

# FLM's AI-POWERED JAVA FULL STACK PROJECT DETAILS

MAJOR PROJECTS (INCLUDING BACKEND & FRONTEND) :7

## Capstone FULL STACK Project

### 1. MedSync - AI-Powered Hospital Management System

A Smart, AI-Integrated Solution for Efficient Healthcare Management

#### 1 Project Overview

**MedSync** is a **full-stack hospital management system** that streamlines hospital operations by efficiently managing doctors, patients, appointments, staff, beds, and notifications. Built using **Spring Boot Microservices, S3 buckets**, and a **modern frontend**, MedSync ensures seamless digital transformation for healthcare institutions.

This system **automates patient and doctor management**, enhances scheduling, and integrates AI-powered Diet suggestions in the discharge summary system for efficient record-keeping.



# MedSync

AI-Powered Hospital Management System



Try The AI-Powered Hospital **Management** System From  
**Healthray** For The Best Patient **Outcomes**!

## 2 Technology Stack

| Component           | Technology Used                  |
|---------------------|----------------------------------|
| Backend             | Java, Spring Boot, Microservices |
| Frontend            | Html, CSS, JS                    |
| Database            | MySQL                            |
| Cloud Storage       | AWS S3 Buckets                   |
| Authentication      | JWT & Spring Security            |
| Email Notifications | SendGrid                         |
| AI Integration      | Spring AI With Open AI           |
| PDF Generation      | Thymeleaf + Flying Saucer        |

## 3 Core Functionalities

### ◆ Doctor Management

- Admins can **register, update, and manage** doctors.

- Doctors can **log in, view their schedules, and access patient details.**
- AI-powered **diagnosis and prescription suggestions** can be integrated.
- ◆ **Patient Management**
  - Receptionists can **register new patients.**
  - Doctors can **view past medical history and update diagnoses.**
  - Patients receive **appointment confirmations via email.**
- ◆ **Appointment Management**
  - Receptionists can **schedule, update, and cancel appointments.**
  - Doctors can **view booked appointments.**
  - Patients receive **automated appointment reminders.**
- ◆ **Staff & Hospital Management**
  - Manage **hospital staff details** (receptionists, nurses, etc.).
  - Allocate and **manage hospital beds and rooms** efficiently.
- ◆ **Notifications & Email Service**
  - Patients receive **email notifications** for appointment bookings and updates.
  - Doctors receive **daily appointment schedules via email.**
- ◆ **Discharge Summary Generation & Storage**
  - **Receptionists generate discharge summaries** in PDF format.
  - The PDF is **stored securely in AWS S3.**
  - The document is **downloadable anytime via a secure link.**

## **AI-Powered Features**

### **AI-Driven Prescription & Diagnosis**

- Doctors receive **AI-based suggestions** for **Diet Recommendations.**
- Integrated with **Spring AI (OpenAI)** for **medical assistance.**



## AI-Based Discharge Summary Generation

- **Dynamic template generation** using Thymeleaf.
- **AI extracts key diagnosis & prescription details** for accurate reporting.
- **Generated PDF stored in AWS S3 and made downloadable.**

## 5

### User Roles & Access Control

| Role         | Permissions                                                                          |
|--------------|--------------------------------------------------------------------------------------|
| Admin        | Manages doctors, staff, and system settings.                                         |
| Receptionist | Registers patients, books appointments, manages beds, generates discharge summaries. |
| Doctor       | Views appointments, updates diagnosis, prescribes medication.                        |

## 6

### System Architecture



#### Microservices Design:

- **Authentication Service:** Handles user login & JWT-based security.
- **Doctor Service:** Manages doctor details & schedules.
- **Patient Service:** Stores patient data & medical history.
- **Appointment Service:** Handles bookings & updates.
- **Notification Service:** Sends automated emails & alerts.
- **Storage Service:** Handles PDF generation & S3 uploads.

## 7

### Project Highlights



Cloud-based storage with S3 for patient records



Secure role-based access with JWT authentication



AI-powered medical recommendations & documentation



Automated email notifications for appointments & reports



Scalable microservices architecture

## 2.StayEase - Accommodation Management System

### Problem Statement

Finding affordable and well-managed hostel/PG accommodations is often a challenge for students and working professionals. PG owners also face difficulties in managing room availability and rent details efficiently.

