SMARTLAB ASSISTANT

-- a virtual lab

Team Members:

- **21A91A04L8 - Backend Developer**
- **21A91A04I5 - Frontend Developer**
- **21A91A04J7 - Backend Developer**
- **21A91A04O9 - Frontend Developer**
- **21MH1A4905 - Cloud Engineer**
- **21MH1A4959 - Cloud Engineer**

OBJECTIVE:

The Project Smart Lab Assistant is designed to enhance students' laboratory knowledge through a comprehensive web application. It consists of several key components aimed at providing detailed explanations of experiments, facilitating communication through a chatbot for doubt clarification, and offering a contact page for user feedback and inquiries.

VIRTUAL LAB

In the realm of modern education, practical learning in laboratories plays a pivotal role in shaping students' understanding of scientific principles and concepts. However, accessing and comprehending lab experiments can often be challenging due to various constraints. To bridge this gap, we introduce the Smart Lab Assistant an innovative web application designed to revolutionize the way students interact with laboratory experiments and enhance their educational experience.

Key Features:

1. Experiment Repository: A centralized database housing a diverse collection of laboratory experiments across different disciplines. Each experiment is meticulously documented with detailed explanations, procedures, and theoretical backgrounds.

2. Detailed Explanations: In-depth guides accompanying each experiment, featuring step-by-step procedures, safety precautions, and explanations of underlying scientific principles. Visual aids such as diagrams and videos enrich the learning experience.

3. Chatbot for Doubt Clarification: An intelligent chatbot integrated within the platform to address students' queries in real-time. Utilizing natural language processing (NLP), the chatbot provides instant responses and clarifications based on the experiment repository and additional resources.

4. User Authentication: Secure login capabilities allowing registered users to personalize their experience, save preferences, track progress, and receive tailored recommendations. This feature ensures data security and enhances user engagement.

5. Contact Page: A dedicated channel for users to provide feedback, report issues, or inquire about the application. This facilitates continuous improvement based on user input and enhances customer satisfaction.

Benefits for Students:

Accessibility: 24/7 access to a wealth of educational resources, enabling students to study at their own pace and convenience.

Comprehensive Learning: Detailed explanations and interactive elements foster deeper understanding of scientific principles beyond traditional classroom settings.

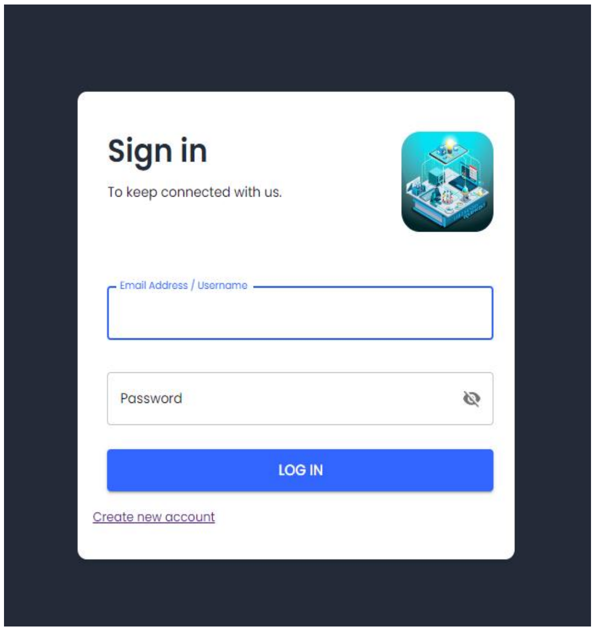
Support and Engagement: Real-time assistance through the chatbot and a user-friendly interface encourage active learning and interaction with course materials.

PDF Viewer for Experiment Documents: Integration of a PDF viewer allowing users to view, download, and print detailed PDF documents related to each experiment. This feature provides offline access and facilitates comprehensive study and preparation.

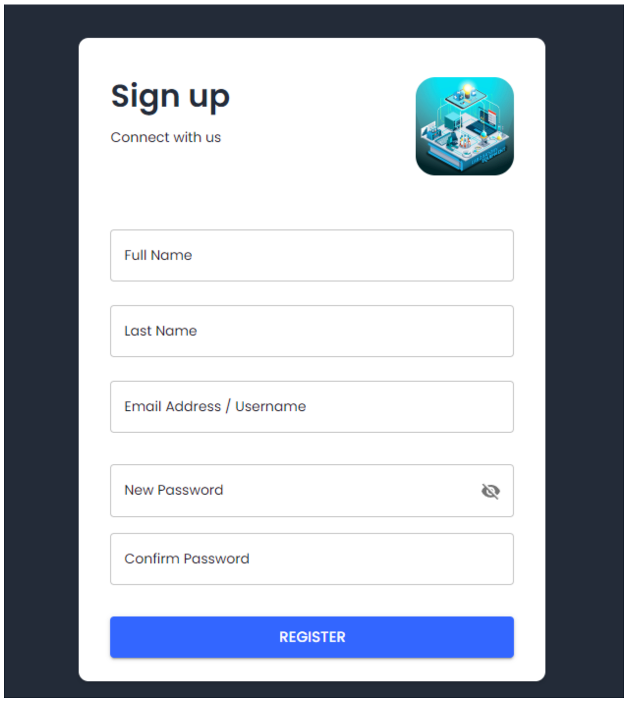
Key Components and Features of Smart Lab Assistant

1. User Authentication:

- **1.1. Login**: The Smart Lab Assistant provides a secure login system, allowing students and educators to access personalized content and track their progress. Users can log in using their credentials to gain access to all features of the application.

