iNeuron

High Level Design(HLD)

Exam proctor (Online exam monitoring system)

Document Version Control

Data issued	Version	Description	Author
08/06/22	1.0	Initial HLD	Yashwant

Table of Content

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 High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

- Present all of the design aspects and define them in detail
- Describe the user interface being implemented
- Describe the software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project
- List and describe the non-functional attributes like:
    - Reliability
    - Maintainability
    - Portability
    - Reusability
    - Application compatibility
    - Resource utilisation
    - Security
    - Serviceability

1.2 Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

1.3 DEFINITIONS

Term	Description
Database	Collection of all the informations monitored by the system
Digital ocean	Web service provided

2 General Description

2.1 Product Perspective

Exam proctor is software which is used to monitor exam candidates through the internet and perform automated actions accordingly.

2.2 Problem statement

Build a web based platform to conduct exams using multiple choice questions. Well recognized organisations such as AWS, Microsoft allowing users to schedule exams based on their availability and their preferred location. In some cases they allow users to schedule an exam at home.

Building a secure platform to evaluate candidates effectively is really important. The Exam Proctor portal should be designed in such a way that it can capture any unwanted activity done by the user and automatically the system should take effective and efficient measures to avoid such activities.

2.3 Proposed solution

The proposed solution is called SLAT (Application name). It is a web based portal where people can create courses and also learn and evaluate themself using MCQ exams which are monitored and consolidated using AI.

The following are the important features of SLAT:

- Create text rich learning content for specific roles
- Learners can learn through content for free and payment will be made only for the examination
- Learners will be monitored in the following ways
    - AI based face matching system
    - AI based head pose estimation
    - Tab switching
    - Maximum attempts
    - Completion of exam in specific time limit
    - Auto submit in case of any malpractice

2.4 Further Improvements

The following factors can be improved:

- UX/UI can be improved to be more interactive
- Allowing used to add video as educational content
- For now all courses are available at 99 rupees. It can be made dynamic
- Adding admin panel

2.5 Technical Requirements

This document addresses the requirements for a web portal where people can create courses and take exams which will be monitored using an AI system.

- A course creator can add a course and create examinations for the courses.
- A Course creator should have a dashboard where he can edit the courses and examinations created by him.

- A course creator should have a dashboard where he can check the number of enrollments happening in his course.
- A learner should have a dashboard where he can search courses according to the role.
- A learner should be able to view content created by the creators for free
- A learner should be able to enrol in an exam associated with any course which he likes to take.
- Payment gateway should be implemented during enrollment
- Monitoring system must be implemented which will auto submit exam when malpractice is found
- Learner must be able to download certificates after course completion

2.6 Tools used

- For web portal
  - Mongo - Nosql database for storing data
  - Express and Node - Create API for communicating with client/front-end
  - React - Frontend/UI library based on javascript for building UI
  - Bootstrap - CSS framework for UI development
  - Draf.js - Used to create a web based text editor where creators can create interactive text rich content. Ref link:(https://draftjs.org/)
  - Caddy - For reverse proxy and load balancing. Ref link(https://caddyserver.com/)

- For monitor system
  - Python - For interacting with the machine learning/deep learning libraries and creating API using flask
  - Deepface - Lite weight face recognition package Ref link: (https://github.com/serengil/deepface)
  - Mediapipe - customizable ML solutions for live and streaming media. Ref link:

(https://google.github.io/mediapipe/)

- Tools used for development
    - Vs code - code editor
    - Compass - GUI for mongodb
    - Atlas - Managed cloud database service
- GIT/Code maintenance
    - Github
- CI/CD
    - Github actions
- Deployment
    - Docker and Docker compose - For application containerization
    - DigitalOcean droplets,name services
    - Godaddy
- Payment gateway
    - Razerpay

2.7.1 Hardware Requirements

- Web camera

2.7 Constraints

The exam proctor application must be user friendly and allow users to create and take exams securely.

2.8 Assumptions

The main objective of the application is to fulfil the requirements mentioned in 2.5.

3 Design Details

3.1 Process Flow

- For creating course we have used react based text editor to create interactive content
- Payment gate is used for course enrollment
- AI is used to monitor candidates while taking exam and auto

submit the exam incase of malpractice
- Learners can download certificate once the course is completed

3.2 Error Handling

- Docker logs is used to monitor the system logs
- Error notification will be shown to the user via toaster messages when something goes wrong.

3.3 Performance

- Exam proctor solution is used for creating courses and mainly monitoring exam candidates while taking exams so the monitoring system used should perform real time and since face analysis plays a main role in it.
- Data privacy and data storage is handled properly.
- Implemented a text editor where people can create blog-like content for their courses and learners can view those content.
- Realtime payment gateway is used for enrollment purposes.
- JWT authentication is being used

3.4 Reusability

The code written and the components used should have the ability to be reused with no problems.

3.5 Resource Utilisation

Load balancer is being used to get the full performance of the server/droplet.

3.6 Deployment

- Digital ocean is being used as the cloud provided to host and deploy the application.
- Github Actions and docker are used to create CI/CD pipelines.
- Caddy is used to create a reverse proxy and implement load balancer.

4 Conclusion

The Designed Exam Proctor (SLAT) will be able to create courses and conduct exams and take actions using AI based monitoring systems. Which will improve the knowledge quality of everyone.

5 References

MERN Stack - https://www.mongodb.com/mern-stack