Thadomal Shahani Engineering College

Bandra (W.), Mumbai- 400 050.

CERTIFICATE

Certify that Mr./~~Miss~~ Akshay Rathod

of IT Department, Semester V with

Roll No. 99 has completed a course of the necessary

experiments in the subject IP Lab under my

supervision in the **Thadomal Shahani Engineering College**

Laboratory in the year 2023 - 2024

Teacher In-Charge

Head of the Department

Date 24/10/23

Principal

TSEC

CONTENTS

SR. NO.	EXPERIMENTS	PAGE NO.	DATE	TEACHERS SIGN.
1.	Develop a web application by using HTML tags.		21/7/23	
2.	Using CSS and CSS3 enhance the web app developed in Assignment 1		28/7/23	
3.	Develop a webpage using Bootstrap framework		4/8/23	
4.	(a) WAP in JS to study conditional statements, loop & functions		11/8/23	
	(b) WAP on inheritence, iterators and generators.		11/8/23	
5.	(a) WAP to study arrow functions, DOM manipulation & CSS manipulation		18/8/23	
	(b) Write a JavaScript Program (a) Implement Promises (b) Fetch (c) Async JS		18/8/23	
6.	(a) WAP to implement the concept of props and State		25/8/23	
	(b) WAP to implement the concept of forms and events.		25/8/23	
7.	(a) WAP to implement ReactJS Router and events		6/10/23	
	(b) WAP to implement concept of React Hooks.		6/10/23	
8.	Implementation of REPL and Node.js filesystem.		1/9/23	
9.	WAP in React to implement the concept of Refs.		20/10/23	
10.	Create a web app that performs CRUD operations.		20/10/23	

Assignment 1

Aim: To design a webpage using HTML tags.

LO Mapped: LO1

Theory:

1. Code for webpage

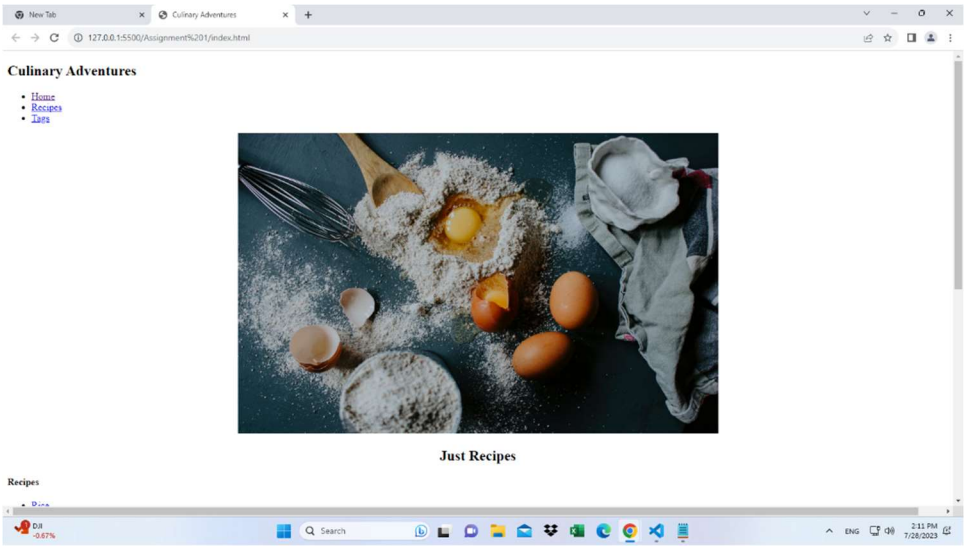
```
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta charset="UTF-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Culinary Adventures</title>
 </head>
 <body>
  <nav>
   <h1>Culinary Adventures</h1>
   <ul>
    <li><a href="index.html">Home</a></li>
    <li><a href="recipes.html">Recipes</a></li>
    <li><a href="tags.html">Tags</a></li>
   </ul>
  </nav>

  <div class="hero">
   <center>
    <img src="assets/main.jpeg" alt="" width="800px" height="500px" />
```

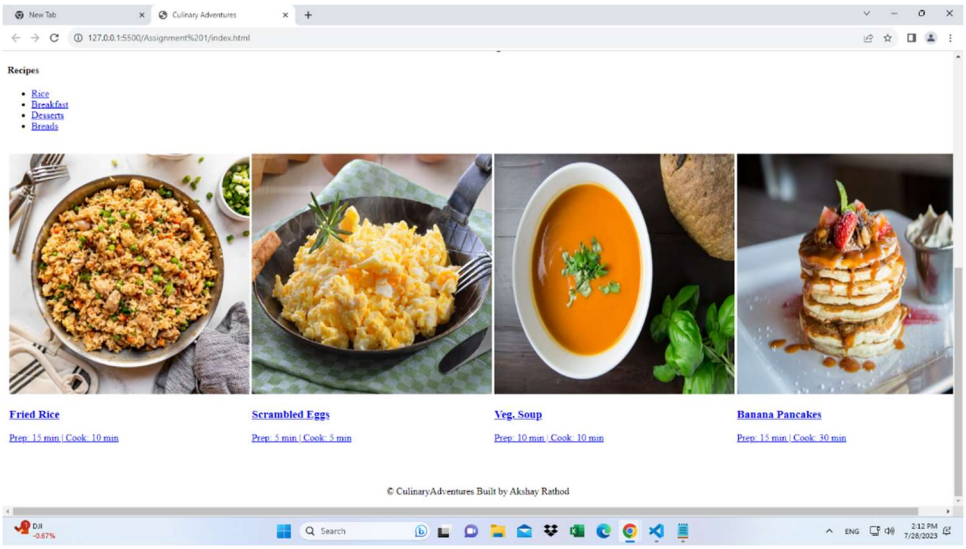
```
   <h2><b>Just Recipes</b></h2>
  </center>
</div>

<section class="recipes-container">
  <div class="tags-container">
    <h4>Recipes</h4>
    <div class="tags-list">
      <ul>
        <li><a href="#">Rice</a></li>
        <li><a href="#">Breakfast</a></li>
        <li><a href="#">Desserts</a></li>
        <li><a href="#">Breads</a></li>
      </ul>
    </div>
    <br />
    <table>
      <tr>
        <td>
          <a href="#">
            <img
              src="assets/recipe-1.jpg"
              alt="Food"
              height="400px"
              width="400px"
            />
            <h3>Fried Rice</h3>
            <p>Prep: 15 min | Cook: 10 min</p>
          </a>
        </td>
```

```
<td>
  <a href="#">
    <img
      src="assets/recipe-2.jpg"
      alt="Food"
      height="400px"
      width="400px"
    />
    <h3>Scrambled Eggs</h3>
    <p>Prep: 5 min | Cook: 5 min</p>
  </a>
</td>
<td>
  <a href="#">
    <img
      src="assets/recipe-3.jpeg"
      alt="Food"
      height="400px"
      width="400px"
    />
    <h3>Veg. Soup</h3>
    <p>Prep: 10 min | Cook: 10 min</p>
  </a>
</td>
<td>
  <a href="#">
    <img
      src="assets/recipe-4.jpeg"
      alt="Food"
      height="400px"
```

```
            width="400px"
           />
           <h3>Banana Pancakes</h3>
           <p>Prep: 15 min | Cook: 30 min</p>
          </a>
         </td>
        </tr>
       </table>
     </div>
   </section>
   <br />
   <br />
   <footer class="page-footer">
    <center>
     <p>
      &copy; <span id="date"></span>
      <span class="footer-logo"> CulinaryAdventures </span>
      Built by Akshay Rathod
     </p>
    </center>
   </footer>
  </body>
</html>
```

2. Screenshots





Conclusion:

In this assignment we understood the basic structure of a webpage and implemented various HTML tags. The HTML tags used in this webpage are as follows:

1.	<nav>	7.	<section>
2.	<h1>, <h4>	8.	<table><tr><td>
3.		9.	
4.	<a>	10.	<footer>
5.	,	11.	
6.	<p>	12.	

Assignment 2

Aim: Use CSS and CSS3 to enhance the web application developed in Assignment #1.

LO Mapped: LO2

Theory:

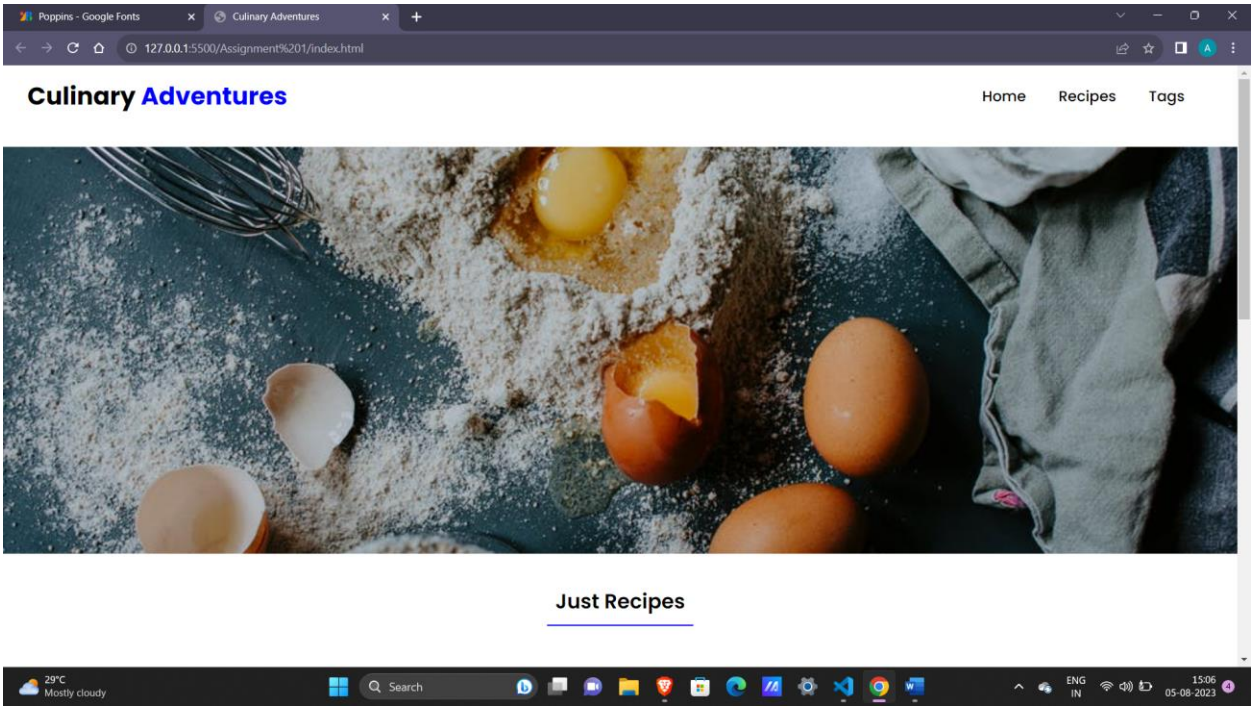
```
@import
url("https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;70
0&display=swap");

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: "Poppins", sans-serif;
}

nav {
  display: flex;
  flex-direction: row;
  justify-content: space-between;
  padding: 15px 30px;
}

nav h1 {
  font-weight: 700;
  font-size: 30px;
}

nav h1 span {
  color: blue;
}

nav > ul {
  display: flex;
  justify-content: space-between;
  align-items: center;
}
```

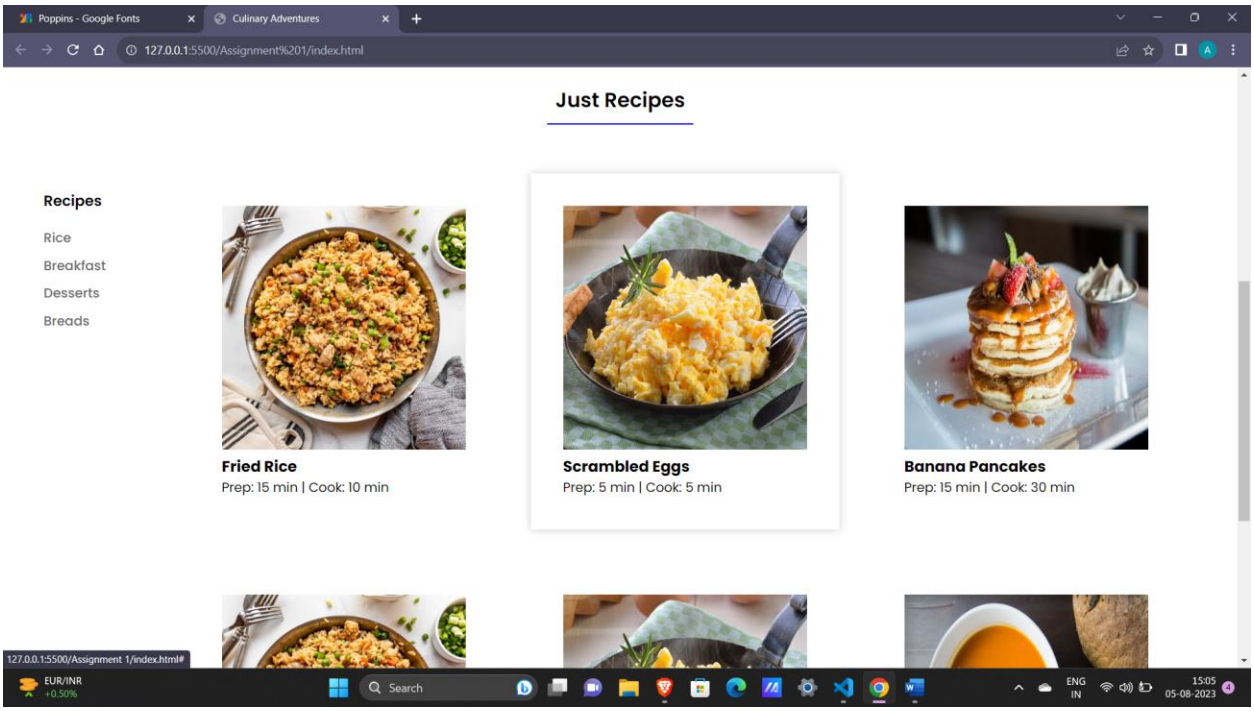
```css
nav > ul > li {
  list-style: none;
  margin-right: 30px;
}

nav > ul > li > a {
  text-decoration: none;
  color: #000;
  font-weight: 500;
  font-size: 18px;
  padding: 10px 5px;
}

nav > ul > li > a:hover {
  border-bottom: 2px solid blue;
}


.hero {
  display: flex;
  flex-direction: column;
  justify-content: space-between;
  align-items: center;
  margin-top: 25px;
}

.hero img {
  width: 100%;
  height: 500px;
  object-fit: cover;
}

.hero h2 {
  font-weight: 600;
  font-size: 25px;
  margin: 30px 0px;
  padding: 10px;
  border-bottom: 2px solid blue;
}

.recipes-container {
  display: flex;
  flex-direction: row;
  margin: 10px 50px;
```

```
  justify-content: space-between;
}

.tags-container {
  display: flex;
  flex-direction: column;
}

.tags-container h4 {
  font-weight: 600;
  font-size: 18px;
  margin-bottom: 20px;
  margin-top: 40px;
}

.tags-list ul {
  list-style: none;
}

.tags-list ul li {
  margin-bottom: 10px;
}

.tags-list ul li a {
  text-decoration: none;
  color: dimgray;
  font-weight: 500;
  font-size: 16px;
}

.recipes {
  display: flex;
  flex-direction: column;
}

.recipes-list {
  display: flex;
  flex-direction: row;
  justify-content: space-around;
}

.recipes-list ul {
  list-style: none;
}
```

```css
.recipes-list a {
  text-decoration: none;
  color: #000;
  padding: 40px;
  margin: 20px;
}

.recipes-list a:hover {
  box-shadow: 0px 0px 10px 0px rgba(0, 0, 0, 0.2);
  transition: 0.2s ease all;
}
```

2. Screenshots





Conclusion:

In this assignment we understood the basic structure of a webpage and implemented various CSS to the HTML page created in Assignment 1. The CSS used in this webpage is as follows:

1.	color	7.	links
2.	background color	8.	Type selector
3.	font-style	9.	Class selector
4.	font-size	10.	Child selector
5.	list-style	11.	Descendant selector
6.	Universal selector	12.	Pseudo selector

Assignment 3

Aim: Develop a webpage using the Bootstrap framework.