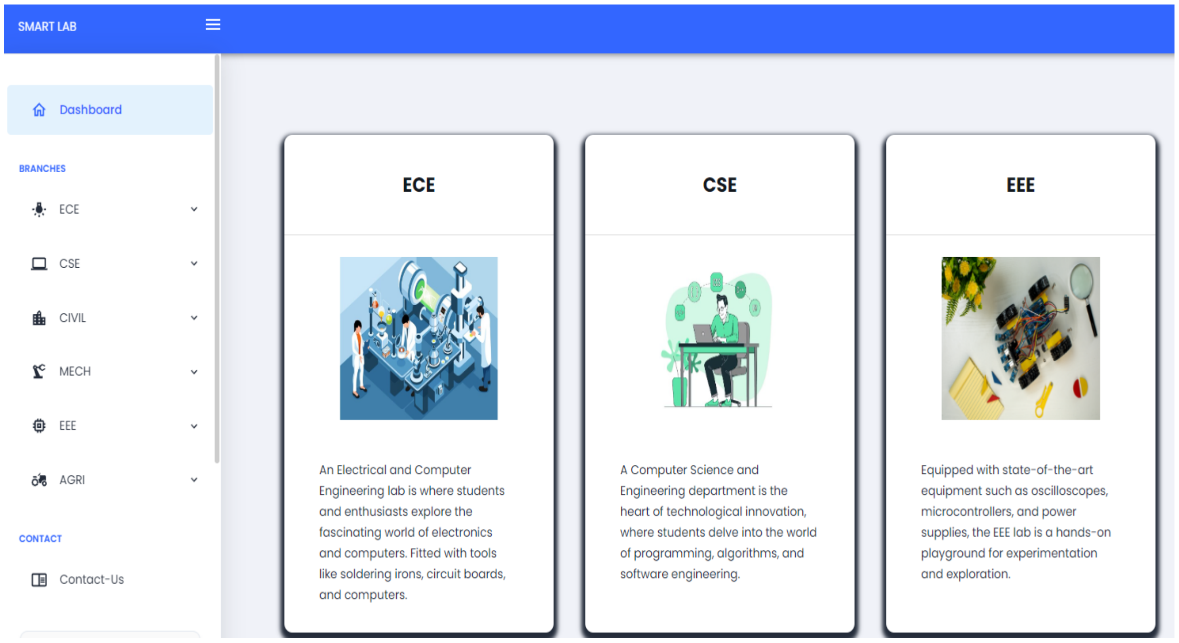


- **1.2. Registration**: New users can easily create an account by providing necessary details such as name, email, and password. The registration process ensures that only authorized users can access the platform's resources.



2. Dashboard:

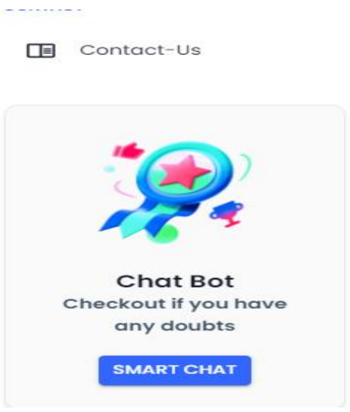
- **2.1. Subject Details**: The dashboard includes detailed information about various subjects, allowing users to navigate through different topics and experiments easily. Each subject section provides a comprehensive overview of available experiments and related resources.



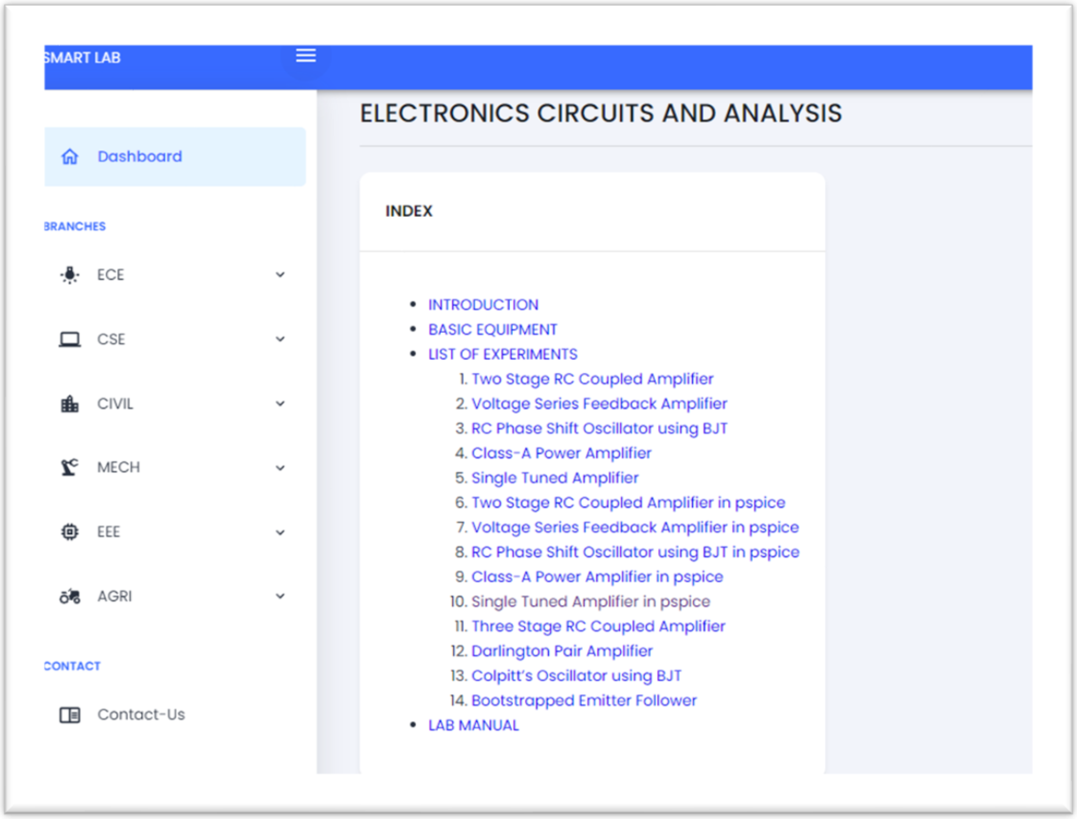
An Electrical and Computer Engineering lab is where students and enthusiasts explore the fascinating world of electronics and computers. Fitted with tools like soldering irons, circuit boards, and computers.

A Computer Science and Engineering department is the heart of technological innovation, where students delve into the world of programming, algorithms, and software engineering.

Equipped with state-of-the-art equipment such as oscilloscopes, microcontrollers, and power supplies, the EEE lab is a hands-on playground for experimentation and exploration.

