RPS
learn..evolve

WIPRO NGA Program – C++ LSP

Capstone Project Presentation – 12 August 2024

Project Title Here - Online Quiz System Using Client Server Architecture

Presented by – Harshitha N

Project overview

The project implements a client-server-based online quiz system using socket programming in C++. This system is designed to facilitate real-time, interactive quizzes where multiple clients can connect to a central server, answer questions, and receive instant feedback on their performance. The system facilitates real-time communication between the client and server, ensuring an interactive and responsive quiz experience. Designed with scalability in mind, it supports multiple clients concurrently.

Introduction

In today's digital age, online quizzes have become a popular way to engage users and assess their knowledge on various subjects. With the rise of online learning platforms and educational websites, there is a growing need for efficient and scalable online quiz systems. This project aims to develop a simple online quiz system using socket programming in C++, which enables real-time communication between the client and server.

Project Scope

The objective is to design and implement a simple client-server based quiz system that allows a client to connect to a server and take a quiz.

1. To design and develop a client-server architecture for the online quiz system that ensures scalability, reliability, and security.
2. To provide a user-friendly interface for users to access quizzes electronically.
3. To ensure the confidentiality and privacy of user data.
4. To provide a convenient and accessible way for users to access quizzes.

System Requirements

Functional Requirements

1. Socket Creation
2. Server Connection
3. User Interaction
4. Data Transmission

Non-Functional Requirements:

1. Performance
2. Reliability
3. Usability

Various Application Tools

1. Programming Language: C++ is used for its efficiency and control over system resources, making it suitable for network programming.

2. Libraries and Headers:- <sys/socket.h>: Provides functions for creating and managing sockets, such as socket(), bind(), listen(), and accept().- <netinet/in.h>: Contains constants and data structures for internet domain addresses, including sockaddr_in.- <arpa/inet.h>: Provides functions for IP address manipulation, like inet_pton() for address conversion.- <unistd.h>: Includes system calls for POSIX operating systems, such as read() and write().

3. Development Tools:*- GCC Compiler: For compiling the C++ code.- Ubuntu Linux: Provides a development environment with support for network programming libraries and tools.

Modules

- Server Module:

1. Manages the quiz process, including storing and sending quiz questions.
2. Handles multiple client connections simultaneously.
3. Tracks and records each client’s score during the quiz.
4. Listens for incoming client connections on a specified port.
5. Manages communication between the server and clients.

- Client module:

1. Connects to the server to participate in the quiz.
2. Receives quiz questions from the server.
3. Submits answers back to the server for evaluation.
4. Displays the final score to the user.
5. Provides a user-friendly interface for seamless interaction with the quiz.

Future Enhancement

- Persistent Score Tracking: Store scores in a database for tracking user progress and leaderboards.
- Graphical User Interface (GUI): Develop a user-friendly GUI for a more engaging experience.
- Question Randomization and Categories: Randomize questions and categorize by topic or difficulty.
- Advanced Features: Add more complex quiz functionalities, such as timed questions, question randomization, and detailed statistics.