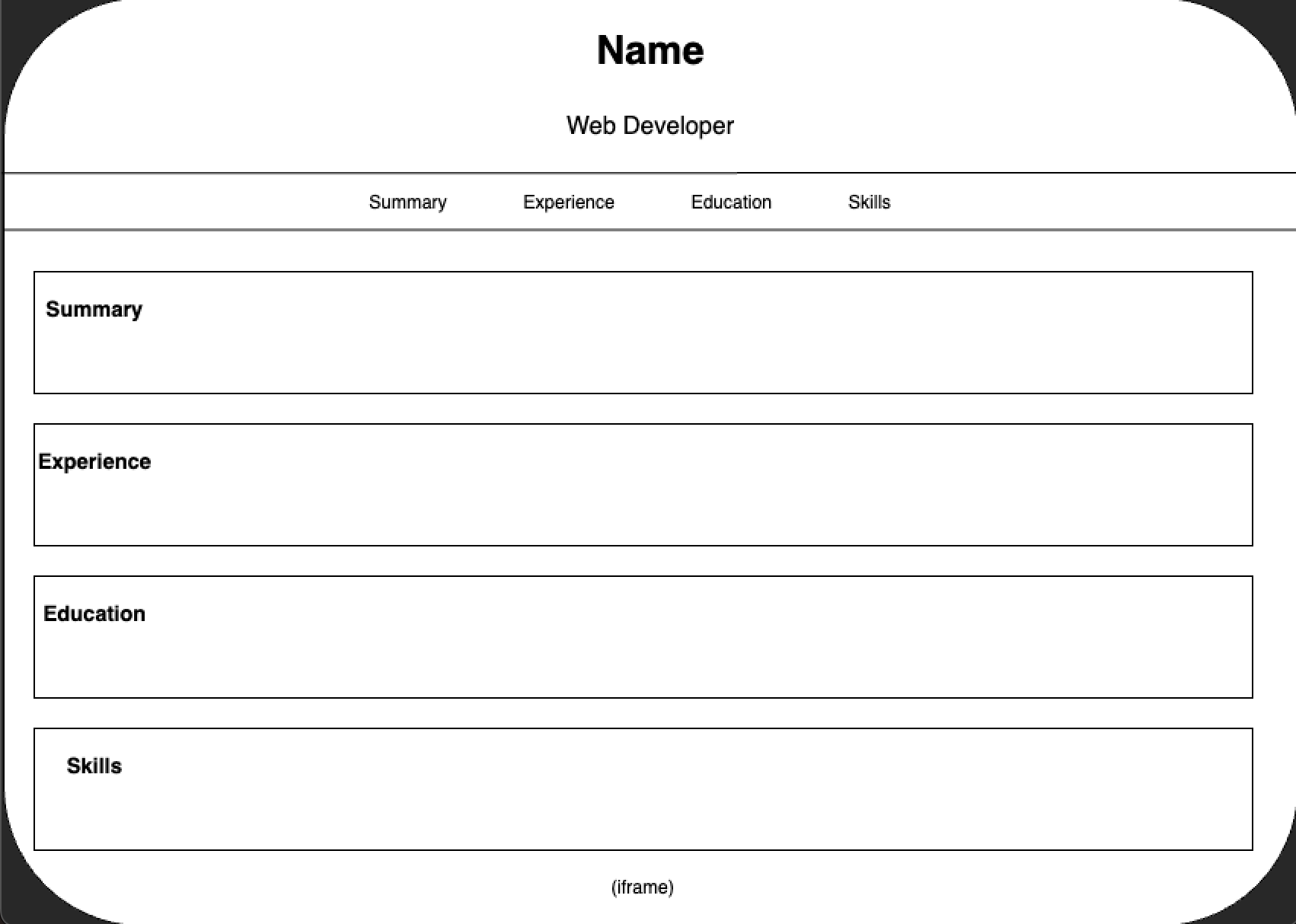
| **Ex. No: 1** | **HTML and CSS - Text Formatting, Hyperlink, Images, Frames, Tables, Links** |
| --- | --- |
| **Date: 17/7/23** |

**Aim:** To design a rough layout of a CV as a webpage and to implement it using HTML and CSS including Text formatting, Hyperlink, Images, Frames, Tables and Links.

**Design:**



**Program :**

HTML:

<!DOCTYPE html>

<html>

<head>

<title>CV\_design</title>

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

</head>

<body>

<header>

<h1>Name</h1>

<p>Web Developer</p>

</header>

<nav>

<a href="#summary">Summary</a>

<a href="#experience">Experience</a>

<a href="#education">Education</a>

<a href="#skills">Skills</a>

</nav>

<section id="summary">

<h2>Summary</h2>

<p>I am a good timekeeper, always willing to learn new skills. I am friendly, helpful and polite, have a good sense of humour. I am able to work independently in busy environments and also within a team setting. I am outgoing and tactful, and able to listen effectively when solving problems.</p>

</section>

<section id="experience">

<h2>Experience</h2>

<table>

<tr>

<th>Position</th>

<th>Company</th>

<th>Year</th>

</tr>

<tr>

<td>Web Developer</td>

<td>ABC Inc.</td>

<td>2020-2023</td>

</tr>

</table>

</section>

<section id="education">

<h2>Education</h2>

<table>

<tr>

<th>Degree</th>

<th>School/University</th>

<th>Year</th>

</tr>

<tr>

<td>B.Tech. AI and Data Science</td>

<td>Shiv Nadar University</td>

<td>2021-2025</td>

</tr>

</table>

</section>

<section id="skills">

<h2>Skills</h2>

<ul>

<li>HTML/CSS</li>

<li>JavaScript</li>

<li>Web Design</li>

<li>Responsive Design</li>

</ul>

</section>

<iframe src="https://www.snuchennai.edu.in/" height="200" width="300" title = "Shiv Nadar University" margin = auto></iframe>

</body>

</html>

CSS:

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

background-color: #f7f7f7;

}

header {

background-color: #333;

color: #fff;

text-align: center;

padding: 20px;

}

h1 {

margin: 0;

font-size: 36px;

}

p {

font-size: 18px;

}

nav {

background-color: #444;

padding: 10px;

text-align: center;

}

nav a {

color: #fff;

text-decoration: none;

font-size: 16px;

margin: 0 20px;

}

nav a:hover {

text-decoration: underline;

}

section {

max-width: 800px;

margin: 20px auto;

background-color: #fff;

padding: 20px;

border-radius: 5px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

h2 {

margin-top: 0;

font-size: 24px;

}

img {

max-width: 100%;

height: auto;

border-radius: 5px;

}

table {

width: 100%;

border-collapse: collapse;

margin-top: 20px;

}

th, td {

border: 1px solid #ccc;

padding: 10px;

text-align: left;

}

th {

background-color: #f2f2f2;

}

ul {

list-style: none;

padding: 0;

}

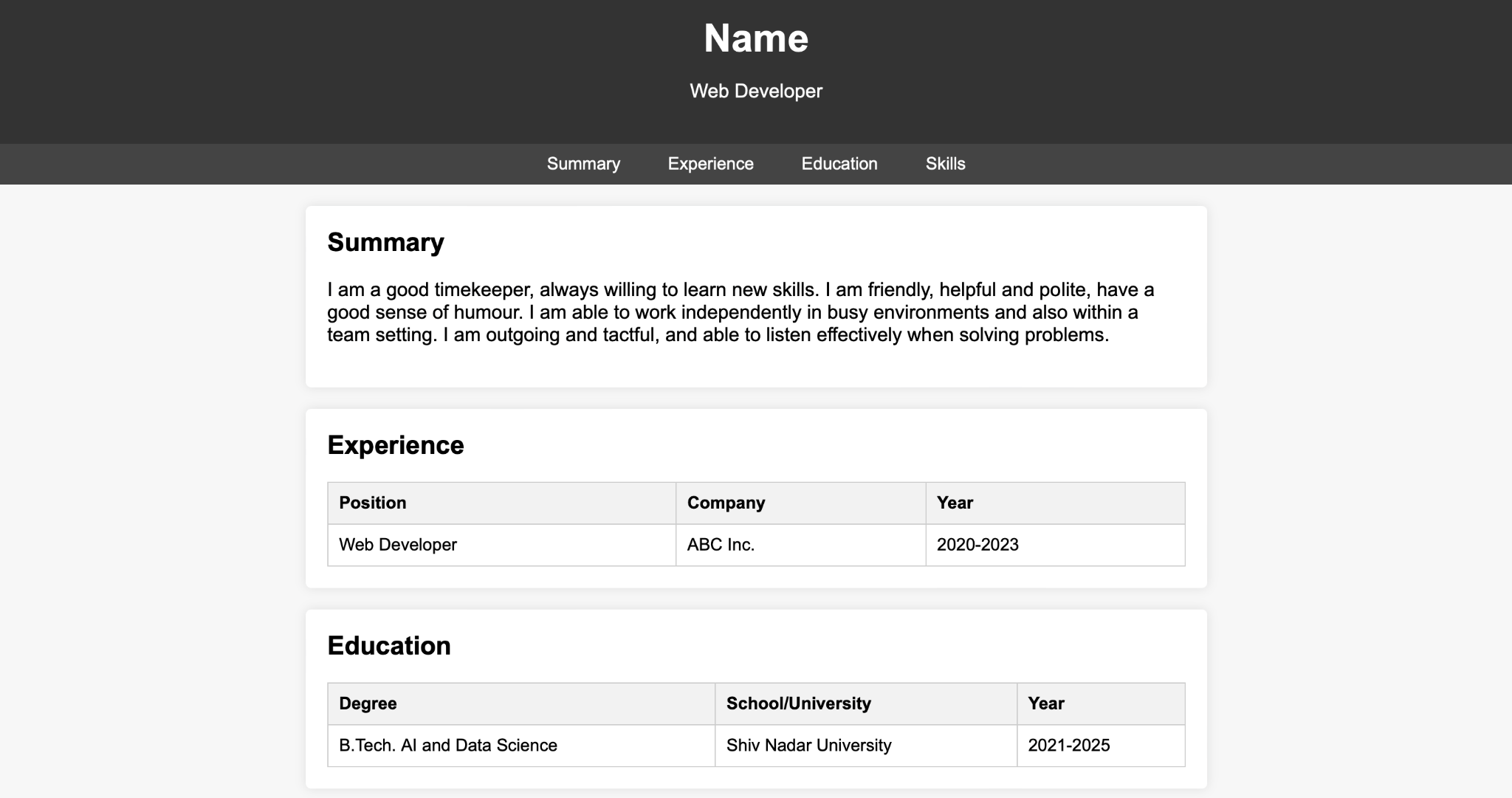
li {

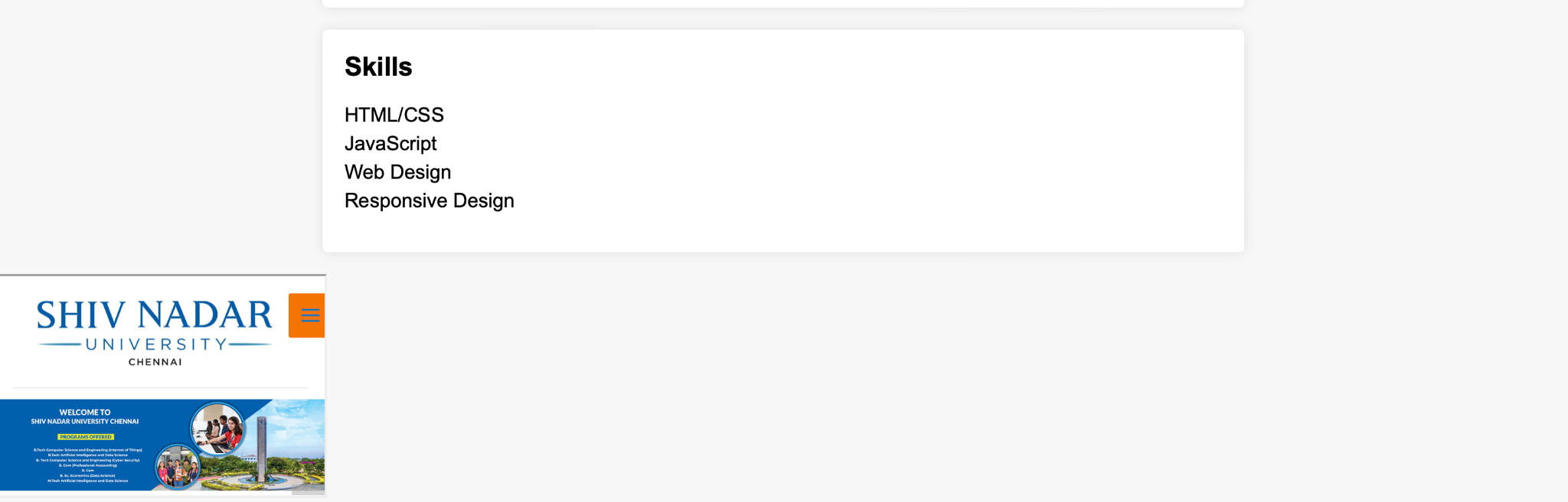
font-size: 18px;

margin-bottom: 5px;

}

**Output:**

****

****

**Result:** The CV has been implemented as a webpage.

| **Ex. No: 2** | **Form Validation** |
| --- | --- |
| **Date: 24/7/23** |

**Aim:** To design and implement a client-side form validation script for a user registration form on a website containing the following fields: Username, Email, Password, Phone Number, Confirm Password.