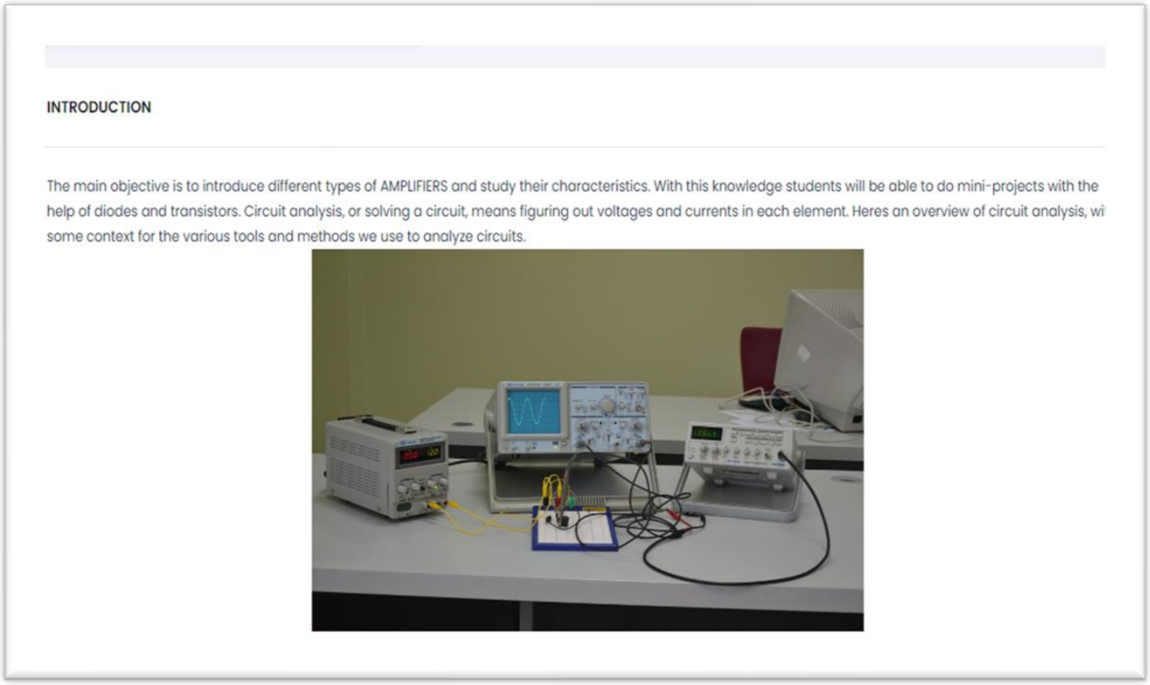


- **2.2. Menu**: The intuitive menu on the dashboard facilitates easy navigation across the platform's features.
  - **2.2.1. Branches and Their Labs**: This section categorizes experiments based on different branches of science and their respective laboratories. Users can explore experiments specific to their branch, making it easier to find relevant content.



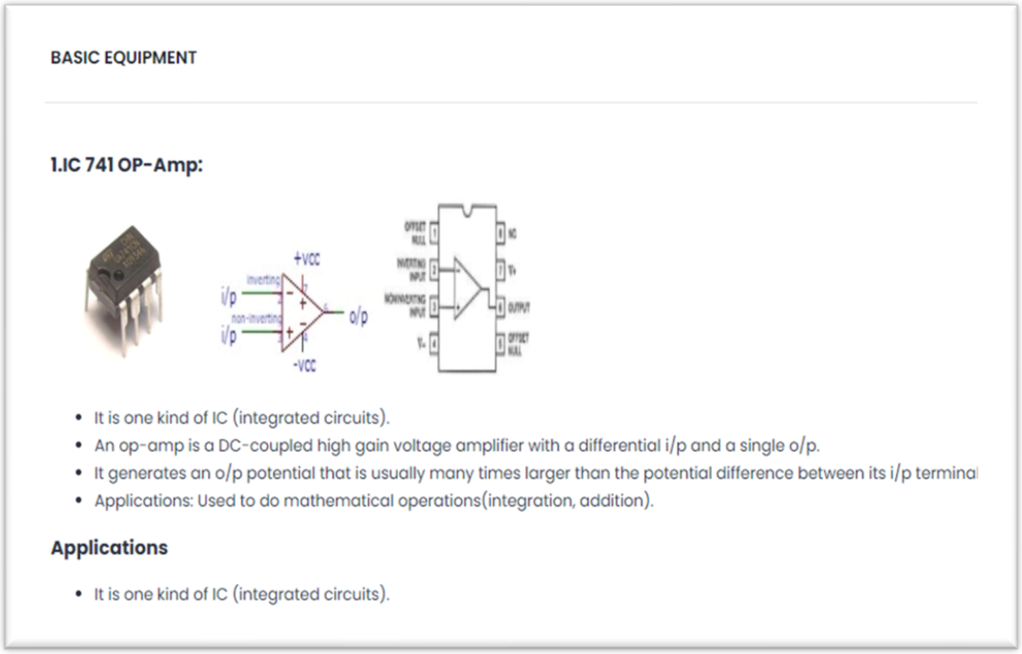
ELECTRONICS CIRCUITS AND ANALYSIS

INDEX



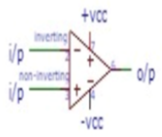
INTRODUCTION

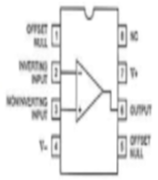
The main objective is to introduce different types of AMPLIFIERS and study their characteristics. With this knowledge students will be able to do mini-projects with the help of diodes and transistors. Circuit analysis, or solving a circuit, means figuring out voltages and currents in each element. Heres an overview of circuit analysis, wi
some context for the various tools and methods we use to analyze circuits.