**StayEase** provides a structured backend solution to bridge this gap, allowing PG owners to manage their properties while enabling users to search for available accommodations effortlessly.

**STAYEASE - ACCOMMODATION MANAGEMENT SYSTEM**

**PROJECT FEATURES:**

- FOR PG OWNERS
- FOR GENERAL USERS

**TECH STACK ;**

- Backend: Spring Boot (REST APIs), Spring Security (Authentication), Spring JPA
- Database: MySQL
- Authentication: JWT Token-based Authentication
- Logging: Log4j

**FRONTLINES EDUTECH**  
Building Trust & Careers

### Tech Stack

- **Backend:** Spring Boot (REST APIs), Spring Security (Authentication), Spring JPA
- **Database:** MySQL
- **Authentication:** JWT Token-based Authentication
- **Logging:** Log4j

## Project Features

### For PG Owners

- **Registration:** Owners can sign up and add their PG details.
- **Property Management:** Add and update rooms and bed availability.
- **Rent Management:** Set and modify rent per room/bed.
- **Status Updates:** Mark beds as occupied or available.
- **Booking Insights:** View total bookings and manage inquiries.

### For General Users

- **Search PGs:** Look for PGs based on location, rent, and availability.
- **View Property Details:** Check PG amenities, rooms, and available beds.
- **Contact Owners:** Reach out to PG owners for booking inquiries.

## 3. Smart Bank Management System

### Problem Statement

Traditional banking systems require customers to visit the bank for many services, leading to inefficiency. This **Smart Bank Management System** provides an online banking platform where users can:

- **Manage accounts**
- **Perform transactions**
- **Apply for loans**



# SMART BANK MANAGEMENT SYSTEM

## Problem Statement

- Manage accounts
- Perform transactions
- Apply for loans

## Features :

- Customer Functionalities
- Admin Functionalities

### Tech Stack

- Backend: Spring MVC, Spring ORM(JPA)
- Frontend: JSP, HTML, CSS
- Database: MySQL
- Server: Apache Tomcat



## Tech Stack

- **Backend:** Spring MVC, Spring ORM(JPA)
- **Frontend:** JSP, HTML, CSS
- **Database:** MySQL
- **Server:** Apache Tomcat

## Features

### Customer Functionalities

- ✓ **User Registration & Login** – Secure login with OTP/email verification.
- ✓ **Account Dashboard** – View balance, transactions, and loan status.
- ✓ **Fund Transfer** – Transfer money using account number or UPI ID.
- ✓ **Mini Statement** – Get last 10 transactions in a detailed view.
- ✓ **Loan Application** – Apply for loans and track approval status.