**Design:**

****

.

**Program :**

HTML:

<!DOCTYPE html>

<html>

<head>

<title>User Registration Form</title>

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

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

</head>

<body>

<form onsubmit="return validateForm()">

<label for="username">Username:</label>

<input type="text" id="username" name="username"><br>

<label for="email">Email:</label>

<input type="text" id="email" name="email"><br>

<label for="password">Password:</label>

<input type="password" id="password" name="password"><br>

<label for="confirmPassword">Confirm Password:</label>

<input type="password" id="confirmPassword" name="confirmPassword"><br>

<label for="phoneNumber">Phone Number:</label>

<input type="text" id="phoneNumber" name="phoneNumber"><br>

<input type="submit" value="Register">

</form>

</body>

</html>

CSS:

body {

font-family: Arial, sans-serif;

background-color: #f7f7f7;

margin: 0;

padding: 0;

}

form {

max-width: 400px;

margin: 0 auto;

background-color: #fff;

padding: 20px;

border-radius: 5px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);

}

label {

display: block;

margin-bottom: 5px;

}

input {

width: 100%;

padding: 10px;

margin-bottom: 10px;

border: 1px solid #ccc;

border-radius: 5px;

}

input[type="submit"] {

background-color: #333;

color: #fff;

border: none;

border-radius: 5px;

padding: 10px 20px;

cursor: pointer;

}

input[type="submit"]:hover {

background-color: #444;

}

.error {

color: red;

font-size: 14px;

}

JavaScript:

function validateForm() {

var username = document.getElementById("username").value;

var email = document.getElementById("email").value;

var password = document.getElementById("password").value;

var confirmPassword = document.getElementById("confirmPassword").value;

var phoneNumber = document.getElementById("phoneNumber").value;

var usernamePattern = /^[a-zA-Z0-9]{3,20}$/;

var emailPattern = /^[a-zA-Z0-9.\_-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}$/;

var passwordPattern = /^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d)(?=.\*[@#$])[A-Za-z\d@#$]{8,}$/;

var phoneNumberPattern = /^\+91 \d{10}$/;

var errors = [];

if (!username.match(usernamePattern)) {

errors.push("Username should be 3-20 alphanumeric characters.");

}

if (!email.match(emailPattern)) {

errors.push("Email should be in the format name@example.com.");

}

if (!password.match(passwordPattern)) {

errors.push("Password should be at least 8 characters with at least one uppercase letter, one lowercase letter, one number, and one special character.");

}

if (password !== confirmPassword) {

errors.push("Passwords do not match.");

}

if (!phoneNumber.match(phoneNumberPattern)) {

errors.push("Phone number should be in the format +91 1234567890.");

}

if (errors.length > 0) {

alert(errors.join("\n"));

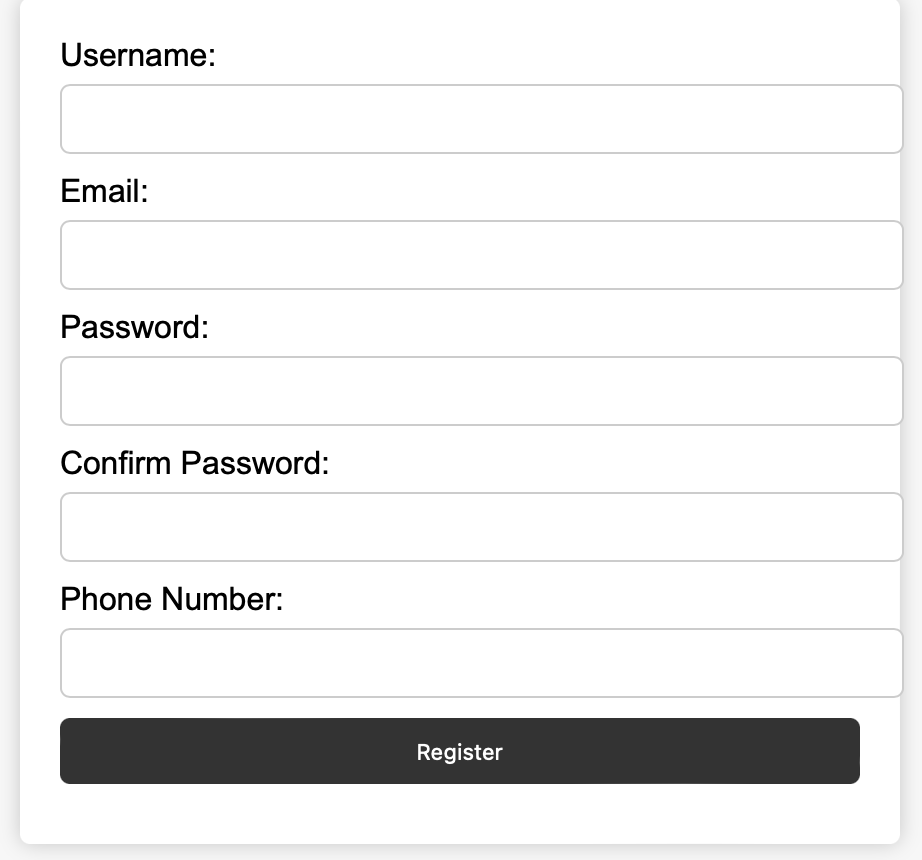
return false;

}

return true;

}

**Output:**

****

**Result:** A form validation script for a user registration form on a website has been implemented.

| **Ex. No: 3** | **Toggling Light Theme & Dark Theme** |
| --- | --- |
| **31/7/23** |

**Aim:**

Design and implement a CSS with dark and light theme using Javascript(The default CSS shall be in light mode)

**Design:**

Target design :

****

.

**Program :**

**index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

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

<link rel="stylesheet" href="style.css" />

<title>Resume</title>

<link rel="preconnect" href="https://fonts.googleapis.com" />

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />

<link

href="https://fonts.googleapis.com/css2?family=Inter:wght@100;200;300;400;500;600;700;800;900&display=swap"

rel="stylesheet"

/>

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>

<script src="https://kit.fontawesome.com/9d47ca84a7.js" crossorigin="anonymous"></script>

<link href="https://fonts.googleapis.com/css2?family=Lora:ital@0;1&display=swap" rel="stylesheet">

</head>

<body id="body-content">

<div class="theme-container">

<div class = "theme-toggle" id="toggleBtn">

<div id = "dark-btn">

<i class="fa-solid fa-moon"></i>

</div>

<div id = "light-btn">

<i class="fa-solid fa-sun" style="color: #ff7800;"></i>

</div>

<div id="custom-btn">

<i class="fa-solid fa-fill-drip" style="color: #be3737;"></i>

</div>

</div>

</div>

<div class = "content-body">

<div class="header">

<div class="header-first">

<div class="header-first-div">

<h1>Surya Sivakumar</h1>

<h2>Shiv Nadar University,Chennai</h2>

</div>

<img class="profile" src="pp.png" alt="" />

</div>

<ul>

<li>

<i class="fa-solid fa-phone fa-lg" style="color: #000000;"></i>

<a href="tel:+9199150746667">9150746667</a>

</li>

<li>

<i class="fa-solid fa-envelope fa-lg" style="color: #000000;"></i>

<a href="mailto:surya21110131@snuchennai.edu.in"

>surya21110199@snuchennai.edu.in</a

>

</li>

<li>

<i class="fa-brands fa-linkedin fa-lg" style="color: #000000;"></i>

<a href="https://www.linkedin.com/in/surya443" target="blank\_">LinkedIn</a>

</li>

<li>

<i class="fa-brands fa-github fa-lg" style="color: #000000;"></i>

<a href="https://www.github.com/surya443" target="blank\_">surya443</a>

</li>

</ul>

</div>

<hr />

<div class="education" id="edu-content">

<h3>Education</h3>

<ul>

<li>

<strong>Shiv Nadar University Chennai (B.Tech in Artificial Intelligence and Data Science)</strong>

<p class="highlighted-line" style="float: right;">(2021 - 2025)</p>

<ul>

<li>

<p>Activities and societies: Active Member - National Sports Organization, Member - SNUC Coding Club</p>

<ul>

</ul>

</li>

</ul>

</li>

<li>

<strong>Arsha Vidya Mandir</strong>

<p class="highlighted-line" style="float: right;">(2019 - 2021)</p>

<ul>

<li>

<p>Activities and societies: Active member - Chess Club, Drama Club</p>

</li>

</ul>

</li>

<li>

<strong>D.A.V Public School</strong>

<p class="highlighted-line" style="float: right;">(2008 - 2019)</p>

<ul>

<li>

<p>Activities and societies: Active Member of DAV MUN, Quiz, Yoga, Robotics and Physics Club - Accomplished track and field athlete, with proven track record of success in 4 X 100m relay and sprint race (2016-2019)</p>

</li>

</ul>

</li>

</ul>

</div>

<hr />

<div>

<h3>Projects </h3>

<ul>

<li>

<strong><u>CapitalCatalyst</u></strong>

<p class="highlighted-line" style="float: right;">(Dec 2022 - Jan 2023)</p>

<br>

<p><u>Technologies:</u> Streamlit, Jupiter notebook, Python, MySQL, Git</p>

<a href="https://github.com/Surya443/NIT-Hackathon2023" style="float: right;"><u>Github</u></a>

- Project for <strong>NIT Trichy Hackathon</strong> 2023<br>

- Startup Investors facilitation system using Machine Learning<br>

- Recommendation algorithm used: cosine Similarity<br>

- Vectorization of text data to ensure better prediction and recommendation<br>

- Deployed on GitHub pages via GitHub Actions

</li><br>

<li>

<strong><u>SUMIT</u></strong>

<p class="highlighted-line" style="float: right;"> (Jan 2023)</p>

<br>

<p><u>Technologies:</u> Streamlit, pytube, gTTS, Visual Code Studio, Git</p>

<a href="https://github.com/Surya443/SUM-IT" style="float: right;"><u>Github</u></a>

- Winning Hackathon Solution - ICamp<br>

- A video summarizing application using Streamlit and Whisper-OpenAI and implemented it using ChatGPT's API<br>

- Whisper architecture - end-to-end approach implemented as an encoder-decoder Transformer<br>

- Input audio is split and converted into a log-Mel spectrogram - passed into an encoder<br>

- A decoder is trained to predict the corresponding text caption, intermixed with special tokens that direct the single model to perform tasks such as language identification, phrase-level timestamps, multilingual speech transcription, and to-English speech translation

</li>

</ul>

<hr class="section-line">

</div>

<hr />

<section>

<h3>Certifications </h3>

<ul>

<li>

<a href="https://nptel.ac.in/courses/106106145"><u>NPTEL Data Structures, Algorithms Using python

(Chennai Mathematical Institute)</u></a>

</li>

<li>

<a href="https://www.linkedin.com/learning/certificates/e501fea7d67818636dc51ad17dfb017f606c3e71e817aa8194593f613f85cb2a"><u>Data

Visualization for Data Analysis and Analytics</u></a>

</li>

<li>

<a href="https://drive.google.com/file/d/13Ynfi055UHSc-YZsM8kUgR38ZRpbDbMP/view"><u>AWS DeepRacer: Driven by