LO Mapped: LO3

Theory:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.mi
n.css"
      rel="stylesheet"
      integrity="sha384-
4bw+/aepP/YC94hEpVNVgiZdgIC5+VKNBQNGCHeKRQN+PtmoHDEXuppvnDJzQIu9"
      crossorigin="anonymous"
    />
    <link rel="stylesheet" href="style.css" />
    <title>Bootstrap</title>
  </head>
  <body>
    <nav class="navbar bg-dark navbar-expand-lg bg-body-tertiary">
      <div class="container-fluid">
        <a class="navbar-brand" href="#">Tekkerz</a>
        <button
          class="navbar-toggler"
          type="button"
          data-bs-toggle="collapse"
          data-bs-target="#navbarScroll"
          aria-controls="navbarScroll"
          aria-expanded="false"
          aria-label="Toggle navigation"
        >
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarScroll">
          <ul
            class="navbar-nav me-auto my-2 my-lg-0 navbar-nav-scroll"
            style="--bs-scroll-height: 100px"
```

```
        >
          <li class="nav-item">
            <a class="nav-link active" aria-current="page" href="#">Home</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="#">Store</a>
          </li>
          <li class="nav-item dropdown">
            <a
              class="nav-link dropdown-toggle"
              href="#"
              role="button"
              data-bs-toggle="dropdown"
              aria-expanded="false"
            >
              Watch Now
            </a>
            <ul class="dropdown-menu">
              <li><a class="dropdown-item" href="#">Premier League</a></li>
              <li><a class="dropdown-item" href="#">LaLiga</a></li>
              <li><a class="dropdown-item" href="#">Bundensliga</a></li>
              <li><a class="dropdown-item" href="#">Serie A</a></li>
              <li><a class="dropdown-item" href="#">Ligue 1</a></li>
              <li><hr class="dropdown-divider" /></li>
              <li>
                <a class="dropdown-item" href="#">Champions League</a>
              </li>
              <li>
                <a class="dropdown-item" href="#">Europa League</a>
              </li>
              <li>
                <a class="dropdown-item" href="#">Conference League</a>
              </li>
            </ul>
          </li>
        </ul>
        <form class="d-flex" role="search">
          <input
            class="form-control me-2"
            type="search"
            placeholder="Search"
            aria-label="Search"
          />
          <button class="btn btn-outline-success" type="submit">
            Search
          </button>
        </form>
      </div>
```

```
    </div>
  </nav>

  <div id="carouselExampleCaptions" class="carousel slide w">
    <div class="carousel-indicators">
      <button
        type="button"
        data-bs-target="#carouselExampleCaptions"
        data-bs-slide-to="0"
        class="active"
        aria-current="true"
        aria-label="Slide 1"
      ></button>
      <button
        type="button"
        data-bs-target="#carouselExampleCaptions"
        data-bs-slide-to="1"
        aria-label="Slide 2"
      ></button>
      <button
        type="button"
        data-bs-target="#carouselExampleCaptions"
        data-bs-slide-to="2"
        aria-label="Slide 3"
      ></button>
    </div>
    <div class="carousel-inner">
      <div class="carousel-item active">
        <img
          src="assets/prem-caro.png"
          class="d-block w-100"
          alt="..."
          height="720px"
        />
        <div class="carousel-caption d-none d-md-block">
          <button class="btn btn-outline-light w-25 mb-5" type="button">
            Watch Now
          </button>
          <h5>Premier League</h5>
        </div>
      </div>
      <div class="carousel-item">
        <img
          src="assets/champs-caro.jpg"
          class="d-block w-100"
          alt="..."
          height="720px"
        />
```

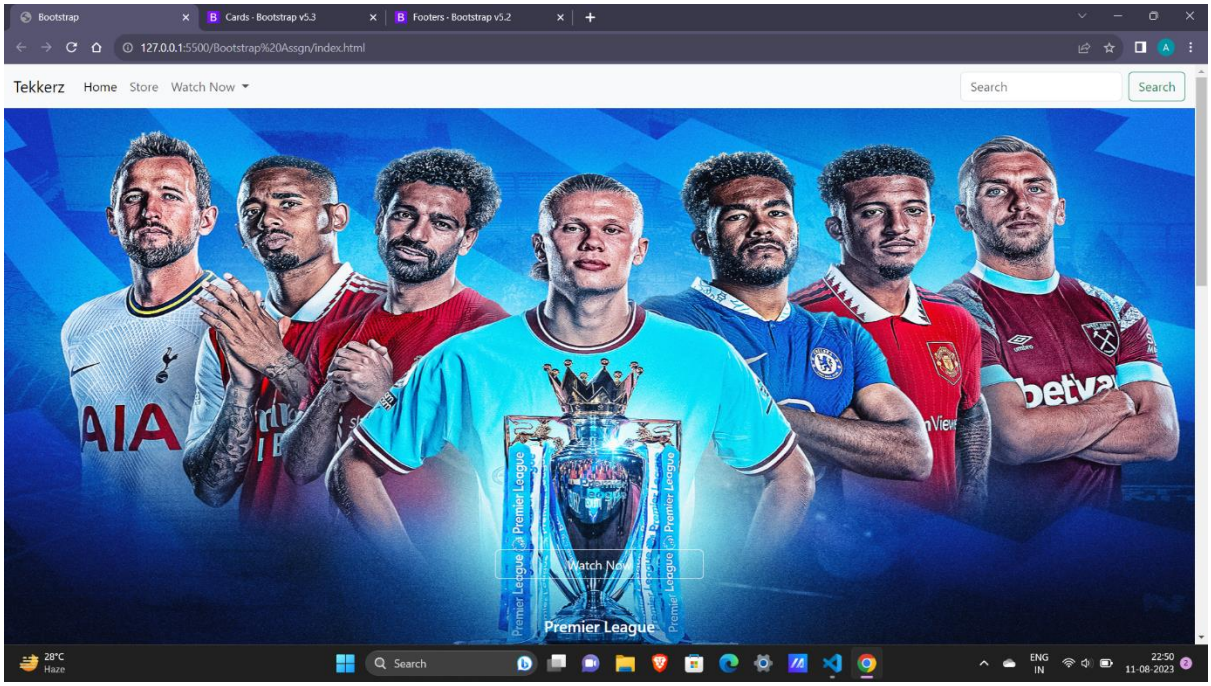
```
          <div class="carousel-caption d-none d-md-block">
            <button class="btn btn-outline-light w-25 mb-5" type="button">
              Watch Now
            </button>
            <h5>Champions League</h5>
          </div>
        </div>
        <div class="carousel-item">
          <img
            src="assets/laliga-caro.jpg"
            class="d-block w-100"
            alt="..."
            height="720px"
          />
          <div class="carousel-caption d-none d-md-block">
            <button class="btn btn-outline-light w-25 mb-5" type="button">
              Watch Now
            </button>
            <h5>La Liga</h5>
          </div>
        </div>
      </div>
      <button
        class="carousel-control-prev"
        type="button"
        data-bs-target="#carouselExampleCaptions"
        data-bs-slide="prev"
      >
        <span class="carousel-control-prev-icon" aria-hidden="true"></span>
        <span class="visually-hidden">Previous</span>
      </button>
      <button
        class="carousel-control-next"
        type="button"
        data-bs-target="#carouselExampleCaptions"
        data-bs-slide="next"
      >
        <span class="carousel-control-next-icon" aria-hidden="true"></span>
        <span class="visually-hidden">Next</span>
      </button>
    </div>
    <div class="store">
      <h1>Visit Our Store</h1>

      <div class="row row-cols-1 row-cols-md-3 g-4 my-5 mx-5">
        <div class="col">
          <div class="card h-100">
            <img src="assets/nike-shoe.jpeg" class="card-img-top" alt="..." />
```

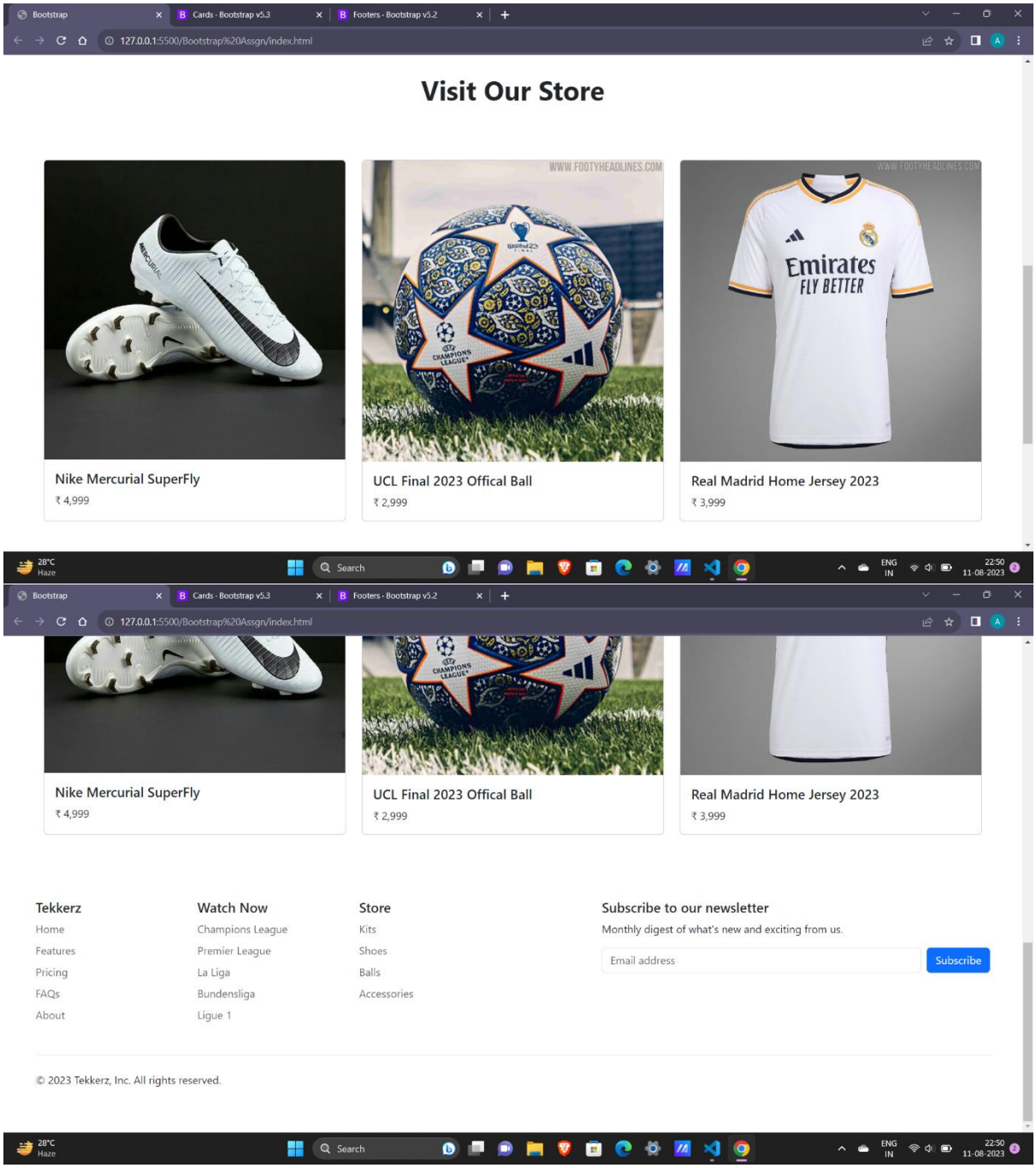
```
            <div class="card-body">
              <h5 class="card-title">Nike Mercurial SuperFly</h5>
              <p class="card-text">
                ₹ 4,999
              </p>
            </div>
          </div>
        </div>
        <div class="col">
          <div class="card h-100">
            <img src="assets/champs-ball.jpg" class="card-img-top" alt="..."
/>
            <div class="card-body">
              <h5 class="card-title">UCL Final 2023 Offical Ball</h5>
              <p class="card-text">₹ 2,999</p>
            </div>
          </div>
        </div>
        <div class="col">
          <div class="card h-100">
            <img src="assets/real-jersey.jpg" class="card-img-top" alt="..."
/>
            <div class="card-body">
              <h5 class="card-title">Real Madrid Home Jersey 2023</h5>
              <p class="card-text">
                ₹ 3,999
              </p>
            </div>
          </div>
        </div>
      </div>
    </div>

    <footer class="pt-5 px-5">
      <div class="row">
        <div class="col-6 col-md-2 mb-3">
          <h5>Tekkerz</h5>
          <ul class="nav flex-column">
            <li class="nav-item mb-2"><a href="#" class="nav-link p-0 text-
muted">Home</a></li>
            <li class="nav-item mb-2"><a href="#" class="nav-link p-0 text-
muted">Features</a></li>
            <li class="nav-item mb-2"><a href="#" class="nav-link p-0 text-
muted">Pricing</a></li>
            <li class="nav-item mb-2"><a href="#" class="nav-link p-0 text-
muted">FAQs</a></li>
            <li class="nav-item mb-2"><a href="#" class="nav-link p-0 text-
muted">About</a></li>
```

```
          </ul>
        </div>

        <div class="col-6 col-md-2 mb-3">
          <h5>Watch Now</h5>
          <ul class="nav flex-column">
            <li class="nav-item mb-2"><a href="#" class="nav-link p-0 text-
muted">Champions League</a></li>
            <li class="nav-item mb-2"><a href="#" class="nav-link p-0 text-
muted">Premier League</a></li>
            <li class="nav-item mb-2"><a href="#" class="nav-link p-0 text-
muted">La Liga</a></li>
            <li class="nav-item mb-2"><a href="#" class="nav-link p-0 text-
muted">Bundensliga</a></li>
            <li class="nav-item mb-2"><a href="#" class="nav-link p-0 text-
muted">Ligue 1</a></li>
          </ul>
        </div>

