KARPAGAM COLLEGE OF ENGINEERING, COIMBATORE-641032

B.E. / B.Tech. CS, CY, CD, CT, EC, EE, ET / IT SEMESTER: IV

21PE04/21YE04 /21DE04/21SE04/21FE04 – ADVANCED JAVA PROGRAMMING

ASSIGNMENT: I

MAX. MARKS: 25 Marks DATE:

As part of the evaluation of the student's understanding and application of database systems and Java programming, the students are required to complete a project assignment that demonstrates his/her skills in database design, CRUD (Create, Read, Update, Delete) operations, and console interface development.

Choose any one among the given scenarios for submitting the assignment. The assignment will be evaluated based on the following criteria:

Criterion	**Marks**	**COs**	**Bloom's Level**
Database Design	7	CO2	K3
CRUD Operations	12	CO3	K3
Console Interface	6	CO2	K3
Total Marks	**25**		

1. University Enrollment System

Description: Develop a system to manage student enrollments, courses, and faculty assignments efficiently. This system should allow administrators to add or remove courses, enroll students in courses, and assign faculty to teach courses.

Requirements:

- **Database Design:** Create tables for Students, Courses, Faculty, and Enrollments. Ensure relationships are properly defined.
- **CRUD Operations:** Implement functionalities to add, update, and delete records in each of the entities. Include the capability to enroll and unenroll students from courses.
- **Console Interface:** Design a user-friendly console interface for interacting with the system, including menu options for each operation.

2. Hospital Management System

Description: An application to manage patient records, appointments, and medical staff information. The system should streamline the process of scheduling appointments and maintaining up-to-date patient and staff records.

Requirements:

- **Database Design:** Develop tables for Patients, Appointments, Staff, and Departments. Include necessary foreign keys and constraints.
- **CRUD Operations:** Facilitate adding, updating, and deleting information for patients and staff. Manage appointments with the ability to schedule, reschedule, or cancel.
- **Console Interface:** Provide a straightforward console interface for various user roles (e.g., receptionist, medical staff) to access relevant functionalities.

3. Employee Payroll System

Description: A system for managing employee details, salaries, and payroll transactions. This includes handling personal information, calculating salaries based on hours worked, and generating payroll reports.

Requirements:

- **Database Design:** Set up tables for Employees, Salaries, and Payroll_Transactions. Design the schema to support different employee types and payment schedules.

- **CRUD Operations:** Ensure capabilities to manage employee records, update salary details, and process payroll transactions are implemented.

- **Console Interface:** Create a console interface that allows HR staff to easily manage employee information and payroll data.

4. Real Estate Property Management System

Description: A system to manage property listings, client information, and sales transactions. Real estate agents should be able to list properties, manage client inquiries, and process sales or rentals.

Requirements:

- **Database Design:** Craft tables for Properties, Clients, Agents, and Transactions. Reflect the relationships between agents, clients, and properties accurately.
- **CRUD Operations:** Provide functionalities to add new listings, update property details, manage client information, and record transactions.

- **Console Interface:** Develop a console-based UI that enables agents to navigate through property listings, client requests, and transactions efficiently.

5. Retail Inventory Management System

Description: An application for tracking product inventory, sales, and supplier details. This system should help in managing stock levels, recording sales transactions, and maintaining supplier information.

Requirements:

- **Database Design:** Construct tables for Products, Sales, Suppliers, and Stock_Levels. Ensure proper tracking of inventory movements and supplier details.
- **CRUD Operations:** Implement operations to manage products, process sales, update stock levels, and handle supplier data.

- **Console Interface:** Offer a user-friendly console interface for store managers and staff to manage inventory and sales data effectively.

6. Hotel Booking System

Description: An application for managing room bookings, guest details, and reservation schedules. This includes handling room availability, guest reservations, and check-in/check-out processes.

Requirements:

- **Database Design:** Design tables for Rooms, Guests, Bookings, and Payments. Model the data to accurately represent room statuses and guest bookings.
- **CRUD Operations:** Enable adding, updating, and deleting room and guest information. Manage booking schedules and process payments.

- **Console Interface:** Construct a console UI that facilitates easy access to booking information, room management, and guest services for hotel staff.

7. Event Management System

Description: An application for organizing event schedules, registrations, and participant information. This includes planning events, managing attendee registrations, and tracking event logistics.

Requirements:

- **Database Design:** Create tables for Events, Participants, Registrations, and Venues. Define relationships to manage event details, venues, and participant information.
- **CRUD Operations:** Allow for the addition, update, and deletion of event details, participant registration, and venue information.

- **Console Interface:** Build a console interface that enables event organizers to manage events, registrations, and venues seamlessly.

Instructions for Submission:

1. **Prepare Your Project for Submission:**
    - Ensure your Java application source code is clean, well-commented, and adheres to the assignment requirements.
    - Organize your project into a logical structure, e.g., separate packages for database connections, entity classes, and user interface classes.
    - Include the SQL script files for creating your database schema and any initial data you've used for testing.
    - Prepare a comprehensive README file that includes:
        - An overview of the project.
        - Detailed setup instructions (including database setup).
        - A user guide for operating the console application.
    - Ensure all personal information and sensitive credentials are removed or obscured in your submission.
    - Submit using the following template: Submission Template

2. **Create a GitHub Repository:**
    - Sign in to your GitHub account (create one if you do not have).
    - Click on the "+" icon in the upper right corner and select "New repository."
    - Name the repository using your roll number (e.g., 2024CS101).
    - Choose "Public" or "Private" based on your preference or instruction given by your course coordinator.
    - Initialize the repository with a README file.
    - Click "Create repository."

3. **Push Your Project to GitHub:**
    - Open a terminal or command prompt on your computer.
    - Navigate to the root directory of your project.
    - Initialize the local directory as a Git repository with git init.
    - Add the GitHub repository as a remote with git remote add origin https://github.com/your_username/your_rollno.git replacing your_username and your_rollno with your actual GitHub username and the roll number repository name.
    - Add your project files to the repository using git add .
    - Commit the changes with git commit -m "Initial project commit".
    - Push the changes to GitHub with git push -u origin master (or main, depending on your Git version).
4. **Submit the details provided in the google form: submission link**

Signature & Name of the Course Coordinator
[Dr. Arul Antran Vijay S., ASP/CSE]

Signature of HoD