Problem 1: Library Management System

Scenario:

You are tasked with designing a library management system for a university. The system should allow librarians to manage books, patrons, and borrowing activities efficiently.

Requirements:

1. Book Management:
    - Books have attributes like Title, Author, ISBN, and Quantity.
    - Patrons can borrow multiple books.
    - A book can be borrowed by multiple patrons.
2. Patron Management:
    - Patrons have attributes like Name, ID, and Contact Information.
    - Each patron can borrow a limited number of books at a time.
    - Patrons can have fines if books are returned late.
3. Borrowing System:
    - Books can be borrowed for a specific duration.
    - Calculate fines for late returns based on a predefined fine rate.

Tasks:

1. Class Diagram:

   Create a class diagram showing the relationship between Book, Patron, and Borrowing.

   Include relevant attributes and relationships between classes.

2. Database Diagram:

   Design a database schema representing the entities Book, Patron, and Borrowing.

   Define the relationships between tables and attributes.

Problem 2: Online Quiz System

Scenario:

You're tasked with developing a quiz application that generates randomized multiple-choice questions for users.

Requirements:

1. **Question Generation:**
   - **Create a pool of 50 questions on various topics (e.g., Science, History, Math).**
   - **Each question should have a unique ID, the question itself, multiple choices (4 options), and the correct answer.**
2. **Quiz Interface:**
   - **Design a web-based interface using HTML, CSS, and JavaScript.**
   - **The interface should display a random question from the pool with multiple-choice options.**
   - **Users can select an option and submit their answer.**
3. **Scoring System:**
   - **Implement a scoring system to calculate the user's score based on correct answers.**
   - **Show the user's score after completing the quiz.**
4. **Logic and Algorithm:**
   - **Develop a JavaScript algorithm to randomly select questions without repeating until all questions have been answered.**

Tasks:

1. **Logical Design:**
   - **Devise a plan for organizing the questions, their structure, and the user's progress.**
   - **Write a logical outline or pseudocode detailing how the system will generate and present questions while maintaining scores and ensuring no repetition.**
2. **Algorithm Implementation (Partial Code):**
   - **Write JavaScript functions to generate random questions, display them on the interface, receive user input, and calculate the score.**
   - **Develop the logic for question randomization without repeating until all questions are answered.**
3. **Class and Database Representation (Explanation Only):**
   - **Explain how you would represent the structure of questions, choices, and user progress in a class diagram and database schema.**
   - **Describe the relationships between entities, emphasizing the flow of data during the quiz.**