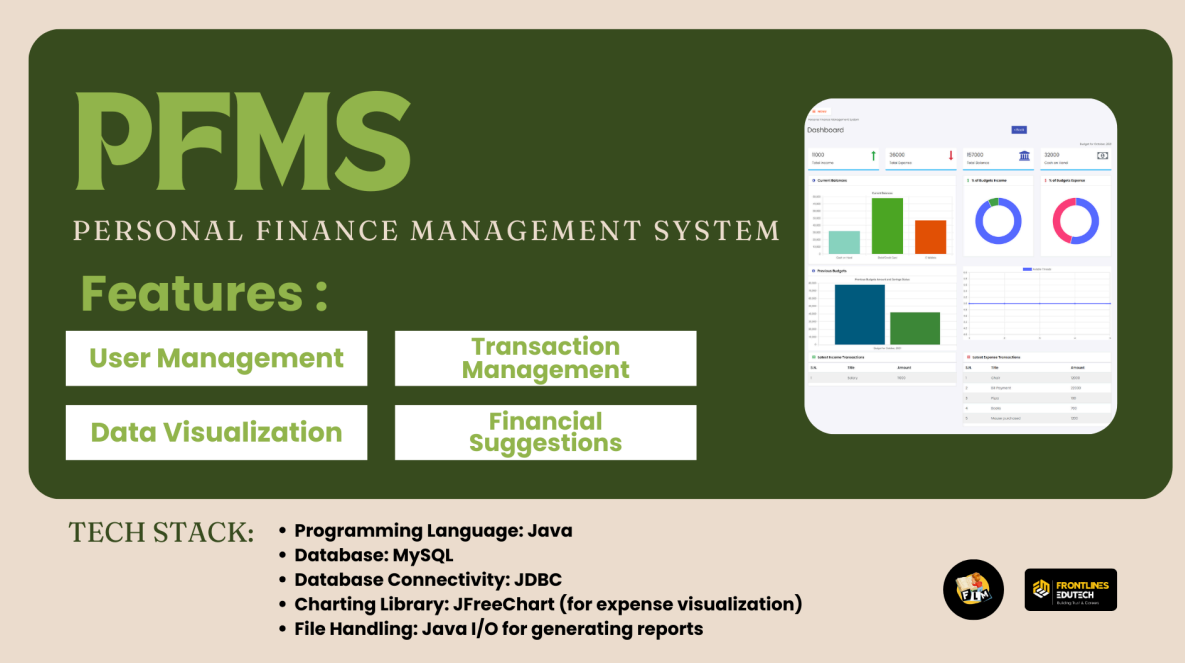
### Admin Functionalities

- ✓ **Approve Loan Requests** – Review and approve/reject loans.
- ✓ **Manage Customers** – View and manage all customer accounts.
- ✓ **Generate Reports** – Monthly/annual transaction reports.

## 4. PFMS-Personal Finance Management System

### Introduction

Managing personal finances effectively is crucial for financial stability. Many individuals struggle with tracking their income and expenses, leading to overspending and poor financial decisions. This project aims to develop a **Personal Expenses Tracker** using Java and JDBC to help users monitor their financial transactions, analyze spending patterns, and receive suggestions to optimize their expenses.



The graphic features a dark green background with the text 'PFMS PERSONAL FINANCE MANAGEMENT SYSTEM' in large, bold, light green letters. Below this, the word 'Features:' is followed by four white boxes containing the features: 'User Management', 'Transaction Management', 'Data Visualization', and 'Financial Suggestions'. To the right is a screenshot of the application's dashboard, which includes various charts and tables. At the bottom, the 'TECH STACK:' is listed with its components, and two logos are displayed on the right.

**PFMS**  
PERSONAL FINANCE MANAGEMENT SYSTEM

**Features :**

- User Management
- Transaction Management
- Data Visualization
- Financial Suggestions

**TECH STACK:**

- Programming Language: Java
- Database: MySQL
- Database Connectivity: JDBC
- Charting Library: JFreeChart (for expense visualization)
- File Handling: Java I/O for generating reports

Logos: PFMS and FRONTLINES EDUTECH

### Tech Stack

- **Programming Language:** Java
- **Database:** MySQL
- **Database Connectivity:** JDBC
- **Charting Library:** JFreeChart (for expense visualization)
- **File Handling:** Java I/O for generating reports

## Features

### User Management

- User **registration and authentication**.
- Secure login credentials stored in the database.

### Transaction Management

- Users can **add income** (e.g., salary, business earnings, investments).
- Users can **record expenses** categorized under predefined or custom categories (e.g., rent, groceries, entertainment).
- Retrieve **transaction history** for a specified period.

### Data Visualization

- Generate a **pie chart** (PNG image) representing expense distribution by category.

### Financial Suggestions

- Analyze **spending patterns**.
- Suggest **budget optimization** strategies.
- Provide alerts for **high spending** in specific categories.

## 5. Spotify Clone – Music Streaming UI

### Description:

TuneVibe is a feature-rich Spotify clone that allows users to stream music, create playlists, and discover trending tracks. The platform offers a seamless audio streaming experience with features like play/pause, skip, shuffle, and a dynamic queue. It also supports user authentication and playlist management.



# Spotify Clone – Music Streaming UI



## Tech Stack:

- **Frontend:** HTML, CSS, JavaScript
- **Media Handling:** HTML5 Audio API
- **Features:**
  - Play/Pause functionality
  - Playlist navigation
  - Basic UI similar to Spotify

## 6. ShopVibe E-commerce

### Description:

ShopVibe is a modern e-commerce platform that enables users to browse, search, and purchase products seamlessly. It offers a smooth shopping experience with secure checkout, product categorization, and user-friendly navigation for customers and admins.