BASIC EQUIPMENT

1.IC 741 OP-Amp:

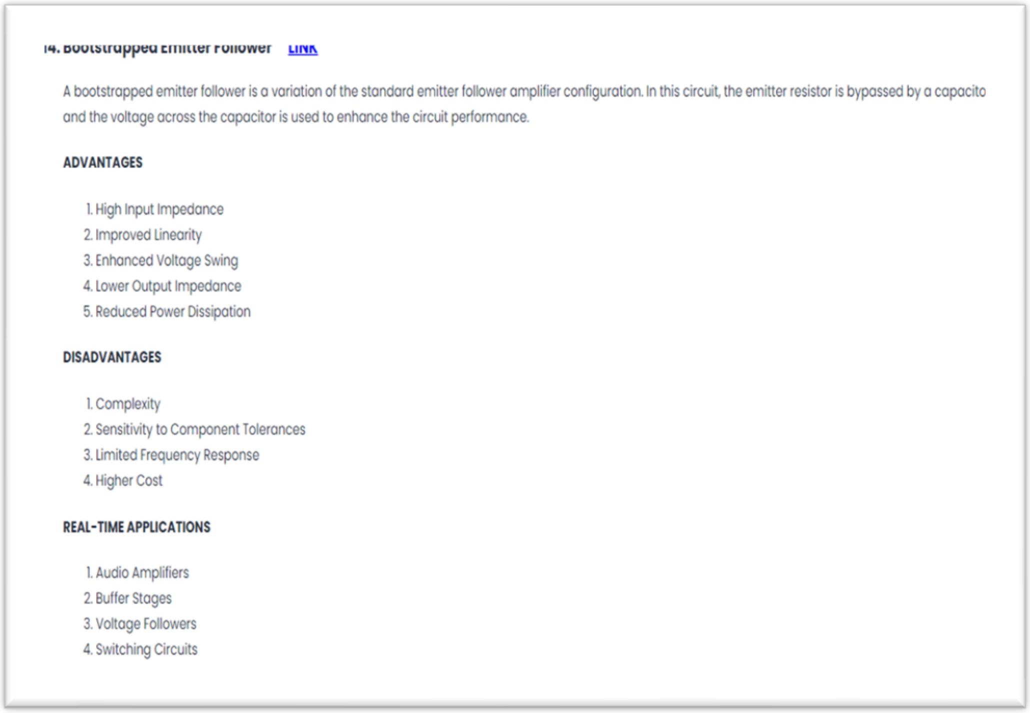




- It is one kind of IC (integrated circuits).
- An op-amp is a DC-coupled high gain voltage amplifier with a differential i/p and a single o/p.
- It generates an o/p potential that is usually many times larger than the potential difference between its i/p terminal
- Applications: Used to do mathematical operations(integration, addition).

Applications

- It is one kind of IC (integrated circuits).



14. Bootstrapped Emitter Follower LINK

A bootstrapped emitter follower is a variation of the standard emitter follower amplifier configuration. In this circuit, the emitter resistor is bypassed by a capacito and the voltage across the capacitor is used to enhance the circuit performance.

ADVANTAGES

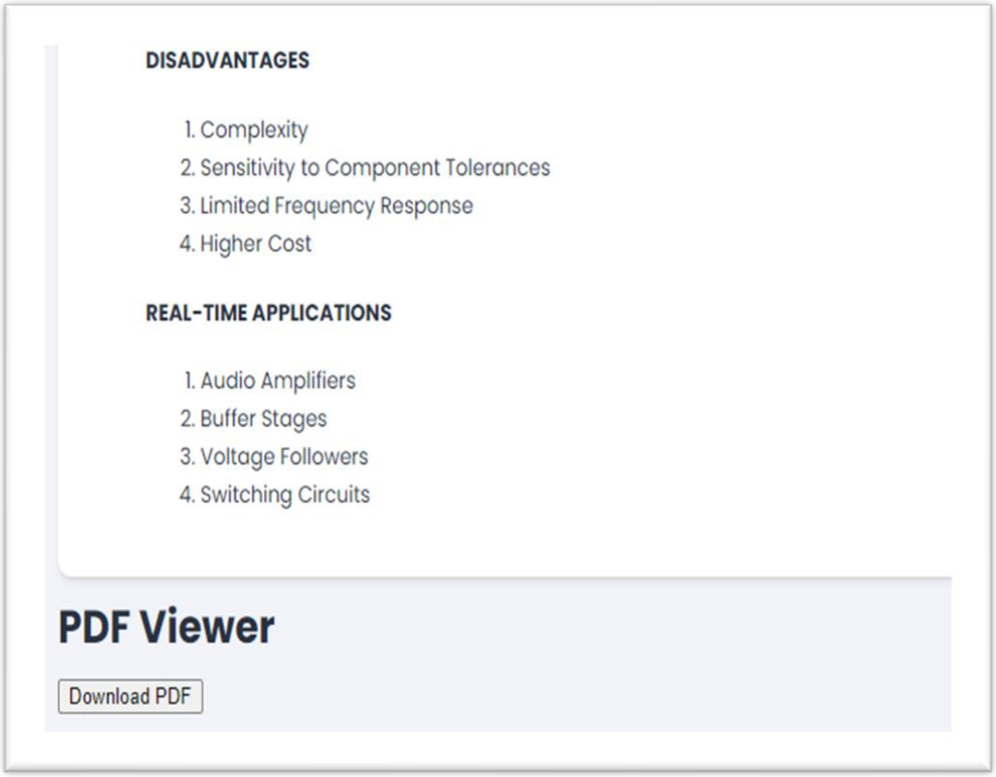
1. High Input Impedance
2. Improved Linearity
3. Enhanced Voltage Swing
4. Lower Output Impedance
5. Reduced Power Dissipation

