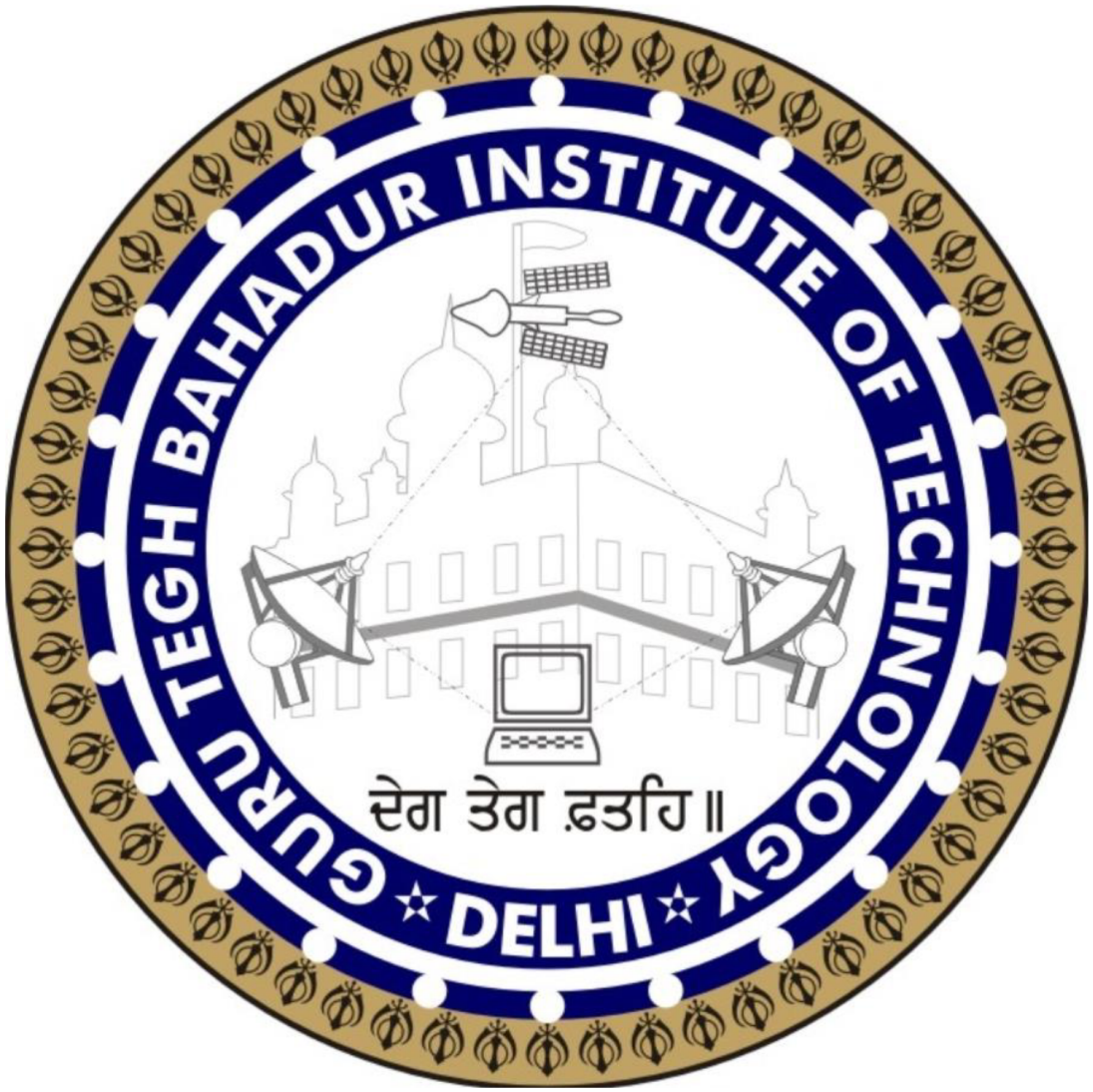
GURU TEGH BAHADUR INSTITUTE OF TECHNOLOGY



WEB DEVELOPMENT USING MERN STACK

SUBJECT CODE: FSD-322P

SEMESTER – VI

Submitted to,

Ms. Kritika

Submitted by,

Name: Abhay Raj
Enrolment No: 00976803122
Branch: IT – 3 (FSD)
Batch: 2022-2026

S.No	Program	Date	Signature
1. (a)	To Introduce the 3 tier architecture of mern stack.		
1. (b)	To elucidate the HTML elements		
2. (a)	To introduce CSS and its types		
2. (b)	Introduction to JavaScript and its types		
3. (a)	To create a time-table using HTML		
3. (b)	To create nested lists using HTML		
4. (a)	To create a registration form using HTML		
4. (b)	To create frames and hyperlinks using HTML		
5. (a)	Create a static website using HTML, CSS, and JavaScript		
5. (b)	Install and set up Node.JS and Express.JS		
6.	Set up React App and print “Hello World”		
7.	To make list components and table components		
8. (a)	To perform use state in React to alter the state of components		
8. (b)	Use props to send data between components		
9.	To create a server in Node.JS and Express.JS and send a get request.		
10. (a)	To install MongoDB Server and Mongosh on the local machine		
10. (b)	Create a MongoDB and perform CRUD operations on it		
11.	Task management tool: Login/Register to the application, add daily tasks, Assign a due date of completion, Mark them as complete/incomplete, and View weekly/monthly statistics of their to-dos.		

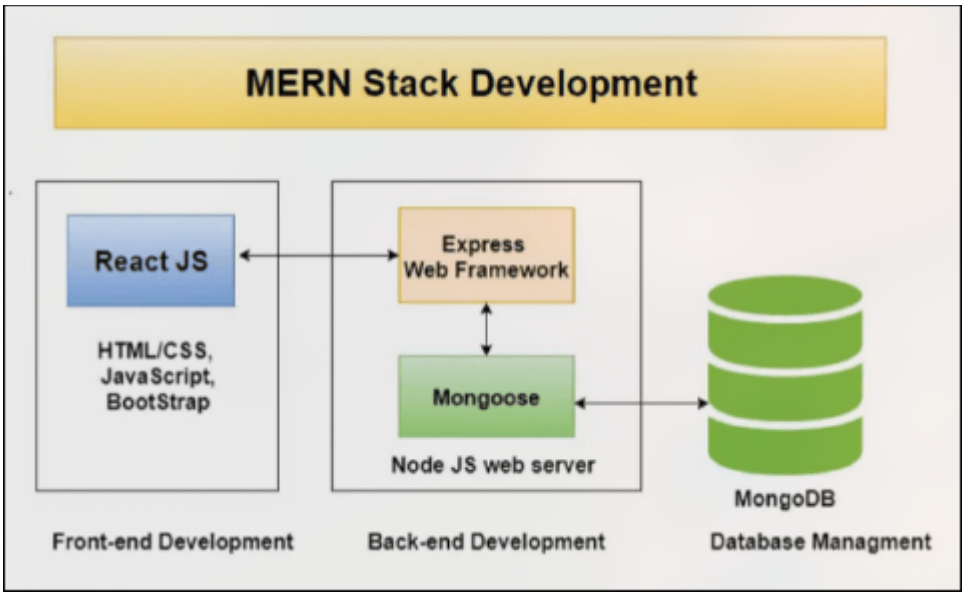
12.	Blogging platform		
13.	Social media platform		
14.	Weather Forecasting App		
15.	Bookstore Library and Stock-Keeping App		

Experiment 1 (a)

Aim:- To Introduce the 3 Tier architecture of Mern Stack.

Although there are many different stacks out there to consider, some have become more common than others due to their popularity and ease of use. One of these popular stacks is the MERN stack. Let us check out MERN stack full form which consists of the following technologies:

1. M stands for MongoDB (Database) which is a Popular NoSQL (Non-Structured ‘
2. Query Language) Database System.
3. E stands for Express, A powerful middleware that sits on top of the server side Node.js web APIs.
4. R stands for React, a client-side JavaScript Library renowned for developing rich user experiences.
5. N stands for Node js, A premier JavaScript web server development environment that is fast and efficient



MERN Stack Components

MERN adheres to a 3-tier architecture system consisting of 3 layers

These layers are as follows:

1. Front-end tier for Web: Basically, the top tier of the MERN stack handled by React.js, which is prominently used open-source front-end JavaScript library for

building Web applications

2. The middle tier is Server: The next level from top layer is handled by two components of the stack, i.e., Express.js and Node.js. They handle it simultaneously where Express.js maintains the server-side framework and middleware, running inside the Node.js server.

3. Backend tier as Database: Being one of the most important levels of the MERN Stack, it is handled by MongoDB. The main role of a database is to store all your application-related data. It is the most popular NoSQL database.

Experiment 1 (b)

Aim:- To elucidate the HTML elements

An **HTML element** is the fundamental building block of a webpage, consisting of a **start tag, content, and an end tag**. Elements define the structure and behavior of web pages.

1. Basic Structure of an HTML Element
<tagname>Content</tagname>

Example:
Hello, World!

2. Types of HTML Elements

a) Block-Level Elements

These elements take up the full width and start on a new line.

<h1>Heading</h1>
Paragraph
Container

These elements take up only as much width as necessary and do not start on a new line.

b) Inline Elements

These elements take up only as much width as necessary and do not start on a new line.

Text
Link
Bold

c) Void (Self-Closing) Elements

These elements do not have closing tags.

<img src="image.jpg" alt="Image">

<input type="text">

**
**

3. Commonly Used HTML Elements

a) Text Formatting Elements

b) List Elements

c) Table Elements

d) Form Elements

```
<!DOCTYPE html>
<html>
<head><title>HTML Example</title></head>
<body>
  <h1>Welcome</h1>
  <p><strong>HTML elements demo</strong></p>
  <ul><li>Web Dev</li><li>App Dev</li></ul>
  <img src="logo.png" alt="Logo" width="200">
  <table border="1">
    <tr><th>Service</th><th>Price</th></tr>
    <tr><td>Web Dev</td><td>$500</td></tr>
  </table>
  <form>
    <input type="text" placeholder="Name">
    <input type="email" placeholder="Email">
    <button>Submit</button>
  </form>
</body>
</html>
```

Output:

Experiment 2 (a)

Aim:- To introduce CSS and its types

CSS (**Cascading Style Sheets**) is used to **style** and **layout** web pages. It controls elements like colors, fonts, spacing, and positioning.

Types of CSS

1. **Inline CSS** (Applied directly to an element using the style attribute)

   This is inline CSS

   - **Pros:** Quick, applies to a single element.
   - **Cons:** Hard to maintain, not reusable.

2. **Internal CSS** (Defined within a <style> tag in the <head> section)

   <style>
   p { color: blue; font-size: 18px; }
   </style>

   - **Pros:** Styles multiple elements in a single file.
   - **Cons:** Not reusable across multiple pages.

3. **External CSS** (Stored in a separate .css file and linked to HTML)

   <link rel="stylesheet" href="styles.css">

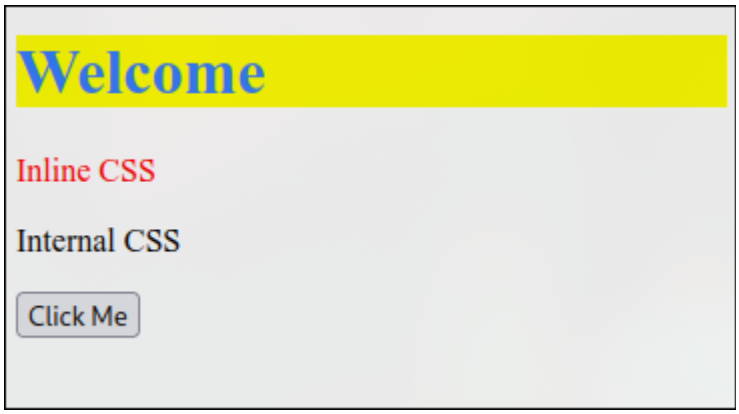
   p { color: green; font-size: 16px; }

   - **Pros:** Best for large projects, reusable across multiple pages.
   - **Cons:** Requires an extra file, but improves maintainability.

Eg: <!DOCTYPE html>
<html>
<head>
<title>CSS Types</title>
<link rel="stylesheet" href="styles.css">
<style> h1 { color: blue; } </style>
</head>
<body>
<h1 style="background: yellow;">Welcome</h1>
Inline CSS

```
    <p>Internal CSS</p>
    <button>Click Me</button>
</body>
</html>
```

Output:



Experiment 2 (b)

Aim:- Introduction to JavaScript and its types

JavaScript is a **dynamic programming language** used for creating **interactive** web pages. It enhances **functionality**, handles **events**, and manipulates the **DOM**.

Types of JavaScript

1. **Internal JavaScript** (Written inside <script> tags in HTML)

   ```
   <script>
      alert("Hello, this is Internal JavaScript!");
   </script>
   ```

   - **Pros:** Easy to use for small scripts.
   - **Cons:** Hard to maintain in large projects.

2. **External JavaScript** (Written in a separate .js file and linked to HTML)

   ```
   <script src="script.js"></script>
   ```

   ```
   console.log("This is External JavaScript.");
   ```

   - **Pros:** Reusable and maintainable.
   - **Cons:** Requires an extra file.

3. **Inline JavaScript** (Used directly inside HTML elements)

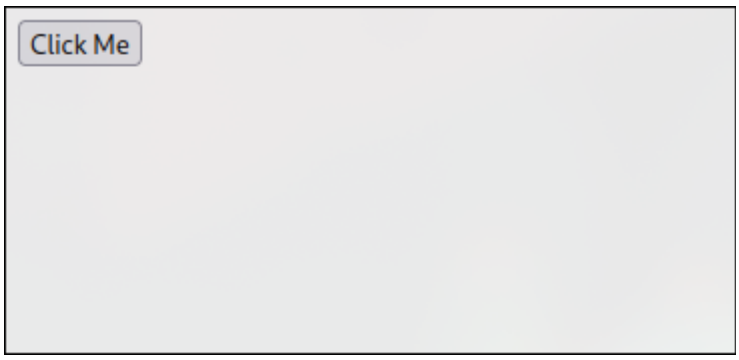
   ```
   <button onclick="alert('Inline JavaScript!')">Click Me</button>
   ```

   - **Pros:** Quick for simple actions.
   - **Cons:** Hard to debug and maintain.

Eg.

HTML (index.html)

```
<!DOCTYPE html>
<html>
<head><title>JS Example</title></head>
<body>
   <button onclick="alert('Hello, JavaScript!')">Click Me</button>
   <script>console.log("Script Loaded!");</script>
</body>
</html>
```


Click Me

Experiment 3 (a)

Aim:- To create a time-table using HTML

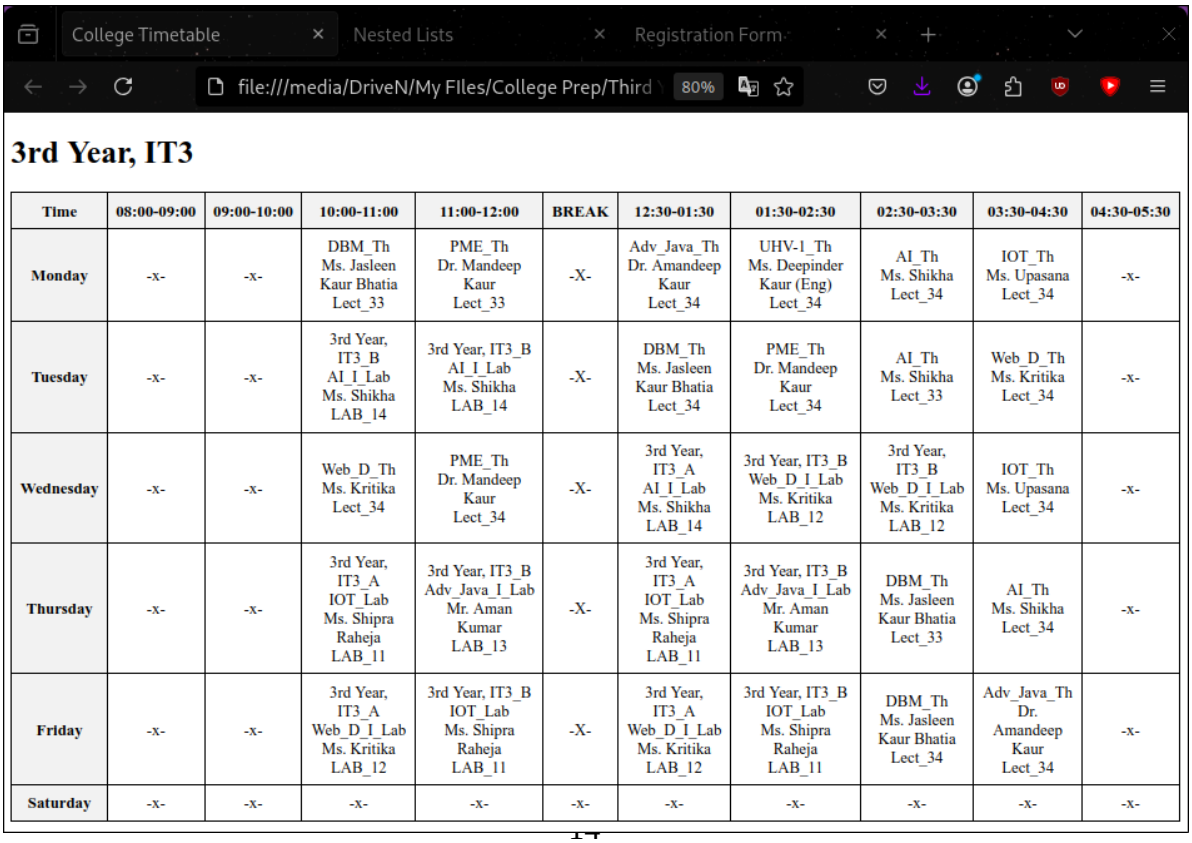
Code:-

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>College Timetable</title>
<style>
table {
width: 100%;
border-collapse: collapse;
}
th, td {
border: 1px solid black;
text-align: center;
padding: 8px;
}
th {
background-color: #f2f2f2;
}
</style>
</head>
<body>
<h1>3rd Year, IT3</h1>
<table>
<tr>
<th>Time</th>
<th>08:00-09:00</th>
<th>09:00-10:00</th>
<th>10:00-11:00</th>
<th>11:00-12:00</th>
<th>BREAK</th>
<th>12:30-01:30</th>
<th>01:30-02:30</th>
<th>02:30-03:30</th>
<th>03:30-04:30</th>
<th>04:30-05:30</th>
</tr>
<tr>
<th>Monday</th>
<td>-x-</td>
<td>-x-</td>
<td>DBM_Th<br>Ms. Jasleen Kaur Bhatia<br>Lect_33</td>
<td>PME_Th<br>Dr. Mandeep Kaur<br>Lect_33</td>
```

<td>-X-</td>
<td>Adv_Java_Th
Dr. Amandeep Kaur
Lect_34</td>
<td>UHV-1_Th
Ms. Deepinder Kaur (Eng)
Lect_34</td>
<td>AI_Th
Ms. Shikha
Lect_34</td>
<td>IOT_Th
Ms. Upasana
Lect_34</td>
<td>-x-</td>
</tr>
<tr>
<th>Tuesday</th>
<td>-x-</td>
<td>-x-</td>
<td>3rd Year, IT3_B
AI_I_Lab
Ms. Shikha
LAB_14</td>
<td>3rd Year, IT3_B
AI_I_Lab
Ms. Shikha
LAB_14</td>
<td>-X-</td>
<td>DBM_Th
Ms. Jasleen Kaur Bhatia
Lect_34</td>
<td>PME_Th
Dr. Mandeep Kaur
Lect_34</td>
<td>AI_Th
Ms. Shikha
Lect_33</td>
<td>Web_D_Th
Ms. Kritika
Lect_34</td>
<td>-x-</td>
</tr>
<tr>
<th>Wednesday</th>
<td>-x-</td>
<td>-x-</td>
<td>Web_D_Th
Ms. Kritika
Lect_34</td>
<td>PME_Th
Dr. Mandeep Kaur
Lect_34</td>
<td>-X-</td>
<td>3rd Year, IT3_A
AI_I_Lab
Ms. Shikha
LAB_14</td>
<td>3rd Year, IT3_B
Web_D_I_Lab
Ms. Kritika
LAB_12</td>
<td>3rd Year, IT3_B
Web_D_I_Lab
Ms. Kritika
LAB_12</td>
<td>IOT_Th
Ms. Upasana
Lect_34</td>
<td>-x-</td>
</tr>
<tr>
<th>Thursday</th>
<td>-x-</td>
<td>-x-</td>
<td>3rd Year, IT3_A
IOT_Lab
Ms. Shipra Raheja
LAB_11</td>
<td>3rd Year, IT3_B
Adv_Java_I_Lab
Mr. Aman Kumar
LAB_13</td>
<td>-X-</td>
<td>3rd Year, IT3_A
IOT_Lab
Ms. Shipra Raheja
LAB_11</td>
<td>3rd Year, IT3_B
Adv_Java_I_Lab
Mr. Aman Kumar
LAB_13</td>
<td>DBM_Th
Ms. Jasleen Kaur Bhatia
Lect_33</td>
<td>AI_Th
Ms. Shikha
Lect_34</td>
<td>-x-</td>
</tr>
<tr>
<th>Friday</th>
<td>-x-</td>

```
<td>-x-</td>
<td>3rd Year, IT3_A<br>Web_D_I_Lab<br>Ms. Kritika<br>LAB_12</td>
<td>3rd Year, IT3_B<br>IOT_Lab<br>Ms. Shipra Raheja<br>LAB_11</td>
<td>-X-</td>
<td>3rd Year, IT3_A<br>Web_D_I_Lab<br>Ms. Kritika<br>LAB_12</td>
<td>3rd Year, IT3_B<br>IOT_Lab<br>Ms. Shipra Raheja<br>LAB_11</td>
<td>DBM_Th<br>Ms. Jasleen Kaur Bhatia<br>Lect_34</td>
<td>Adv_Java_Th<br>Dr. Amandeep Kaur<br>Lect_34</td>
<td>-x-</td>
</tr>
<tr>
<th>Saturday</th>
<td>-x-</td>
<td>-x-</td>
<td>-x-</td>
<td>-x-</td>
<td>-x-</td>
<td>-x-</td>
<td>-x-</td>
<td>-x-</td>
<td>-x-</td>
<td>-x-</td>
</tr>
</table>
</body>
</html>
```

Output:-



3rd Year, IT3

Time	08:00-09:00	09:00-10:00	10:00-11:00	11:00-12:00	BREAK	12:30-01:30	01:30-02:30	02:30-03:30	03:30-04:30	04:30-05:30
Monday	-x-	-x-	DBM_Th Ms. Jasleen Kaur Bhatia Lect_33	PME_Th Dr. Mandeep Kaur Lect_33	-X-	Adv_Java_Th Dr. Amandeep Kaur Lect_34	UHV-1_Th Ms. Deepinder Kaur (Eng) Lect_34	AI_Th Ms. Shikha Lect_34	IOT_Th Ms. Upasana Lect_34	-x-
Tuesday	-x-	-x-	3rd Year, IT3_B AI_I_Lab Ms. Shikha LAB_14	3rd Year, IT3_B AI_I_Lab Ms. Shikha LAB_14	-X-	DBM_Th Ms. Jasleen Kaur Bhatia Lect_34	PME_Th Dr. Mandeep Kaur Lect_34	AI_Th Ms. Shikha Lect_33	Web_D_Th Ms. Kritika Lect_34	-x-
Wednesday	-x-	-x-	Web_D_Th Ms. Kritika Lect_34	PME_Th Dr. Mandeep Kaur Lect_34	-X-	3rd Year, IT3_A AI_I_Lab Ms. Shikha LAB_14	3rd Year, IT3_B Web_D_I_Lab Ms. Kritika LAB_12	3rd Year, IT3_B Web_D_I_Lab Ms. Kritika LAB_12	IOT_Th Ms. Upasana Lect_34	-x-
Thursday	-x-	-x-	3rd Year, IT3_A IOT_Lab Ms. Shipra Raheja LAB_11	3rd Year, IT3_B Adv_Java_I_Lab Mr. Aman Kumar LAB_13	-X-	3rd Year, IT3_A IOT_Lab Ms. Shipra Raheja LAB_11	3rd Year, IT3_B Adv_Java_I_Lab Mr. Aman Kumar LAB_13	DBM_Th Ms. Jasleen Kaur Bhatia Lect_33	AI_Th Ms. Shikha Lect_34	-x-
Friday	-x-	-x-	3rd Year, IT3_A Web_D_I_Lab Ms. Kritika LAB_12	3rd Year, IT3_B IOT_Lab Ms. Shipra Raheja LAB_11	-X-	3rd Year, IT3_A Web_D_I_Lab Ms. Kritika LAB_12	3rd Year, IT3_B IOT_Lab Ms. Shipra Raheja LAB_11	DBM_Th Ms. Jasleen Kaur Bhatia Lect_34	Adv_Java_Th Dr. Amandeep Kaur Lect_34	-x-
Saturday	-x-	-x-	-x-	-x-	-x-	-x-	-x-	-x-	-x-	-x-

Experiment 3 (b)

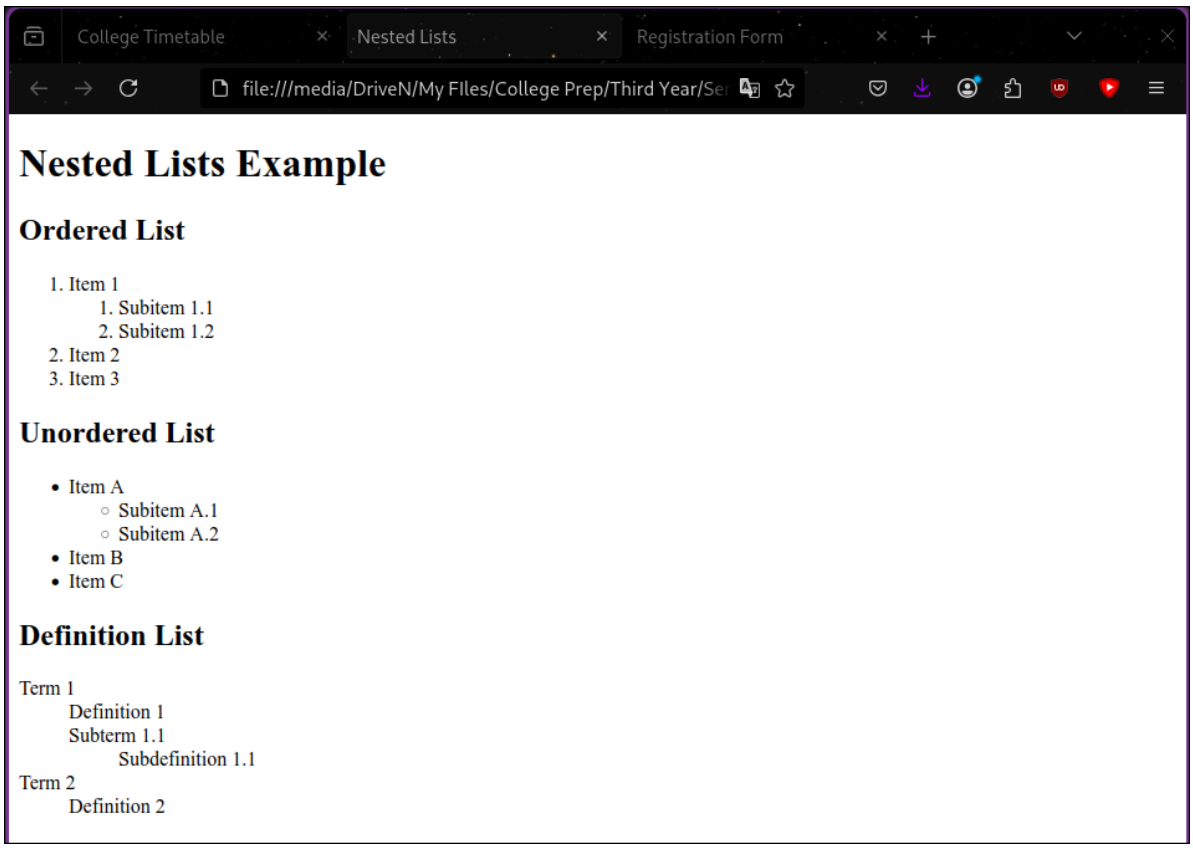
Aim:- To create nested lists using HTML

Code:-

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Nested Lists</title>
</head>
<body>
<h1>Nested Lists Example</h1>

<h2>Ordered List</h2>
<ol>
<li>Item 1
<ol>
<li>Subitem 1.1</li>
<li>Subitem 1.2</li>
</ol>
</li>
<li>Item 2</li>
<li>Item 3</li>
</ol>

<h2>Unordered List</h2>
<ul>
<li>Item A
<ul>
<li>Subitem A.1</li>
<li>Subitem A.2</li>
</ul>
</li>
<li>Item B</li>
<li>Item C</li>
</ul>

<h2>Definition List</h2>
<dl>
<dt>Term 1</dt>
<dd>Definition 1
<dl>
<dt>Subterm 1.1</dt>
<dd>Subdefinition 1.1</dd>
</dl>
</dd>
```

```
<dt>Term 2</dt>
<dd>Definition 2</dd>
</dl>
</body>
</html>
```