        <div class="col-6 col-md-2 mb-3">
          <h5>Store</h5>
          <ul class="nav flex-column">
            <li class="nav-item mb-2"><a href="#" class="nav-link p-0 text-
muted">Kits</a></li>
            <li class="nav-item mb-2"><a href="#" class="nav-link p-0 text-
muted">Shoes</a></li>
            <li class="nav-item mb-2"><a href="#" class="nav-link p-0 text-
muted">Balls</a></li>
            <li class="nav-item mb-2"><a href="#" class="nav-link p-0 text-
muted">Accessories</a></li>
          </ul>
        </div>

        <div class="col-md-5 offset-md-1 mb-3">
          <form>
            <h5>Subscribe to our newsletter</h5>
            <p>Monthly digest of what's new and exciting from us.</p>
            <div class="d-flex flex-column flex-sm-row w-100 gap-2">
              <label for="newsletter1" class="visually-hidden">Email
address</label>
              <input id="newsletter1" type="text" class="form-control"
placeholder="Email address">
              <button class="btn btn-primary" type="button">Subscribe</button>
            </div>
          </form>
        </div>
      </div>
```

```
      <div class="d-flex flex-column flex-sm-row justify-content-between py-4
my-4 border-top">
        <p>© 2023 Tekkerz, Inc. All rights reserved.</p>
        <ul class="list-unstyled d-flex">
          <li class="ms-3"><a class="link-dark" href="#"><svg class="bi"
width="24" height="24"><use xlink:href="#twitter"></use></svg></a></li>
          <li class="ms-3"><a class="link-dark" href="#"><svg class="bi"
width="24" height="24"><use xlink:href="#instagram"></use></svg></a></li>
          <li class="ms-3"><a class="link-dark" href="#"><svg class="bi"
width="24" height="24"><use xlink:href="#facebook"></use></svg></a></li>
        </ul>
      </div>
    </footer>

    <script
      src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/js/bootstrap.bund
le.min.js"
      integrity="sha384-
HwwvtgBNo3bZJJLYd8oVXjrBZt8cqVSpeBNS5n7C8IVInixGAoxmnlMuBnhbgrkm"
      crossorigin="anonymous"
    ></script>
  </body>
</html>
```

2. Screenshots





Conclusion:

In this assignment we understood and implemented various components of the bootstrap framework which include navbar, carousel, cards, buttons and footer.

Assignment 4(a)

Aim: WAP in JS to study conditional statements, loops and functions.

LO Mapped: LO4

Theory:

```
const checkPalindrome = (string) => {
  const len = string.length;

  for (let i = 0; i < len / 2; i++) {
    if (string[i] !== string[len - 1 - i]) {
      return `"${string}" is not a palindrome`;
    }
  }
  return `"${string}" is a palindrome`;
};

const string = "racecar";
const value = checkPalindrome(string);
console.log(value);
```

2. Screenshots

```
PS D:\Code\Web Dev\Sem-5-IP> node "d:\Code\Web Dev\Sem-5-IP\JavaScript\4a-condition.js"
"racecar" is a palindrome
```

Conclusion:

In this assignment, we understood the concept of conditional statements, loops and functions in JavaScript by writing a program to check if a string is a palindrome.

Assignment 4(b)

Aim: WAP on inheritance, iterators and generators.

LO Mapped: LO4

Theory:

Program

1. Inheritence

```
class Vehicle {
  constructor() {
    this.company = "Honda";
  }
}
class Bike extends Vehicle {
  constructor(name, price) {
    super();
    this.name = name;
    this.price = price;
  }
}
class Car extends Vehicle {
    constructor(name, price) {
      super();
      this.name = name;
      this.price = price;
    }
  }
const bike = new Bike("Rebel", "70000");
const car = new Car("City", "1000000");
console.log(bike.company + " " + bike.name + " " + bike.price);
console.log(car.company + " " + car.name + " " + car.price);
```

```
PS D:\Code\Web Dev\Sem-5-IP> node "d:\Code\Web Dev\Sem-5-IP\JavaScript\4b-inheritence.js"
Honda Rebel 70000
Honda City 1000000
```

2. Iterators

```
const courses = ["JavaScript", "ReactJS", "NodeJS"];

for (let item in courses) {
  console.log(item);
}

for (let item of courses) {
  console.log(item);
}

const nums = [1, 2, 3, 4, 5, 6, 7, 8, 9];
const copy = [];
nums.forEach((element) => {
  copy.push(element * element);
});

console.log(copy);

const array = ["a", "b", "c"];
const it = array[Symbol.iterator]();
console.log(it.next());
console.log(it.next());
console.log(it.next());
console.log(it.next());
```

```
PS D:\Code\Web Dev\Sem-5-IP> node "d:\Code\Web Dev\Sem-5-IP\JavaScript\4b-iterators.js"
0
1
2
JavaScript
ReactJS
NodeJS
[
   1,  4,  9, 16, 25,
  36, 49, 64, 81
]
{ value: 'a', done: false }
{ value: 'b', done: false }
{ value: 'c', done: false }
{ value: undefined, done: true }
```

3. Generator

```
function* genFunction() {
  console.log("First");
  yield 10;
  console.log("Second");
  yield 20;
  console.log("Third");
  yield 30;
}

let gen = genFunction();
console.log(gen.next().value);
console.log(gen.next().value);
console.log(gen.next().value);
```

```
PS D:\Code\Web Dev\Sem-5-IP> node "d:\Code\Web Dev\Sem-5-IP\JavaScript\4b-generator.js"
First
10
Second
20
Third
30
```

Conclusion:

In this assignment, we understood and implemented various programs on inheritance, iterators and generator in JavaScript.

Assignment 5(a)

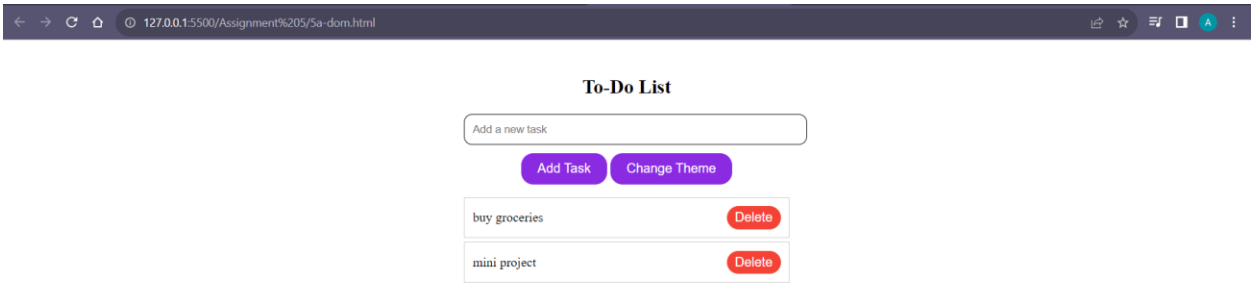
Aim: Write a JavaScript program to study arrow functions, DOM manipulation and CSS manipulation.

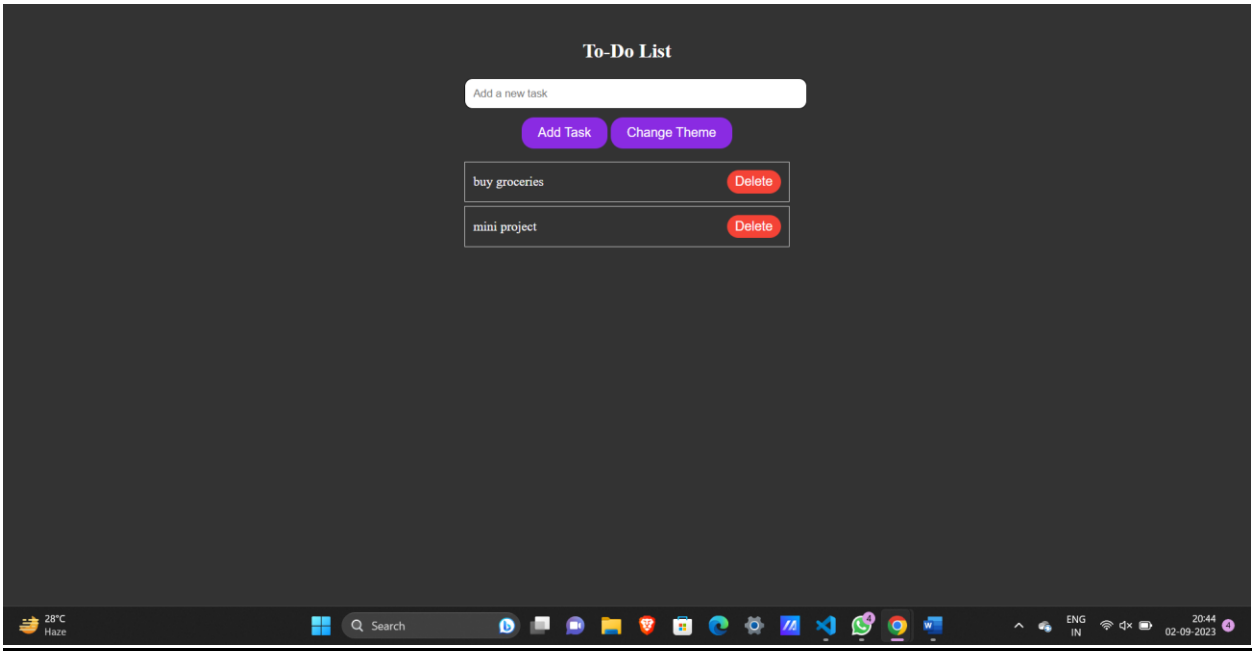
Lo Mapped: LO4

Theory:

```
const taskInput = document.querySelector("#taskInput");
const addButton = document.querySelector("#addButton");
const taskList = document.querySelector("#taskList");
const changeTheme = document.getElementById("change-theme");

addButton.addEventListener("click", () => {
  const taskText = taskInput.value.trim();

  if (taskText === "") {
    alert("Please enter a task.");
    return;
  }

  const listItem = document.createElement("li");
  listItem.innerHTML = `
        ${taskText}
        <button class="delete-button">Delete</button>
    `;

  taskList.appendChild(listItem);
  taskInput.value = "";

  const deleteButton = listItem.querySelector(".delete-button");
  deleteButton.addEventListener("click", () => {
    listItem.remove();
  });
});

