Architecture Design

Exam Proctor

Contents

Abstract 3
1 Introduction 4
1.1 Why this Low-Level Design Document? 4
1.2 Scope 4
1.3 Constraints 5
2 Technical specifications 5
2.1 UI/Development 5
2.2 Backend Development 5
2.3 AI Monitoring System 5
2.4 Database 5
2.5 Containerization **5**
2.6 GIT and CI/CD 6
2.7 Deployment 6
3 Proposed Solution 6
4 User I/O workflow 6
4.1 Creator Flow 7
4.2 Learner Flow 8
5 Architecture Design 9
6 Architectural Description 9

Abstract

During the covid and post covid period there is a lot of development in the e-learning platforms but there are very few portals which will evaluate the knowledge which is acquired by the learners. This work discusses the implementation of both creating content rich courses and how to implement a system which will monitor learners while they are taking exams online.

1 Introduction

1.1 Why this Architecture Design Document?

Any software needs the architectural design to represent the design of software. IEEE defines architectural design as “the process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system.” The software that is built for computer-based systems can exhibit one of these many architectures.

Each style will describe a system category that consists of :

- A set of components (eg: a database, computational modules) that will perform a function required by the system.
- The set of connectors will help in coordination, communication, and cooperation between the components.
- Conditions that how components can be integrated to form the system.
- Semantic models that help the designer to understand the overall properties of the system.

1.2 Scope

This is a web based application which will be able to add courses and also act as an online exam proctor which will monitor candidates with its AI based monitoring system. People can create courses and design exams where many can come and learn and examine their knowledge. Certification will be provided once the examination is completed.

1.3 Constraints

We will be only focusing mainly on the course creation and the exam monitoring part.

2 Technical specifications

2.1 UI/Development:

React js along with bootstrap is used for front end development. The main reason for going with react is for the reusability and light weight and bootstrap came out of the box for styling.

Draft.js is being used as an inline text editor to create courses.

Fetch and axios is used for making API interactions

2.2 Backend development

For backend i have used express and with node for creating REST API which will interact seamlessly with the UI and Mongo

2.3 AI monitoring system

For monitoring exams I have used python along with deepface and mediapipe and exposed it in REST API using flask.

2.4 Database

I have used mongo db Atlas which is a managed cloud database for storing and retrieving data.

2.5 Containerization

I have used docker and docker compose to containerize the application.

2.6 GIT and CI/CD:

I have used git for code maintenance and github actions to perform CI/CD operations

2.7 Deployment:

I have used the following for deployment purposes.

- Digital ocean droplets - Deploying our application
- Caddy - For Https, reverse proxy and load balancer
- Godaddy - For name server and domain

