Learning Management Platform

sdmay24-46

Sam DeFrancisco - Backend Engineer
Nikhil Kuricheti - Frontend Engineer
Nick Erickson - Backend Engineer
Jennifer Robles - Frontend Engineer
Brayton Rude - Backend Engineer
Deepika Vempati - Frontend Engineer
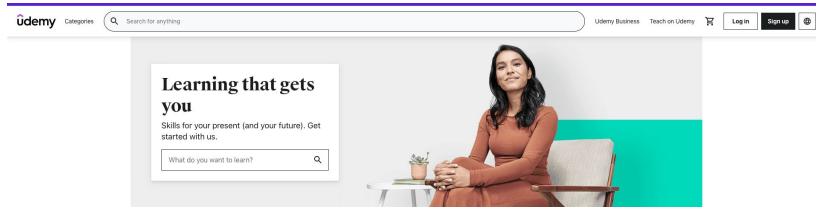
Client/Advisor: Judith Islam

Project Vision

Our goal is to provide a centralized platform for students to access quality educational resources and user-driven study tools.

Our platform will make learning more efficient and effective by organizing content into lessons, tracking user progress, and offering study tools like flashcards and quizzes.

Concept Visual

System Design - Overall Architecture

Client Side
- ReactJS

Server
- Java (SpringBoot)

Database
- S3 (AWS)
- Neptune (AWS)
- RDS (AWS)