changeTheme.addEventListener("click", () => {
  document.body.classList.toggle("dark");
});
```

Screenshots





Conclusion:

In this assignment, we understood and implemented arrow functions, DOM manipulation and CSS manipulation in JavaScript.

Assignment 5(a)

Aim: Write a JavaScript program (a) Implement concept of Promises (b) Fetch (Client-Server communication) and (c) Asynchronous JavaScript.

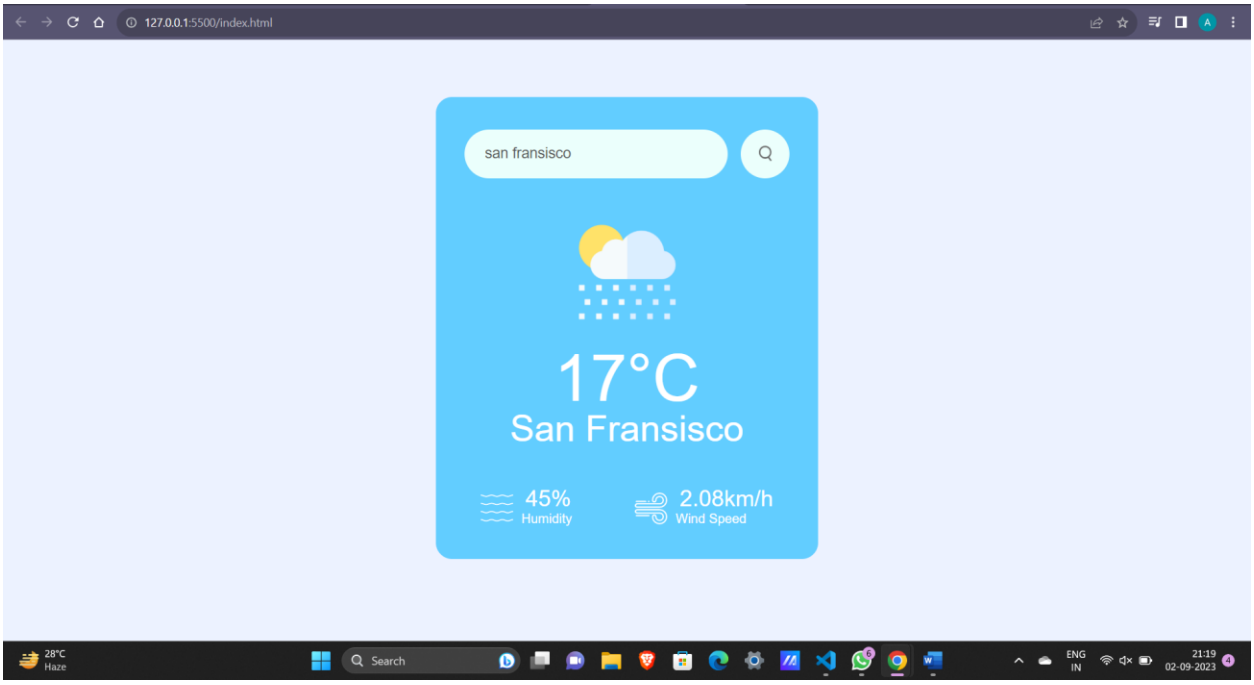
Lo Mapped: LO4

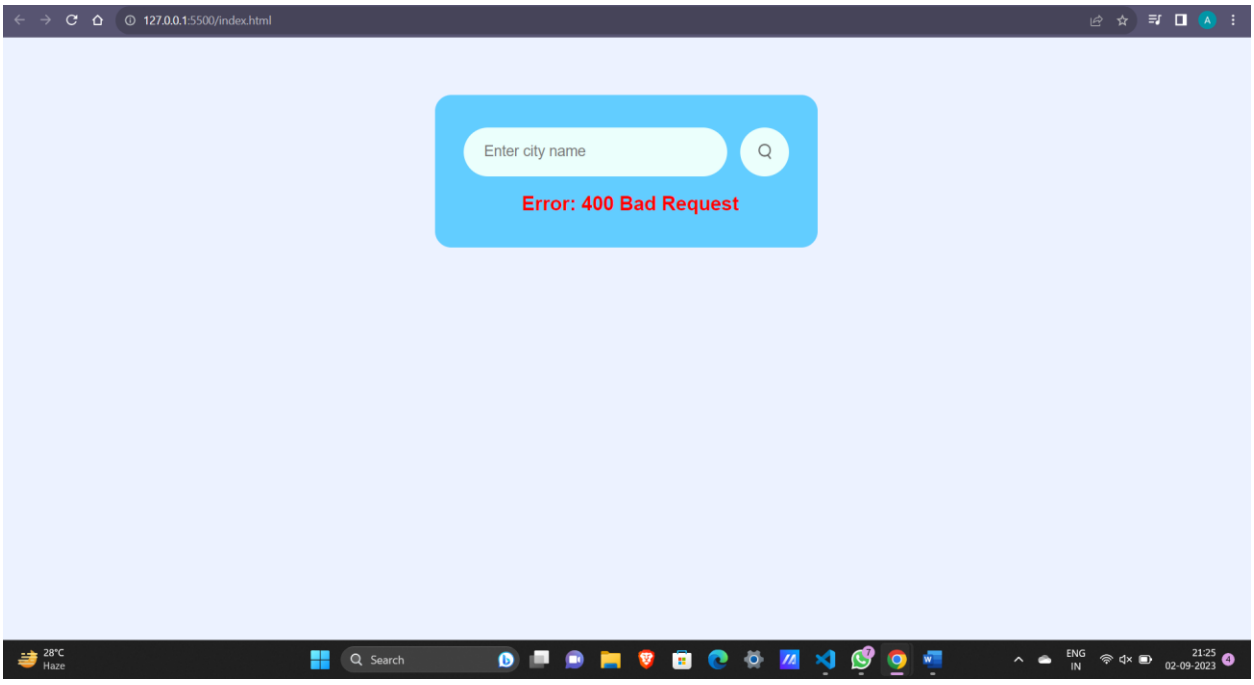
Theory:

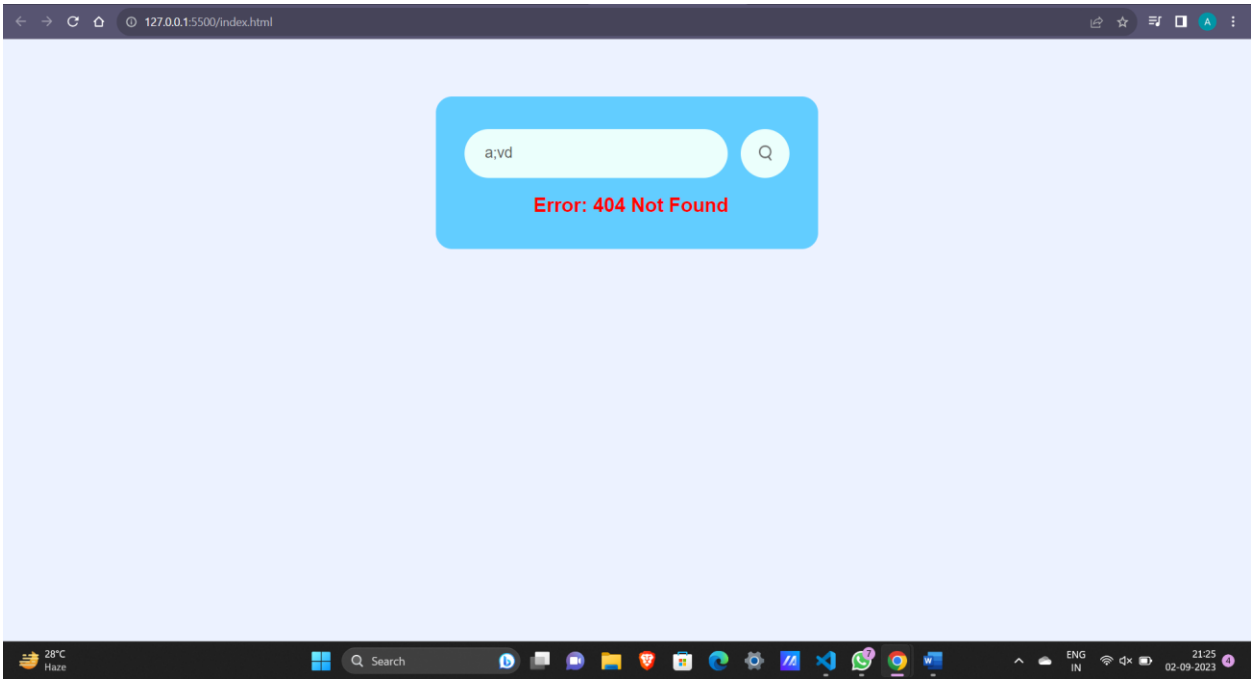
```
const apiKey = "";
const apiUrl = "";
const searchBox = document.querySelector(".search input");
const searchBtn = document.querySelector(".search button");
const weatherIcon = document.querySelector(".weather-icon");

function fetchWeather(city) {
  return new Promise((resolve, reject) => {
    fetch(apiUrl + city + `&appid=${apiKey}`)
      .then((response) => {
        if (!response.ok) {
          reject(new Error("City not found"));
        } else {
          resolve(response.json());
        }
      })
      .catch((error) => {
        reject(error);
      });
  });
}

function displayWeather(data) {
  document.querySelector(".city").innerHTML = data.name;
  document.querySelector(".temp").innerHTML = Math.round(data.main.temp) + "°C";
  document.querySelector(".humidity").innerHTML = data.main.humidity + "%";
  document.querySelector(".wind").innerHTML = data.wind.speed + "km/h";

  switch (data.weather[0].main) {
    case "Clouds":
      weatherIcon.src = "images/clouds.png";
      break;
    case "Clear":
      weatherIcon.src = "images/clear.png";
      break;
    case "Rain":
      weatherIcon.src = "images/rain.png";
```

```
      break;
    case "Drizzle":
      weatherIcon.src = "images/drizzle.png";
      break;
    case "Mist":
      weatherIcon.src = "images/mist.png";
      break;
    case "Snow":
      weatherIcon.src = "images/snow.png";
      break;
    default:
      weatherIcon.src = ""; // Set a default image or clear the image
  }

  document.querySelector(".weather").style.display = "block";
  document.querySelector(".error").style.display = "none";
}

function handleError(error) {
  document.querySelector(".error").style.display = "block";
  document.querySelector(".weather").style.display = "none";
  console.error("Fetch error:", error);
}

searchBtn.addEventListener("click", () => {
  const city = searchBox.value;
  fetchWeather(city).then(displayWeather).catch(handleError);
});
```

Screenshots







Conclusion:

In this assignment, we understood and implemented the concept of Promises, Fetch (Client-Server communication) and Asynchronous JavaScript.

Assignment 6(a)

Aim: WAP to implement the concept of props and state.

Lo Mapped: LO5

Theory:

Displaying Date passed as props

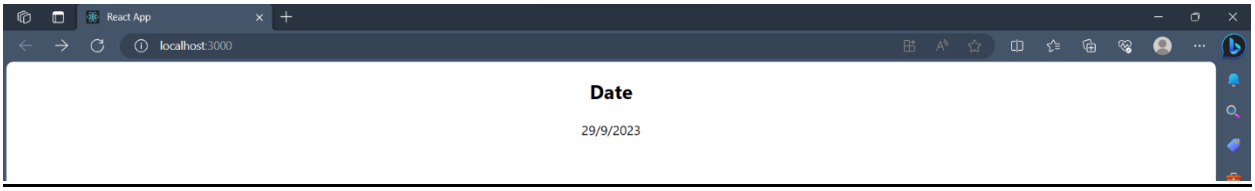
```
import React from "react";

function DateDisplay(props) {
  return (
    <div>
      <h2>Date</h2>
      <p>{props.currentDate}</p>
    </div>
  );
}

export default DateDisplay;
```

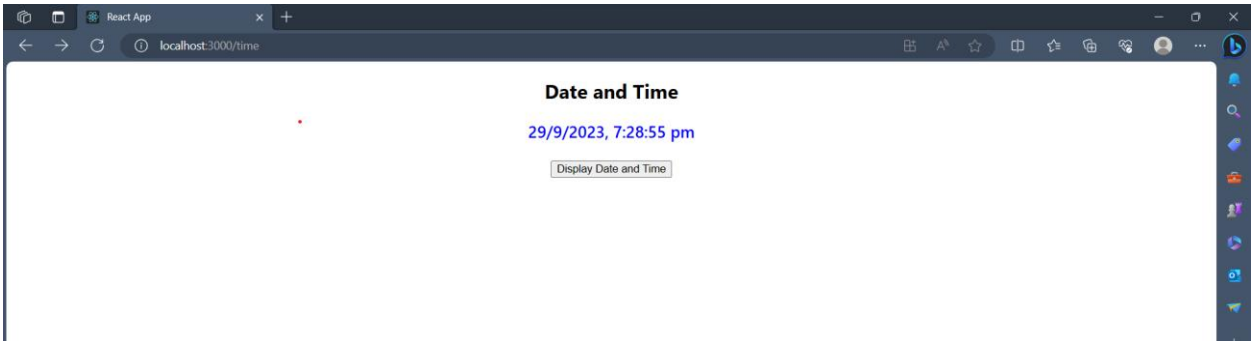
Displaying Date and Time on click

```
import React, { Component } from "react";

class DateTime extends Component {
  constructor(props) {
    super(props);
    this.state = {
      dateTime: null,
    };
  }

  handleDisplay = () => {
    const currentDateTime = new Date().toLocaleString();
    this.setState({ dateTime: currentDateTime });
  };

  render() {
    return (
      <div>
        <h2>Date and Time</h2>
        <p style={{ fontSize: "20px", fontWeight: "500", color: "blue" }}>
```

```
          {this.state.dateTime}
        </p>
        <button onClick={this.handleDisplay}>Display Date and Time</button>
      </div>
    );
  }
}

export default DateTime;
```

Screenshots





Conclusion:

In this assignment we understood and implemented the concept of props and state. We used props to display the current date and state to display the current date and time.

Assignment 6(b)

Aim: WAP to implement the concept of forms and events.

Lo Mapped: LO5

Theory:

```
import React, { Component } from "react";
import "./Registration.css";

class RegistrationForm extends Component {
  constructor(props) {
    super(props);

    this.state = {
      firstName: "",
      lastName: "",
      address: "",
      branch: "it",
      languages: {
        javascript: false,
        python: false,
        java: false,
        cpp: false,
      },
      gender: "male",
    };
  }

  handleInputChange = (event) => {
    const { name, value, type, checked } = event.target;
    if (type === "text" && !/^[a-zA-Z0-9]*$/.test(value)) {
      return;
    }

    // Update state based on input type
    if (type === "checkbox") {
      this.setState((prevState) => ({
        languages: {
          ...prevState.languages,
          [name]: checked,
        },
      }));
    } else {
      this.setState({ [name]: value });
```

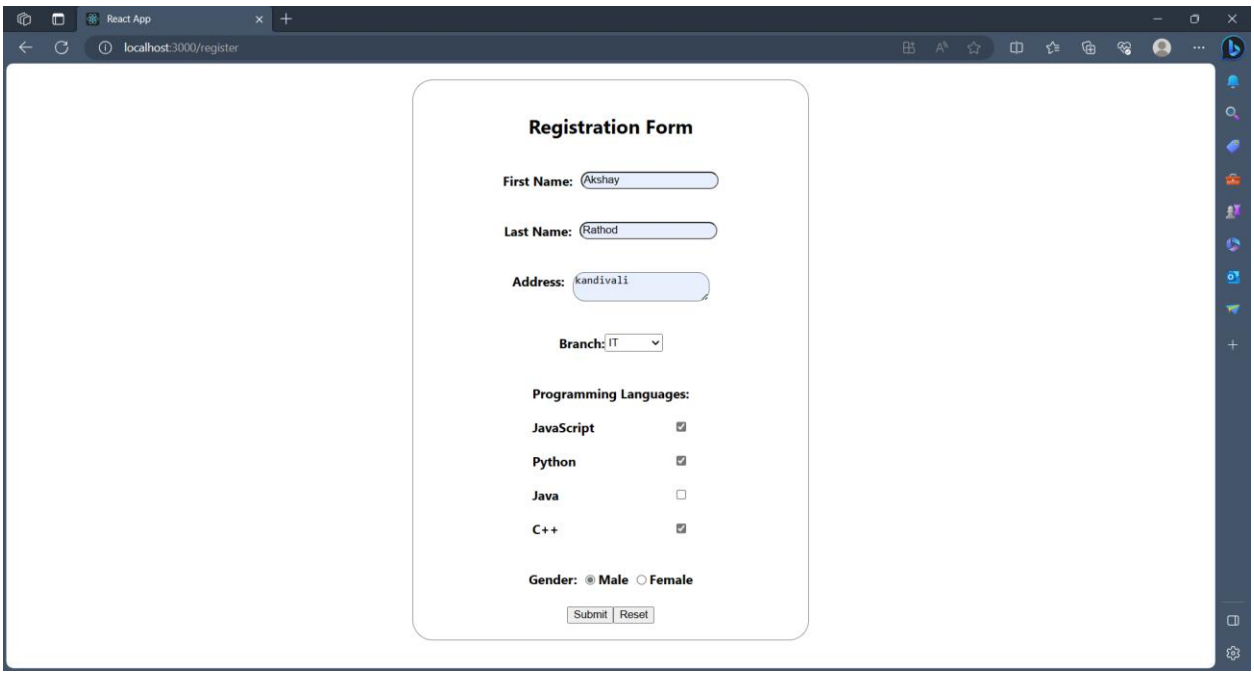
```
  }
};

handleSubmit = (event) => {
  event.preventDefault();
  // Display an alert with form data
  alert("Form submitted successfully!");
};

handleReset = () => {
  this.setState({
    firstName: "",
    lastName: "",
    address: "",
    branch: "it",
    languages: {
      javascript: false,
      python: false,
      java: false,
      cpp: false,
    },
    gender: "male",
  });
};

handleKeyPress = (e) => {
  if (e.key === "Enter") {
    this.handleSubmit(e);
  }
};

