PES UNIVERSITY, BANGALORE
Department of Computer Science and Engineering

B.TECH. (CSE)

VI SEMESTER

UE20CS352 – Object Oriented Analysis and Design with Java

Mini-Project Report

on

Kaun Banega Crorepati: An AI-based Quiz Resource

SUBMITTED BY

Name	SRN	Roll No
Yashas R Kashyap	PES1UG20CS519	6
Vijit Kumar	PES1UG20CS500	2
Vishal J Lodha	PES1UG20CS507	3
Yash Vardhan Soni	PES1UG20CS518	1

Section I

January – May 2023

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

BENGALURU – 560100, KARNATAKA, INDIA

PES UNIVERSITY, BANGALORE
Department of Computer Science and Engineering

Project Synopsis:

Project Synopsis:

The proposed project is a web-based application that allows users to take a quiz consisting of five questions. The application uses the React front-end framework and the Spring Boot back-end framework to build a responsive, scalable, and secure platform.

The user first needs to login to the website using their username and password. Once logged in, the user can start the quiz. The quiz is generated based on the user's preferred difficulty level: easy, medium, or hard. The questions are fetched from the OpenAI API called DALLE, which uses natural language processing (NLP) to generate questions.

The user reads the question and selects their answer from the available choices. After answering all five questions, the user submits their answers to the server for grading. The server processes the answers and returns the user's score.

The user can view their score along with the correct answers and explanations for the questions they got wrong. The application also provides feedback to the user on their performance and suggests ways to improve their score.

The project aims to provide an engaging and interactive way for users to test their knowledge and learn new things. The use of NLP technology in generating questions makes the quiz more challenging and interesting. The application's responsive and scalable design ensures that users can access the quiz from any device and that the system can handle multiple users simultaneously. The security features of the Spring Boot framework ensure that user data is protected and that the application is safe to use.

Features:

- *User Authentication and Authorization*

  The project includes a user authentication and authorization system that allows users to create an account, login, and logout. The authentication system uses Spring Security, which provides authentication and access control features out of the box. When a user logs in, Spring Security creates a session and sets a session ID in the user's browser. The session ID is used to authenticate subsequent requests from the user.

- *Integration with OpenAI API*

  The project integrates with the OpenAI API called DALLE, which generates natural language questions. The integration involves making HTTP requests to the DALLE API using the Spring RestTemplate class. The RestTemplate class is configured to send requests with the correct headers and parameters, and to parse the JSON responses returned by the API.

- *React-based User Interface*

  The project's user interface is built using the React front-end framework, which provides a fast and responsive interface for the user. React components are used to build the quiz form, display the questions and answers, and provide feedback to the user. The user interface is designed to be accessible and easy to use, with clear instructions and intuitive controls.

- *Server-side Quiz Grading*

The project grades the user's quiz answers on the server side using Spring Boot. When the user submits their answers, the server compares the user's answers with the correct answers and calculates a score. The score is returned to the user along with feedback on their performance. The server also stores the user's quiz results in a database for future reference.

- *Scalable Architecture*

  The project is designed to be scalable, with a modular architecture that separates the front-end and back-end components.

Use cases:

Here are some possible use cases for this project, given that the quiz contains general knowledge questions:

1. Educational institutions
Educational institutions such as schools, colleges, and universities can use this project as a tool for assessing their students' knowledge in various subjects. Teachers can create custom quizzes based on the subjects they teach and assign them to their students. The quiz results can help teachers identify areas where their students need more help and provide feedback to improve their performance.

2. Training and development programs
Organizations that provide training and development programs for their employees can use this project to assess their employees' knowledge in various domains. The quiz can be customized to match the organization's training curriculum, and the quiz results can be used to identify areas where employees need further training.