Output:-



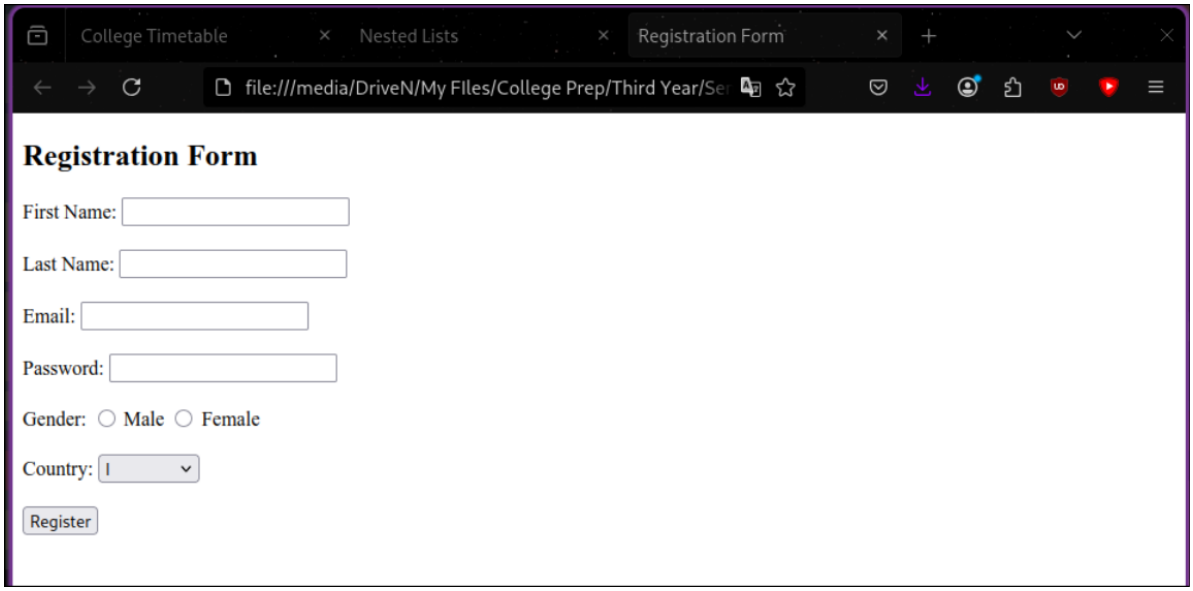
Experiment 4 (a)

Aim:- To create a registration form using HTML

Code:-

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Registration Form</title>
</head>
<body>
<h2>Registration Form</h2>
<form action="/submit_registration" method="post">
<label for="fname">First Name:</label>
<input type="text" id="fname" name="fname" required><br><br>
<label for="lname">Last Name:</label>
<input type="text" id="lname" name="lname" required><br><br>
<label for="email">Email:</label>
<input type="email" id="email" name="email" required><br><br>
<label for="password">Password:</label>
<input type="password" id="password" name="password" required><br><br>
<label for="gender">Gender:</label>
<input type="radio" id="male" name="gender" value="male">
<label for="male">Male</label>
<input type="radio" id="female" name="gender" value="female">
<label for="female">Female</label><br><br>
<label for="country">Country:</label>
<select id="country" name="country">
<option value="india">I</option>
<option value="canada">Canada</option>
<option value="uk">UK</option>
<option value="australia">Australia</option>
</select><br><br>
<input type="submit" value="Register">
</form>
</body>
</html>
```

Output:-


College Timetable
Nested Lists
Registration Form
file:///media/DriveN/My FIles/College Prep/Third Year/Se
Registration Form
First Name:
Last Name:
Email:
Password:
Gender: Male Female
Country: I
Register

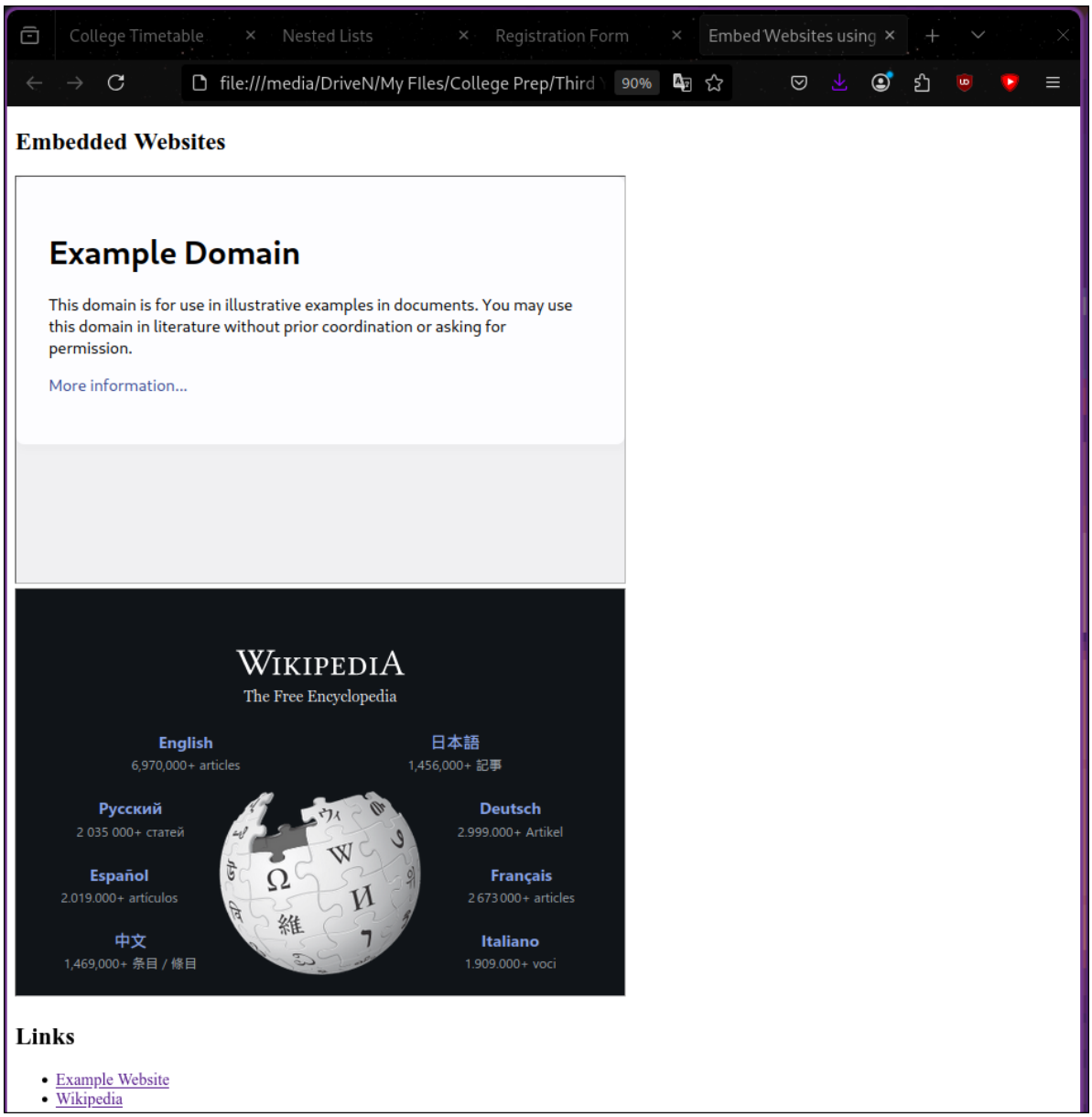
Experiment 4 (b)

Aim:- To create frames and hyperlinks using HTML

Code:-

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Embed Websites using Iframes</title>
</head>
<body>
<h2>Embedded Websites</h2>
<iframe src="https://www.example.com" width="600" height="400"
title="Example Website"></iframe>
<iframe src="https://www.wikipedia.org" width="600" height="400"
title="Wikipedia"></iframe>
<h2>Links</h2>
<ul>
<li><a href="https://www.example.com" target="_blank">Example
Website</a></li>
<li><a href="https://www.wikipedia.org"
target="_blank">Wikipedia</a></li>
</ul>
</body>
</html>
```

Output:-


College Timetable
Nested Lists
Registration Form
Embed Websites using
file:///media/DriveN/My FIles/College Prep/Third
90%
Embedded Websites
Example Domain
This domain is for use in illustrative examples in documents. You may use this domain in literature without prior coordination or asking for permission.
More information...
WIKIPEDIA
The Free Encyclopedia
English
6,970,000+ articles
日本語
1,456,000+ 記事
Русский
2 035 000+ статей
Deutsch
2.999.000+ Artikel
Español
2.019.000+ artículos
Français
2 673 000+ articles
中文
1,469,000+ 条目 / 條目
Italiano
1.909.000+ voci
Links
Example Website
Wikipedia

Experiment 5 (a)

Aim:- Create a static website using HTML, CSS, and JavaScript

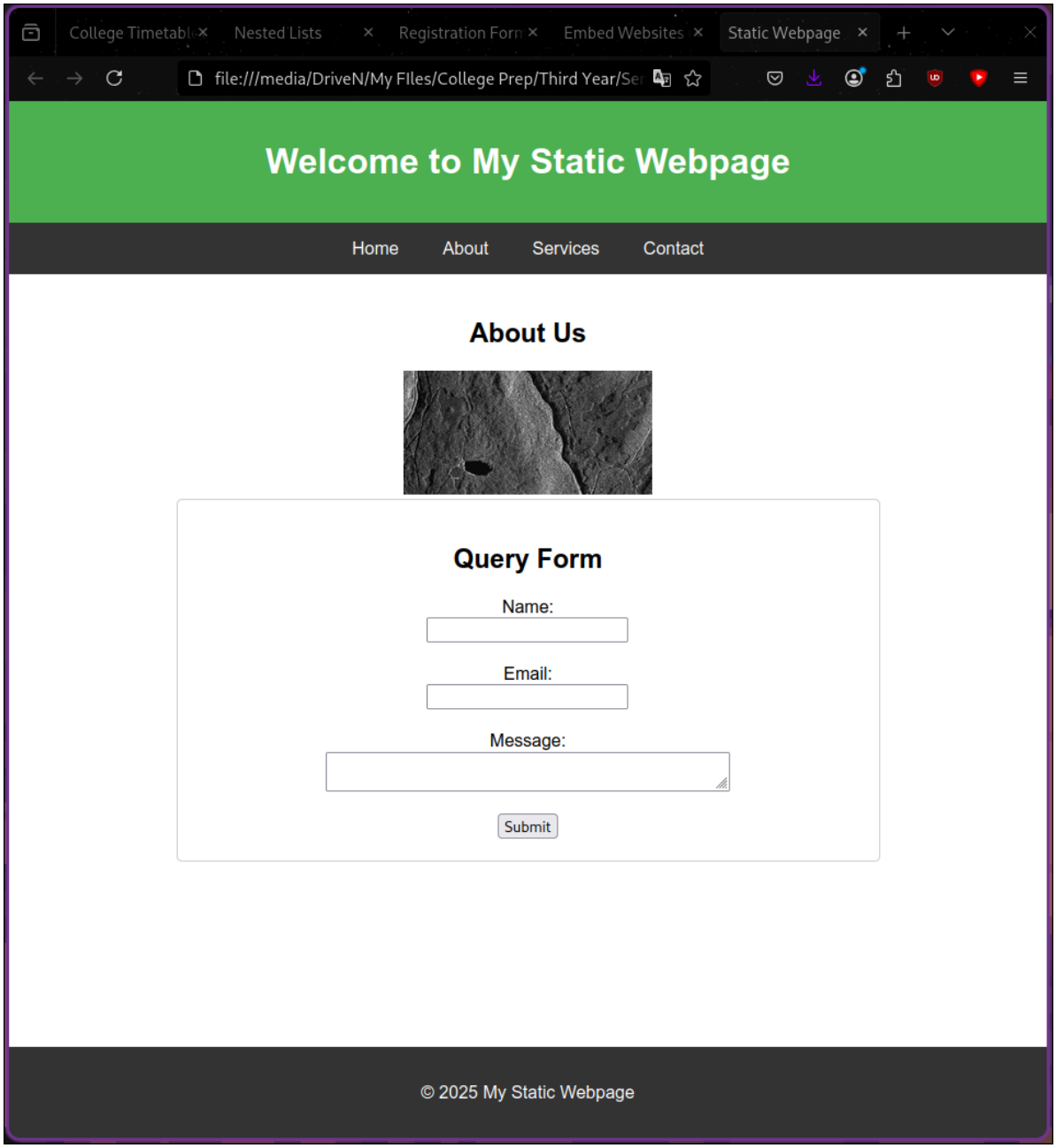
Code:-

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Static Webpage</title>
<style>
body {
font-family: Arial, sans-serif;
margin: 0;
padding: 0;
}
header {
background-color: #4CAF50;
color: white;
text-align: center;
padding: 1em 0;
}
nav {
display: flex;
justify-content: center;
background-color: #333;
}
nav a {
color: white;
padding: 14px 20px;
text-decoration: none;
text-align: center;
}
nav a:hover {
background-color: #ddd;
color: black;
}
.content {
padding: 20px;
text-align: center;
}
.content img {
max-width: 100%;
height: auto;
}
.form-container {
max-width: 600px;
```

```
margin: 0 auto;
padding: 20px;
border: 1px solid #ccc;
border-radius: 5px;
}
footer {
background-color: #333;
color: white;
text-align: center;
padding: 1em 0;
position: fixed;
width: 100%;
bottom: 0;
}
img {
width: 25%;
height: 25%;
}
</style>
</head>
<body>
<header>
<h1>Welcome to My Static Webpage</h1>
</header>
<nav>
<a href="#home">Home</a>
<a href="#about">About</a>
<a href="#services">Services</a>
<a href="#contact">Contact</a>
</nav>
<div class="content">
<h2>About Us</h2>
<img
src="https://www.researchgate.net/publication/288825767/figure/fig1/
AS:650042314596387@1531993573969/The-image-of-the-Montmorency-forest-
airborne-C-X-SAR-580-image-7-looks-800x400-pixel.png" alt="Sample
Image">
<div class="form-container">
<h2>Query Form</h2>
<form>
<label for="name">Name:</label><br>
<input type="text" id="name" name="name"><br><br>
<label for="email">Email:</label><br>
<input type="email" id="email" name="email"><br><br>
<label for="message">Message:</label><br>
<textarea id="message" name="message" rows="2"
cols="50"></textarea><br><br>
<input type="submit" value="Submit">
</form>
```

```
</div>
</div>
<footer>
<p>&copy; 2025 My Static Webpage</p>
</footer>
</body>
</html>
```

Output:-



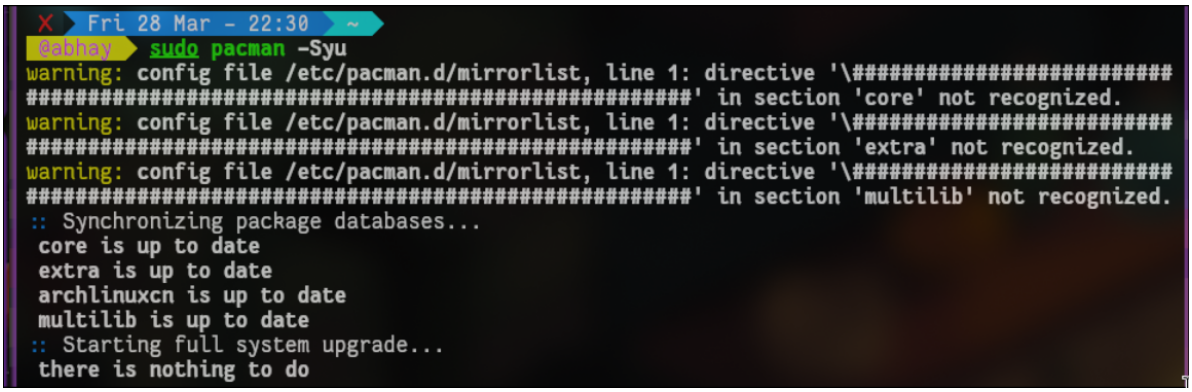
Experiment 5 (b)

Aim:- Install and set up Node.JS and Express.JS

Step 1: Install Node.js and npm

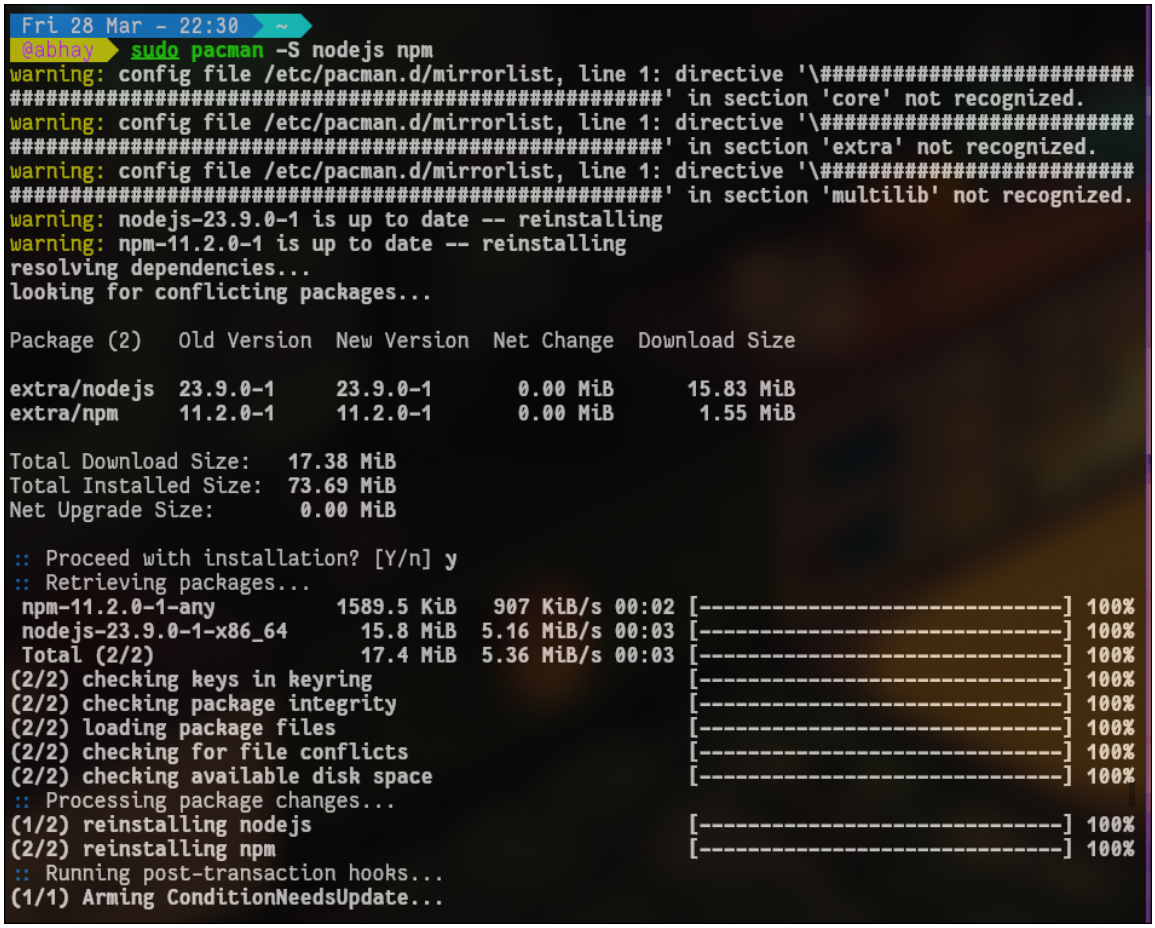
1. Open a terminal and update the package database:

   sudo pacman -Syu



2. Install Node.js and npm:

   sudo pacman -S nodejs npm



3. Verify the installation:

   node -v

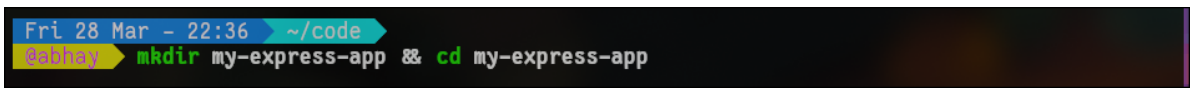
   npm -v

   If both commands return version numbers, Node.js and npm are installed successfully.

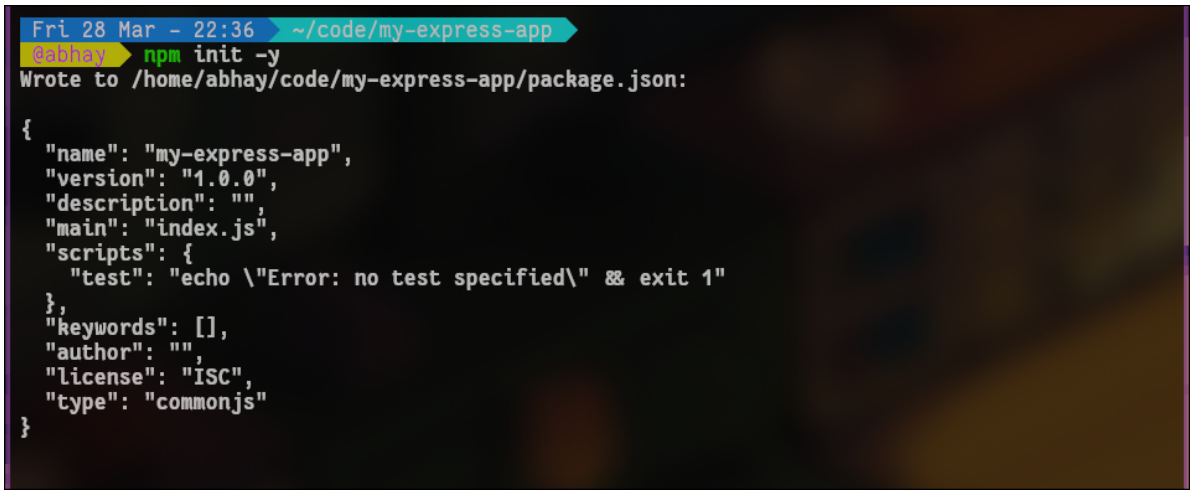
Step 2: Set Up an Express.js Project

1. Create a new project folder:

   mkdir my-express-app && cd my-express-app



2. Initialize a Node.js project:

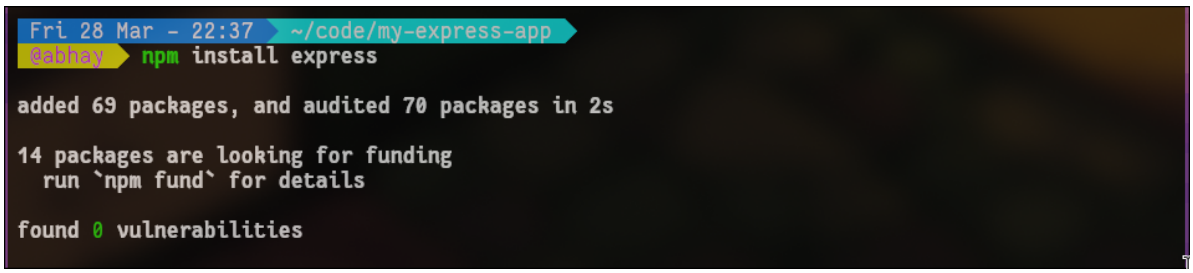


   npm init -y

   This generates a package.json file.

3. Install Express.js:

   npm install express



Step 3: Create an Express.js Server

1. Create an index.js file:

   touch index.js

2. Open it in a text editor (e.g., nano):

   nano index.js

   Add the following code:

   const express = require('express');

   const app = express();

   const port = 3002;

   app.get('/', (req, res) => {

   res.send('Hello, Express!');

   });

   app.listen(port, () => {

   console.log(`Server running at http://localhost:${port}`);

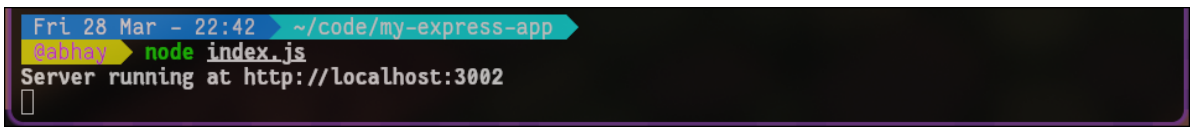
   });

   Save and exit (Ctrl + X, then Y, then Enter).

Step 4: Run the Server

Start the Express server:

node index.js



Now, open a browser and go to:

http://localhost:3000

You should see **"Hello, Express!"**.

Experiment 6

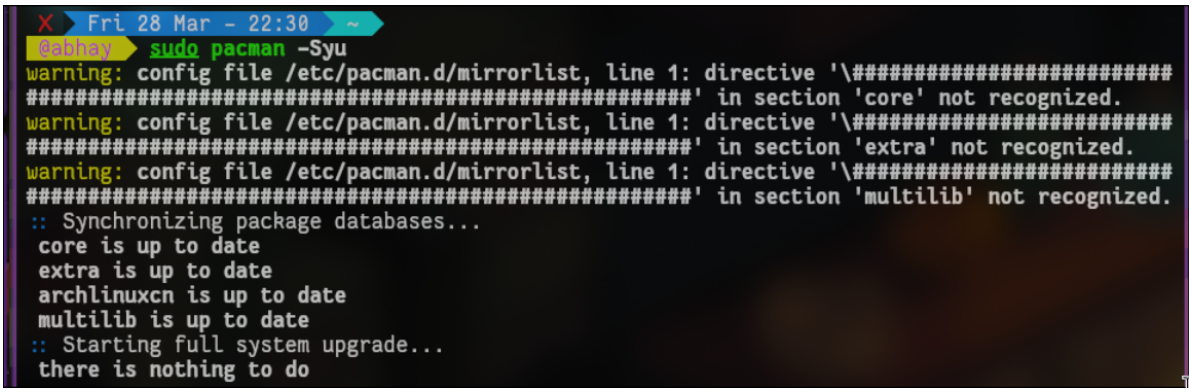
Aim:- Set up React App and print “Hello World”

Step 1: Install Node.js and npm

(If not installed yet, follow these steps)

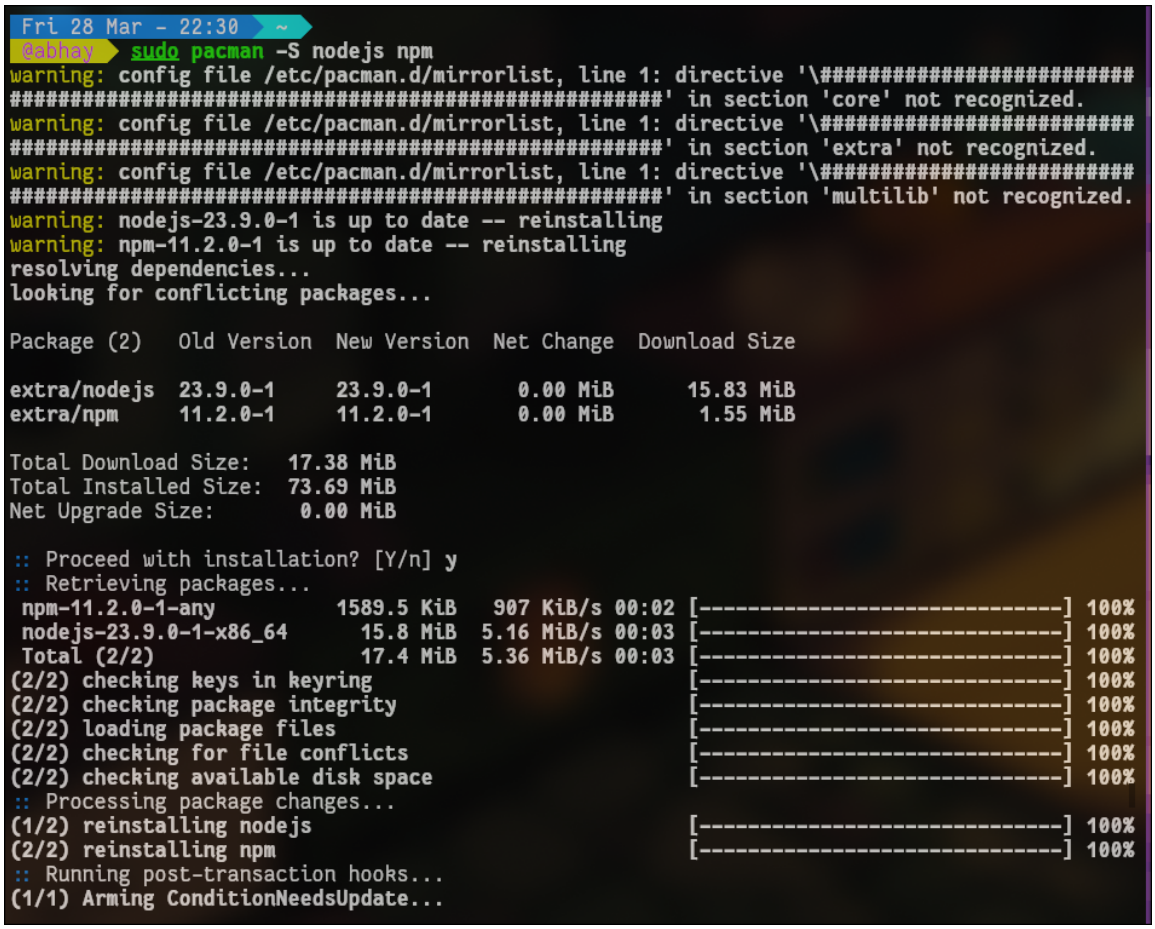
1. Open a terminal and update packages:

   sudo pacman -Syu



2. Install Node.js and npm:

   sudo pacman -S nodejs npm



3. Verify the installation:

   node -v

   npm -v

Step 2: Create a React App

1. Create a new React app:

   npx create-react-app my-react-app

