Experiment 1

Student Name: Prabhav Poddar **UID: 22BCS11373**
Branch: CSE **Section/Group: KRG IOT2B**
Semester: 6th **Date of Performance: 16-1-25**
Subject Name: AP LAB-II **Subject Code: 22CSP-351**

1. Aim:

Full Stack Development (MERN). The primary aim of this experiment is to provide students or developers with an understanding of full-stack development involving MongoDB, Node.js, React, and Express.

1. Problem 1.1.1: Give understanding of MongoDB, Nodejs, React, Express.
2. Problem 1.1.2: Create a Frontend design of Login/Signup pages and create a backend of it.
3. Problem 1.1.3: Test the Backend API Using Postman

2. Objective:

- Understand the fundamentals of MongoDB, Node.js, React, and Express
- Create a functional frontend for Login/Signup pages
- Develop a backend using Express and MongoDB
- Test the backend API using Postman

3. Implementation/Code:

Backend:

- mkdir backend cd backend
- npm init -y npm install
- express mongoose cors bcryptjs jsonwebtoken

server.js

```
import express from 'express'
import cors from 'cors'
const app = express();
const corsoption = {
    origin : "*",
    Credentials:true
}
app.use(cors(corsoption))
 //Routing in Express
app.get('/',(req,res)=>{
    res.send("Your are requested for Home Route")
})
app.post('/CU',(req,res)=>{
    res.send("This CU project Learning")
})
const prot = 2000;
app.listen(prot, ()=> console.log(`server is Running ${prot}`))
```

DB.js

```
import mongoose from "mongoose";
export const connectDB = async()=>{
    try{
        console.log("DB is Connected");
    } catch(error){
        console.log(error);
    }
}
```

authRoutes.js // Signup Route

```
router.post('/signup', async (req, res) => {

const { email, password } = req.body;
```

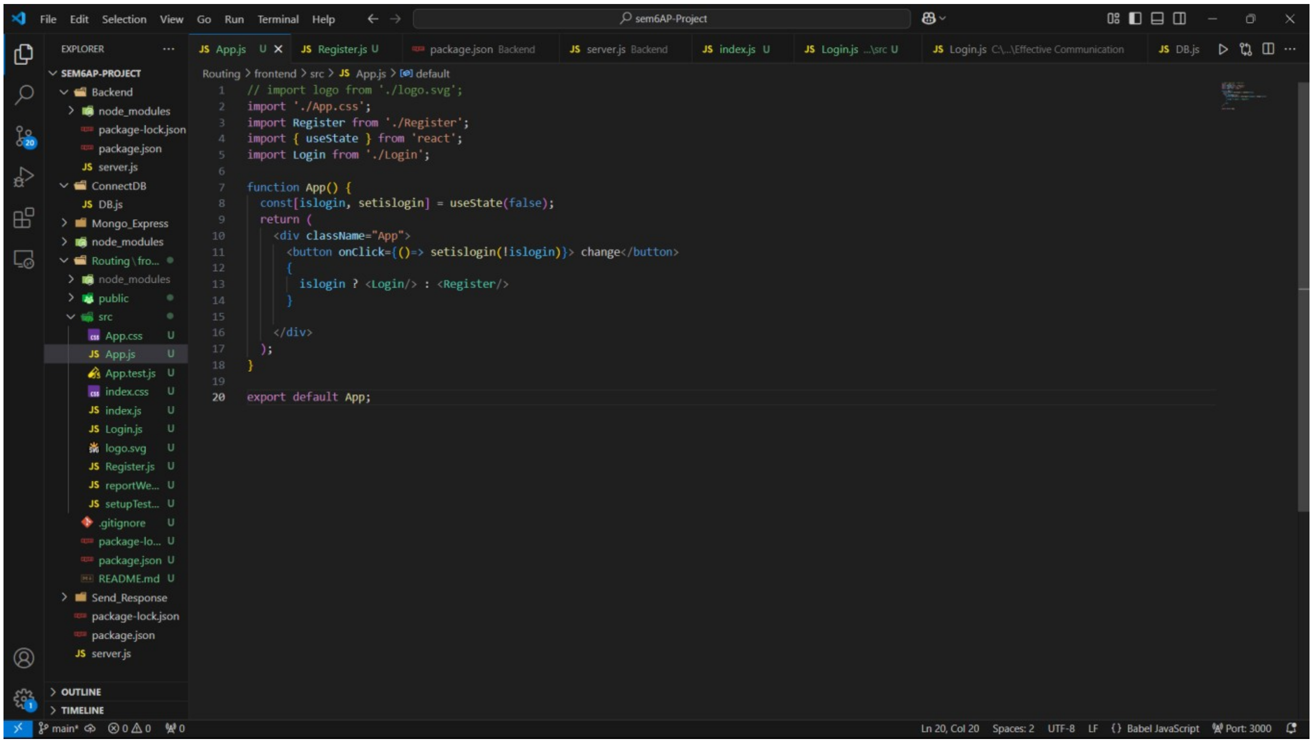
```
try {
    const existingUser = await User.findOne({ email });
   if (existingUser) {
 return res.status(400).json({ message: 'User already exists' });
    }
    const hashedPassword = await bcrypt.hash(password, 10);
   const newUser = new User({ email, password: hashedPassword });
   await newUser.save();
    res.status(201).json({ message: 'User created successfully' });
   }
   catch (error) {
   console.error('Signup error:', error); // Log error to console
    res.status(500).json({ message: 'Server error', error: error.message }); // Include the error
   message in the response
    }   });
    router.post('/login', async (req, res) => {
   const { email, password } = req.body;
       try {
  const user = await User.findOne({ email });
       if (!user) {        return res.status(404).json({ message: 'User not
    found' });
       }
    const isMatch = await bcrypt.compare(password, user.password);
if (!isMatch) {        return res.status(400).json({ message: 'Invalid credentials'

}); }

      const token = jwt.sign({ userId: user._id }, process.env.JWT_SECRET, { expiresIn: '1h' });
res.json({ message: 'Login successful', token });

    } catch (error) {       console.error('Login error:', error); // Log error

to console       res.status(500).json({ message: 'Server error', error:

error.message }); // Include the error message in the response

    }   });
```

