthi Sarthi, M.Tech for his cooperation and unflinching Devotion, which motivated us to complete this project.

We also thankful to our Respected Principal Sri N.Janardhana Rao M.tech, for his kind support in successful complete on this project.

We also thankful to all teaching and non-teaching staff who has helped us either directly or indirectly in completion of this project.

PROJECT ASSOCIATES

PROJECT TEAM:

PIN NO:

Ch. Rohini	22010-CM-023
B. Molisha	22010-CM-013
P. Venu	22635-CM-029
Ch. Shanmukha Vamsi	22010-CM-028
B. Sailendra Kumar	22010-CM-017
K. Anil Kumar	22010-CM-057
B. Siddardha	22010-CM-020
B. Gideon	22010-CM-021
A.Tharun	21202-CM-001

TABLE OF CONTENT

1. Abstract.....	!
1.1 Existing System.....	!
1.2 Proposed System.....	!
2. Software Requirements.....	!
3. Introduction.....	!
4. System Analysis.....	!
4.1 Requirement Analysis.....	!
4.2 Feasibility Study.....	!
5. Software Requirements & Specifications.....	!
6. Design.....	!
6.1 System Modeling.....	!
7. Frontend Installation.....	!
7.1 Install React.....	!
8. Backend Installation.....	!
8.1 Install Node.js.....	!
9. Output & Screenshots.....	!

ABSTRACT

The traditional examination seating arrangement process is often time-consuming, prone to errors, and lacks flexibility, especially in handling large student populations. Presently, institutions rely on manual seat allocation or basic software tools, which may not effectively prevent malpractices such as impersonation and unfair seating patterns. Modern advancements integrate AI-powered automation, cloud-based systems, and facial recognition to streamline seating assignments, ensuring fairness and security.

This project aims to develop an **Online Exam Seating Arrangement System** that automates the allocation of seats based on predefined rules, reducing manual effort and minimizing human errors. The system dynamically assigns seats while considering factors like subject codes, roll numbers, and hall capacities. Additionally, features like QR-based verification and real-time updates enhance efficiency. By implementing this technology, educational institutions can conduct examinations seamlessly, ensuring transparency, security, and ease of management.

Our **Online Exam Seating Arrangement System** includes various essential features to streamline the examination process. The system considers multiple branches, subjects, and student lists to generate an optimized seating plan. It takes input such as student roll numbers, registered subjects, and hall capacities to allocate seats systematically. The system ensures that students from the same branch or subject are distributed evenly across different halls to prevent malpractice. Additionally, it includes an admin panel for uploading student lists, managing seating preferences, and making real-time modifications if needed. Features like hall mapping, QR-based verification, and automatic seat allocation make the process efficient, reducing manual workload and ensuring a smooth examination experience.

SOFTWARE REQUIREMENTS

SOFTWARE SPECIFICATION:

OS: MS WINDOWS 11
FRONT END: HTML,CSS,REACT JS
BACKEND: NODE
DATABASE: MYSQL
PLATFORM: VISUAL STUDIO CODE

HARDWARE SPECIFICATION:

PROCESSOR: 12TH Gen Intel(R) Core (TM) i5-1235U 1.30GHz
RAM: 16 GB RAM

WEB BROWSER:

Chrome, Internet Explorer (or) any Browser (or) Opera

INTRODUCTION

Examination seating arrangement system gives page out results under all circumstances, once the data collection is over. Our system manages everything perfectly. Our system reduces the manual effect and speeds up the processing of results. The present project has been developed to meet the aspirations indicated in the modern age. An attempt has been made through this project to do all work ease & fast. It provides current add, Update, Move Next, Move Previous, Move Last, Find & Delete all facilities to accomplish the desired objectives. The facility Include in this project and the suggested activities have been organized to impart knowledge & develop skill & attitude in the College official works.

Logic Description:

We might be clear with the problem which we are going to solve from the above figure. As you can see, we need to obtain output of seating numbers using input of starting number and ending number. In most of the schools and colleges, exams are having a block of 30 students, so in that case, let's have a looping variable 'j' which will indicate a row for each iteration. For a block containing 30 students we require 5 rows. So range of 'j' is 1 to 5. There is a relation of some distance between 1st and 2nd column, which is same for 3rd and 4th column, 5th and 6th column. This distance is decreased by 2 for each iteration. We can add a facility of printing Block no., it can be incremented after each 5 iterations of this loop. Concepts have used some file functions which are required to generate a notepad file containing this seating arrangement, which can be easily printed out.

Seating Arrangement System

Seating Arrangement System provides a lot of facility to their user. The objective and scope of my Project Seating Arrangement System is to record the details various activities of user. It will simplify the task and reduce the paper work. During implementation every user will be given appropriate training to suit their specific needs. Specific support will also be provided at key points within the academic calendar. Training will be provided on a timely basis, and you will be trained as the new is Seating Arrangement System rolled out to your area of responsibility. Hence the management system for the College management has been designed to remove all the deficiency from which the present system is suffering and to ensure.

Seating Arrangement System Usages

The client uses MS Excel, and maintains their records, however it is not possible them to share the data from multiple system in multi user environment, there is lot of duplicate work, and chance of mistake. When the records are changed they need to update each and every excel file. There is no option to find and print previous saved records. There is no security; anybody can access any report and sensitive data, also no reports to summary report. This Seating Arrangement System is used to overcome the entire problem which they are facing currently, and making complete atomization of manual system to computerized system.

HTML:

HTML, or Hypertext Markup Language, is **a markup language used to create and structure content on the web.**²³ It defines the structure of web pages and is the foundational building block for websites.²³ HTML is not a programming language but rather a markup language that annotates text to define how it is structured and displayed by web browsers.³ It consists of elements, which are typically enclosed in tags, and these elements are responsible for creating and structuring the content of a webpage.²³

HTML elements usually consist of a start tag, content, and a closing tag.²³ The basic structure of an HTML document includes the document type declaration, the HTML root element, the head element containing metadata, and the body element containing visible content.²³ The head element includes elements like the title, which defines the title of the webpage, and the meta tag, which provides information such as the character set or viewport settings.²

HTML was initially created by Tim Berners-Lee in 1991 and has evolved through various versions, with HTML5 being the latest and most advanced version.

CSS:

Cascading Style Sheets (CSS) is **a style sheet language used for specifying the presentation and styling of a document written in a markup language such as HTML or XML** (including XML dialects like SVG, MathML, or XHTML).⁶ CSS is designed to enable the separation of content and presentation, including layout, colors, and fonts.¹ This separation can improve content accessibility, since the content can be written without concern for its presentation, and it provides more flexibility and control in the specification of presentation characteristics.¹

CSS allows you to apply styles to HTML documents by prescribing colors, fonts, spacing, and positioning.² It simplifies the process of making web pages presentable and enhances user experience by controlling the appearance of HTML elements.²⁶ CSS separates content from design, improving accessibility and providing responsive designs for different devices and screen sizes.²⁶

CSS consists of style rules that are interpreted by the browser and applied to the corresponding elements. A style rule set includes a selector and a declaration block. The selector targets specific HTML elements to apply styles, and the declaration block contains one or more declarations separated by semicolons. Each declaration includes a CSS property name and a value, separated by a colon.²⁶

CSS resolves conflicts by prioritizing rules based on specificity and order. Inline CSS has the highest priority, so it overrides internal and external styles.

REACT:

React is a **JavaScript library for building user interfaces (UIs) on the web**. It allows developers to create large web applications that can change data without reloading the page, focusing on being fast, scalable, and simple.²³ React is declarative, meaning developers describe how the UI should look at different points in time, and React handles updating the UI to match the description.³ Components are the building blocks of React applications, and they are JavaScript functions that can be combined to create complex UIs.³ React uses a Virtual DOM to optimize rendering performance by minimizing DOM updates, which enhances the application's performance.

