OLP – Design Documentation

**OLP – Design Documentation**

The design of the OLP (Online Learning Platform) is built on a modern, modular architecture, ensuring scalability, maintainability, and an optimal user experience for all stakeholders — students, teachers, and administrators.

**1. Frontend:**

* **Technologies Used:** HTML, CSS, Bootstrap, and Material UI.
* **Design Goals:**
  + Ensure full responsiveness across devices (mobile, tablet, desktop).
  + Use Material UI components for modern visual consistency and interactivity.
  + Implement intuitive navigation with minimal clicks to reach core features.

**2. Backend:**

* **Framework:** Node.js with Express.js.
* **Responsibilities:**
  + Handle API routing, user authentication, and data manipulation.
  + Maintain separation of concerns using modular controllers and services.
  + Enable easy scalability through structured API endpoints.

**3. Database:**

* **Database Used:** MongoDB Atlas (NoSQL).
* **Data Models Include:**
  + Users (with roles: Student, Teacher, Admin)
  + Courses (title, modules, content, resources)
  + Certificates
  + Progress tracking

**4. Integration Layer:**

* **Tool:** Axios (on the frontend).
* **Purpose:**
  + Seamlessly connect frontend components to backend APIs.
  + Enable secure and efficient data exchange (GET, POST, PUT, DELETE).
  + Display real-time updates on the frontend based on API responses.

**5. UI/UX Focus:**

* **Core Design Features:**
  + Dashboard-specific layouts for each user role.
  + Visual indicators like progress bars, badges, and alerts.
  + Simple CTA buttons (Start Course, Submit Quiz, Download Certificate).
  + Light and dark theme support (optional enhancement).
* **Accessibility:** Semantic HTML and ARIA tags for screen readers.

**6. Security:**

* **Authentication:**
  + JWT (JSON Web Tokens) to manage sessions and protect API routes.
  + Role-based route guarding to restrict access to sensitive pages and actions.
* **Validation & Error Handling:**
  + Frontend form validations using regex and real-time feedback.
  + Backend input sanitization and proper status code handling.