System Design
More Detailed
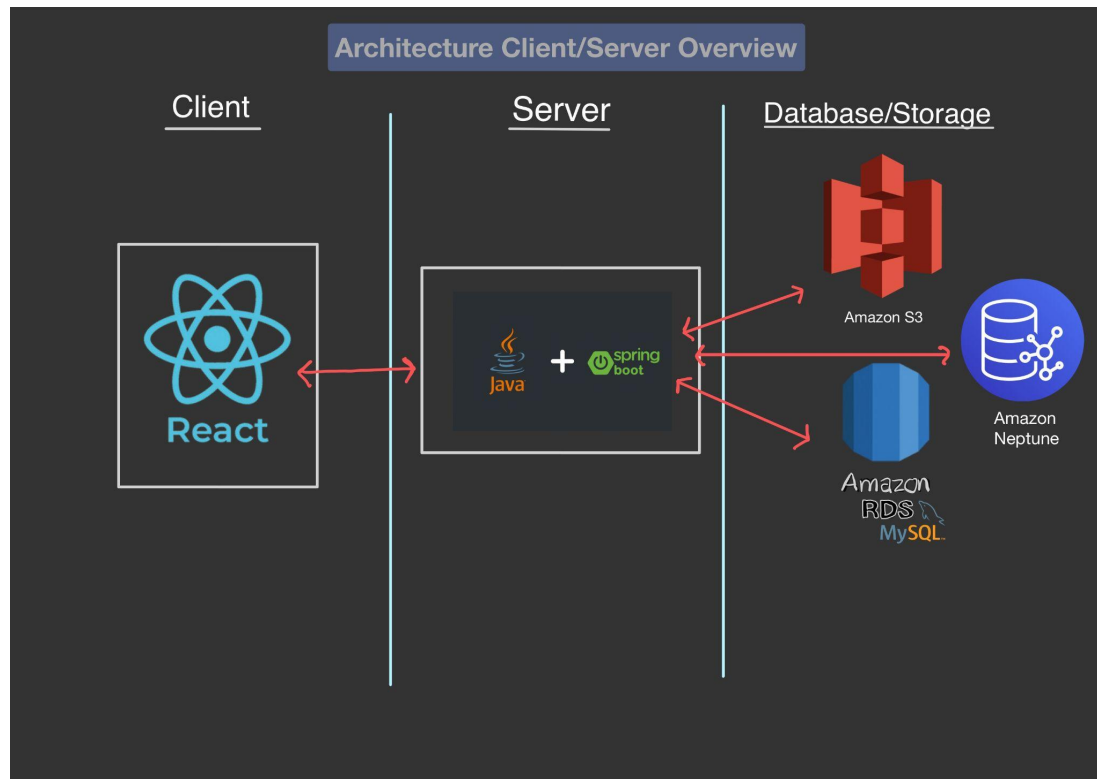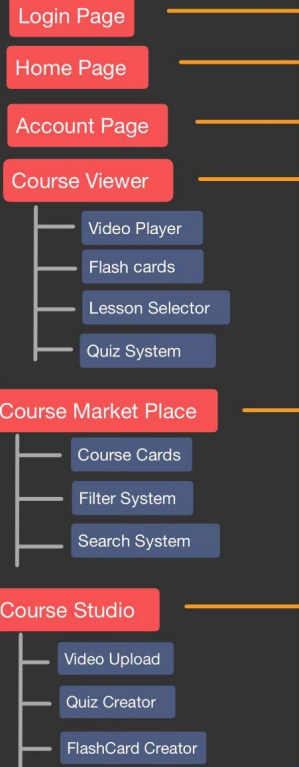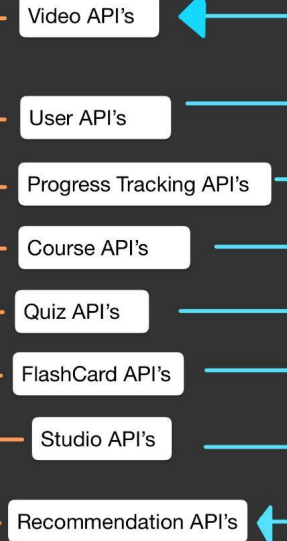