Reinforcement Learning</u></a>

</li>

<li>

<a href="https://www.linkedin.com/learning/certificates/ad81086e747ca4ec4f9fba7bd6e40e79137d8aef292abe32a18f6099e29c40b1"><u>Convolution

Neural Network essential training</u></a>

</li>

<li>

<a href="https://www.udemy.com/certificate/UC-54d85610-78ca-4e60-b2ed-835b37b5f846/"><u>SQL and

PostgreSQL Developer's Guide</u></a>

</li>

<li>

<a href="https://www.cloudskillsboost.google/public\_profiles/cf2a909a-07a8-4440-9099-bce8474a7a3a"><u>Google

Cloud Badges (Generative AI, BERT, Image Captioning)</u></a>

</li>

</ul>

<hr class="section-line">

</section>

<hr />

<div class="tskills">

<h3>Technical Skills</h3>

<ul>

<li>

<span class="heading-skills">Programming Languages</span

><span>&nbsp;: C, Python, C++, Java</span>

</li>

<li>

<span class="heading-skills">Web Technologies</span

><span>&nbsp;: Html, CSS, Javascript, Flask</span>

</li>

<li>

<span class="heading-skills">Courses</span

><span>&nbsp;: Operating Systems, Data Structures, Design and Algorithms Analysis, Artificial Intelligence, Machine Learning, Networking, Databases

</li>

</ul>

</div>

<hr />

<div class="communication">

<h3>Communication</h3>

<ul>

<li>English (<em> Full professional proficiency </em>)</li>

<li>Hindi (<em> Elementary proficiency </em>)</li>

<li>Tamil (<em> Native </em>)</li>

</ul>

</div>

</div>

</body>

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

</html>

**style.css**

:root {

--light-black: #9b9999;

--medium-black: #7e7d7d;

}

html{

background:#f4f4f4;

}

.content-body {

justify-content: center;

margin: 0% auto;

max-width: 1280px;

font-family: "Lora", sans-serif;

font-weight: 300;

border: 1px solid var(--light-black);

padding: 0 2%;

}

div {

margin: 0 0 3%;

}

h1 {

font-weight: 700;

margin: 2% 0 1px;

}

h2 {

margin: 0;

}

h2,

h3 {

font-weight: 500;

}

h3{

padding-left: 2%;

}

hr {

border: 1px solid;

color: var(--medium-black);

}

table {

border-collapse: collapse;

display: flex;

justify-content: space-around;

margin-top: 30px;

}

td,

th {

border: 1px solid black;

padding: 10px 20px;

width: 105%;

}

a {

color: #4c4c7ee7;

text-decoration: none;

}

a:hover {

background-color: var(--light-black);

}

ul {

list-style: disc;

}

.header ul {

display: flex;

justify-content: space-around;

list-style: none;

margin: 0;

}

.header ul li {

display: flex;

align-items: center;

}

.header ul li i {

padding: 0 5%;

}

.header ul li:not(:first-child) {

padding: 0 1%;

/\* margin: 10%; \*/

}

.header-first {

display: flex;

flex-direction: row;

justify-content: space-between;

padding: 0;

margin: 0;

}

.header-first div {

padding: 2%;

}

.profile {

height: 70px;

width: 70px;

padding: 3%;

border-radius: 30%;

}

.heading-skills {

font-style: italic;

font-weight: 400;

}

.project .subheading {

font-size: 14px;

padding: 0px;

margin: 0 0 3px;

}

.education iframe {

display: flex;

margin: 0 auto;

border: none;

height: 350px;

width: 1000px;

}

.theme-container{

display: flex;

flex-direction: row;

align-items: center;

justify-content: flex-end;

margin: 0 15% 0 0;

}

.theme-toggle{

display: flex;

flex-direction: row;

align-items: center;

width: 4%;

border:2px solid black;

border-radius: 40px;

margin-top: 10px;

}

.theme-toggle div{

padding: 6% 0 0 18%;

}

.dark-btn-active{

background-color: #000000;

height:16px;

width: 15px;

border-radius: 50%;

}

.light-btn-active{

background-color: rgb(255, 255, 255);

border: 2px solid black;

height:16px;

width: 15px;

border-radius: 50%;

}

#custom-btn{

padding-left: 50%;

}

**src.js**

const lightBtn = document.getElementById("light-btn");

const darkBtn = document.getElementById("dark-btn");

const bodyContent = document.getElementById("body-content");

const eduBodyContent = document.getElementById("edu-content");

const toggleButton = document.getElementById("toggleBtn");

const iconList = document.getElementsByClassName("fa-lg");

const customTheme = document.getElementById('custom-btn');

let customToggle = false;

darkBtn.addEventListener("click",()=>{

darkBtn.innerHTML="<i class='fa-solid fa-moon'></i>"

lightBtn.innerHTML="<div class='light-btn-active'></div>"

bodyContent.style.backgroundColor = "#121212";

bodyContent.style.color = "#ffffff";

// eduBodyContent.style.backgroundColor = "#fff";

toggleButton.style.borderColor = "#fff"

for(let i=0;i<iconList.length;i++){

iconList[i].style.color = "#fff";

}

});

lightBtn.addEventListener("click",()=>{

darkBtn.innerHTML="<div class='dark-btn-active'></div>"

lightBtn.innerHTML="<i class='fa-solid fa-sun' style='color: #ff7800;''></i>"

bodyContent.style.backgroundColor = "#f4f4f4";

bodyContent.style.color = "#000";

toggleButton.style.borderColor = "#000"

for (let i = 0; i < iconList.length; i++) {

iconList[i].style.color = "#000";

}

});

customTheme.addEventListener("click",()=>{

if(!customToggle){

const contentBody = document.getElementsByClassName("content-body")[0];

contentBody.style.padding = "0";

contentBody.style.border = "None";

const headingList = document.getElementsByTagName("H3");

for (let i = 0; i < headingList.length; i++) {

headingList[i].style.backgroundColor = "#393E46";

headingList[i].style.color = "#EEEEEE";

}

const headerFirst = document.getElementsByClassName("header")[0];

headerFirst.style.color = "#2f6f72";

headerFirst.style.height = "150%";

headerFirst.style.backgroundColor = "#222831";

headerFirst.margin = "0";

const Img = document.getElementsByTagName("IMG")[0];

Img.style.backgroundColor = "#EEEEEE";

Img.style.borderRadius = "5%";

Img.style.padding = "3%";

Img.style.margin = "1% 2%";

for (let i = 0; i < iconList.length; i++) {

iconList[i].style.color = "#EEEEEE";

}

const linksList = document.getElementsByTagName("A");

for (let i = 0; i < linksList.length; i++) {

linksList[i].style.color = "#8162d672";

}

customToggle = true;

}

else{

const contentBody = document.getElementsByClassName("content-body")[0];

contentBody.style.padding = "0 2%";

contentBody.style.border = "1px solid #9b9999";

const headingList = document.getElementsByTagName("H3");

for (let i = 0; i < headingList.length; i++) {

headingList[i].style.backgroundColor = "#f4f4f4";

headingList[i].style.color = "#000";

}

const headerFirst = document.getElementsByClassName("header")[0];

headerFirst.style.color = "#000";

headerFirst.style.height = "100%";

headerFirst.style.backgroundColor = "#f4f4f4";

headerFirst.margin = "0";

const Img = document.getElementsByTagName("IMG")[0];

Img.style.backgroundColor = "#EEEEEE";

Img.style.borderRadius = "30%";

Img.style.padding = "3%";

Img.style.margin = "0";

for (let i = 0; i < iconList.length; i++) {

iconList[i].style.color = "#000";

}

const linksList = document.getElementsByTagName("A");

for (let i = 0; i < linksList.length; i++) {

linksList[i].style.color = "#4c4c7ee7";

}

customToggle = false;

}

});

window.addEventListener("load",()=>{

darkBtn.innerHTML = "<div class='dark-btn-active'></div>"

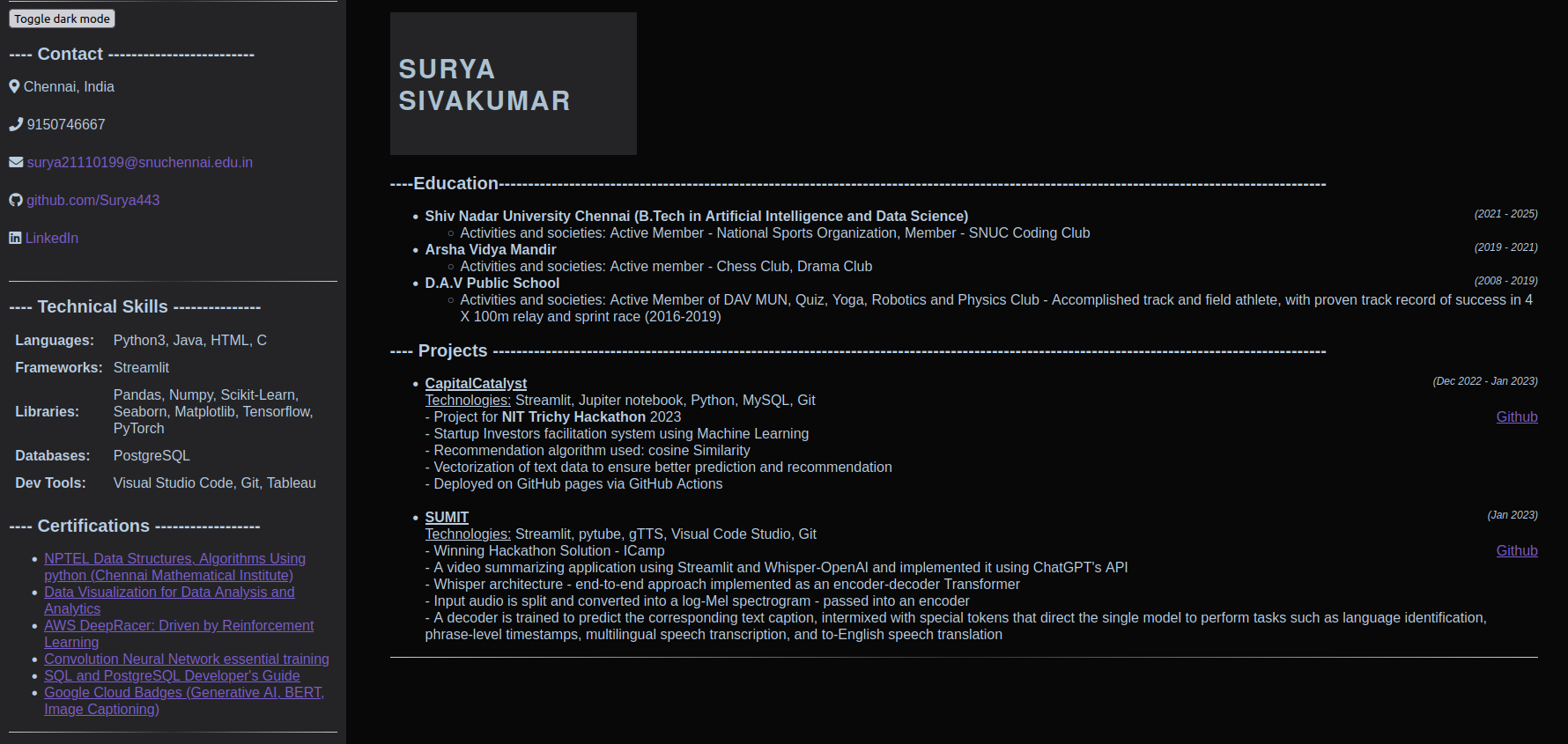
});

**Output:**

Light Theme:



Dark theme:

****

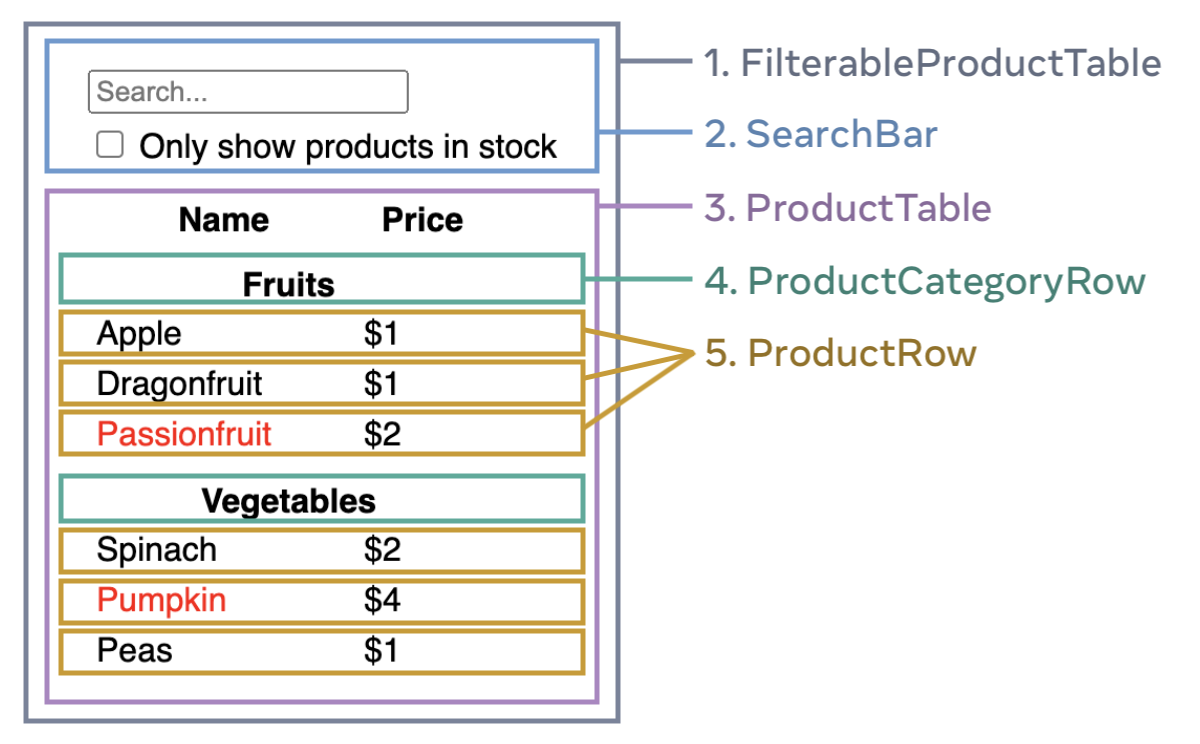
**Result:**

Successfully implemented a CSS with light theme and dark theme toggle button using javascript

| **Ex. No: 4** | **Search and Filter using React** |
| --- | --- |
| **7/8/23** |

**Aim:** Implement the search and filter application using react.

**Mockup:**

****

**Program :**

Components:

1. **filterTableProduct.js:**

import React, { useState } from 'react';

import SearchBar from './SearchBar';

import ProductTable from './ProductTable';

function FilterableProductTable({ products }) {

const [filterText, setFilterText] = useState('');

const [inStockOnly, setInStockOnly] = useState(false);

return (

<div>

<SearchBar

filterText={filterText}

inStockOnly={inStockOnly}

onFilterTextChange={setFilterText}

onInStockOnlyChange={setInStockOnly} />

<ProductTable

products={products}

filterText={filterText}

inStockOnly={inStockOnly} />

</div>

);

}

export default FilterableProductTable;

1. **ProductCategoryRow.js**

import React from 'react';

function ProductCategoryRow({ category }) {

return (

<tr>

<th colSpan="2">

{category}

</th>

</tr>

);

}

export default ProductCategoryRow;

**3. ProductRow.js**

import React from 'react';

function ProductRow({ product }) {

const name = product.stocked ? product.name :

<span style={{ color: 'red' }}>

{product.name}

</span>;

return (

<tr>

<td>{name}</td>

<td>{product.price}</td>

</tr>

);

}

export default ProductRow;

**4. ProductTable.js**

import React from 'react';

import ProductCategoryRow from './ProductCategoryRow';

import ProductRow from './ProductRow';

function ProductTable({ products, filterText, inStockOnly }) {

const rows = [];

let lastCategory = null;

products.forEach((product) => {

if (

product.name.toLowerCase().indexOf(

filterText.toLowerCase()

) === -1

) {

return;

}

if (inStockOnly && !product.stocked) {

return;

}

if (product.category !== lastCategory) {

rows.push(

<ProductCategoryRow

category={product.category}

key={product.category} />

);

}

rows.push(

<ProductRow

product={product}

key={product.name} />

);

lastCategory = product.category;

});

return (

<table>

<thead>

<tr>

<th>Name</th>

<th>Price</th>

</tr>

</thead>

<tbody>{rows}</tbody>

</table>

);

}

export default ProductTable;

**5. SearchBar.js**

import React from 'react';

import '../styles.css';

function SearchBar({

filterText,

inStockOnly,

onFilterTextChange,

onInStockOnlyChange

}) {

return (

<div>

<form>

<input

type="text"

value={filterText} placeholder="Search..."

onChange={(e) => onFilterTextChange(e.target.value)} />

<label>

<input

type="checkbox"

checked={inStockOnly}

onChange={(e) => onInStockOnlyChange(e.target.checked)} />

{' '}

Only show products in stock

</label>

</form>

</div>

);

}

export default SearchBar;

**app.js**

import React from 'react';

import FilterableProductTable from './components/FilterableProductTable';

const PRODUCTS = [

{category: "Fruits", price: "$1", stocked: true, name: "Apple"},

{category: "Fruits", price: "$1", stocked: true, name: "Dragonfruit"},

{category: "Fruits", price: "$2", stocked: false, name: "Passionfruit"},

{category: "Vegetables", price: "$2", stocked: true, name: "Spinach"},

{category: "Vegetables", price: "$4", stocked: false, name: "Pumpkin"},

{category: "Vegetables", price: "$1", stocked: true, name: "Peas"}

];

function App() {

return <FilterableProductTable products={PRODUCTS} />;

}

export default App;

**index.js**

import React from 'react';

import ReactDOM from 'react-dom/client';

import App from './App';

import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

<React.StrictMode>

<App />

</React.StrictMode>

);

reportWebVitals();

**reportWebVitals.js**

const reportWebVitals = onPerfEntry => {

if (onPerfEntry && onPerfEntry instanceof Function) {

import('web-vitals').then(({ getCLS, getFID, getFCP, getLCP, getTTFB }) => {

getCLS(onPerfEntry);

getFID(onPerfEntry);

getFCP(onPerfEntry);

getLCP(onPerfEntry);

getTTFB(onPerfEntry);

});

}

};

export default reportWebVitals;

**style.css**

.form-container {

margin-bottom: 400px;

border-bottom: 10px solid black;

}

/\* ProductTable component styles \*/

table {

width: 100%;

border-collapse: collapse;

background-color: antiquewhite;

}

th, td {

padding: 8px;

text-align: left;

border-bottom: 1px solid #1e1919;

}

th {

background-color: #f2f2f2;

}

/\* ProductRow component styles \*/

.red-text {

color: red;

}

input {

border-radius: 5px;

align-self: start;

}

tbody tr:nth-child(odd) {

background: #1b61a9;

}

tbody tr:nth-child(even) {

background: #eab716;

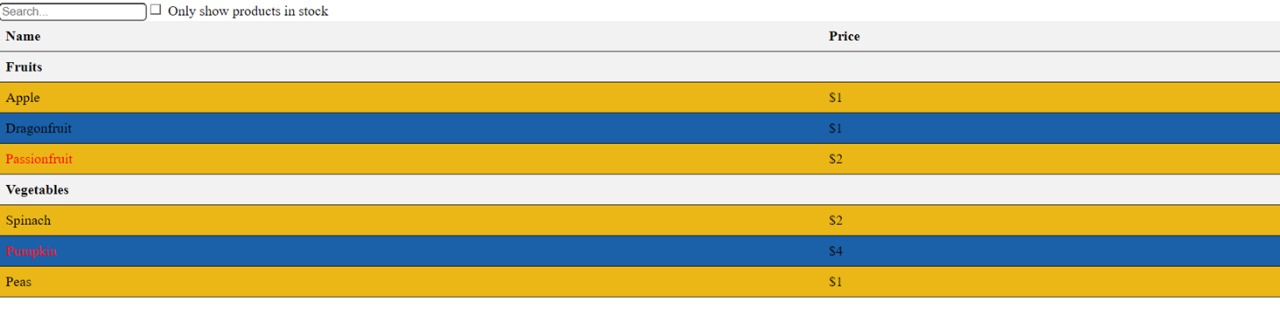
}

caption {

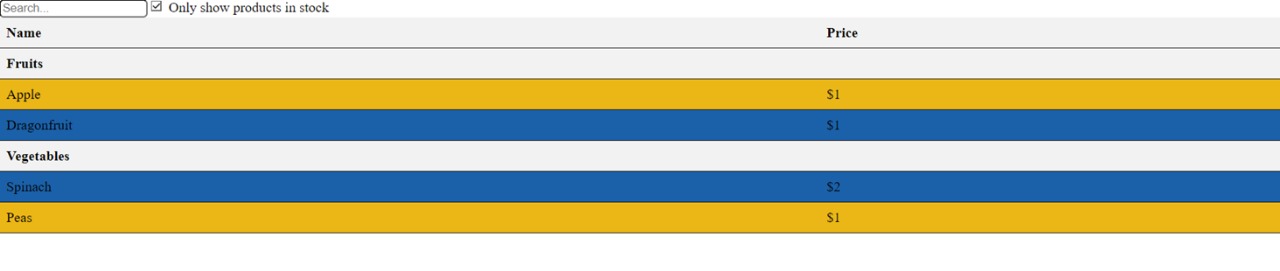
font-size: 0.8rem;