my-auth-app:

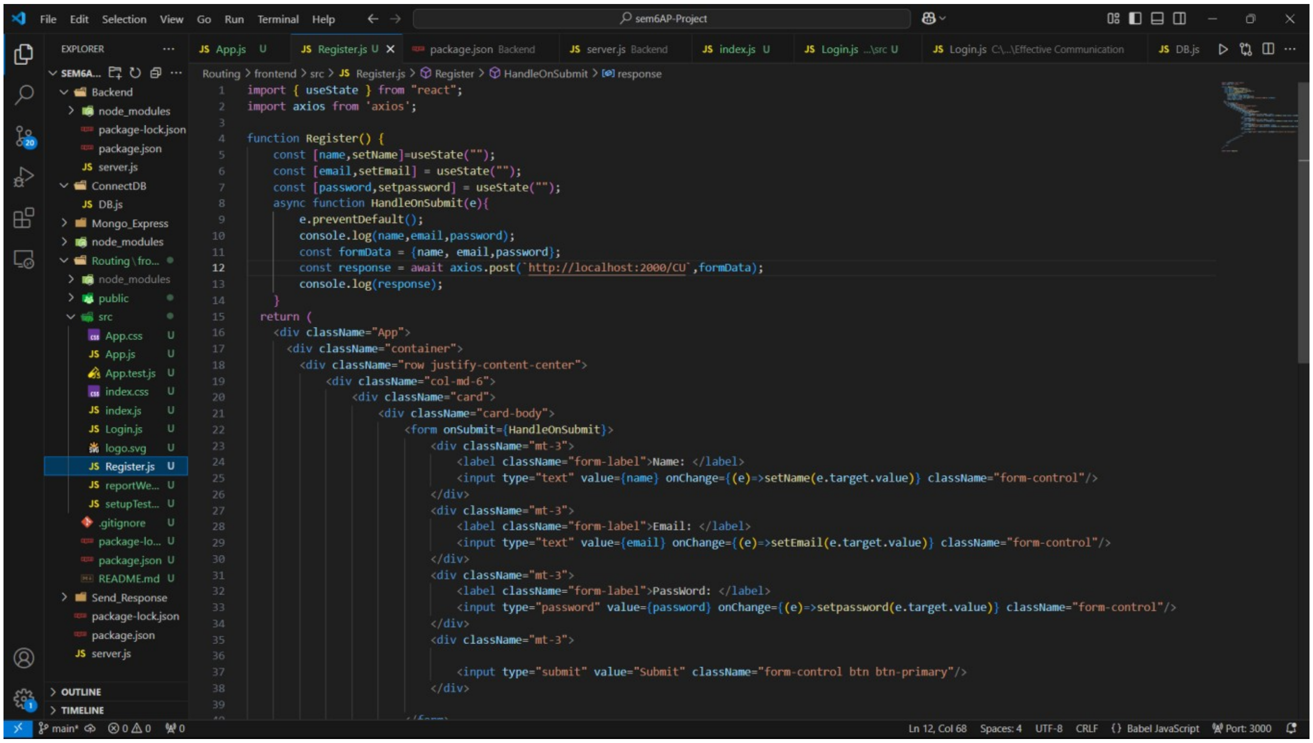
- npm create vite@latest my-auth-app --template react
- cd my-auth-app
- npm install

