A2 Batch ODD Roll Numbers

Learning Management System (LMS)

Develop an LMS where students can enroll in courses, complete lessons, and track their progress.

Frontend:

The React UI will include dashboards for students to view enrolled courses, course progress, and upcoming deadlines. Lessons will have videos, quizzes, and downloadable content. Use React Router for navigation and conditional rendering for role-based access (e.g., admin, student).

Backend:

Build APIs using Node.js for managing courses, quizzes, and user progress:

- `GET /courses` to fetch available courses.
- `POST /enroll` to enroll in a course.
- `GET /progress` to fetch course progress.

MongoDB/MySQL will store course details, lesson content, and student progress.

A Learning Management System (LMS) could enable students to enroll in courses, complete lessons, and track their progress. The React-based frontend would feature a student dashboard displaying enrolled courses, lessons, and progress metrics, while role-based access would provide admins with options to manage course content. The backend, powered by Node.js, would manage course data, user progress, and quiz submissions. With a database like MongoDB or MySQL, the system would efficiently store course materials, lessons, and user progress, creating a scalable platform for online education.

A2 Batch EVEN Roll Numbers

Job Portal Application

Develop a job portal where users can search for jobs, apply, and manage their applications.

Frontend:

Use React to build a search interface for job listings with filters like location, experience level, and salary range. Applicants can upload resumes and track the status of their applications. Employers can post and manage job openings.

Backend:

Node.js APIs will manage job listings and applications:

- `GET /jobs` to fetch job listings.
- `POST /jobs` for employers to post new jobs.
- `POST /applications` for users to apply for a job.
- `GET /applications/:userId` to fetch user applications.

Use MongoDB/MySQL to store job and application data.

A Job Portal Application could provide a platform for users to search for jobs, apply, and track application status. The React frontend would feature a search interface with filters like location and experience level, and a dashboard for users to view application history. Employers could post and manage job openings using the same platform. On the backend, Node.js APIs would handle job listings, applications, and user data, with MongoDB or MySQL managing storage. Features such as automated notifications and role-based dashboards could be added.