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.

Reminder app

Code

App.js

import React, { useState } from 'react';

import TaskForm from './TaskForm';

import TaskList from './TaskList';

import Filter from './Filter';

import './App.css';

function App() {

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

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

  const addTask = (task) => {

    setTasks([...tasks, task]);

  };

  const handleFilterChange = (status) => {

    setFilter(status);

  };

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

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

    if (filter === 'not-completed') return !task.completed;

    return true;

  });

  return (

    <div className="App">

      <h1>Task Reminder</h1>

      <TaskForm addTask={addTask} />

      <Filter setFilter={handleFilterChange} />

      <TaskList tasks={filteredTasks} setTasks={setTasks} />

    </div>

  );

}

export default App;

TaskForm.js

import React, { useState } from 'react';

function TaskForm({ addTask }) {

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

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

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

  const handleSubmit = (e) => {

    e.preventDefault();

    if (taskName && dueDate) {

      const newTask = {

        id: Date.now(),

        name: taskName,

        dueDate: dueDate,

        description,

        completed: false,

      };

      addTask(newTask);

      setTaskName('');

      setDueDate('');

      setDescription('');

    }

  };

  return (

    <form onSubmit={handleSubmit}>

      <div>

        <input

          type="text"

          placeholder="Task Name"

          value={taskName}

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

        />

      </div>

      <div>

        <input

          type="date"

          value={dueDate}

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

        />

      </div>

      <div>

        <textarea

          placeholder="Description (optional)"

          value={description}

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

        />

      </div>

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

    </form>

  );

}

export default TaskForm;

Filter.js

import React from 'react';

function Filter({ setFilter }) {

  return (

    <div>

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

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

      <button onClick={() => setFilter('not-completed')}>Pending Tasks</button>

    </div>

  );

}

export default Filter;

Tasklist.js

import React from 'react';

function TaskList({ tasks, setTasks }) {

  const toggleTaskCompletion = (taskId) => {

    setTasks(

      tasks.map((task) =>

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

      )

    );

  };

  const deleteTask = (taskId) => {

    setTasks(tasks.filter((task) => task.id !== taskId));

  };

  return (

    <div>

      {tasks.length > 0 ? (

        <ul>

          {tasks.map((task) => (

            <li key={task.id}>

              <h3>{task.name}</h3>

              <p>Due Date: {task.dueDate}</p>

              {task.description && <p>Description: {task.description}</p>}

              <p>Status: {task.completed ? 'Completed' : 'Not Completed'}</p>

              <button onClick={() => toggleTaskCompletion(task.id)}>

                {task.completed ? 'Mark as Not Completed' : 'Mark as Completed'}

              </button>

              <button onClick={() => deleteTask(task.id)}>Delete</button>

            </li>

          ))}

        </ul>

      ) : (

        <p>No tasks available!</p>

      )}

    </div>

  );

}

export default TaskList;