MCA Semester – IV
Interim Report

Name	**ARUN B**
USN	**231VMTR00045**
Elective	**FULL STACK DEVOLOPMENT**
Date of Submission	**16/05/2025**

Project Title

Research Project submitted to Jain Online (Deemed-to-be University) In partial fulfillment of the requirements for the award of:

Master of Computer Applications

Submitted by

ARUN B

USN

231VMTR00045

Under the guidance of

Chandra Bhanu

(Faculty-JAIN Online)

DECLARATION

I, *Arun B,* hereby declare that this Project Report has been prepared by me under the guidance of *Chandra Bhanu.* I declare that this Project is towards the partial fulfillment of the credit requirement for the course "Capstone Project," which is part of the Master of Computer Applications degree given by Jain University, Bengaluru. I declare that the work done by me towards this Project is original in nature and is my own contribution.

Place: Bengaluru

Date: 16/05/2025

_____

Arun B

USN:231VMTR00045

EXECUTIVE SUMMARY

Fragrance Fusion: *A world of Aromas in one place.*

Overview:

Fragrance Fusion is a full-stack e-commerce platform targeting the fragmented perfume market. It provides a centralized marketplace for **Customers**, **Sellers**, and **Admins**, improving product discovery, personalization, and transaction security.

Tech Stack:

- **Frontend:** React.js, Redux, Axios
- **Backend:** Spring Boot, Spring Security, JWT, Spring Data JPA
- **Database:** MySQL
- **Deployment:** Docker, AWS

Scope & Objectives

Scope:

- Build a responsive web app with role-based access
- Enable product browsing, cart management, payments, and order tracking
- Implement AI recommendations and smart search

Objectives by Role:

- **Customers:** Browse/filter products, manage cart, place orders
- **Sellers:** Manage product listings, track inventory/sales
- **Admins:** Oversee users, platform security, and performance

Progress & Features Implemented

- **Authentication:** JWT-based login/registration, secured endpoints
- **Customer Features:** Product listings, category filters, basic cart
- **Seller Features:** Add/edit/delete products, view dashboard
- **UI:** Responsive design, Redux for global state

Challenges Solved:

- Redux sync with backend
- Role-based access with Spring Security

Next Steps:

- Admin features
- Stripe/Razorpay integration
- AI recommendations
- Deployment on AWS

Conclusion

Fragrance Fusion is on track to become a niche e-commerce leader. Core modules like authentication and seller management are complete. Focus now shifts to payments, admin tools, and deployment. The platform reflects scalable full-stack design and industry best practices.

.

TABLE OF CONTENTS

1.Introduction

Overview, Goals, and Significance

Fragrance Fusion is a full-stack e-commerce platform designed to transform the fragmented perfume industry by providing a **centralized marketplace** for customers, sellers, and administrators.
The **goals** of the project are:

- To offer seamless product browsing, cart management, secure payments, and order tracking.
- To empower sellers to manage inventory and track sales.
- To give administrators control over users and platform security.

Significance:
The project highlights the practical application of full-stack development skills, emphasizing **scalability, security, user experience,** and **industry-relevant practices** like cloud deployment and AI-driven features.

Problem Statement / Purpose of the Project

The perfume industry lacks a unified digital platform where multiple sellers and customers can interact easily. Customers often face difficulty in discovering products based on preferences (like scent notes or occasions), and sellers lack digital tools to manage and analyze sales effectively.

Purpose:
Fragrance Fusion aims to solve this by providing:

- Centralized access to a wide range of products.
- Personalized recommendations powered by AI.
- Secure, role-based access for customers, sellers, and admins.

3. Background and Relevance in Full Stack Development

In today's digital era, businesses demand end-to-end solutions covering frontend, backend, database management, security, and deployment.
Fragrance Fusion applies key full-stack concepts:

- **Frontend (React.js + Redux):** For dynamic, responsive user interfaces.
- **Backend (Spring Boot + JWT Authentication):** For secure business logic and API management.
- **Database (MySQL):** For structured data storage and retrieval.
- **Deployment (Docker, AWS):** For real-world cloud hosting practices.