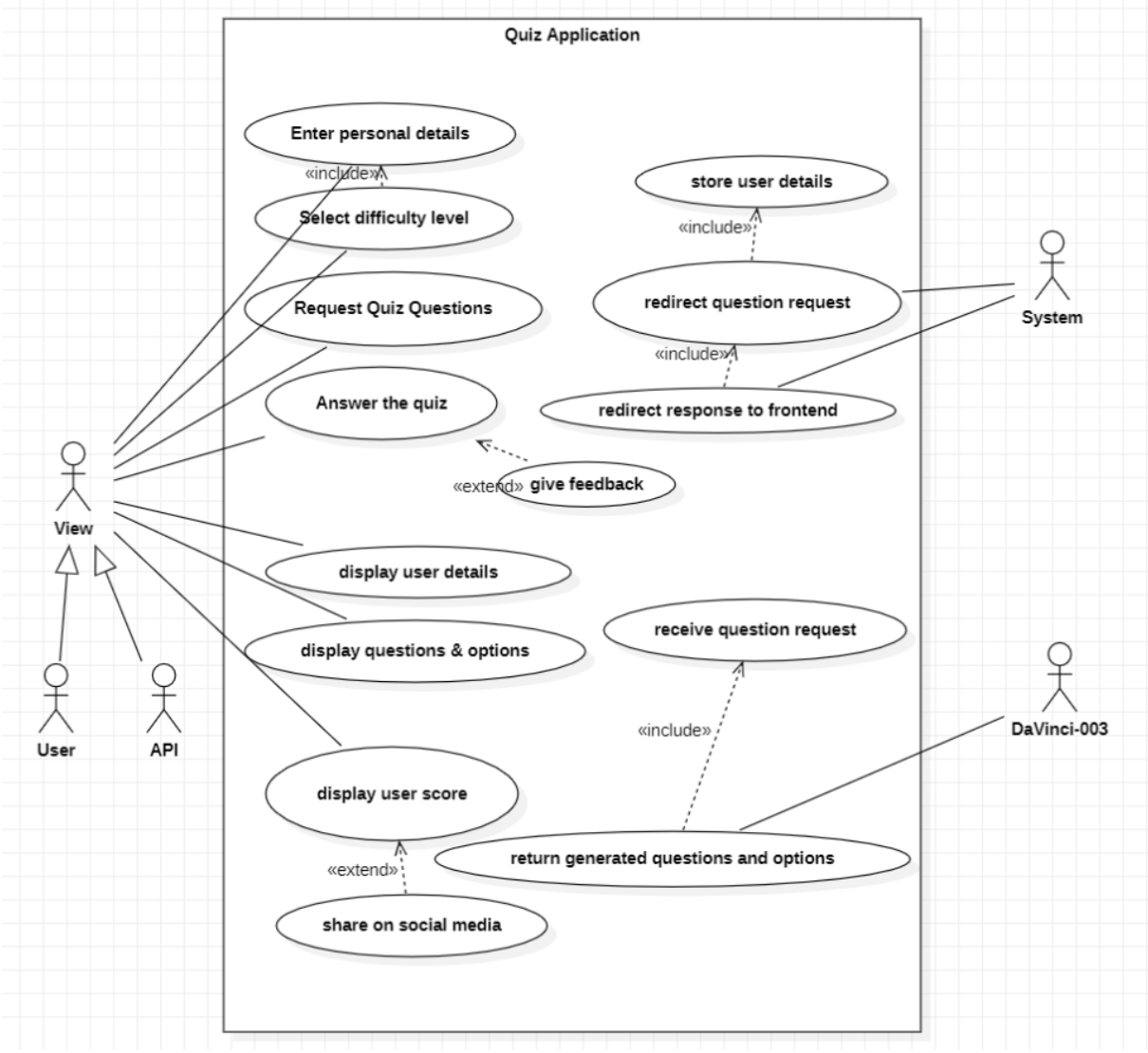
3. Online education platforms
Online education platforms can use this project as a tool for assessing their learners' knowledge in various subjects. The quiz can be integrated into the platform's learning management system (LMS) and used to evaluate learners' progress. The quiz results can be used to recommend personalized learning paths for each learner based on their strengths and weaknesses.

4. Recruitment and selection processes(UPSE)
Organizations can use this project as part of their recruitment and selection processes to evaluate candidates' general knowledge. The quiz can be customized to match the job requirements and used to filter out candidates who do not meet the minimum knowledge criteria.

