Name : Jenil Vaghasiya

Enroll. No. : 92310133012

System Design and Architecture

Project Title : Ignite Perfume Web Application

1. Introduction

The **Ignite Perfume Web Application** is designed as a modern e-commerce platform for perfumes. It focuses on **modularity, scalability, and reliability** to ensure smooth operation, maintainability, and a good user experience.

2. Modular Design

The system is divided into **three main modules**:

1. **Frontend (React.js)**
   - Handles UI, pages (Home, Products, Cart, Checkout), and user interactions.
   - Uses reusable components like Navbar, Product Card, and Footer.
2. **Backend (Node.js + Express)**
   - Manages APIs, authentication, business logic, and error handling.
   - Receives requests from frontend, processes them, and interacts with the database.
3. **Database (MongoDB Atlas)**
   - Stores users, products, orders, and cart details.
   - Uses indexing for fast searches.

Integration: Frontend → Backend → Database → Response to frontend.

3. Technology Stack

Component	Technology	Why Used
Frontend	React.js	Fast, responsive, reusable components
Backend	Node.js + Express	Handles multiple API requests efficiently
Database	MongoDB Atlas	Cloud-based, scalable, flexible NoSQL
Hosting	Vercel	Easy deployment, auto-builds from GitHub
Auth & Security	JWT + bcrypt	Secure user authentication

4. Scalability Plan

- **Frontend:** Multiple instances can run via Vercel to handle more users.
- **Backend:** Can be replicated; APIs handle high load efficiently.
- **Database:** Indexing, connection pooling, and future sharding for large data.
- **Monitoring:** Vercel Analytics and MongoDB Atlas track performance and uptime.
- **Caching & Rate Limiting:** Ensures faster responses and protects from overload.

5. Conclusion

The system is **modular, reliable, and scalable**, making it easy to maintain, upgrade, and handle growth. The chosen technology stack supports a **fast, secure, and smooth user experience**, while the architecture ensures clear separation between frontend, backend, and database.

Visuals to include:

- System Architecture Diagram
- Data Flow Diagram (Frontend → Backend → Database)