NODE.JS:

Node.js is a **JavaScript runtime environment built on Chrome's V8 JavaScript engine, designed to run JavaScript outside the browser on the server side**.⁵⁶ It is an open-source, cross-platform tool that enables developers to build scalable network applications using asynchronous, event-driven programming.⁵⁶ Node.js uses a single-threaded architecture with event loops to handle multiple requests concurrently without blocking, making it efficient for real-time and high-performance applications.⁵⁶ This runtime environment allows developers to use JavaScript for both client-side and server-side programming, streamlining the development process.

MYSQL:

MySQL is an **open-source relational database management system (RDBMS) that uses Structured Query Language (SQL) for database operations.**^{23,4} It is designed to store, manage, and retrieve structured data, making it a popular choice for various applications, from small websites to large-scale enterprise systems.²⁴

A relational database organizes data into tables with rows and columns, allowing for efficient data storage and retrieval.³ MySQL supports ACID transactions, ensuring data integrity and consistency.⁴ It is known for its reliability, speed, and ease of use, and it can scale both vertically and horizontally to accommodate growing data and user loads.²⁴

MySQL is versatile and can run on various platforms, including UNIX, Linux, and Windows.³ It is often used in web applications, such as content management systems, e-commerce platforms, and social media sites.²⁴ Large companies like Facebook, Netflix, Airbnb, and Amazon also use MySQL for their database needs.³

MySQL is available under the GNU General Public License (GPL) and is also available under a variety of proprietary licenses.²³ Oracle Corporation currently develops, distributes, and supports MySQL.

SYSTEM STUDY AND ANALYSIS

EXISTING SYSTEM

- The existing system only provides text-based interface, which is not as user-friendly as Graphical user Interface.
- Since the system is implemented in Manual, so the response is very slow.
- The transactions are executed in off-line mode, hence on-line data capture and modification is not possible.
- Off-line reports cannot be generated due to batch mode execution

PROPOSED SYSTEM

Hence, there is a need of reformation of the system with more advantages and flexibility. The Seating Arrangement System eliminates most of the limitations of the existing software. It has the following objectives:

Enhancement:

The main objective of Seating Arrangement System is to enhance and upgrade the existing system by increasing its efficiency and effectiveness. The software improves the working methods by replacing the existing manual system with the computer-based system.

Automation:

The Seating Arrangement System automates each and every activity of the manual system and increases its throughput. Thus the response time of the system is very less and it works very fast.

Accuracy:

The Seating Arrangement System provides the uses a quick response with very accurate information regarding the users etc. Any details or system in an accurate manner, as and when required.

User-Friendly:

The software Seating Arrangement System has a very user-friendly interface. Thus the users will feel very easy to work on it. The software provides accuracy along with pleasant interface. Make the present manual system more interactive, speedy and usefriendly.

Availability:

The transaction reports of the system can be retried as and when required. Thus, there is no delay in the availability of any information, whatever needed, can be captured very quickly and easily.

Maintenance Cost:

Reduce the cost of maintenance.

FEASIBILITY STUDY

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. The Three key considerations involved in the feasibility analysis are

- 1.ECONOMICAL FEASIBILITY
- 2.TECHNICAL FEASIBILITY
- 3.SOCIAL FEASIBILITY

1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

2. TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed

system must have a modest requirement, as only minimal or null changes are required for implementing this system.

3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

DESIGN

SYSTEM DESIGN

The system design for examination seating arrangement **aims to automate the manual process of assigning seats to students, making it more efficient and reducing errors.** Such systems typically include a database to store student registration details, such as name, roll number, branch, year, and subjects, which can be used to generate seating arrangements for exams.

The design involves several key components:

- **Database Management System (DBMS):** Utilizes a DBMS like MySQL to manage and control data in a structured manner.⁵
- **Graph Colouring Algorithm:** Used to ensure that students taking the same course are not seated together, thereby minimizing cheating opportunities.⁵
- **SMS Notification Feature:** Allows for timely communication of seating details to students.⁵
- **GUI for Data Entry:** Enables the addition of class and course data, as well as student data, through a graphical user interface.

These systems also include functionalities for administrative login, automatic seating arrangement generation, and printing of seating charts.

The implementation of such systems can significantly streamline the examination process, making it more secure and efficient.

System Modelling:

System modelling for examination seating arrangement involves **automating the process of assigning students to exam halls and seats**. This system aims to simplify the traditional manual process, which is often tedious and error-prone.

The model typically includes the following components:

1. **Student Details Module:** Collects and stores information about students, such as name, roll number, branch, year, and subjects registered for the exam.
2. **Room Allocation Module:** Allocates rooms based on the number of students and the size of the rooms.
3. **Seating Arrangement Module:** Generates seating charts for each room, ensuring that students from the same school do not sit next to each other.
4. **Report Generation Module:** Automatically generates seating arrangement reports for each room, which can be accessed by students and invigilators.
5. **User Interface:** Provides an interface for administrators to input exam details and view seating arrangements.

The system uses machine learning techniques to improve predictions and behaviours, making the seating arrangement process more efficient and accurate.

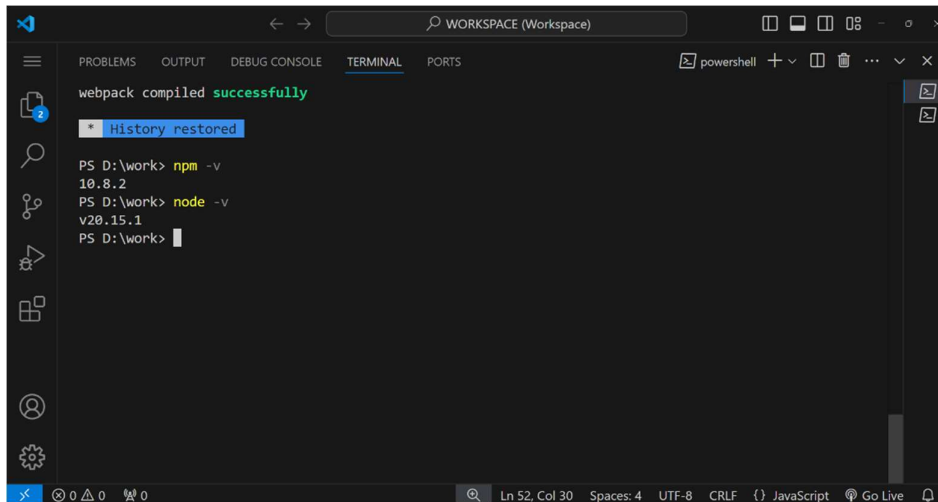
FRONTEND

1. Install Node.js

React requires Node.js and npm (Node Package Manager).

Download and install Node.js from Node.js official site.

After installation, verify Node.js and npm versions:



```
webpack compiled successfully
History restored
PS D:\work> npm -v
10.8.2
PS D:\work> node -v
v20.15.1
PS D:\work>
```

node -v

npm -v

Option 1: Using create-react-app (Recommended for Beginners)

create-react-app is a boilerplate to quickly set up a React project.

1. Open your terminal and run:

```
npx create-react-app my-app
```

Replace my-app with your desired project name.

This sets up a new React project with all dependencies.

2. Navigate to the project directory:

```
cd my-app
```

3. Start the development server:

npm start

Open <http://localhost:3000> in your browser to see your React app.

