Core Requirement (AI Flavor & Submission) (Submit Any 1 assignment of your choice)

Candidates must document their use of AI-based development tools (e.g., Gemini, GitHub Copilot, ChatGPT, Cursor AI) throughout the coding process.

Component	**Detail**
Tool Usage	Use AI tools for code generation (e.g., helper functions, boilerplate), test case creation, or debugging assistance.
Final Submission	Include a dedicated section in your **README.md** file detailing:
	1. AI-Assisted Tasks: A list of all tasks where an AI tool was used (e.g., *Generated Python helper class for input validation*, *Wrote Jest test cases for the login API endpoint*).
	2. Effectiveness Score: A simple score (e.g., 1 to 5) reflecting the overall efficiency gained from using AI tools, with a brief justification.

Assignment 1: Secure User Profile & Access Control System

The Challenge: Building an Identity Management Microservice

Develop a secure system to manage user identity and profile data, incorporating encryption for sensitive fields.

Area	Component	Requirements & Engineering Focus
Backend	**User Profile Service**	1. **User Authentication:** Implement a secure Login/Registration API using **JWT (JSON Web Tokens)** for stateless authentication. 2. **Data Security:** Design the database model (SQL or NoSQL) for user profiles. Ensure the **Aadhaar/ID Number** field is stored **encrypted** (e.g., AES-256) at rest. 3. **API Design:** Create an authenticated API endpoint to fetch the user profile data, which requires **decrypting** the Aadhaar/ID number before sending it to the client.
Frontend	**Profile Management Dashboard**	1. **Login/Registration Pages:** Implement the UI to interact with the authentication APIs. 2. **Profile View:** Create a dashboard that securely fetches and displays the user's details, including the decrypted Aadhaar/ID Number. 3. **Error Handling:** Implement robust client-side and server-side error handling for failed logins or data fetches.
AI Flavor	**Token Validation & Encryption Tests**	Use an AI tool to generate the **utility function** for token validation on the backend and to create **comprehensive unit tests** for the encryption/decryption logic.

Assignment 2: Real-time Transaction & Audit Log System
The Challenge: Simple Fund Transfer and Audit Trail

Develop a system that simulates a simple peer-to-peer fund transfer and maintains a mandatory, immutable audit log.

Area	Component	Requirements & Engineering Focus
Backend	**Transaction & Audit Service**	1. **Transaction API:** Implement a single endpoint for `POST /transfer`. The logic must deduct the amount from one user and credit it to another. **Crucially, this must be wrapped in a database transaction** to ensure atomicity (both debit and credit succeed, or both fail). 2. **Audit Log:** After a successful transfer, asynchronously (or in a separate, dedicated table) log the full transaction details (sender ID, receiver ID, amount, timestamp, success status). 3. **Read API:** Create an authenticated endpoint to fetch the transaction history (the Audit Log) for a specific user.
Frontend	**Transfer & History Interface**	1. **Transfer Form:** A simple UI to input the receiver's ID and the amount. 2. **Real-time Update:** After a successful transfer, the frontend must immediately update to show the new balance and the latest entry in the history. 3. **History View:** Display the user's transaction history in a clear, sortable table format.
AI Flavor	**DB Transaction Boilerplate & Frontend Component**	Use an AI tool to generate the **database transaction boilerplate** for the `POST /transfer` endpoint (e.g., using Spring's `@Transactional` or Python's ORM session management) and to create the initial **sortable table component** for the transaction history on the frontend.

Technology Stack (Recommended but Flexible):

- **Backend:** Java/Spring Boot, Python/Django/Flask, Go, or Node.js (with Express/NestJS).
- **Database:** PostgreSQL, MySQL, MongoDB, or SQLite (Must be persisted).
- **Frontend:** React, Angular, or Vue.js.

Submission Guidelines

1. **Code Repository:** Submit the final work as a link to a **public GitHub repository**.
2. Also provide a video link/screen recording for end solutions which are completed.
3. **README.md File (Crucial):** Your repository **must** include a comprehensive README.md file with the following sections:
    - **Project Overview:** A brief description of the chosen assignment and your implementation approach.
    - **Setup/Run Instructions:** Clear, step-by-step instructions on how to set up the project (including dependencies) and run both the backend and frontend components.
    - **API Documentation:** A brief description or collection (e.g., a Postman collection) of your core API endpoints.
    - **Database Schema:** A visual representation or text description of your database tables/models.
    - **AI Tool Usage Log (MANDATORY)**

AI Flavor: Tool Usage & Documentation (Mandatory)

In modern engineering, efficiency is key. We require you to utilize an AI-based development tool (e.g., Gemini, GitHub Copilot, ChatGPT, Cursor AI) during your coding process.

Section	**Detail**
AI-Assisted Tasks	List all specific tasks where an AI tool was used (e.g., *Generated Python helper class for input validation*, *Wrote Jest test cases for the login API endpoint*, *Refactored error handling block*).
Effectiveness Score	Provide a score (1 to 5, where 5 is highly effective) for the overall efficiency gained from using AI tools, with a brief, honest justification (e.g., "*Score 4: Saved 2 hours on boilerplate code, but debugging AI-generated SQL took time*").

Evaluation Criteria

Criteria	Description	Weight
Backend Implementation	API design (RESTful/clean), security (JWT/encryption), correct use of database transactions (Option B), and robust error handling.	High
Frontend Implementation	Clean, responsive UI/UX, correct state management, and effective communication with the backend APIs.	High
Code Quality	Clean code, proper structure, naming conventions, use of design patterns, and adequate commenting.	High
Documentation & Submission	Clear README.md, easy-to-run setup, and complete adherence to the **AI Tool Usage Log**.	Medium
AI Tool Utilization	Strategic and thoughtful use of AI tools to enhance productivity, as evidenced by the log.	Medium