Project 1: Online Retail Store Database System

Objective: Design and implement a database system for an online retail store. The system should manage inventory, sales, and customer information efficiently to support both front-end user interactions and back-end management activities.

Description:

- **Inventory Management:** Design tables to store information about products, including product ID, name, description, price, and stock levels. Implement features to add, update, or delete product information.
- **Sales Transaction Processing:** Create a sales transaction system that records each purchase, including transaction ID, customer ID, product IDs, quantities, prices, and date of transaction. Ensure the system supports viewing and managing past transactions.
- **Customer Management:** Develop a database schema to store customer information, such as customer ID, name, email, and address. Include functionalities for adding new customers and updating existing customer details.
- **Security and Authentication:** Design a simple authentication mechanism for different types of users (e.g., admin, staff, customer) to access and manage their relevant parts of the database securely.

Technical Requirements:

- Use a relational database management system (RDBMS) such as MySQL, PostgreSQL, or SQLite.
- Develop a simple front-end interface (optional) for interacting with the database, using a programming language or framework of your choice (e.g., Python with Flask/Django, Java with Spring, PHP).
- Ensure normalization of database tables to at least the third normal form (3NF) to reduce redundancy and improve data integrity.

Deliverables:

- Database schema and design documents, including ER diagrams and table definitions.
- SQL scripts for database creation, including tables, constraints, and indexes.
- Source code for any developed front-end interface and database interaction logic.

- A final report detailing the project design, implementation, challenges faced, and solutions implemented.

Project 2: University Course Registration System

Objective: Develop a comprehensive database system for managing course registration at a university. This system should handle courses, students, instructors, and enrollments efficiently.

Description:

- **Course Management:** Design a database schema to store information about courses offered, including course code, title, description, credits, instructor ID, and schedule. Implement functionality to add, update, and delete course listings.
- **Student Information System:** Create tables to manage student information, such as student ID, name, major, year, and contact details. Include features for enrolling students in courses, tracking their progress, and managing academic records.
- **Instructor Management:** Develop a system to manage instructor details, including instructor ID, name, department, and courses taught. Allow for adding new instructors and updating existing information.
- **Enrollment and Grades:** Design a system for enrolling students in courses, recording grades, and managing course prerequisites. Ensure integrity constraints are in place to prevent enrollment in courses with unmet prerequisites.
- **Search and Query Capabilities:** Implement search functionalities for students to find courses, view schedules, and check enrollment status. Instructors should be able to view enrolled students and enter grades.

Technical Requirements:

- Utilize a RDBMS such as MySQL, PostgreSQL, or Microsoft SQL Server.
- Create a web-based interface for users to interact with the system, using technologies like HTML, CSS, JavaScript, and a server-side scripting language.
- Apply database normalization principles to ensure efficient and reliable data storage.

Deliverables:

- ER diagrams and database schema documentation outlining the design.
- SQL scripts for creating the database environment, including all tables, views, stored procedures, and necessary constraints.
- Source code for the web interface and database interaction (if implemented).
- A comprehensive final report describing the system architecture, functionalities, user guide, and reflections on the project execution.