3 Proposed Solution

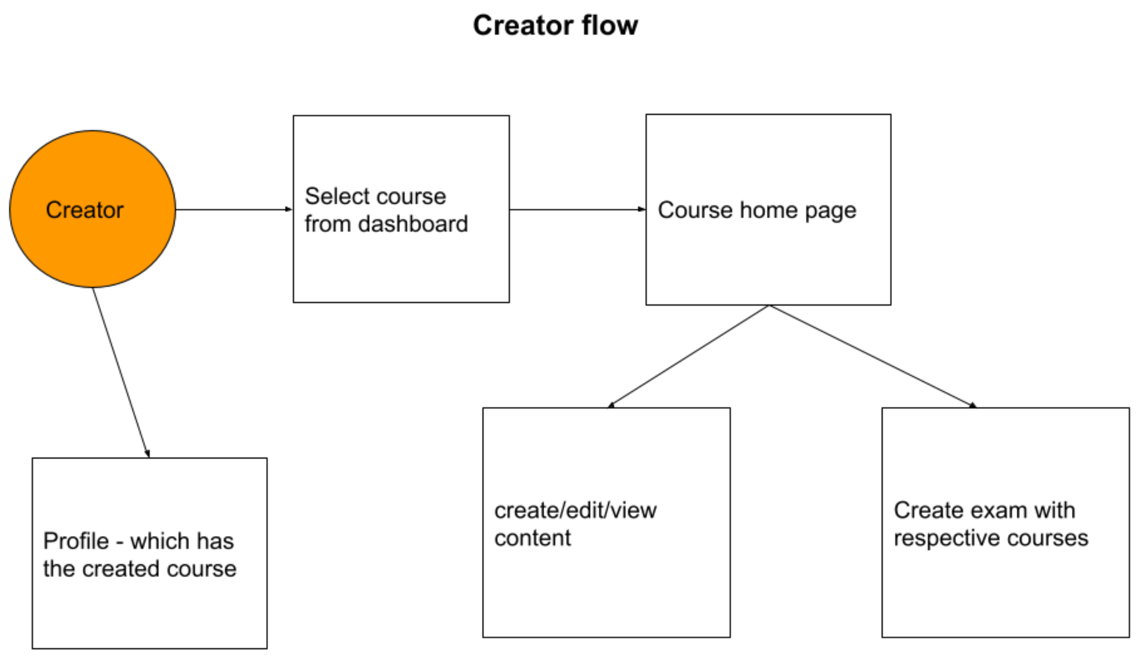
I will be creating a web based portal using mern stack and AI based automatic online proctor in order to monitor the candidates which will be written in python,deepface and mediapipe. The web portal will be interacting along with the monitoring system in order to examine the candidates.

4 User I/O workflow

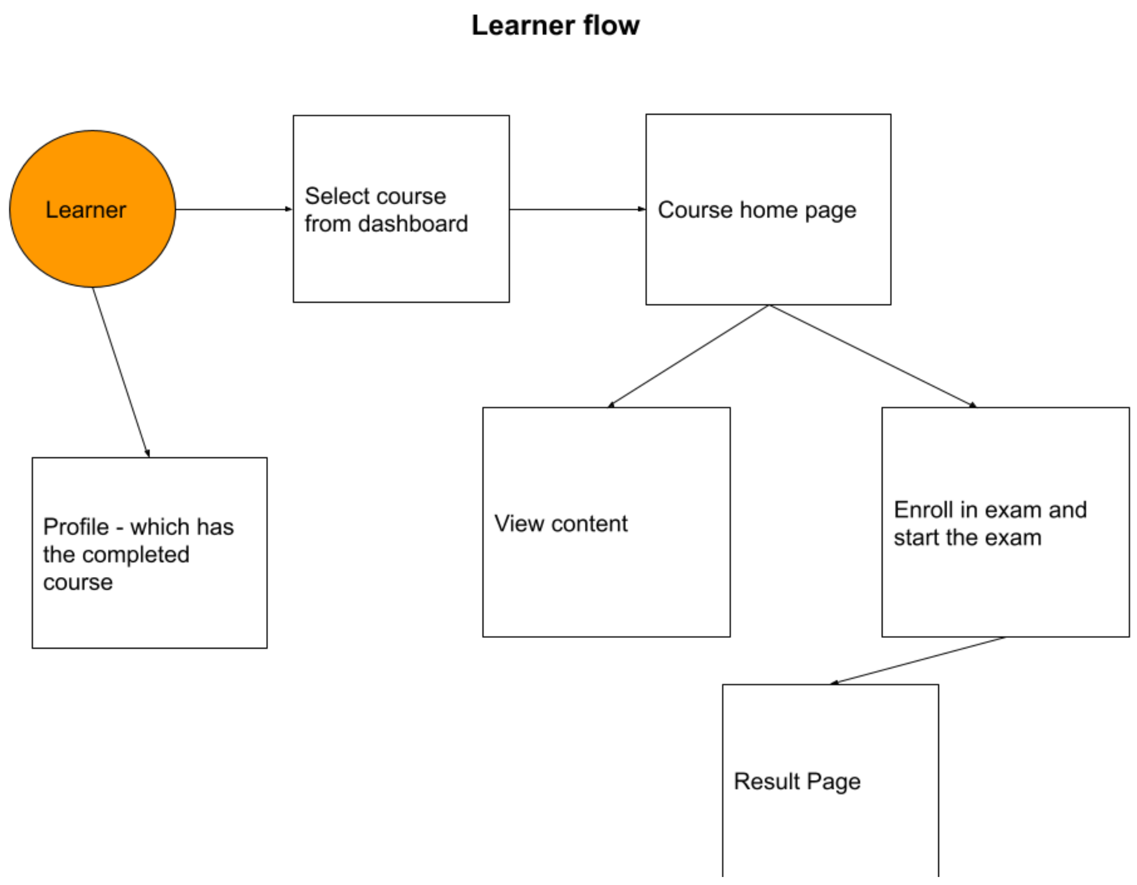
Since there are two types of users in our application

