Building an Al shopping assistant that identifies affordable product alternatives using images, text, or links, leveraging Al models and a scalable e-commerce database to generate affiliate revenue streams. Users submit product visuals, links, or descriptions. Al models convert inputs to embeddings Core Functionality for similarity search. Project Overview and Objectives System retrieves affordable alternative products with affiliate links. Affiliate commissions (5-10% per sale). Integrated Google Ads within the app Revenue Model interface. Planned premium subscription with enhanced features like exact search and price alerts. The project employs a modern full-stack architecture with AI and scalable backend services to deliver fast and accurate product recommendations. Next.js and React for UI development. Tailwind CSS for styling, ensuring Frontend Technologies responsive design. TypeScript to enforce type safety and code robustness. FastAPI framework with Python 3.11 for efficient API endpoints. Initial use of SQLite for development, migrating to MySQL for production Backend and Database scalability. Technology Stack and Components Database stores product metadata including affiliate links and pricing. CLIP model for extracting image embeddings from product visuals. MiniLM or similar models for generating Al and Search Infrastructure text embeddings from descriptions. FAISS library utilized for fast similarity search on vector embeddings. Scraping performed via Python libraries like BeautifulSoup, Requests, or through affiliate APIs. Cloud hosting starting with DigitalOcean or Azure, with plans for AWS/GCP Data Acquisition and Hosting scalability. Docker containers with CI/CD pipelines for consistent deployment and updates. A seamless multi-input system enables users to discover deals via images, links, or text with AI-powered backend processing. User uploads an image or screenshot of a product. Or pastes a product link from social or e-Input Stage commerce platforms. Or types the product name or a brief description. Images processed by CLIP to generate embedding vectors. Text inputs handled by MiniLM to produce Processing Stage embeddings. User Interaction and Workflow Embeddings sent to the backend for similarity searching. FAISS index locates top matching product embeddings from the database. Mapping from embeddings to product IDs Search and Retrieval enables detailed lookups. Details like price, image URL, and affiliate link are fetched from the database. Backend returns structured JSON containing product alternatives. Frontend presents options with prices and Output and Display "Buy" buttons linked to affiliates. A "Cheapest option" is subtly blurred to tease premium features. Outlined steps show the end-to-end data flow from user input to displayed results with revenue integration. Develop frontend UI for multipath input (image, link, text). Implement backend endpoints to process inputs and generate embeddings. Build and maintain product vector Action Items database indexed by FAISS. Implementation Plan and Workflow Diagram Set up data scraping and integration with affiliate APIs for product info. Deploy scalable hosting with containerization and CI/CD pipelines. User submits image/text/link → Backend (FastAPI) processes input → Al models (CLIP/MiniLM) generate embeddings → FAISS similarity search → Simplified Workflow Retrieve product IDs > Query MySQL database → Fetch product details → Send structured data to frontend → Display affordable alternatives with affiliate links.

Al Engines: CLIP for images, MiniLM for text

Storage: From SQLite in development to

Search Technology: FAISS for fast vector

Infrastructure: FastAPI backend paired

Monetization: Affiliate commissions,

Google ads, and upcoming premium

User Outcome: Easy discovery of affordable

alternatives linked to trusted e-commerce

MySQL in production for product data.

embeddings.

Summary of Core Elements

similarity matching.

with Next.js frontend.

subscriptions.

platforms.

Al-Powered Product
Discovery & Deal
Finder