```
8 vulnerabilities (2 moderate, 6 high)

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.

Created git commit.

Success! Created my-react-app at /home/abhay/code/my-react-app
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

  cd my-react-app
  npm start

Happy hacking!
```

2. Move into the project folder:

   cd my-react-app

```
Fri 28 Mar - 22:47  ~/code
@abhay  cd my-react-app

Fri 28 Mar - 22:49  ~/code/my-react-app  master ✓
@abhay
```

Step 3: Modify React App to Print "Hello World"

1. Open the file src/App.js in a text editor:

   nano src/App.js

2. Replace its content with:

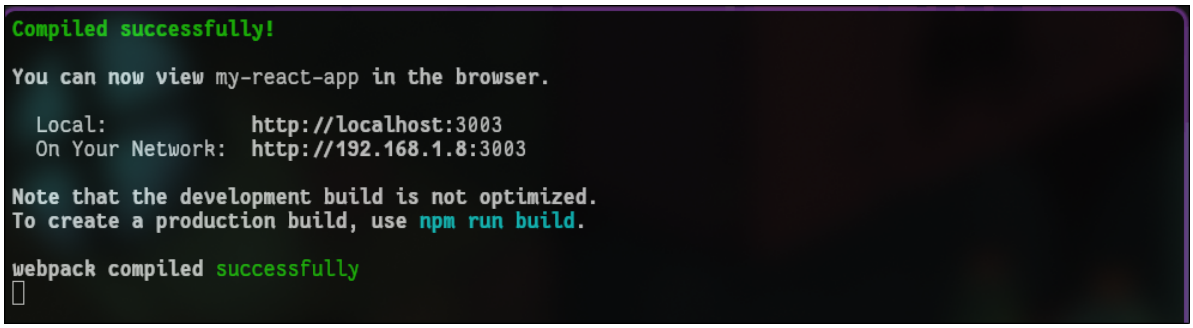
```
function App() {
  return (
    <div>
      <h1>Hello World</h1>
    </div>
  );
}
export default App;
```

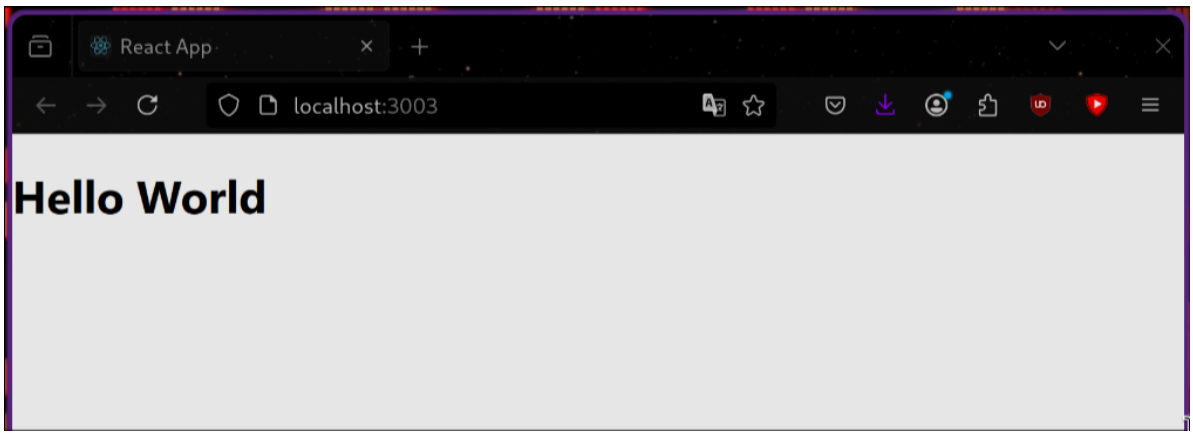
3. Save and exit (Ctrl + X, then Y, then Enter).

Step 4: Start the React App

Run the development server:

npm start





This will start the app on http://localhost:3000/, and you should see **"Hello World"** displayed on the page.

Experiment 7

Aim:- To make list components and table components

Code:-

/home/abhay/code/my-mui-app/src/App.js

```
import React from "react";
import ListComponent from "./ListComponent";
import TableComponent from "./TableComponent";
import { Container } from "@mui/material";

function App() {
return (
<Container sx={{ mt: 5 }}>
<ListComponent />
<TableComponent />
</Container>
);
}

export default App;
```

/home/abhay/code/my-mui-app/src/ListComponent.js

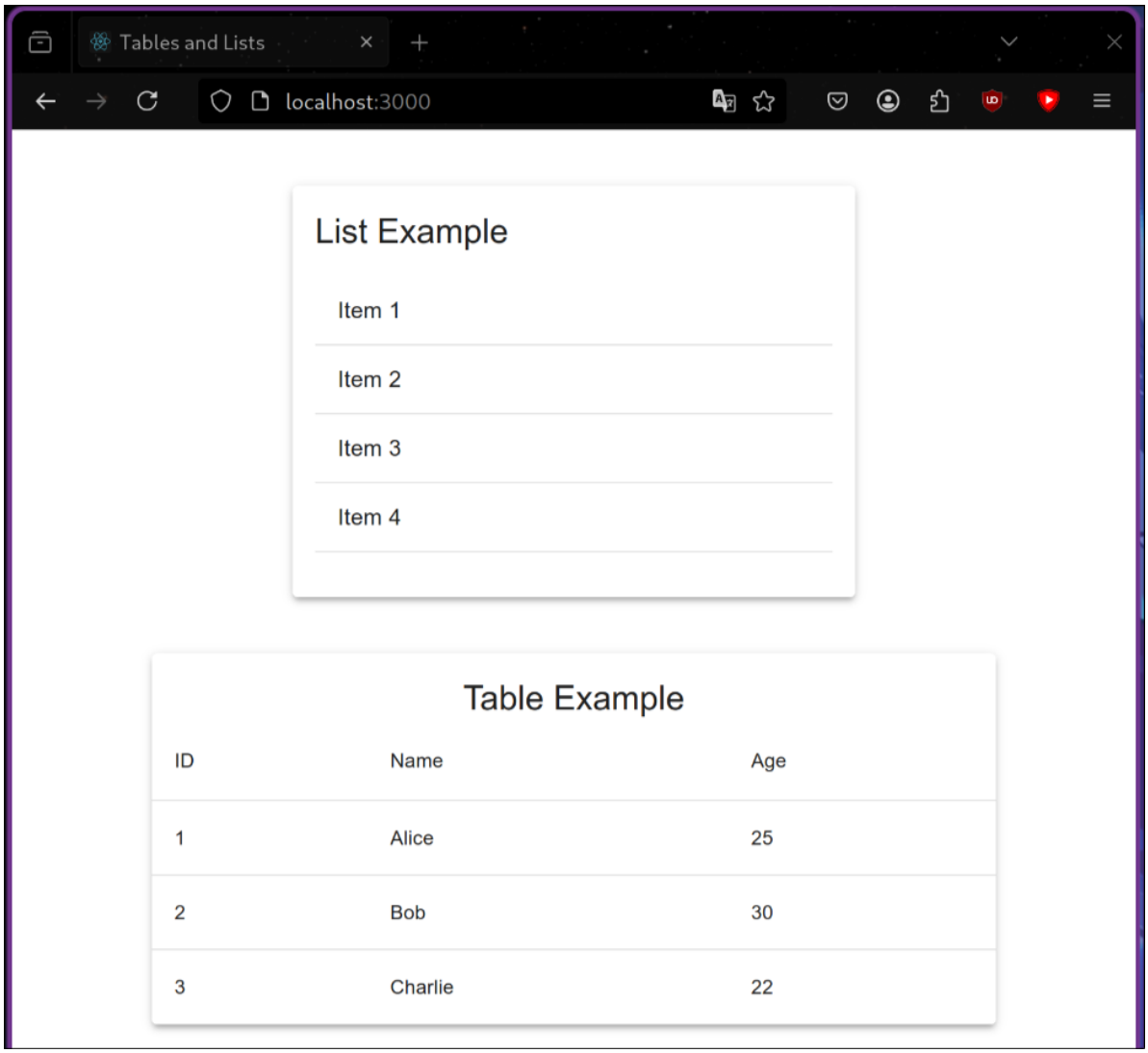
```
import React from "react";
import { Table, TableBody, TableCell, TableContainer, TableHead, TableRow, Paper,
Typography } from "@mui/material";

const data = [
{ id: 1, name: "Alice", age: 25 },
{ id: 2, name: "Bob", age: 30 },
{ id: 3, name: "Charlie", age: 22 },
];

const TableComponent = () => {
return (
<TableContainer component={Paper} sx={{ maxWidth: 600, margin: "auto", mt: 5,
boxShadow: 3 }}>
<Typography variant="h5" align="center" sx={{ mt: 2 }}>
Table Example
</Typography>
<Table>
<TableHead>
<TableRow>
<TableCell>ID</TableCell>
<TableCell>Name</TableCell>
```

```
<TableCell>Age</TableCell>
</TableRow>
</TableHead>
<TableBody>
{data.map((row) => (
<TableRow key={row.id}>
<TableCell>{row.id}</TableCell>
<TableCell>{row.name}</TableCell>
<TableCell>{row.age}</TableCell>
</TableRow>
))}
</TableBody>
</Table>
</TableContainer>
);
};

export default TableComponent;
```

/home/abhay/code/my-mui-app/src/ListComponent.js

```
import React from "react";
import { List, ListItem, ListItemText, Card, CardContent, Typography } from
"@mui/material";

const items = ["Item 1", "Item 2", "Item 3", "Item 4"];

const ListComponent = () => {
return (
<Card sx={{ maxWidth: 400, margin: "auto", mt: 5, boxShadow: 3 }}>
<CardContent>
<Typography variant="h5" gutterBottom>
List Example
</Typography>
<List>
{items.map((item, index) => (
<ListItem key={index} divider>
<ListItemText primary={item} />
</ListItem>
))}
</List>
</CardContent>
</Card>
);
};

export default ListComponent;
```

Output:-


Tables and Lists
localhost:3000
List Example
Item 1
Item 2
Item 3
Item 4
Table Example
ID
Name
Age
1
Alice
25
2
Bob
30
3
Charlie
22

Experiment 8 (a)

Aim:- To perform use state in React to alter the state of components.

Code:-

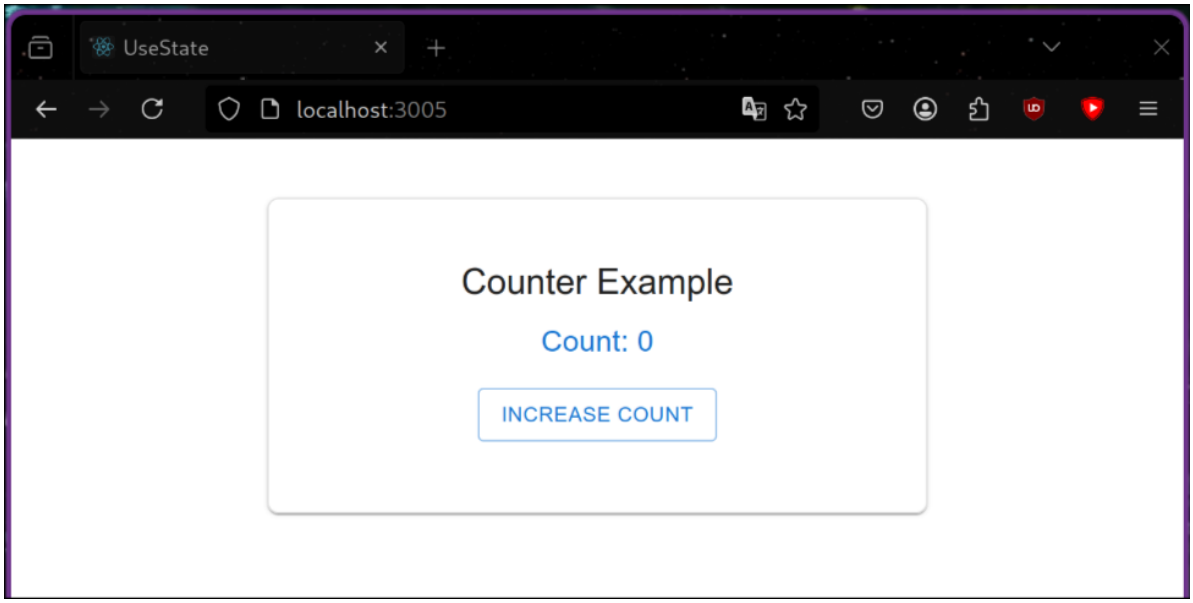
/home/abhay/code/use-state-app/use-state-app/src/StateComponent.js

```
import React, { useState } from "react";
import { Button, Typography, Container, Card, CardContent } from "@mui/material";

const StateComponent = () => {
const [count, setCount] = useState(0);

return (
<Container sx={{ mt: 5 }}>
<Card sx={{ maxWidth: 400, margin: "auto", textAlign: "center", padding: 3,
border: "1px solid #E0E0E0", borderRadius: 2, backgroundColor: "white" // Light grey
background
}}>
<CardContent>
<Typography variant="h5" gutterBottom>
Counter Example
</Typography>
<Typography variant="h6" color="primary">
Count: {count}
</Typography>
<Button variant="outlined" color="primary" sx={{ mt: 2 }} onClick={() => setCount(count
+ 1)}>
Increase Count
</Button>
</CardContent>
</Card>
</Container>
);
};

export default StateComponent;
```

/home/abhay/code/use-state-app/use-state-app/src/App.js

```
import React from "react";
import StateComponent from "./StateComponent";
import ListStateComponent from "./ListStateComponent";
import ToggleComponent from "./ToggleComponent";
import { Container } from "@mui/material";

function App() {
return (
<Container sx={{ mt: 5 }}>
<StateComponent />
</Container>
);
}

export default App;
```

Output:-



Experiment 8 (b)

Aim:- Use props to send data between components.

Code:-

/home/abhay/code/use-state-app/props-app/src/App.js

```
import React from "react";
import ParentComponent from "./ParentComponent";
import { Container } from "@mui/material";

function App() {
return (
<Container sx={{ mt: 5 }}>
<ParentComponent />
</Container>
);
}

export default App;
```

/home/abhay/code/use-state-app/props-app/src/ChildComponent.js

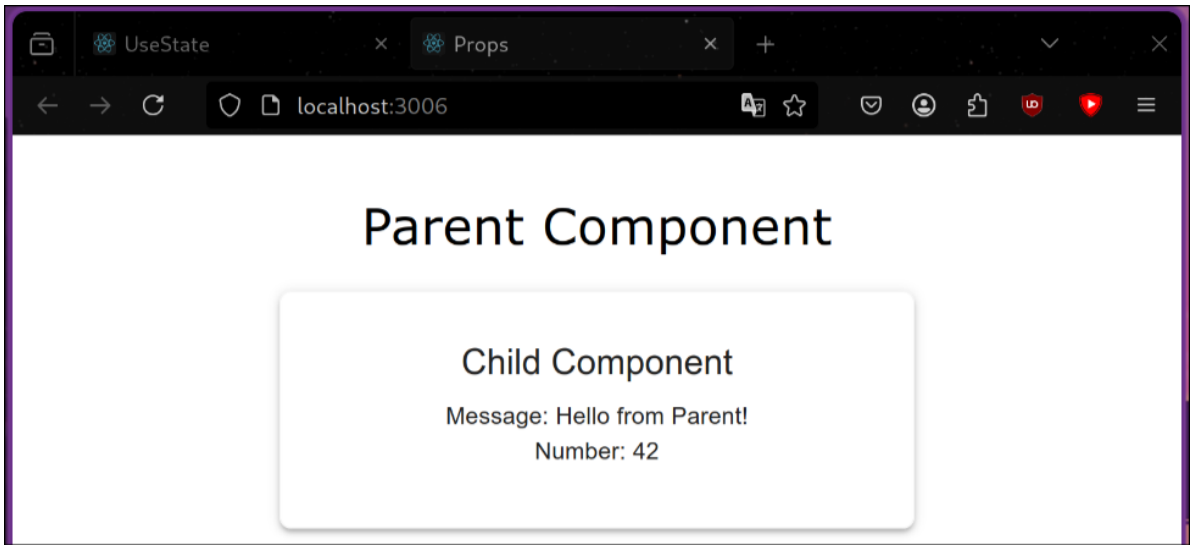
```
import React from "react";
import ChildComponent from "./ChildComponent";
import { Container, Typography } from "@mui/material";

const ParentComponent = () => {
return (
<Container sx={{ mt: 5, textAlign: "center" }}>
<Typography variant="h4" gutterBottom fontFamily={"sans-serif"}>
Parent Component
</Typography>
<ChildComponent message="Hello from Parent!" number={42} />
</Container>
);
};

export default ParentComponent;
```

/home/abhay/code/use-state-app/props-app/src/ParentComponent.js