Relevance:

This project prepares developers for real industry scenarios by integrating **authentication, payment processing, role management,** and **scalable architecture** — all essential for modern full-stack applications.

2. Project Scope and Objectives

Scope and Boundaries

The project aims to build a **full-stack e-commerce platform** specifically for the perfume industry, supporting three types of users: **Customers**, **Sellers**, and **Administrators**.

In Scope:

- Responsive web application with user authentication and role-based access
- Product browsing, advanced search/filtering, cart management, secure order placement
- Seller dashboards for inventory and sales management
- Admin panel for managing users and monitoring platform activity

Specific Objectives and Deliverables

Objectives:

- Enable customers to discover, purchase, and track fragrance products easily
- Provide sellers tools to manage their products and monitor sales performance
- Equip administrators with control over platform users and system security

Key Deliverables:

- Functional frontend (React.js + Redux) and backend (Spring Boot)
- MySQL database with normalized tables for users, products, and orders
- Secure JWT-based authentication and role-based access control
- Stripe or Razorpay payment gateway integration
- Dockerized deployment hosted on AWS

Constraints and Limitations

Constraints:

- **Third-party dependency:** Reliance on payment gateways (Stripe/Razorpay) may introduce downtime or API changes.
- **Scalability challenges:** Initial database and server configuration may need upgrades as user and product volume grows.
- **Budget constraints:** Limited hosting and API usage costs during early deployment stages.

3. Methodology

Methodologies, Frameworks, and Tools Used

The project adopted an **Agile development methodology**, with short, iterative sprints to deliver features incrementally. Regular testing, feedback, and continuous integration helped refine functionalities at every stage.

The main frameworks and tools used include **React.js** for building the frontend user interface, **Redux** for state management, and **Axios** for handling API requests. The backend was developed using **Spring Boot**, combined with **Spring Security** and **JWT authentication** to secure user sessions. For database operations, **Spring Data JPA** was used to simplify interactions with the **MySQL** database. Deployment is managed through **Docker** containers, and cloud hosting is handled on **AWS EC2**. Version control throughout the project was maintained using **Git and GitHub**.

Development Process

The project followed an **agile and iterative development process**. Work was divided into multiple sprints, each targeting specific features:

- In the first sprint, the project structure was set up with authentication mechanisms like login and signup.
- The second sprint focused on building customer-side features such as product browsing and cart management.
- The third sprint implemented seller functionalities, including product addition and inventory management.
- The fourth sprint is currently underway, focusing on admin tools, payment integration, and order tracking.

Technologies and Programming Languages Used

For the frontend, **React.js** was used to build dynamic and responsive interfaces, supported by **Redux** for managing application state and **Axios** for API communication.

On the backend, **Spring Boot** was chosen to create RESTful APIs, with **Spring Security** and **JWT** ensuring secure authentication and role-based access control.

The database was developed using **MySQL**, with **Spring Data JPA** simplifying database interaction through object-relational mapping.

Deployment technologies include **Docker** for containerization and **AWS** for cloud-based hosting. The core programming languages used in the project are **JavaScript** (for frontend) and **Java** (for backend), along with **SQL** for database queries.

4. System Architecture and Design

System Architecture and High-Level Design

Fragrance Fusion is based on a **three-tier architecture** separating the frontend, backend, and database layers:

- **Frontend:** Developed using React.js with Redux for state management. It handles the user interface, user interactions, and communicates with the backend APIs through Axios.
- **Backend:** Built with Spring Boot, it manages business logic, user authentication via JWT, and role-based access control for Customers, Sellers, and Administrators.
- **Database:** MySQL stores structured data related to users, products, orders, and reviews. Data access is managed through Spring Data JPA in the backend.

