**PROGRAM 8**

Develop a Reminder Application that allows users to efficiently manage their tasks. The application should include the following functionalities: Provide a form where users can add tasks along with due dates. The form includes task name,Due date,An optional description. Display a list of tasks dynamically as they are added. Show relevant details like task name, due date, and completion status. Include a filter option to allow users to view all Tasks and Display all tasks regardless of status. Show only tasks marked as completed. Show only tasks that are not yet completed.

**App.jsx**

import React, { useState } from 'react';

import './style.css';

const App = () => {

  const [tasks, setTasks] = useState([]);

  const [taskName, setTaskName] = useState('');

  const [dueDate, setDueDate] = useState('');

  const [description, setDescription] = useState('');

  const [filter, setFilter] = useState('all');

  const handleAddTask = (e) => {

    e.preventDefault();

    if (!taskName || !dueDate) return;

    const newTask = {

      id: Date.now(),

      name: taskName,

      date: dueDate,

      desc: description,

      completed: false,

    };

    setTasks([...tasks, newTask]);

    setTaskName('');

    setDueDate('');

    setDescription('');

  };

  const toggleComplete = (id) => {

    setTasks(

      tasks.map((task) =>

        task.id === id ? { ...task, completed: !task.completed } : task

      )

    );

  };

  const filteredTasks = tasks.filter((task) => {

    if (filter === 'completed') return task.completed;

    if (filter === 'incomplete') return !task.completed;

    return true;

  });

  return (

    <div className="reminder-app">

      <h2>Reminder Application</h2>

      <form onSubmit={handleAddTask} className="task-form">

        <input

          type="text"

          placeholder="Task name"

          value={taskName}

          onChange={(e) => setTaskName(e.target.value)}

          required

        />

        <input

          type="date"

          value={dueDate}

          onChange={(e) => setDueDate(e.target.value)}

          required

        />

        <textarea

          placeholder="Description (optional)"

          value={description}

          onChange={(e) => setDescription(e.target.value)}

        />

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

      </form>

      <div className="filter-buttons">

        <button onClick={() => setFilter('all')}>All</button>

        <button onClick={() => setFilter('completed')}>Completed</button>

        <button onClick={() => setFilter('incomplete')}>Incomplete</button>

      </div>

      <ul className="task-list">

        {filteredTasks.map((task) => (

          <li key={task.id} className={task.completed ? 'completed' : ''}>

            <h3>{task.name}</h3>

            <p>Due: {task.date}</p>

            {task.desc && <p>{task.desc}</p>}

            <label>

              <input

                type="checkbox"

                checked={task.completed}

                onChange={() => toggleComplete(task.id)}

              />

              {task.completed ? 'Completed' : 'Mark as complete'}

            </label>

          </li>

        ))}

      </ul>

    </div>

  );

};

export default App;

------------------------------------------------------------------------------------

**style.css**

body {

  background-color: white;

  margin: 0;

  padding: 20px;

  display: flex;

  justify-content: center;

  align-items: center;

}

.reminder-app {

  max-width: 700px;

  margin: auto;

  background: white;

  padding: 25px;

  border-radius: 12px;

  background-color: black;

  box-shadow: 0 0 15px rgba(0, 0, 0, 0.1);

}

h1 {

  text-align: center;

  color: white;

  margin-bottom: 20px;

}

form {

  display: flex;

  flex-direction: column;

  gap: 12px;

}

input, textarea, select {

  padding: 10px;

  border: 1px solid #ccc;

  border-radius: 6px;

  font-size: 16px;

}

button {

  background-color: green;

  color: white;

  padding: 10px;

  font-size: 16px;

  border: none;

  border-radius: 6px;

  cursor: pointer;

  transition: background-color 0.2s ease-in-out;

}

button:hover {

  background-color: darkgreen;

}

.task-list {

  margin-top: 20px;

}

.task {

  background-color: #f9f9f9;

  padding: 15px;

  margin-bottom: 12px;

  border-radius: 8px;

  border-left: 6px solid #4caf50;

  display: flex;

  justify-content: space-between;

  align-items: center;

}

.task-details {

  flex-grow: 1;

  margin-left: 10px;

}

.task h3 {

  margin: 0;

  color: #333;

}

.task p {

  margin: 5px 0;

  color: #666;

}

.task input[type="checkbox"] {

  transform: scale(1.4);

  cursor: pointer;

}

.filter-options {

  margin-top: 20px;

  display: flex;

  justify-content: center;

  gap: 10px;

}

.filter-options button {

  background-color: #2196f3;

