**Project Report  
Online Quiz System using Socket Programming**

**1. Introduction:**

The Online Quiz System is a network-based application developed using socket programming. It consists of three main components: Main Server, Sub Server, and Client. The system allows multiple clients to connect to the servers simultaneously and participate in the quiz. The servers are responsible for managing the quiz questions and scores, while the client interface provides the users with the questions and accepts their responses.

**2.Objectives:**

* Develop a distributed application using socket programming.
* Implement a single-threaded server architecture to handle clients.
* Create a user-friendly client interface for taking the quiz.
* Design a robust system for delivering quiz questions and recording scores accurately.

**3. System Architecture:**

The system architecture comprises two levels of servers: Main Server and Sub Server.

* **Main Server:** This server acts as the central point of contact for all clients. It manages the overall flow of the quiz, gets questions from the sub-servers, and sends them to the client and sends back the response of the client to the sub Server.
* **Sub Server:** Each sub-server is responsible for handling a subset of questions. The main server provides sub-servers the response of clients that sub-servers judge the answer and assign the proper score.
* **Client:** The client interface provides users with the quiz questions and accepts their answers. It communicates with the main server to join the quiz and submits the answers to the sub-server.

**4. Implementation Details:**

* **Socket Programming:** Implemented using Python's socket library for establishing network connections between servers and clients.
* **GUI Implementation:** implemented using Python’s tinkers Library for giving easy to use interface to clients.
* **Score Management:** Scores are maintained at the sub-server level. Each sub-server tracks the scores of clients participating in its assigned questions.
* **Communication Protocol:** Defined a custom communication protocol between servers and clients for exchanging messages related to quiz participation, question delivery, and score submission.

**5. Functionality:**

* **Client Interface:** Provides a user-friendly interface for displaying questions, accepting user responses, and displaying scores.
* **Server Coordination:** The main server coordinates the distribution of questions among sub-servers and collects scores from them to calculate the final results.
* **Error Handling:** Implemented robust error handling mechanisms to handle network failures, client disconnects, and other potential issues gracefully.

**6. Conclusion:**

The Online Quiz System developed using socket programming provides an interactive platform for conducting quizzes over a network. Its distributed architecture ensures scalability and performance, allowing multiple users to participate simultaneously. With further enhancements and refinements, the system can offer an engaging and educational experience for users of all ages.