App.jsx:

```
// import logo from './logo.svg';

import './App.css';

import Register from './Register';

import { useState } from 'react';

import Login from './Login';

function App() {

  const[islogin, setislogin] =

useState(false);

  return (
    <div className="App">
      <button onClick={()=>
setislogin(!islogin)}>
change</button>
      {
        islogin ? <Login/> :
<Register/>
      }
    </div>
  );
}
export default App;
```

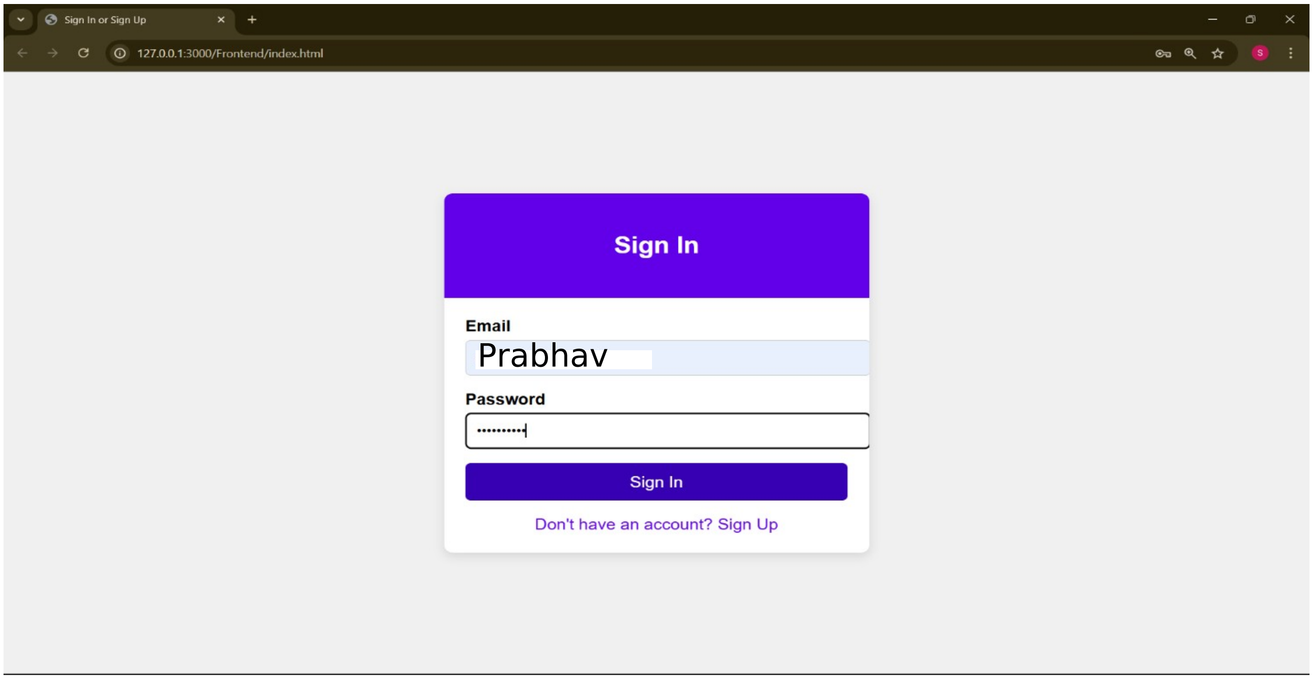


```
// import logo from './logo.svg';
import './App.css';
import Register from './Register';
import { useState } from 'react';
import Login from './Login';

function App() {
  const[islogin, setislogin] = useState(false);
  return (
    <div className="App">
      <button onClick={()=> setislogin(!islogin)}> change</button>
      {
        islogin ? <Login/> : <Register/>
      }

    </div>
  );
}

export default App;
```



```
import { useState } from "react";
import axios from 'axios';

function Register() {
    const [name,setName]=useState("");
    const [email,setEmail] = useState("");
    const [password,setpassword] = useState("");
    async function HandleOnSubmit(e){
        e.preventDefault();
        console.log(name,email,password);
        const formData = {name, email,password};
        const response = await axios.post(`http://localhost:2000/CU`,formData);
        console.log(response);
    }
  return (
    <div className="App">
      <div className="container">
        <div className="row justify-content-center">
            <div className="col-md-6">
                <div className="card">
                    <div className="card-body">
                        <form onSubmit={HandleOnSubmit}>
                            <div className="mt-3">
                                <label className="form-label">Name: </label>
                                <input type="text" value={name} onChange={(e)=>setName(e.target.value)} className="form-control"/>
                            </div>
                            <div className="mt-3">
                                <label className="form-label">Email: </label>
                                <input type="text" value={email} onChange={(e)=>setEmail(e.target.value)} className="form-control"/>
                            </div>
                            <div className="mt-3">
                                <label className="form-label">Password: </label>
                                <input type="password" value={password} onChange={(e)=>setpassword(e.target.value)} className="form-control"/>
                            </div>
                            <div className="mt-3">

                                <input type="submit" value="Submit" className="form-control btn btn-primary"/>
                            </div>

                        </form>
```

