

PROBLEM STATEMENT 1:

Library Management System (FastAPI)

Project Name

library

Scenario Description

A public library wants to build a **Library Management System** using REST APIs to manage:

1. Library users
2. Book inventory

The application must be developed using **FastAPI**, must store all data in memory using **HashMaps (Python dictionaries)**, and must not use **Pydantic models** or any database.

The system must support **full CRUD operations** and must validate data using **Path parameters**, **Query parameters**, and **Request Body**.

Mandatory Project Structure

library/

|

├── main.py # Application entry point (mandatory)

├── users.py # User APIs (router only)

└── inventory.py # Inventory APIs (router only)

main.py Requirements (Mandatory)

- FastAPI app instance must be created **only in main.py**
- users and inventory routers must be included using `include_router()`
- All APIs must be accessed **only via main.py**
- Application must be started using:
 - `uvicorn main:app --reload`

USERS MODULE

Base Route

/users

CRUD Functional Requirements

1. **Create User (POST)**
 - `user_id` → Path (int, > 0)
 - `name` → Body (min 3 characters)
 - `email` → Body (valid email format)

- age → Body (≥ 12)
- membership_type → Query (regular, premium)
- 2. **Read User (GET)**
 - Fetch user details using user_id
- 3. **Update User (PUT)**
 - Update name, email, or age using user_id
- 4. **Delete User (DELETE)**
 - Delete user using user_id

Validation Rules

- Duplicate user_id must not be allowed
- Invalid email format must be rejected
- Membership type must be validated using query parameters
- Proper error messages and HTTP status codes must be returned

INVENTORY MODULE

Base Route

/inventory

CRUD Functional Requirements

1. **Create Book (POST)**
 - book_id → Path (int, > 0)
 - title → Body (min 3 characters)
 - author → Body (non-empty)
 - quantity → Body (≥ 1)
 - category → Query (fiction, non-fiction, academic)
2. **Read Book (GET)**
3. **Update Book Quantity (PUT)**
4. **Delete Book (DELETE)**

Validation Rules

- Quantity must not go below zero
- Category must be validated via query parameter
- Proper error responses required