Assessment

**Module 8:  JavaScript Essentials And Advanced**

**Case Overview:**

**Scenario:**

You are an **intermediate JavaScript developer at Hashcrypt technologies Pvt. ltd**. Your team has been assigned create a **modular, smart, and interactive Job Application Web App** for a tech recruitment firm. The form should be responsive, perform real-time validation, and follow clean coding practices.

|  |  |
| --- | --- |
| **Core functionality** | |
| |  | | --- | | 1. **Responsive Task**   **Manager App Form** |  |  | | --- | |  | | Create a form with the following fields:   * **Task Name** (Text) * **Descriptions**  (textarea) * **Due date**  (date) * **Priority (High, medium and low apply** Dropdown) * Use **Flexbox/Grid** layout with **media queries** for responsiveness on mobile and tablets. |
| 2.Form Handling with Advanced JavaScript | **Class: TaskManagerHandler**   | **Method** | **Functionality** | | --- | --- | | validateForm() | Validate fields using regex and rules: • task Name ≥ 10 characters • descriptions  • due date(date formate) • priority (dropdown) | | saveToLocalStorage() | Store validated applications as an array  in localStorage | | clearForm() | Reset all form fields after successful submission | | showMessage(type, message) | Display styled success/error alerts | | addEventListeners() | Use **event delegation** and attach real-time validation to inputs (oninput or onblur) | |
| 3. View Applications Page (view-applications.html) | **Class: TaskManager Viewer**   * Fetch and display data from localStorage in a table. * Add a **search input** to filter applications by Name or Position. * Show “No applications found” message when data is empty or filtered results don’t match. |

# Optional Bonus Tasks

# Add Delete button beside each application row to remove it from localStorage.

# Modularize code using a separate utility module (storageUtils.js) for getItem() / setItem().

# Structure JavaScript using ES Modules and optionally use a bundler like Vite/Webpack.

# Validation Rules

| Field | Rule |
| --- | --- |
| Taskname | Taskname ≥ 10 characters |
| Descriptions | Any type of data |
| Due date | Date type |
| Priority | dropdown |

# Key Competencies Tested

# ES6 Classes and Modular Code

# DOM Manipulation and Clean UI

# Event Delegation and Real-Time Input Validation

# localStorage CRUD operations

# Responsive Design with CSS Grid/Flexbox

# Search and Filter Functionality

# Reusable Component and Utility Design

# Reflective Thinking Questions

# How would this app differ if data were stored on a backend (e.g., Firebase, Node API)?

# What improvements can be made to handle validation errors more accessibly (e.g., ARIA)?

# How secure is storing job applications in localStorage? What are safer alternatives?

# How can this form be scaled into a Progressive Web App (PWA) for offline job fairs?