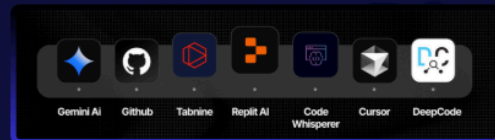
# SHOPVIBE E-COMMERCE

## Description:

ShopVibe is a modern e-commerce platform that enables users to browse, search, and purchase products seamlessly. It offers a smooth shopping experience with secure checkout, product categorization, and user-friendly navigation for customers and admins.

## Tech stack :

HTML, CSS, JavaScript, VibeCoding



## Tech Stack:

- **Frontend:** HTML, CSS, JavaScript, VibeCoding
- **Features:**
  - User registration and login
  - Browse and search products by category
  - Add to cart and place orders
  - Manage product listings (Admin)
  - Order tracking and history

## 7. Blog Application

### Description:

Blog Application is a responsive web app built with Bootstrap that allows users to create, read, update, and delete blog posts. It provides an easy-to-use interface for managing content and engaging with readers through organized blog listings.

## BLOG APPLICATION



**Tech Stack:**  
**Frontend: HTML, CSS, Bootstrap**

### **Tech Stack:**

- **Frontend:** HTML, CSS, Bootstrap
- **Features:**
  - Create, edit, and delete blog posts
  - Responsive design using Bootstrap
  - View all blogs in a clean card layout
  - Search and filter blogs by title or category
  - Comment section for reader interaction

# MINI PROJECTS (INCLUDING BACKEND & FRONTEND):7

## 1. Music Player Application

### Problem Statement

Traditional music applications rely on complex data structures and frameworks. However, for a beginner-friendly Object-Oriented Programming (OOP) approach, we need a lightweight music player that allows users to add, update, delete, and play songs, as well as manage playlists without using collections. This project demonstrates how to build a basic music management system using pure OOP concepts like Encapsulation, Inheritance, and Abstraction in Java.



### Music Player Application

**Project Features**

- Add Songs – Users can add songs with a title, artist, and duration.
- Update Songs – Modify existing song details.
- Delete Songs – Remove a song from the list.
- Play a Song – Select and play a song.
- Pause & Stop – Pause or stop a song anytime.
- Create Playlists – Users can create a playlist.
- Add Songs to Playlist – Assign songs to a specific playlist.
- Play a Playlist – Play all songs in a particular playlist.

**TECH STACK ;**

- Java – Core OOP principles
- Object-Oriented Programming – Encapsulation, Abstraction, Inheritance
- Console-based Application – Runs in the terminal



### Tech Stack

Java – Core OOP principles

Object-Oriented Programming – Encapsulation, Abstraction, Inheritance

Console-based Application – Runs in the terminal

## Project Features

- ✓ **Add Songs** – Users can add songs with a title, artist, and duration.
- ✓ **Update Songs** – Modify existing song details.
- ✓ **Delete Songs** – Remove a song from the list.
- ✓ **Play a Song** – Select and play a song.
- ✓ **Pause & Stop** – Pause or stop a song anytime.
- ✓ **Create Playlists** – Users can create a playlist.
- ✓ **Add Songs to Playlist** – Assign songs to a specific playlist.
- ✓ **Play a Playlist** – Play all songs in a particular playlist.

## 2. Online Food Ordering System

### Problem Statement

The goal of this project is to build a menu-driven console-based food ordering system where:

1. Admins can add restaurants and their menu items.
2. Users can browse restaurants, place orders, and cancel them.

**ONLINE FOOD ORDERING SYSTEM**

**Features :**

- Admin Functionalities
- User Functionalities

**Tech Stack:**

- Java (Core OOP) – Encapsulation, Inheritance, Polymorphism
- Collections – HashMap, ArrayList
- Exception Handling – To handle invalid inputs

**FRONTLINES EDUTECH**  
Building Talent & Careers

## Tech Stack

- **Java (Core OOP)** – Encapsulation, Inheritance, Polymorphism
- **Collections** – `HashMap`, `ArrayList`
- **Exception Handling** – To handle invalid inputs

## Features



### Admin Functionalities

- ✓ **Add Restaurants** – Admins can add restaurants with details like name and location.
- ✓ **Add Menu Items** – Add food items under each restaurant with a name and price.
- ✓ **View All Restaurants** – Display all available restaurants.
- ✓ **View Restaurant Menu** – See the menu of a particular restaurant.



