MetaHunt Project Documentation

Etsy Web Crawling & Data Extraction Project

Team Members: 4 | **Platform**: Etsy | **Tools**: Selenium, Streamlit, RSS, Public API, Graphviz

Team Structure & Responsibilities

🔍 Member 1 -- Crawlability & API Specialist

- 🧭 **Step 1: Fetch robots.txt** -- Accessed and read `https://www.etsy.com/robots.txt` to identify crawl rules such as allowed/disallowed URLs, crawl-delay, and sitemap entries.
- 🕵️ **Step 2: Check Crawl Permissions** -- Verified if crawling is allowed for specific Etsy category pages using `rp.can_fetch()` logic.
- 📰 **Step 3: Parse RSS Feeds** -- Extracted latest product updates from Etsy shops like *CaitlynMinimalist* using `feedparser`.
- 🔗 **Step 4: Access Public API** -- Queried Etsy's internal API to retrieve product data like title, price, and image using `requests`.

🛠️ Member 2 -- Content Extractor & Scraper

- 💻 **Step 1: Launch Chrome Headless** -- Used `undetected_chromedriver` to bypass bot detection and render dynamic JavaScript content.
- ⏳ **Step 2: Wait for Content** -- Implemented `WebDriverWait` to ensure page is fully loaded before scraping.
- 🛍️ **Step 3: Extract Product Data** -- Retrieved product titles, prices, images, review counts, and links using CSS selectors.
- 📊 **Step 4: Store & Export** -- Structured the extracted content into a pandas DataFrame and exported it to an Excel file.

📊 Member 3 -- Dashboard & Visualization Designer

- 📈 **Step 1: Setup Streamlit App** -- Designed a user-friendly dashboard using Streamlit with wide layout configuration.
- 🚦 **Step 2: Show Crawlability Score** -- Used a formula to score the site's crawlability and displayed it using metrics and progress bars.
- 📉 **Step 3: Visualize Product Data** -- Leveraged Plotly to plot price comparisons of top products.
- 🗺️ **Step 4: Visual Sitemap** -- Built a mock sitemap using Graphviz to represent the site's structure from homepage to product pages.
- 💡 **Step 5: Recommendations** -- Listed interactive, styled crawling tips and best practices.

📄 Member 4 -- Documentation & Deployment

- 📝 **Step 1: Write Docs** -- Documented all group roles, tools, and processes in an organized Word file.
- 📑 **Step 2: Create README** -- Outlined project setup, requirements, usage instructions, and key insights.
- 🚀 **Step 3: Deploy Dashboard** -- Ran the dashboard locally and prepared it for Streamlit Cloud deployment.
- 🧪 **Step 4: Final Testing** -- Verified all visual components and data sources work correctly post-deployment.

Project Structure Diagrams

Web Crawling Solution for Etsy Marketplace using Selenium, Streamlit, RSS, Public API, and Graphviz

Team Structure & Responsibilities:

- **Crawlability & API Specialist**: Fetch robots.txt, Check crawl permissions, Parse RSS feeds, Access public API
  - Tools: Requests, RobotParser, Feedparser
- **Content Extractor & Scraper**: Launch Chrome headless, Wait for dynamic content, Extract product data, Store & export data
  - Tools: Selenium, undetected_chromedriver, Pandas
- **Dashboard & Visualization Designer**: Setup Streamlit app, Show crawlability score, Visualize product data, Create visual sitemap
  - Tools: Streamlit, Plotly, Graphviz
- **Documentation & Deployment**: Write documentation, Create README, Deploy dashboard, Final testing
  - Tools: Word, Markdown, Streamlit Cloud

Streamlit Dashboard - Interactive Web Interface

Main Components:

- **Crawlability Score**: Visual representation of site accessibility metrics
- **Product Comparisons**: Interactive price and popularity charts
- **Visual Sitemap**: Graphical representation of Etsy's structure
- **Best Practices**: Recommendations for effective crawling
- **Data Export**: Options to download structured data

Built with: Streamlit • Plotly • Graphviz • Pandas • CSS

Five-Layer Architecture:

1. **Data Sources Layer**: Etsy Product Pages, Etsy Public API, Shop RSS Feeds, robots.txt
2. **Collection Layer**: Selenium, Undetected Chrome, Feedparser, Requests
3. **Processing Layer**: Pandas, NumPy, Data Cleaning
4. **Visualization Layer**: Streamlit, Plotly, Graphviz, Interactive UI
5. **User Interface**: Final dashboard interface