}

**Output:**

****

****

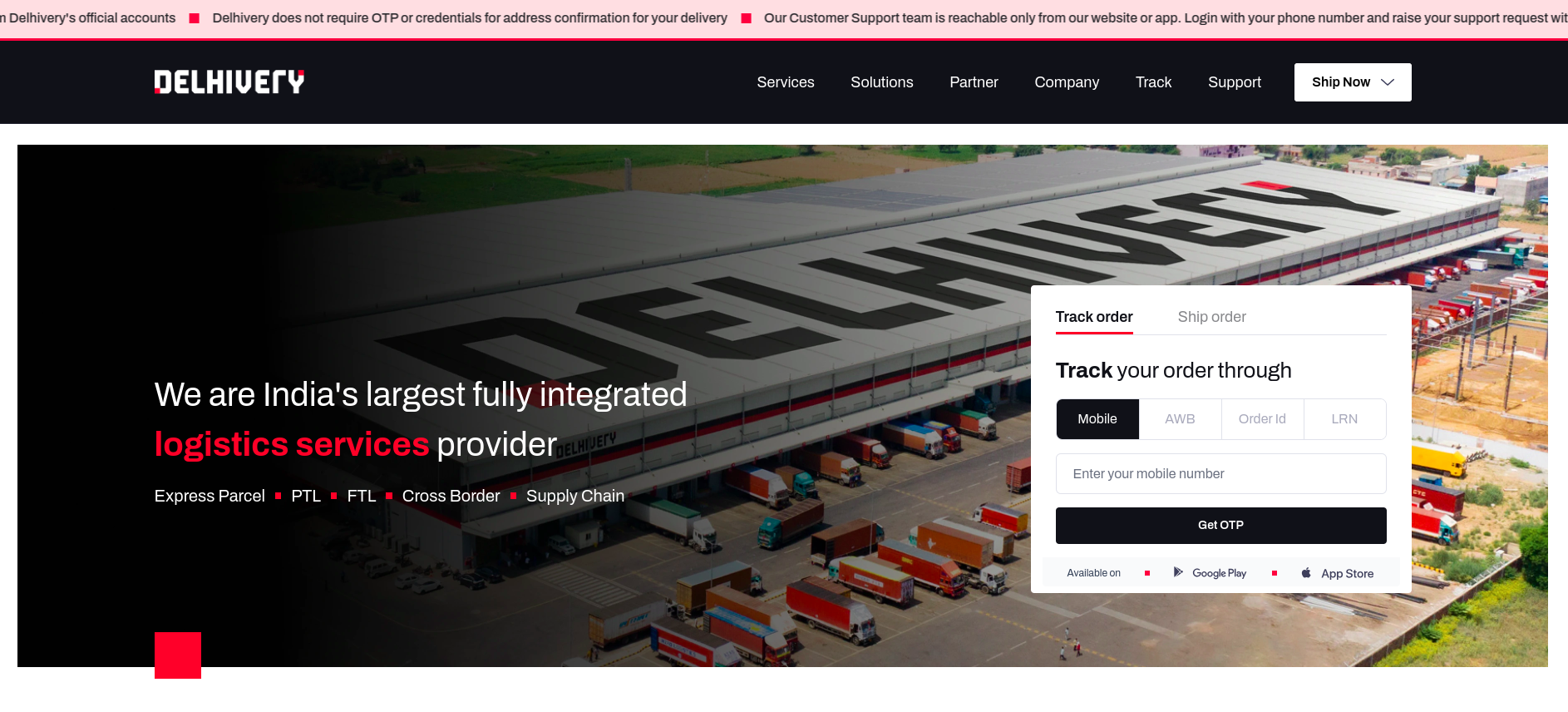
****

**Result:** Thus search and filter functions in the react application have been successfully executed.

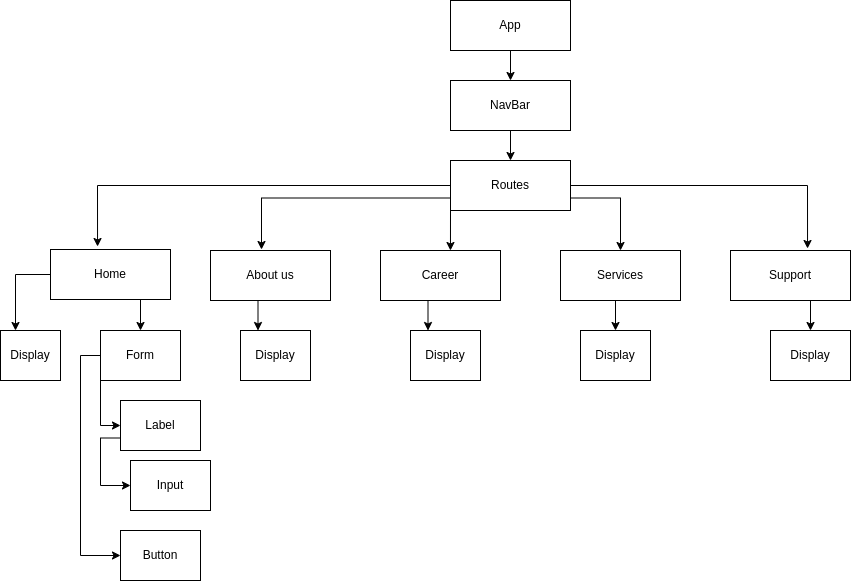
| **Ex. No: 5** | **React - Menu Navigation** |
| --- | --- |
| **21/8/23** |

**Aim:** To write a program using React.js that focuses on Menu Navigation for a Nifty 200 company.

**Mockup:**

****

**Component Hierarchy:**



**Program :**

**App.js**

import Navbar from "./Navbar"

import Home from "./pages/Home"

import Company from "./pages/Company"

import Solutions from "./pages/Solutions"

import Services from "./pages/Services"

import Partner from "./pages/Partner"

import Support from "./pages/Support"

import Track from "./pages/Track"

import {Route, Routes} from "react-router-dom"

function App() {

return (<>

<Navbar />

<div className="container">

<Routes>

<Route path="/" element={<Home/>} />

<Route path="/services" element={<Services/>} />

<Route path="/solutions" element={<Solutions/>} />

<Route path="/partner" element={<Partner/>} />

<Route path="/company" element={<Company/>} />

<Route path="/track" element={<Track/>} />

<Route path="/support" element={<Support/>} />

</Routes>

</div>

</>)

}

**Navbar.js**

import {Link, useMatch, useResolvedPath} from "react-router-dom"

export default function Navbar(){

return <nav className="nav">

<Link to="/" className="site-title">

<div className="logo-container">

<img src="https://www.delhivery.com/\_nuxt/img/Delhivery\_logo.dcef706.png" className="logo"/>

</div></Link>

<ul>

<CustomLink to="/services">Services</CustomLink>

<CustomLink to="/solutions">Solutions</CustomLink>

<CustomLink to="/partner">Partner</CustomLink>

<CustomLink to="/company">Company</CustomLink>

<CustomLink to="/track">Track</CustomLink>

<CustomLink to="/support">Support</CustomLink>

</ul>

</nav>

}

function CustomLink({to,children, ...props}){

const resolvedPath = useResolvedPath(to)

const isActive = useMatch({path : resolvedPath.pathname, end: true})

return (

<li className={isActive? "active" : ""}>

<Link to={to} {...props}>

{children}

</Link>

</li>

)

}

export default App;

**Pages/**

**Home.js**

export default function Home(){

return (<>

<div className="main-home-content" style={{ backgroundImage: `url("https://www.delhivery.com/\_nuxt/img/homepage.ca1e7bc.webp")` }}>

{/\* <img src={background} height="40%" width="100%" className="homepage-img"></img> \*/}

<div className="home-element">

<p>We are India's largest fully integrated <br/> <strong className='red'>logistics services</strong> provider</p><br/>

<ul className="points">

<li>Express Parcel</li>

<li>PTL</li>

<li>FTL</li>

<li>Cross Border Supply Chain</li>

</ul>

</div>

<div className="form" >

{/\* <p className="p1">Track Order</p> \*/}

<p>

<b>Track</b> your order

</p><br/><br/>

<form className="trackfrom">

<label>

<input className="number" name="Number" value="Enter your mobile number" />

</label>

<br/>

<button className="submit" type="submit">Submit</button>

</form>

</div>

</div>

</>)

}

**Services.js**

export default function Services(){

return (<>

<div className="main-content" style={{ backgroundImage: `url(" https://www.delhivery.com/\_nuxt/img/dataIntelligence.dff1d63.webp")` }}>

{/\* <img src={background} height="40%" width="100%" className="homepage-img"></img> \*/}

<div className="element">

<p><strong className='red'>Boost your Business</strong> with India's

<br/> most accurate AI powered location stack</p><br/>

<ul className="points" style={{paddingTop: '0px', marginTop: '0px'}}>

Improve delivery success with deep insights from user-generated addresses

</ul>

</div>

</div>

</>)

}

**Solutions.js**

export default function Solutions(){

return (<>

<div className="main-content" style={{ backgroundImage: `url("https://www.delhivery.com/\_nuxt/img/d2c.24360ef.webp")` }}>

{/\* <img src={background} height="40%" width="100%" className="homepage-img"></img> \*/}

<div className="element">

<p>Grow your <strong className='red'> Direct To Consumer brand</strong> with

<br/> Delhivery’s end-to-end logistics solutions</p><br/>

<ul className="points" style={{paddingTop: '0px', marginTop: '0px'}}>

Faster deliveries, lower returns and improved consumer

experience that drives your brand's success

</ul>

</div>

</div>

</>)

}

**Partner.js**

export default function Partner(){

return (<>

<div className="main-content" style={{ backgroundImage: `url("https://www.delhivery.com/\_nuxt/img/Constellation.47ffc3d.webp")` }}>

{/\* <img src={background} height="40%" width="100%" className="homepage-img"></img> \*/}

<div className="element">

<p>Join India's <strong className='red'> largest integrated logistics </strong> provider

</p><br/>

<ul className="points" style={{paddingTop: '0px', marginTop: '0px'}}>

Register as a partner and provide local expertise to extend

Delhivery’s reach across India

</ul>

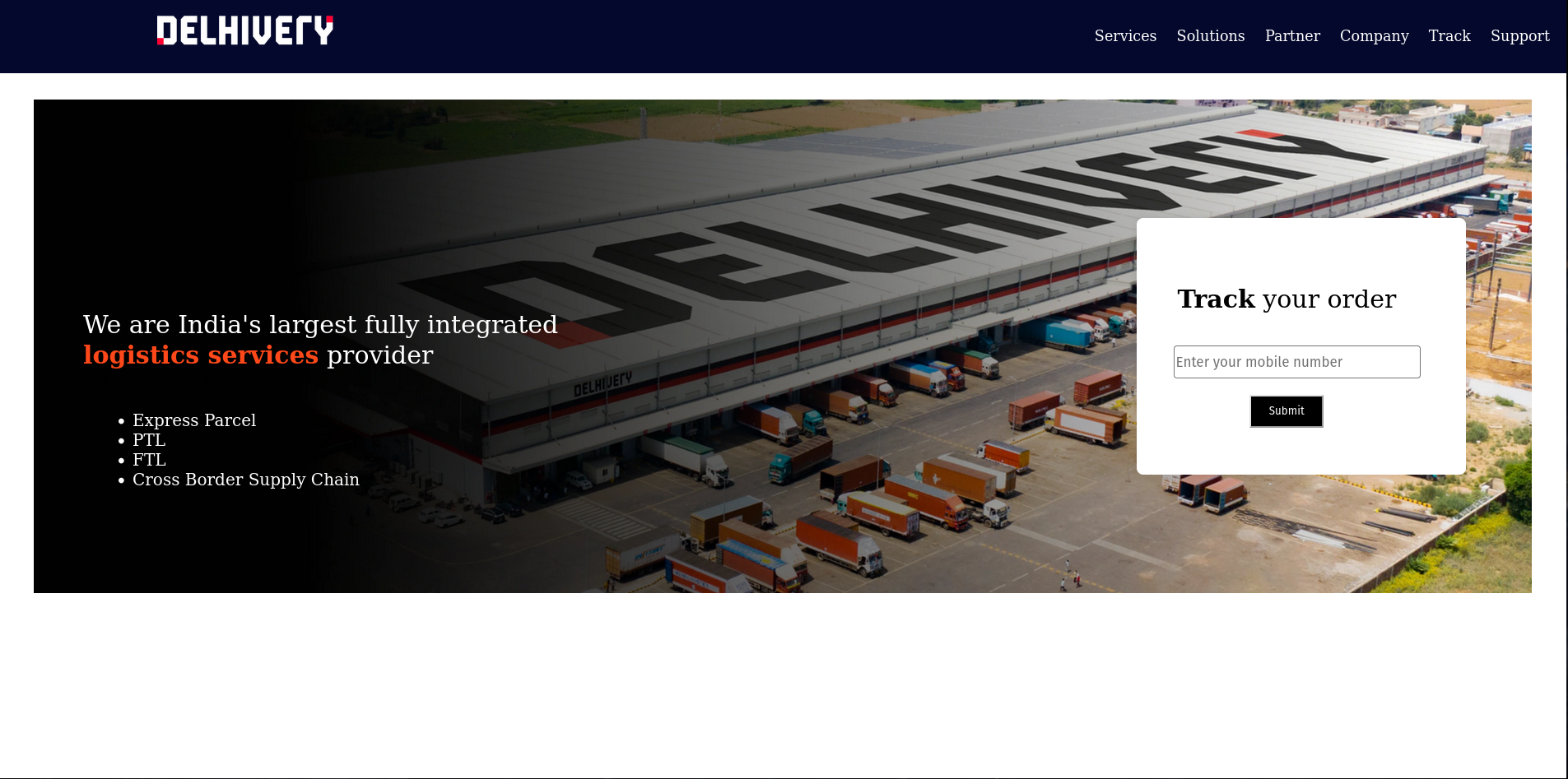
</div>

</div>

</>)

}

**Output:**

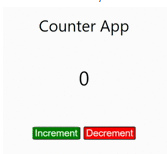
****

**Result:** The React App with Menu Navigation using routers has been implemented successfully.

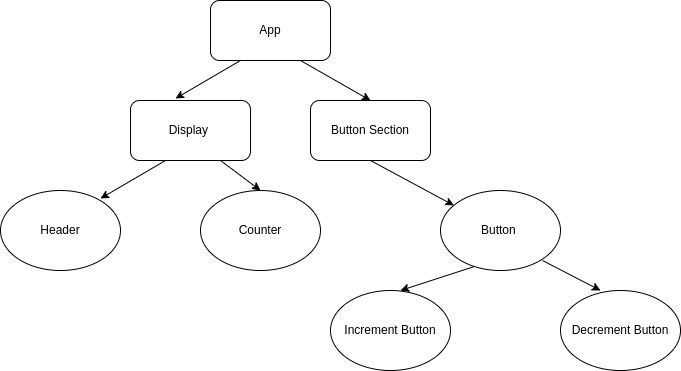
| **Ex. No: 6** | **React- Simple Applications** |
| --- | --- |
| **28/7/23** |

**Aim:** Implement a react counter app as below.

**Mockup:**

****

**Component Hierarchy Diagram:**

****

.