```
import React from "react";
import { Card, CardContent, Typography } from "@mui/material";

const ChildComponent = ({ message, number }) => {
return (
```

```
<Card sx={{ maxWidth: 400, margin: "auto", mt: 3, boxShadow: 3,
borderRadius: 2,border:'1px solid', borderColor: "#f5f5f5", padding: 2, textAlign: "center"
}}>
<CardContent>
<Typography variant="h5" gutterBottom>
Child Component
</Typography>
<Typography variant="body1">Message: {message}</Typography>
<Typography variant="body1">Number: {number}</Typography>
</CardContent>
</Card>
);
};

export default ChildComponent;
```

Output:-



Experiment 9

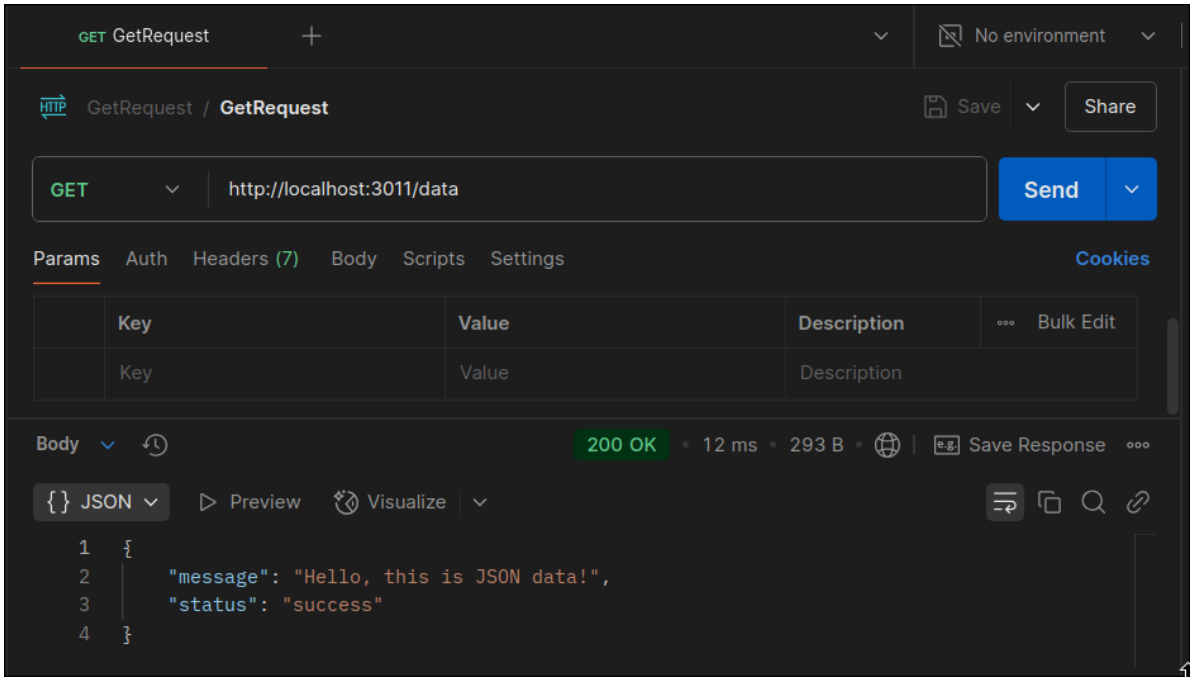
Aim:- To create a server in Node.JS and Express.JS and send a get request.

Code:-

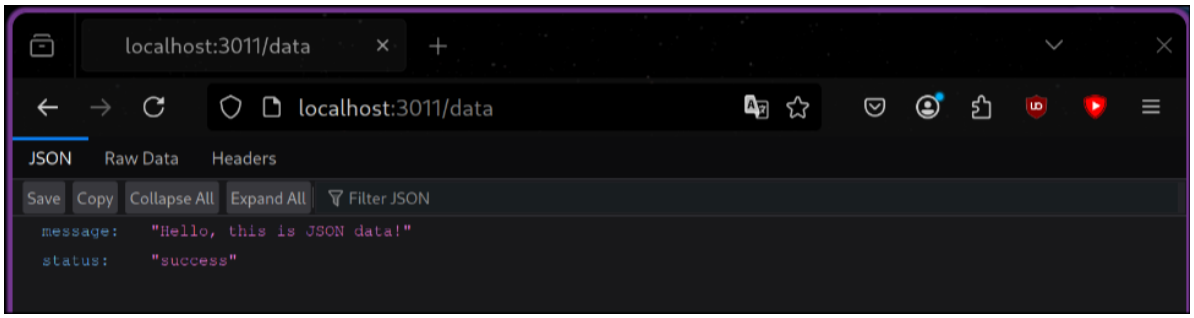
```
const express = require("express");
const app = express();
const PORT = 3011;

app.get("/data", (req, res) => {
res.json({ message: "Hello, this is JSON data!", status: "success" });
});

app.listen(PORT, () => {
console.log(`Server is running on http://localhost:${PORT}`);
});
```

Output:-

Postman:



Browser:



Experiment 10 (a)

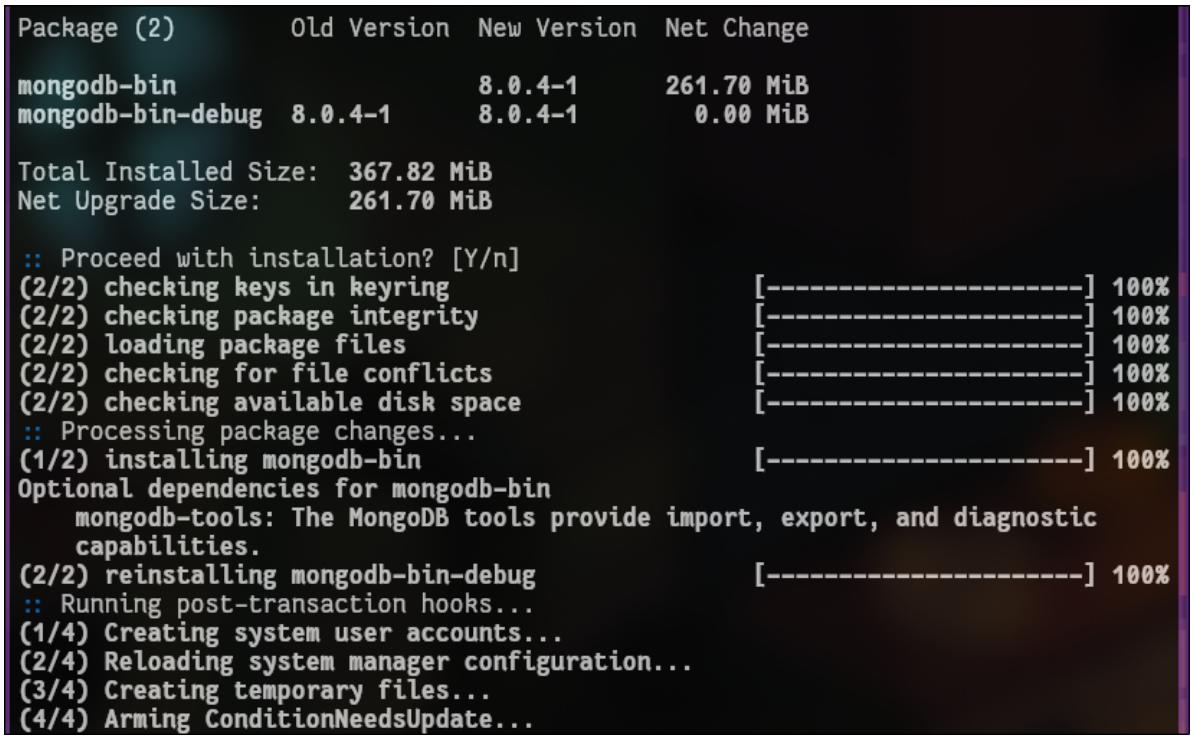
Aim:- To install MongoDB Server and Mongosh on the local machine.

Step 1: Install MongoDB

MongoDB is available in the **Arch User Repository (AUR)**. You need an **AUR helper** like yay or paru to install it.

1.1 Install MongoDB Using yay

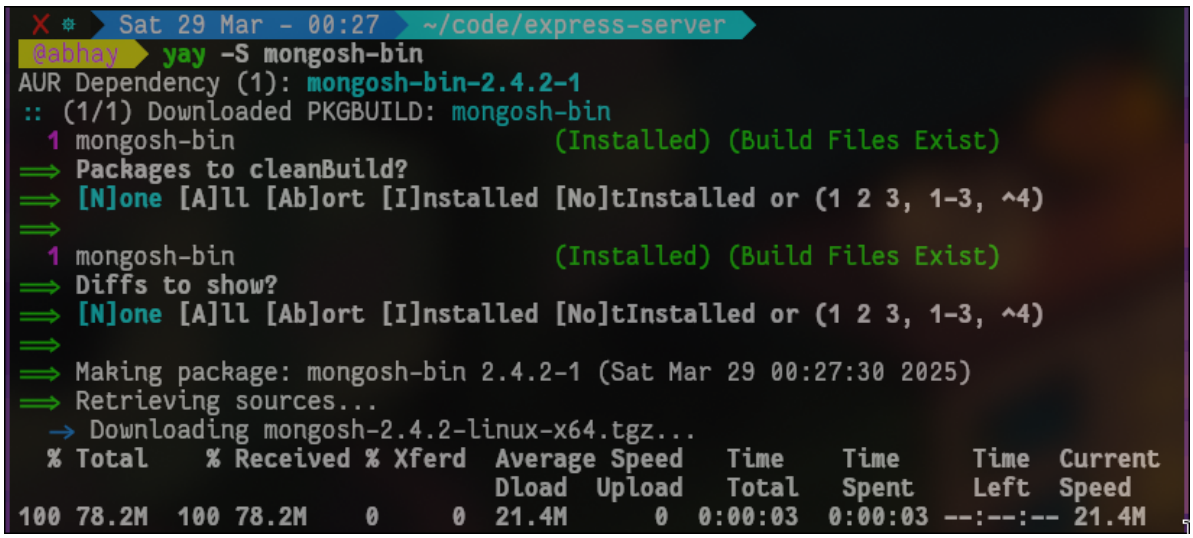
yay -S mongodb-bin

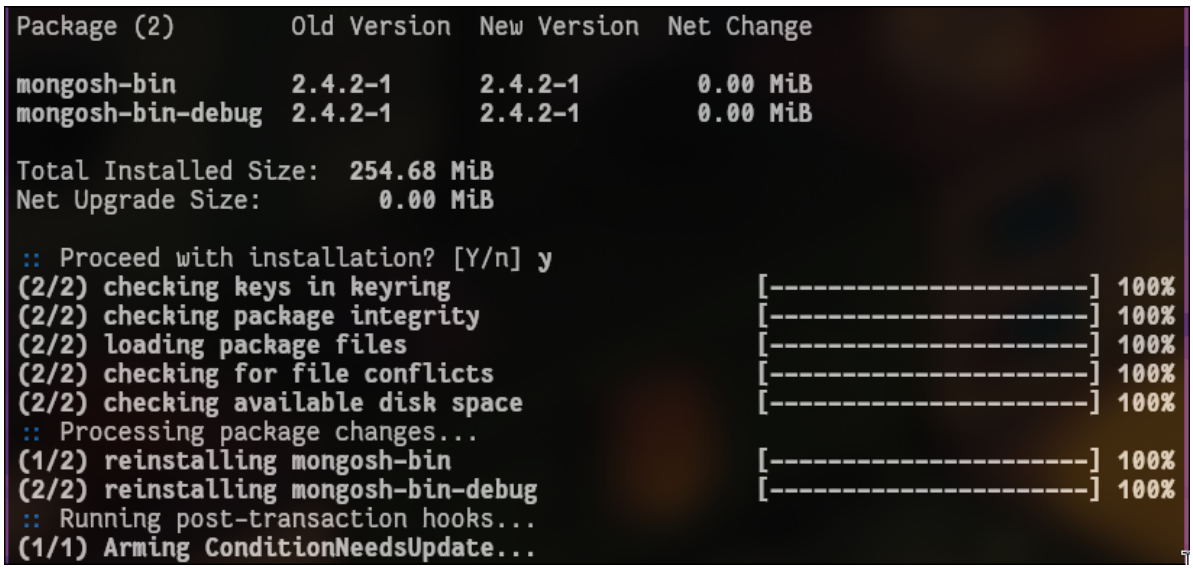


Step 2: Install MongoDB Shell (mongosh)

To install MongoDB Shell:

yay -S mongosh-bin





Step 3: Start and Enable MongoDB Service

After installation, start the MongoDB service:

sudo systemctl start mongodb

To enable it on startup:

sudo systemctl enable mongodb

Check if it's running:

sudo systemctl status mongodb

If MongoDB is running, you should see:

Active: active (running)

```
 Sat 29 Mar - 00:18  ~/code/express-server
@abhay  sudo systemctl start mongodb

 Sat 29 Mar - 00:18  ~/code/express-server
@abhay  sudo systemctl enable mongodb

 Sat 29 Mar - 00:18  ~/code/express-server
@abhay  sudo systemctl status mongodb

● mongodb.service - MongoDB Database Server
     Loaded: loaded (/usr/lib/systemd/system/mongodb.service; enabled; preset: d
isabled)
     Active: active (running) since Sat 2025-03-29 00:18:40 IST; 9s ago
 Invocation: f33a2ec0d6b44b1bb31f3acdcbd1594f
       Docs: https://docs.mongodb.org/manual
   Main PID: 114040 (mongod)
     Memory: 227.7M (peak: 285.6M)
        CPU: 626ms
     CGroup: /system.slice/mongodb.service
             └─114040 /usr/bin/mongod --config /etc/mongodb.conf

Mar 29 00:18:40 192.168.1.8 systemd[1]: Started MongoDB Database Server.
Mar 29 00:18:40 192.168.1.8 mongod[114040]: {"t":{"$date":"2025-03-28T18:48:40.3
28Z"},"s":"I",  "c":"CONTROL",  "id":7484500, "ctx":"main","msg":"Environment va
riable MONGODB_CONFIG_OVERRIDE_NOFORK == 1, overriding \"processManagement.fork\
" to false"}
```

Step 4: Connect to MongoDB

Now, open **MongoDB Shell (mongosh)** and connect:

mongosh

```
 Sat 29 Mar - 00:18  ~/code/express-server
@abhay  mongosh

Current Mongosh Log ID: 67e6ef1b9a38e803606b140a
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverS
electionTimeoutMS=2000&appName=mongosh+2.4.2
Using MongoDB:          8.0.4
Using Mongosh:          2.4.2

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

------
   The server generated these startup warnings when booting
   2025-03-29T00:18:40.386+05:30: Using the XFS filesystem is strongly recommend
ed with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodno
tes-filesystem
   2025-03-29T00:18:40.806+05:30: Access control is not enabled for the database
. Read and write access to data and configuration is unrestricted
   2025-03-29T00:18:40.806+05:30: For customers running the current memory alloc
ator, we suggest changing the contents of the following sysfsFile
   2025-03-29T00:18:40.806+05:30: We suggest setting the contents of sysfsFile t
o 0.
   2025-03-29T00:18:40.806+05:30: We suggest setting swappiness to 0 or 1, as sw
apping can cause performance problems.
------

test> use testDB
... db.testCollection.insertOne({ name: "ArchUser", age: 25 })
... db.testCollection.find()
...
switched to db testDB
testDB>
```

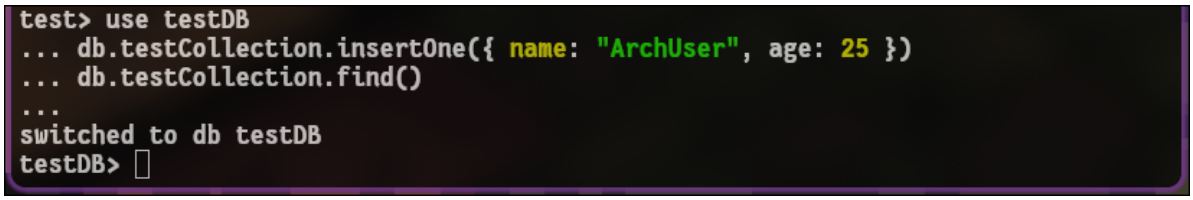
If connected successfully, you will see:

Current Mongosh Log ID: ...
Connecting to: mongodb://127.0.0.1:27017/

Step 5: Create a Test Database

Inside mongosh, run:

use testDB
db.testCollection.insertOne({ name: "ArchUser", age: 25 })
db.testCollection.find()



This inserts a test document into the testDB database and retrieves it.

Step 6: Stop MongoDB (Optional)

If you need to stop MongoDB:

sudo systemctl stop mongodb

To disable it from starting on boot:

sudo systemctl disable mongodb

Experiment 10 (b)

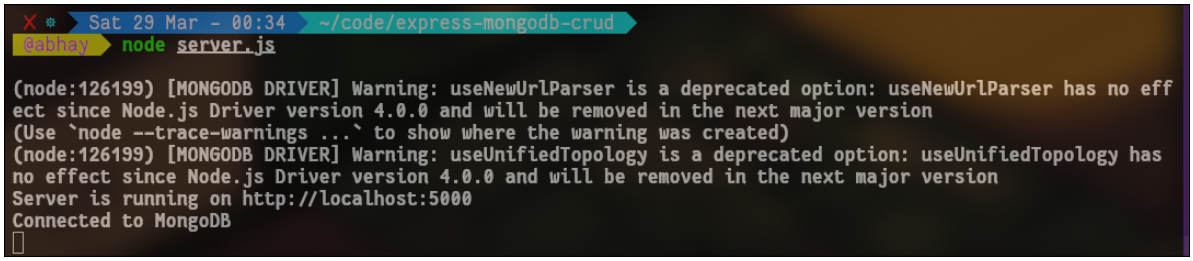
Aim:- Create a MongoDB and perform CRUD operations on it.

Code:-

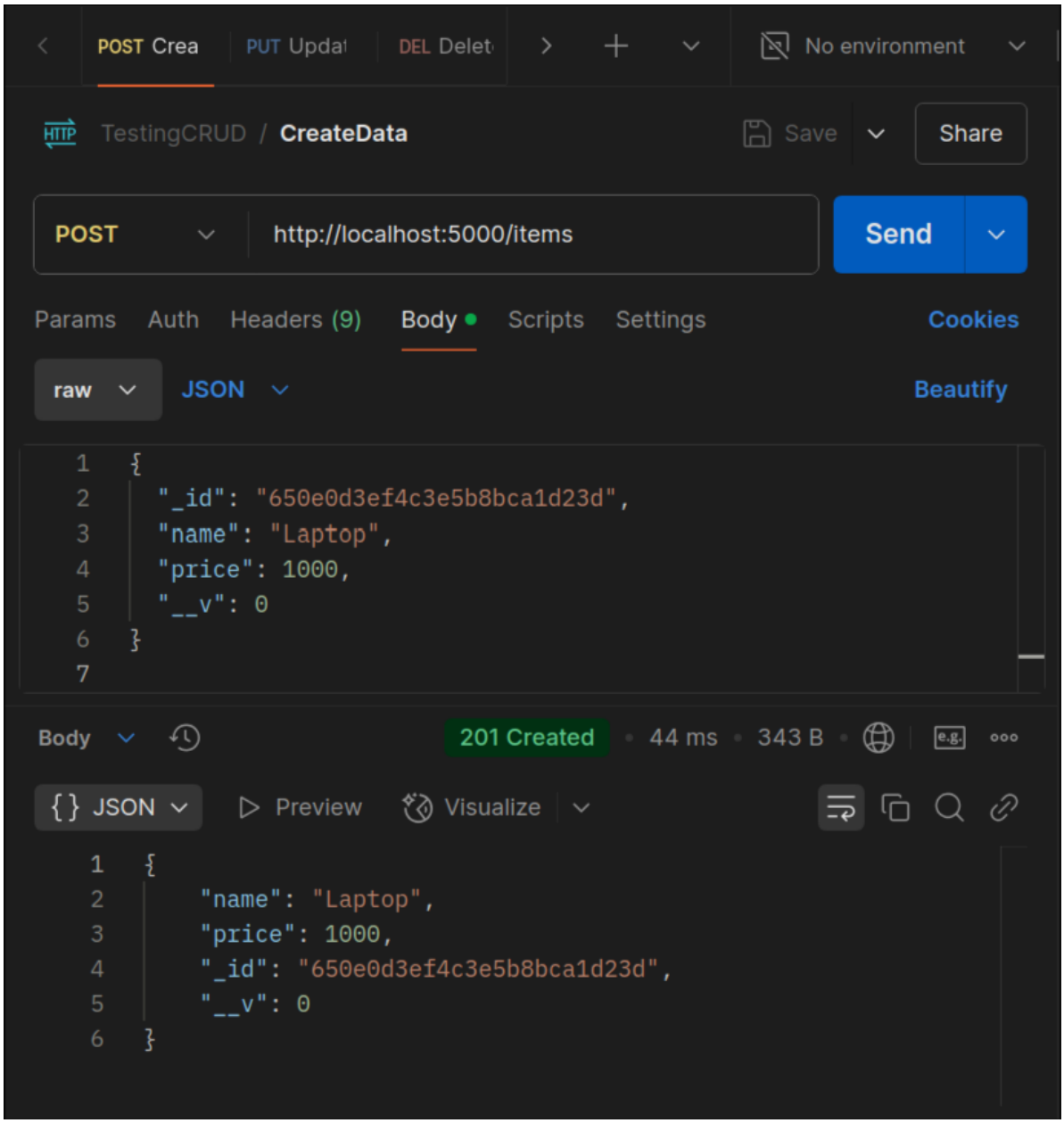
```
const express = require("express");
const mongoose = require("mongoose");
const bodyParser = require("body-parser");
const cors = require("cors");

const app = express();
const PORT = 5000;

app.use(bodyParser.json());
app.use(cors());

mongoose.connect("mongodb://127.0.0.1:27017/mydatabase", {
useNewUrlParser: true,
useUnifiedTopology: true,
});

const db = mongoose.connection;
db.on("error", console.error.bind(console, "MongoDB connection
error:"));
db.once("open", () => console.log("Connected to MongoDB"));

const ItemSchema = new mongoose.Schema({
name: String,
price: Number,
});

const Item = mongoose.model("Item", ItemSchema);

app.post("/items", async (req, res) => {
try {
const newItem = new Item(req.body);
await newItem.save();
res.status(201).json(newItem);
} catch (error) {
res.status(400).json({ error: error.message });
}
});

app.get("/items", async (req, res) => {
try {
const items = await Item.find();
res.json(items);
} catch (error) {
```

```
res.status(500).json({ error: error.message });
}
});

app.put("/items/:id", async (req, res) => {
try {
const updatedItem = await Item.findByIdAndUpdate(req.params.id,
req.body, { new: true });
res.json(updatedItem);
} catch (error) {
res.status(400).json({ error: error.message });
}
});

app.delete("/items/:id", async (req, res) => {
try {
await Item.findByIdAndDelete(req.params.id);
res.json({ message: "Item deleted successfully" });
} catch (error) {
res.status(500).json({ error: error.message });
}
});

app.listen(PORT, () => {
console.log(`Server is running on http://localhost:${PORT}`);
});
```

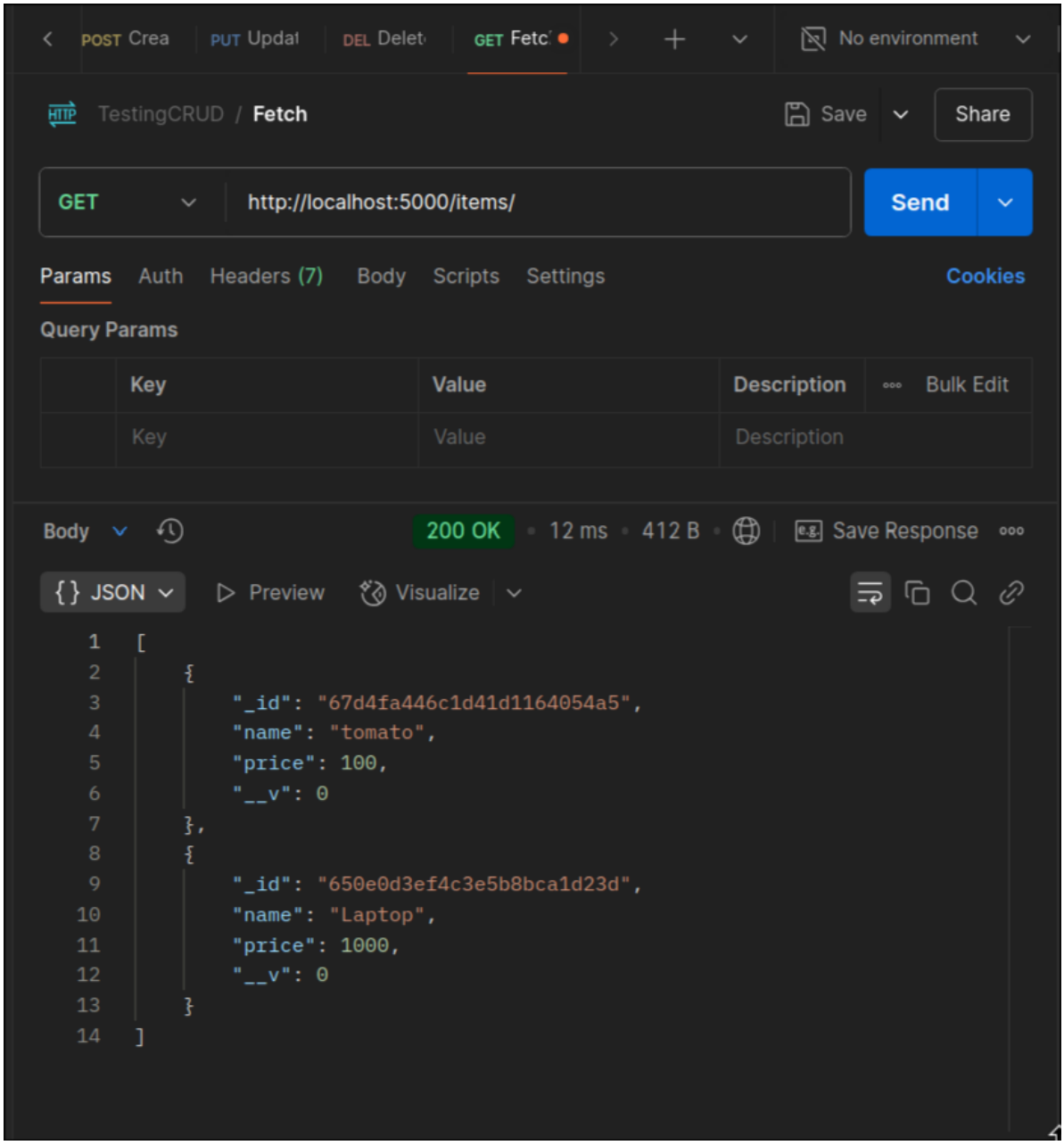
Output:-



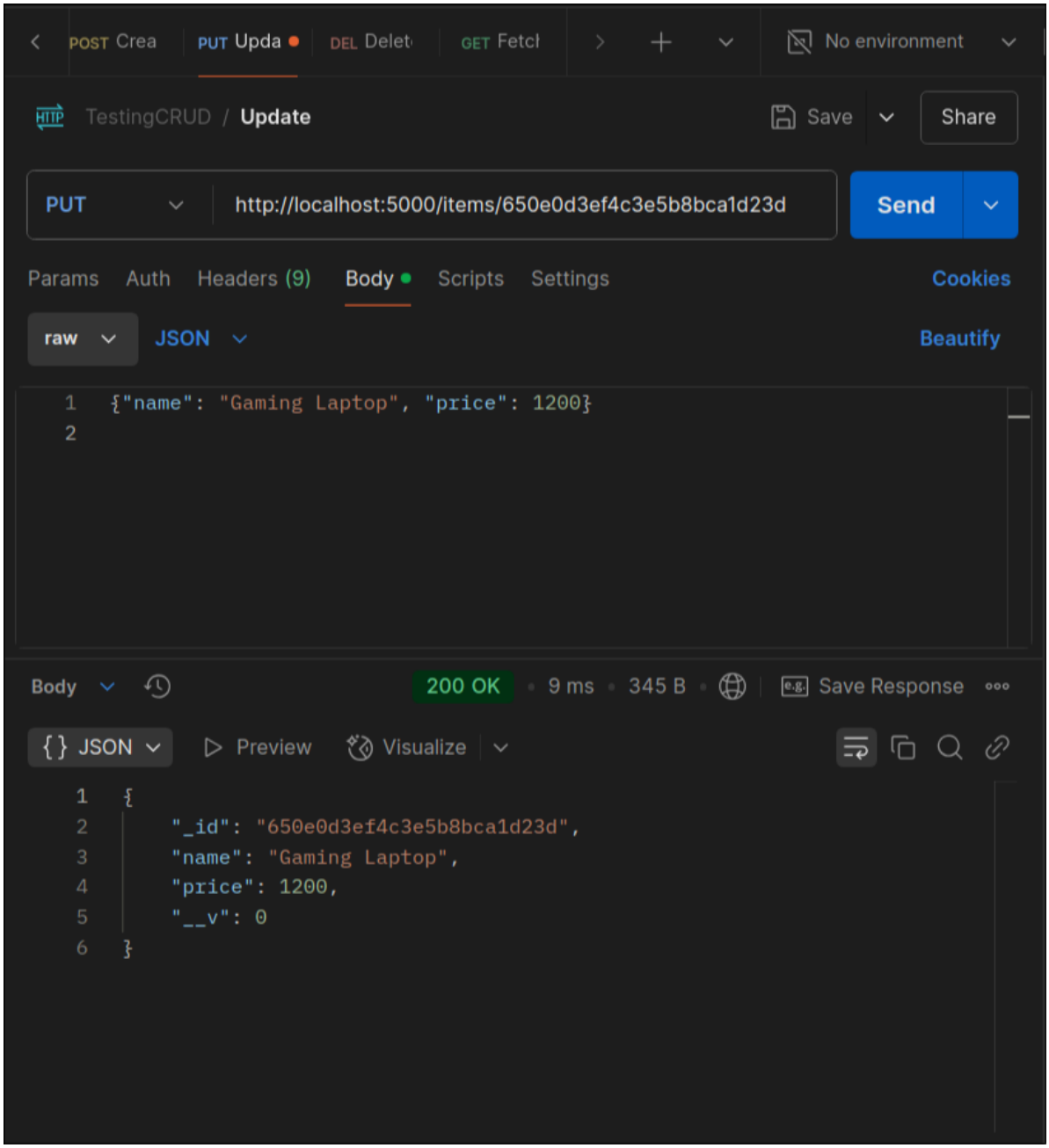
POST REQUEST:



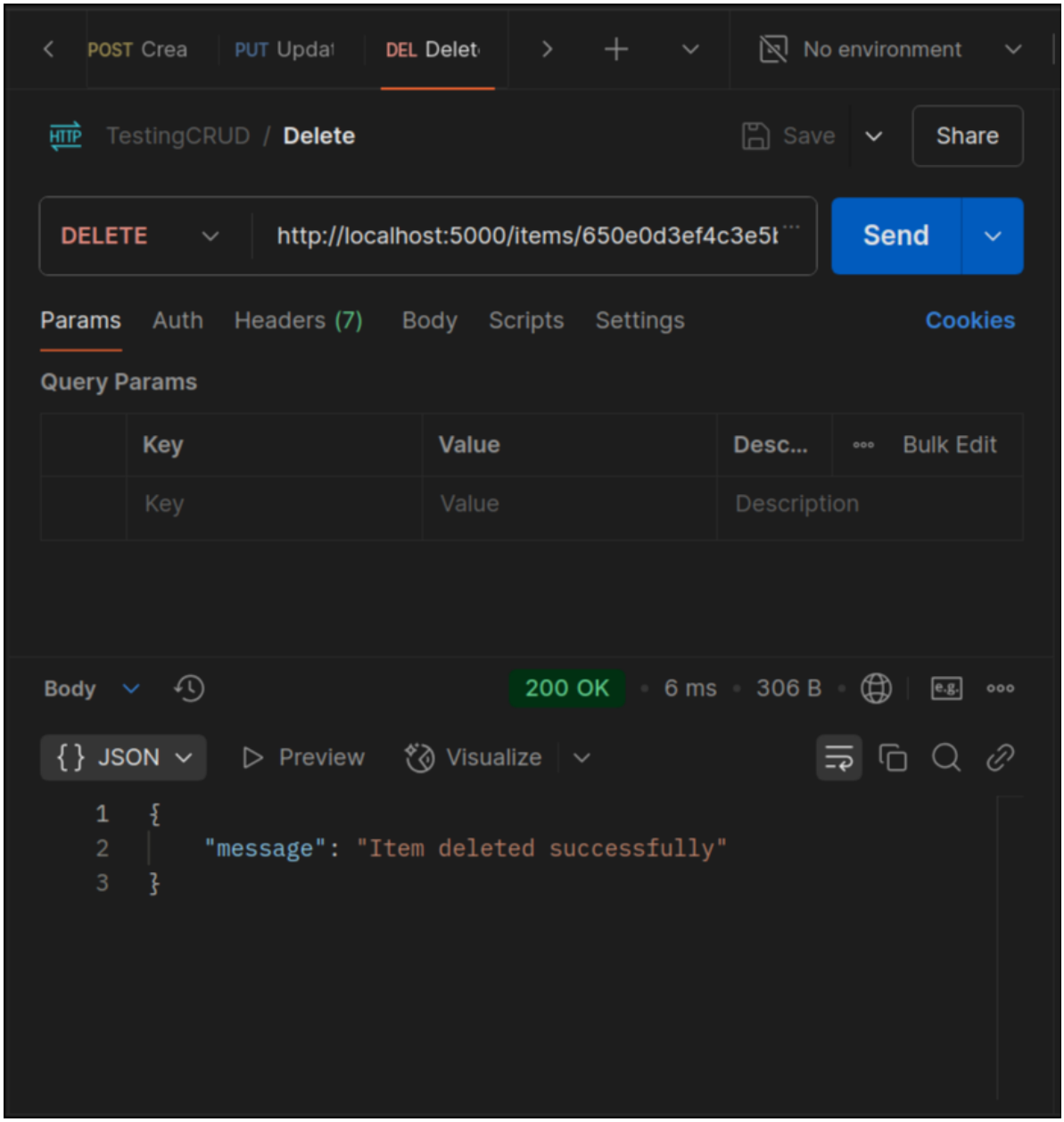
GET REQUEST:


POST Crea
PUT Updat
DEL Delet
GET Fetc
No environment
TestingCRUD / Fetch
Save
Share
GET
http://localhost:5000/items/
Send
Params
Auth
Headers (7)
Body
Scripts
Settings
Cookies
Query Params
Key
Value
Description
Bulk Edit
Key
Value
Description
Body
200 OK
12 ms
412 B
Save Response
{} JSON
Preview
Visualize
[
{
"_id": "67d4fa446c1d41d1164054a5",
"name": "tomato",
"price": 100,
"__v": 0
},
{
"_id": "650e0d3ef4c3e5b8bca1d23d",
"name": "Laptop",
"price": 1000,
"__v": 0
}
]

PUT REQUEST:


POST Crea
PUT Upda
DEL Delet
GET Fetcl
No environment
TestingCRUD / Update
Save
Share
PUT
http://localhost:5000/items/650e0d3ef4c3e5b8bca1d23d
Send
Params
Auth
Headers (9)
Body
Scripts
Settings
Cookies
raw
JSON
Beautify
1 {"name": "Gaming Laptop", "price": 1200}
2
Body
200 OK
9 ms
345 B
Save Response
{} JSON
Preview
Visualize
1 {
2 "_id": "650e0d3ef4c3e5b8bca1d23d",
3 "name": "Gaming Laptop",
4 "price": 1200,
5 "__v": 0
6 }

DELETE REQUEST:



Experiment 11

Aim:- Task management tool: Login/Register to the application, add daily tasks, Assign a due date of completion, Mark them as complete/incomplete, and View weekly/monthly statistics of their to-dos.

Code:-

FRONTEND:

/home/abhay/WebstormProjects/task_management_tool/src/App.js