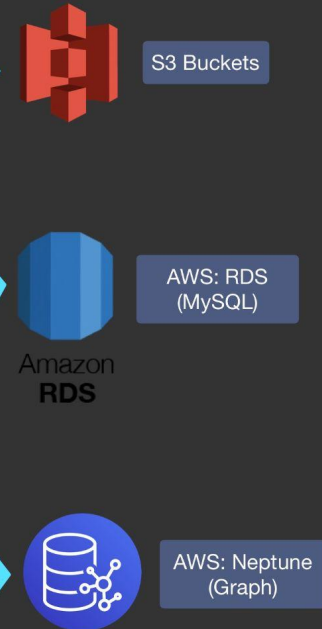
System Design - Databases
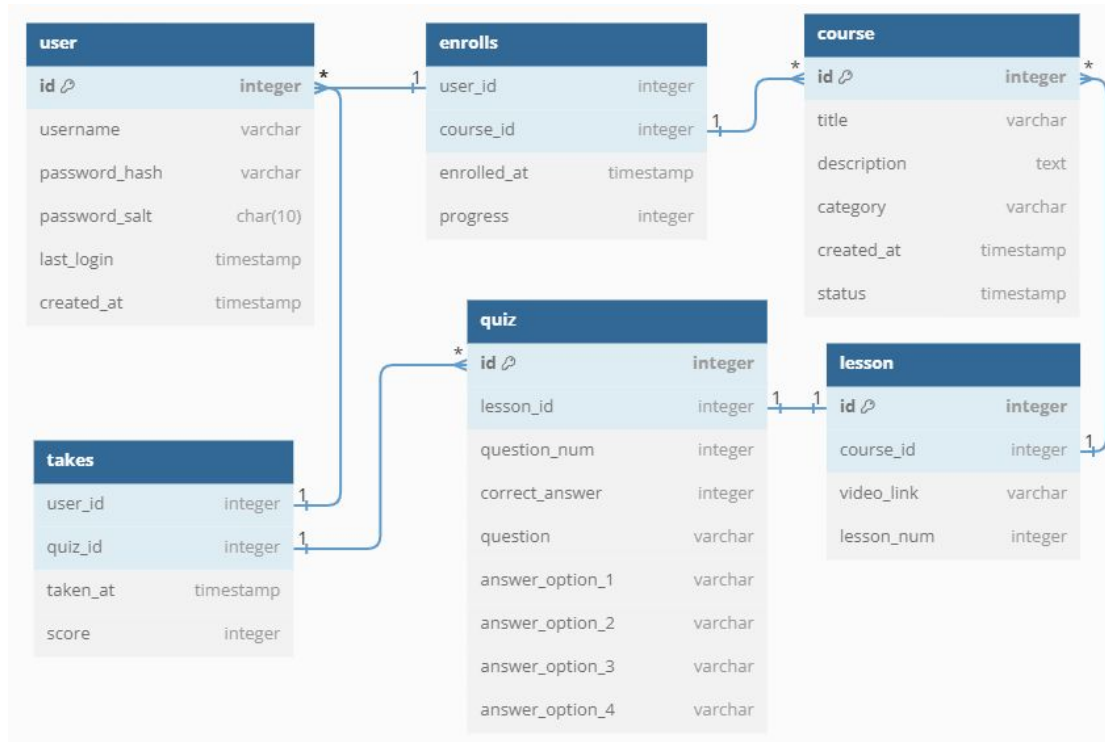
System Design - Databases