5. General entertainment and learning
Individuals can use this project as a fun and engaging way to test their knowledge and learn new things. The quiz can be accessed from anywhere with an internet connection and used to challenge oneself and others. The quiz results can be used to track one's progress and identify areas where one needs further learning.

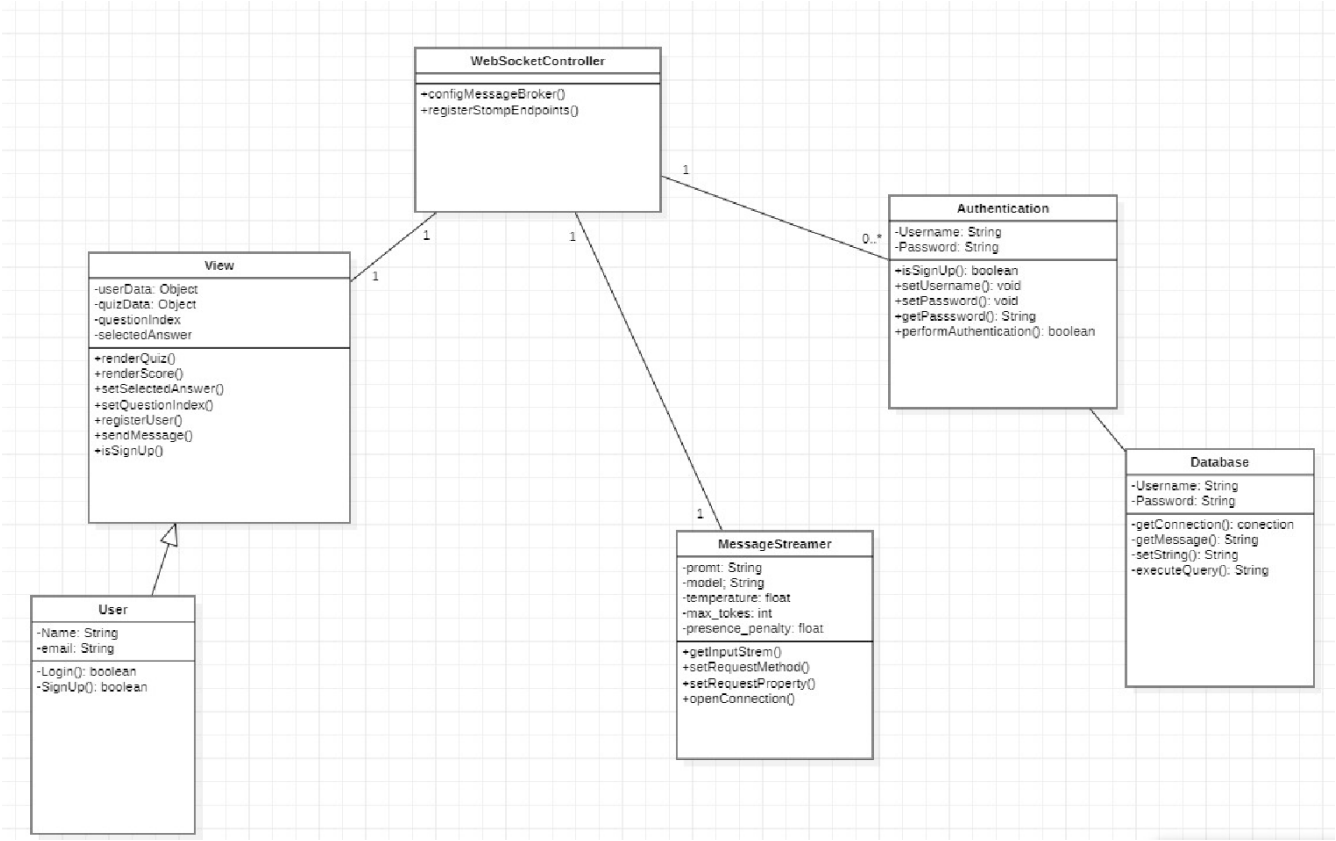
Models

Use Case Model



PES UNIVERSITY, BANGALORE
Department of Computer Science and Engineering

Class Model (based on MVC)



Architectural Pattern

Model-View-Controller

MVC (Model-View-Controller) is a software architectural pattern commonly used in developing user interfaces for web applications. The pattern separates an application into three interconnected components: the model, the view, and the controller.
The Model component represents the data and the business logic of the application. It is responsible for managing the data and performing operations on it.