```
import { Routes, Route } from "react-router-dom";
import React, {} from 'react';
import SignUp from "./SignUp";
import Home from "./Home";
import CreateTask from "./CreateTask";
import {
    Box, Divider, Link,
    Typography, Grid, Grid2
} from "@mui/material";
import {ThemeProvider, createTheme} from '@mui/material/styles';
import CssBaseline from '@mui/material/CssBaseline';
import Login from "./Login";
import Dashboard from "./Dashboard";
import ButtonAppBar from "./ButtonAppBar";
import ProtectedRoute from "./ProtectedRoute";
import AuthContext, {AuthProvider} from "./AuthContext";

const darkTheme = createTheme({
    palette: {
        mode: 'light',
    },
});

function App() {
    //const { token } = useContext(AuthContext);
    return (
        <ThemeProvider theme={darkTheme}>
            <CssBaseline/>
            <React.Fragment>
                    <AuthProvider>
                        <ButtonAppBar/>
                        <Routes>
                            <Route path="/" element={<Home/>}/>
                            <Route path="/login" element={<Login/>}/>
                            <Route path="/signup"
element={<SignUp/>}/>
                            <Route path="/createtask"
element={<CreateTask/>}/>
```

```
                            <Route path="/dashboard"
element={<Dashboard/>}/>
                            <Route element={<ProtectedRoute />}>
                                <Route path="/dashboard"
element={<Dashboard />} />
                            </Route>
                            <Route path="*" element={<Login />} />
                        </Routes>
                    </AuthProvider>
                <Footer/>
            </React.Fragment>
        </ThemeProvider>
    );
}

function Footer() {
    return (

        <Box component="footer"
             sx={{
                 backgroundColor: "#1976d2",
                 textAlign: "center",
                 fontFamily: "sans-serif",
                 fontSize: 25,
                 boxShadow: "0px -2px 10px rgba(0, 0, 0, 0.3)", //
Darker shadow
                 zIndex: 1000, // Ensures it's above other elements
                 color: 'white',
                 bottom: 0,
                 width: "auto%",
                 minHeight: "200px",


             }}>
            <Grid2 sx={{
                alignContent: "center",
                alignItems: "center",

                justifyContent: "center",
                m: 1, p: 1

            }} container spacing={2}>
                <Grid2 sx={{
                    '--Grid-borderWidth': '1px',
                    borderRight: 'var(--Grid-borderWidth) solid',
                    borderColor: 'divider'
                }} size={5}>
                    <Typography variant={"body1"} sx={{m: 1, p: 1}}
>Tasks.Web.App
                        © {new Date().getFullYear()} Copyright
```

```
</Typography>
                    <Typography variant={"body1"} sx={{m: 1, p: 1}}
>Made by abhay-byte</Typography>
                </Grid2>
                <Grid2 size={5}>
                    <Typography variant={"body1"} sx={{m: 1, p: 1}}>
                        <Link href="/src/privacy-policy"
color="inherit" underline="hover">
                            Privacy Policy
                        </Link>
                    </Typography>
                    <Typography variant={"body1"} sx={{m: 1, p: 1}}>
                        <Link href="/src/terms-conditons"
color="inherit" underline="hover">
                            Terms and Conditions
                        </Link>
                    </Typography>

                </Grid2>
            </Grid2>


        </Box>
    );
}
export default App;
```

/home/abhay/WebstormProjects/task_management_tool/src/ButtonAppBar.js

```
import {useNavigate} from "react-router-dom";
import {AppBar, Box, IconButton, Toolbar, Typography} from
"@mui/material";
import tmsicon from "./Images/tms1024.png";
import React from "react";
import Logout from "./Logout";

export default function ButtonAppBar() {

    let navigate = useNavigate();
    const naviagteToHome = (e) => {
        navigate("/");

    }


    return (
        <Box sx={{flexGrow: 1}}>
            <AppBar position="fixed">
                <Toolbar>
```

```
                    <Typography variant="h6" component="div"
sx={{flexGrow: 1, display:"flex", fontFamily: 'sans-serif', fontSize:
15, justifyContent: 'space-between'}}>
                        <Box sx={{flexGrow: 1}}>
                            <IconButton color="inherit"
onClick={naviagteToHome}>
                                <img
                                    srcSet={`${tmsicon}?
w=164&h=164&fit=crop&auto=format&dpr=2 2x`}
                                    src={`${tmsicon}?
w=164&h=164&fit=crop&auto=format`}
                                    alt={'Tms Icon'}
                                    loading="lazy"
                                    width="24"
                                    height="24"
                                />
                            </IconButton>
                            Tasks.Web.App
                        </Box>

                        <Box alignContent={"right"}>
                            <Logout/>
                        </Box>

                    </Typography>
                </Toolbar>
            </AppBar>
        </Box>
    );
}
```

/home/abhay/WebstormProjects/task_management_tool/src/Home.js

```
import {motion} from "motion/react";
import {useNavigate} from "react-router-dom";
import React, {useState, useEffect, useRef, useContext} from 'react';
import Button from '@mui/material/Button';
import {
    AppBar,
    Box, ImageList, ImageListItem,
    Typography
} from "@mui/material";
import {ThemeProvider, createTheme} from '@mui/material/styles';
import CssBaseline from '@mui/material/CssBaseline'
import AuthContext from "./AuthContext";

const darkTheme = createTheme({
    palette: {
        mode: 'light',
```

```
    },
});

function Home() {
    const {token} = useContext(AuthContext);
    const itemData = [
        {
            img: 'https://storage.googleapis.com/profit-prod/wp-
content/uploads/2022/07/d4eaf149-task-management.jpg',
            title: 'Bed',
        },
        {
            img: 'https://thumbs.dreamstime.com/z/business-task-
management-vector-illustration-set-isolated-white-background-
effective-corporate-time-tasks-planning-scheduling-157821557.jpg',
            title: 'Books',
        },
        {
            img: 'https://storage.googleapis.com/profit-prod/wp-
content/uploads/2022/07/d4eaf149-task-management.jpg',
            title: 'Bed',
        },
        {
            img:
'https://www.cflowapps.com/wp-content/uploads/2018/07/task-management-
process.png',
            title: 'Sink',
        },
        {
            img:
'https://www.ntaskmanager.com/wp-content/uploads/2021/02/Task-
management-vs-project-management.jpg',
            title: 'Kitchen',
        },
        {
            img: 'https://storage.googleapis.com/profit-prod/wp-
content/uploads/2022/07/d4eaf149-task-management.jpg',
            title: 'Bed',
        },
        {
            img: 'https://images.unsplash.com/photo-1588436706487-
9d55d73a39e3',
            title: 'Blinds',
        },
        {
            img:
'https://www.cflowapps.com/wp-content/uploads/2018/07/task-management-
process.png',
            title: 'Storage',
        },
```

```
        {
            img: 'https://wallpaperaccess.com/full/5137774.jpg',
            title: 'Candle',
        },
        {
            img: 'https://storage.googleapis.com/profit-prod/wp-
content/uploads/2022/07/d4eaf149-task-management.jpg',
            title: 'Bed',
        },

    ];
    const navigate = useNavigate();
    const navigateToSignUp = (e) => {
        navigate("/signup");
    }
    const navigateToLogin = (e) => {
        navigate("/login");
    }
    const navigateToDashboard = (e) => {
        navigate("/dashboard");
    }
    return (
        <ThemeProvider theme={darkTheme}>
            <CssBaseline/>
            <React.Fragment>
                <motion.div initial={{scale: 0}} animate={{scale: 1}}>
                    <Box
                        sx={{
                            display: 'flex',
                            flexDirection: 'column',
                            justifyContent: 'space-between',
                            alignItems: 'center',
                        }}>
                        <Box component="section" sx={{
                            display: "flow",
                            justifyContent: "center",
                            alignItems: "center",
                            m: 1, p: 1,
                            width: 'auto%',
                            maxWidth: 800,
                            minWidth: 300,
                            textAlign: "center",
                            fontFamily: 'sans-serif',
                        }}>

                            <ImageList variant="masonry" cols={3}
gap={8} sx={{mt: 10}}>
                                {itemData.map((item) => (
                                    <ImageListItem key={item.img}>
                                        <img
```

```
                                        srcSet={`${item.img}?
w=248&fit=crop&auto=format&dpr=2 2x`}
                                        src={`${item.img}?
w=248&fit=crop&auto=format`}
                                        alt={item.title}
                                        loading="lazy"
                                    />
                                </ImageListItem>
                            ))}
                        </ImageList>

                        <Typography variant={'h4'} sx={{m: 3}}
>Welcome to Tasks.Web.App</Typography>

                        <Typography variant={'body1'} sx={{m: 2}}
>A place where you can create tasks, manage your
                            task and do it on time.</Typography>

                        <Typography variant={'body1'} sx={{m: 2}}
>Sign up, if you are new or login to
                            use.</Typography>

                        {token &&
                            <Button variant={"outlined"}
style={{margin: 10, marginBottom: 50, display: "inline"}}
                                    onClick={navigateToDashboard}
>Go To Dashboard</Button>}

                        {!token &&
                            <Button variant={"outlined"}
style={{margin: 10, marginBottom: 50, display: "inline"}}
                                    onClick={navigateToSignUp}
>Register</Button>}

                        {!token &&
                            <Button variant={"outlined"}
style={{margin: 10, marginBottom: 50, display: "inline"}}
                                    onClick={navigateToLogin}
>Login</Button>}


                    </Box>
                </Box>

            </motion.div>
        </React.Fragment>
    </ThemeProvider>
  );
}
```

```
export default Home;
```

/home/abhay/WebstormProjects/task_management_tool/src/SignUp.js

```
import React, {useState} from "react";
import {motion} from "motion/react";
import {useNavigate} from "react-router-dom"
import {
    Box, Checkbox,
    FormControlLabel, FormGroup,
    TextField,
    Typography
} from "@mui/material";
import {createTheme, ThemeProvider} from "@mui/material/styles";
import CssBaseline from "@mui/material/CssBaseline";
import Button from "@mui/material/Button";
import ScrollToTop from "./ScrollToTop";
import axios from "axios";
const darkTheme = createTheme({
    palette: {
        mode: 'light',
    },
});

function SignUp() {
    ScrollToTop();
    const label = {inputProps: {'aria-label': 'Checkbox demo'}};
    const [showPassword, setShowPassword] = React.useState(false);
    const navigate = useNavigate();
    const handleClickShowPassword = () => setShowPassword((show) => !
show);

    const handleSignup = async (e) => {
        e.preventDefault();
        try {
            const data = new FormData(e.currentTarget);
            const username = data.get('username');
            const password = data.get('password');
            const email = data.get('email');
            const res = await
axios.post("http://localhost:5000/signup", { username, password
,email});
            navigate("/login");
            alert(res.data.message);
        } catch (error) {
            alert(error.response?.data?.message || "Signup failed");
        }
    };
    const handleMouseDownPassword = (event) => {
```

```
        event.preventDefault();
    };

    const handleMouseUpPassword = (event) => {
        event.preventDefault();
    };

    const navigateToLogin = (e) => {
        navigate("/login");
    }
    return (
        <div>
            <ThemeProvider theme={darkTheme}>
                <CssBaseline/>
                <React.Fragment>

                    <motion.div initial={{scale: 0}} animate={{scale:
1}} className="App">
                        <Box sx={{
                            alignItems: "center",
                            justifyContent: "center",
                            display: "flex",

                        }}>
                            <Box component="section" sx={{
                                display: "flex",
                                flexDirection: "column",
                                m: 4, p: 4,
                                borderRadius: 4,
                                boxShadow: 3,
                                maxWidth: 500,
                                minWidth: 300,
                                width: "auto%",
                                fontSize: 15,
                                mt: 10,
                                fontFamily: 'sans-serif',
                                textAlign: "center",


                            }}>
                                <Typography variant={'inherit'}>Sign
Up</Typography>
                                <Box
                                    component="form"
                                    sx={{
                                        '& .MuiTextField-root': {
                                            m: 1, p: 1, width: '25ch',
                                            display: "flex",

                                            color: "grey",
```

```
                                    }
                                }}
                                autoComplete="off"
                                onSubmit={handleSignup}>

                                <div style={{}}>
                                    <img
src="https://static.vecteezy.com/system/resources/previews/003/689/228
/original/online-registration-or-sign-up-login-for-account-on-
smartphone-app-user-interface-with-secure-password-mobile-application-
for-ui-web-banner-access-cartoon-people-illustration-vector.jpg"
                                        alt="Description"
width="300"/>
                                </div>
                                <TextField sx={{display: "inline-
block"}} name="username" required label="Username"
                                        variant="standard"
                                        placeholder="Enter
Username" id='username' margin="normal"></TextField>
                                <TextField sx={{display: "block"}}
name="email" required label="Email"
                                        variant="standard"
                                        placeholder="Enter
Email" id='email' margin="normal"></TextField>
                                <TextField sx={{display: "block"}}
                                        name="password"
required label="Password"
                                        variant="standard"
placeholder="Password" id='password'
                                        type={showPassword ?
'text' : 'password'}

                                />
                                <TextField sx={{display: "block"}}
name="confirm_password" required
                                        label="Confirm
Password"
                                        variant="standard"
placeholder="Re-Enter Password" id='confirm_password'
                                        margin="normal"
type={showPassword ? 'text' : 'password'}></TextField>

                                <FormGroup sx={{textAlign: 'left',
fontSize: 16, color: 'grey', m: 1}}>
                                    <FormControlLabel sx={{mt: 1}}
required control={<Checkbox/>}
                                                      label="Agree
```

```
to our Terms and Conditions."/>
                                        <FormControlLabel sx={{mb: 1}}
required control={<Checkbox/>}
                                                          label="Agree
to our Privacy Policy."/>
                                    </FormGroup>
                                    <Button sx={{}}
variant={"outlined"} type={"submit"}>Create Account</Button>

                                    <Typography sx={{display: 'block',
mt: 2}} variant={'overline'}>Already have an
                                        account,
                                    </Typography>
                                    <Button sx={{fontSize: 12}}
onClick={navigateToLogin}>Sign in here.</Button>

                                </Box>
                            </Box>
                        </Box>
                    </motion.div>
                </React.Fragment>
            </ThemeProvider>
        </div>
    );
}

export default SignUp;
```

/home/abhay/WebstormProjects/task_management_tool/src/Login.js

```
import React, {useContext, useState} from "react";
import {useNavigate} from "react-router-dom";
import { motion } from "motion/react";
import {
    Box, Checkbox,
    FormControlLabel, FormGroup,
    TextField,
    Typography
} from "@mui/material";
import {createTheme, ThemeProvider} from "@mui/material/styles";
import CssBaseline from "@mui/material/CssBaseline";
import Button from "@mui/material/Button";
import ScrollToTop from "./ScrollToTop";
import axios from "axios";
import AuthContext from "./AuthContext";
const darkTheme = createTheme({
    palette: {
        mode: 'light',
    },
});
```

```
function Login(){
    ScrollToTop();
    const navigate = useNavigate();
    const label = { inputProps: { 'aria-label': 'Checkbox demo' } };
    const [showPassword, setShowPassword] = React.useState(false);
    const [username, setUsername] = useState("");
    const [password, setPassword] = useState("");
    const { login } = useContext(AuthContext);

    const handleLogin = async (e) => {
        e.preventDefault();
        try {
            const data = new FormData(e.currentTarget);
            const username = data.get("username");
            const password = data.get("password");

            const res = await
axios.post("http://localhost:5000/login", { username, password });
            login(res.data.token);
            navigate("/dashboard");
        } catch (error) {
            alert(error.response?.data?.message || "Login failed");
        }
    };
    const handleClickShowPassword = () => setShowPassword((show) => !
show);

    const handleMouseDownPassword = (event) => {
        event.preventDefault();
    };

    const handleMouseUpPassword = (event) => {
        event.preventDefault();
    };

    const navigateToSignup= (e) => {
        navigate("/signup");
    }
    return (
        <div>
            <ThemeProvider theme={darkTheme}>
                <CssBaseline />
                <React.Fragment>

                    <motion.div initial={{ scale: 0 }} animate={{
scale: 1 }} className="App">
                        <Box sx={{alignItems: "center",justifyContent:
"center",display: "flex",alignContent: "center"}}>
```

```
                            <Box component="section" sx={{
                                display: "flex",
                                flexDirection: "column",
                                justifyContent: "center",
                                alignItems: "center",
                                m:4, p: 4,
                                borderRadius: 4,
                                boxShadow: 3,
                                maxWidth: 500,
                                minWidth: 300,
                                width: "auto%",
                                fontSize: 15,
                                mt:10,
                                fontFamily: 'sans-serif',
                                textAlign: "center",

                            }}>

                                <Box
                                    component="form"
                                    sx={{ '& .MuiTextField-root': { m:
1, p: 1 , width: '25ch',
                                            display: "flex" ,}}}
                                    autoComplete="off"
                                    onSubmit={handleLogin}
                                >

                                    <Typography variant={'inherit'}
>Login</Typography>
                                    <div style={{display:"inline",
alignContent:"right"}}>
                                        <img src =
"https://static.vecteezy.com/system/resources/previews/011/432/528/
original/enter-login-and-password-registration-page-on-screen-sign-in-
to-your-account-creative-metaphor-login-page-mobile-app-with-user-
page-flat-illustration-vector.jpg"
                                            alt="Description"
width="250" />
                                    </div>
                                    <TextField sx={{display:"inline"}}
name="username" required label="Username" variant="standard"
placeholder="Enter Username" id = 'username' margin =
"normal"></TextField>
                                    <TextField sx={{display:"block"}}
name="password" required label="Password" variant="standard"
placeholder="Enter Password" id = 'password' margin = "normal"
                                        type={showPassword ?
'text' : 'password' } ></TextField>
                                    <FormGroup sx={{textAlign: 'left',
fontSize: 16, color: 'grey', m: 1}}>
```

```
                                        <FormControlLabel sx={{mt: 1}}
control={<Checkbox/>}
                                                          label="Stay
Logged in for the session."/>
                                    </FormGroup>

                                    <Button sx={{mt:4}}
variant={"outlined"} type={"submit"}>Login Account</Button>
                                    <Typography sx={{display: 'block',
mt: 2}} variant={'overline'}>Don't have an
                                        account,
                                    </Typography>
                                    <Button sx={{fontSize: 12}}
onClick={navigateToSignup}>Sign up here.</Button>
                                </Box>
                            </Box>
                        </Box>

                    </motion.div>
                </React.Fragment>
            </ThemeProvider>
        </div>
    );
}

export default Login;
```

/home/abhay/WebstormProjects/task_management_tool/src/Dashboard.js

```
import {useNavigate} from "react-router-dom";
import * as React from 'react';
import Box from '@mui/material/Box';
import IconButton from '@mui/material/IconButton';
import Grid2 from '@mui/material/Grid';
import Typography from '@mui/material/Typography';
import {
    Backdrop,
    Card,
    CardActions,
    CardContent,
    Chip,
    FormControl, FormControlLabel,
    FormLabel, Radio,
    RadioGroup,
    useMediaQuery
} from "@mui/material";
import Button from "@mui/material/Button";
import EditOutlinedIcon from '@mui/icons-material/EditOutlined';
import DeleteOutlineOutlinedIcon from
'@mui/icons-material/DeleteOutlineOutlined';
```

```
import {PieChart} from '@mui/x-charts/PieChart';
import AddBoxOutlinedIcon from '@mui/icons-material/AddBoxOutlined';
import {motion} from "motion/react";
import ScrollToTop from "./ScrollToTop";
import AuthContext from "./AuthContext";
import {useContext, useEffect, useState} from "react";
import axios from "axios";


export default function Dashboard() {
    ScrollToTop();
    const months =
['Jan','February','March','April','May','June','July','August','Septem
ber','October','November','December'];
    const {user} = useContext(AuthContext);
    const userId = user?.userId;
    const navigate = useNavigate();
    const [dense, setDense] = React.useState(false);
    const [secondary, setSecondary] = React.useState(true);
    const isMobile = useMediaQuery("(max-width:600px)"); // Mobile:
Width ≤ 60
    const [tasks, setTasks] = useState([]);
    const [open, setOpen] = React.useState(false);
    const handleClose = () => {
        setOpen(false);
    };
    const [currentTask, setCurrentTask] = useState("");
    const [currentTaskId, setCurrentTaskId] = useState("");

    const handleOpen = (id, name) => {
        setCurrentTaskId(id);
        setCurrentTask(name);
        setOpen(true);
    };
    useEffect(() => {
        if(userId != null){
            const params = new
URLSearchParams([["userid",user["userId"]]]);
            axios.get('http://localhost:5000/tasks',{ params: {
userid: userId ,sort: 'asc'} })
                .then((response) => {
                    setTasks(response.data);
                })
                .catch((error) => {
                    console.error('There was an error fetching the
items!', error);
                });
        }