render() {
  return (
    <form onSubmit={this.handleSubmit} onKeyPress={this.handleKeyPress}>
      <h2>Registration Form</h2>
      <div>
        <label>
          First Name:
          <input
            type="text"
            name="firstName"
            value={this.state.firstName}
            onChange={this.handleInputChange}
            required
          />
```

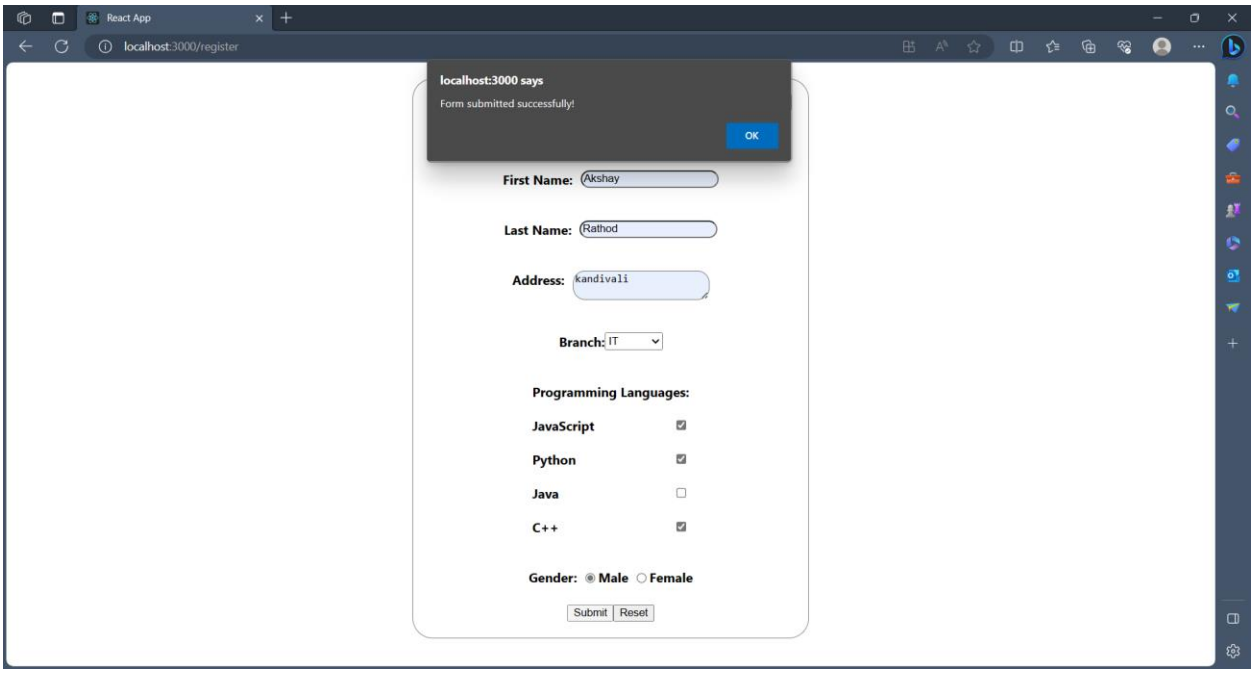
```
          </label>
        </div>
        <div>
          <label>
            Last Name:
            <input
              type="text"
              name="lastName"
              value={this.state.lastName}
              onChange={this.handleInputChange}
              required
            />
          </label>
        </div>
        <div>
          <label>
            Address:
            <textarea
              name="address"
              value={this.state.address}
              onChange={this.handleInputChange}
            />
          </label>
        </div>
        <div>
          <label>
            Branch:
            <select
              name="branch"
              value={this.state.branch}
              onChange={this.handleInputChange}
            >
              <option value="it">IT</option>
              <option value="comps">COMPS</option>
              <option value="aids">AIDS</option>
            </select>
          </label>
        </div>
        <div className="check">
          <label>Programming Languages:</label>
          <div>
            <label>
              JavaScript
              <input
                type="checkbox"
```

```
            name="javascript"
            checked={this.state.languages.javascript}
            onChange={this.handleInputChange}
          />
        </label>
        <label>
          Python
          <input
            type="checkbox"
            name="python"
            checked={this.state.languages.python}
            onChange={this.handleInputChange}
          />
        </label>
        <label>
          Java
          <input
            type="checkbox"
            name="java"
            checked={this.state.languages.java}
            onChange={this.handleInputChange}
          />
        </label>
        <label>
          C++
          <input
            type="checkbox"
            name="cpp"
            checked={this.state.languages.cpp}
            onChange={this.handleInputChange}
          />
        </label>
      </div>
    </div>
    <div>
      <label>
        Gender:
        <input
          type="radio"
          name="gender"
          value="male"
          checked={this.state.gender === "male"}
          onChange={this.handleInputChange}
        />
        Male
```

```
            <input
              type="radio"
              name="gender"
              value="female"
              checked={this.state.gender === "female"}
              onChange={this.handleInputChange}
            />
            Female
          </label>
        </div>
        <div>
          <button type="submit">Submit</button>
          <button type="button" onClick={this.handleReset}>
            Reset
          </button>
        </div>
      </form>
    );
  }
}

export default RegistrationForm;
```

Screenshots





Conclusion:

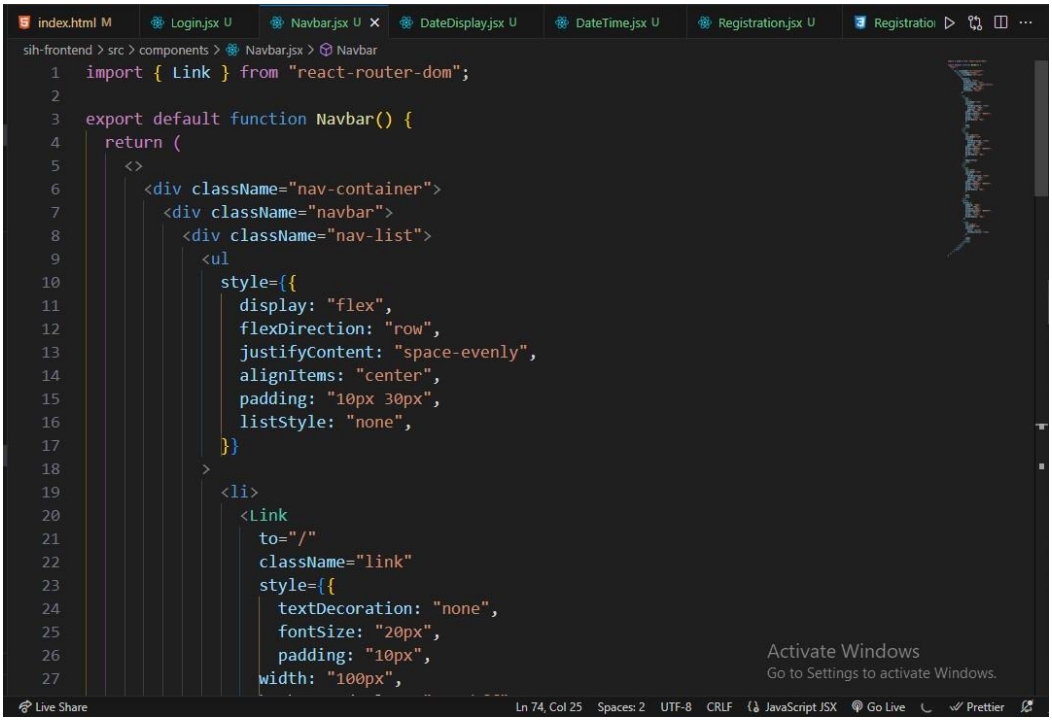
In this assignment we understood and implemented the concept of forms and events. We used forms and events to create a registration form in React

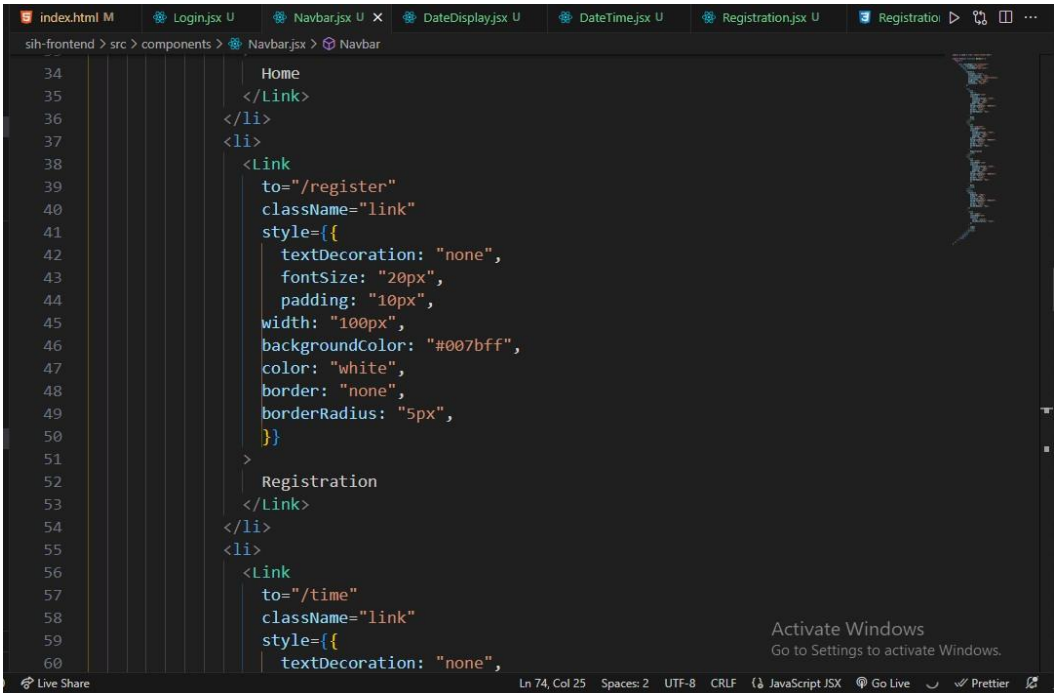
ASSIGNMENT 7(a)

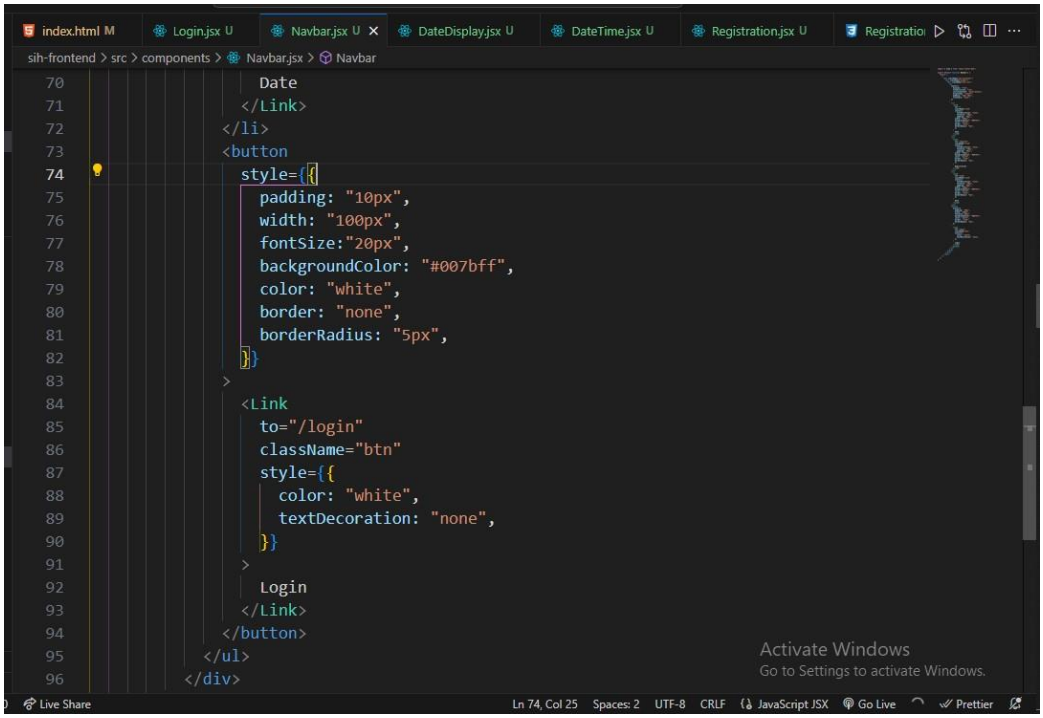
AIM: WAP to implement ReactJS Router and Animation.

LO MAPPED: LO 5 THEORY:

Code-

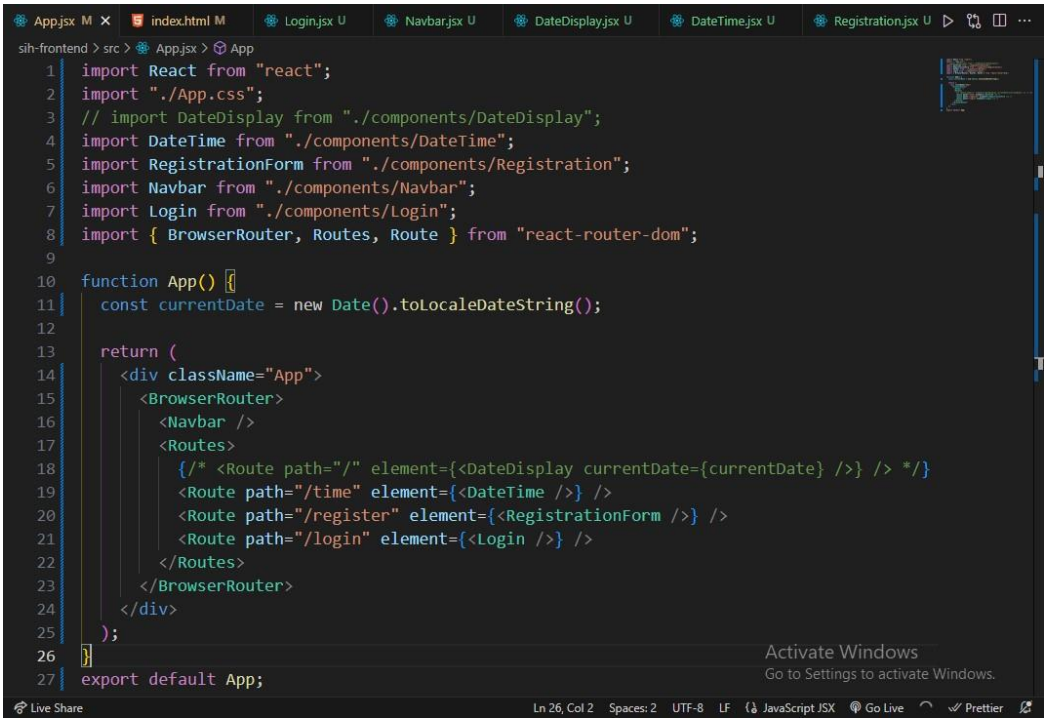





index.html M
Login.jsx U
Navbar.jsx U
DateDisplay.jsx U
DateTime.jsx U
Registration.jsx U
sih-frontend > src > components > Navbar.jsx > Navbar
Date
</Link>

<button
style={{
padding: "10px",
width: "100px",
fontSize:"20px",
backgroundColor: "#007bff",
color: "white",
border: "none",
borderRadius: "5px",
}}
>
<Link
to="/login"
className="btn"
style={{
color: "white",
textDecoration: "none",
}}
>
Login
</Link>
</button>

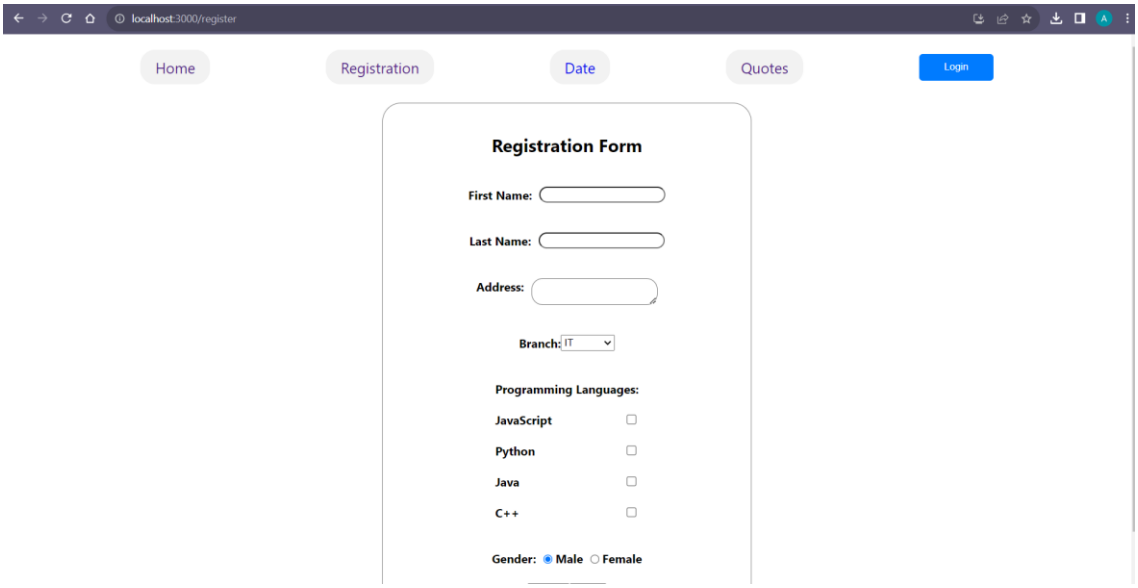
Activate Windows
Go to Settings to activate Windows.
Live Share
Ln 74, Col 25
Spaces: 2
UTF-8
CRLF
JavaScript JSX
Go Live
Prettier


App.jsx M
index.html M
Login.jsx U
Navbar.jsx U
DateDisplay.jsx U
DateTime.jsx U
Registration.jsx U
sih-frontend > src > App.jsx > App
import React from "react";
import "./App.css";
// import DateDisplay from "./components/DateDisplay";
import DateTime from "./components/DateTime";
import RegistrationForm from "./components/Registration";
import Navbar from "./components/Navbar";
import Login from "./components/Login";
import { BrowserRouter, Routes, Route } from "react-router-dom";
function App() {
const currentDate = new Date().toLocaleDateString();
return (

<BrowserRouter>
<Navbar />
<Routes>
{/* <Route path="/" element={<DateDisplay currentDate={currentDate} />} /> */}
<Route path="/time" element={<DateTime />} />
<Route path="/register" element={<RegistrationForm />} />
<Route path="/login" element={<Login />} />
</Routes>
</BrowserRouter>

);
}
export default App;
Activate Windows
Go to Settings to activate Windows.
Live Share
Ln 26, Col 2
Spaces: 2
UTF-8
LF
JavaScript JSX
Go Live
Prettier

Output-



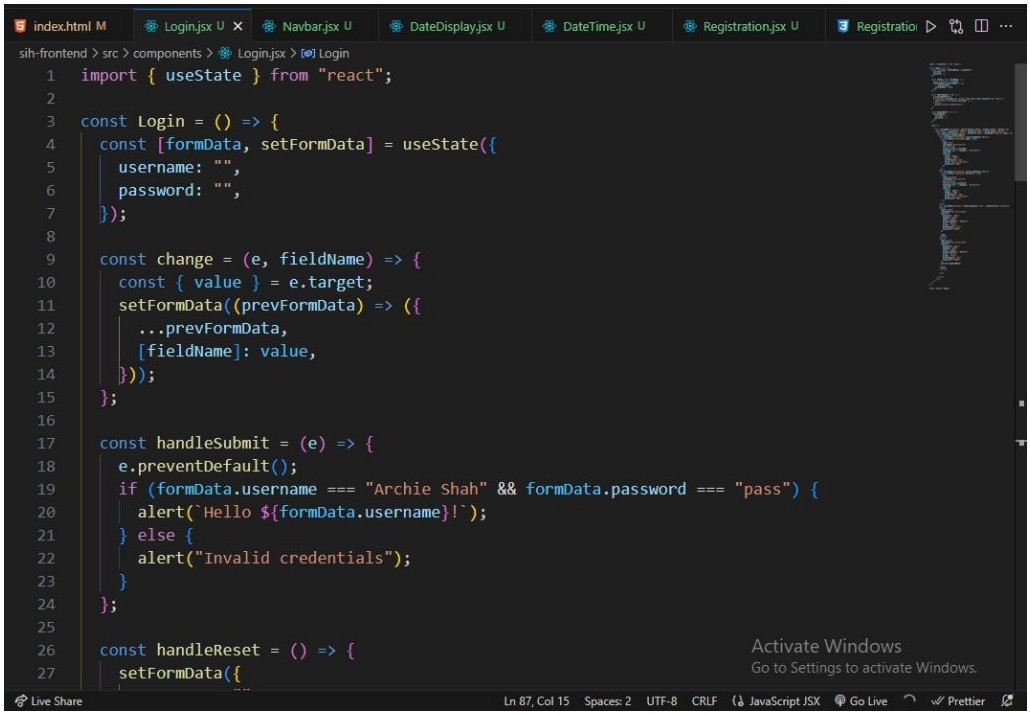
CONCLUSION: In this assignment we learnt about routing in React and implemented it using React-router-dom.

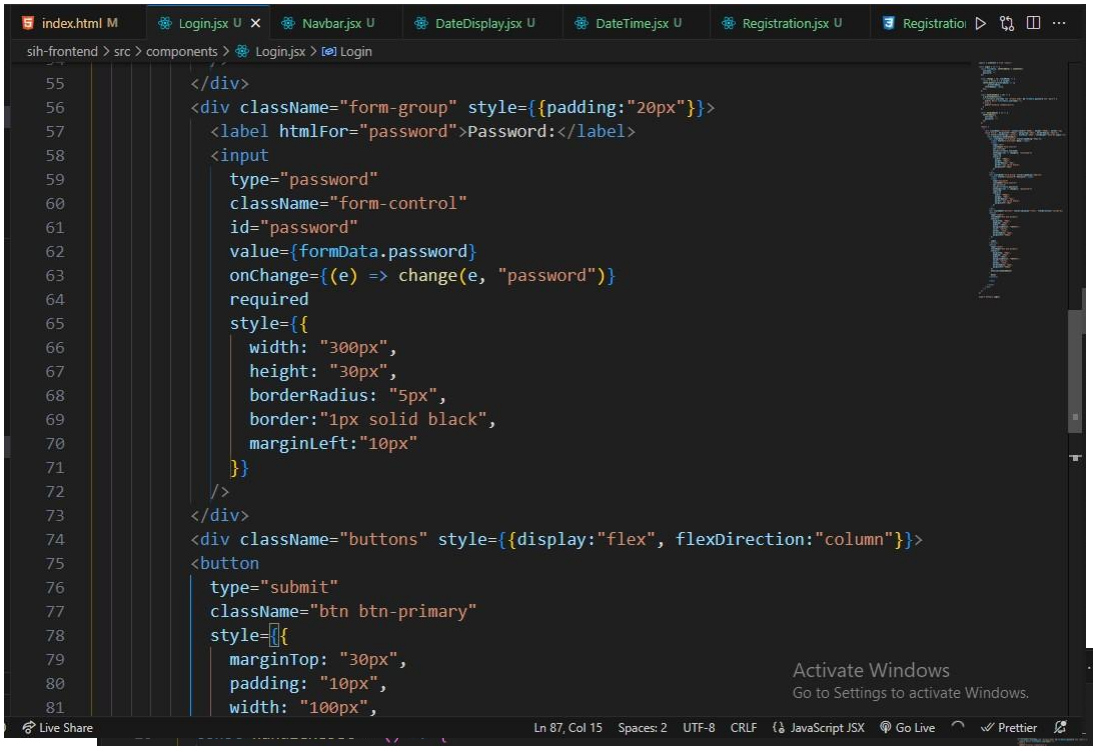
ASSIGNMENT 7(b)

AIM: WAP to implement the concept of React Hooks

LO MAPPED: LO5 THEORY:

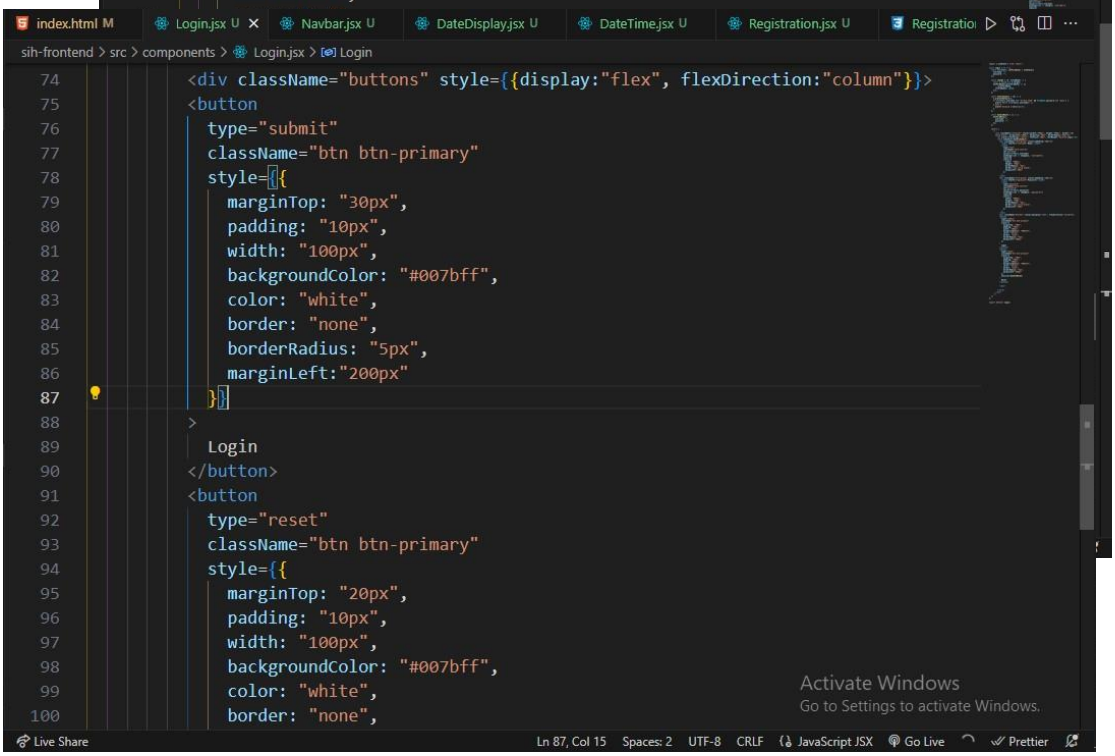
Code-

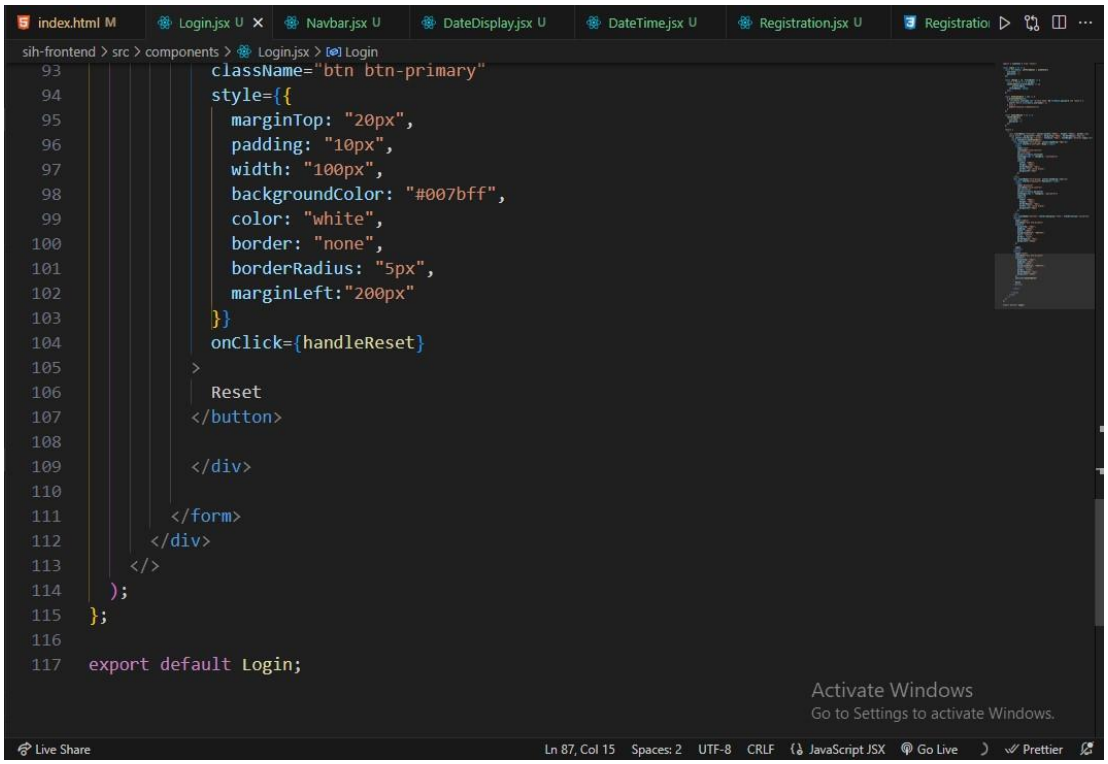


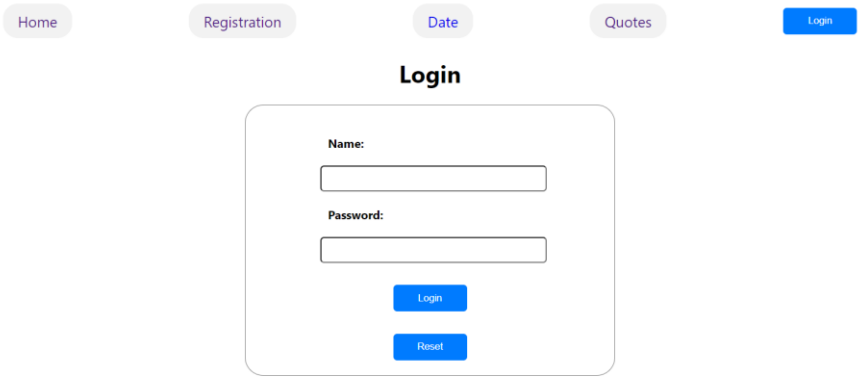

index.html M
Login.jsx U
Navbar.jsx U
DateDisplay.jsx U
DateTime.jsx U
Registration.jsx U
sih-frontend > src > components > Login.jsx > Login

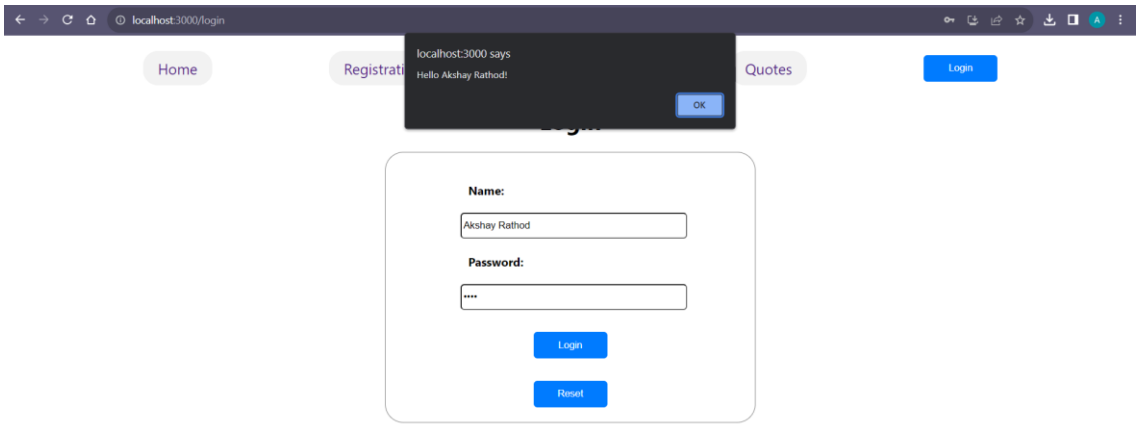
<div className="form-group" style={{padding:"20px"}}>
<label htmlFor="password">Password:</label>
<input
type="password"
className="form-control"
id="password"
value={formData.password}
onChange={(e) => change(e, "password")}
required
style={{
width: "300px",
height: "30px",
borderRadius: "5px",
border:"1px solid black",
marginLeft:"10px"
}}
/>

<div className="buttons" style={{display:"flex", flexDirection:"column"}}>
<button
type="submit"
className="btn btn-primary"
style={{
marginTop: "30px",
padding: "10px",
width: "100px",
Activate Windows
Go to Settings to activate Windows.
Live Share
Ln 87, Col 15
Spaces: 2
UTF-8
CRLF
JavaScript JSX
Go Live
Prettier


setFormData({
username: "",
index.html M
Login.jsx U
Navbar.jsx U
DateDisplay.jsx U
DateTime.jsx U
Registration.jsx U
sih-frontend > src > components > Login.jsx > Login
<div className="buttons" style={{display:"flex", flexDirection:"column"}}>
<button
type="submit"
className="btn btn-primary"
style={{
marginTop: "30px",
padding: "10px",
width: "100px",
backgroundColor: "#007bff",
color: "white",
border: "none",
borderRadius: "5px",
marginLeft:"200px"
}}
>
Login
</button>
<button
type="reset"
className="btn btn-primary"
style={{
marginTop: "20px",
padding: "10px",
width: "100px",
backgroundColor: "#007bff",
color: "white",
border: "none",
Activate Windows
Go to Settings to activate Windows.
Live Share
Ln 87, Col 15
Spaces: 2
UTF-8
CRLF
JavaScript JSX
Go Live
Prettier


index.html M
Login.jsx U
Navbar.jsx U
DateDisplay.jsx U
DateTime.jsx U
Registration.jsx U
sih-frontend > src > components > Login.jsx > Login
93 className="btn btn-primary"
94 style={{
95 marginTop: "20px",
96 padding: "10px",
97 width: "100px",
98 backgroundColor: "#007bff",
99 color: "white",
100 border: "none",
101 borderRadius: "5px",
102 marginLeft:"200px"
103 }}
104 onClick={handleReset}
105 >
106 Reset
107 </button>
108
109
110
111 </form>
112
113 </>
114);
115 };
116
117 export default Login;
Activate Windows
Go to Settings to activate Windows.
Live Share
Ln 87, Col 15
Spaces: 2
UTF-8
CRLF
JavaScript JSX
Go Live
Prettier


Home
Registration
Date
Quotes
Login
Login
Name:
Password:
Login
Reset



CONCLUSION: In this assignment we learnt about the concept of hooks in React and implemented it using a log-in form.

IP Assignment 8

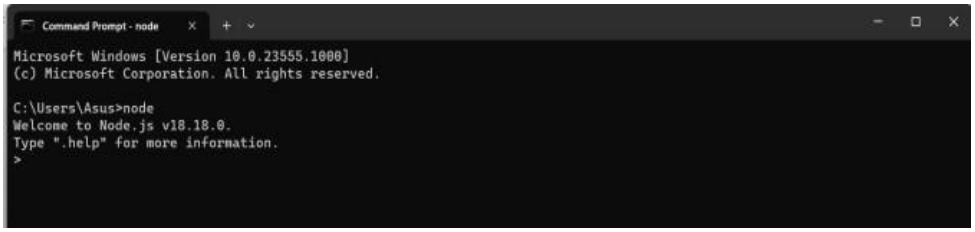
Aim: REPL and Node.JS Filesystem commands Implementation.

LO Mapped: LO6

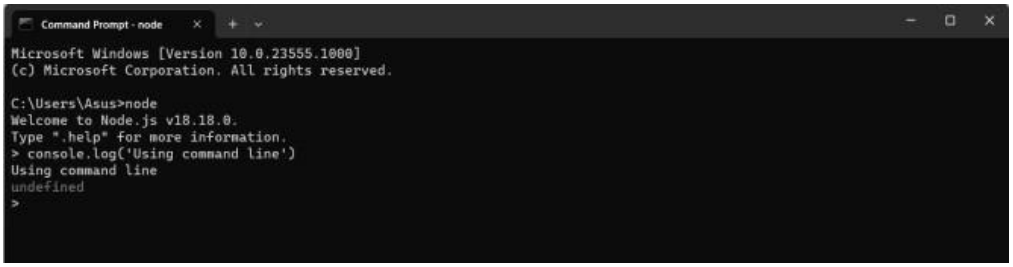
Theory:

1. **Accessing the REPL:**

   Type **node** and press Enter to enter the Node.js REPL.



2. **Basic Commands:**



3. **Multiline Commands:**

   - Use the .editor command to enter multiline mode for writing longer code snippets.
   - Press **Ctrl + D** to run the code.

4. **Special Commands:**

   - Use **.help** to view a list of special commands.
   - **.exit** or **Ctrl + D** to exit the REPL.

Node.js Filesystem Commands

1. **Include the fs Module:**

$const\ fs$ = $require('fs')$;

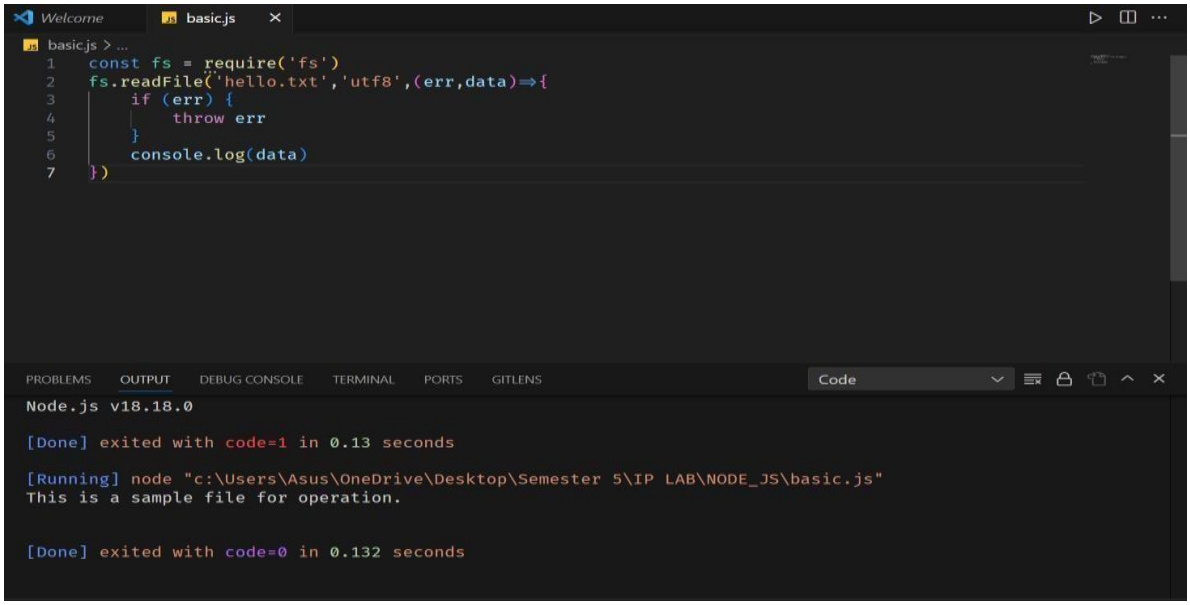
2. **Reading a File:**

$fs.readFile('filename.txt', 'utf8', (err, data)$ => {

$if\ (err)\ throw\ err;$

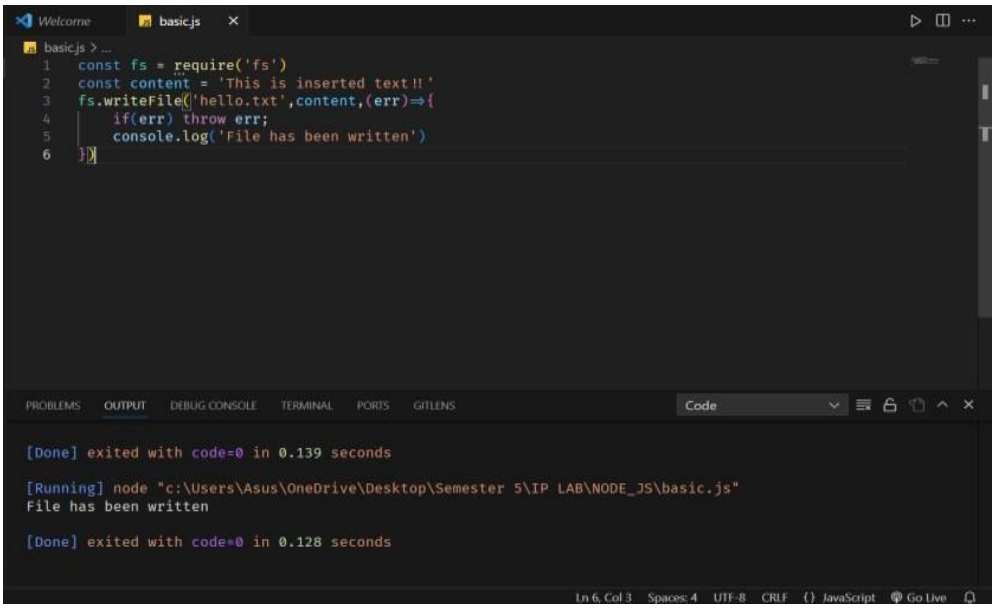
$console.log(data);$

});



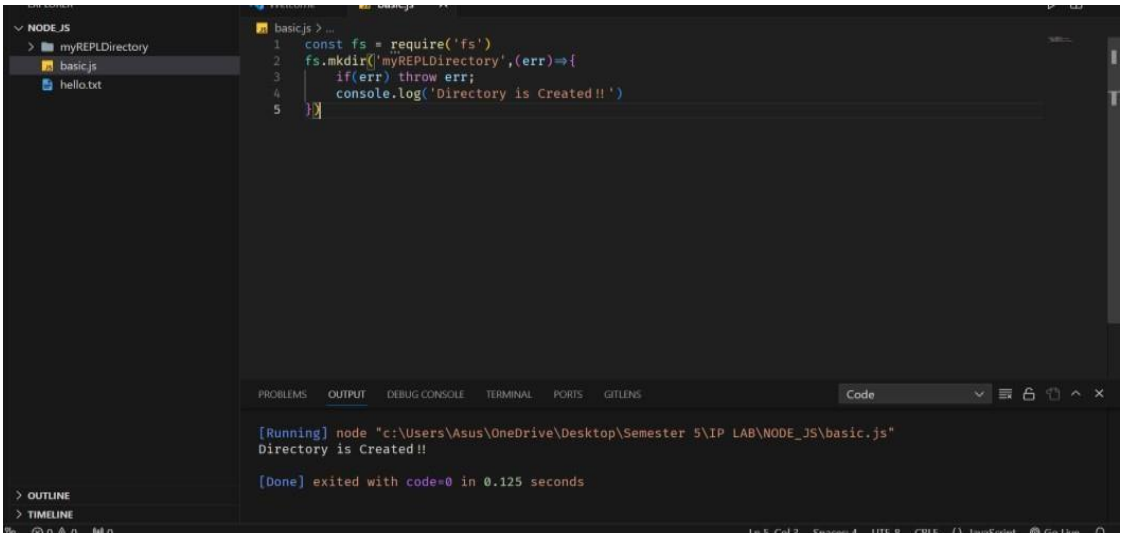
3. **Writing to a File:**

