Library Management System

INTRODUCTION:-

→ A library management system (LMS) is a software application designed to manage and automate the various operations of a library

→ It helps librarians, students & staff efficiently organize, track and maintain books, digital resources & user records.

Sunday 04

PURPOSE:-

→ The primary purpose of an LMS is to improve the efficiency of library operations, reduce manual workload and enhance the user experience. It allows users to search for books, borrow & return them and manage their accounts online.

KEY FEATURES:-

→ BOOK MANAGEMENT → Cataloging, adding & updating book records

→ USER MANAGEMENT → Tracking student, faculty and staff memberships.

→ BORROWING & RETURNING → Automate book issuance and returns

SEARCH & CATALOGING :- Allowing users to search for books easily

DIGITAL RESOURCE MANAGEMENT- Managing e-books, journals and research papers

BENEFITS:-

→ TIME SAVING:- Automates repetetive tasks

→ ACCURACY & EFFICIENCY :- Reduce human errors in record keeping

→ SECURITY :- Prevents unauthorized access & book losses

→ SCALABILITY :- Can be expanded to accomodate more users and books

1) Setting up a Spring Boot project in VS code

→ Download & Install Java JDK

→ Download JDK 17 or later from oracle or open JDK

→ Verify Installation:
java version

2) Install maven

→ Install from Apache maven

→ Verify Installation

mvn - version

3) Install visual studio code & extensions

→ Install the VS code extensions

* Create a Spring Boot Project :-
→ Using spring Initializer
→ Open spring initializer in a browser
→ Select
→ Dependencies
→ Click & Generate, download the ZIP file
→ Extract the project and open it in VS code

14/1/25

→ Issue solving of VS ode
→ Environment setting of MySQL
→ Basic Probm about springboot

→ Demonstration on springboot project
→ Creating folder on flavourclub,
→ Creating home.html on template folder.

15/1/25

Landing page → serves as the first point of interaction for users.

Navigation bar: added a responsive navigation bar with links to key sections such as HTML, menu, about us

16/1/25

→ Applied consistent color palette & reflecting a food centric theme

→ Responsive design

17/1/25

Added concise & engaging content section highlighting the platforms

Used HTML & CSS and added javascript for interactive elements such as smooth scrolling and hover effect

20/1/25

→ Designing of PDP
Worked on designing the PDP for flavour hub web app.

→ Header & navigation

Suruthi
20.01.25

21/1/25

Designing of PDP page

Added key details including product name, price, description rating & review.

Added an register with issue name, book name, issue date, etc

22/1/25

Admin Dashboard design

→ Worked on designing the admin dashboard for flavours hub web application

→ Applied a fixed sidebar menu with link to homepage, students

23/1/25

Worked on dashboard overview section

→ Worked on issuing management section

24/1/25

Worked on category design page
Applied product & customer page

25/1/25

Worked on overall design

Added ~~items~~ & icons & rever effect

26/1/25

Git & Git Hub

→ Installation & Download of Git & Git hub

→ How to push code to github

Gitpull & Github push commands:
to get & github

27|1|25

Creation of Controller Page

→ Worked on developing the controller layer for the application

→ The controller handles user reques processes them and return appropriate response

→ Created controller classes.

Suresh
28.01.

28/1/25

Controller Class
Used HTTP methods
GET (Retrive data)
POST (Add new data)
PUT (Upate excusting data)
DELETE (Remove data)

29/1/25

Controller Class
Implemented input validation
using spring boot's valid
annotation to prevent incorrect
data submission
Spring MVC is used

31/1/25

Entity class

→ Developed separate entity classes like category entity, produce entity, delivery, order

1/2/25

Created audibable entity class

The getter, setter methods used

3/2/25

Entity class

→ Utilized JPA annotations for table mapping

→ Entity → Specifies the class as a JPA entity

Sumukh
4.02.25

4/2/25

Added validation annotation to ensure data integrity

5/2/25

→ Creating Repositories

→ developed separate repository for key entities

6/2/25

Used spring data JPA to extend the JiaRepository interface providing built in methods.