    }, []);
```

```
    const currentMonth = new Date().getMonth(); // Get current month
(0-11)
    const currentYear = new Date().getFullYear(); // Get current year

    const filteredTasks = tasks.filter(task => {
        const taskDate = new Date(task.startdate); // Convert
startdate to Date
        return taskDate.getMonth() === currentMonth &&
taskDate.getFullYear() === currentYear;
    });

    const taskCounts = filteredTasks.reduce(
        (acc, task) => {
            acc[task.status] = (acc[task.status] || 0) + 1;
            return acc;
        },
        { completed: 0, pending: 0, due: 0 }
    );

    const taskItems = [
        { id: 0, value: taskCounts.completed, label: "Completed",
color: "blue" },
        { id: 1, value: taskCounts.pending, label: "Pending", color:
"#1976d2" },
        { id: 2, value: taskCounts.due, label: "Due", color: "#E0E0E0"
},
    ];

    const [status, setStatus] = useState("Ongoing");

    const handleStatusChange = (event) => {
        setStatus(event.target.value);
    };

    const updateTask = async () => {
        try {
            const response = await
axios.put("http://localhost:5000/tasks", {
                taskid: currentTaskId,
                status
            });
            console.log("Task Updated:", response.data);
            handleClose(); // Close modal after update
            window.location.reload();
        } catch (error) {
            console.error("Error updating task:", error.response?.data
|| error.message);
        }
    };
```

```
    const deleteTask = async (taskId) => {
        try {
            await axios.delete(`http://localhost:5000/tasks/${taskId}
`); // Pass taskId in the URL
            console.log("Task Deleted:", taskId);

            window.location.reload(); // Refresh page after deletion
        } catch (error) {
            console.error("Error deleting task:", error.response?.data
|| error.message);
        }
    };

    const navigateToCreateTask = () => {
        navigate("/createTask");
    };
    return (
        <Box sx={{
            display: "flex", flexDirection: "Column", justifyContent:
"space-between", mt: 10,
            fontFamily: "sans-serif"
        }}>

            <motion.div initial={{scale: 0}} animate={{scale: 1}}
style={{margin: 20}}>
                <Box sx={{display: "flex", flexDirection: "column",
justifyContent: "space-between"}}>

                    <Typography color={"grey"} variant={'body2'}>
Welcome, </Typography>
                    <Box display="flex" alignItems="center" gap={2}
justifyContent={"space-between"} >
                        <Typography p={1} align={"left"}
variant={'h4'} gutterBottom>{user['username']}</Typography>

                        <Button onClick={navigateToCreateTask}
variant="outlined" startIcon={<AddBoxOutlinedIcon />} sx={{ display:
"flex", alignItems: "center" }}>
                            Create Task
                        </Button>
                    </Box>

                    <Box sx={{border:'1px
solid',borderColor:'#E0E0E0', borderRadius:2, mt:3,mb:3, display:
'flex', flexDirection: isMobile?'column':'row', justifyContent:
'space-around'
                    ,minWidth:'80%'}}>

                        <Typography sx={{m:2}} >Statistics Of
```

```
{months[new Date().getMonth()]}</Typography>
                        <PieChart sx={{p:2}}

                                  series={[
                                      {
                                          data:taskItems,
                                          innerRadius: 20,
                                          outerRadius: isMobile ? 90 :
100,

                                          paddingAngle: 5,
                                          cornerRadius: 5,
                                          startAngle: -45,
                                          endAngle: 270,
                                          cx: isMobile ? 120 : 150,
                                          cy: isMobile ? 120 : 150,

                                      },
                                  ]}
                                  labelPosition="outside"
                                  width={isMobile ? 360 : 400}
                                  height={isMobile ? 280 : 300}

                        />
                        <Box display={"flex"} flexDirection={'column'}
justifyContent={'space-around'} alignItems={'center'} sx={{m:4, p:4}}>
                            <Typography color={"grey"}
variant={"body1"}>Task completed: {taskItems[0].value}</Typography>
                            <Typography color={"grey"}
variant={"body1"}>Task Ongoing: {taskItems[1].value}</Typography>
                            <Typography color={"grey"}
variant={"body1"} gutterBottom={4}>Task Due:
{taskItems[2].value}</Typography>
                        </Box>

                    </Box>

                </Box>
                <Typography variant="h4" gutterBottom>
                    Task List
                </Typography>
                <Grid2 container spacing={2}>
                    {tasks.map((task, index) => (
                        <Grid2 item xs={12} sm={6} md={4} key={index}>
                            <Card variant="outlined" sx={{ display:
"block", flexDirection: "column", justifyContent: "space-between" }}>
                                <CardContent>
                                    <Typography variant="body2"
color="grey">Task Name</Typography>
                                    <Typography
variant="h6">{task.taskname}</Typography>
```

```
                                    <Typography variant="body2"
color="text.secondary">
                                        Description:
{task.taskdescription}
                                    </Typography>

                                    <Typography variant="body2"
display="block">
                                        <strong>Start:</strong>
{task.startdate} | <strong>End:</strong> {task.enddate}
                                    </Typography>

                                    <Chip
                                        variant="outlined"
                                        label={task.status}
                                        color={task.status ===
"completed" ? "success" : "warning"}
                                        sx={{ marginTop: 1 }}
                                    />

                                    <CardActions>
                                        <IconButton onClick={() =>
handleOpen(task.taskid, task.taskname)} color="primary" aria-
label="edit">
                                            <EditOutlinedIcon />
                                        </IconButton>
                                        <IconButton onClick={() =>
deleteTask(task.taskid)} color="error" aria-label="delete">
                                            <DeleteOutlineOutlinedIcon
/>
                                        </IconButton>
                                    </CardActions>
                                </CardContent>
                            </Card>
                        </Grid2>
                    ))}
                </Grid2>

                {/* Backdrop for displaying selected task */}
                <Backdrop
sx={{alignContent:'center',direction:'flex',flexDirection: 'column',
color: "#fff", zIndex: (theme) => theme.zIndex.drawer + 1 }}
open={open} onClick={handleClose}>
                    <Card
sx={{alignContent:'center',direction:'flex',flexDirection: 'column',
padding: 2, backgroundColor: "white", borderRadius: 2 }} onClick={(e)
=> e.stopPropagation()}>
                        <Typography variant="h6" color="black">Task
Name: {currentTask}</Typography>
                        <Typography variant="body2"
```

```
color="text.secondary">Task ID: {currentTaskId}</Typography>

                        <FormControl margin="normal"
sx={{alignContent:'center',direction:'flex',flexDirection: 'column'}}>
                            <FormLabel id="task-status-label">Task
Status</FormLabel>
                            <RadioGroup
                                aria-labelledby="task-status-label"
                                value={status}
                                name="taskstatus"
                                onChange={handleStatusChange}
                            >
                                <FormControlLabel value="completed"
control={<Radio />} label="Completed" />
                                <FormControlLabel value="pending"
control={<Radio />} label="Pending" />
                            </RadioGroup>
                        </FormControl>

                        <Button variant="outlined"
onClick={updateTask}>Update</Button>
                    </Card>
                </Backdrop>

            </motion.div>
        </Box>
    );
};
```

/home/abhay/WebstormProjects/task_management_tool/src/CreateTask.js

```
import { BrowserRouter as Router, Routes, Route, Link, useNavigate }
from "react-router-dom";
import React, {useState, useEffect, useContext} from 'react';
import axios from 'axios';
import Button from '@mui/material/Button';
import {
    Alert, AlertTitle,
    Box,
    FormControl, FormControlLabel, FormLabel,
    Radio, RadioGroup,
    TextField,
} from "@mui/material";
import { ThemeProvider, createTheme } from '@mui/material/styles';
import CssBaseline from '@mui/material/CssBaseline';
import AuthContext from "./AuthContext";
const darkTheme = createTheme({
    palette: {
```

```
        mode: 'light',
    },
});

function App() {
    const navigation = useNavigate();
    const [tasks, setTasks] = useState([]);
    const [formData, setFormData] = useState(new FormData());
    const [responseMessage, setResponseMessage] = useState("");
    const [alertContent, setAlertContent] = useState('');
    const [s_alert, setSAlert] = useState(false);
    const [e_alert, setEAlert] = useState(false);
    const {user} = useContext(AuthContext);


    const CreateNewTask = async (e) => {
        e.preventDefault(); // Prevent page reload
        let data = new FormData(e.currentTarget);

        let taskdata = {
            taskname: data.get('taskname'),
            taskdescription: data.get('taskdescription'),
            status: data.get('taskstatus'),
            startdate: data.get('startdate'),
            enddate: data.get('enddate'),
            userid: user?.userId, // Ensure user object exists
        };

        try {
            const response = await
axios.post('http://localhost:5000/tasks', taskdata, {
                headers: { "Content-Type": "application/json" },
            });

            setResponseMessage(response.data.message || "Success!");
            setSAlert(true);
            navigation("/dashboard");

        } catch (error) {
            setResponseMessage("Error submitting data");
            setEAlert(true);
            console.error("Error:", error);
        }
    };

    return (

        <ThemeProvider theme={darkTheme}>
            <CssBaseline />
            <React.Fragment>
```

```
                {s_alert ? <Alert severity="success">
                    <AlertTitle>Success</AlertTitle>
                    Task Created Successfully!
                </Alert>:<></>}
                {e_alert ? <Alert severity="error">
                    <AlertTitle>Error</AlertTitle>
                    Task Not Created!
                    </Alert>:<></>}

                <div className="App">
                    <Box component="section" sx={{
                        display: "flex",
                        justifyContent: "center",
                        alignItems: "center",
                        m:4, p: 4,
                        borderRadius: 4,
                        boxShadow: 5,
                        width: 'auto%',
                    mt:12}}>

                        <Box
                            component="form"
                            sx={{ '& .MuiTextField-root': { m: 1, p: 1
, width: '25ch' } }}
                            autoComplete="off"
                            onSubmit={CreateNewTask}
                        >
                            <h1 style={{ textAlign: "center" }}>Create
Task</h1>
                            <TextField name="taskname" required
label="Task Name" variant="outlined" placeholder="Enter Task" id =
'task' margin = "dense"></TextField>

                            <TextField required name="taskdescription"
label="Task Desctiption" variant="outlined" placeholder="Enter
Description" id = 'desc' margin = "dense"></TextField>
                            <br></br>
                            <TextField label="" name="startdate"
type={"date"} variant="outlined" id = 's_date' margin = "dense"
helperText={"Enter Start Date"}></TextField>

                            <TextField label="" name="enddate"
type={"date"} variant="outlined"  id = 's_date' margin = "dense"
helperText={"Enter End Date"}></TextField>
                            <br></br>
                            <FormControl margin={"normal"}>
                                <FormLabel id="demo-radio-buttons-
group-label">Task Status</FormLabel>
                                <RadioGroup
                                    aria-labelledby="demo-radio-
```

```
buttons-group-label"
                                    defaultValue="Ongoing"
                                    name="taskstatus"
                                >
                                    <FormControlLabel
value="completed" control={<Radio />} label="Completed" />
                                    <FormControlLabel value="pending"
control={<Radio />} label="Pending" />
                                </RadioGroup>
                            </FormControl>
                            <br></br>
                            <Button   style={{
                                display: 'block',
                                margin: '10px auto',

                            }} variant={"outlined"} type={"submit"}
>Add Task</Button>

                        </Box>
                    </Box>
                </div>
            </React.Fragment>
        </ThemeProvider>
    );
}


function Main() {
    const navigate = useNavigate(); // Hook for navigation

    return (
        <div>
            <h1>Welcome to App.js</h1>
            <button onClick={() => navigate("/home")}>Go to
Home</button>
        </div>
    );
}

export default App;
```

/home/abhay/WebstormProjects/task_management_tool/src/AuthContext.js

```
import { createContext, useState, useEffect } from "react";
import { jwtDecode } from 'jwt-decode'

const AuthContext = createContext(undefined);
```

```
const GetUser = (token)=>{
    try{
        if (token) {
            const user = jwtDecode(token);
            return user;
        }
    }
    catch(err){
        console.log(err);
    }

};

export const AuthProvider = ({ children }) => {
    const [token, setToken] = useState(localStorage.getItem("token")
|| "");
    const user = GetUser(token);

    useEffect(() => {
        localStorage.setItem("token", token);
        console.log(user);
    }, [token]);

    const login = (newToken) => {
        setToken(newToken);
    };

    const logout = () => {
        setToken("");
        localStorage.removeItem("token");
    };

    return (
        <AuthContext.Provider value={{user, token, login, logout }}>
            {children}
        </AuthContext.Provider>
    );
};

export default AuthContext;
```

BACKEND:

/home/abhay/WebstormProjects/tms_mongo_backend/server.js

```
import express from 'express';
import mongoose from 'mongoose';
import cors from 'cors';
import Tasks from "./models/tasks.js";
import jwt from 'jsonwebtoken';
import bcrypt from "bcryptjs";
import dotenv from "dotenv";
import crypto from 'crypto';
import 'dotenv/config'
import tasks from "./models/tasks.js";

const output = dotenv.config();
console.log(output);
const app = express();
const PORT = 5000;

// Middleware
app.use(cors());
app.use(express.json());

const { JWT_SECRETS, MONGO_URI } = process.env;
// MongoDB Connection
mongoose.connect(MONGO_URI, {
    useNewUrlParser: true,
    useUnifiedTopology: true,
});

const userSchema = new mongoose.Schema({
    username: { type: String, unique: true},
    password: { type: String, required: true },
    email: { type: String, unique: true, required: true },
});

const User = mongoose.model("User", userSchema);

const db = mongoose.connection;
db.on('error', (error) => console.error(error));
db.once('open', () => console.log('Connected to MongoDB'));

// Routes
app.get('/', (req, res) => {
    res.send('Hello from Express');
});

app.listen(PORT, () => {
    console.log(`Server running on port ${PORT}`);
```

```
});


// Create Item
app.post('/tasks', async (req, res) => {
    const { taskname, taskdescription, startdate,enddate,status,userid
} = req.body;

    const taskid = crypto.randomUUID();
    const task = new Tasks({
        taskname,
        taskdescription,
        startdate,
        enddate,
        status,
        userid,
        taskid,
    });

    try {
        const newtask = await task.save();
        res.status(201).json(newtask);
    } catch (err) {
        res.status(400).json({ message: err.message });
    }
});

// Get all Items
app.get('/tasks', async (req, res) => {
    try {
        const id = req.query.userid;
        const taskid = req.query.taskid;
        const sortBy = req.query.sort ?? 'enddate';
        const order = req.query.order === 'asc' ? 1 : -1;
        const argument = {[`${sortBy}`]: order };
        const taskStatus = req.query.status;
        const limit = parseInt(req.query.limit);
        const page = parseInt(req.query.page);

        if (taskid)
        {
            const task = await Tasks.find({'taskid':taskid})
                .exec();
            res.json(task);
            console.log(task);
        }
        else{
            const task = await
Tasks.find({'userid':id}).sort(argument)
                .exec();
```

```
            res.json(task);
            console.log(task);
        }

    } catch (err) {
        res.status(500).json({ message: err.message });
    }
});

// Update Item
app.put('/tasks', async (req, res) => {
    const { status,taskid } = req.body;

    try {
        const updatedTask = await Tasks.findOneAndUpdate(
            { taskid },  // Find by taskid, not _id
            { status },  // Update only the status field
            { new: true } // Return the updated document
        );

        if (!updatedTask) {
            return res.status(404).json({ message: "Task not found"
});
        }

        res.status(200).json(updatedTask);
    } catch (error) {
        res.status(400).json({ message: error.message });
    }

})

//Delete item
app.delete('/tasks/:id', async (req, res) => {
    const taskid = req.params.id;
    try {

        await Tasks.findOneAndDelete(taskid);
        res.status(200);
    }
    catch (error) {
        res.status(400).json({ message: error.message });
    }
})

// **Signup Route**
app.post("/signup", async (req, res) => {
    try {
        const { username, password, email } = req.body;
```

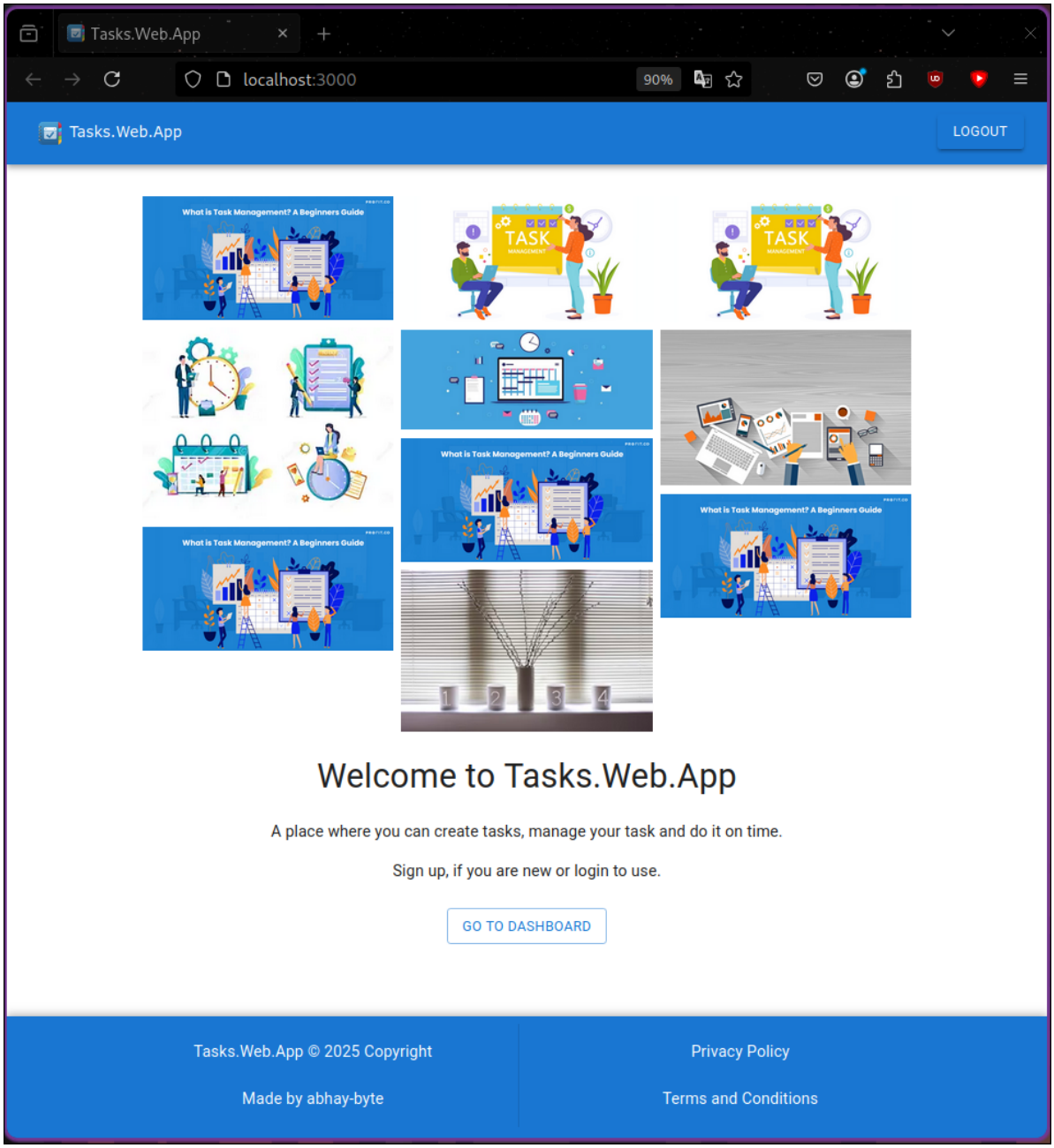
```
        const existingUser = await User.findOne({ username });
        if (existingUser) return res.status(400).json({ message: "User
already exists" });

        const hashedPassword = await bcrypt.hash(password, 10);
        const newUser = await User.create({ username, password:
hashedPassword, email });

        res.json({ message: "User registered successfully", user:
newUser });
    } catch (error) {
        res.status(500).json({ message: "Error signing up", error });
    }
});

// **Login Route**
app.post("/login", async (req, res) => {
    try {
        const { username, password } = req.body;

        const user = await User.findOne({ username });
        if (!user) return res.status(401).json({ message: "Invalid
credentials" });

        const isMatch = await bcrypt.compare(password, user.password);
        if (!isMatch) return res.status(401).json(
            { message: "Invalid credentials" }
        );

        const token = jwt.sign({ userId: user._id, username },
JWT_SECRETS, { expiresIn: "1h" });
        res.json({ token });
    } catch (error) {
        res.status(500).json({ message: "Error logging in", error });
    }
});

const authenticate = (req, res, next) => {
    const token = req.headers.authorization?.split(" ")[1];
    if (!token) return res.status(401).json({ message: "Unauthorized"
});

