Project Report: Online Course and Exam Platform

1. Project Overview

The Online Course and Exam Platform is a web-based system designed to enable digital learning, automated testing, and real-time performance tracking. It supports multiple user roles (admin, instructor, student) and offers tools for course management, assessments, and progress reporting.

2. Technology Stack

1. Frontend: React.js
2. Backend: Spring Boot (Java)
3. Database: PostgreSQL
4. Authentication: JWT-based
5. Deployment: Docker + Kubernetes (optional), NGINX, CI/CD (e.g., GitHub Actions)
6. Hosting: AWS / Azure / Local Serve

3. Key Modules:

3.1 Admin Module

1. Manage users (CRUD)
2. Approve instructor applications
3. Monitor platform metrics
4. Control categories and tags

3.2 Instructor Module

1. Create/manage courses
2. Upload materials (PDFs, videos)
3. Create exams and quizzes
4. Track student performance

3.3 Student Module

1. Browse and enroll in courses
2. View course content
3. Attempt exams and quizzes
4. View grades and progress

3.4 Exam Engine

1. Support for MCQ-type questions
2. Timer and auto-submission
3. Shuffling of questions/options
4. Evaluation and instant scoring

3.5 Reporting Module

1. Progress reports for students
2. Performance analytics for instructors

Exam statistics

Export to PDF/CSV

4. System Design

4.1 High-Level Architecture

1. Frontend (React): Handles UI/UX, uses Axios or Fetch API for REST calls.
2. Backend (Spring Boot): Exposes REST endpoints, handles logic, security, and service orchestration
3. Database (PostgreSQL): Stores users, courses, exams, results, logs.

4.2 Low-Level Design (Key Components)

a) User Management

- Entity: User (fields: id, name, email, passwordHash, role, status)
- Roles: ENUM (ADMIN, INSTRUCTOR, STUDENT)
- Authentication: Spring Security with JWT

b) Course Module

1. Entities: Course, Module, Material
2. Relationships:
    - Instructor (1) → Course (many)
    - Course (1) → Module (many)
    - Module (1) → Material (many)

c) Exam Engine

1. Entities: Exam, Question, Option, Submission, Answer
2. Features:
    - Shuffle questions/options (using utility service)
    - Timer using React useEffect
    - Auto-save answers (AJAX)
    - Score calculation (in backend)

d) Progress Tracking

- Store Progress with: user_id ,course_id, completed_modules, scores

e) Reporting

- Spring services to generate reports
- Scheduled tasks for weekly progress summary
- Export via Apache POI (Excel) or iText (PDF)

4. **Database Schema Overview**
   User (id, name, email, password, role, status)
   Course (id, title, description, instructor_id)
   Module (id, course_id, title, order)
   Material (id, module_id, type, content_url)
   Exam (id, course_id, duration, created_by)
   Question (id, exam_id, question_text, correct_option_id)

Option (id, question_id, option_text, is_correct)
Submission (id, user_id, exam_id, score, submitted_at)
Answer (id, submission_id, question_id, selected_option_id)
Progress (id, user_id, course_id, completed_modules, last_accessed)

6. Security and Roles

- JWT Auth for secure access
- Role-based access in backend via Spring Security annotations (@PreAuthorize)
- Passwords hashed with bcrypt
- Rate-limiting for login API

7. Responsive UI/UX

- Built with React + Tailwind/Bootstrap
- Responsive design for desktop, tablet, and mobile
- Dashboard with stats, progress bar, and action items
- Timer and live exam view for students
- Dark mode optional

8. Evaluation Criteria

Component Description:

- MCQ Exam Engine
- Fully automated with timer, shuffle, scoring
  Backend Design RESTful API
- modular service layers, secure auth
- UI/UX Clean, responsive, intuitive
- Reporting Real-time progress and downloadable reports

9. Deployment Strategy

- Dockerize Spring Boot and React apps
- Use NGINX as reverse proxy
- PostgreSQL volume with backup
- CI/CD with GitHub Actions
- Environment variables via .env files

10. Future Enhancements

- Video conferencing integration (Zoom API)
- Subjective exam support
- AI-based plagiarism detection
- Mobile app (React Native)
- Gamification (badges, leaderboards)