**Program :**

**App.js:**

import logo from './logo.svg';

import './App.css';

import Display from './components/display';

import ButtonSection from './components/buttonsection';

import { useState } from 'react';

import Stack from 'react-bootstrap/Stack'

function App() {

let [count,setCount] = useState(0);

const incrementCount = () =>{

setCount(count+1);

}

const decrementCount = () =>{

setCount(count-1);

}

return (

<div className="App">

<Stack direction='vertical' className='container'>

<Display className="header" text="Counter App"></Display>

<Display className="counter" text={count}></Display>

<ButtonSection className='buttonSection' increment={incrementCount}

decrement={decrementCount} c={count}></ButtonSection>

</Stack>

</div>

);

}

export default App;

**button.js**

import {Button} from "react-bootstrap";

function Buttonboot(props){

return(

<Button variant={props.color} onClick={props.onClick}

disabled={props.flag}>{props.text}</Button>

)

}

export default Buttonboot;

**buttonSection.js**

import Buttonboot from "./button";

import {Stack} from "react-bootstrap";

function ButtonSection(props){

return(

<div className={props.className}>

<Stack direction="horizontal" gap={3}>

<Buttonboot color="success" text="Increment"

onClick={props.increment} ></Buttonboot>

<Buttonboot color="danger" text="Decrement"

onClick={props.decrement} flag={props.c==0? 1:0}></Buttonboot>

</Stack>

</div>

)

}

export default ButtonSection;

**display.js**

import Buttonboot from "./button";

function Display(props){

return(

<div className={props.className}>

<h1>{props.text}</h1>

</div>

)

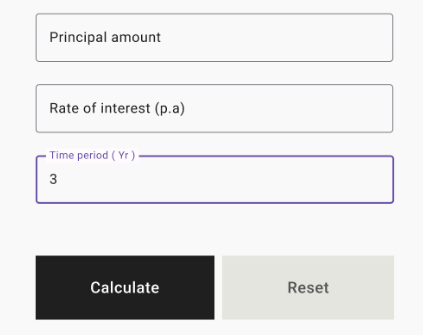
}

export default Display;

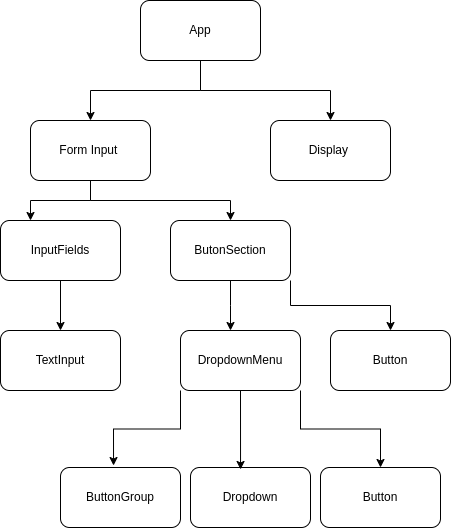
**Aim:**

Implement a simple and compound interest calculator as below.

**Mockup:**

****

**Component Hierarchy Diagram:**

****

**Program:**

**App.js**

import logo from './logo.svg';

import './App.css';

import { useState,useEffect } from 'react';

import Display from './components/display';

import FormInput from './components/form';

function App() {

let [resultMessage,setResultMessage] = useState('');

let [displayStatus,setDisplayStatus] = useState('true');

return (

<div className="App">

<Display display={true} resultMessage = {"Simple and Compound Interest

Calculator"}></Display>

<div className='wrapper'>

<FormInput className={"form-box"} setResult={setResultMessage}

setDisplayStatus = {setDisplayStatus}

displayStatus={displayStatus}></FormInput>

</div>

<Display display = {displayStatus} resultMessage={resultMessage}></Display>

</div>

);

}

export default App;

**ButtonSection.js**

import Button from "react-bootstrap/Button";

import DropdownMenu from "./dropdownbutton";

import Stack from "react-bootstrap/Stack";

import React, { useEffect } from "react";

function ButtonSection(props){

const handleReset = () =>{

props.setDisplayStatus(false);

}

return(

<Stack direction="horizontal" gap={3} className="button-section">

<DropdownMenu formData = {props.formData} setFormData =

onSubmit= {props.onSubmit}></DropdownMenu>

<Button variant="secondary" type="reset" onClick={handleReset}>Reset</Button>

</Stack>

)

}

export default ButtonSection;

**display.js**

function Display(props){

const displayStyle = {

display: props.display ? 'block' : 'none', // Set display property based

on the display prop

};

return(

<div className="result" style={displayStyle}>

<h1>{props.resultMessage}</h1>

</div>

)

}

export default Display;

**dropdownbutton.js**

import Button from 'react-bootstrap/Button';

import ButtonGroup from 'react-bootstrap/ButtonGroup';

import Dropdown from 'react-bootstrap/Dropdown';

function DropdownMenu(props) {

const setSimpleInterest = (e) =>{

e.preventDefault();

props.setFormData({...props.formData,interestFunction:1});

console.log(props.formData.interestFunction)

}

function setCompoundInterest(e){

e.preventDefault();

props.setFormData({...props.formData,interestFunction:2});

console.log(props.formData.interestFunction)

}

return(

<Dropdown as={ButtonGroup}>

<Button type='submit' variant='primary'

onClick={props.onSubmit}>Calculate</Button>

<Dropdown.Toggle split variant='primary'

id='dropdown-split-basic'></Dropdown.Toggle>

<Dropdown.Menu>

<Dropdown.Item as="button" onClick={setSimpleInterest}>

Simple Interest</Dropdown.Item>

<Dropdown.Item as = "button"

onClick={setCompoundInterest}>Compound

Interest</Dropdown.Item>

</Dropdown.Menu>

</Dropdown>

)

}

export default DropdownMenu;

**form.js**

import InputFields from "./inputfields";

import ButtonSection from "./buttonsection";

import { useState,useEffect } from "react";

import Form from 'react-bootstrap/Form';

function FormInput(props){

let [formData, setFormData] = useState({

// Initialize form fields with default values

principal: '',

rate: '',

years:'',

interestFunction:1

});

const calculateSimpleInterest = (p,r,n) =>{

let result = (p\*n\*r)/100;

console.log(result)

props.setResult("The simple interest is: "+result);

}

const calculateCompoundInterest = (p,r,n) =>{

r = r/100;

let result = p\*((1+r)\*\*n);

console.log(p,r,n)

console.log(result)

props.setResult("The compound interest is: "+result);

}

const calculate = (e) => {

e.preventDefault();

console.log('calculate');

console.log(formData)

let {principal,rate,years} = {...formData}

if (formData.interestFunction===1){

calculateSimpleInterest(principal,rate,years)

}

else if (formData.interestFunction===2){

calculateCompoundInterest(principal,rate,years)

}

props.setDisplayStatus('true')

}

return (

<Form className={props.className}>

<InputFields formData = {formData} setFormData = {setFormData}></InputFields>

<ButtonSection formData = {formData} setFormData = {setFormData} onSubmit={calculate} setDisplayStatus = {props.setDisplayStatus}></ButtonSection>

</Form>

)

}

export default FormInput;

**inputFields.js**

import TextInput from './textinput';

import Stack from 'react-bootstrap/Stack';

function InputFields(props) {

const handleInputChange = (e) =>{

// e.preventDefault();

let id = e.target.id;

let value = e.target.value;

props.setFormData({...props.formData,[id]:value});

}

var {principal,rate,years} = props.formData

return(

<Stack className='inputFields' gap={3}>

<TextInput label="Principal amount" value={principal} id={"principal"} onChange={handleInputChange}></TextInput>

<TextInput label="Rate of interest (p.a)" id={"rate"} value = {props.formData.rate} onChange={handleInputChange}></TextInput>

<TextInput label="Time period(Yr)" id={"years"} onChange={handleInputChange} value={props.formData.years}></TextInput>

</Stack>

)

}

export default InputFields;

**textinput.js**

import FloatingLabel from 'react-bootstrap/FloatingLabel';

import Form from 'react-bootstrap/Form';

function TextInput(props){

return(

<FloatingLabel

controlId={props.id}

label={props.label}

className="text-input"

value = {props.value}

>

<Form.Control type="number" onChange={props.onChange}/>

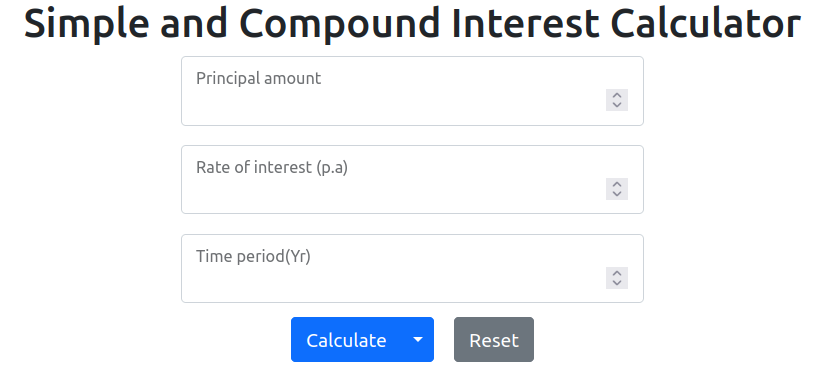
</FloatingLabel>

)

}

export default TextInput;

**Output:**

****

**Result:**

Thus, simple react applications were written and studied.

| **Ex. No: 7** | React - Real Time Form Validation, and Submission Validation |
| --- | --- |
| **4/9/23** |

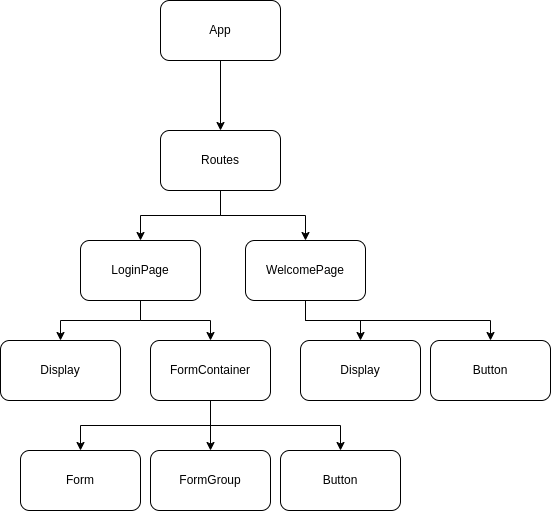
**Aim:** Implement a react app as below, (K4)

**Mockup:**

****

.