The View component represents the user interface of the application. It is responsible for presenting the data to the user in a way that is understandable and meaningful.

The Controller component acts as an intermediary between the Model and the View. It receives input from the user and manipulates the Model to perform the requested operation. It also updates the View with the changes made to the Model.

Using the MVC pattern, developers can create software that is modular, reusable, and easy to maintain. It also allows for easier testing and debugging, as each component can be tested separately.
MVC is widely used in developing web applications, but it can also be used

in other software development projects where there is a need to separate the presentation layer from the business logic and data.

MVC has been applied in our project in the following manner:

Model:

- performAuthentication is the primary model of our project
- SpringBoot framework was used for the backend

View:

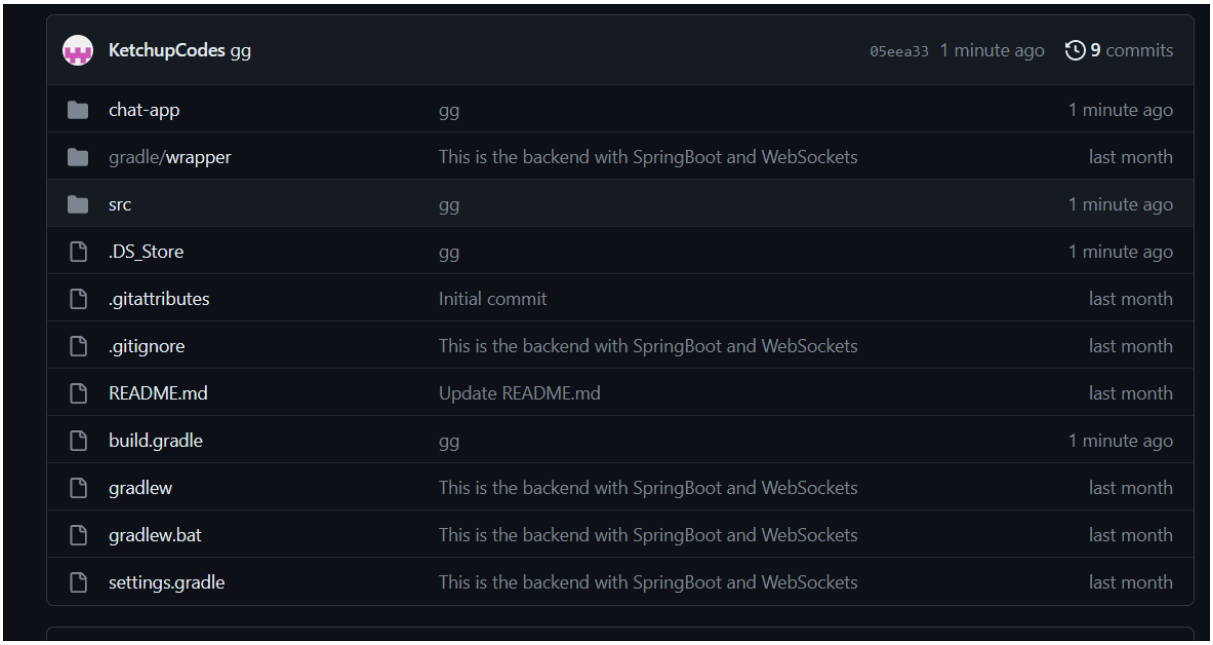
Front-end using ReactJS + CSS

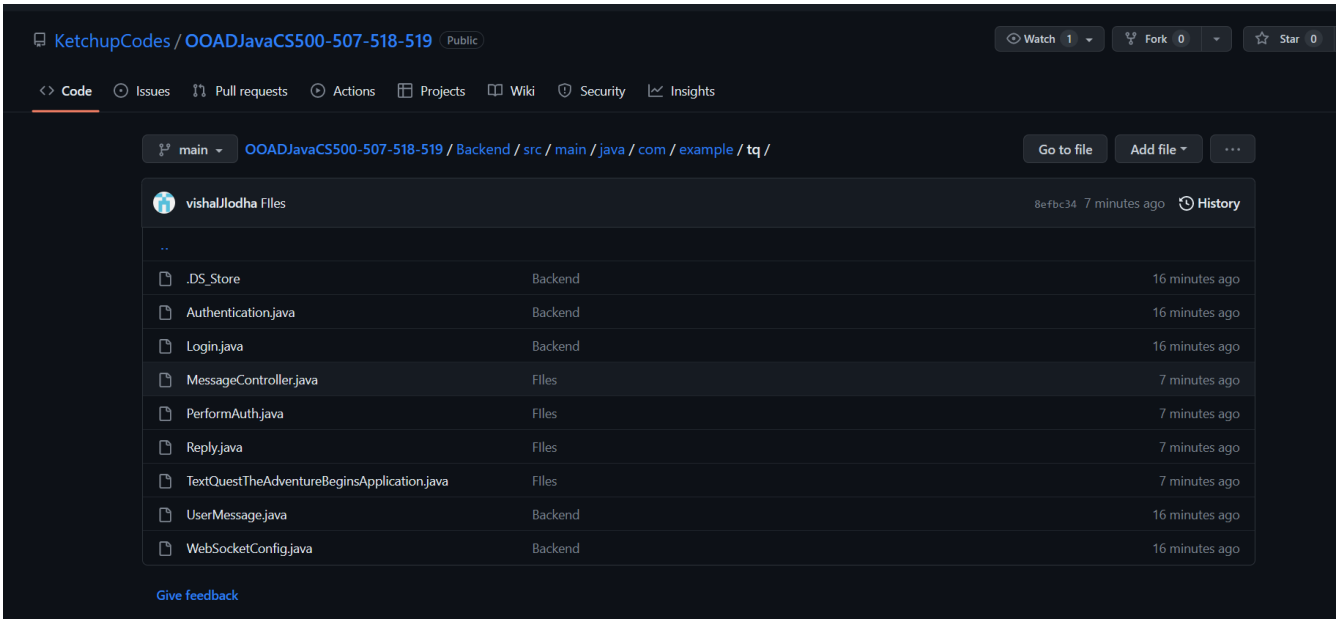
Controller:

The WebSocket acts as the primary controller and allocates tasks to each model.

PES UNIVERSITY, BANGALORE
Department of Computer Science and Engineering

Snapshot of the directory structure of our project which gives a clear demarcation of MVC models.





PES UNIVERSITY, BANGALORE
Department of Computer Science and Engineering

Design Principle and Patterns

SOLID

1. SRP – Single Responsibility Principle

   The performAuthentication class is purely and solely responsible for verifying the credentials of the user.

2. ISP – Interface Segregation Principle

   The interface behaves differently for an existing user and new user, on the basis of the UserState.

3. Open-Closed Principle

   The Login class extends the performAuthentication class, without making any modifications to it. It just adds its own additional features.

GRASP

1. Controller

   There is a controller class called WebSocketConfig to allocate tasks to all adequate models.

2. Indirection

   The same CSS code is used for all pages of the web application.

3. Low Coupling and High Cohesion

   All tasks are capable on functioning independently and work in tandem with each other.

Design Patterns (Structural)

1. Bridge

   The ReactJS frontend implements the bridge between the user and the backend.

2. Flyweight

   The Reply class object is passed on and reused multiple times by the MessageController and the WebSocket.

GitHub link to Code Repository - https://github.com/KetchupCodes/OOADJavaCS500-507-518-519

Individual Contribution

Yashas R Kashyap	Backend
Vijit Kumar	Front-end
Vishal J Lodha	Authentication
Yash Vardhan Soni	Database