

Mock Practice Assessment – Front-End & Database Integration

Total Marks: 60
Duration: 120 Minutes
Mode: Individual Assessment
Submission Format: ZIP File or PDF Report

Objective

This practice assessment evaluates your ability to design and integrate a complete web solution combining front-end technologies (HTML, CSS, Bootstrap, JavaScript, TypeScript) and database interaction. You will be assessed on the implementation quality, responsiveness, and overall integration approach.

Assessment Overview

User Story	Focus Area	Complexity	Marks
User Story 1	HTML, CSS, Bootstrap	Easy	20
User Story 2	JavaScript (ES6), Fetch API	Medium	20
User Story 3	TypeScript (Interfaces, Classes, Decorators)	Difficult	20
Total			60

User Story 1 – Event Management Homepage (HTML, CSS, Bootstrap)

Scenario

Design a responsive homepage for an Event Management company showcasing upcoming events, event highlights, and a registration form. The page should use semantic HTML5 elements and Bootstrap components for layout and styling.

Requirements

Criteria	Description	Marks
Structure & Semantics	Use proper HTML5 structure (header, nav, main, footer, sections)	5
Styling & Layout	Apply Flexbox or Grid with responsive Bootstrap classes	5
Forms & Validation	Include a registration form with basic validation (e.g., email, required fields)	4
Bootstrap Components	Use at least two Bootstrap components (carousel, cards, modal, etc.)	3
Presentation & Creativity	Consistent color palette, font choices, and overall layout	3
Total Marks (User Story 1)		20

User Story 2 – Event Dashboard (JavaScript + ES6)

Scenario

Develop an interactive dashboard that displays a list of events, allows filtering by category or date, and supports live updates using JavaScript and ES6. Simulate API responses using Fetch API with a mock JSON file.

Requirements

Criteria	Description	Marks
Data Logic & Structure	Use arrays, objects, and iteration effectively	4
ES6 Features	Apply let/const, arrow functions, destructuring, and template literals	5
Fetch API + Async/Await	Implement simulated API calls with async/await and error handling	4

DOM Manipulation	Dynamically render event cards and update on user actions	4
Code Readability	Proper indentation, modular structure, and clear comments	3
Total Marks (User Story 2)		20

User Story 3 – TypeScript Customer Registration Module

Scenario

Create a TypeScript-based module that handles customer registration and management for the event platform. Demonstrate the use of interfaces, classes, decorators, enums, and tuples. Ensure the module compiles cleanly to JavaScript.

Requirements

Criteria	Description	Marks
Interface & Class Design	Define interfaces and implement classes with proper inheritance	5
Type Safety	Use annotations, tuples, and enums effectively	4
Decorators	Implement at least one custom decorator (e.g., logging, input validation)	4
Iterators & Modularity	Use iterators and maintain organized file structure	4
Compilation	Code compiles successfully using the TypeScript compiler	3
Total Marks (User Story 3)		20

Bonus Section (Optional – 10 Marks)

Database Integration Challenge (MySQL or MongoDB)

Extend your dashboard or TypeScript module to connect to a database. Implement CRUD operations for managing events or customers.

Criteria	Description	Marks
Database Connectivity	Establish connection to MySQL or MongoDB	3
CRUD Operations	Implement Create, Read, Update, Delete functions	5
Error Handling	Apply input validation and error management	2
Total Marks (Bonus)		10

Submission Instructions

Format Options

Submit your work as a ZIP file containing:

- HTML, CSS, JS, and TS files
- Screenshots of outputs
- Compiled JavaScript files from TypeScript
- Database scripts or screenshots (for bonus task)

OR

Submit a PDF report containing:

- Description of each user story
- Key code snippets
- Screenshots of final output
- 2-3 line explanation of approach

Naming Convention

ParticipantName_MockPractice_Frontend.zip

Example: AmitKumar_MockPractice_Frontend.zip

Folder Structure Example

```
/MockPracticeAssessment/  
|-- userstory1/  
|   ├── index.html  
|   └── style.css  
|-- userstory2/  
|   └── script.js  
|-- userstory3/  
|   ├── CustomerModule.ts  
|   └── CustomerModule.js  
|-- screenshots/  
|   ├── homepage.png  
|   ├── dashboard.png  
|   └── typescript_output.png
```

Execution Requirements

- All HTML files must run correctly in a browser.
- TypeScript code must compile using tsc.
- Include a README.txt for setup or dependencies.

Deadline

Submit all files within 120 minutes.

Late submissions will result in a 10% deduction for every 15-minute delay.