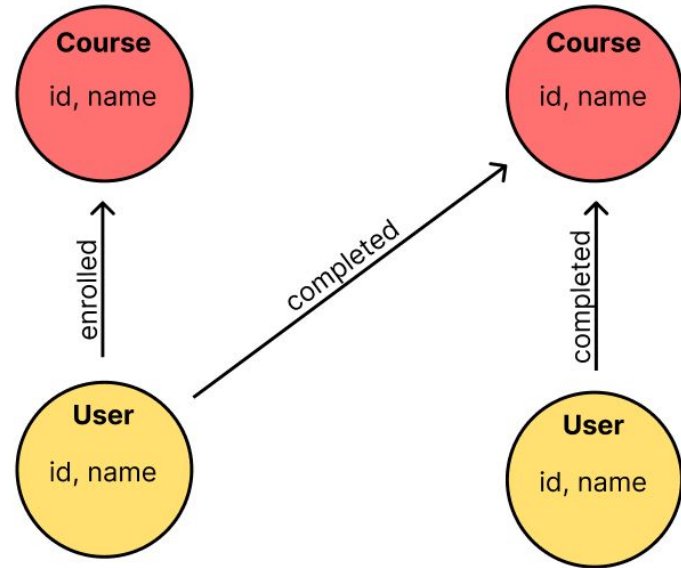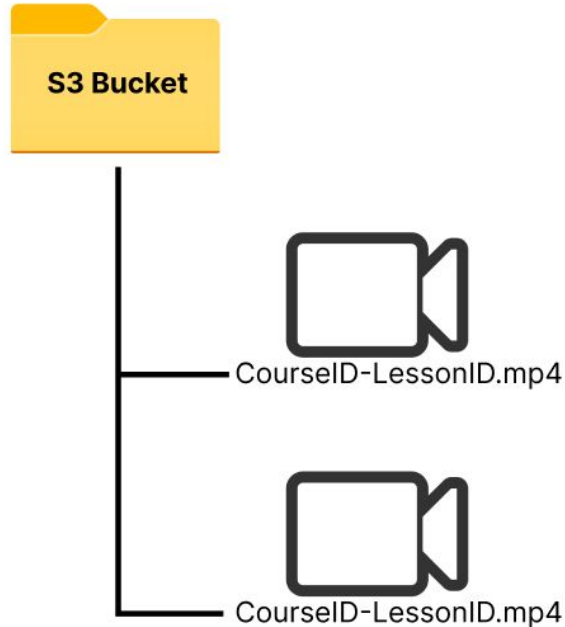
System Design - CI/CD Pipeline
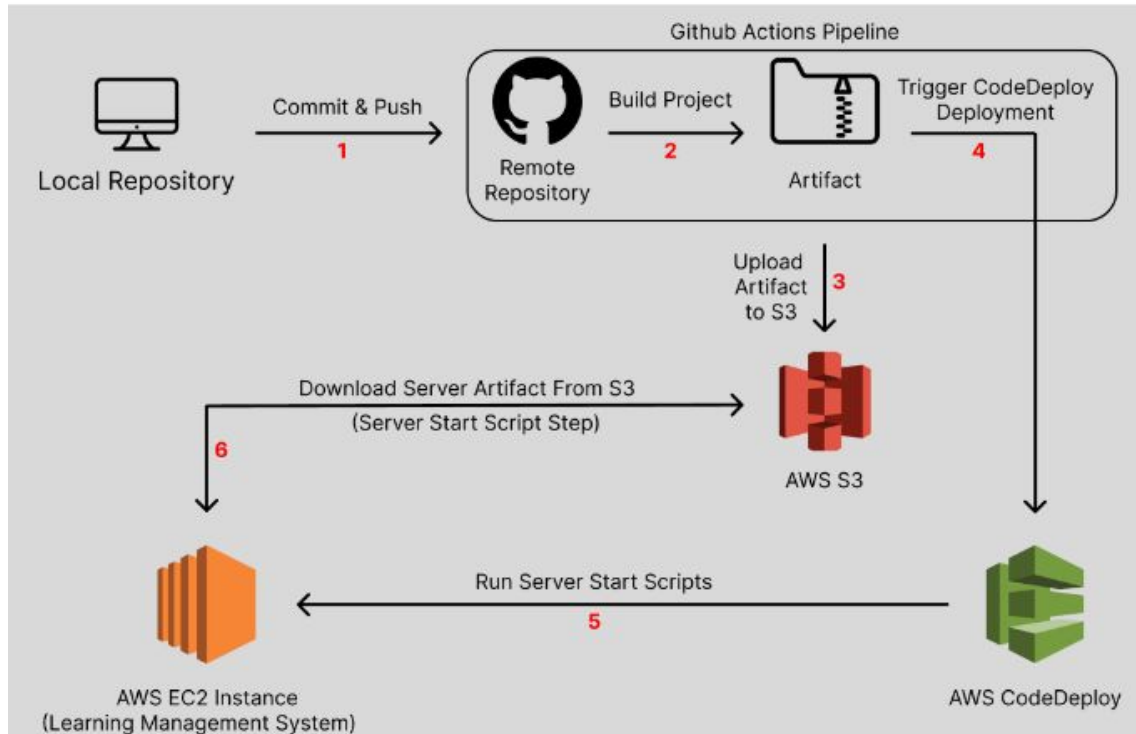
Work Progress