4. Output:



Sign In

Email

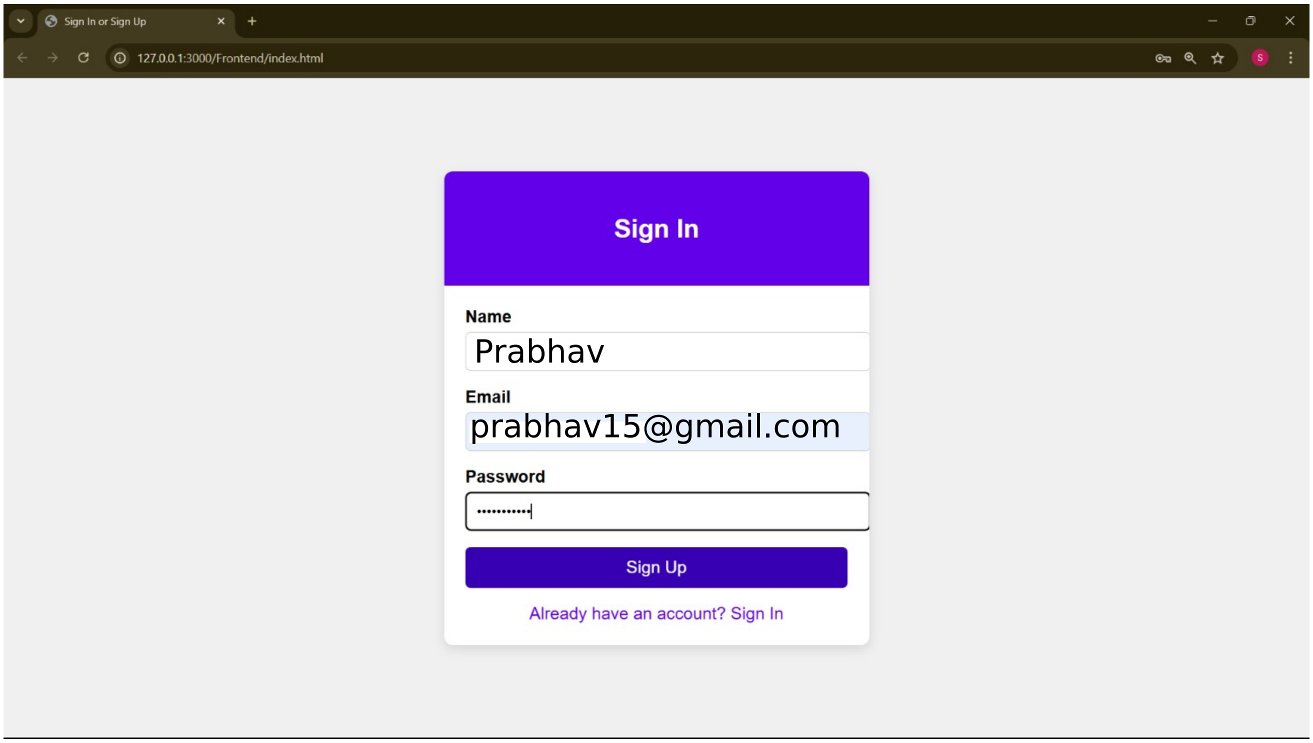
Prabhav

Password

••••••••••

Sign In

Don't have an account? Sign Up


Sign In or Sign Up
127.0.0.1:3000/Frontend/index.html
Sign Up
Already have an account? Sign In

Sign In

Name: Prabhav

Email: prabhav15@gmail.com

Password: •••••••••••

5. Learning Outcome:

- Design user-friendly forms for user login and registration using React..
- Learn how each technology works individually and how they integrate to form a full-stack application.
- Set up a server with Express to handle HTTP requests for user registration and login
- Verify that the backend API functions as expected by testing the registration and login endpoints with Postman