LMS Development Progress Report

# Day 1 – 8 April: Environment Setup

Set up the initial development environment. Installed and configured Django for the backend and React for the frontend. Created a local MySQL database using XAMPP and named it `lms\_db`. Decided to structure the project into Django (backend) and React (frontend) components. This marked the official start of the LMS development.

# Day 2 – 9 April: React Frontend Initialization

Initialized the React frontend project and installed key dependencies like `react-router-dom`. Created the RoleSelect landing page with login options for Student, Instructor, and Funder. Successfully implemented basic routing for each user login page.

# Day 3 – 10 April: Login Pages & Routing

Built individual login components for each role (Student, Instructor, Funder). Added visual styling using plain CSS. Ensured all login routes work correctly with React Router. Also improved the role selection UX with dropdown enhancements.

# Day 4 – 11 April: Student Dashboard Setup

Started developing the student dashboard. Created the sidebar layout and navigation structure. Decided to use basic CSS for UI styling rather than Tailwind. Prepared the routing system to allow nested dashboard components.

# Day 5 – 12 April: Functional Dashboard Pages

Created several student dashboard pages including StudentProfile, StudentGrades, and initial Message components. Setup the sidebar to consistently appear across all student pages. Connected login to redirect into the dashboard view.

# Day 6 – 13 April: MySQL Table Setup on XAMPP

Created and implemented SQL scripts for all required tables using MySQL via XAMPP. Tables included: students, instructors, funders, courses, enrollments, grades, announcements, quizzes, and messages. This laid the foundation for dynamic backend integration.

# Day 7 – 14 April: Messaging & Grade Details

Implemented messaging pages: Messages.js and MessageChat.js. Also built a dynamic subpage for grades (StudentGradesSub.js) with mock data. Improved styling across pages and centralized styling using DashboardStyles.css.

# Day 8 – 15 April: Course Details & Final Touches

Finalized the StudentCourses.js page and created dynamic routing to StudentCourseDetails.js. Implemented individual content views per course. Also prepared the system for scalability, using PostgreSQL going forward and evaluating deployment options.

Tools & Resources Needed Going Forward

The following tools and platforms are suggested for future development, deployment, and collaboration:

|  |  |  |
| --- | --- | --- |
| Tool | Purpose | Estimated Price (ZAR) |
| GitHub | Version control & team collaboration | Free |
| VS Code | Source code editor | Free |
| PostgreSQL | Relational database (used for backend) | Free (Local) / R0–R400 (Cloud) |
| Google Cloud Storage | Storage for files & documents | Free Tier (5GB) / ~R20–R50+ |
| Render | Recommended deployment option for full-stack apps | ~R100–R300 |
| Railway | Deployment & database hosting with ease of use | Free (limited) / ~R200+ |
| Heroku | Classic PaaS for deploying Django/PostgreSQL apps | Free (sleep mode) / ~R100+ |