The frontend and backend interact through RESTful APIs, ensuring clear communication between layers. The backend interacts with the database to fetch, update, or delete records as required.

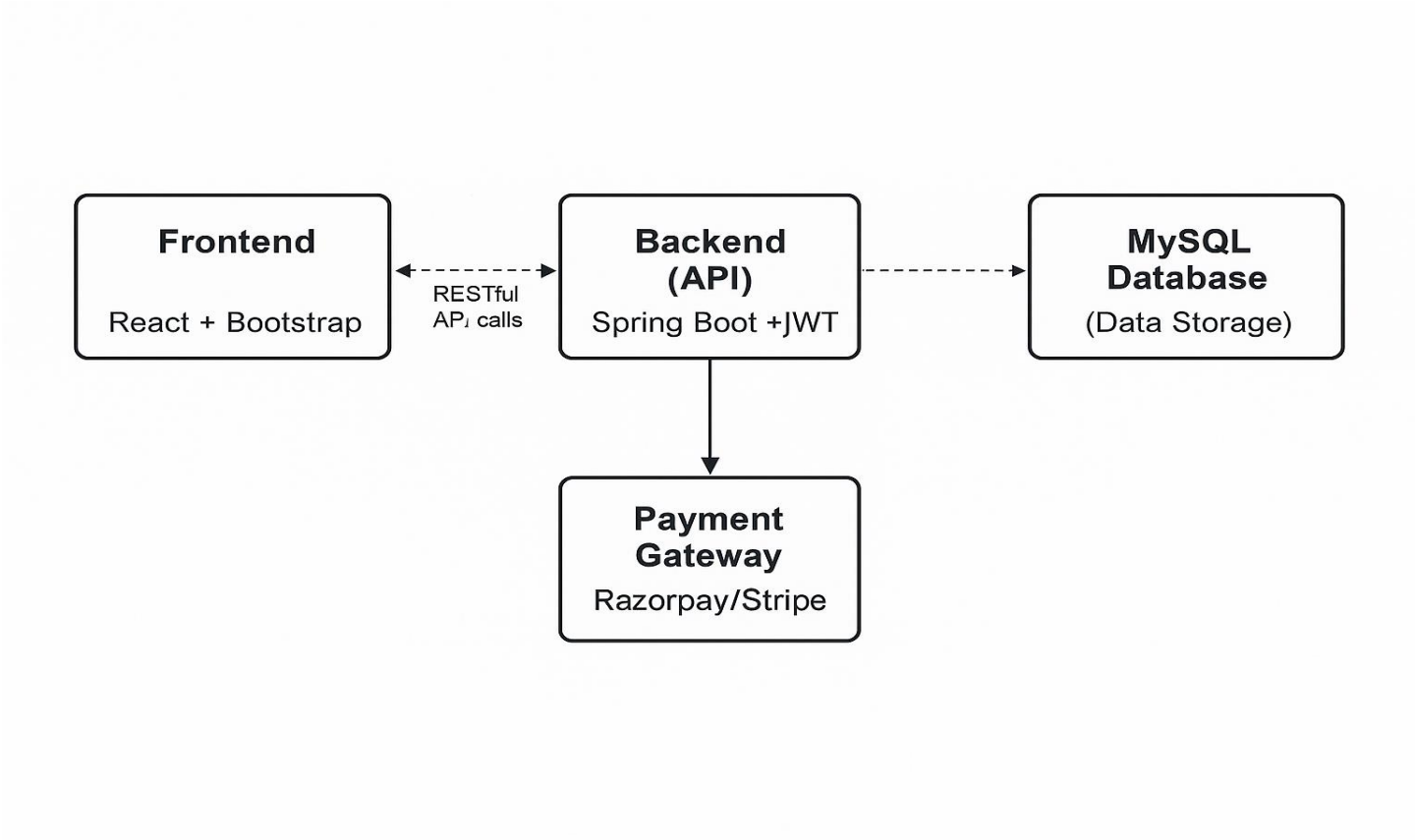
Architectural Patterns and Design Principles

Architectural Pattern:

- The project uses a **Layered Architecture** (also called N-tier architecture), which organizes code into distinct layers such as presentation, service/business logic, and data access. This improves maintainability, scalability, and security.