DISADVANTAGES

1. Complexity
2. Sensitivity to Component Tolerances
3. Limited Frequency Response
4. Higher Cost

REAL-TIME APPLICATIONS

1. Audio Amplifiers
2. Buffer Stages
3. Voltage Followers
4. Switching Circuits



DISADVANTAGES

1. Complexity
2. Sensitivity to Component Tolerances
3. Limited Frequency Response
4. Higher Cost

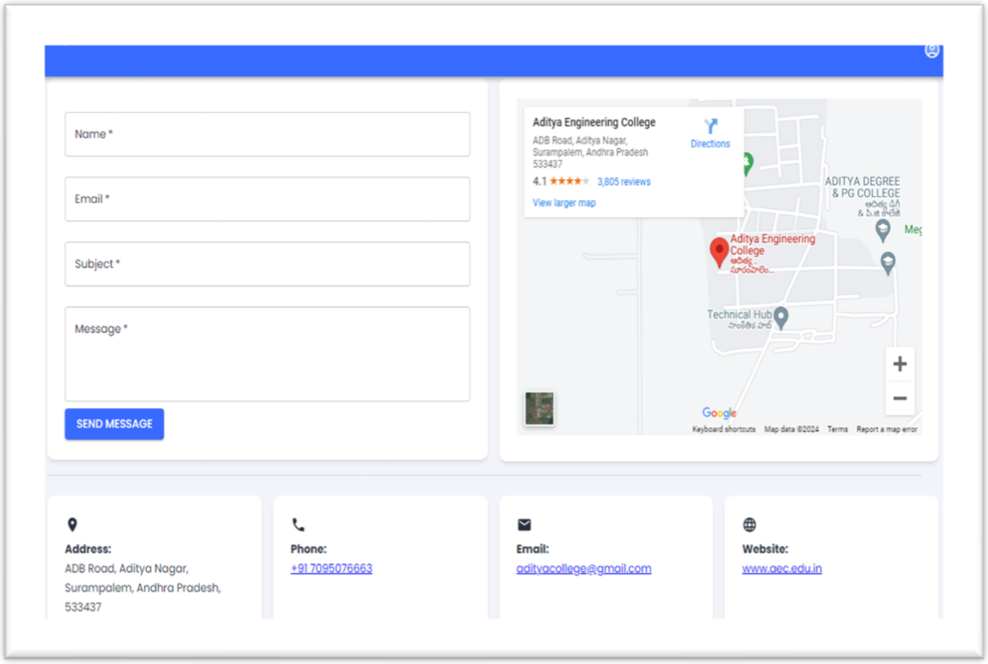
REAL-TIME APPLICATIONS

1. Audio Amplifiers
2. Buffer Stages
3. Voltage Followers
4. Switching Circuits

PDF Viewer

Download PDF

- **2.2.2. Contact Us**: The contact page allows users to provide feedback, report issues, or make inquiries about the web application. It includes a form where users can submit their questions or comments, which are directed to the support team for response.



Name *

Email *

Subject *

Message *

SEND MESSAGE

Aditya Engineering College
ADB Road, Aditya Nagar, Surampalem, Andhra Pradesh 533437
4.1 ★★★★ 3,805 reviews
Directions
View larger map
ADITYA DEGREE & PG COLLEGE
Aditya Engineering College
Technical Hub
Google
Keyboard shortcuts
Map data ©2024
Terms
Report a map error

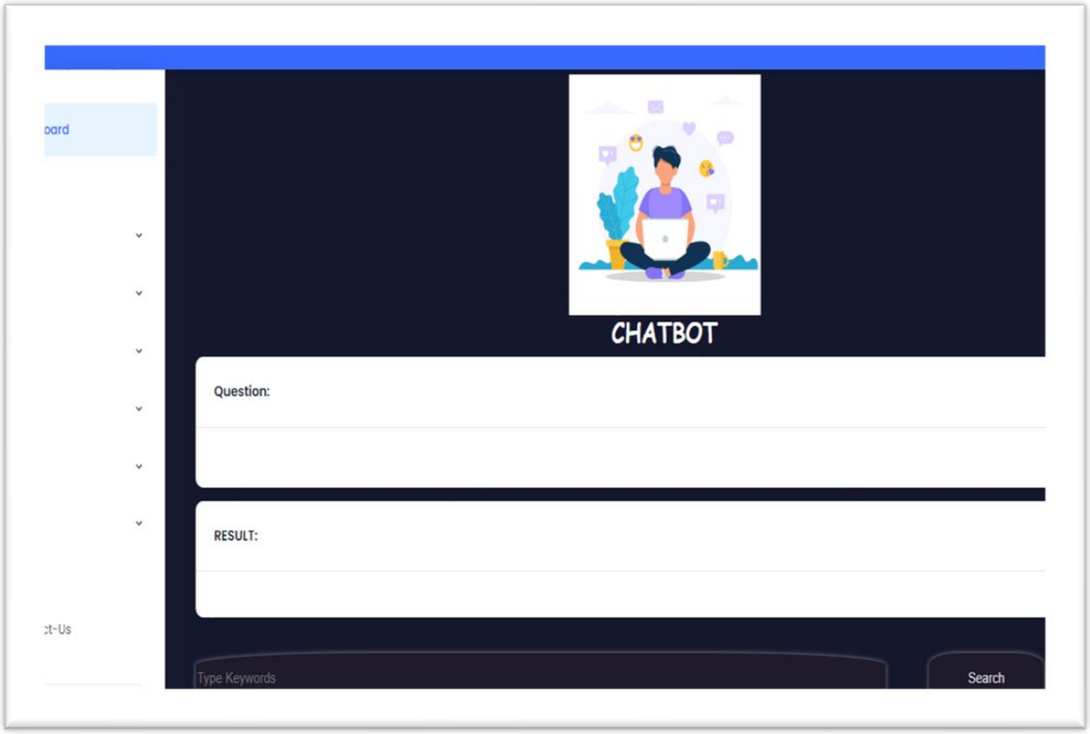
Address: ADB Road, Aditya Nagar, Surampalem, Andhra Pradesh, 533437

