Architecture Design CodeX learner

TECH STACK

1. Frontend: React.js

React.js is a popular choice for building dynamic and responsive user interfaces. It offers a component-based architecture, which is ideal for creating the frontend of a complex application like an e-learning platform. React's virtual DOM enables efficient updates, and its large community provides a wealth of reusable components.

2. Backend: Node.js with Express.js

-Justification: Node.js is a suitable choice for building the backend. Express.js is a lightweight web framework for Node.js that simplifies routing, middleware management, and handling HTTP requests, making it well-suited for building RESTful APIs.

3. Database: PostgreSQL

PostgreSQL is a robust, open-source relational database management system. It provides excellent support for complex queries, which can be important for managing e-learning content, user data, and course information.

4. Authentication and Authorization: JSON Web Tokens (JWT)

- JWTs are a secure way to handle user authentication and authorization. They can be used to verify the identity of users and control access to various parts of the application, such as course content and instructor features.

5. Content Management System (CMS): Custom-built or Use an Existing CMS

Depending on the complexity of your content management needs, you can either build a custom CMS or utilize an existing one. A custom CMS allows you to tailor the content management experience to your exact requirements, while existing solutions like Strapi or WordPress can provide a head start.

6. Video Hosting: Embedded YouTube Videos

Embedding YouTube videos is a cost-effective and user-friendly way to handle video content. It offloads video storage and streaming to YouTube, which is a reliable platform for hosting and serving videos.

7. Cloud Storage: Amazon S3 (for other content)

-For non-video content like documents, images, and resources, Amazon S3 can be used for secure and scalable cloud storage. It allows you to store and serve these files efficiently.

8. Instructor Panel: Admin Dashboard

Create an admin dashboard for instructors to manage courses, upload content, and monitor student progress. You can build this panel using React.js or a frontend framework of your choice and secure it with appropriate authentication and authorization checks.

9. API Gateway (Optional):

Consider using an API gateway like NGINX or AWS API Gateway to manage API endpoints, enforce security, and handle rate limiting and caching. This can improve the scalability and security of your application.

10. Deployment and Hosting: Docker and a Cloud Provider {AWS}

Use Docker containers for consistent deployment and scalability. Choose a cloud provider based on your team's preferences and needs. These providers offer scalable infrastructure, automated deployment, and management tools that simplify hosting.