Backend

- Core APIs (User, Course, Lesson) [100%]
- Video Upload/Streaming [100%]
- Quiz Endpoints [10%]
  - Agree on final quiz model / dataflow
  - Implement Quiz API
- Course Recommendations [25%]
  - Find new place to host graph database
  - Implement a *simple* endpoint to retrieve course recommendations for a user
- Flashcards [0%]
  - Determine who has access to edit flashcards / where accessed from
  - Implement Flashcard API

Frontend

- Homepage [80%]
  - Improve formatting of text and add more information about the site
- Course Marketplace [60%]
  - Use courses API to interface with stored course information to populate the marketplace with course data.
  - Setup marketplace search bar.
- Course Viewer [60%]
  - Use lesson API and mock data to populate data and load videos on lesson select
  - Connect course cards in marketplace to direct to course viewer screen
- Login/Signup Pages [50%]
  - Make improvements to the login screen
  - Work on the Signup page

CI/CD Demo
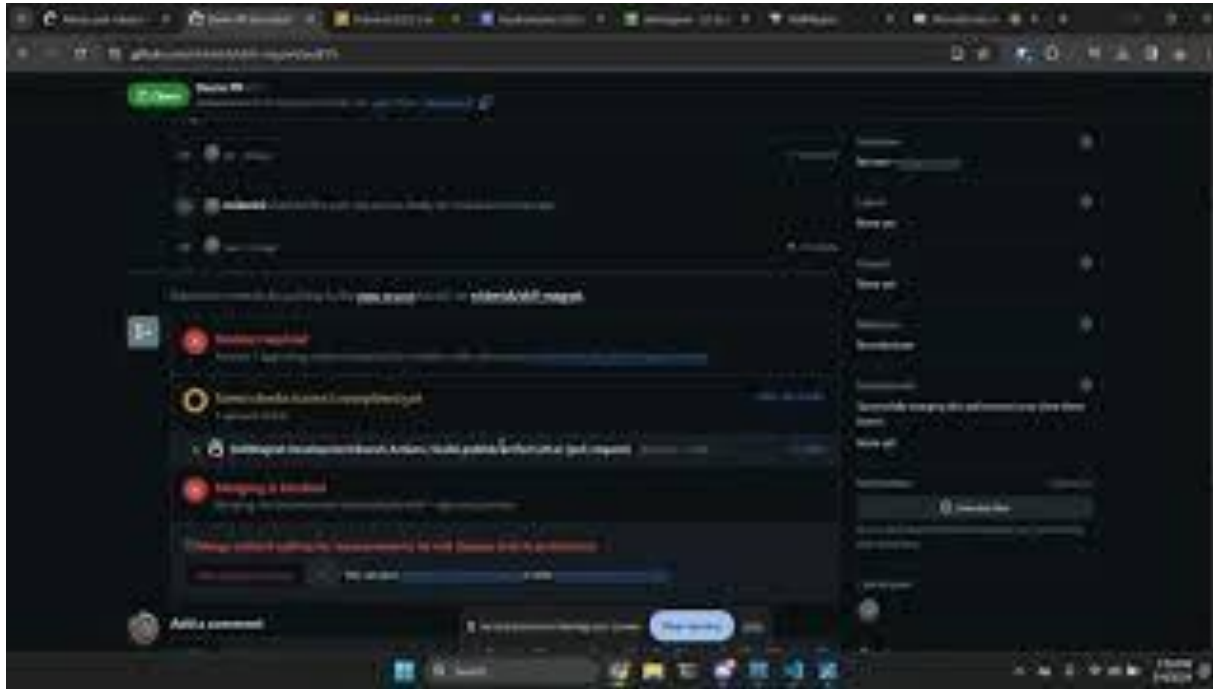
Backend Demo
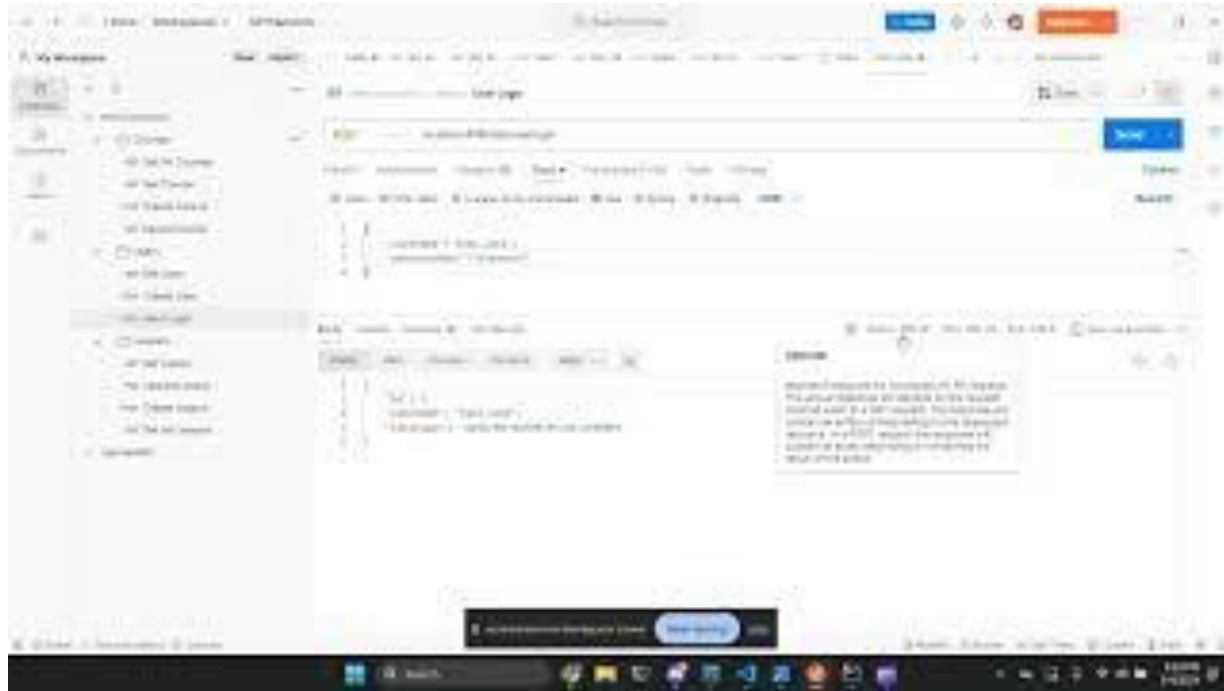
Frontend Demo
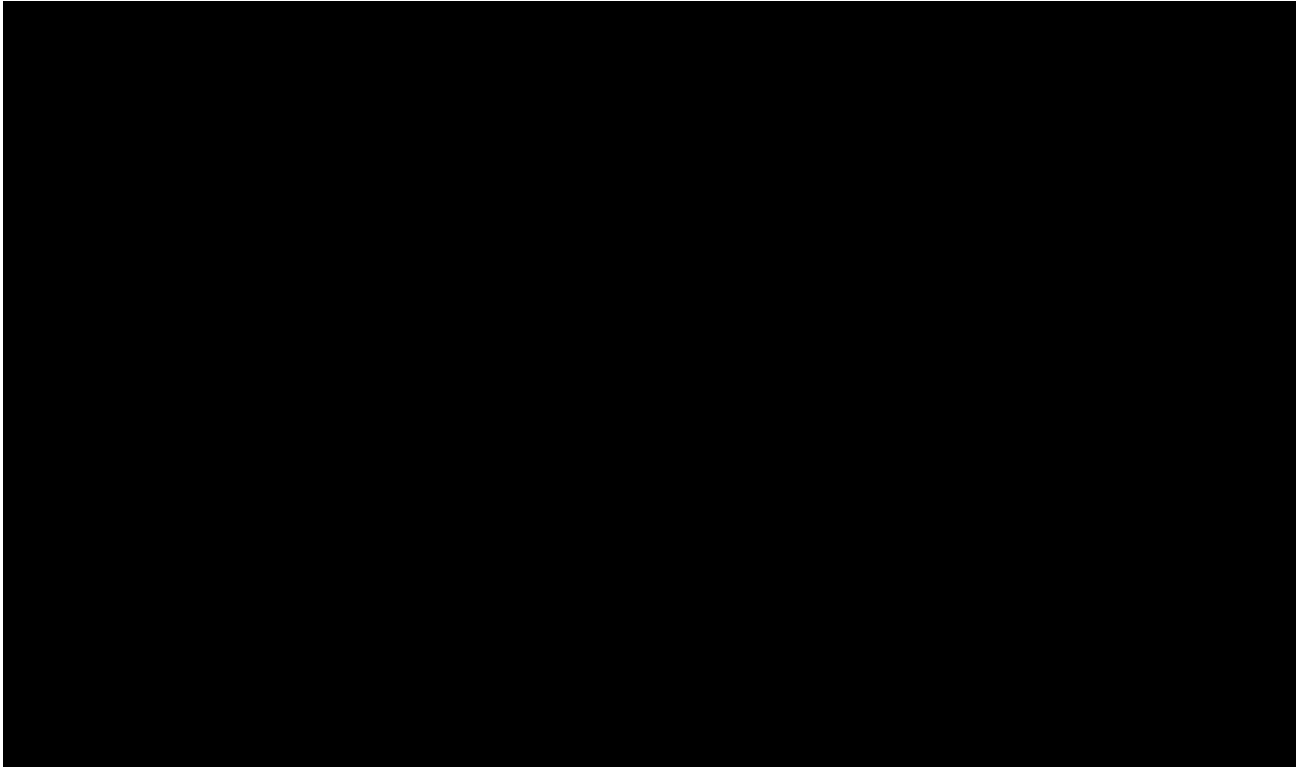
Challenges & Solutions

Resolved Challenges
- Playing video on client is difficult and expensive to transport large video files from a server
  - Discovered how to authenticate AWS S3 requests from React client
  - Override the request headers to stream the video from AWS rather than download
- When deciding how to hash passwords, there are many varying solutions ranging from out of date algorithms to expensive resource consuming algorithms.
  - Found an easy implementation through the Spring Security Library
  - Selected the Argon2 algorithm, which can be customized to match your use case.

Remaining Challenges
- When designing our course recommendations system we had planned to utilize the free tier of Amazon's Neptune graph DB, but Amazon recently added restrictions and it is no longer free.

Conclusion

Overall Progress

- Core APIs & Video Streaming Complete
- Frontend Pages 50-80% Complete

Remaining Work

- Use local mock data to populate web pages
- Connect navigation bar buttons with corresponding pages
- Setup course recommendations via the graph DB
- Build out Flashcards and Quizzes with APIs and decide on access rights

Team's Plan

- Make some major strides in the next few weeks in order to prepare the system for user feedback via surveys.
- Get the recommendations system up and running.
- Populate the frontend with live data.
- Creator Studio, Flashcards, and Quizzes to be implemented.
- If we check these boxes in the next few weeks, we can gather feedback from users, and have time to act on the feedback.