|  |  |
| --- | --- |
| Criteria | Description |
| OOP Principles | Proper use of classes, methods, encapsulation, and inheritance to model and organize code. |
| Functionality | Implementation of all required features, handling edge cases, and ensuring the program works as intended. |
| DOM Manipulation | Effective use of JavaScript to dynamically add, update, and remove elements from the HTML and CSS. |
| Event Handling | Proper use of event listeners (e.g., *click, submit*) to enable interactivity in the application. |
| Asynchronous Programming | Handling delays with *setTimeout, setInterval*, or *Promise*. Fetching or simulating data from APIs. |
| Storage Management | Effective use of *localStorage* or *sessionStorage* to persist data between sessions or page reloads. |
| API Integration | Fetching, displaying, and managing data from mock APIs, or using JSON files to simulate server communication. |
| UI/UX | Designing a user-friendly, intuitive interface with responsive design principles. |