Phone: +91 7095076663

Email: adityacollege@gmail.com

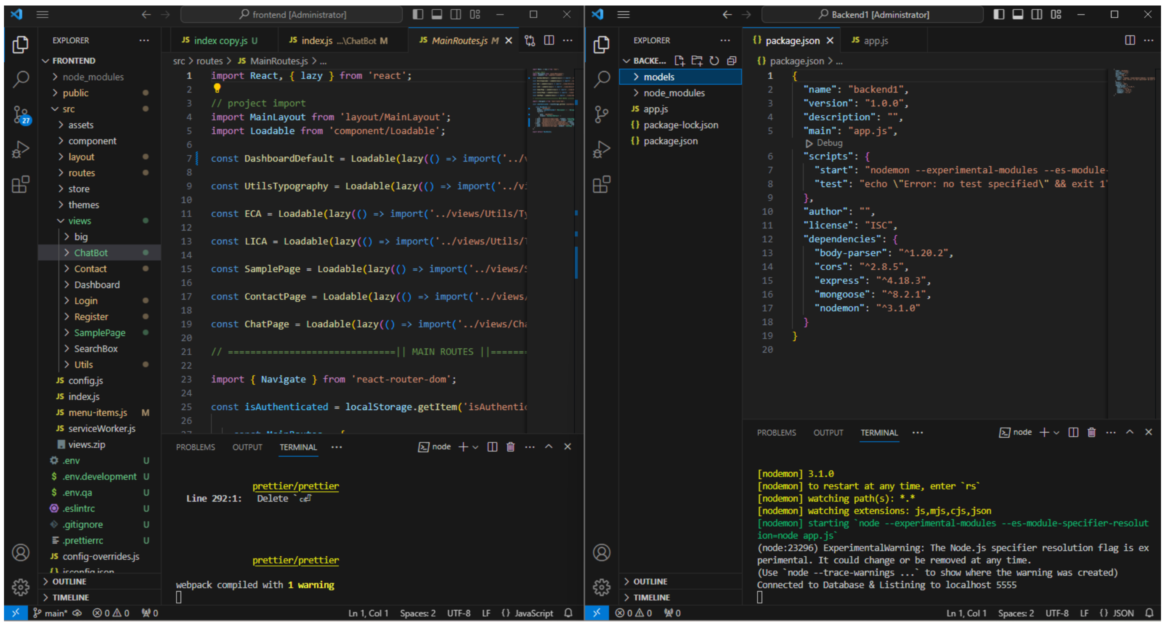
Website: www.aec.edu.in

- **2.2.3. Chat Bot**: Integrated with OpenAI, the chatbot provides instant, AI-driven responses to student inquiries. This feature supports real-time doubt clearing, helping students understand complex concepts and procedures.



Development Platform: Visual Studio Code

The Smart Lab Assistant is developed using Visual Studio Code (VS Code), a versatile and lightweight code editor that supports multiple programming languages and frameworks. VS Code's extensive features and extensions enhance productivity and streamline the development process.



```
import React, { lazy } from 'react';

// project import
import MainLayout from 'layout/MainLayout';
import Loadable from 'component/Loadable';

const DashboardDefault = Loadable(lazy(() => import('../v

const UtilsTypography = Loadable(lazy(() => import('../vi

const ECA = Loadable(lazy(() => import('../views/Utils/Ty

const LICA = Loadable(lazy(() => import('../views/Utils/T

const SamplePage = Loadable(lazy(() => import('../views/S

const ContactPage = Loadable(lazy(() => import('../views/

const ChatPage = Loadable(lazy(() => import('../views/Cha

// ==============================|| MAIN ROUTES ||======

import { Navigate } from 'react-router-dom';

const isAuthenticated = localStorage.getItem('isAuthentic
```

```
          prettier/prettier
  Line 292:1:   Delete `⏎`



          prettier/prettier

webpack compiled with 1 warning
```

```
{
  "name": "backend1",
  "version": "1.0.0",
  "description": "",
  "main": "app.js",
  "scripts": {
    "start": "nodemon --experimental-modules --es-module-
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "body-parser": "^1.20.2",
    "cors": "^2.8.5",
    "express": "^4.18.3",
    "mongoose": "^8.2.1",
    "nodemon": "^3.1.0"
  }
}
```

```
[nodemon] 3.1.0
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node --experimental-modules --es-module-specifier-resolut
ion=node app.js`
(node:23296) ExperimentalWarning: The Node.js specifier resolution flag is ex
perimental. It could change or be removed at any time.
(Use `node --trace-warnings ...` to show where the warning was created)
Connected to Database & Listining to localhost 5555
```

Technology Stack

- **Front-end Technologies:**
    1. The frontend of the Smart Lab Assistant is developed using React.js, a powerful JavaScript library that enables the creation of dynamic and responsive user interfaces through reusable components and a virtual DOM
    2. This ensures a smooth, efficient, and interactive user experience across devices.

- **Back-end Technologies:**
    1. The backend of the Smart Lab Assistant uses Node.js with Express.js, providing a robust environment for handling asynchronous operations and building scalable APIs.
    2. This setup ensures efficient data processing and real-time interaction capabilities

- **Data Base:**
    1. The Smart Lab Assistant utilizes MongoDB, a flexible NoSQL database that stores data in JSON-like documents, allowing for dynamic and scalable data handling.
    2. This ensures efficient storage and retrieval of diverse and evolving data structures.

- **Artificial Intelligence**
    1. The Smart Lab Assistant integrates an OpenAI-powered chatbot to provide instant, AI-driven responses to student inquiries.
    2. This enhances the learning experience by offering real-time doubt clearing and support.

- **DevOps :**
    1. The Smart Lab Assistant is hosted on AWS, leveraging its scalable cloud infrastructure to ensure high availability and performance.
    2. AWS services provide secure and reliable storage, compute power, and network capabilities for the application.
    3. Docker containers enable containerized deployments for microservices architecture, promoting scalability and portability.

- The chosen technologies contribute to:
    - **Enhanced User Experience:** Modern front-end frameworks ensure a responsive and user-friendly interface.
    - **Scalability and Performance:** Back-end technologies provide a robust foundation for handling large user bases and project data.
    - **Efficient Development & Maintenance:** DevOps practices streamline development workflows and ensure faster deployments.

React Packages:

1. **react-router-dom**: This module provides routing capabilities for React applications, including components like `BrowserRouter`, `Route`, `Switch`, and `Link` to enable navigation between different views or pages within the application.
2. **axios**: Axios is a popular HTTP client for making asynchronous HTTP requests in React applications. It simplifies the process of sending and handling HTTP requests to backend servers or APIs.
3. **redux** and **react-redux**: Redux is a predictable state container for JavaScript apps, commonly used with React for managing application state. `react-redux` provides bindings to use Redux with React components, enabling efficient data flow and state management across the application.
4. **styled-components**: This module allows you to write CSS-in-JS in your React components. It enables the creation of styled components with scoped styles that are easy to maintain and reuses.
5. **react-icons**: This library offers a collection of popular icons as React components, allowing easy integration of icons from various icon sets into your application.

