Full Stack Development

Assignment 2:

Assignment Objective

Students will develop a Dynamic Task Manager Dashboard web app that allows users to add, edit, delete, and filter tasks using pure JavaScript (ES6) and Bootstrap 5.
The UI must be responsive and modular, focusing on logic, interactivity, and UI management.

Assignment Description

You are required to build a Task Manager Web App that behaves like a mini productivity dashboard.
Users can:

i. Add tasks with a title, category, and deadline.

ii. Mark tasks as completed.

iii. Filter or sort tasks (e.g., by category or status).

iv. Persist tasks in browser storage.

v. View details in a Bootstrap modal.

Assignment Requirements

1. HTML Structure

i. Create a single-page app (index.html).

ii. Use a Bootstrap navbar with a brand name and theme toggle (dark/light mode).

iii. Include:

- A Task Input Section (form with fields: Task Title, Category, Due Date)
- A Task Display Section using Bootstrap Cards inside a responsive grid.

- A Bootstrap Modal for editing task details.
- A Filter Section (dropdown or buttons to filter tasks).

---

2. CSS Styling

i. Use Bootstrap's grid system for responsiveness.

ii. Apply custom CSS for subtle animations and hover effects.

iii. Bonus: Add CSS transitions when tasks are added or deleted.

---

3. JavaScript Logic (Main Focus)

Implement using modern ES6 features:

i. Use let, const, arrow functions, template literals, and array methods like filter(), map(), and forEach().

ii. Use DOM manipulation to:

   a. Dynamically create task cards.

   b. Update the DOM when tasks are added/edited/deleted.

iii. Use event listeners for buttons and form actions.

iv. Use localStorage or sessionStorage to store tasks persistently.

v. Implement a search or filter feature (by category or completion status).

vi. Add a Dark/Light theme toggle using JS and Bootstrap classes.

---

4. Bootstrap Components (Required Use)

i. Navbar (with collapse on small screens)

ii. Form controls for adding tasks

iii. Cards for displaying tasks

iv. Modal for editing tasks

v. Buttons & badges for actions and status

vi. Responsive grid system

Example Functional Flow

1. User fills form → clicks "Add Task" → new Bootstrap card appears below.
2. Clicking "Edit" opens modal → user updates details → card updates dynamically.
3. Clicking "Complete" changes card border color and adds a ✅ badge.
4. Clicking "Delete" removes the card with a fade-out animation.
5. Tasks persist on page reload via localStorage.

Submission Requirements:

Students are required to submit their completed assignment following the instructions below:
Submit a **single PDF document** containing:

i. Complete **HTML, CSS, and JavaScript code** of your project.

ii. **Screenshots of the website output** clearly showing:

   a. The **main interface (homepage or dashboard)**

   b. **Working features** (e.g., add/edit/delete actions)

   c. **Responsive layout views** for **mobile, tablet, and desktop screens**.

iii. The PDF should be **well-formatted** with clear headings for:

   a. HTML Code
   b. CSS Code
   c. JavaScript Code
   d. Output Screenshots
   e. Responsive View Screenshots
   f. Screenshots should be **clear and labeled** for easy evaluation.

iv. Late submissions may result in mark deductions as per course policy.