Option 2: Manual Installation (Advanced Users)

1. Create a new project folder:

```
mkdir my-app
```

```
cd my-app
```

2. Initialize a new Node.js project:

```
npm init -y
```

3. Install React and React DOM:

```
npm install react react-dom
```

4. Install a development server and Babel for JSX:

```
npm install webpack webpack-cli webpack-dev-server @babel/core @babel/preset-react babel-loader --save-dev
```

5. Create necessary files:

public/index.html:

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8" />
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

```
<title>React App</title>
```

```
</head>
<body>
<div id="root"></div>
</body>
</html>
```

src/index.js:

```
import React from 'react';
import ReactDOM from 'react-dom';
const App = () => <h1>Hello, React!</h1>;
ReactDOM.render(<App />, document.getElementById('root'));
```

webpack.config.js:

```
const path = require('path');
module.exports = {
  entry: './src/index.js',
  output: {
    path: path.resolve(__dirname, 'dist'),
    filename: 'bundle.js',
  },
  module: {
    rules: [
      {
        test: /\.jsx?$/,
        exclude: /node_modules/,
        use: {
          loader: 'babel-loader',
          options: {
            presets: ['@babel/preset-react'],
```

```
},  
},  
},  
],  
},  
devServer: {  
  contentBase: path.join(__dirname, 'public'),  
  compress: true,  
  port: 3000,  
}  
};
```

6. Start the development server:

`npx webpack serve`

Open <http://localhost:3000> to view the app.

Frontend:

```
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\boddu\OneDrive\Desktop>cd project
C:\Users\boddu\OneDrive\Desktop\project>cd frontend
C:\Users\boddu\OneDrive\Desktop\project\frontend>code .
C:\Users\boddu\OneDrive\Desktop\project\frontend>npm start

> frontend@0.1.0 start
> react-scripts start

(node:20560) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.
(Use 'node --trace-deprecation ...' to show where the warning was created)
(node:20560) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.
Starting the development server...
Compiled successfully!

You can now view frontend in the browser.

Local:      http://localhost:3000
On Your Network:  http://192.168.29.209:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

BACKEND

1. Install Node.js on Windows

1. Download the Node.js Installer:

- Go to the <https://nodejs.org/en>
- You will see two versions available: LTS (Long Term Support) and Current. Choose LTS for better stability, especially if you're starting a new project.

2. Run the Installer:

- After downloading, run the installer (.msi file) and follow the prompts.
- Ensure you check the option to add Node.js to your system PATH during installation. This will allow you to run Node.js commands from the command line.

3. Verify the Installation:

Open Command Prompt or PowerShell and run the following commands to check if Node.js and npm (Node Package Manager) were installed correctly:

- `node -v`
- `npm -v`

2. Install Node.js and npm

If you haven't already installed Node.js and npm, please follow the instructions from the previous response. Node.js installation includes npm, which you'll need to create and manage React projects.

3. Create a New React App Using create-react-app

The easiest way to get started with React is by using the create-react-app tool. This tool sets up a new React project with all the necessary configurations and dependencies. Follow these steps:

1. Open Terminal or Command Prompt:

- Navigate to the directory where you want to create your React project.
- Run the create-react-app command
npx create-react-app my-react-app
- Replace my-react-app with your desired project name.
- npx is a package runner tool that comes with npm 5.2 and later versions. It will download and run create-react-app without installing it globally.

Wait for the Installation to Complete:

- This command will take a few minutes to set up your project, install all the required packages, and create the initial project structure.

Navigate to your folder : cd my-react-app

Start the React Application:
npm start

- This will start the development server, and your new React application will open in your default browser at <http://localhost:3000>.

4. Install React and ReactDOM:

npm install react react-dom

5. Installing Additional Packages

After setting up your React project, you may want to install additional packages to enhance its functionality. Some common packages include:

ReactRouterforrouting:
npm install react-router-dom

Axios for making HTTP requests:

`npm install axios`

- Bootstrap for styling:

`npm install bootstrap`

- React Recharts:

`npm install recharts`

- React Sweetalert:

`npm install sweetalert`

- React Mysql:

`npm install mysql`

- React nodemon:

`npm install nodemon`

- React Bodyparser:

`npm install bodyparser`

- React express:

`npm install express`

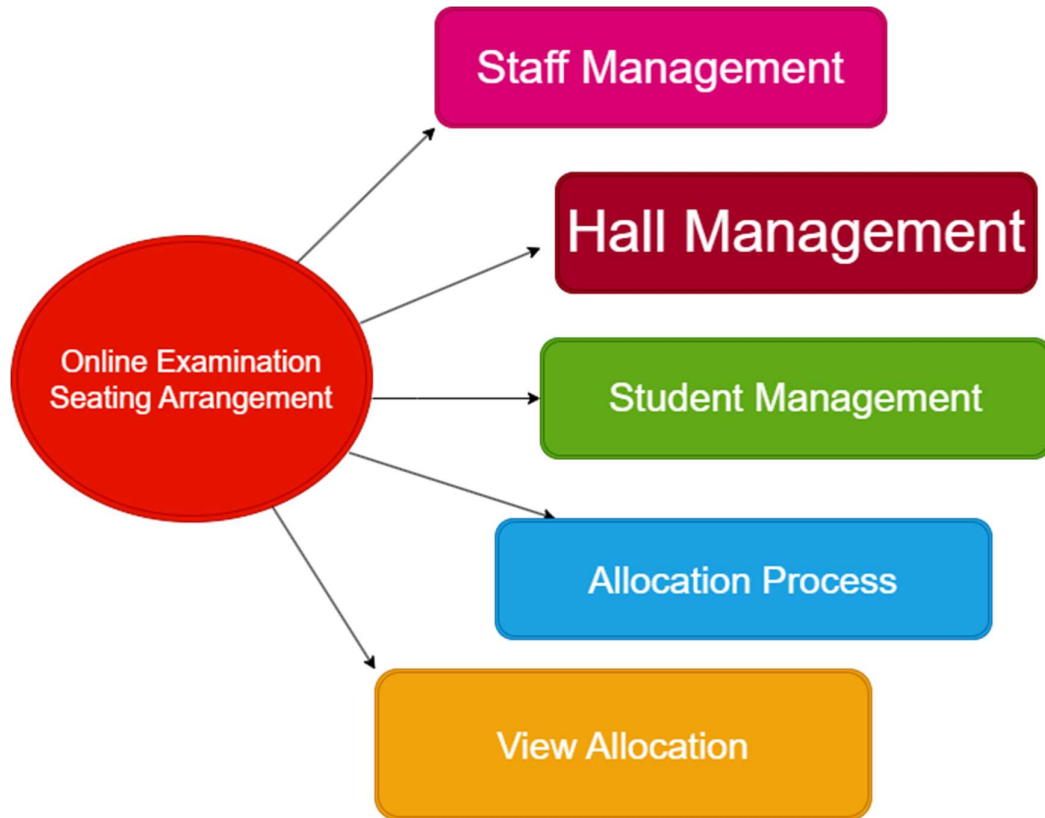
- React cors:

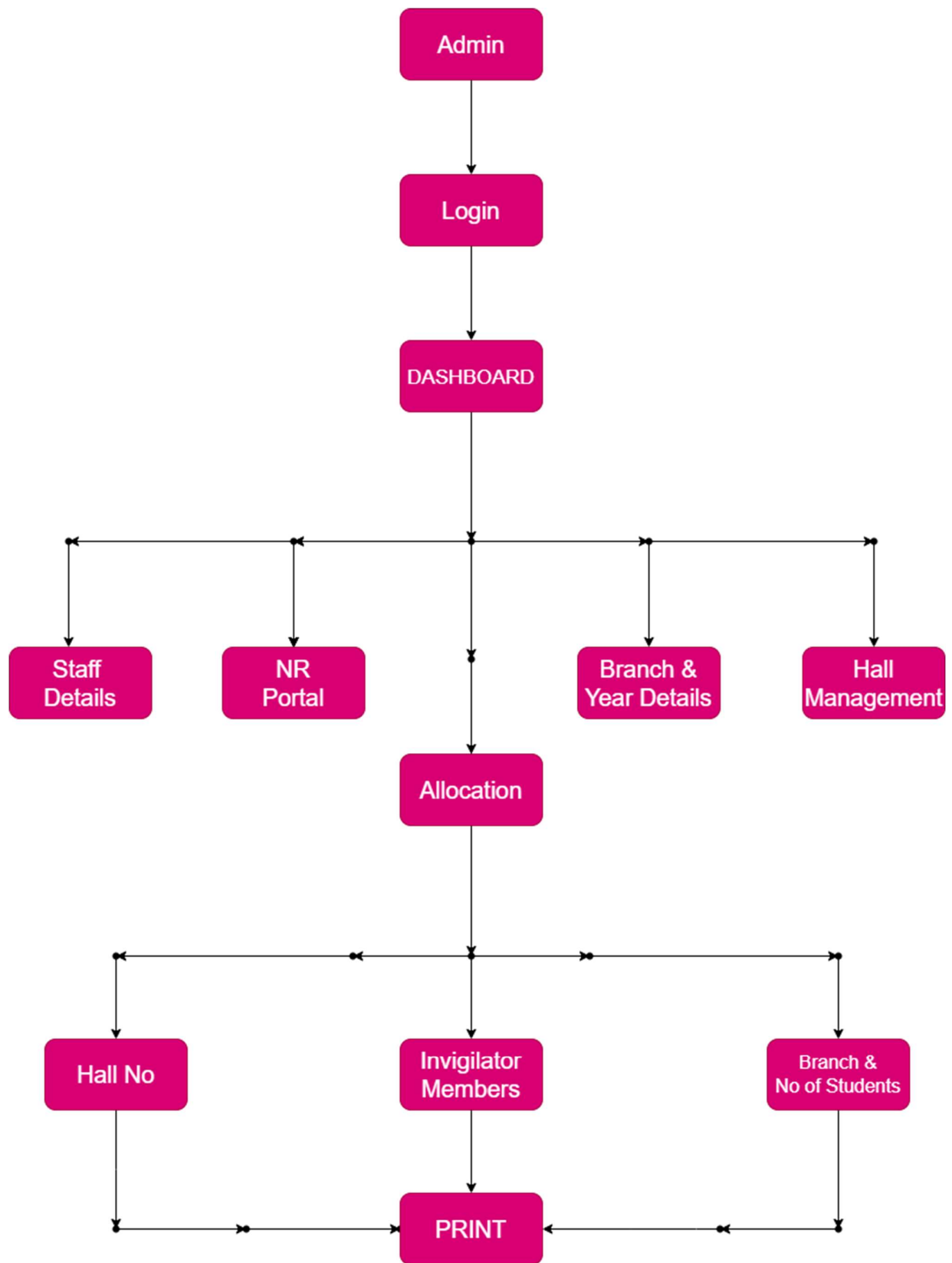
`npm install cors`

- React http:

`npm install http`

FLOWCHARTS





1. Testing Objectives

- Ensure correct seat allocation for students.
- Verify that hall capacities are not exceeded.
- Check for proper distribution of students across different halls.
- Ensure that view allocation functions correctly.
- Validate system performance under various conditions (e.g., large number of students).

2. Test Cases

- **Functional Testing**
 - Test if students are correctly assigned to halls.
 - Check if the hall allocation respects maximum seating capacity.
 - Ensure that students from the same department or roll number series are distributed correctly (if applicable).
- **User Interface Testing**
 - Check if the UI displays seat allocations clearly.
 - Verify that all buttons, dropdowns, and navigation options work properly.

3. Testing Methodology

- **Manual Testing:** Assign a small dataset of students and verify allocations manually.
- **Automated Testing (if applicable):** Use scripts to check if allocations match expected results.
- **Load Testing:** Simulate a large number of students to test performance.

4. Test Results and Observations

- Document results of each test case.
- Mention any issues found and how they were fixed.
- Highlight improvements based on testing feedback.

Backend:

//EXAM SEATING ARRANGMENT ALLOCATION PROCESS

