- *Introduction to the Library Management System*

  Content:

- **Overview**:
  - The Library Management System aims to streamline library operations by automating user and book management.
  - Provides a user-friendly interface for managing transactions like issuing and returning books.
- **Objective**:
  - Simplify library workflows.
  - Enhance user experience for both library staff and visitors.
- **Technologies Used**:
  - Backend: Java (Maven-based).
  - Frontend: HTML, CSS, Bootstrap, JavaScript.
  - Database: MySQL for storing and managing data.

- **Features**
- **Title**: *Key Features of the System*
  Content:

1. **User Management**:
    1. User registration with validation for required fields.
    2. Login functionality with secure password handling.
    3. Profile management for updating user details.
2. **Library Management**:
    1. CRUD operations for books (Create, Read, Update, Delete).
    2. Display available books with search and filter options.
3. **Transaction Management**:
    1. Issue and return books with automatic due date calculation.
    2. Track transaction history for all users.
4. **Reports and Analytics**:
    1. Generate usage reports for admins.
    2. Summary of issued and returned books.
5. **Validation and Security**:
    1. Client-side and server-side validation for all forms.
    2. Role-based access control for admin and user accounts.

- **Title**: *Testing and Validation Process*
  Content:
- **Unit Testing**:
  - Tested service and DAO layers for core logic (e.g., adding books, updating profiles).
- **Integration Testing**:
  - Ensured that controllers interact correctly with services and DAOs.
- **Validation Testing**:
  - Tested email, password strength, and required fields in forms.
  - Checked proper error messages for invalid inputs.
- **Example Test Cases**:
  - Successfully registering a new user with valid data.
  - Preventing duplicate book entries.
  - Handling invalid login credentials gracefully.
- **Tools Used**:
  - JUnit for automated testing of Java components.

- **Challenges**
- **Title**: *Development Challenges and Solutions*
  Content:

1. **Database Synchronization**:
    1. Issue: Maintaining consistency during simultaneous transactions.
    2. Solution: Used Spring transaction management to handle concurrency.
2. **Responsive Design**:
    1. Issue: Ensuring the UI works smoothly across devices.
    2. Solution: Leveraged Bootstrap for responsiveness.
3. **Error Handling**:
    1. Issue: Showing meaningful error messages for invalid inputs.
    2. Solution: Implemented live feedback with JavaScript for better user experience.
4. **Time Constraints**:
    1. Issue: Balancing development and testing phases.
    2. Solution: Followed agile principles with small, iterative updates.

- **Demo**
- **Title**: *Application Walkthrough*
  Content:
- **Screenshots**:
  - Login Page: Shows the user authentication interface.
  - Registration Form: Captures new user details.
  - Dashboard: Displays book inventory and user actions.
  - Issue/Return Form: Demonstrates transaction workflows.
- **Video Demo**:
  - Highlights:
    - User login and registration.
    - Adding and searching for books.
    - Issuing a book and returning it with a due date check.
  - Tools: Record using screen capture software (e.g., OBS, Camtasia).
- **Key Takeaways**:
  - Simple navigation and intuitive UI.
  - All features integrated seamlessly for real-world usability.