```
const content = 'Hello, Node.js!';
fs.writeFile('output.txt', content, (err) => {
  if (err) throw err;
  console.log('File has been written.');
});
```



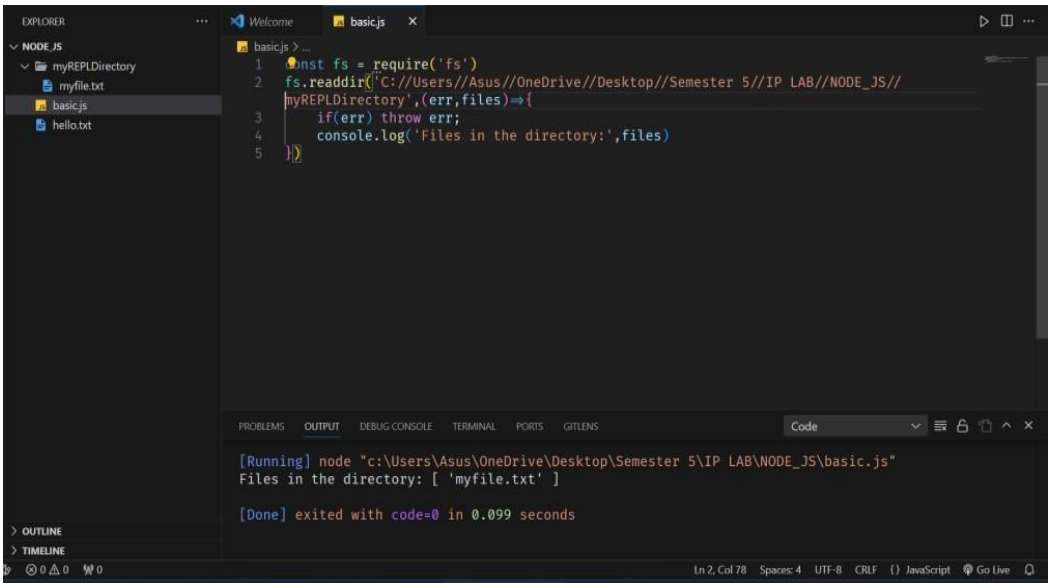
4. **Creating a Directory:**