```
app.post("/get-seating-arrangement", (req, res) => {  
  const { selectedBranches, hallId, date, time } = req.body;  
  
  if (!selectedBranches.length || !hallId || !date || !time) {  
    return res.status(400).json({ message: "Missing required fields" });  
  }  
  
  const hallQuery = SELECT * FROM hall WHERE hallId = ?;  
  connection.query(hallQuery, [hallId], (err, hallResults) => {  
    if (err || hallResults.length === 0) {  
      return res.status(500).json({ message: "Hall not found" });  
    }  
  
    const { rows, columns } = hallResults[0];  
    let seatingMatrix = Array.from({ length: rows }, () => Array(columns).fill(null));  
  
    const studentQuery =  
    `SELECT s.pin_no, s.branch  
    FROM student2 s  
    WHERE s.branch IN (?)  
    AND NOT EXISTS (  
    SELECT 1 FROM seating_allocation sa WHERE sa.pin_no = s.pin_no  
    AND sa.date = ? AND sa.time = ?  
    )  
    ORDER BY s.branch, s.pin_no`;  
  
    connection.query(studentQuery, [selectedBranches, date, time], (err, students) => {  
      if (err || students.length === 0) {
```

```
return res.status(400).json({ message: "No students available" });  
}
```

```
let branchQueues = {};  
selectedBranches.forEach(branch => branchQueues[branch] = []);  
students.forEach(student => branchQueues[student.branch].push(student));
```

```
function isSafeToSit(matrix, row, col, branch) {  
  return !(  
    (row > 0 && matrix[row - 1][col]?.branch === branch) ||  
    (col > 0 && matrix[row][col - 1]?.branch === branch) ||  
    (row < rows - 1 && matrix[row + 1][col]?.branch === branch) ||  
    (col < columns - 1 && matrix[row][col + 1]?.branch === branch)  
  );  
}
```

```
let branchOrder = [...selectedBranches];  
let branchIndex = 0;  
let remainingStudents = Object.values(branchQueues).flat();  
  
for (let r = 0; r < rows; r++) {  
  for (let c = 0; c < columns; c++) {  
    if (remainingStudents.length > 0) {  
      let foundStudent = null;  
      for (let i = 0; i < branchOrder.length; i++) {  
        let branch = branchOrder[(branchIndex + i) % branchOrder.length];  
        if (branchQueues[branch].length > 0) {  
          let student = branchQueues[branch][0];  
          if (isSafeToSit(seatingMatrix, r, c, student.branch)) {  
            foundStudent = branchQueues[branch].shift();
```

```
branchIndex = (branchIndex + 1) % branchOrder.length;
```

```
break;
```

```
}
```

```
}
```

```
}
```

```
seatingMatrix[r][c] = foundStudent;
```

```
}
```

```
}
```

```
}
```

```
let finalSeating = [];
```

```
for (let r = 0; r < rows; r++) {
```

```
  for (let c = 0; c < columns; c++) {
```

```
    let seat = seatingMatrix[r][c];
```

```
    finalSeating.push(seat ? { row: r, col: c, ...seat } : { row: r, col: c, pin_no: null });
```

```
  }
```

```
}
```

```
const insertStudentQuery = INSERT INTO seating_allocation (pin_no, branch, hallId, date, time)  
VALUES ?;
```