}

.filter-options button:hover {

  background-color: #1976d2;

}

.completed {

  text-decoration: line-through;

  color: gray;

}

**Program 9**

Design a React application that demonstrates the implementation of routing using the react-router-dom library. The application should include the Navigation Menu: Create a navigation bar with links to three distinct pages, Home, About, Contact. Develop separate components for each page (Home, About, and Contact) with appropriate content to differentiate them. Configure routes using react-router-dom to render the corresponding page component based on the selected link. Use BrowserRouter and Route components for routing. Highlight the active link in the navigation menu to indicate the current page

**Home.jsx ( inside component folder)**

import React from 'react';

const Home = () => {

  return (

    <div>

      <h2>Home Page</h2>

      <p>Welcome to the Home Page!</p>

    </div>

  );

};

export default Home;

**About.jsx ( inside component folder)**

import React from 'react';

const About = () => {

  return (

    <div>

      <h2>About Page</h2>

      <p>Learn more about us on the About Page!</p>

    </div>

  );

};

export default About

**Contact.jsx ( inside component folder)**

import React from 'react';

const Contact = () => {

  return (

    <div>

      <h2>Contact Page</h2>

      <p>Get in touch with us through the Contact Page!</p>

    </div>

  );

};

export default Contact;

**Navbar.jsx ( inside component folder)**

import React from 'react';

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

const Navbar = () => {

  return (

    <nav>

      <ul>

        <li>

          <NavLink

            to="/"

            className={({ isActive }) => (isActive ? 'active' : '')}

          >

            Home

          </NavLink>

        </li>

        <li>

          <NavLink

            to="/about"

            className={({ isActive }) => (isActive ? 'active' : '')}

          >

            About

          </NavLink>

        </li>

        <li>

          <NavLink

            to="/contact"

            className={({ isActive }) => (isActive ? 'active' : '')}

          >

            Contact

          </NavLink>

        </li>

      </ul>

    </nav>

  );

};

export default Navbar;

**App.jsx**

import React from 'react';

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

import Navbar from './components/Navbar';

import Home from './components/home';

import About from './components/about';

import Contact from './components/contact';

import './style.css'

const App = () => {

  return (

    <Router>

      <div>

        <Navbar />

        <div style={{ padding: '20px' }}>

          <Routes>

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

            <Route path="/about" element={<About />} />

            <Route path="/contact" element={<Contact />} />

          </Routes>

        </div>

      </div>

    </Router>

  );

};

export default App;

**style.css ( inside src folder)**

body {

  font-family: Arial, sans-serif;

  background-color: #f4f4f4;

  margin: 0;

  padding: 0;

}

div {

  margin: 0 auto;

  max-width: 960px;

  padding: 20px;

  width: 100%;

  height: 100vh;

}

h2 {

  color: #333;

  padding-bottom: 20px;

}

nav {

  background-color: #333;

  padding: 10px;

  border-radius: 5px;

  margin-bottom: 20px;

}

ul {

  list-style: none;

  display: flex;

  gap: 10%;

  justify-content: center;

  margin: 0;

  padding: 0;

}

li {

  display: inline;

}

a {

  text-decoration: none;

  color: white;

  padding: 8px 16px;

  border-radius: 4px;

}

a:hover {

  background-color: #444;

}

a.active {

  background-color: #1e90ff;

  color: white;

  font-weight: bold;

}

p {

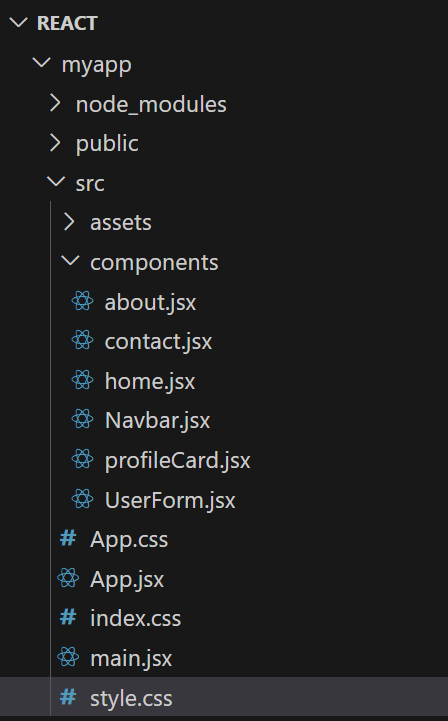
  color: #555;

  font-size: 1.1rem;

  line-height: 1.6;

}

File srtucture



OUTPUT

