Module 8: JavaScript Essentials And Advanced

## Case Overview

**Scenario:** You are an intermediate JavaScript developer at **Hashcrypt technologies Pvt. ltd.** The company wants a smart, error-resilient, and modular **Task Manager Web App**. The app must be responsive, interactive, and follow good programming practices.

## Core Functionality

### Advanced JavaScript Task Manager App

## Tech Stack

## **HTML5** – For structure and form inputs.

**CSS3** – For layout and styling.

**JavaScript (ES6+)** – For dynamic behavior, form handling, DOM manipulation, and local storage.

## Features

Task creation form with fields:

Task Title (text)

Description (textarea)

Due Date (date)

Priority (select: High, Medium, Low)

Status (select: To Do, In Progress, Done)

Form validation (required fields)

Task display in a list

Edit and Delete task actions

Persist tasks using localStorage

Real-time UI update on actions

* Mobile-friendly layout using **Flexbox/Grid** and **media queries**

### 2. Form Handling with Advanced JavaScript

**Class:** **CustomerFormHandler**

* **validateForm()** – Validate:  
  + task title ≥ 10 characters
  + Descriptions
  + Due date
  + Priority
  + status
* saveToLocalStorage() – Save data as array in localStorage
* clearForm() – Reset form after submission
* showMessage() – Reusable UI message box for success/error

Additional Requirements:

* Use **event delegation** for handling form events
* Implement **real-time validation** (on input or blur)

### 3. View Submissions Page (view.html)

**Class:** SubmissionViewer

* Read from localStorage
* Dynamically generate table of submissions
* Search bar to filter by **Name** or **Vehicle**
* Show **“No data found”** message if results empty

## Optional Bonus Tasks

* Add **Delete** button to remove individual records
* Use **utility module** for setItem() and getItem() (modularized code)
* Implement **import/export** (<script type="module"> or bundler like Vite/Webpack)

## Validation Rules

* All fields required
* Descriptions
* Due date
* Priority
* status

## Key Competencies Tested

* ES6 Classes & Modules
* DOM Manipulation & Event Handling
* Real-Time & Manual Form Validation
* LocalStorage CRUD Operations
* Dynamic HTML Rendering & Filtering
* Clean, Modular, and Reusable Code

## Reflective Thinking Questions

* How could this app be enhanced by integrating with an **API** (e.g., saving data to a backend)?
* How can error handling be improved for a better **user experience**?
* What security concerns arise when storing customer data in localStorage?
* Could the app be scaled into a **PWA** (Progressive Web App) for offline use?