```
const studentValues = finalSeating
```

```
.filter(student => student.pin_no)
```

```
.map(student => [student.pin_no, student.branch, hallId, date, time,]);
```

```
const facultyQuery =
```

```
`SELECT id, faculty_name
```

```
FROM faculty
```

```
WHERE faculty_designation IN ('Senior Lecturer', 'Lecturer')
```

```
AND id NOT IN (
```

```
SELECT faculty_id FROM faculty_allocation WHERE date = ? AND time = ?
```

```
)
```

ORDER BY id

LIMIT 2`;

```
connection.query(facultyQuery, [date, time], (err, facultyResults) => {  
  if (err) {  
    return res.status(500).json({ message: "Database error while fetching faculty" });  
  }  
  
  if (facultyResults.length < 2) {  
    return res.status(400).json({ message: "Not enough available faculty for this hall" });  
  }  
}
```

```
let facultyAllocation = facultyResults.map((faculty, index) => ({  
  hallId: hallId,  
  id: faculty.id,  
  faculty_name: faculty.faculty_name,  
  date: date,  
  time: time,  
  faculty_designation: index === 0 ? "Supervisor 1" : "Supervisor 2"  
})));
```

```
const insertFacultyQuery = INSERT INTO faculty_allocation (faculty_id, hallId, date, time,  
faculty_designation) VALUES ?;
```

```
const facultyValues = facultyAllocation.map(faculty => [faculty.id, hallId, date, time,  
faculty.faculty_designation]);
```

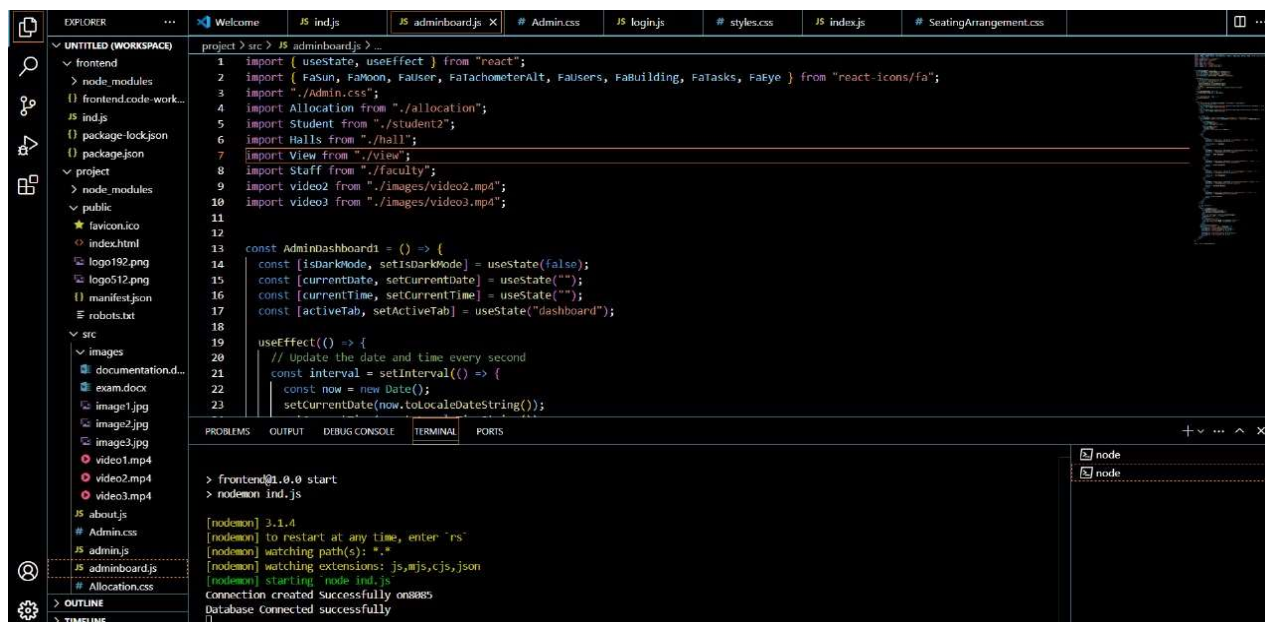
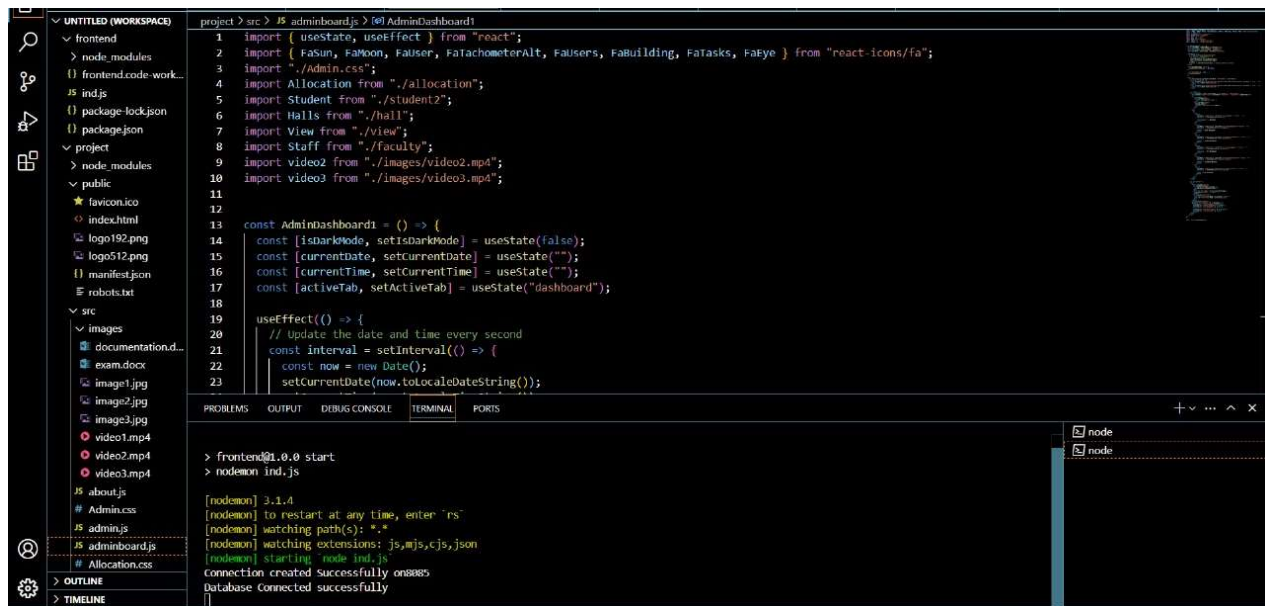
```
if (studentValues.length > 0) {  
  connection.query(insertStudentQuery, [studentValues], (err) => {  
    if (err) {  
      console.error("Database error while saving student allocation:", err);  
      return res.status(500).json({ message: "Database error." });  
    }  
  }  
}
```

```
}
```

```
connection.query(insertFacultyQuery, [facultyValues], (err) => {  
  if (err) {  
    console.error("Database error while saving faculty allocation:", err);  
    return res.status(500).json({ message: "Database error." });  
  }  
}
```

```
res.status(200).json({  
  seating: finalSeating,  
  faculty: facultyAllocation,  
  rows: rows,  
  columns: columns  
});  
});  
});  
}  
});  
});  
});  
});
```

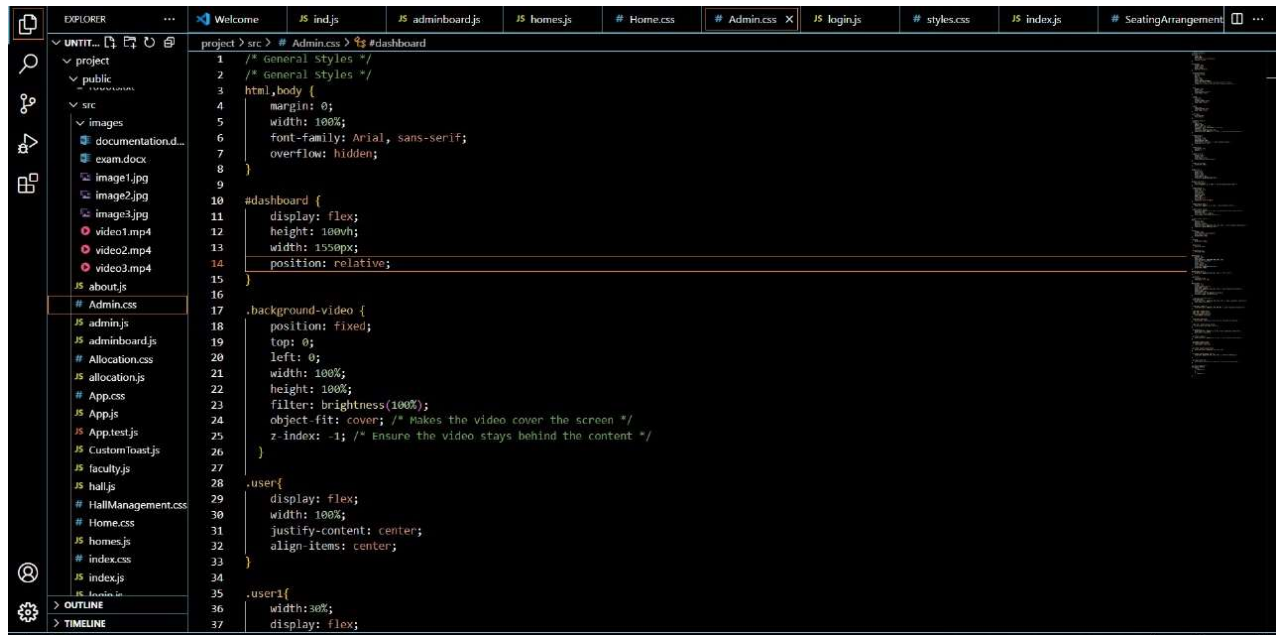
SCREENSHOTS:



```
EXPLORER  Welcome  JS ind.js  JS adminboard.js  JS homes.js  # Admin.css  JS login.js  # styles.css  JS index.js  # SeatingArrangement.css  ...

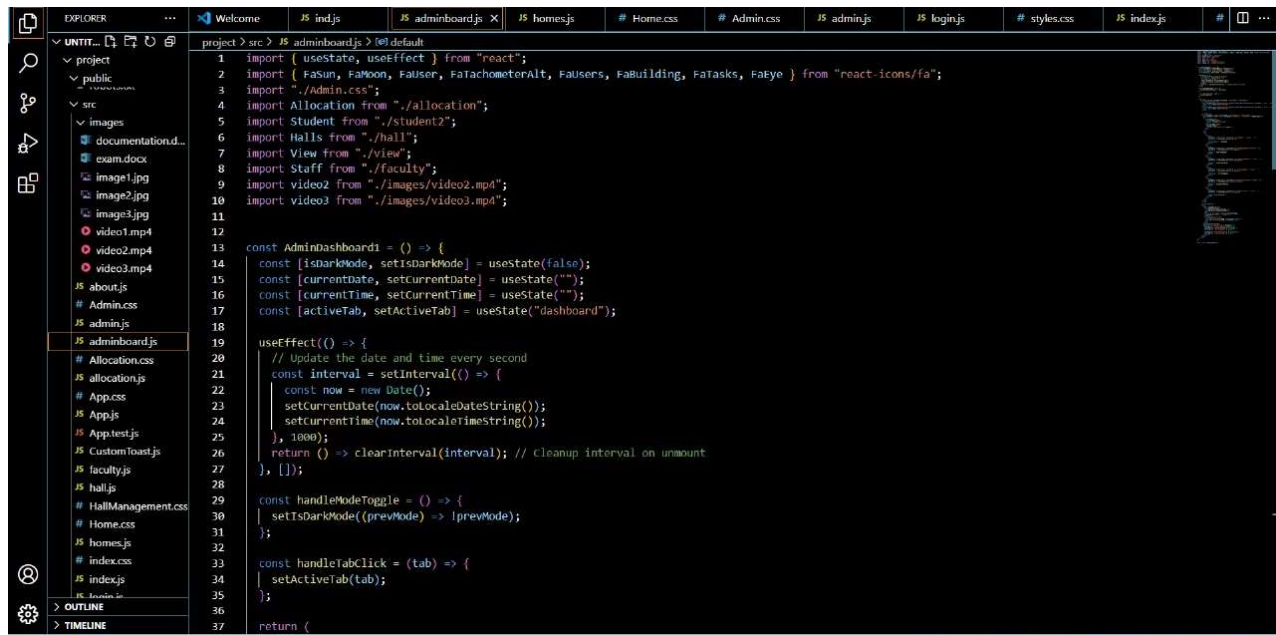
project > src > JS homes.js > Home
1 import React, { useState, useEffect } from 'react';
2 import { FaInstagram, FaFacebook, FaTwitter, FaWhatsapp, FaSun, FaMoon } from 'react-icons/fa';
3 import './Home.css';
4 import { useNavigate } from 'react-router-dom';
5 import Login from './login.js';
6
7 import image1 from './images/image1.jpg';
8 import image2 from './images/image2.jpg';
9 import image3 from './images/image3.jpg';
10 import video1 from './images/video1.mp4';
11 import video3 from './images/video3.mp4';
12 import video2 from './images/video2.mp4';
13 const Home = () => {
14   const [isDarkMode, setIsDarkMode] = useState(false);
15   const [currentTab, setCurrentTab] = useState('home'); // Default tab
16   const [currentDateTime, setCurrentDateTime] = useState(new Date().toLocaleString()); // Initialize with current date and time
17   const navigate = useNavigate();
18
19   // Update date and time every second
20   useEffect(() => {
21     const interval = setInterval(() => {
22       setCurrentDateTime(new Date().toLocaleString());
23     }, 1000);
24
25     return () => clearInterval(interval); // Cleanup interval on component unmount
26   }, []);
27
28   // Toggle between light and dark mode
29   const toggleMode = () => setIsDarkMode(!isDarkMode);
30
31   // Handle tab change
32   const handleTabChange = (tab) => setCurrentTab(tab);
33
34   return (
35     <div className="home-container ${isDarkMode ? 'dark-mode' : 'light-mode'}">
36       <div className="background-video" autoPlay loop muted style={{ display: isDarkMode ? 'none' : 'block' }}>
37
```

