

III. Development environment

A. Choice of software development platform

Which platform and why? (e.g., Windows, Linux, Web, or etc.)

The best platform for this project is a web-based platform. This choice allows for cross-platform accessibility, enabling users on any device with a browser (desktop, tablet, smartphone) to scan barcodes and compare prices without needing a specific OS like Windows or Linux. It also ensures real-time data access for comparing supermarket prices, easy updates, and scalability—important for handling data across multiple supermarkets efficiently.

Which programming language and why?

For this project, Python is an ideal programming language. Here's why:

1. **Libraries for Barcode Scanning and Data Handling:** Python has powerful libraries such as ZBar and pyzbar for barcode scanning and pandas for data manipulation, which are essential for retrieving and handling product information.
2. **Web Development Compatibility:** Python works well with web frameworks like Flask or Django, which can simplify the development of the web-based platform and API endpoints for retrieving product and price data.
3. **Ease of Integration with Databases and APIs:** Python's extensive library support makes it easy to integrate with various databases (like MySQL or PostgreSQL) and external APIs, which is essential for retrieving real-time price data.
4. **Community Support and Flexibility:** Python's popularity and extensive community support mean there are many resources and libraries available, which can help with rapid development and troubleshooting.

In summary, Python's versatility and the tools it offers make it well-suited for developing a barcode scanning app with real-time data capabilities.

Provide a cost estimation for your build. (including any purchase of software/hardware)

1. Infrastructure and Hosting

- **Web Hosting and Server Costs:** For initial deployment, a basic cloud instance on platforms like AWS, Google Cloud, or Heroku can range from \$100–\$200/month.
- **Database Hosting:** A managed database (e.g., Firebase, MongoDB Atlas, or a small MySQL instance) could be around \$20–\$50/month.

Estimated Total for 6 months: \$720–\$1,200

2. Barcode Scanning API (if using a third-party service)

- Optional Cost: If we decide to use a third-party API for barcode scanning (instead of developing your own or using open-source libraries), it could cost about \$50–\$100/month. This option can save development time and improve scanning accuracy.

6-Month Total: \$300–\$600

3. Miscellaneous

- Domain Registration: Around \$10–\$15 per year for a custom domain.
- SSL Certificate: Either free (e.g., Let's Encrypt) or \$50–\$100/year for a premium SSL certificate.
- Testing and Deployment Tools: Minimal costs if free tools or student licenses are used; otherwise, set aside \$100–\$200 for any necessary testing resources.

Revised Total Estimated Cost

- Infrastructure & Hosting: \$720–\$1,200
- Barcode Scanning API (Optional): \$300–\$600
- Miscellaneous: \$60–\$215
- Overall Total: \$1,080–\$2,015

This revised estimate covers the essential costs for launching and maintaining the app over the first six months, focusing only on necessary operational expenses.

Provide clear information of your development environment (e.g., version of software, OS version, your computer resources). Using any commercial cloud platform (e.g., AWS, GCP) is definitely a BONUS.

1. Operating System

- OS: Windows 11 Home

2. Computer Hardware Specifications

- Processor: 12th Gen Intel® Core™ i5-12500H @ 2.50 GHz
- Installed RAM: 16.0 GB (15.7 GB usable)
- System Type: 64-bit Operating System, x64-based processor

3. Development Tools and Software Versions

- IDE: Microsoft Visual Studio Code (latest version) with extensions for Python, SQL, Git, and web development.

- Python: Version 3.10+ for compatibility with web frameworks and data libraries.
- Node.js: Version 16.x or later, if using JavaScript-based front-end frameworks.
- Git: Version 2.35 or later for version control.
- Browser: Latest versions of Chrome for web app testing.

4. Frameworks and Libraries

- Backend Framework: Flask 2.x or Django 4.x for building RESTful APIs.
- Frontend Framework: React 17.x or Vue 3.x for a responsive user interface.
- Database: MySQL 8.x or PostgreSQL 13.x, depending on the specific requirements (either installed locally or on a cloud instance).
- Barcode Scanning Library: pyzbar for Python, providing functionality for scanning barcodes.
- API Documentation: Swagger (OpenAPI 3.0) to document and test API endpoints.

5. Cloud Platform (Bonus)

- Platform: AWS or Google Cloud Platform (optional for deployment and scalability)
- Recommended Services:
 - Compute: EC2 (AWS) or Compute Engine (GCP) to host the backend.
 - Database: Amazon RDS or Google Cloud SQL specifically with MySQL or PostgreSQL, offering a managed database service.
 - Storage: S3 (AWS) or Google Cloud Storage for storing app assets or any uploaded data.

This environment is tailored to leverage Microsoft Visual Studio Code and a choice between MySQL and PostgreSQL, providing flexibility and scalability suitable for development and deployment.

B. Software in use

Any existing software or algorithm in use? (doing a similar task as your proposal; provide a proper reference if there is any)

1. Amazon Price Comparison and Product Scanner

- Description: Amazon's mobile app allows users to scan product barcodes and instantly access product information, reviews, and pricing within Amazon's marketplace.

- Reference: Amazon Mobile App (Available on iOS and Android).
- Relevance: This app uses barcode scanning to retrieve product information, enabling users to compare Amazon's prices with in-store prices, which is similar to your project's functionality.

2. ShopSavvy Barcode Scanner

- Description: ShopSavvy is an app that lets users scan barcodes to see pricing and availability across multiple online and local stores.
- Reference: ShopSavvy (App Store, Google Play Store).
- Relevance: ShopSavvy's barcode scanning technology and price comparison feature across multiple retailers align closely with the goals of your app.

3. Google Lens

- Description: Google Lens can scan barcodes and provide detailed product information, including prices, from various online sources.
- Reference: Google Lens (Integrated in Google Search and Google Photos).
- Relevance: Google Lens leverages advanced image recognition and search capabilities, retrieving product information through barcodes, which resembles the core functionality of your proposed app.

4. Barcode Scanner Algorithm

- Description: Algorithms such as ZBar and pyzbar are commonly used in Python for barcode detection and decoding in apps and services.
- Reference: pyzbar Python library (<https://pypi.org/project/pyzbar/>).
- Relevance: pyzbar provides an open-source solution for reading barcodes in images, which could be part of the foundation for the barcode scanning functionality in your app.

These references illustrate existing solutions and technologies that provide similar barcode scanning and price comparison features. For your app, integrating similar barcode decoding libraries and developing a comparison feature across supermarkets would differentiate it by focusing on price comparisons specifically within the grocery retail sector.