Design Principles:

- **Separation of Concerns:** Different parts of the application handle distinct responsibilities (UI, logic, and data access).
- **Modularity:** React components and Spring services/controllers are modular, making the system easier to test and maintain.
- **Secure by Design:** Security measures like JWT authentication and role-based authorization are integrated early into the system.
- **Scalability:** The system is designed for future growth with Docker-based containerization and cloud deployment on AWS.
- **DRY (Don't Repeat Yourself):** Code is reusable, with centralized error handling and reusable UI components and backend services.



Data Flow Summary:

- The **Frontend** sends user requests (like login, product search) to the **Backend**.
- The **Backend** processes these requests, applies business logic, checks authentication/authorization, and interacts with the **Database** if needed.
- The **Database** responds with the necessary data, which the **Backend** then sends back to the **Frontend** for display to the user.

5. Progress and Accomplishments

1. Completed Tasks and Achievements:

- **User Authentication:**
  - **JWT-based login/signup** functionality was implemented to allow secure user authentication for Customers, Sellers, and Admins.
  - **Spring Security** was integrated to handle authorization, ensuring that role-based access controls are correctly applied.
  - **Evidence:** Login/Signup forms and successful user authentication via JWT (Fig. 1).
- **Customer Functionality:**
  - Developed **product listing page** with category filters (e.g., Floral, Citrus) to allow customers to easily browse products.
  - Implemented **basic cart management** — customers can add and remove items from their cart.
  - **Evidence:** Product listing UI and working cart functionality (Fig. 2).
- **Seller Functionality:**
  - Implemented **product addition form** for sellers to create new product listings, with fields for name, description, price, and inventory.
  - Created a **seller dashboard** to allow sellers to view and manage their products.
  - **Evidence:** Seller dashboard interface with the option to add products (Fig. 3).

