**WEEK 6 MINIPROJECT**

1)

import React, { useState } from &#39;react&#39;;

function TaskForm({ addTask }) {

const [taskText, setTaskText] = useState(&#39;&#39;);

const handleInputChange = (e) =&gt; {

setTaskText(e.target.value);

};

const handleSubmit = (e) =&gt; {

e.preventDefault();

if (taskText.trim()) {

addTask({ id: Date.now(), text: taskText, completed: false });

setTaskText(&#39;&#39;);

}

};

return (

&lt;form onSubmit={handleSubmit}&gt;

&lt;input

type=&quot;text&quot;

placeholder=&quot;Add a new task&quot;

value={taskText}

onChange={handleInputChange}

/&gt;

&lt;button type=&quot;submit&quot;&gt;Add Task&lt;/button&gt;

&lt;/form&gt;

);

}

export default TaskForm;

TaskItem.jsx

import React from &#39;react&#39;;

function TaskItem({ task, removeTask, toggleComplete }) {

return (

&lt;li&gt;

&lt;input

type=&quot;checkbox&quot;

checked={task.completed}

onChange={() =&gt; toggleComplete(task.id)}

/&gt;

&lt;span className={task.completed ? &#39;completed&#39; : &#39;&#39;}&gt;{task.text}&lt;/span&gt;

&lt;button onClick={() =&gt; removeTask(task.id)}&gt;Remove&lt;/button&gt;

&lt;/li&gt;

);

}

export default TaskItem;

TaskList.jsx

import React from &#39;react&#39;;

import TaskItem from &#39;./TaskItem&#39;;

function TaskList({ tasks, removeTask, toggleComplete }) {

return (

&lt;ul&gt;

{tasks.map((task) =&gt; (

&lt;TaskItem

key={task.id}

task={task}

removeTask={removeTask}

toggleComplete={toggleComplete}

/&gt;

))}

&lt;/ul&gt;

);

}

export default TaskList;

App.js

import React, { useState } from &#39;react&#39;;

import TaskForm from &#39;./TaskManager/TaskForm&#39;;

import TaskList from &#39;./TaskManager/TaskList&#39;;

import &quot;./App.css&quot;;

function App() {

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

const addTask = (task) =&gt; {

setTasks([...tasks, task]);

};

const removeTask = (taskId) =&gt; {

const updatedTasks = tasks.filter((task) =&gt; task.id !== taskId);

setTasks(updatedTasks);

};

const toggleComplete = (taskId) =&gt; {

const updatedTasks = tasks.map((task) =&gt;

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

);

setTasks(updatedTasks);

};

return (

&lt;div className=&quot;App&quot;&gt;

&lt;h1&gt;Dynamic Task Manager&lt;/h1&gt;

&lt;TaskForm addTask={addTask} /&gt;

&lt;TaskList

tasks={tasks}

removeTask={removeTask}

toggleComplete={toggleComplete}

/&gt;

&lt;/div&gt;

);

}

export default App;

OUTPUT:

