Frontend Development Requirements for Moodle-Based LMS Platform

Note: It is the basic theme being shared, when you confirm that you can manage to do the task, then complete requirements will be shared.

1. Project Overview

We are building an online education platform where **Moodle LMS** will serve as the **core backend system**, while a **custom-coded frontend** will replace the current WordPress + Edwiser Bundle Pro setup.

The objective is to achieve:

- Better performance
- Greater flexibility
- Scalable architecture
- Advanced dashboards and analytics
- Long-term maintainability

---

2. Existing System (For Reference)

Currently implemented using:

- Moodle LMS (Backend)
- WordPress (Frontend)
- Edwiser Bundle Pro (Connector)

Current features:

- Course synchronization from Moodle
- Student enrolment
- Single Sign-On (SSO)
- Payment gateway integration
- Basic progress dashboards

---

3. Proposed Architecture

Backend

- **Moodle LMS (latest stable version)**
- Moodle Web Services (REST APIs)
- Custom External Functions if required

Frontend

- Fully coded frontend (React / Next.js / Vue / equivalent)
- No WordPress dependency

---

4. Core Functional Requirements

4.1 User Management

- User registration & login via frontend
- Single Sign-On with Moodle
- Role handling:
  - Student
  - Teacher
  - Admin
- Password reset & profile management

4.2 Course Management

- Fetch courses dynamically from Moodle
- Display:
  - Categories

    - Course descriptions
    - Pricing
- Course detail pages
- Enrollment triggers via frontend

4.3 Enrollment & Access Control

- Auto-enrol users in Moodle after:
    - Successful payment
    - Admin assignment
- Restrict course access based on enrollment
- Sync enrollment status in real time

4.4 Payment Integration

- Integration with payment gateways (to be finalized)
- Payment success → enrollment automation
- Transaction logging
- Refund / failure handling (if supported)

4.5 Student Dashboard

Custom dashboard showing:

- Enrolled courses
- Course progress (percentage)
- Completed vs pending activities
- Quiz scores

- Certificates (if enabled)

📊 Visual representations:

- Progress charts
- Completion graphs
- Performance trends

4.6 Instructor Dashboard

- View assigned courses
- Student progress overview
- Quiz / activity analytics
- Reports export (CSV / PDF)

4.7 Admin Dashboard

- User overview
- Enrollment statistics
- Revenue reports
- Course analytics
- System health indicators

---

5. Technical Requirements

API Integration

- Use Moodle REST APIs
- Secure token-based authentication
- Custom Moodle plugins if necessary

Security

- HTTPS enforcement
- Secure token handling
- Role-based access control
- Input validation

---

6. Non-Functional Requirements

- High performance (fast page loads)
- Scalable for future growth
- SEO-friendly frontend
- Mobile responsive UI
- Clean, maintainable codebase

---

7. Deliverables Expected

- Fully functional frontend
- API integration documentation
- Deployment guide
- Code handover
- Basic admin usage documentation

---

8. Future Expansion (Phase-2 Ready)

- Mobile app integration
- AI chatbot integration
- Advanced learning analytics

- Multi-language support

🔚 Final Note to Developer

The solution must **replace WordPress and Edwiser Bundle Pro entirely** while **meeting or exceeding current functionality**, with improved scalability, performance, and customization.