2. Milestones Achieved:

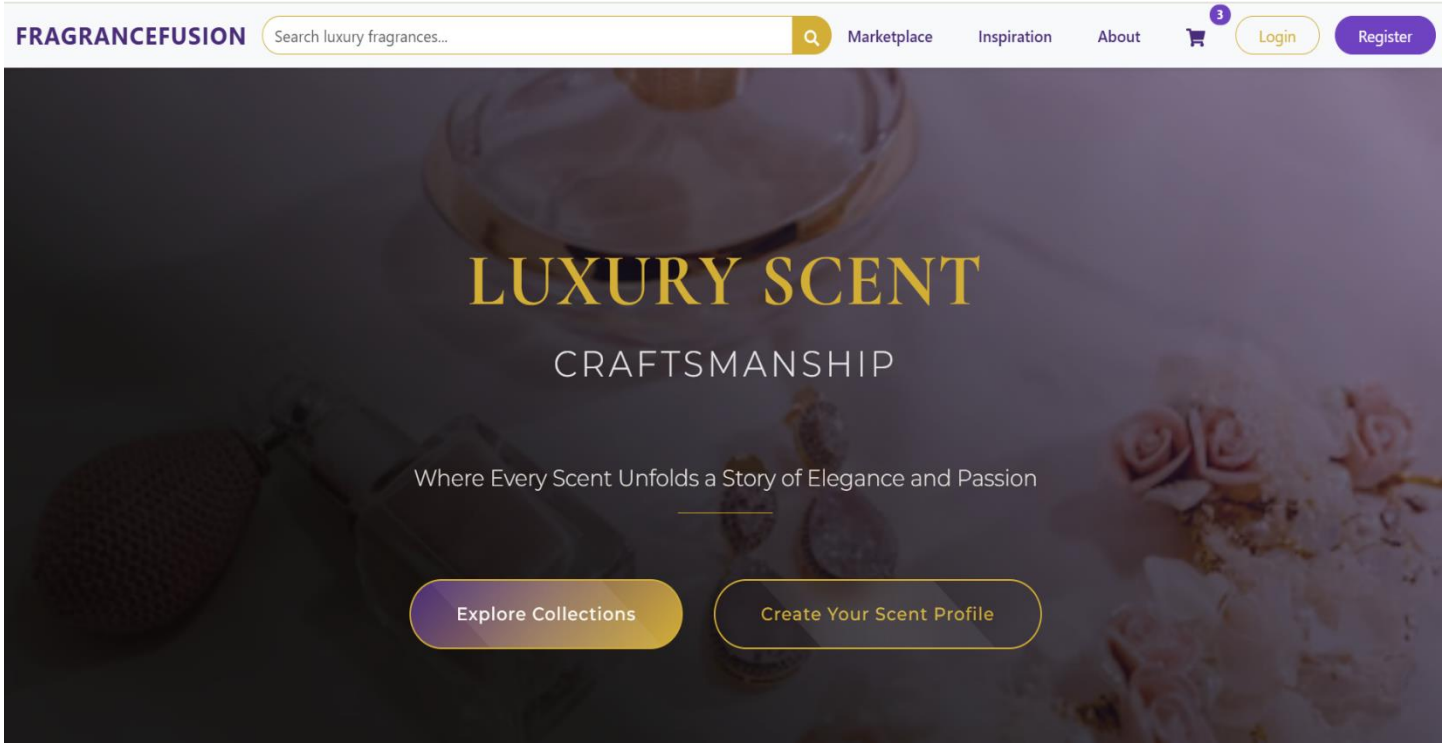
- **Authentication and Security:**
  The initial security and authentication setup was successfully completed, allowing users to securely log in and access their respective roles.
- **Basic Customer and Seller Interfaces:**
  The product listing page, cart functionality, and seller dashboard are fully functional.

3. Challenges Encountered:

- **Redux State Synchronization:**
  One of the challenges faced was syncing the cart state between the frontend and backend. This required careful handling of API responses and updates to ensure data consistency across sessions.

- **Spring Security and Role-based Access:**
  Configuring Spring Security to correctly enforce role-based access control (Customer, Seller, Admin) posed some difficulties in ensuring secure API endpoints.