### User Functionalities

- ✓ **Browse Restaurants** – Users can view available restaurants.
- ✓ **View Menu & Prices** – Users can check the menu of any restaurant.
- ✓ **Place an Order** – Select food items and place an order.
- ✓ **Cancel an Order** – Remove an order if needed.
- ✓ **View Order History** – Show past orders placed by the user.

## 3. Gated Community - Complaint Tracker Application

### Problem Statement

Managing complaints in apartment complexes is often inefficient, with complaints getting lost or unresolved due to a lack of proper tracking. This system allows residents to log complaints, track their status, and receive updates from the apartment administration team in a streamlined way.

# GATED COMMUNITY

Complaint Tracker Application



## Features :

- Resident Functionalities
- Admin Functionalities

## Tech Stack:

- Backend: Java Servlets, JSP
- Database: MySQL (for storing complaints and user details)
- Server: Apache Tomcat
- JDBC: For database connectivity



## Tech Stack

- **Backend:** Java Servlets, JSP
- **Database:** MySQL (for storing complaints and user details)
- **Server:** Apache Tomcat
- **JDBC:** For database connectivity

## Features

### Resident Functionalities

- ✓ **Register & Login** – Residents can sign up and log in to the system.
- ✓ **Submit Complaints** – Raise complaints for issues like water supply, electricity, maintenance, etc.
- ✓ **View Complaint Status** – Check if complaints are pending, in progress, or resolved.
- ✓ **Update/Cancel Complaints** – Modify or withdraw complaints if needed.
- ✓ **Complaint History** – View past complaints and their resolution.

## **Admin Functionalities**

- ✓ **View All Complaints** – Admin can see complaints submitted by all residents.
- ✓ **Update Complaint Status** – Change status from "Pending" → "In Progress" → "Resolved".
- ✓ **Assign Complaints** – Assign maintenance staff for resolution.
- ✓ **Generate Reports** – Track complaint resolution times and trends.

## **4. AI-Based Resume Screening System using Spring AI**

### **Project Overview**

The AI-Based Resume Screening System automates resume evaluation by analyzing resumes against job descriptions. It uses Spring Boot, Spring AI (OpenAI API integration), Thymeleaf, and MongoDB to provide an intelligent, web-based solution for HR teams.

### **Problem Statement**

Recruiters receive a high volume of resumes, making manual screening time-consuming and subjective. This system aims to:

- Reduce manual effort by automating resume evaluation.
- Improve accuracy in shortlisting candidates.
- Ensure fair and unbiased screening.

## AI-BASED RESUME SCREENING SYSTEM USING SPRING AI

### Tech Stack:

- Backend: Spring Boot, Spring AI (OpenAI API), Java
- Frontend: Thymeleaf, HTML, CSS, JavaScript
- Database: MongoDB
- File Handling: Apache PDFBox (for extracting text from PDFs)



### Tech Stack

- **Backend:** Spring Boot, Spring AI (OpenAI API), Java
- **Frontend:** Thymeleaf, HTML, CSS, JavaScript
- **Database:** MongoDB
- **File Handling:** Apache PDFBox (for extracting text from PDFs)

### Key Features

- **Resume Upload:** Candidates upload resumes (PDF format).
- **Job Description Input:** Recruiters enter job descriptions.
- **AI Analysis:**
  - Extracts text from resumes.
  - Analyzes skills, experience, and qualifications.
  - Compare resume content with job description.
- **Candidate Scoring:** AI assigns a relevance score (0-100%).
- **Shortlisting:** Displays top candidates based on scores.
- **Reports & Insights:** Provides feedback on candidate suitability.

## 5. Portfolio Grid Gallery

### Description:

Portfolio Grid Gallery is a visually appealing web project that showcases creative works or projects in a responsive grid layout. Built using HTML, CSS, and JavaScript, it allows users to browse through a collection of portfolio items with smooth hover effects, lightbox previews, and category-based filtering for an interactive viewing experience.



