N-Tier Software Architecture Diagram of Library Management System

Presentation layer (Frontend)

- Components:
- Frontend components (e.g., login, register, book search, borrowing books).
- Templates:
- HTML: add_book.html, my_books.html, rate_book.html, etc.
- CSS: styles.css, my_books.css, available_books.css.
- Static assets: Videos, icons, images (referenced in templates).
- Endpoints to Backend:
- /login, /signup, /borrow-book, /return-book, /view-books.

Application Layer

- Purpose: Processes requests and contains business logic.
- Components:
- Python Files:
  - main.py: Entry point for the Flask app.
  - auth.py: Manages authentication routes and logic.
  - views.py: Defines routes for various user actions.
  - Key Routes:
    - /login (from auth.py): User login.
    - /add-book (from views.py): Adding new books.
    - /borrow-book (from views.py): Borrowing books.
  - Business Logic:
    - User authentication: Login, register, password reset.
    - Book management: Adding, editing, and recommending books.
    - Borrowing logic: Borrow, return, overdue fines.
- Services (Integrated into the logic of views.py and auth.py):
  - UserService: Logic for user authentication and session handling.
  - BookService: Logic for managing book data.
  - BorrowService: Handles borrowing and return operations.

Persistence Layer

- Purpose: Manages database interactions (ORM logic).
- Components:
- Python Files:
  - models.py: Defines database schema using SQLAlchemy.
    - User2: User data.
    - Book: Book data.
    - BorrowedBook: Borrowing records.
    - Note: Notes attached to users.
    - UserMessage: User-specific messages.
- Data Access Logic:
  - SQLAlchemy relationships and CRUD operations (handled via models.py).

Database Layer

- Purpose: Stores persistent data for the application.
- Components:
- Database Tables:
  - User2 Table: Stores user details (id, email, password, etc.).
  - Book Table: Stores book metadata (id, name, author, genre, etc.).
  - BorrowedBook Table: Tracks borrowing records (id, user_id, book_id, due_date, etc.).
  - Note Table: Stores user-specific notes (e.g., annotations).
  - UserMessage Table: Stores user messages.