```
project > src > # Home.css > .current-datetime.dark-mode
1 /* General Styling */
2 html, body {
3   width: 100%;
4   height: fit-content;
5   overflow-y: auto;
6 }
7
8 a {
9   text-decoration: none;
10 }
11
12 .home-container {
13   position: relative;
14   width: 1500px;
15   min-height: 100vh;
16   font-family: 'Arial', sans-serif;
17   overflow-x: hidden;
18   transition: all 0.3s ease;
19 }
20
21 .btn1 {
22   width: 50px;
23   height: 35px;
24   justify-content: center;
25   display: flex;
26   align-items: center;
27 }
28
29 .light-mode {
30   color: #0000;
31   background-color: rgba(255, 255, 255, 0);
32 }
33
34 .dark-mode {
35   color: #ffff;
36   background: rgba(0, 0, 0, 0.8); /* Dark transparent background */
37
```

The screenshot shows the Visual Studio Code editor with a project named 'project'. The Explorer sidebar on the left shows a file structure with folders 'public' and 'src', and various files including 'documentation.d...', 'exam.docx', 'image1.jpg', 'image2.jpg', 'image3.jpg', 'video1.mp4', 'video2.mp4', 'video3.mp4', 'about.js', 'Admin.css', 'admin.js', 'adminboard.js', 'Allocation.css', 'allocation.js', 'App.css', 'App.js', 'AppTest.js', 'CustomToast.js', 'faculty.js', 'hall.js', 'HallManagement.css', 'Home.css', 'homes.js', 'index.css', and 'index.js'. The main editor area shows the 'Admin.css' file with the following CSS code:

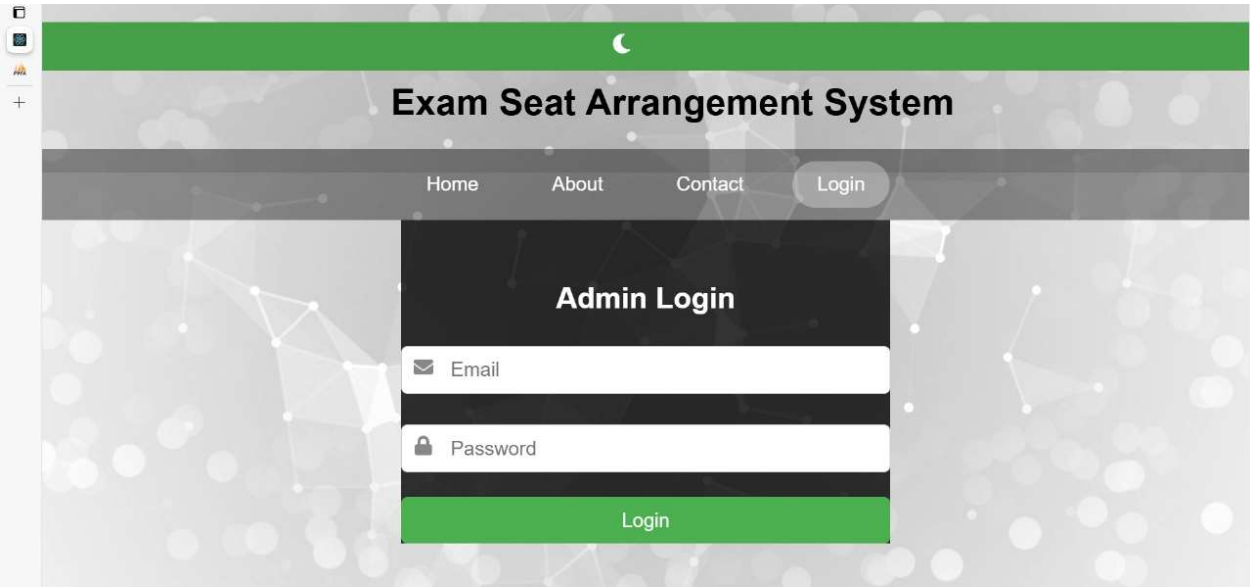
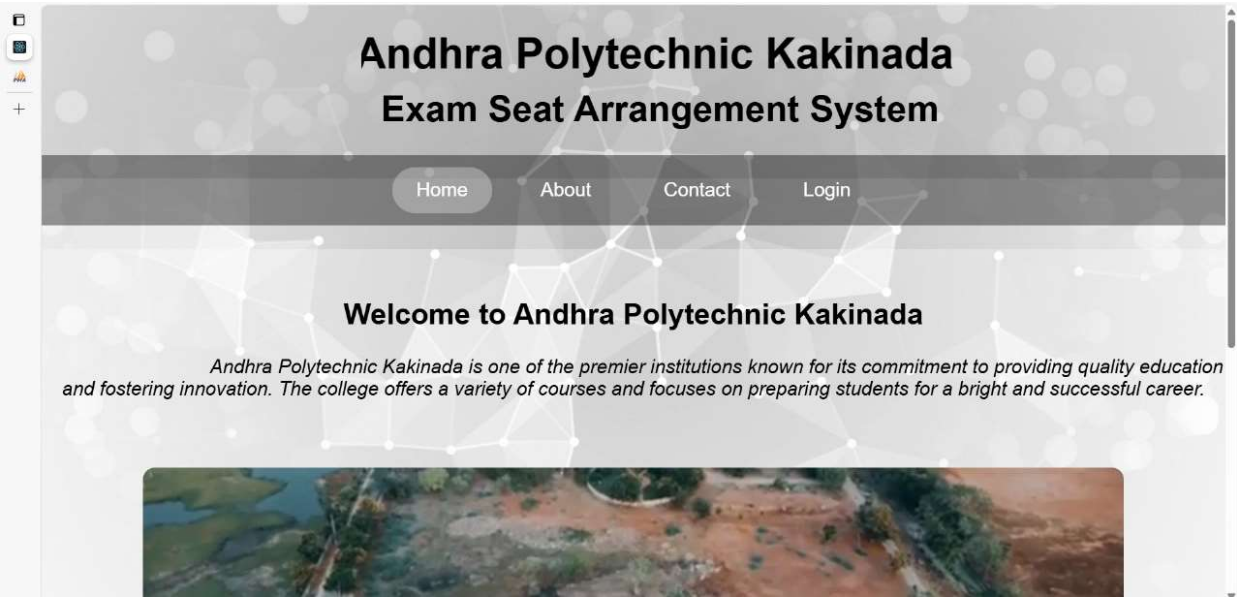
```
1 /* general styles */
2 /* general styles */
3 html, body {
4   margin: 0;
5   width: 100%;
6   font-family: Arial, sans-serif;
7   overflow: hidden;
8 }
9
10 #dashboard {
11   display: flex;
12   height: 100vh;
13   width: 1550px;
14   position: relative;
15 }
16
17 .background-video {
18   position: fixed;
19   top: 0;
20   left: 0;
21   width: 100%;
22   height: 100%;
23   filter: brightness(100%);
24   object-fit: cover; /* Makes the video cover the screen */
25   z-index: -1; /* Ensure the video stays behind the content */
26 }
27
28 .user {
29   display: flex;
30   width: 100%;
31   justify-content: center;
32   align-items: center;
33 }
34
35 .user1 {
36   width: 50%;
37   display: flex;
```



The screenshot shows the Visual Studio Code editor with the same project structure as the first image. The Explorer sidebar shows the same file structure. The main editor area shows the 'adminboard.js' file with the following JavaScript code:

