The Institute Management System is a software application designed to streamline the core administrative tasks of an educational institution. The system allows administrators to manage key operations, such as adding, editing, and deleting staff members and students, while securely storing all the data in a database. This project implements a complete solution for handling student and staff information, including features for authentication and authorization.

The system's architecture emphasizes simplicity and ease of use, providing functionalities for adding institutes, managing staff and student records, and retrieving the stored data from the database.

Key Features:

- Authentication & Authorization: Secure login system to control access.
- Staff & Student Management: Add, edit, and delete staff and student records.
- Database Integration: All data is securely stored in a SQL Server database.
- Data Display: List all staff and students.
- Edit & Delete Operations: Modify or remove existing records.

Step-by-Step Guide to Implement:

1. System Architecture Design

- Design a simple client-server model.
- The client side provides an interface for interacting with the database, while the server handles the database queries.

2. Database Design

Database Creation:

- Create a SQL Server database named InstituteDB.
- Define the necessary tables: Users, Staff, and Students.
- Include columns for storing staff and student details such as name, position, institute_name, and location.

3. Backend Implementation

- Use Python with pyodbc for database interactions.
- Classes:
  - User: Handles user registration, login, and authentication.
  - Staff and Student: Manages staff and student details.

4. Features and Functionalities

Authentication:

- Register users (admin and staff).
- Log in by verifying usernames and hashed passwords.

Staff & Student Management:

- Add new records
- Edit existing records
- Display Data: List all staff and students by querying the database and using a Data Frame for easier viewing.
- Delete Records: Remove staff or students by their unique ID.

5. Front-End Implementation

- Interface: Use a basic command-line interface for input and output.
- Add options for adding, editing, displaying, and deleting staff and students, and for user authentication.

6. Final Testing

Test all functionalities:

- User registration and login.
- Adding, editing, and deleting staff and student records.
- Displaying lists of all staff and students.

Ensure data consistency in the database and that only authorized users can perform certain actions (e.g., only admins can edit staff).

7. Deployment

- Upload the complete project to GitHub.
- Include a README file explaining how to set up the environment, connect to the SQL Server, and run the system.