**Project Design Phase**

**Solution Architecture**

|  |  |
| --- | --- |
| Date | 28 June 2025 |
| Team ID | LTVIP2025TMID56240 |
| Project Name | **ShopSmart** |
| Maximum Marks | 4 Marks |

**🧠 Solution Architecture: ShopSmart**

**The solution architecture for the ShopSmart project is designed to offer an intelligent shopping assistant that helps users compare products, find the best deals, and make informed purchasing decisions. The system integrates real-time product data with AI-driven recommendations through a user-friendly web interface.**

**🔹 Architecture Objectives**

* **To help users compare products and prices across multiple platforms in real-time.**
* **To offer AI-powered recommendations based on user preferences and behavior.**
* **To streamline the shopping experience with an intuitive and responsive interface.**

| **Component** | **Description** |
| --- | --- |
| **User Interface** | **A web-based UI (using React or HTML/CSS/JS) for users to search, view, and compare products.** |
| **Product Scraper/API Layer** | **Gathers product data from e-commerce platforms using APIs or web scraping.** |
| **Recommendation Engine** | **Analyzes user data to recommend products using ML models (e.g., collaborative filtering).** |
| **Price Comparison Module** | **Compares prices for similar products across different stores.** |
| **User Data Store** | **Stores user profiles, preferences, search history, and cart information.** |
| **Notification System** | **Alerts users on price drops or deals through email or push notifications.** |

**🔁 Architecture Flow**

1. **User searches for a product via UI.**
2. **Product Scraper/API fetches data from various platforms.**
3. **Price Comparison Module ranks products.**
4. **Recommendation Engine suggests similar/better products.**
5. **Results are displayed on the UI with sorting/filtering options.**

**🛠 Technology Stack**

* **Frontend: HTML, CSS, React.js**
* **Backend: Node.js / Express.js**
* **Database: MongoDB / Firebase**
* **AI/ML: Python (scikit-learn, TensorFlow for recommendation system)**
* **Deployment: Vercel / Heroku / AWS / Netlify**

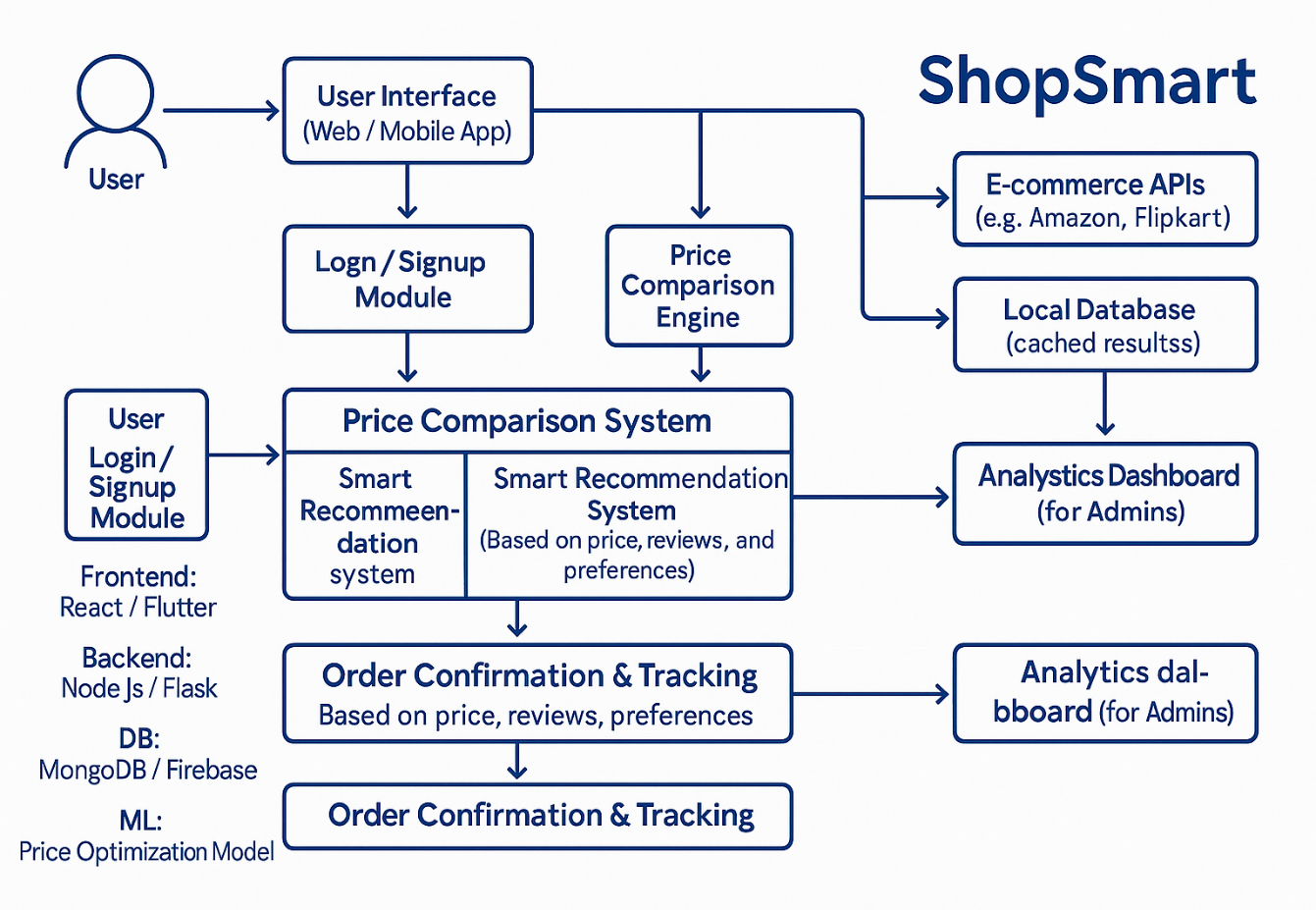
**🚀 Development Phases**

1. **Data Gathering from E-commerce APIs**
2. **Backend and Scraper/API Setup**
3. **Recommendation Model Training**
4. **Frontend Integration with Backend**
5. **Testing & Deployment**

**📈 Scalability Considerations**

* **Modular architecture allows adding more e-commerce platforms.**
* **Can scale horizontally to handle large user traffic.**
* **Can integrate mobile support and real-time user analytics.**

**Solution Architecture Diagram:**

****