4. Evidence of Implemented Features



Home Page

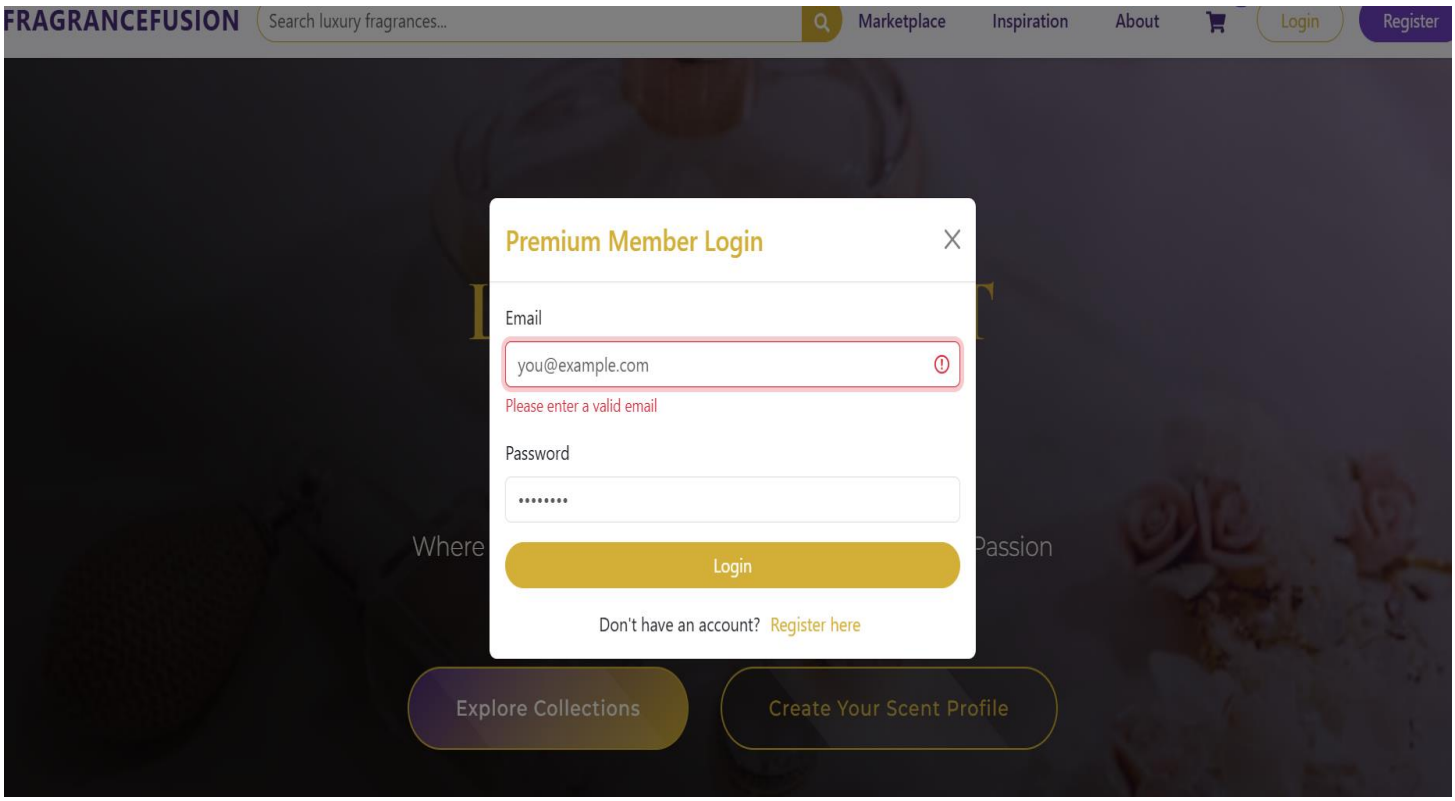


Fig 1:-User Login

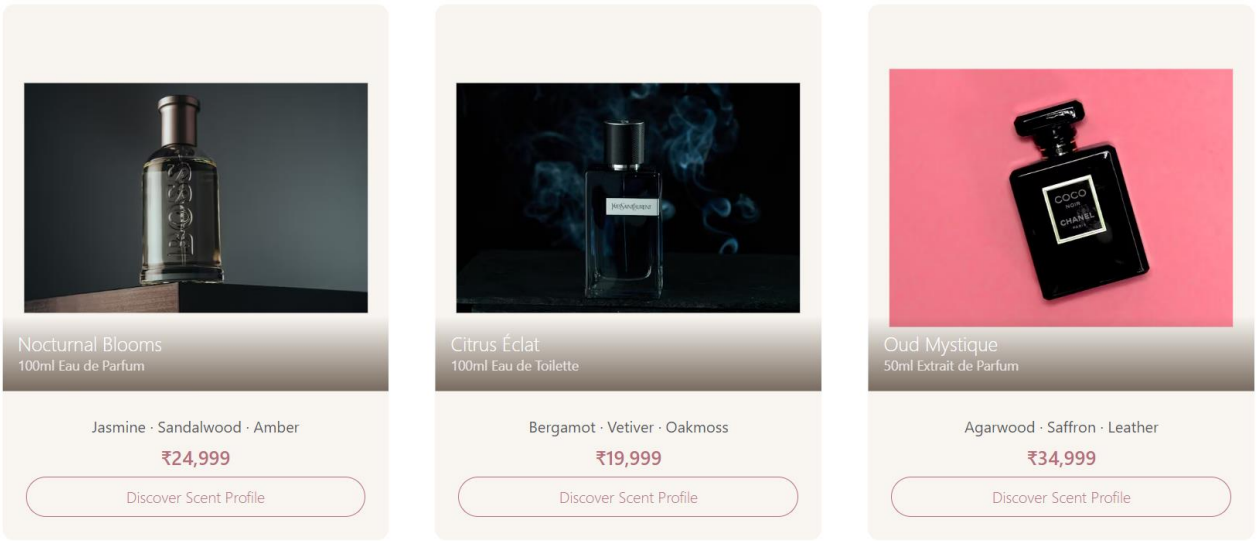


Fig 2:-customer collections

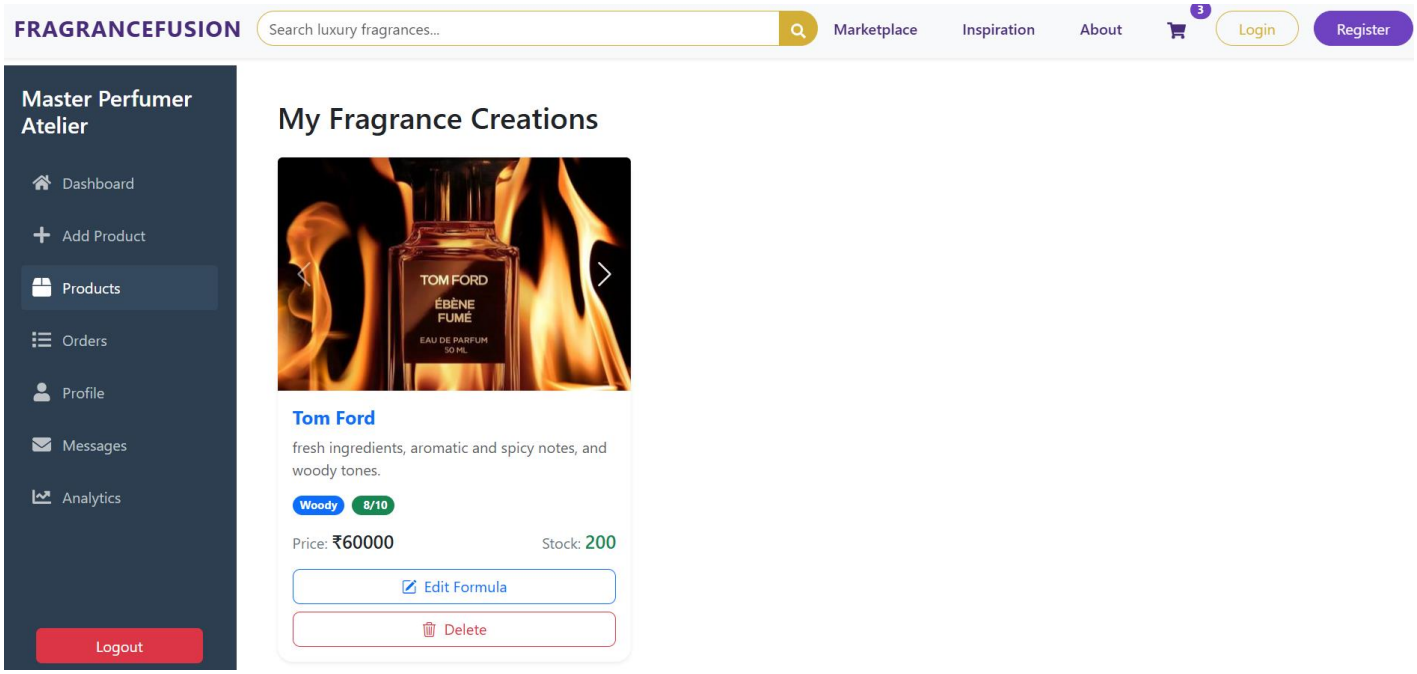


Fig 3:-Seller Dashboard

6. Future Work and Timeline

1.Remaining Tasks and Deliverables

- Marketplace Page Implementation

  Develop a dynamic page showcasing all fragrance products with filters and sorting.
- Cart Overflow Handling

