PYTHON PROJECT
INT216
FINAL PROJECT REPORT

Topic: QuizApp

Submitted By: **Submitted To:**

Muhammad Yousuf Khan   Ved Prakash
Section - K22UP   Chaubey
Roll no. : 45   UID: 63892
12204116

DECLARATION

I hereby declare that the project work entitled ("QuizApp") is an authentic record of my own work carried out as requirements of Capstone Project for the award of B.Tech degree in Computer Science and Engineering (AI & ML) from Lovely Professional University, Phagwara, under the guidance of **Ved Prakash Chaubey** sir, during September to November 2023. All the information furnished in this capstone project report is based on my own intensive work and is genuine.

Name of Student : Muhammad Yousuf Khan
Registration Number: 12204116

_____
Signature of Student
Date: 06/11/23

CERTIFICATE

This is to certify that the declaration statement made by this student is correct to the best of my knowledge and belief. He has completed this Capstone Project under my guidance and supervision. The present work is the result of their original investigation, effort and study. No part of the work has ever been submitted for any other degree at any University. The Capstone Project is fit for the submission and partial fulfillment of the conditions for the award of B.Tech degree in Computer Science and Engineering (AI & ML) from Lovely Professional University, Phagwara.

Signature and Name of the Mentor:
Designation:
School of Computer Science and Engineering, Lovely Professional University, Phagwara, Punjab.
Date :

ACKNOLEDGEMENT

ABSTRACT

The Quiz Application is a comprehensive platform developed using cutting-edge technologies such as React Native for the frontend, FastAPI Python for the backend, and MongoDB as the database. The application is designed to provide an interactive and engaging experience for users who are interested in testing their knowledge across various topics.

The home screen of the application is user-friendly and displays a list of available quizzes. It also includes a search bar, allowing users to find quizzes based on their interests. This feature enhances the usability of the application and allows users to navigate through the quizzes effortlessly.

Upon selecting a quiz, users are directed to the quiz page where they can attempt multiple-choice questions. After answering all the questions and clicking on submit, a modal displays the user's score. This instant feedback mechanism motivates users to improve their performance and learn from their mistakes.

One of the unique features of the application is the ability to create new quizzes. Users can set any number of questions and their correct answers. After submitting the quiz, it gets uploaded to the database and appears on the home screen, ready for other users to attempt. This feature fosters a sense of community among users and encourages knowledge sharing.

The application also includes a profile page where users can view their details. This personal touch makes the application more engaging and gives users a sense of ownership.

In conclusion, the Quiz Application is not just a tool for learning, but also a platform for users to share their knowledge, track their progress, and engage with a community of learners. The application's design principles focus on usability, interactivity, and community-building, making it a fun and educational tool for users.

Introduction

The Quiz Application is an interactive platform developed using React Native, FastAPI Python, and MongoDB. It provides a dynamic environment for users to participate in quizzes across various topics, create their own quizzes, and track their progress. The home screen displays a list of available quizzes and a search bar for easy navigation. Upon selecting a quiz, users can attempt the questions and view their score instantly. The application also allows users to create new quizzes, fostering a sense of community and encouraging knowledge sharing. A profile page is included for users to view their details, adding a personal touch to the application. The Quiz Application aims to provide a fun and educational tool for users.

Problem Statement

The problem this project addresses is the lack of an interactive platform that allows users to participate in quizzes across various topics, create their own quizzes, and track their progress, all in one place. The application aims to provide a fun, educational, and community-driven tool for learning.

Problem Statement Solution Approach

The solution to the problem statement is the development of an interactive Quiz Application. The application is designed using React Native for the frontend, FastAPI Python for the backend, and MongoDB as the database. These technologies were chosen for their robustness, scalability, and ease of use.

The application provides a dynamic environment where users can participate in quizzes across various topics. The home screen displays a list of available quizzes and a search bar for easy navigation. This design ensures a user-friendly experience and encourages active participation.