1) Creator- who creates courses
2) Learner - who learns and take exam

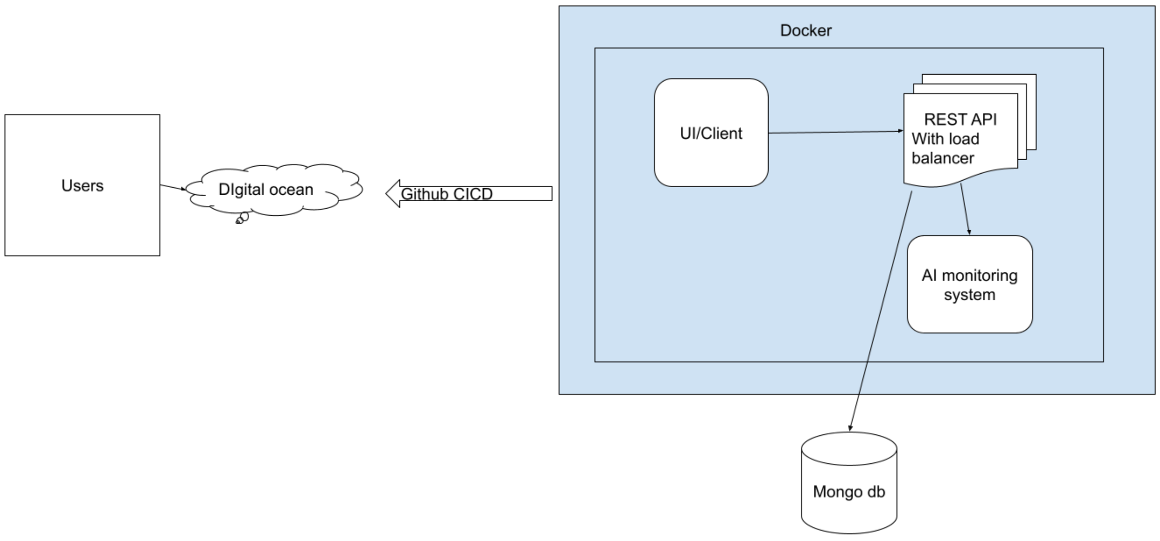
4.1 Creator flow


Creator flow
Creator
Select course from dashboard
Course home page
Profile - which has the created course
create/edit/view content
Create exam with respective courses

4.2 Learner flow


Learner flow
Learner
Select course from dashboard
Course home page
Profile - which has the completed course
View content
Enroll in exam and start the exam
Result Page

5 Architecture design:



6 Architectural description

1. Frontend and backend communicate via Rest API and the results are stored in the mongo db.
2. For storing course i am using mongodb grid which acts file a file system to store and retrieve files which may be generated above 16 mb
3. I containerized the application using docker file and serve it using docker-compose
4. I use caddy for reverse proxy and applying load balancer to our backend application
5. I have used JWT based authentication to validate users
6. I have used github for code maintenance and github actions for cicd.
7. Digital ocean droplets is used for hosting and deploying our docker container
8. Godaddy is used for name server and domain management.