**Component Hierarchy Diagram :**



**Program:**

**App.js**

import logo from './logo.svg';

import './App.css';

import LoginPage from './components/LoginPage';

import WelcomePage from './components/WelcomePage';

import { Routes, Route } from 'react-router-dom';

import { useState } from 'react';

function App() {

let [validation,setValidation] = useState('false');

return (

<div className="App">

<Routes>

<Route path="/" Component={LoginPage} index={true}></Route>

<Route path='/welcome' Component={WelcomePage}/>

</Routes>

</div>

);

}

export default App;

**Display.js**

function Display(props){

return(

<h1 style={{ userSelect: 'none',...(props.displayStyle || {})}} >{props.message}</h1>

)

}

export default Display;

**FormContainer.js**

import Display from "./Display"

import FormContainer from "./FormContainer";

import { useState } from "react";

function LoginPage(){

let[flag,setFlag] = useState(true)

let displayStyle = {display: flag? "none":"block",

flexShrink:0};

return(

<div className='login-page'>

<Display message={"Login"}></Display>

<FormContainer setFlag = {setFlag}></FormContainer>

<Display message={'Sorry! Your name and password do not match!'} displayStyle = {displayStyle}></Display>

</div>

)

}

export default LoginPage;

**WelcomePage.js**

import Display from "./Display";

import { Button } from "react-bootstrap";

import { useNavigate } from "react-router-dom";

function WelcomePage(){

let navigate=useNavigate();

const handleClick = (e) =>{

e.preventDefault();

navigate("/")

}

return(

<div className="welcome-page">

<Display message={'Welcome!'}></Display>

<Button variant="danger" onClick={handleClick}>Logout</Button>

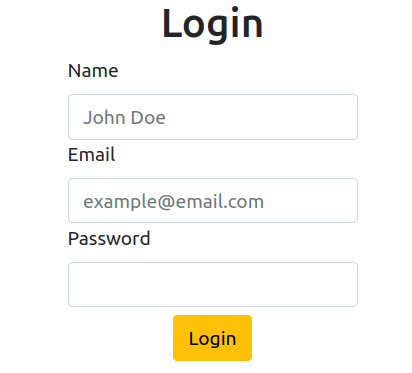
</div>

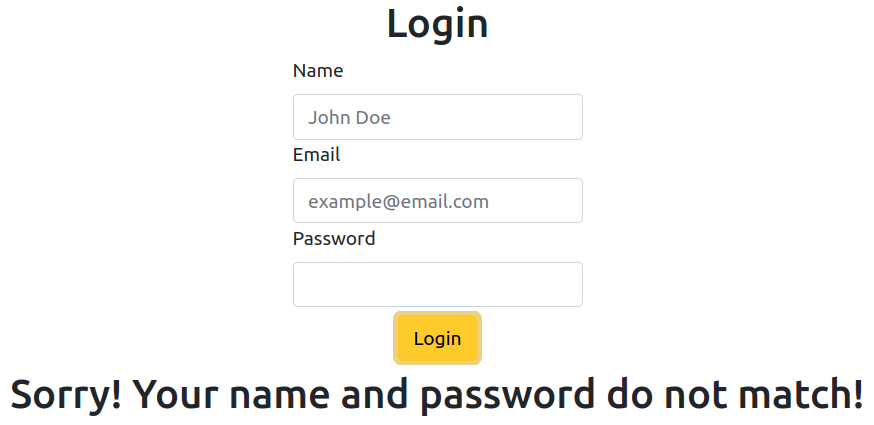
)

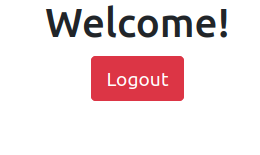
}

export default WelcomePage;

**Output:**

****

****

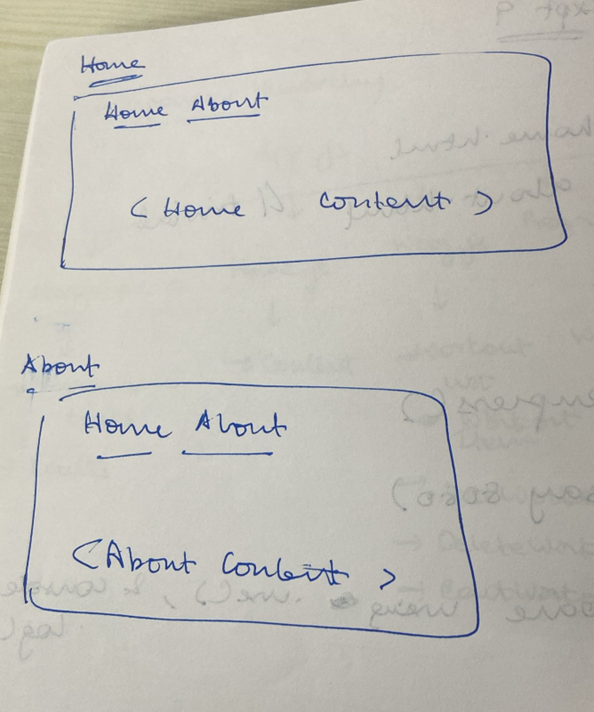
****

**Result:** Thus, form validation in react was studied and implemented.

| **Ex. No: 8** | **Node JS** |
| --- | --- |
| **9/10/23** |

**Aim:** To create a Node JS server that has 2 pages - Homepage, and About and to implement proper error handling for invalid routes (e.g., return a 404 status code for unknown routes), and also, Implement a middleware to log incoming requests to the console.

**MockUp:**

****

**Algorithm:**

1. Import necessary modules: http, fs, path.

2. Construct file path based on request URL (filePath).

3. If URL is root or index.html, set path to public/index.html.

4. If URL is about.html, set path to public/about.html.

5. Read file asynchronously.

6. If file not found (error), return 404 response.

7. Determine file extension and set content type accordingly.

8. Write 200 OK header with appropriate content type.

9. Send file content as response using binary encoding.

10. Create an HTTP server instance using http.createServer().

11. Handle server errors and log error messages.

12. Server listens on port 3000.

13. Pass request and response objects to handleRequest function.

14. If requested file not found, respond with 404 status code.

15. Use appropriate content type for valid files (text/html, images).

16. Serve index.html for root and about.html for /about.html URL.

17. Access the server at http://localhost:3000/ for Homepage.

18. Access the server at http://localhost:3000/about.html for About page.

19. Invalid routes will return a 404 status code as expected.

.

**Program :**

***Main.js***

const http = require('http');

const fs = require('fs');

const path = require('path');

function handleRequest(req, res) {

let filePath = '.' + req.url;

if (filePath == './' || filePath == './index.html') {

filePath = './public/index.html'; // Default to index.html for the root URL

}

else if (filePath == "./about.html") {

filePath = './public/about.html';

}

fs.readFile(filePath, (err, data) => {

if (err) {

res.writeHead(404, { 'Content-Type': 'text/html/css' });

res.end('<h1>404 Not Found</h1>');

}

else {

const extname = path.extname(filePath);

let contentType = 'text/html';

res.writeHead(200, { 'Content-Type': contentType });

res.end(data, 'binary');

}

});

}

const server = http.createServer((req, res) => {

console.log('Request received:', req.method, req.url);

handleRequest(req, res);

});

server.on('error', (error) => {

console.error('Server error:', error.message);

});

server.listen(3000, () => {

console.log('Server is listening on port 3000 - localhost:3000/');

});

***index.html***

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

</head>

<body>

<a href="index.html">Home</a>

<a href="about.html">About</a>

<img src="./public/images/home1.png" name="Home1">

<img src="./public/images/home2.png" name="Home2">

</body>

</html>

***about.html***

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

\* {

background-color: white;

}

h1 {

color: red;

}

p {

border: 5px solid black;

padding: 20px;

box-shadow: inset;

}

</style>

</head>

<body>

<a href="index.html">Home</a>

<a href="about.html">About</a>

<h1>About Arsenal F.C</h1>

<p>Arsenal Football Club is an English professional football club based in Holloway, North London. Arsenal compete in the Premier League, the top flight of English football. The club has won 13 league titles (including one unbeaten title), a record 14 FA Cups, two League Cups, 17 FA Community Shields, the Football League Centenary Trophy, one European Cup Winners' Cup and one Inter-Cities Fairs Cup. In terms of trophies won, it is the third-most successful club in English football.

Arsenal was the first club from the South of England to join the Football League in 1893,

and they reached the First Division in 1904. Relegated only once, in 1913, they continue the longest streak in the top division,

[2] and have won the second-most top-flight matches in English football history.[3] In the 1930s, Arsenal won five League Championships

and two FA Cups, and another FA Cup and two Championships after the war. In 1970–71, they won their first League and FA Cup Double.

Between 1989 and 2005, they won five League titles and five FA Cups, including two more Doubles.

They completed the 20th century with the highest average league position.[4] Between 1998 and 2017, Arsenal qualified for the UEFA Champions League for nineteen consecutive seasons.

In 1886, munitions workers at the Royal Arsenal in Woolwich founded the club as Dial Square. In 1913 the club crossed the city to Arsenal Stadium in Highbury, becoming close neighbours of Tottenham Hotspur, and creating the North London derby.

Herbert Chapman, who changed the fortunes of Arsenal forever, won the club its first silverware, and his legacy led the club to dominate the 1930s.

He helped introduce the WM formation, floodlights, and shirt numbers;[5] he also added the white sleeves and brighter red to the club's jersey.[6]

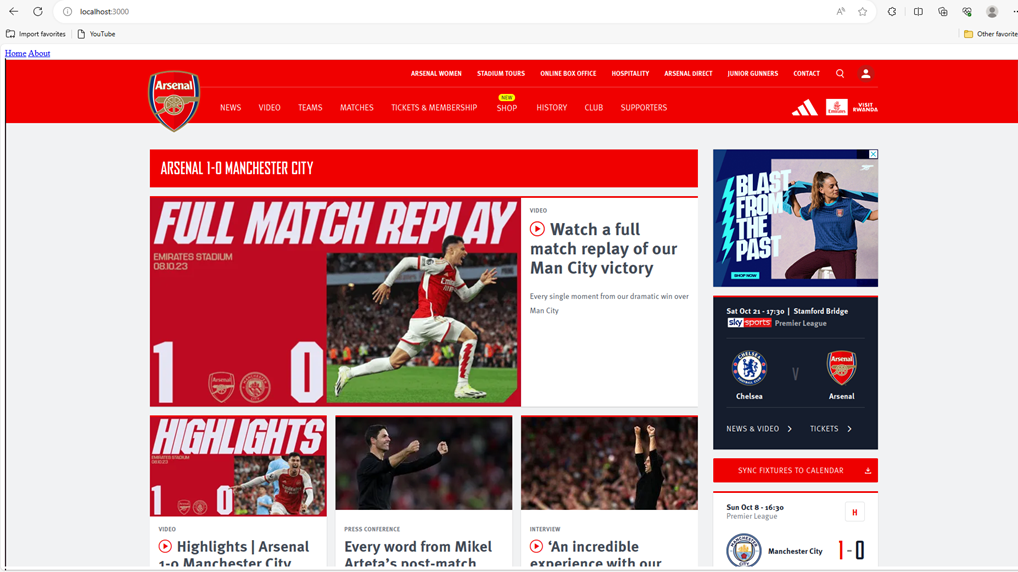
Arsène Wenger is the club's longest-serving manager and has won the most trophies for it. He won a record seven FA Cups, and his

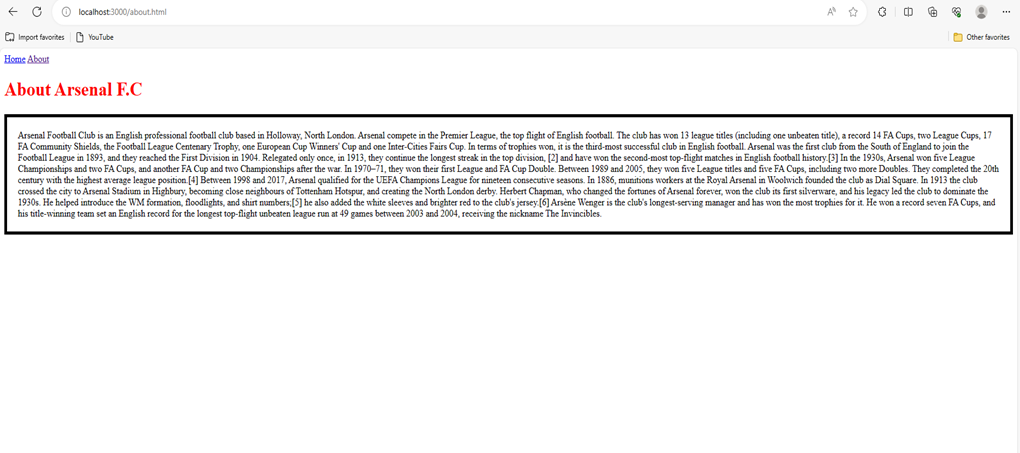
title-winning team set an English record for the longest top-flight unbeaten league run at 49 games between 2003 and 2004, receiving the nickname The Invincibles. </p>

</body>

</html>

**Output Screenshots:**







**Result:** Using Node JS, a web server of a website, containing Home and About pages were successfully implemented and errors were successfully handled.

| **Ex. No: 9** | **MERN STACK - 1** |
| --- | --- |
| **16/10/2023** |

**Aim :** To learn and implement the MERN stack based on the provided documentation and convert one of the previous React and Node.js projects into a MERN application

**Design :**

**Algorithm :**

1. Start the application.

2. Initialize the state in the client-side React component, including an array called todos to store tasks and a string called newTodo to store the text of the new task.

3. In the componentDidMount lifecycle method, send a GET request to the server to fetch existing tasks from the MongoDB database.