```
fs.mkdir('newDirectory', (err) => {
  if (err) throw err;
  console.log('Directory created.'); });
```



5. **Listing Files in a Directory:**

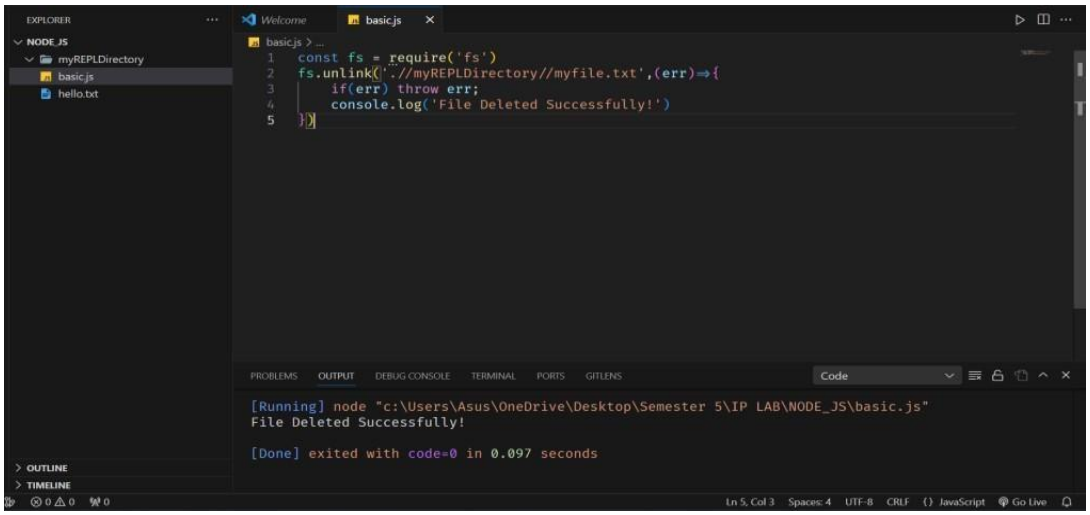
```
fs.readdir('directoryPath', (err, files) => {
  if (err) throw err;
  console.log('Files in the directory:', files);
});
```



6. **Deleting a File:**

```
fs.unlink('filename.txt', (err) => {
  if (err) throw err;
  console.log('File deleted.');
```

});



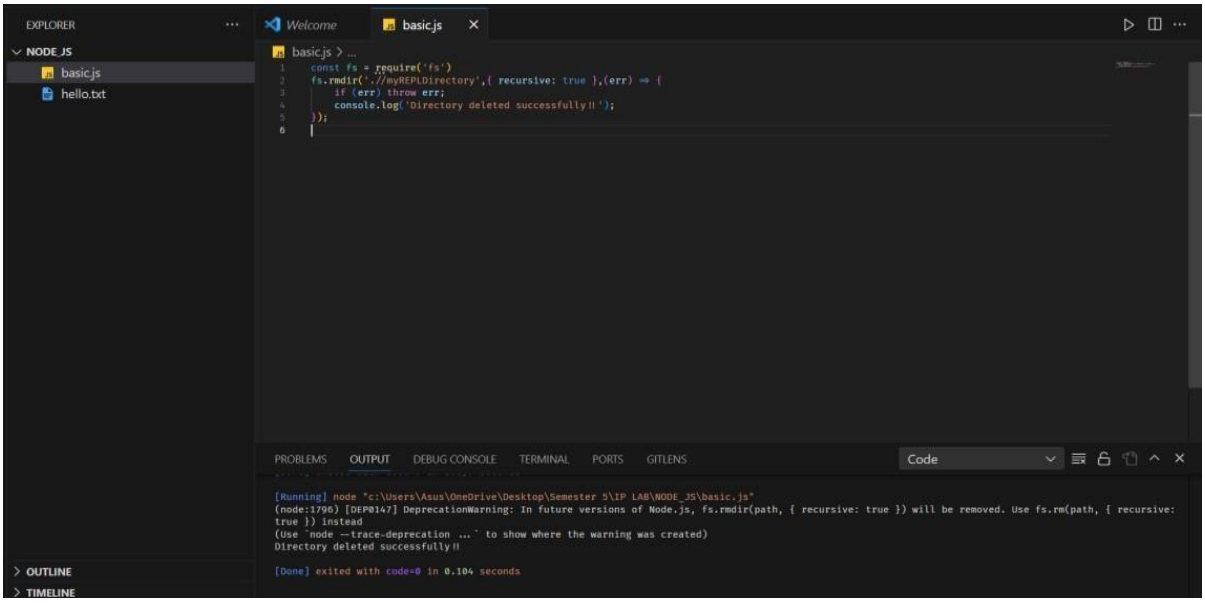
7. **Deleting a Directory:**

$fs.rmdir('directoryPath', \{ recursive: true \}, (err) => \{$

$if\ (err)\ throw\ err;$

$console.log('Directory\ deleted.');$

});



Conclusion: By this assignment we learned concept of REPL and Node.JS Filesystem commands Implementation.

ASSIGNMENT 9

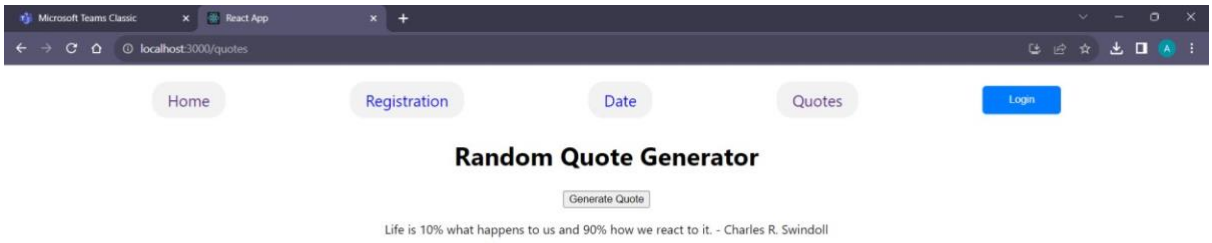
Aim: WAP to understand the concept of References in react and implement it.

Lo Mapped: LO 5

Theory:

```
import React, { Component } from "react";
 class Quotes extends Component
{   constructor(props) {
super(props);     this.state =
{       quotes: [
        "Life is 10% what happens to us and 90% how we react to it. - Charles
R. Swindoll",
        "The only way to do great work is to love what you do. - Steve Jobs",
        "In the middle of every difficulty lies opportunity. - Albert Einstein",
        "Success is not final, failure is not fatal: It is the courage to
continue that counts. - Winston Churchill",
        "Don't watch the clock; do what it does. Keep going. - Sam Levenson",
      ],
    };      this.myRef =
React.createRef();
  }    generateRandomQuote = () => {     const randomIndex =
Math.floor(Math.random() * this.state.quotes.length);     const randomQuote
= this.state.quotes[randomIndex];     this.myRef.current.textContent =
randomQuote;
  };
render() {
return (
      <div>
        <h1>Random Quote Generator</h1>
        <button onClick={this.generateRandomQuote}>Generate Quote</button>
        <p ref={this.myRef}>
          Press the button to generate a random quote.
        </p>
      </div>
    );
  }
}  export default
Quotes;
```

Screenshots:



Conclusion:

In this assignment we learnt about references in react and implemented it using class compo

Assignment 10

Branch: IT/T2 Roll No:96,99,115,117 Date:20/10/2023

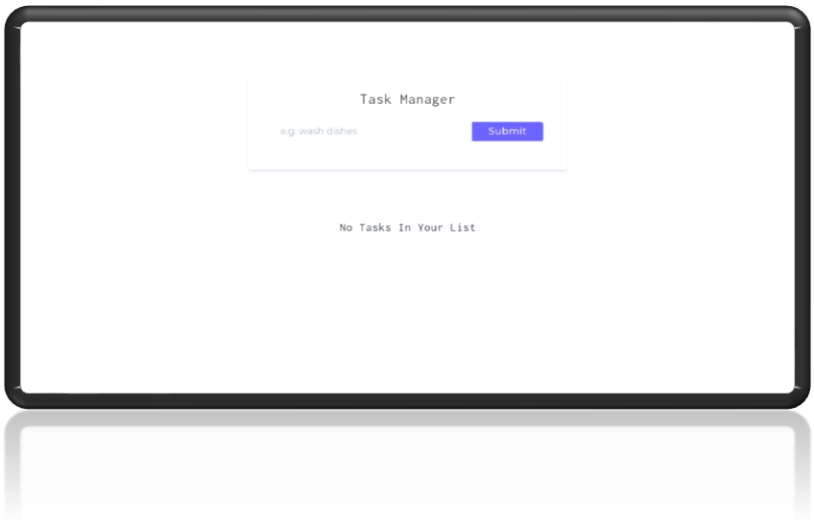
Aim: Create a web application that performs crud operations (database connectivity).

LO mapped: All Lab outcomes were covered

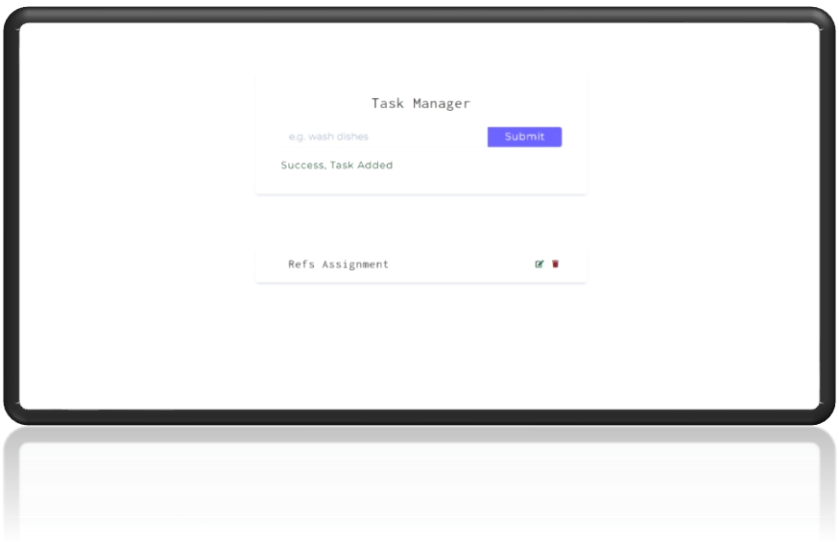
Theory:

A task manager, often referred to as a task management tool or task management software, is a digital or physical system that helps individuals, teams, or organizations organize, prioritize, and track their tasks and activities. Task managers are invaluable tools for boosting productivity, managing workloads, and ensuring that important tasks are completed efficiently and on time.

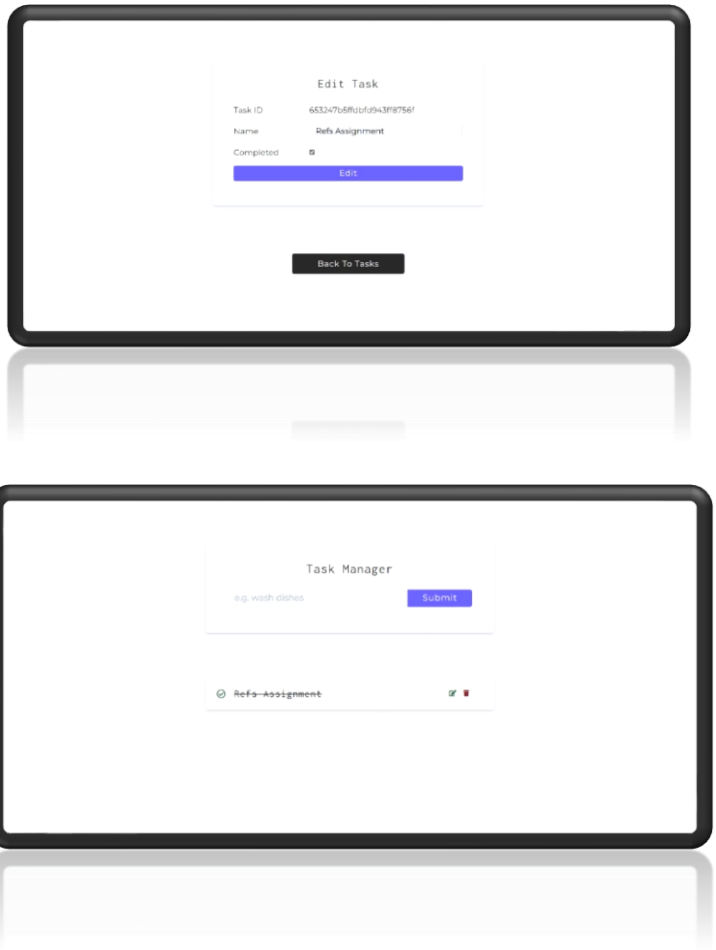
The app provides with following things:



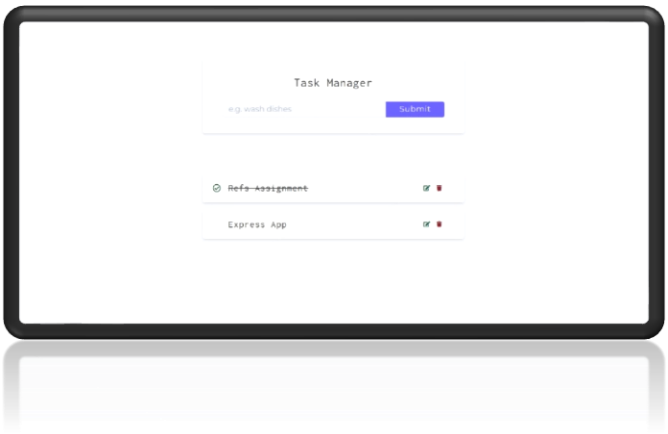
1. **Task Creation:** Users can create and add new tasks to their task list.

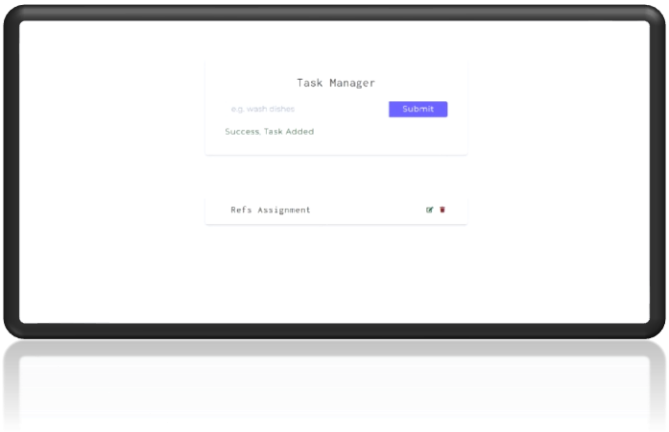


2. **Task Editing:** Ability to edit and update task details, including title, description, due date, priority, and tags.



3. **Task Deletion:** Ability to delete tasks that are no longer relevant or needed.





These features can make a task manager app versatile and user-friendly, catering to a wide range of task management needs, from individual task tracking to team collaboration on complex projects. The specific features you choose to include in your app should align with your target audience and their requirements.

Conclusion: In this assignment we learnt about CRUD operations and created an Express working app.