  Ensure the shopping cart can manage large numbers of items without errors or UI issues.
- Payment Integration

  Connect a secure payment gateway (like Stripe Test Mode) to enable mock transactions.
- Order Summary & Confirmation Page

  After payment, display a detailed order summary including product details, prices, and order ID.
- Final Testing and Optimization

  Conduct full testing on different devices and browsers, fix any final bugs, and optimize performance.
- Deployment

  Deploy the complete website to a live production server with a custom domain (optional but ideal).

2.Updated Timeline (Deadline: May 10, 2025)

Task	Start Date	End Date	Duration	Status
Marketplace Page Development	May 15	May 17	2 days	Planned
Cart Overflow Handling	May 15	May 18	3 days (parallel)	Planned
Payment Integration (Test Mode)	May 18	May 19	1 day	Planned
Order Summary and Confirmation Page	May 18	May 19	1 day	Planned
Final Testing and Optimization	May 19	May 20	1 day	Planned
Deployment and Launch	May 19	May 20	1 day	Planned

3.Risks and Challenges

- Tight Schedule:

  Very little room for unexpected delays — any small issue could impact the May 10 deadline.
- Payment Gateway Issues:

  Even test-mode payments sometimes have integration bugs that might need quick fixes.
- Parallel Development Stress:

  Developing Marketplace and Cart Overflow in parallel requires good task management.
- Testing Time Limited:

  Only a few days are available for thorough testing, so early testing during development is recommended.

7. Conclusion

Fragrance Fusion was conceptualized to address a growing gap in the online luxury fragrance market — the lack of a dedicated, user-friendly platform where fragrance enthusiasts can discover, explore, and purchase premium scents in one elegant space. Traditional online shopping platforms often mix luxury fragrances with everyday products, making it difficult for consumers to experience the exclusivity and uniqueness these brands represent. *Fragrance Fusion* aims to redefine this experience by offering a specialized marketplace focused solely on the art and culture of fine fragrances.

The website has been designed with a strong emphasis on visual appeal, seamless navigation, and a mobile-responsive structure, ensuring that users receive a smooth and enjoyable browsing experience across all devices. From showcasing premium brands to providing smart filtering options and a dynamic cart system, every feature has been thoughtfully developed to enhance user engagement and satisfaction.
Throughout the project, a modern, space-inspired UI theme was chosen to symbolize exploration — just as users explore and find fragrances that match their identity. The design reflects not only luxury but also the idea of fragrance as a personal journey.

Key functionalities such as brand showcases, dynamic product listings, filtering systems, cart management, and responsive layouts have already been successfully completed. Upcoming phases, including Marketplace development, payment gateway integration, and final testing, are strategically scheduled to ensure a polished product is ready for launch by May 10, 2025.

Despite facing technical challenges around design optimization, payment security, and cross-device performance, the development process has been both resilient and adaptive. Every obstacle has been approached with innovation and a focus on user-centric solutions.

In conclusion, *Fragrance Fusion* is not just an e-commerce site — it is an experience built for fragrance lovers, luxury seekers, and anyone passionate about self-expression through scent. Upon completion and deployment, it will stand as a premium destination offering quality, beauty, and convenience, successfully bridging the gap between luxury fragrance brands and modern online consumers.

Project Repository (Interim Report):

The complete source code for the “Fragrance Fusion” project is available at the following GitHub repository:

https://github.com/arunraj-0627/ecommerce_.git