A unique feature of the application is the ability for users to create their own quizzes. This fosters a sense of community and encourages knowledge sharing. The quizzes are then added to the database and made available for other users to attempt.

Users can track their progress through an instant feedback mechanism. After attempting a quiz, the user's score is displayed, providing an opportunity for self-assessment and learning.

The application also includes a profile page where users can view their details. This personal touch enhances user engagement and provides a sense of ownership.

In conclusion, the Quiz Application addresses the problem by providing a fun, educational, and community-driven platform for learning. It leverages modern technologies to deliver a smooth and engaging user experience.

Methodology

The methodology for developing the Quiz Application involved several steps:

Requirement Analysis: The first step was to understand the requirements and define the problem statement. The need for an interactive platform for users to participate in quizzes, create their own quizzes, and track their progress was identified.

Design: The application was designed with a user-friendly interface. The home screen displays a list of available quizzes and a search bar. The quiz page allows users to attempt the quiz and view their score. A feature to create new quizzes was also included.

Development: The application was developed using React Native for the frontend, FastAPI Python for the backend, and MongoDB as the database. These technologies were chosen for their robustness, scalability, and ease of use.

Testing: The application was thoroughly tested to ensure its functionality and performance. Any bugs identified were fixed, and improvements were made based on feedback.

Deployment: After successful testing, the application was deployed and made available for users.

This methodology ensured a systematic approach to the development of the Quiz Application, resulting in a robust, user-friendly, and interactive platform for learning.

Result and Analysis

The Quiz Application was developed using a combination of React Native for the frontend, FastAPI Python for the backend, and MongoDB as the database. The code was written in a modular and scalable manner, ensuring that each component of the application could be developed and tested independently before being integrated.

The frontend was developed using React Native, a popular framework for building mobile applications. The code for the frontend was structured around various screens such as the home screen, quiz page, and profile page. Each screen was developed as a separate component, allowing for easy testing and debugging.

The backend was developed using FastAPI Python, a modern, fast (high-performance), web framework for building APIs with Python based on standard Python type hints. The code for the backend was organized into different modules such as controllers, models, and routes. This modular approach made it easy to manage the complexity of the backend code.

MongoDB, a NoSQL database, was used to store the quizzes and user data. The choice of MongoDB allowed for flexibility in the data model and scalability as the number of quizzes and users grew.

The application was tested thoroughly during the development process. Various test cases were written and executed to ensure that all components of the application were working as expected. The tests covered various scenarios such as creating a new quiz, attempting a quiz, and viewing the score.

The analysis of the code and test results showed that the application was functioning as expected. The performance of the application was satisfactory, and the user experience was smooth and intuitive. The modular and scalable code structure ensures that the application can be easily maintained and enhanced in the future. Overall, the development of the Quiz Application was a success.

Conclusion

In conclusion, the Quiz Application serves as an interactive platform for users to participate in quizzes, create their own quizzes, and track their progress. Developed using React Native, FastAPI Python, and MongoDB, the application provides a user-friendly interface and a dynamic environment for learning. The application's design principles focus on usability, interactivity, and community-building, making it a fun and educational tool for users. The code for the application is modular and scalable, ensuring easy maintenance and potential enhancements in the future. The successful development and testing of the Quiz Application demonstrate the effectiveness of the chosen technologies and the robustness of the application. Overall, the Quiz Application stands as a testament to the potential of modern technologies in creating engaging and educational platforms.

References

React Native Documentation: Official documentation for React Native, providing a detailed guide on how to use the framework for building mobile applications.

FastAPI Python Documentation: Official documentation for FastAPI Python, providing a comprehensive guide on how to use the framework for building APIs.

MongoDB Documentation: Official documentation for MongoDB, providing instructions on how to use the NoSQL database for storing and retrieving data.

Python Documentation: Official documentation for Python, providing a detailed guide on the Python programming language.

JavaScript Documentation: Official documentation for JavaScript, providing a comprehensive guide on the JavaScript programming language.