OLP – Requirement Analysis Documentation

The **Online Learning Platform (OLP)** is built to provide a rich, flexible, and personalized learning experience for users from diverse backgrounds. It focuses on seamless navigation, efficient content delivery, and role-specific interactions to promote effective learning and teaching.

**1. User Roles**

* **Students**: Learners who browse, enroll, and participate in courses.
* **Teachers**: Content creators and instructors responsible for course management.
* **Admins**: System overseers who ensure smooth operation and platform integrity.

**2. Student Features**

* **User Registration & Login**: Students must register and authenticate securely.
* **Course Browsing**: Filter and search for courses by category, difficulty, or instructor.
* **Enrollment**: Ability to enroll in both free and paid courses.
* **Progress Tracking**: View module completion, quiz attempts, and time spent.
* **Interactive Learning**: Access discussion forums, attend live webinars, and receive notifications.
* **Certification**: Generate and download certificates after successful course completion.
* **Payment Integration**: Use secure payment gateways to purchase premium courses.

**3. Teacher Features**

* **Profile Setup**: Teachers create detailed profiles highlighting expertise.
* **Course Management**: Upload videos, PDFs, quizzes, and structure curriculum.
* **Live Sessions**: Host and manage real-time webinars and Q&A sessions.
* **Progress Monitoring**: View analytics of student engagement and performance.
* **Student Interaction**: Respond to forum queries, post announcements, and offer feedback.

**4. Admin Features**

* **User Management**: Approve/ban users, reset credentials, assign roles.
* **Course Oversight**: Moderate course content, approve new courses.
* **Transaction Logs**: Monitor and track financial transactions securely.
* **Issue Handling**: Resolve disputes, respond to complaints, and manage reports.
* **Platform Analytics**: Monitor user activity, engagement metrics, and platform health.

**5. Functional Requirements**

* **Authentication & Authorization**: Role-based login system using JWT tokens.
* **Course Management API**: RESTful APIs to fetch, update, and delete course content.
* **Discussion Forums & Messaging**: Enable real-time interaction using sockets or polling.
* **Certificate Generator**: Auto-generate PDF certificates upon completion.
* **Mobile Responsiveness**: Fully functional across devices (web, tablet, mobile).

**6. Non-Functional Requirements**

* **Availability**: 99.9% uptime using cloud-hosted infrastructure.
* **Performance**: Optimized API response (<300ms) and lazy-loaded assets.
* **Security**: Encrypted passwords, secure login, HTTPS protocols.
* **Scalability**: Microservices-ready design for easy horizontal scaling.
* **Maintainability**: Modular codebase and consistent documentation.
* **Accessibility**: WCAG 2.1-compliant design for inclusive learning.