**Title: Build a Simple Blockchain Simulation**

**Objective:** Demonstrate your basic understanding of programming, problem-solving, and blockchain concepts by creating a simple blockchain simulation in your preferred programming language (Python, JavaScript, or any other).

#### **Task Description:**

You are required to build a **basic blockchain simulation** that mimics the core features of a blockchain. The simulation should include the following components:

1. **Block Structure**:  
   * Each block should contain the following:
     + Block number or index.
     + Timestamp of block creation.
     + A list of transactions (can be dummy data).
     + Hash of the previous block.
     + Current block hash.
2. **Hashing**:  
   * Use any hashing algorithm (e.g., SHA-256) to generate the block’s hash based on its data and the previous block’s hash.
3. **Blockchain Class**:  
   * Create a class or module to manage the chain of blocks.
   * The class should include:
     + A method to add new blocks.
     + A method to validate the chain’s integrity (ensure hashes link correctly).
4. **Simple Proof-of-Work** (Optional for bonus points):  
   * Implement a basic proof-of-work mechanism to make block creation computationally intensive (e.g., a condition on the hash value).
5. **Output:**
   * Print the blockchain showing the details of each block in the chain.
   * Include a mechanism to tamper with the data and demonstrate how the chain detects tampering.

#### **Evaluation Criteria:**

1. **Functionality**:  
   * Does the program correctly simulate a blockchain with linked blocks?
   * Is the validation method able to detect tampered data?
2. **Code Quality**:  
   * Is the code well-structured, modular, and readable?
   * Are comments added where necessary to explain logic?
3. **Optional Bonus**:  
   * Implementation of proof-of-work.
   * Ability to dynamically add transactions to blocks before mining.

### **Submission Guidelines**

1. **GitHub Repository**: Share the project repository link with:
   * A clear README.md file for setup and execution instructions.
   * Properly commented and organized code.
2. Submit your project via [this form](https://docs.google.com/forms/d/e/1FAIpQLScJIaxeMSCvXs6ShInKj60H1ZWumBbgJcdBKaCzmnVV4IuZgQ/viewform?usp=dialog).

### **Evaluation Criteria**

1. **Functionality**: The application meets the described features and flows.
2. **Code Quality**: Clean, modular, and well-documented code.
3. **Dockerization**: Proper Docker setup for frontend, backend, and database.

### **Timeline**

You have **72 hours** to complete the task. After the deadline, your submission will not be accepted.

### **Contact Information**

For queries, email **hr@quadbtech.com** with the subject **Full Stack Developer Task Query**.

Good luck!