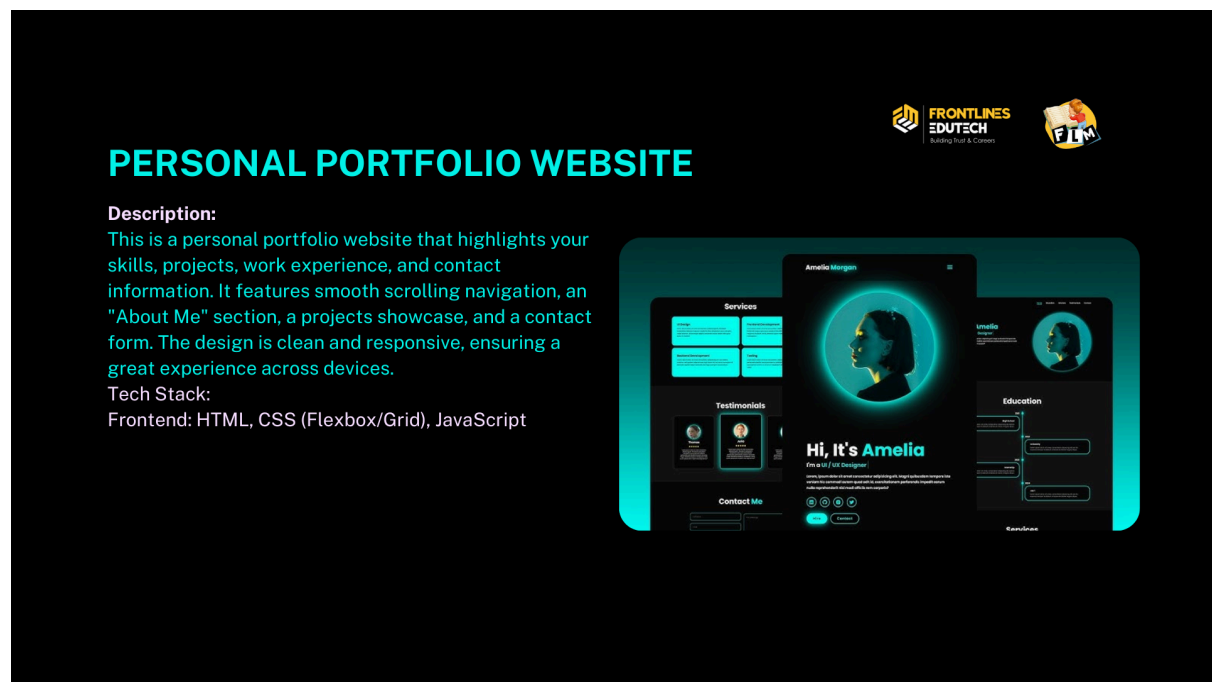
### Tech Stack:

- **Frontend:** HTML, CSS, Grid

## 6. Personal Portfolio Website

### Description:

This is a personal portfolio website that highlights your skills, projects, work experience, and contact information. It features smooth scrolling navigation, an "About Me" section, a projects showcase, and a contact form. The design is clean and responsive, ensuring a great experience across devices.



### Tech Stack:

- **Frontend:** HTML, CSS (Flexbox/Grid), JavaScript

## 7. Product Showcase Carousel

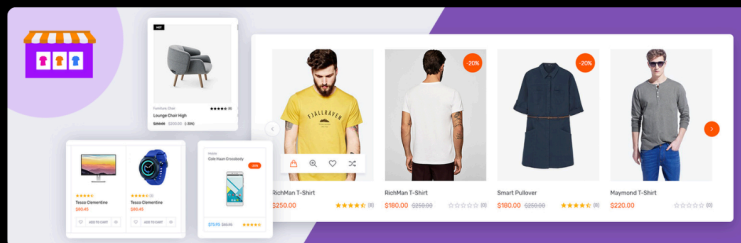
### Description:

ProductView is an interactive product showcase carousel designed for e-commerce or portfolio sites. It allows users to scroll through multiple products with images, descriptions, and prices. It includes navigation arrows, autoplay functionality, and responsive design for seamless interaction across devices.

### Tech Stack:

- **Frontend:** HTML, CSS, JavaScript

### PRODUCT SHOWCASE CAROUSEL



#### Description:

Product View is an interactive product showcase carousel designed for e-commerce or portfolio sites. It allows users to scroll through multiple products with images, descriptions, and prices. It includes navigation arrows, autoplay functionality, and responsive design for seamless interaction across devices.

#### Tech Stack:

- **Frontend:** HTML, CSS, JavaScript