High Level Design Document

Introduction

This High Level Design (HLD) document outlines the architecture and core components for **ShopSmart - AI Discount Finder**. The project is a web application that enables users to input product names and receive machine-learning-based suggestions on the best time to buy, leveraging historical price trends. The solution demonstrates data analysis, trend prediction, and practical e-commerce applications of machine learning.

---

1. System Architecture Overview

Architecture Description:
ShopSmart is a web-based application with a React frontend, a Node.js backend, and a machine-learning module for price trend analysis. The system interacts with a product price history data source and exposes a REST API for frontend-backend communication.

Module/Component	Role/Responsibility
Frontend (React)	User interface, input collection, result display
Backend (Node.js)	API endpoints, request handling, business logic
ML Engine	Analyzes price data, predicts optimal purchase timing
Data Source	Stores historical product price data

---

2. Component Interactions

Sequence Step	Interaction Description
1	User enters product name in the React frontend
2	Frontend sends request to Node.js backend API
3	Backend retrieves historical price data from Data Source
4	Backend invokes ML Engine with price data
5	ML Engine returns best time-to-buy prediction to backend
6	Backend sends prediction result to frontend
7	Frontend displays recommendation to user

---

3. Data Flow Overview

Data Flow	Source	Destination	Description
Product Query	User (Frontend)	Backend API	Product name input by user

Price Data Retrieval	Backend	Data Source	Fetch historical price data
Prediction Request	Backend	ML Engine	Send price data for analysis
Prediction Result	ML Engine	Backend	Best time-to-buy suggestion
Recommendation Delivery	Backend	Frontend	Display result to user

---

4. Technology Stack

Layer/Component	Technology/Framework
Frontend	React
Backend	Node.js, Express
ML Engine	Python (scikit-learn or similar)
Data Source	JSON/CSV files or simple DB
API	REST

---

5. Scalability & Reliability

- **Scalability:** Modular design allows independent scaling of frontend, backend, and ML engine. Stateless backend enables horizontal scaling.
- **Reliability:** Input validation and error handling at API level. ML predictions are stateless and can be retried. Data source can be upgraded to a managed database for higher reliability.
- **Security:** Basic input sanitization and API endpoint protection recommended.

End of Document