

Milestone-1 Assessment – Front-End Development (E-Commerce Product Management System)

Duration: 90 Minutes

Total Marks: 60

Mode: Individual (Manual Submission)

Submission Format: ZIP Folder or PDF Report

Objective

This assessment evaluates your ability to design and implement a functional, responsive, and data-driven **E-Commerce Product Management Interface** using **HTML5, CSS3, Bootstrap, JavaScript (ES6), and TypeScript**.

You will demonstrate hands-on coding, UI structuring, data handling, and modular logic using best coding practices.

Assessment Overview

User Story	Focus Area	Complexity	Marks
User Story 1	HTML5, CSS3, Bootstrap Layout	Easy	20
User Story 2	JavaScript, Fetch API, ES6 Concepts	Medium	20
User Story 3	TypeScript with Classes, Interfaces, and Decorators	Difficult	20
Total			60 Marks

User Story 1 – Product Showcase Home Page (20 Marks)

Duration: 25 Minutes

Scenario

Create a **Home Page** for an E-Commerce Product Portal.

The page should display featured products, a category navigation bar, and a newsletter subscription form.

Requirements

- Use **semantic HTML5** structure (header, nav, main, section, footer).
- Design layout using **Bootstrap grid** for responsiveness.
- Represent products using **cards** (image, name, and price).
- Implement a **form with validation** for newsletter subscription.
- Use **Flexbox or Grid** for product alignment.
- Apply **CSS transitions** for hover effects.

Evaluation Criteria (20 Marks)

Criteria	Description	Marks
HTML Structure	Correct and semantic use of HTML elements	4
Layout & Responsiveness	Proper use of Bootstrap grid	4
Form & Validation	Working and intuitive form	3
Visual Design	Aesthetic alignment and color consistency	5
Code Readability	Indentation and clean structure	2
Creativity	Innovative styling or layout	2
Total		20

User Story 2 – Product Listing with API Fetch and Filtering (20 Marks)

Duration: 30 Minutes

Scenario

Develop a **Product Listing Page** that dynamically fetches data from a mock API (products.json or FakeStoreAPI) and allows filtering by category and price.

Requirements

- Implement **Fetch API** with **async/await**.
- Dynamically display product data using **DOM manipulation**.
- Handle API errors gracefully using **try...catch**.
- Add **filter or sort** functionality.
- Display a **loading message** during API calls.

Evaluation Criteria (20 Marks)

Criteria	Description	Marks
API Handling	Fetch API with async/await	5
Error Handling	Use of try...catch and fallback UI	3
DOM Manipulation	Dynamic product rendering	5
Filter/Sort Logic	Correct filtering or sorting mechanism	4
Responsiveness	Instant updates and smooth UI	3
Total		20

User Story 3 – Product Management Module (TypeScript) (20 Marks)

Duration: 35 Minutes

Scenario

Build a **Product Management Module** using **TypeScript** to model and track product inventory.

Requirements

- Create an **Interface** `IProduct` with fields: `id`, `name`, `category`, `price`, `stock`.
- Implement a **Class** `Product` using this interface.
- Use a **Decorator** to log changes in price or stock.
- Store products in a **Map or Array of Tuples**.
- Iterate and display products using **for...of** or **iterators**.
- Ensure successful **compilation** from TypeScript to JavaScript.

Evaluation Criteria (20 Marks)

Criteria	Description	Marks
Interface & Class	Correct interface and class implementation	5
Type Annotations	Appropriate use of enums, tuples, and types	4
Decorator	Functional and meaningful decorator	4
Iteration & Logic	Proper traversal and display logic	4
Code Execution	Code compiles and runs successfully	3
Total		20

Submission Instructions

Format

Submit as either:

- A **ZIP folder** containing all code files, or
- A **PDF report** with:
 - Screenshots of outputs for each user story
 - Key code snippets
 - 2-3 line explanations per functionality

File Naming Convention

ParticipantName_ECommerceAssessment.zip

Example: RohanVerma_ECommerceAssessment.zip

Folder Structure Example

```
/ECommerceAssessment/  
├── userstory1/  
│   ├── index.html  
│   └── style.css  
├── userstory2/  
│   ├── script.js  
│   └── products.json  
├── userstory3/  
│   ├── productManager.ts  
│   └── productManager.js  
└── screenshots/  
    ├── home.png  
    ├── api_output.png  
    └── ts_output.png
```

Submission Mode

- Upload on LMS/Portal or share drive link.
- Ensure files run without dependency issues.
- TypeScript must compile successfully using tsc.

Late Submission Policy

- 10% mark deduction per 15-minute delay.

Evaluation Summary (60 Marks)

Component	Marks
User Story 1 - HTML/CSS/Bootstrap	20
User Story 2 - JavaScript + Fetch API	20
User Story 3 - TypeScript Module	20
Total	60 Marks