7/2/25

Added custom query method to fetch specific data efficiently
→ find By category
→ find by status

8/2/25

→ Worked on enum class
→ created delevery status, order, roll no, etc

9/2/25

→ Worked on Product controller class the fetch the page

10/2/25

→ Created admin booked page which include all information
→ Created student page

11/2/25

→ Worked on database table
→ Insert value

12/2/25

→ Worked on admin dashboard

→ overall space adjustment & other design

13/2/25

→ Learn about entity relationship

→ Collection interface and about list like book list, issuing list

14|2|25

→ Learn on JDBC connection which include http request dispatcher / front controller handler wrapper, controller view revolver

15|2|25

Learn & implement fragment on admin dashboard.

→ Created a fragment file of html of css which can include all common codes which can be reuse in other html file

16/2/25

Merge the code on github
- the conflict occure, and resolve the conflict

17/2/25

Learn about UUID universally Unique identifier which are 128 bit used to uniquely identify objects in computer system

→ Implement it on entity class as generated value strategy UUID

17/2/25

18/2/25

Worked on datatable of students book name, along with its backend using post mapping & get mapping

19/2/25

→ Work on category page and product page.

→ Using thymleaf, springboot dependency, render the data

20/2/25

Completed admin category page and product page design

Worked on its data table

21/2/25

Implemented CRUD operations Create, Read, Update, Delete, to manage categories and products allowing seamless interaction with MySQL Database

22/2/25

Utilized Thymleaf template engine for rendering dynamic HTML content a displaying real time data from the backend

23/2/25

① Create new ~~row~~ ticket and new branch through github

Merge the existing code to the master branch

24/2/25

Develop dynamic data tables on the category and product page of the web application using spring boot and thymleaf

Sweta
25-02-25

25/2/25

Integrated custom search & filter option, enabling user to easily locate specific categories and product

26/2/25

Ensured data consistency & integrity using Hibernate's cascading operation and validation annotation
Worked on CSS of admin dashboard.

27/2/25

Created widget table
Learn about spring boot security

Implement it using csrf token

28/2/25

Integrated spring boot security to protect the widget, category product.

→ Applied role based authentication and authorization

1/3/25

Used Bcrypt password encoding to securely store and verify user credentials

Added custom login and function functionality with proper session handlin

2/3/25

Ensured CSRF protection (Cross-site request forgery) by enabling CSRF tokens in forms

3/3/25

Created config folder & security config class under it where used @Bean which will convert the password to hash form

Surya
03.03.25

4/3/25

Developed the userservice class as part of Flavour Hub application using Spring Boot handling business logic related to user operation

5/3/25

Added the getuser method to fetch the entire user entity, providing detailed user information
Applied exception handling using or Else throw ( ).

6|3|25

Worked on upload image
Created a folder upload image
in the explorer
→ Put it on git - ignore
→ Upload images for the category
and product page

7|3|25

Created place order page
which will confirm the
order placed successfully

→ Link it to the place order
button in add to cart page.

8/3/25

Completed place order page design and links

Worked on overall admin dashboard design page

9/3/25

Worked on search button

→ Used JPA Repository <Book list, String> to leverage built-in-CRUD operation and simplify database interaction.

10/3/25

Implemented the find by book name containing ignore case method to perform case insensitive search for books by name.

→ Used spring Data JPA's method.

Sanat
10-3-25

11/3/25

Worked on backend of widget page and data table.

12/3/25

Created widget repository ~~entity~~ & Render data from database to front end.

13/3/25

Sunday 0

→ Created widget entity class
→ Link it to the database
→ Worked on login & signup error and fix it.

14/3/25

Created widget list front end page

→ Create CSV file of widget list which have product id and widget id

15/3/25

On Admin dashboard render the dynamic data from the database and show it on front end

16/3/25

→ Worked on spring security

→ Applied javascript in the front end for enhanced design

→ Worked on Css of widget page under admin dashboard

17/3/25

Space adjustment and operate design of widget front end and widget list page

15.03.25

→ Add model to the widget html page to show list page

18/3/25

Created a repository under repository folder to extend JPA repository

→ Worked on backend of widget and widget list page.

19/3/25

Worked on delete button

→ Created edit and delete button on category and product page under admin dashboard.

20/3/25

Created edit & delete button on library page.

→ Created dto repository under repository folder

21/3/25

worked on backend of edit and delete

Merge code to the github

Worked on entity class of security of product page

22/3/25

Worked on overall edit & delete button.

→ fix the error

→ completion of all design & backend of admin dashboard.

Done
22.03.25