```
1 import { useState, useEffect } from "react";
2 import { FaSun, FaMoon, FaUser, FaTachometerAlt, FaUsers, FaBuilding, FaTasks, FaEye } from "react-icons/fa";
3 import "./Admin.css";
4 import Allocation from "./allocation";
5 import Student from "./student2";
6 import Halls from "./hall";
7 import View from "./view";
8 import Staff from "./faculty";
9 import video2 from "../images/video2.mp4";
10 import video3 from "../images/video3.mp4";
11
12
13 const AdminDashboard1 = () => {
14   const [isDarkMode, setIsDarkMode] = useState(false);
15   const [currentDate, setCurrentDate] = useState("");
16   const [currentTime, setCurrentTime] = useState("");
17   const [activeTab, setActiveTab] = useState("dashboard");
18
19   useEffect(() => {
20     // Update the date and time every second
21     const interval = setInterval(() => {
22       const now = new Date();
23       setCurrentDate(now.toLocaleDateString());
24       setCurrentTime(now.toLocaleTimeString());
25     }, 1000);
26     return () => clearInterval(interval); // Cleanup interval on unmount
27   }, []);
28
29   const handleModeToggle = () => {
30     setIsDarkMode((prevMode) => !prevMode);
31   };
32
33   const handleTabClick = (tab) => {
34     setActiveTab(tab);
35   };
36
37   return (
```

OUTPUT:



Admin Panel

Welcome, Admin

Logout

Dashboard

Staff Management

Nr List

Branch wise list

year wise list

Hall Management

Allocation Process

View Allocation

Andhra Polytechnic Kakinada

Exam Seat Arrangement System

Date:

11/3/2025

Time:

9:13:00 pm

Faculty data

Id	FACULTY NAME	FACULTY DESIGNATION	DEPARTMENT
1001	parthasaradhi sir	head of department	Dcme
1002	Anasuya mam	senior lecturer	Dcme
1003	pavani mam	lecturer	Dcme
1004	pravalika mam	lecturer	Dcme
1005	sandhya rani mam	lecturer	Dcme
1006	Lakshmi devi mam	lecturer	Dcme

Admin Panel

Welcome, Admin

Logout

Dashboard

Staff Management

Nr List

Branch wise list

year wise list

Hall Management

Allocation Process

View Allocation

Andhra Polytechnic Kakinada

Exam Seat Arrangement System

Date:

11/3/2025

Time:

9:13:26 pm

Branch Wise List

Dcme

Dce

Dece

Deee

Dme

Id	Name	PIN NO	BRANCH	YEAR	SCHEME	DATE	TIME
1	A.gideon	22010-cm-001	Dcme	3rd year	c-20	1/4/2025	9:30
2	A.raju	22010-cm-002	Dcme	3rd year	c-20	01-04-2025	09:30
3	C. Singh	22010-cm-003	Dcme	3rd year	c-20	01-04-2025	09:30
4	D. Sharma	22010-cm-004	Dcme	3rd year	c-20	01-04-2025	09:30
5	E. Patel	22010-cm-005	Dcme	3rd year	c-20	01-04-2025	09:30
6	F. Gupta	22010-cm-006	Dcme	3rd year	c-20	01-04-2025	09:30
7	G. Mehta	22010-cm-007	Dcme	3rd year	c-20	01-04-2025	09:30
8	H. Nair	22010-cm-008	Dcme	3rd year	c-20	01-04-2025	09:30
9	I. Joshi	22010-cm-009	Dcme	3rd year	c-20	01-04-2025	09:30
10	J. Verma	22010-cm-010	Dcme	3rd year	c-20	01-04-2025	09:30
11	K. Kapoor	22010-cm-011	Dcme	3rd year	c-20	01-04-2025	09:30

Admin Panel

Welcome, Admin

Logout

Dashboard

Staff Management

Nr List

Branch wise list

year wise list

Hall Management

Allocation Process

View Allocation

Andhra Polytechnic Kakinada

Exam Seat Arrangement System

Date:

11/3/2025

Time:

9:13:44 pm

Student List by Branch & Year

Select Branch:

Dce

Select Year:

2 Year

Id	Name	PIN NO	BRANCH	YEAR	SCHEME	DATE	TIME
326	M.Modakalpana	23010-c-001	Dce	2nd year	c-23	02-04-2025	09:30
327	M.Maharshi	23010-c-002	Dce	2nd year	c-23	02-04-2025	09:30
328	N.Narayana	23010-c-003	Dce	2nd year	c-23	02-04-2025	09:30
329	N.Nikil	23010-c-004	Dce	2nd year	c-23	02-04-2025	09:30
330	K.Kalpana	23010-c-005	Dce	2nd year	c-23	02-04-2025	09:30
331	S.Sunitha	23010-c-006	Dce	2nd year	c-23	02-04-2025	09:30
332	G.sailendra	23010-c-007	Dce	2nd year	c-23	02-04-2025	09:30
333	G.Gopi	23010-c-008	Dce	2nd year	c-23	02-04-2025	09:30

Admin Panel

Welcome, Admin

Logout

Dashboard

Staff Management

Nr List

Branch wise list

year wise list

Hall Management

Allocation Process

View Allocation

Andhra Polytechnic Kakinada

Exam Seat Arrangement System

Date:

11/3/2025

Time:

9:14:15 pm

Hall list

Id	NAME	BRANCH	YEAR	SCHEME	CAPACITY	Actions
1					30	View
2					36	View
3					30	View
4	R4		18	6	36	View
5	R5		18	6	36	View
6	R6		15	5	30	View
7	R7		15	5	30	View

Seat Layout for R1

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

Admin Panel

Welcome, Admin

Logout

Dashboard

Staff Management

Nr List

Branch wise list

year wise list

Hall Management

Allocation Process

View Allocation

Faculty Allocation:

Name: Lakshmi devi mam
Designation: Supervisor 1

Name: veera babu sir
Designation: Supervisor 2

Seating Arrangement:
Hall ID: 3

22010-ec-023 Dece Row: 1, Col: 1	22010-ee-023 Dece Row: 1, Col: 2	22010-cm-023 Dcme Row: 1, Col: 3	22010-ec-024 Dece Row: 1, Col: 4	22010-ee-024 Dece Row: 1, Col: 5	22010-cm-024 Dcme Row: 1, Col: 6
22010-ee-025 Dece Row: 2, Col: 1	22010-cm-025 Dcme Row: 2, Col: 2	22010-ec-025 Dece Row: 2, Col: 3	22010-ee-026 Dece Row: 2, Col: 4	22010-cm-026 Dcme Row: 2, Col: 5	22010-ec-026 Dece Row: 2, Col: 6
22010-ec-027 Dece Row: 3, Col: 1	22010-ee-027 Dece Row: 3, Col: 2	22010-cm-027 Dcme Row: 3, Col: 3	22010-ec-028 Dece Row: 3, Col: 4	22010-ee-028 Dece Row: 3, Col: 5	22010-cm-028 Dcme Row: 3, Col: 6
22010-ee-029 Dece Row: 4, Col: 1	22010-cm-029 Dcme Row: 4, Col: 2	22010-ec-029 Dece Row: 4, Col: 3	22010-ee-030 Dece Row: 4, Col: 4	22010-cm-030 Dcme Row: 4, Col: 5	22010-ec-030 Dece Row: 4, Col: 6
22010-ec-031 Dece Row: 5, Col: 1	22010-ee-031 Dece Row: 5, Col: 2	22010-cm-031 Dcme Row: 5, Col: 3	22010-ec-032 Dece Row: 5, Col: 4	22010-ee-032 Dece Row: 5, Col: 5	22010-cm-032 Dcme Row: 5, Col: 6

Print

3/11/25, 9:18 PM

Go to any page between 1 and 1. (Ctrl+Alt+G)
about:blank

Hall Name: 3

Exam Date: 2025-04-01

Exam Time: 09:30

Faculty Allocation

Name	Designation
Lakshmi devi mam	Supervisor 1
veera babu sir	Supervisor 2

Seating Arrangement

Col 1	Col 2	Col 3	Col 4	Col 5	Col 6
22010-ec-023	22010-ee-023	22010-cm-023	22010-ec-024	22010-ee-024	22010-cm-024
22010-ec-025	22010-cm-025	22010-ec-025	22010-ec-026	22010-cm-026	22010-ec-026
22010-ec-027	22010-ee-027	22010-cm-027	22010-ec-028	22010-ee-028	22010-cm-028
22010-ec-029	22010-cm-029	22010-ec-029	22010-ee-030	22010-cm-030	22010-ec-030
22010-ec-031	22010-ec-031	22010-cm-031	22010-ec-032	22010-ec-032	22010-cm-032