11. Version Control: Git

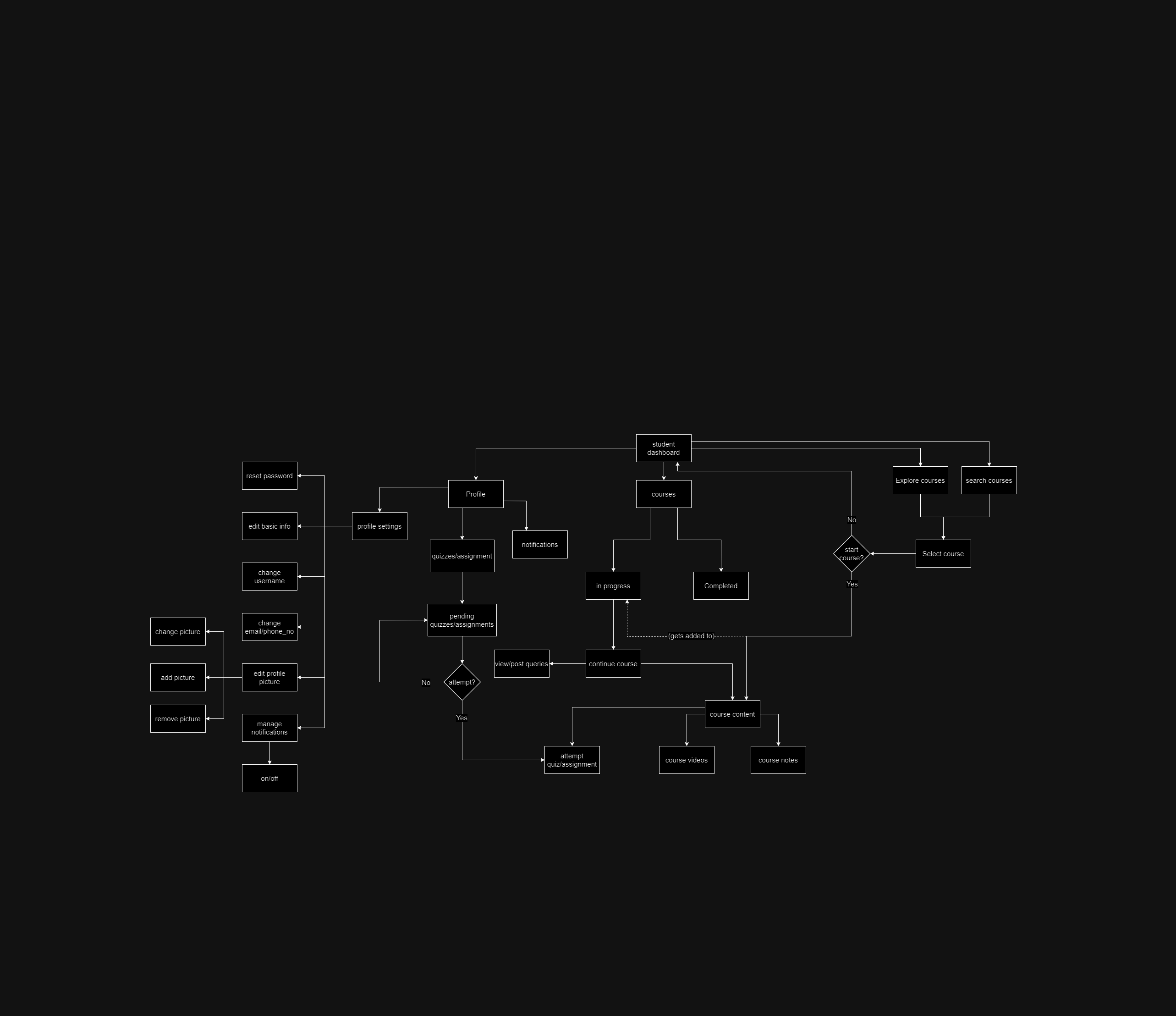
This architecture provides a solid foundation for building an e-learning platform that meets your requirements while ensuring scalability, security, and maintainability. Make sure to also implement best practices for security, performance optimization, and regular testing to ensure a robust and user-friendly application.

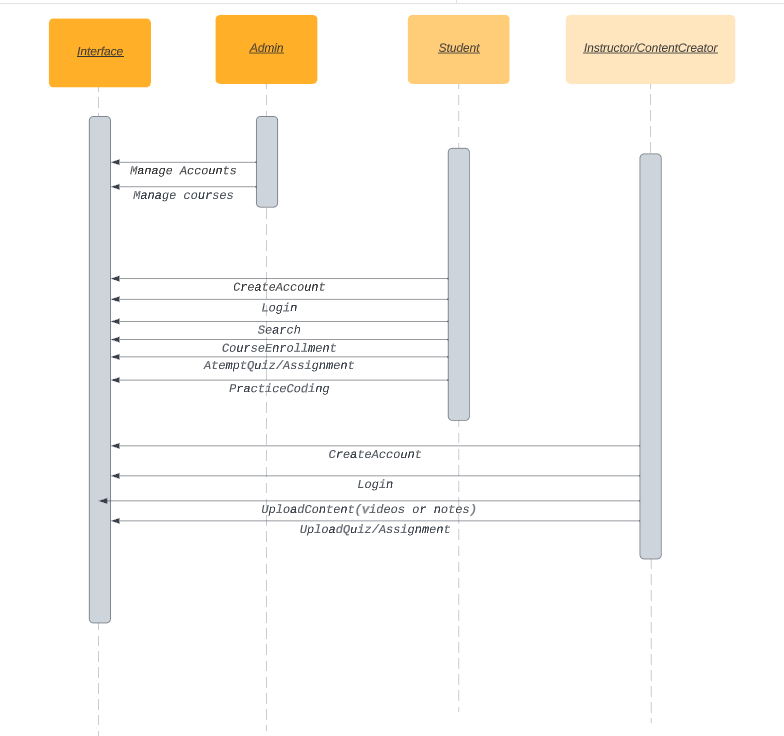
Use Case Diagrams

1. E learning System

2. Student system

3. Instructor System



****