React Modules:

1. **Components**: Modular building blocks that encapsulate UI elements and logic, promoting reusability and maintainability.
2. **JSX**: Syntax extension for JavaScript that allows HTML-like code within React components, facilitating the creation of UI elements.
3. **Props**: Short for properties, props are used to pass data from one component to another, enabling dynamic and flexible component behavior.
4. **State**: State allows components to manage and store internal data, enabling them to update and render based on changing conditions or user interactions.
5. **Hooks**: Introduced in React 16.8, hooks like `useState` and `useEffect` provide a way to use state and lifecycle features in functional components, improving code organization and reusability.

HTTP Methods:

HTTP methods provide a standardized way for clients (such as web browsers or mobile apps) to interact with backend servers and perform CRUD (Create, Read, Update, Delete) operations on resources. Proper use and adherence to HTTP methods help maintain clarity, consistency, and scalability in API design and development.

- **GET**: Retrieves data from a specified resource.

- **POST**: Submits data to be processed to a specified resource.

- **PUT**: Updates a specified resource with new data.

- **DELETE**: Deletes a specified resource.

Backend Packages:

Express: This is the core web application framework for Node.js. It simplifies the process of handling HTTP requests, defining routes, and managing middleware. Express is crucial for building robust APIs and web applications.

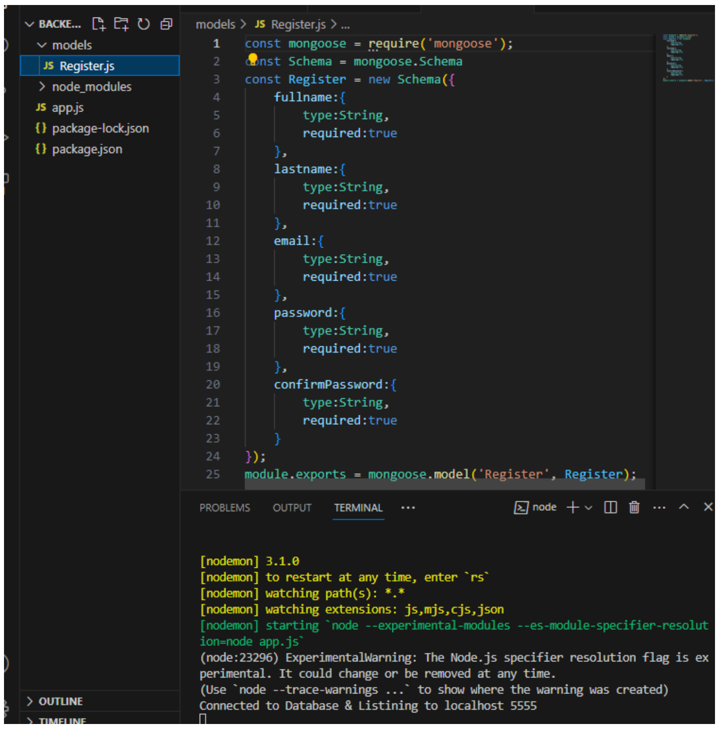
Mongoose: Mongoose is an ODM (Object-Document Mapper) for MongoDB and Node.js. It provides a schema-based solution to model application data, perform CRUD operations, and interact with MongoDB databases using JavaScript objects.

cors: CORS (Cross-Origin Resource Sharing) middleware allows control over how resources from different origins (domains) can interact with your server. It's essential for handling requests from frontend applications that are hosted on different domains than your backend.

body-parser: While included in Express.js as `express.json()` in newer versions, `body-parser` was traditionally used to parse incoming request bodies in middleware. It's crucial for accessing POST, PUT, and PATCH request parameters or JSON payloads in your routes.

Schemas (MongoDB):

1. **Mongoose Schema**:In a Node.js application using MongoDB with Mongoose, schemas define the structure of documents within a collection. They specify the fields, their types, validation rules, and default values.



```
const mongoose = require('mongoose');
const Schema = mongoose.Schema
const Register = new Schema({
    fullname:{
        type:String,
        required:true
    },
    lastname:{
        type:String,
        required:true
    },
    email:{
        type:String,
        required:true
    },
    password:{
        type:String,
        required:true
    },
    confirmPassword:{
        type:String,
        required:true
    }
});
module.exports = mongoose.model('Register', Register);
```

```
[nodemon] 3.1.0
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node --experimental-modules --es-module-specifier-resolut
ion=node app.js`
(node:23296) ExperimentalWarning: The Node.js specifier resolution flag is ex
perimental. It could change or be removed at any time.
(Use `node --trace-warnings ...` to show where the warning was created)
Connected to Database & Listining to localhost 5555
```

EC2 Instance Connection:

Set Up an EC2 Instance: Begin by launching an EC2 instance using an appropriate Amazon Machine Image (AMI), such as Ubuntu. Configure security groups to allow necessary traffic (typically HTTP and SSH).

Connect to the Instance: Use SSH to connect to your EC2 instance securely**.**

Install Necessary Software: On the EC2 instance, install essential software like Git, Node.js, npm, and a web server like Nginx.

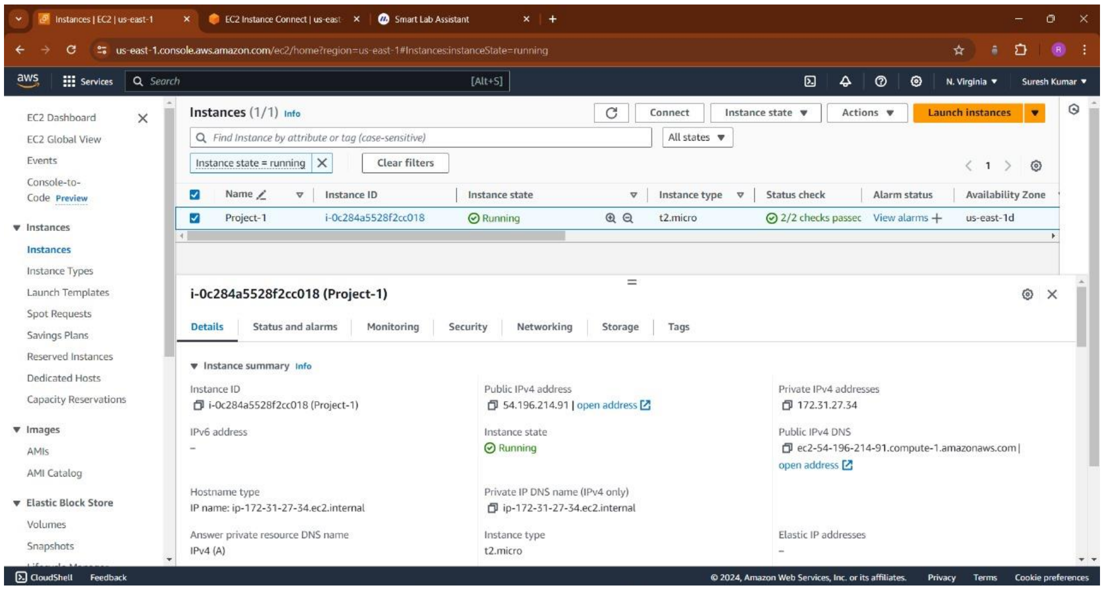
Clone Your React App: Use Git to clone your React application repository from a version control system like GitHub.

