Product Requirements & Specification Document

Project Name

ShopSmart - AI Discount Finder

Description

ShopSmart is a web application that enables users to input product names and receive AI-driven suggestions for the optimal time to purchase, based on historical price trends. The project demonstrates data analysis, trend prediction, and practical e-commerce applications of machine learning.

---

1. Objectives

- Allow users to search for products and receive recommendations on the best time to buy.
- Teach users about data analysis and trend prediction in e-commerce.
- Provide a simple, intuitive interface suitable for entrepreneurship-themed learning.

---

2. Stakeholders

Role	Responsibility
Product Owner	Define requirements, approve features
Developers	Implement frontend, backend, ML model
Users	Search products, view recommendations
Data Provider	Supply historical price data

---

3. Functional Requirements

ID	Requirement
FR1	Users can input a product name via a search bar.
FR2	System fetches and displays historical price trends for the product.
FR3	ML model analyzes trends and predicts the best time to buy.
FR4	Display recommendation with supporting data visualization (e.g., line chart).
FR5	Users can view brief explanations of the prediction.
FR6	Responsive design for desktop and mobile.

---

4. Non-Functional Requirements

ID	Requirement
NFR1	System responds within 2 seconds per request.
NFR2	Secure handling of user input and data.
NFR3	Codebase is modular and maintainable.
NFR4	Accessible UI (WCAG 2.1 AA compliance).

---

5. User Stories

As a…	I want to…	So that…
User	Search for a product	I can see when to buy it at the best price
User	View price trend charts	I understand the price history
User	Get a clear recommendation	I can make informed purchase decisions

---

6. Technical Specifications

6.1 Architecture

- **Frontend:** React (with charting library, e.g., Chart.js)
- **Backend:** Node.js (Express)
- **ML Model:** Simple regression or time-series model (e.g., linear regression)
- **Data Source:** Public e-commerce price APIs or static sample datasets

6.2 Data Flow

```
graph TD
A[User Input] --> B[Backend API]
B --> C[Fetch Historical Data]
C --> D[ML Model Prediction]
D --> E[Return Recommendation & Data]
E --> F[Frontend Display]
```

6.3 Key Endpoints

Endpoint	Method	Description
`/api/search`	POST	Search product, fetch price data
`/api/predict`	POST	Run ML prediction on price trends

---

7. UI/UX Requirements

- Simple search bar on homepage

- Results page with:
  - Product name
  - Price trend chart
  - “Best time to buy” recommendation
  - Brief explanation of prediction
- Mobile-friendly layout

---

8. Acceptance Criteria

ID	Criteria
AC1	User can search and receive a recommendation within 2 seconds.
AC2	Price trend chart is clear and accurate.
AC3	Recommendation is based on actual trend analysis.
AC4	UI is responsive and accessible.

---

9. Constraints & Assumptions

- Only public or sample price data will be used.
- ML model will be simple and interpretable.
- No user authentication required.

---

10. Milestones

Milestone	Description	Due Date
Requirements Finalized	PRD approved	Week 1
Frontend MVP	Search & display UI	Week 2
Backend & ML Integration	Data fetching & prediction	Week 3
Testing & QA	Functional & UX testing	Week 4
Launch	Public release	Week 5

---

11. Out of Scope

- Real-time price tracking
- User accounts or personalization
- Advanced ML models (deep learning, etc.)

---

12. Appendix

Example ML Pseudocode

```
# Simple linear regression for price trend
import numpy as np
from sklearn.linear_model import LinearRegression

X = np.array(time_stamps).reshape(-1, 1)
y = np.array(prices)
model = LinearRegression().fit(X, y)
prediction = model.predict([[future_time]])
```

End of Document