4. Handle the response from the server by updating the todos state with the retrieved data.

5. Create two methods, handleInputChange and handleSubmit, to handle user input for adding new tasks.

6. In the handleInputChange method, update the newTodo state as the user types.

7. In the handleSubmit method, check if the input is empty and return if so.

8. If the input is not empty, create a new task object with the task and completed properties.

9. Send a POST request to the server to add the new task to the database.

10. Update the state with the newly added task and reset the newTodo input field.

11. Render the UI, displaying the task list, input field, and a button to add tasks.

12. Map through the todos array to display each task in the list.

**Frontend Components (React) :**

1. App.js component for the main application.

2. Task List component to display the list of tasks.

3. Task Item component to represent individual tasks.

4. Task Form component for adding new tasks.

**Frontend Functionality (React) :**

1. Fetching tasks from the backend.

2. Adding a new task to the list.

3. Handling user input for tasks.

**Backend (Node.js with Express):**

1. Defining API endpoints for fetching and adding tasks.

2. Connecting to the MongoDB database.

**Database (MongoDB) :**

1. Creating a schema for tasks and a model.

2. Storing task data with the Todo schema.

3. Performing CRUD operations for tasks.

**Program :**

**Client – App.js**

import React, { Component } from 'react';

import axios from 'axios';

class App extends Component {

constructor(props) {

super(props);

this.state = {

todos: [],

newTodo: '',

};

}

componentDidMount() {

axios.get('http://127.0.0.1:5000/todos')

.then(response => {

this.setState({ todos: response.data });

})

.catch(function (error) {

console.log(error);

});

}

handleInputChange = (e) => {

this.setState({ newTodo: e.target.value });

};

handleSubmit = (e) => {

e.preventDefault();

if (this.state.newTodo.trim() === '') {

return;

}

const newTask = {

task: this.state.newTodo,

completed: false,

};

axios

.post('http://127.0.0.1:5000/todos/add', newTask)

.then((response) => {

console.log(response.data);

this.setState({ newTodo: '', todos: [...this.state.todos, newTask] });

})

.catch((error) => {

console.log(error);

});

};

render() {

return (

<div>

<h1>To-Do List</h1>

<form onSubmit={this.handleSubmit}>

<input

type="text"

placeholder="New Task"

value={this.state.newTodo}

onChange={this.handleInputChange}

/>

<button type="submit">Add Task</button>

</form>

<ul>

{this.state.todos.map((todo) => (

<li key={todo.\_id}>{todo.task}</li>

))}

</ul>

</div>

);

}

}

export default App;

**Server.js**

const express = require('express');

const bodyParser = require('body-parser');

const cors = require('cors');

const mongoose = require('mongoose');

const app = express();

const port = process.env.PORT || 5000;

app.use(cors());

app.use(bodyParser.json());

mongoose.connect('mongodb://127.0.0.1/todo-list', { useNewUrlParser: true, useUnifiedTopology: true });

const connection = mongoose.connection;

connection.once('open', () => {

console.log('MongoDB database connection established successfully');

});

const todoSchema = new mongoose.Schema({

task: String,

completed: Boolean

});

const Todo = mongoose.model('Todo', todoSchema);

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

Todo.find((err, todos) => {

if (err) {

console.log(err);

} else {

res.json(todos);

}

});

});

app.post('/todos/add', (req, res) => {

const todo = new Todo({

task: req.body.task,

completed: false

});

todo.save()

.then(todo => {

res.status(200).json({ 'todo': 'todo added successfully' });

})

.catch(err => {

res.status(400).send('adding new todo failed');

});

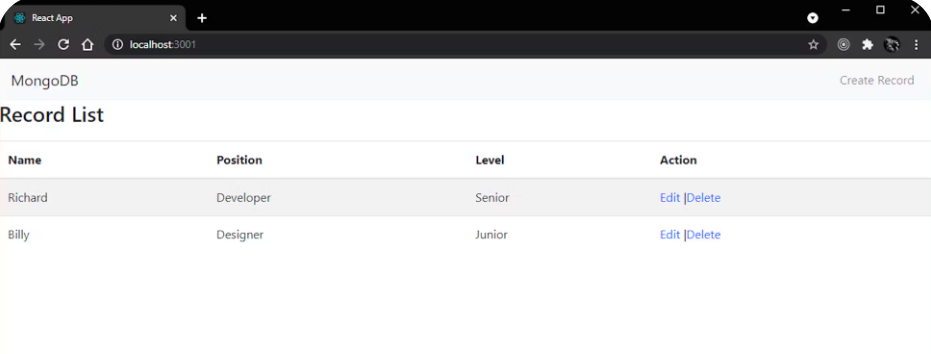
});

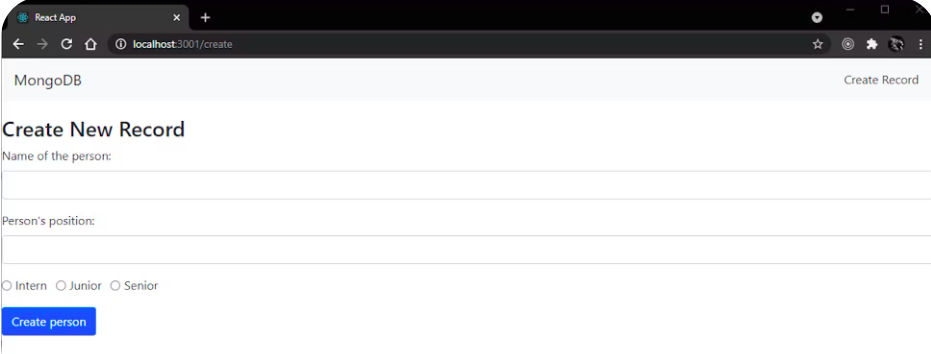
app.listen(port, () => {

console.log(`Server is running on port: ${port}`);

});

**Output :**

****

****

**Result :** A functional MERN stack to-do list application that allows users to add and display tasks stored in a MongoDB database has been implemented.

| **Ex No: 10** | **MERN Stack-II** |
| --- | --- |
| **Date:** |

**Aim:**  To implement the Express.js And MongoDB REST API Tutorial | MongoDB

**Design (Project Tree):**

**.**

**├── app**

**│ └── src**

**│ ├── App.tsx**

**│ ├── components**

**│ │ └── PostSummary.js**

**│ └── pages**

**│ ├── Archive.js**

**│ ├── Create.js**

**│ ├── Home.js**

**│ └── Post.js**

**└── server**

**├── .env**

**├── db**

**│ └── conn.mjs**

**├── index.mjs**

**├── loadEnvironment.mjs**

**└── routes**

**└── posts.mjs**

**Algorithm:**

1) First you will need to deploy an Atlas cluster.

2) Create a server directory and install npm.

3) Run the server.

4) Setup environment variable to establish connection.

5) Make a file to access .env file.

6) Make a file that will be the entry point of the script.

7) Now create the database module

8) Define post, read, delete, update routes.

9) Set up the front-end using browsers native fetch object.

10) Test the application.

**Program:**

App.tsx:

import "./styles.css";

import "./fonts.css";

import LeafygreenProvider from '@leafygreen-ui/leafygreen-provider';

import Layout from "./components/Layout";

import { BrowserRouter as Router, Route, Routes } from "react-router-dom";

import Home from "./pages/Home";

import Archive from "./pages/Archive";

import Create from "./pages/Create";

import Post from "./pages/Post";

function App() {

return (

<LeafygreenProvider>

<Router>

<Routes>

<Route path="/" element={<Layout />}>

<Route path="/" element={<Home />} />

<Route path="/archive" element={<Archive />} />

<Route path="/create" element={<Create />} />

<Route path="/post/:id" element={<Post />} />

</Route>

</Routes>

</Router>

</LeafygreenProvider>

);

}

export default App;

index.mjs:

import express from "express";

import cors from "cors";

import "./loadEnvironment.mjs";

import "express-async-errors";

import posts from "./routes/posts.mjs";

const PORT = process.env.PORT || 5050;

const app = express();

app.use(cors());

app.use(express.json());

// Load the /posts routes

app.use("/posts", posts);

// Global error handling

app.use((err, \_req, res, next) => {

res.status(500).send("Uh oh! An unexpected error occured.")

})

// start the Express server

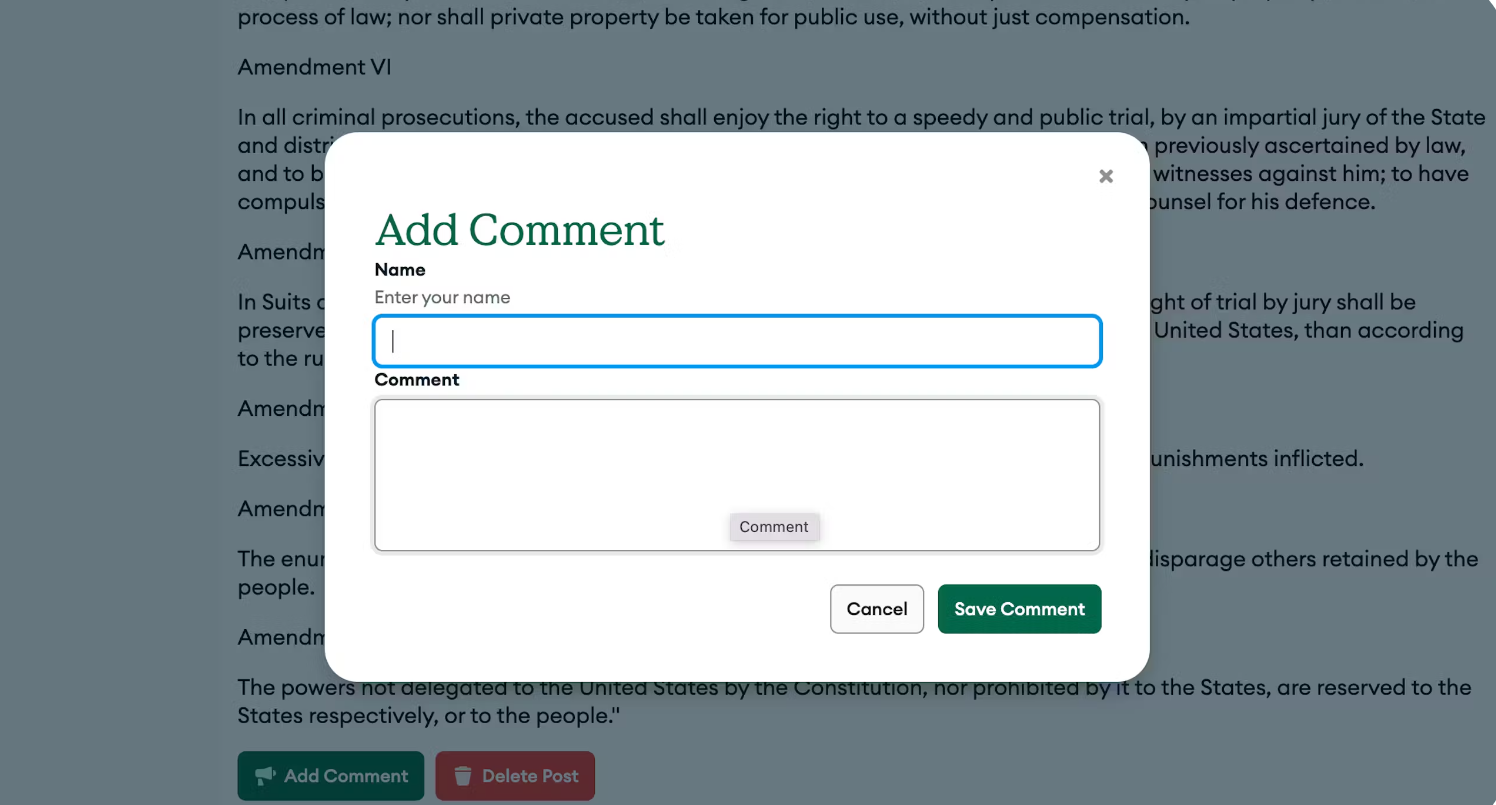
app.listen(PORT, () => {

console.log(`Server is running on port: ${PORT}`);

});

**Output:**

****

****

**Result:** The Express.js And MongoDB REST API Tutorial | MongoDB has been implemented.