Build Your React App: Navigate to your application's directory and use npm to install dependencies and build the React application for production.

Configure the Web Server: Set up the web server (Nginx) to serve your React application's build files. This typically involves modifying configuration files to ensure Nginx points to your app's build directory.

Deploy and Test: Start the web server and navigate to your EC2 instance's public IP or DNS in a web browser to ensure the React app is running correctly.

Optimize for Production: For a production deployment, consider additional steps like setting up a process manager (e.g., PM2) to keep your application running, configuring SSL certificates for secure HTTPS access, and using a domain name for your application.



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Project-1	i-0c284a5528f2cc018	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1d

i-0c284a5528f2cc018 (Project-1)

Instance summary Info

Instance ID
i-0c284a5528f2cc018 (Project-1)

Public IPv4 address
54.196.214.91 | open address

Private IPv4 addresses
172.31.27.34

IPv6 address
–

Instance state
Running

Public IPv4 DNS
ec2-54-196-214-91.compute-1.amazonaws.com | open address

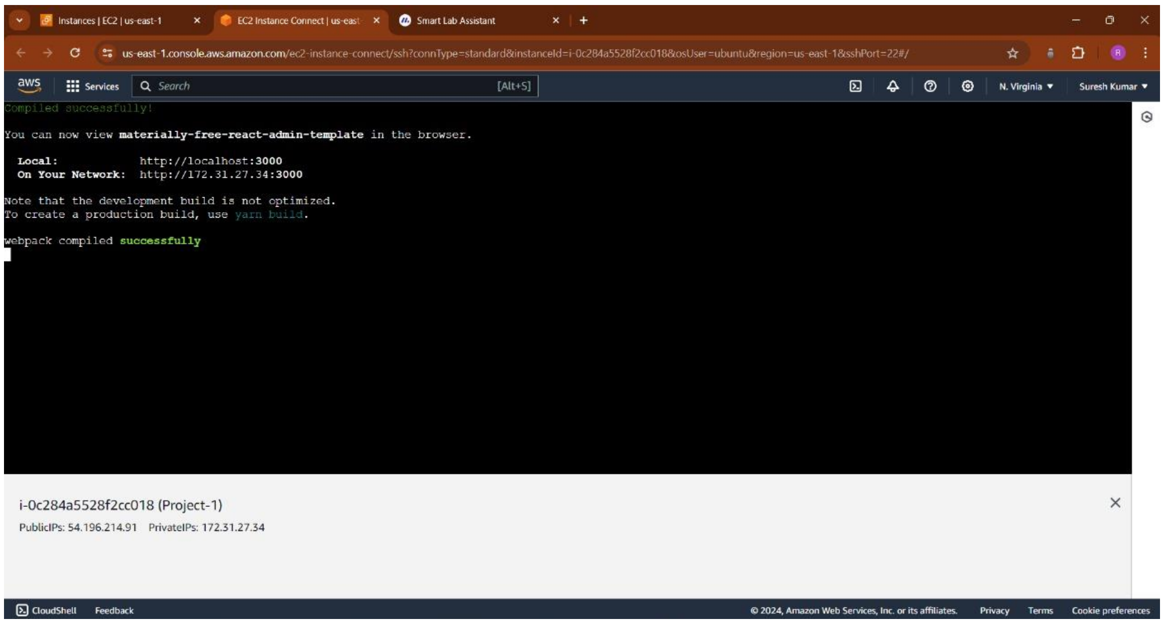
Hostname type
IP name: ip-172-31-27-34.ec2.internal

Private IP DNS name (IPv4 only)
ip-172-31-27-34.ec2.internal

Answer private resource DNS name
IPv4 (A)

Instance type
t2.micro

Elastic IP addresses
–



Instances | EC2 | us-east-1
EC2 Instance Connect | us-east-
Smart Lab Assistant
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0c284a5528f2cc018&osUser=ubuntu&region=us-east-1&sshPort=22#/
Services
Search
[Alt+S]
N. Virginia
Suresh Kumar

```
Compiled successfully!

You can now view materially-free-react-admin-template in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://172.31.27.34:3000

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

webpack compiled successfully
```

i-0c284a5528f2cc018 (Project-1)
PublicIPs: 54.196.214.91 PrivateIPs: 172.31.27.34

CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Front-end and Back-end Installation

- **npx create-react-app app_name**
- **npm config set legacy-peer-deps**
- **npm i --legacy-peer-deps**
- **npm cache clean --force**
- **npm i styled-components**
- **npm i react-router-dom**
- **npm i axios**
- **npm i react-bootstrap bootstrap**
- **npm init**
- **npm i express**
- **npm i mongoose**
- **npm i cors**
- **npm i body-parser**

Conclusion:

Smart Lab Assistant represents a comprehensive solution aimed at enhancing students' laboratory learning experiences through innovative web technologies. By integrating features such as an extensive experiment repository, detailed explanations, a chatbot for clearing doubts, and user-friendly interfaces, the application facilitates accessible and interactive learning. Through continuous improvement and user feedback mechanisms, Smart Lab Assistant strives to empower students with in-depth scientific knowledge and practical skills, making complex concepts more understandable and engaging.