    try {
        const decoded = jwt.verify(token, JWT_SECRET);
        req.user = decoded;
        next();
    } catch (error) {
        res.status(401).json({ message: "Invalid token" });
    }
};
```

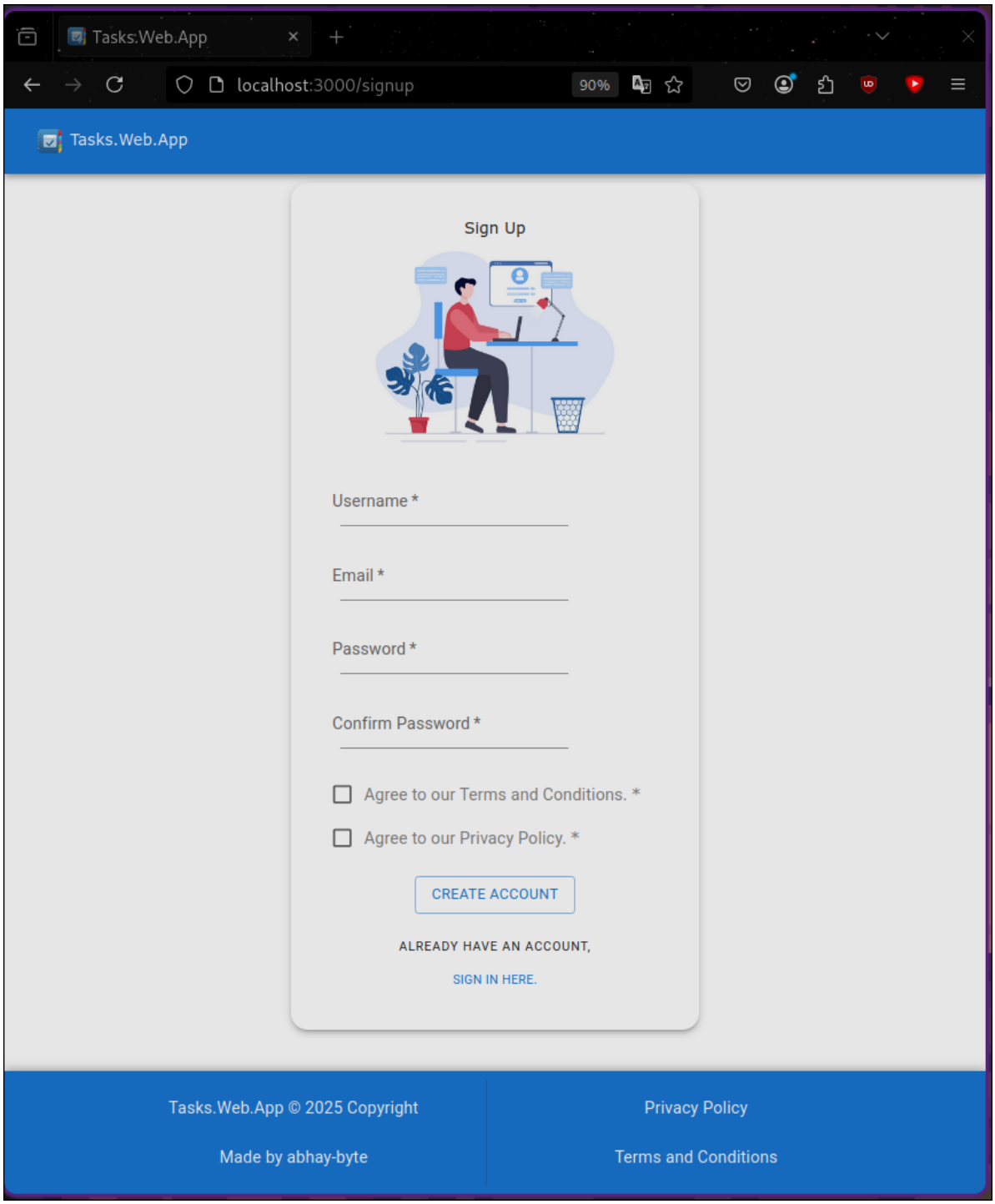
```
// **Protected Route**
app.get("/protected", authenticate, (req, res) => {
    res.json({ message: "Access granted", user: req.user });
});
```

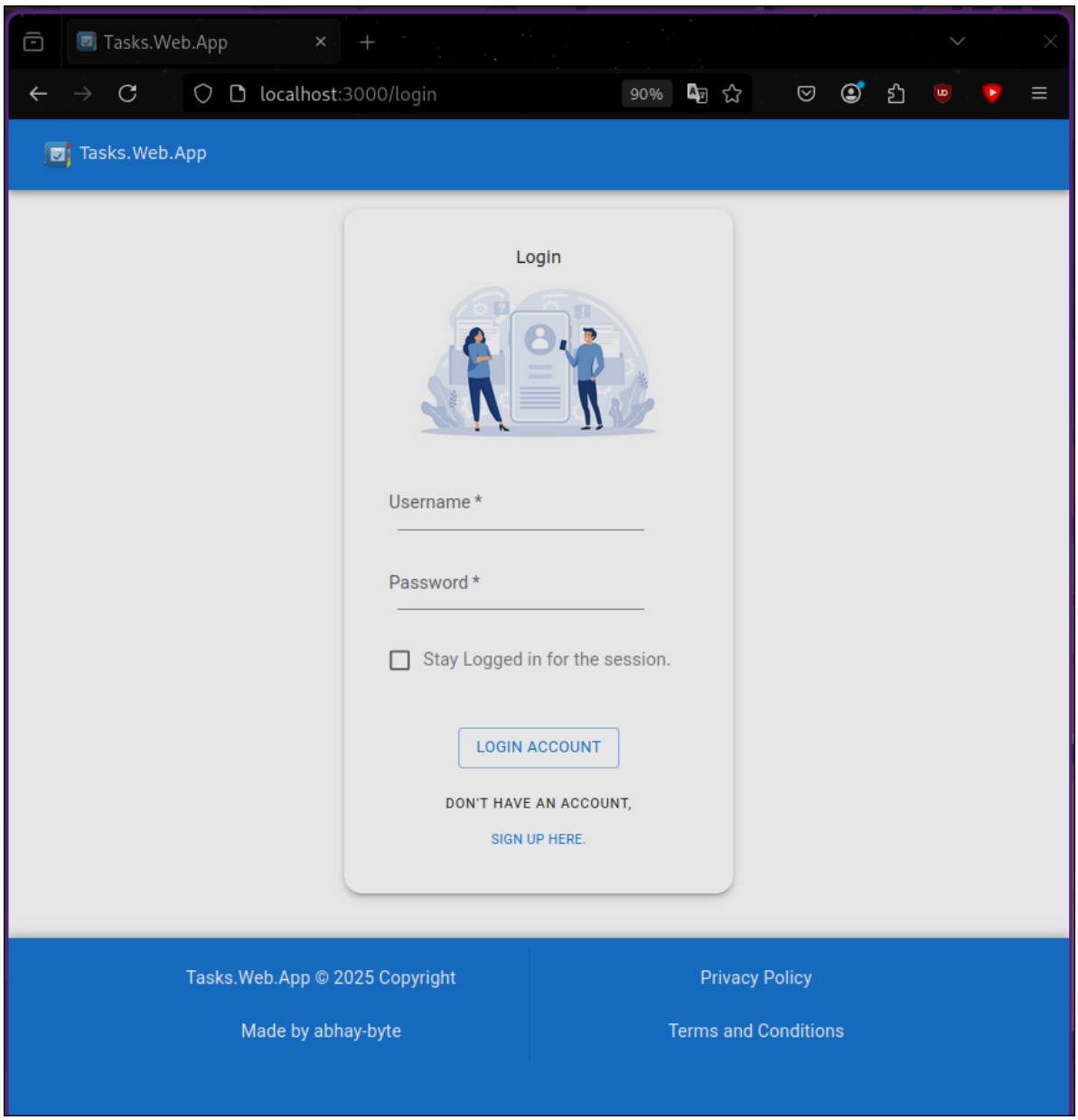
/home/abhay/WebstormProjects/tms_mongo_backend/models/tasks.js

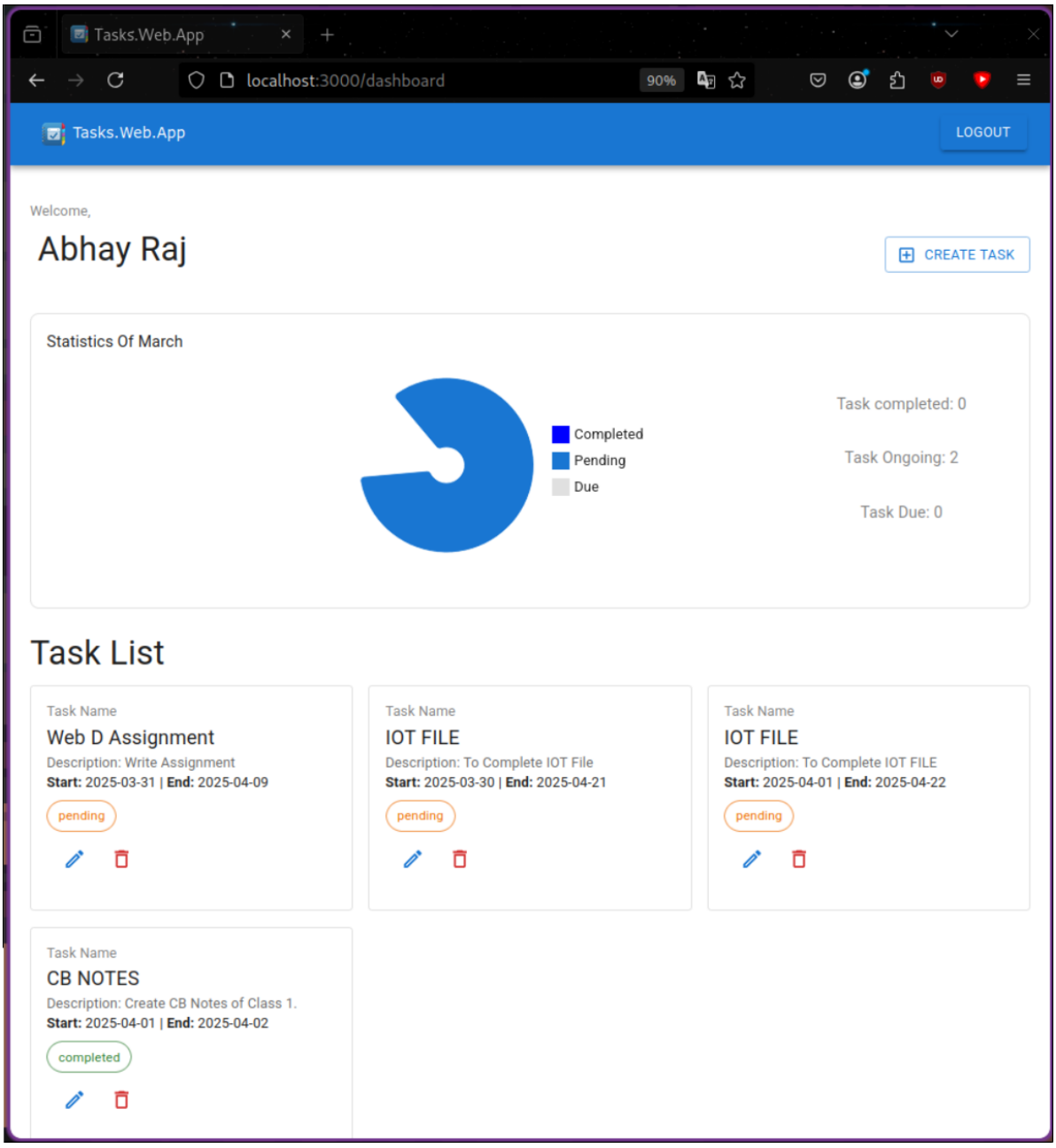
```
import mongoose, {Schema} from 'mongoose';

const taskSchema = new Schema({
    taskname: {
        type: String,
        required: true
    },
    taskdescription: {
        type: String,
        required: true
    },
    startdate: {
        type: String,
        required: true
    },
    enddate: {
        type: String,
        required: true
    },
    status: {
        type: String,
        enum: ['completed', 'pending', 'in-progress'],  // Add
possible statuses here
        required: true
    },
    userid: {
        type: String,  // Reference to the User model
        ref: 'User',
        required: true
    },
    taskid: {
        type: String,
    }
}, { timestamps: true });  // Automatically add createdAt and
updatedAt fields


export default mongoose.model('task', taskSchema);
```

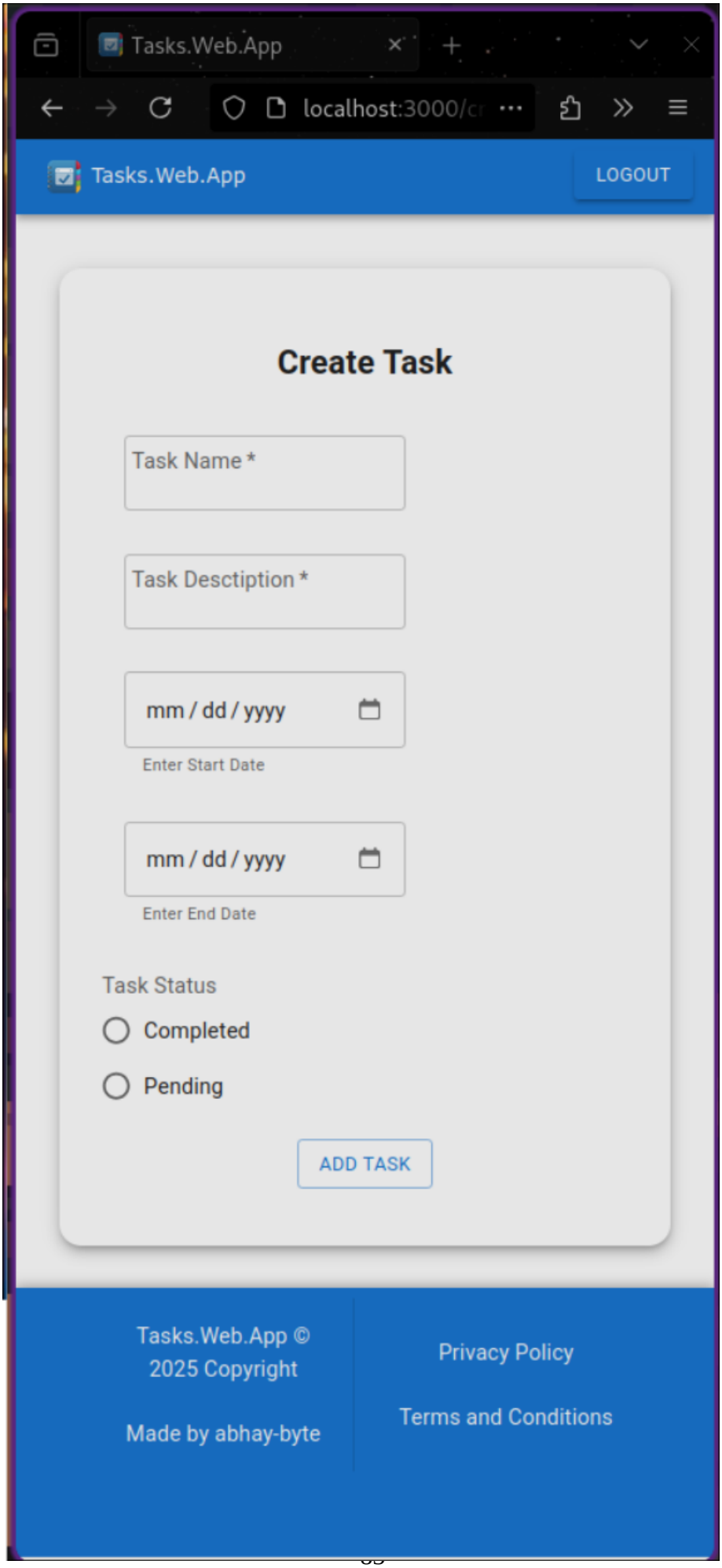
Output:-

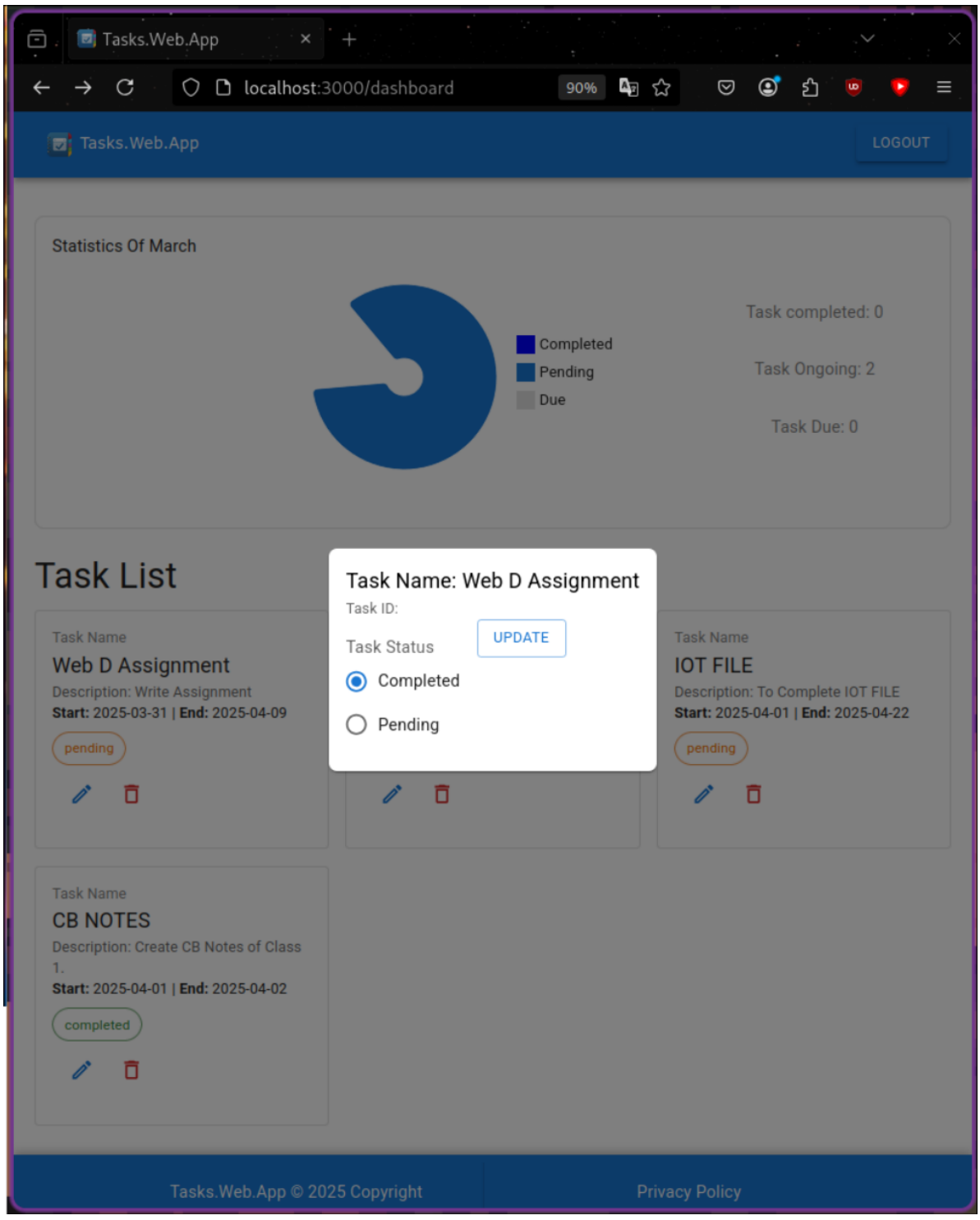

Tasks.Web.App
localhost:3000
90%
Tasks.Web.App
LOGOUT
Welcome to Tasks.Web.App
A place where you can create tasks, manage your task and do it on time.
Sign up, if you are new or login to use.
GO TO DASHBOARD
Tasks.Web.App © 2025 Copyright
Made by abhay-byte
Privacy Policy
Terms and Conditions


Tasks:Web.App
localhost:3000/signup
90%
Tasks.Web.App
Sign Up
Username *
Email *
Password *
Confirm Password *
Agree to our Terms and Conditions. *
Agree to our Privacy Policy. *
CREATE ACCOUNT
ALREADY HAVE AN ACCOUNT,
SIGN IN HERE.
Tasks.Web.App © 2025 Copyright
Made by abhay-byte
Privacy Policy
Terms and Conditions


Tasks.Web.App
localhost:3000/login
90%
Tasks.Web.App
Login
Username *
Password *
Stay Logged in for the session.
LOGIN ACCOUNT
DON'T HAVE AN ACCOUNT,
SIGN UP HERE.
Tasks.Web.App © 2025 Copyright
Made by abhay-byte
Privacy Policy
Terms and Conditions


Tasks.Web.App
localhost:3000/dashboard
90%
Tasks.Web.App
LOGOUT
Welcome,
Abhay Raj
CREATE TASK
Statistics Of March
Completed
Pending
Due
Task completed: 0
Task Ongoing: 2
Task Due: 0
Task List
Task Name
Web D Assignment
Description: Write Assignment
Start: 2025-03-31 | End: 2025-04-09
pending
Task Name
IOT FILE
Description: To Complete IOT File
Start: 2025-03-30 | End: 2025-04-21
pending
Task Name
IOT FILE
Description: To Complete IOT FILE
Start: 2025-04-01 | End: 2025-04-22
pending
Task Name
CB NOTES
Description: Create CB Notes of Class 1.
Start: 2025-04-01 | End: 2025-04-02
completed


Tasks.Web.App
localhost:3000/cr
Tasks.Web.App
LOGOUT
Create Task
Task Name *
Task Desctiption *
mm / dd / yyyy
Enter Start Date
mm / dd / yyyy
Enter End Date
Task Status
Completed
Pending
ADD TASK
Tasks.Web.App © 2025 Copyright
Made by abhay-byte
Privacy Policy
Terms and Conditions


Tasks.Web.App
localhost:3000/dashboard
90%
Tasks.Web.App
LOGOUT
Statistics Of March
Completed
Pending
Due
Task completed: 0
Task Ongoing: 2
Task Due: 0
Task List
Task Name: Web D Assignment
Task ID:
Task Status
UPDATE
Completed
Pending
Task Name
Web D Assignment
Description: Write Assignment
Start: 2025-03-31 | End: 2025-04-09
pending
Task Name
IOT FILE
Description: To Complete IOT FILE
Start: 2025-04-01 | End: 2025-04-22
pending
Task Name
CB NOTES
Description: Create CB Notes of Class 1.
Start: 2025-04-01 | End: 2025-04-02
completed
Tasks.Web.App © 2025 Copyright
Privacy Policy

Experiment 12

Aim:- To create a Blogging platform in MERN.

Code:-

FRONTEND:

```
//p12/frontend/src/pages/Home.js
import React, { useEffect, useState } from "react";
import { Container, Card, CardContent, Typography, Button } from
"@mui/material";
import axios from "axios";

const Home = () => {
const [posts, setPosts] = useState([]);

useEffect(() => {
axios.get("http://localhost:5000/api/posts")
.then(res => setPosts(res.data))
.catch(err => console.log(err));
}, []);

const deletePost = (id) => {
axios.delete(`http://localhost:5000/api/posts/${id}`)
.then(() => setPosts(posts.filter(post => post._id !== id)));
};

return (
<Container>
<h2>Blog Posts</h2>
{posts.map((post) => (
<Card key={post._id} style={{ marginBottom: "10px" }}>
<CardContent>
<Typography variant="h5">{post.title}</Typography>
<Typography variant="subtitle1">By {post.author}</Typography>
<Typography variant="body1">{post.body}</Typography>
<Button color="error" onClick={() => deletePost(post._id)}
>Delete</Button>
</CardContent>
</Card>
))}
</Container>
);
};

export default Home;
```

```
//p12/frontend/src/pages/CreatePost.js
import React, { useState } from "react";
import { Container, TextField, Button } from "@mui/material";
import axios from "axios";
import { useNavigate } from "react-router-dom";

const CreatePost = () => {
const [post, setPost] = useState({ title: "", author: "", body: "" });
const navigate = useNavigate();

const handleChange = (e) => {
setPost({ ...post, [e.target.name]: e.target.value });
};

const submitPost = () => {
axios.post("http://localhost:5000/api/posts", post)
.then(() => navigate("/"))
.catch(err => console.log(err));
};

return (
<Container>
<h2>Create a New Post</h2>
<TextField fullWidth name="title" label="Title"
onChange={handleChange} />
<TextField fullWidth name="author" label="Author"
onChange={handleChange} />
<TextField fullWidth multiline rows={4} name="body" label="Body"
onChange={handleChange} />
<Button variant="contained" color="primary" onClick={submitPost}
>Submit</Button>
</Container>
);
};

export default CreatePost;


//p12/frontend/src/App.js
import { BrowserRouter as Router, Routes, Route, Link } from "react-
router-dom";
import { AppBar, Toolbar, Button } from "@mui/material";
import Home from "./pages/Home";
import CreatePost from "./pages/CreatePost";

function App() {
return (
<Router>
<AppBar position="static">
```

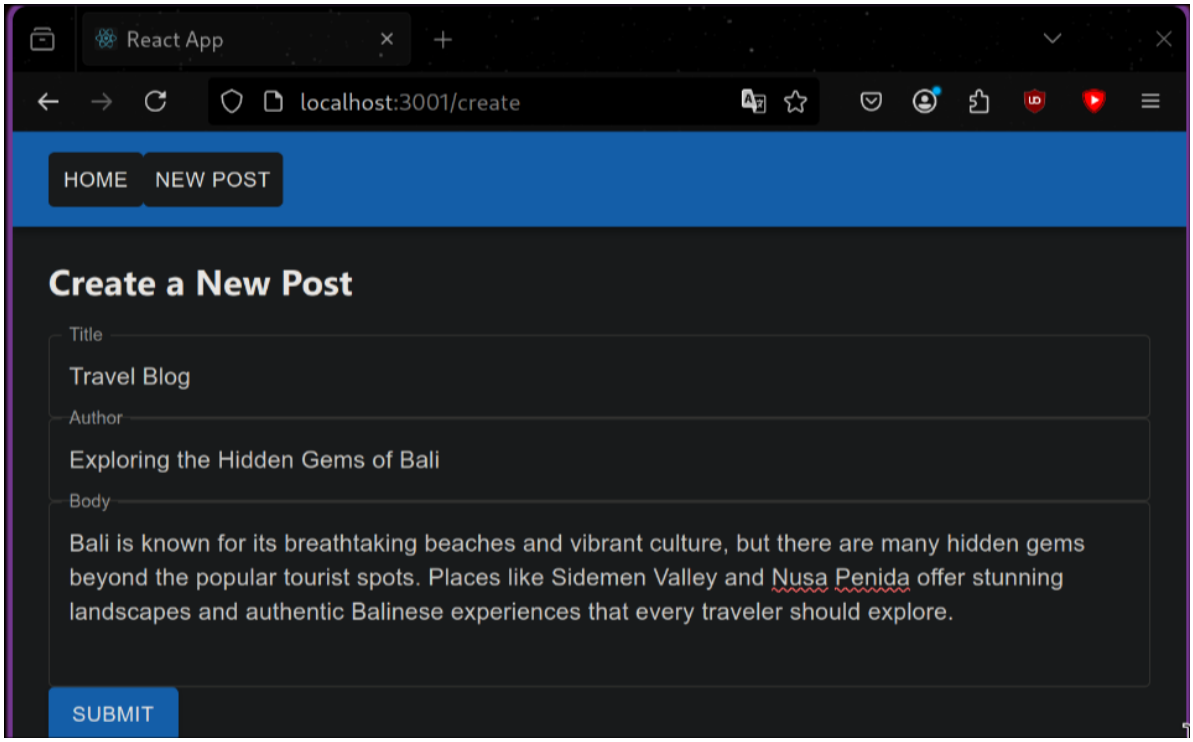
```
<Toolbar>
<Button color="inherit" component={Link} to="/">Home</Button>
<Button color="inherit" component={Link} to="/create">New
Post</Button>
</Toolbar>
</AppBar>
<Routes>
<Route path="/" element={<Home />} />
<Route path="/create" element={<CreatePost />} />
</Routes>
</Router>
);
}

export default App;
```

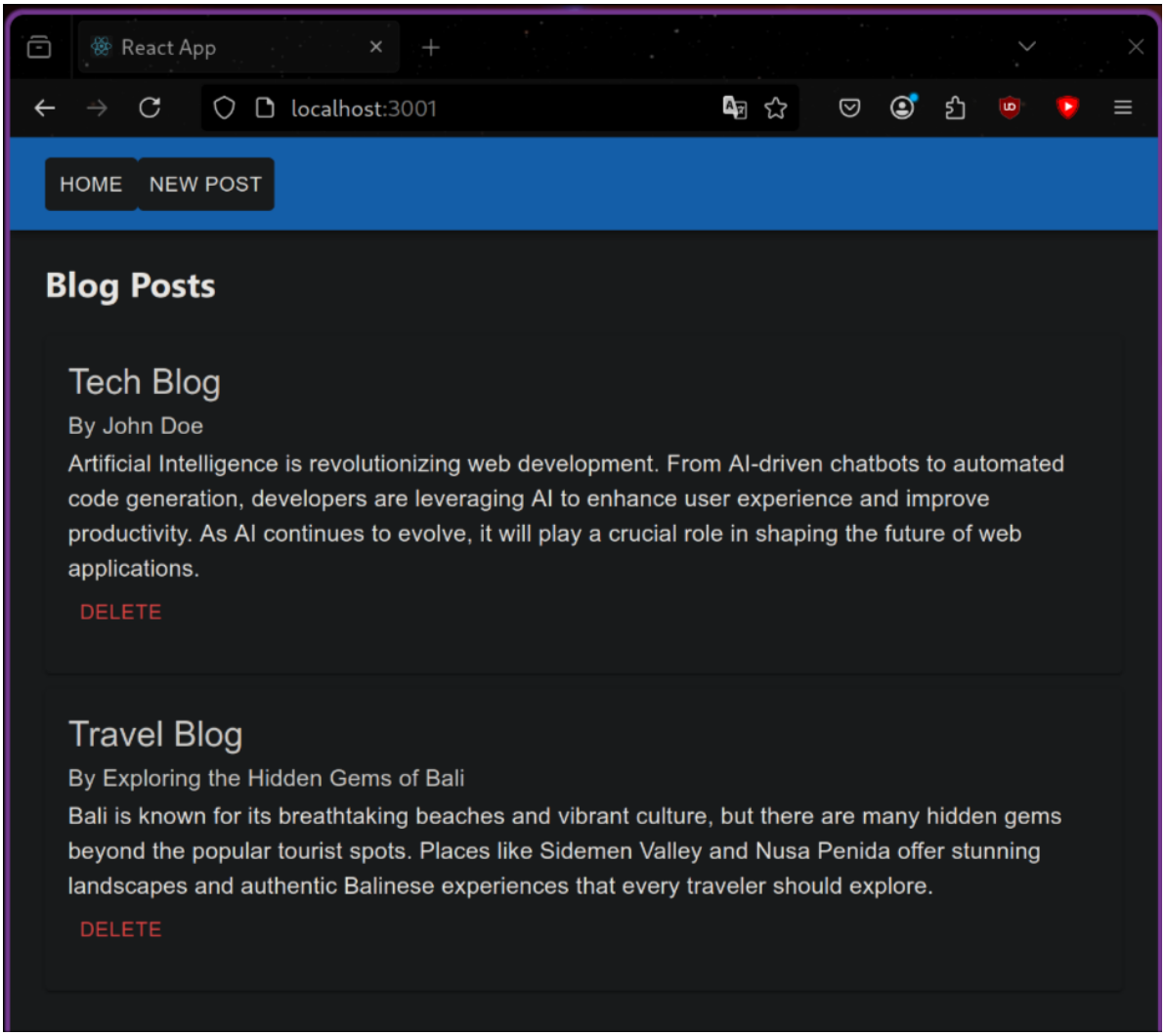
BACKEND:

```
//p12/backend/server.js
const express = require("express");
const mongoose = require("mongoose");
const cors = require("cors");
const dotenv = require("dotenv");

dotenv.config();
const app = express();
app.use(express.json());
app.use(cors());

mongoose.connect(process.env.MONGO_URI, { useNewUrlParser: true,
useUnifiedTopology: true })
.then(() => console.log("MongoDB Connected"))
.catch(err => console.log(err));

const postRoutes = require("./routes/postRoutes");
app.use("/api/posts", postRoutes);

app.listen(5000, () => console.log("Server running on port 5000"));


//p12/backend/routes/postRoutes.js

const express = require("express");
const router = express.Router();
const Post = require("../models/Post");

// Get all posts
router.get("/", async (req, res) => {
```

```
const posts = await Post.find();
res.json(posts);
});

// Create a post
router.post("/", async (req, res) => {
const { title, author, body } = req.body;
const newPost = new Post({ title, author, body });
await newPost.save();
res.json(newPost);
});

// Delete a post
router.delete("/:id", async (req, res) => {
await Post.findByIdAndDelete(req.params.id);
res.json({ message: "Post deleted" });
});

module.exports = router;

//p12/backend/models/Post.js


const mongoose = require("mongoose");

const postSchema = new mongoose.Schema({
title: String,
author: String,
body: String
});

module.exports = mongoose.model("Post", postSchema);
```

Output:-


React App
localhost:3001/create
HOME
NEW POST
Create a New Post
Title
Travel Blog
Author
Exploring the Hidden Gems of Bali
Body
Bali is known for its breathtaking beaches and vibrant culture, but there are many hidden gems beyond the popular tourist spots. Places like Sidemen Valley and Nusa Penida offer stunning landscapes and authentic Balinese experiences that every traveler should explore.
SUBMIT


React App
localhost:3001
HOME
NEW POST
Blog Posts
Tech Blog
By John Doe
Artificial Intelligence is revolutionizing web development. From AI-driven chatbots to automated code generation, developers are leveraging AI to enhance user experience and improve productivity. As AI continues to evolve, it will play a crucial role in shaping the future of web applications.
DELETE
Travel Blog
By Exploring the Hidden Gems of Bali
Bali is known for its breathtaking beaches and vibrant culture, but there are many hidden gems beyond the popular tourist spots. Places like Sidemen Valley and Nusa Penida offer stunning landscapes and authentic Balinese experiences that every traveler should explore.
DELETE

Experiment 13

Aim:- To create a Social media platform in MERN.

Code:-

FRONTEND:

```
//p13/social-media-app/frontend/src/App.js
import { BrowserRouter as Router, Route, Routes } from "react-router-
dom";
import Home from "./pages/Home";
import Post from "./pages/Post";

function App() {
return (
<Router>
<Routes>
<Route path="/" element={<Home />} />
<Route path="/post" element={<Post />} />
</Routes>
</Router>
);
}

export default App;


//p13/social-media-app/frontend/src/pages/Home.js
import { useEffect, useState } from "react";
import axios from "axios";
import { Container, Typography, Card, CardContent, Button } from
"@mui/material";
import { useNavigate } from "react-router-dom";

const Home = () => {
const [posts, setPosts] = useState([]);
const navigate = useNavigate();

useEffect(() => {
axios.get("http://localhost:5000/posts")
.then(res => setPosts(res.data))
.catch(err => console.error(err));
}, []);

return (
<Container>
<Typography variant="h4" sx={{ my: 2 }}>All Posts</Typography>
```

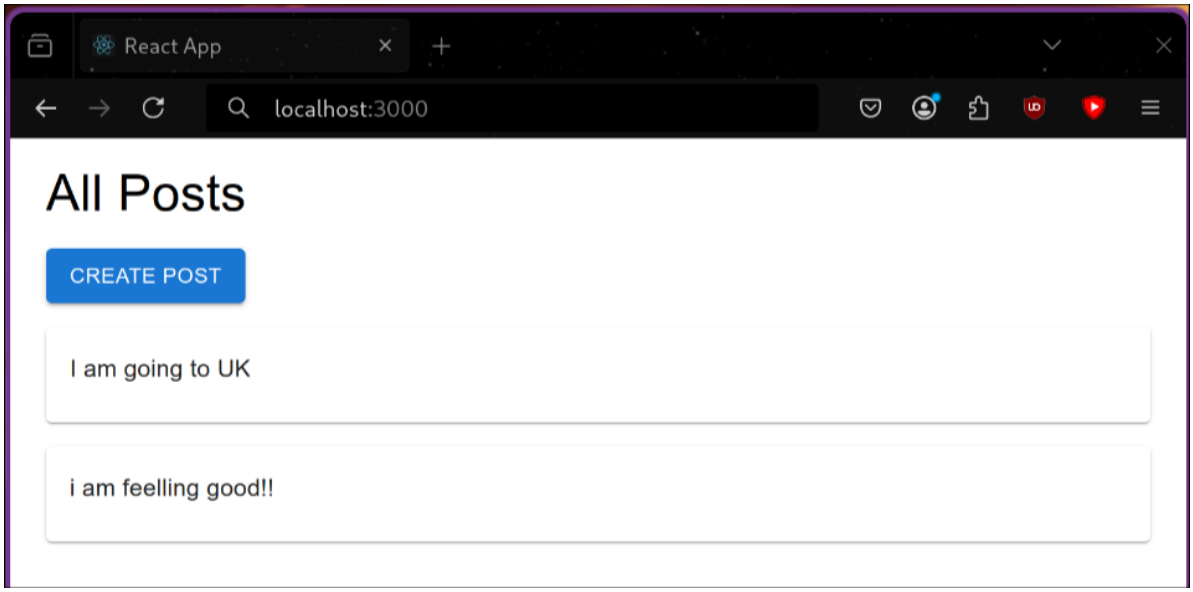
```
<Button variant="contained" onClick={() => navigate("/post")}>Create
Post</Button>
{posts.map(post => (
<Card key={post._id} sx={{ my: 2 }}>
<CardContent>
<Typography>{post.content}</Typography>
</CardContent>
</Card>
))}
</Container>
);
};

export default Home;

//p13/social-media-app/frontend/src/pages/Post.js
import { useState } from "react";
import axios from "axios";
import { Container, TextField, Button, Typography } from
"@mui/material";
import { useNavigate } from "react-router-dom";

const Post = () => {
const [content, setContent] = useState("");
const navigate = useNavigate();

const handleSubmit = async () => {
if (!content) return;
await axios.post("http://localhost:5000/posts", { content });
navigate("/");
};

return (
<Container>
<Typography variant="h4" sx={{ my: 2 }}>Create Post</Typography>
<TextField
fullWidth
label="What's on your mind?"
value={content}
onChange={(e) => setContent(e.target.value)}
sx={{ my: 2 }}
/>
<Button variant="contained" onClick={handleSubmit}>Post</Button>
</Container>
);
};

export default Post;
```

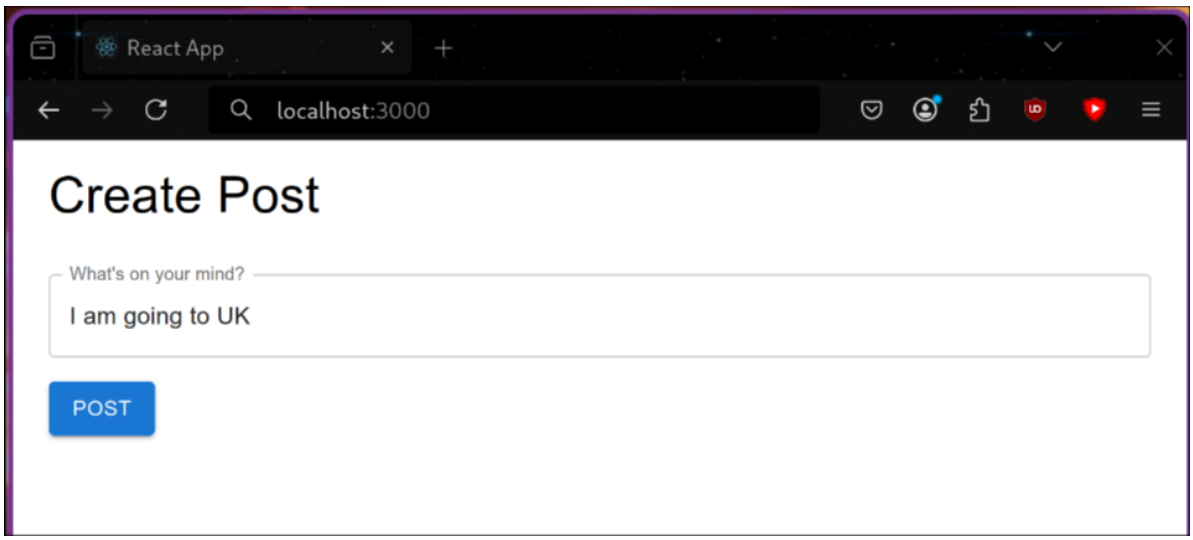
BACKEND:

```
//p13/social-media-app/backend/server.js
require("dotenv").config();
const express = require("express");
const mongoose = require("mongoose");
const cors = require("cors");

const app = express();

// Middleware
app.use(cors());
app.use(express.json());

// MongoDB Connection
mongoose
.connect(process.env.MONGO_URI, { useNewUrlParser: true,
useUnifiedTopology: true })
.then(() => console.log("MongoDB Connected"))
.catch((err) => console.log(err));

// Routes
app.use("/posts", require("./routes/postRoutes"));

// Start Server
const PORT = process.env.PORT || 5000;
app.listen(PORT, () => console.log(`Server running on port ${PORT}`));

//p13/social-media-app/backend/models/Post.js
const mongoose = require("mongoose");

const PostSchema = new mongoose.Schema({
content: { type: String, required: true },
createdAt: { type: Date, default: Date.now },
});

module.exports = mongoose.model("Post", PostSchema);

//p13/social-media-app/backend/routes/postRoutes.js
const express = require("express");
const router = express.Router();
const Post = require("../models/Post");

// Create Post
router.post("/", async (req, res) => {
try {
const newPost = new Post({ content: req.body.content });
```

```
await newPost.save();
res.status(201).json(newPost);
} catch (err) {
res.status(500).json({ error: err.message });
}
});

// Get All Posts
router.get("/", async (req, res) => {
try {
const posts = await Post.find().sort({ createdAt: -1 });
res.json(posts);
} catch (err) {
res.status(500).json({ error: err.message });
}
});

// Delete Post
router.delete("/:id", async (req, res) => {
try {
await Post.findByIdAndDelete(req.params.id);
res.json({ message: "Post deleted" });
} catch (err) {
res.status(500).json({ error: err.message });
}
});

module.exports = router;
```

Output:-


React App
localhost:3000
All Posts
CREATE POST
I am going to UK
i am feelling good!!


React App
localhost:3000
Create Post
What's on your mind?
I am going to UK
POST

Experiment 14

Aim:- To create a Weather Forecasting App.

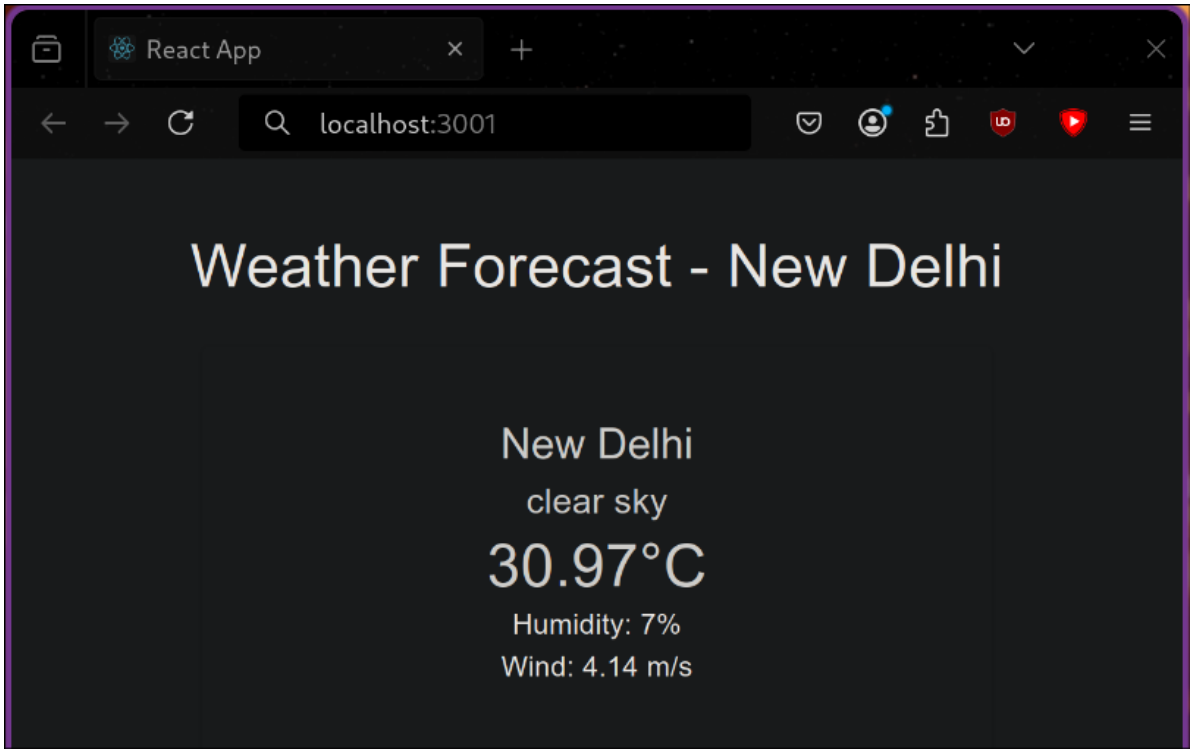
Code:-

FRONTEND:

```
//p14/weather-app/frontend/src/App.js

import Weather from "./components/Weather";

function App() {
return <Weather />;
}

export default App;


//p14/weather-app/frontend/src/components/Weather.js

import { useEffect, useState } from "react";
import axios from "axios";
import { Container, Typography, Card, CardContent, CircularProgress }
from "@mui/material";

const API_KEY = "2984a11f27cc4183f0e09bec287a72be"; // Replace with
your API Key
const CITY = "New Delhi";
const API_URL = `https://api.openweathermap.org/data/2.5/weather?q=$
{CITY}&units=metric&appid=${API_KEY}`;

const Weather = () => {
const [weather, setWeather] = useState(null);
const [loading, setLoading] = useState(true);

useEffect(() => {
axios.get(API_URL)
.then(res => {
setWeather(res.data);
setLoading(false);
})
.catch(err => console.error(err));
}, []);

return (
<Container sx={{ textAlign: "center", mt: 5 }}>
<Typography variant="h4">Weather Forecast - New Delhi</Typography>
{loading ? (
```

```
<CircularProgress sx={{ mt: 3 }} />
) : (
<Card sx={{ mt: 3, p: 3, maxWidth: 400, mx: "auto" }}>
<CardContent>
<Typography variant="h5">{weather.name}</Typography>
<Typography variant="h6">{weather.weather[0].description}</Typography>
<Typography variant="h4">{weather.main.temp}°C</Typography>
<Typography>Humidity: {weather.main.humidity}%</Typography>
<Typography>Wind: {weather.wind.speed} m/s</Typography>
</CardContent>
</Card>
)}
</Container>
);
};

export default Weather;
```

Output:-



Experiment 15

Aim:- Bookstore Library and Stock-Keeping App

Code:-

FRONTEND:

```
//p15/bookstore-app/frontend/src/App.js
import { BrowserRouter as Router, Routes, Route } from "react-router-
dom";
import BookList from "./components/BookList";
import StockManager from "./components/StockManager";
import { Container, AppBar, Toolbar, Typography, Button } from
"@mui/material";
import { Link } from "react-router-dom";

function App() {
return (
<Router>
<Container>
<AppBar position="static">
<Toolbar>
<Typography variant="h6" sx={{ flexGrow: 1 }}>Bookstore</Typography>
<Button color="inherit" component={Link} to="/">Library</Button>
<Button color="inherit" component={Link} to="/stock">Stock
Manager</Button>
</Toolbar>
</AppBar>
<Routes>
<Route path="/" element={<BookList />} />
<Route path="/stock" element={<StockManager />} />
</Routes>
</Container>
</Router>
);
}

export default App;


//p15/bookstore-app/frontend/src/components/BookList.js
import { useEffect, useState } from "react";
import axios from "axios";
import { List, ListItem, ListItemText, Typography, Container } from
"@mui/material";

const BookList = () => {
const [books, setBooks] = useState([]);
```

```
useEffect(() => {
axios.get("http://localhost:5000/api/books")
.then(res => setBooks(res.data))
.catch(err => console.error(err));
}, []);

return (
<Container>
<Typography variant="h4" sx={{ my: 3 }}>Library Books</Typography>
<List>
{books.map((book) => (
<ListItem key={book._id}>
<ListItemText primary={`${book.title} by ${book.author}`}
secondary={`Genre: ${book.genre}, Stock: ${book.stock}`} />
</ListItem>
))}
</List>
</Container>
);
};

export default BookList;


//p15/bookstore-app/frontend/src/components/StockManager.js
import { useState } from "react";
import axios from "axios";
import { TextField, Button, Container, Typography } from
"@mui/material";

const StockManager = () => {
const [book, setBook] = useState({ title: "", author: "", genre: "",
publishedYear: "", stock: "" });

const handleChange = (e) => {
setBook({ ...book, [e.target.name]: e.target.value });
};

const handleSubmit = () => {
axios.post("http://localhost:5000/api/books", book)
.then(() => alert("Book added successfully"))
.catch(err => console.error(err));
};

return (
<Container>
<Typography variant="h4" sx={{ my: 3 }}>Stock Management</Typography>
```

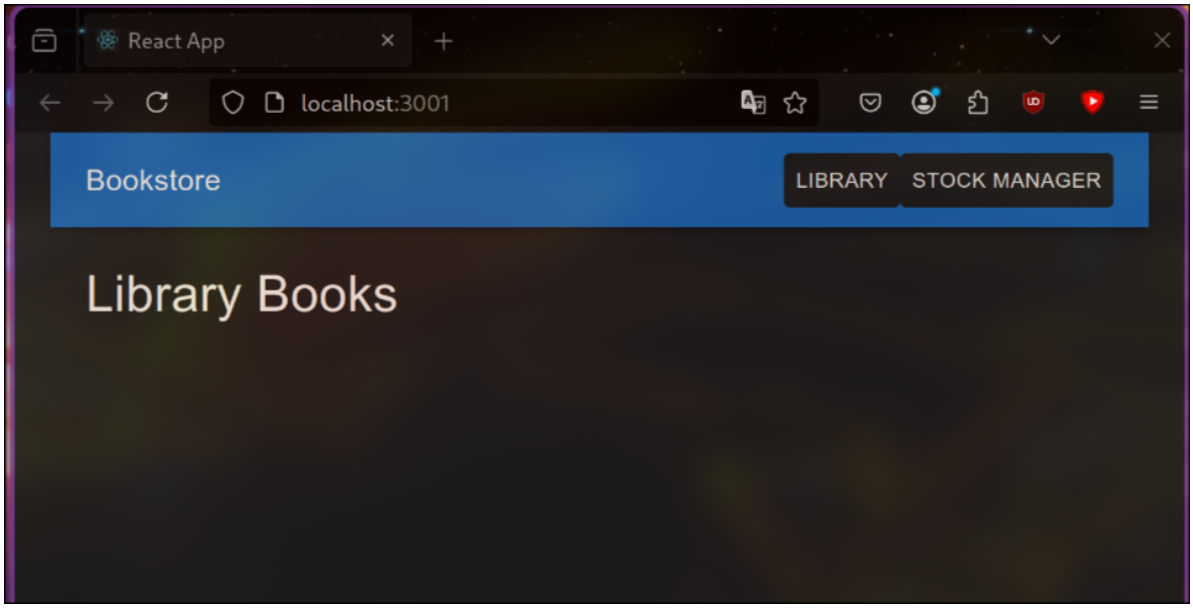
```
<TextField label="Title" name="title" fullWidth margin="normal"
onChange={handleChange} />
<TextField label="Author" name="author" fullWidth margin="normal"
onChange={handleChange} />
<TextField label="Genre" name="genre" fullWidth margin="normal"
onChange={handleChange} />
<TextField label="Published Year" name="publishedYear" fullWidth
margin="normal" onChange={handleChange} />
<TextField label="Stock" name="stock" fullWidth margin="normal"
onChange={handleChange} />
<Button variant="contained" color="primary" sx={{ mt: 2 }}
onClick={handleSubmit}>Add Book</Button>
</Container>
);
};

export default StockManager;
```

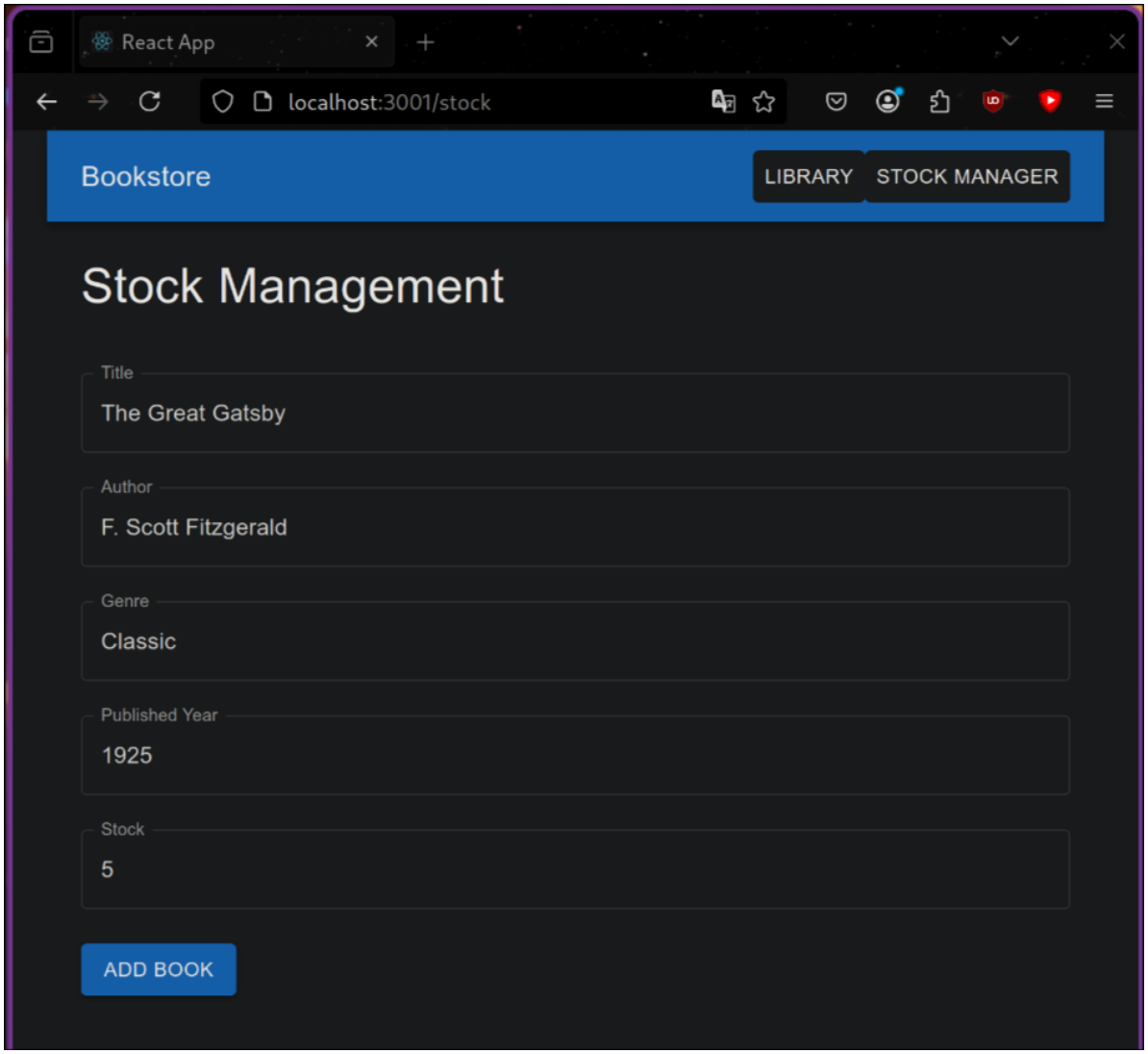
BACKEND:

```
//p15/bookstore-app/backend/server.js
const express = require("express");
const mongoose = require("mongoose");
const cors = require("cors");
require("dotenv").config();

const app = express();
app.use(cors());
app.use(express.json());

// Connect to MongoDB
mongoose.connect(process.env.MONGO_URI, { useNewUrlParser: true,
useUnifiedTopology: true })
.then(() => console.log("MongoDB Connected"))
.catch(err => console.error(err));

// Import book routes
const bookRoutes = require("./routes/bookRoutes");
app.use("/api/books", bookRoutes);

const PORT = process.env.PORT || 5000;
app.listen(PORT, () => console.log(`Server running on port ${PORT}`));
```

```
//p15/bookstore-app/backend/routes/bookRoutes.js
const express = require("express");
const { getBooks, addBook, deleteBook } =
require("../controllers/bookController");

const router = express.Router();

router.get("/", getBooks);
router.post("/", addBook);
router.delete("/:id", deleteBook);

module.exports = router;


///home/abhay/code/WEBD/p15/bookstore-app/backend/models/Book.js
const mongoose = require("mongoose");

const BookSchema = new mongoose.Schema({
title: String,
author: String,
genre: String,
publishedYear: Number,
stock: Number
});

module.exports = mongoose.model("Book", BookSchema);


//p15/bookstore-app/backend/controllers/bookController.js
const Book = require("../models/Book");

// Get all books
exports.getBooks = async (req, res) => {
const books = await Book.find();
res.json(books);
};

// Add a new book
exports.addBook = async (req, res) => {
const newBook = new Book(req.body);
await newBook.save();
res.json({ message: "Book added successfully" });
};

// Delete a book
exports.deleteBook = async (req, res) => {
await Book.findByIdAndDelete(req.params.id);
res.json({ message: "Book deleted" });
};
```

Output:-


React App
localhost:3001
Bookstore
LIBRARY
STOCK MANAGER
Library Books


React App
localhost:3001/stock
Bookstore
LIBRARY
STOCK MANAGER
Stock Management
Title
The Great Gatsby
Author
F. Scott Fitzgerald
Genre
Classic
Published Year
1925
Stock
5
ADD BOOK