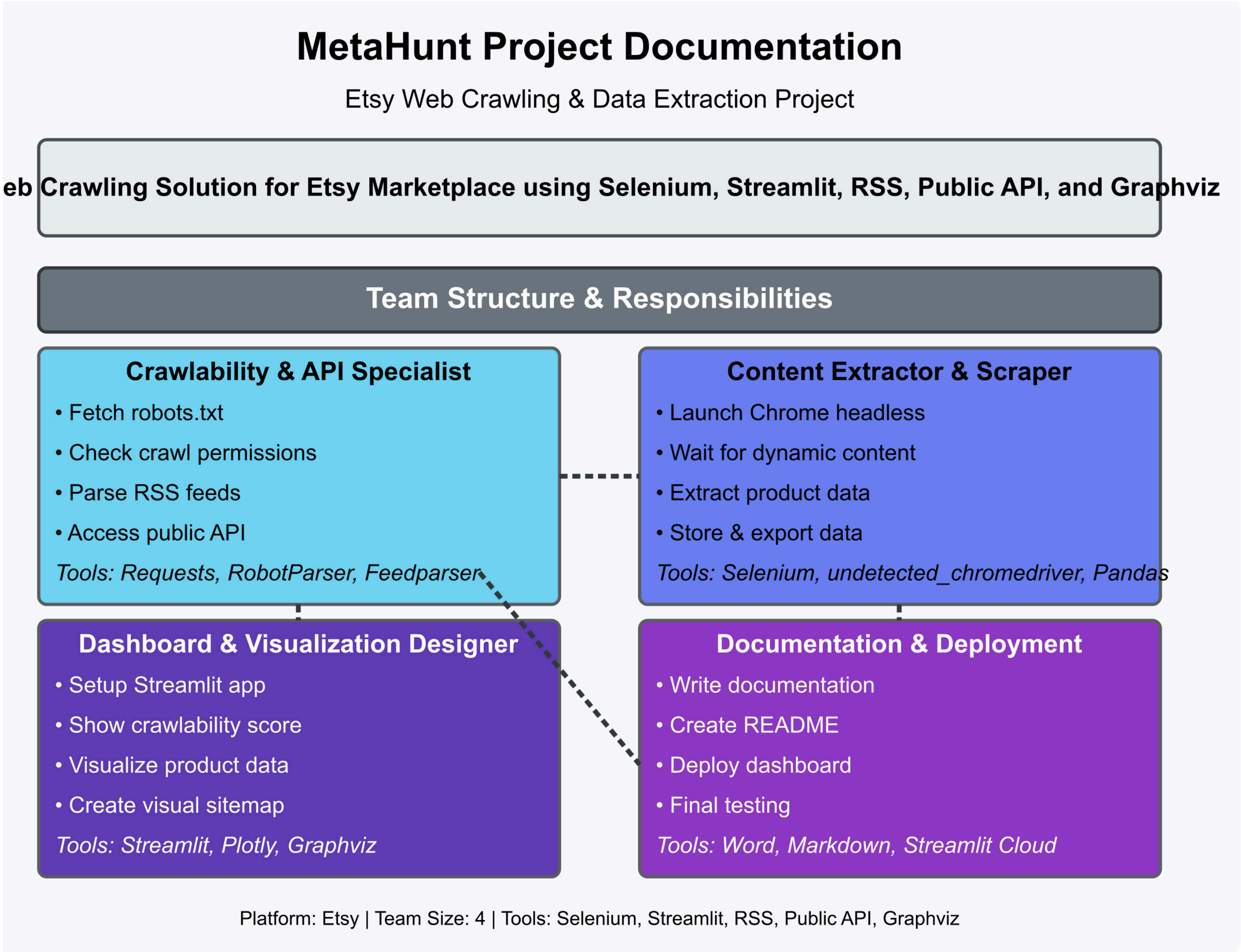
Five-Stage Implementation Process:

1. **Preparation Phase**: Research Etsy's structure • Identify crawlability rules • Select appropriate tools
2. **Data Collection Phase**: Configure crawlers • Implement browser automation • Set up API connections
3. **Processing Phase**: Clean extracted data • Calculate metrics • Generate site structure maps
4. **Presentation Phase**: Design dashboard interface • Create visualizations • Implement filtering
5. **Deployment Phase**: Test all components • Prepare cloud configuration • Document usage

Summary

The MetaHunt project provides a comprehensive web crawling and data extraction solution for the Etsy marketplace. Through a coordinated effort of four team members with specialized roles, the project successfully implements a complete data pipeline from collection to visualization. The solution adheres to ethical web crawling practices by respecting robots.txt directives and employs multiple data sources including direct page extraction, RSS feeds, and public APIs.

The final product is delivered as an interactive Streamlit dashboard that presents Etsy product data through intuitive visualizations while also providing insights about the site's structure and crawlability metrics. The five-layer technical architecture ensures clean separation of concerns from data acquisition to user presentation.



MetaHunt Project Documentation

Etsy Web Crawling & Data Extraction Project

eb Crawling Solution for Etsy Marketplace using Selenium, Streamlit, RSS, Public API, and Graphviz

Team Structure & Responsibilities

Crawlability & API Specialist

- Fetch robots.txt
- Check crawl permissions
- Parse RSS feeds
- Access public API

Tools: Requests, RobotParser, Feedparser

Content Extractor & Scraper

- Launch Chrome headless
- Wait for dynamic content
- Extract product data
- Store & export data

Tools: Selenium, undetected_chromedriver, Pandas

Dashboard & Visualization Designer

- Setup Streamlit app
- Show crawlability score
- Visualize product data
- Create visual sitemap

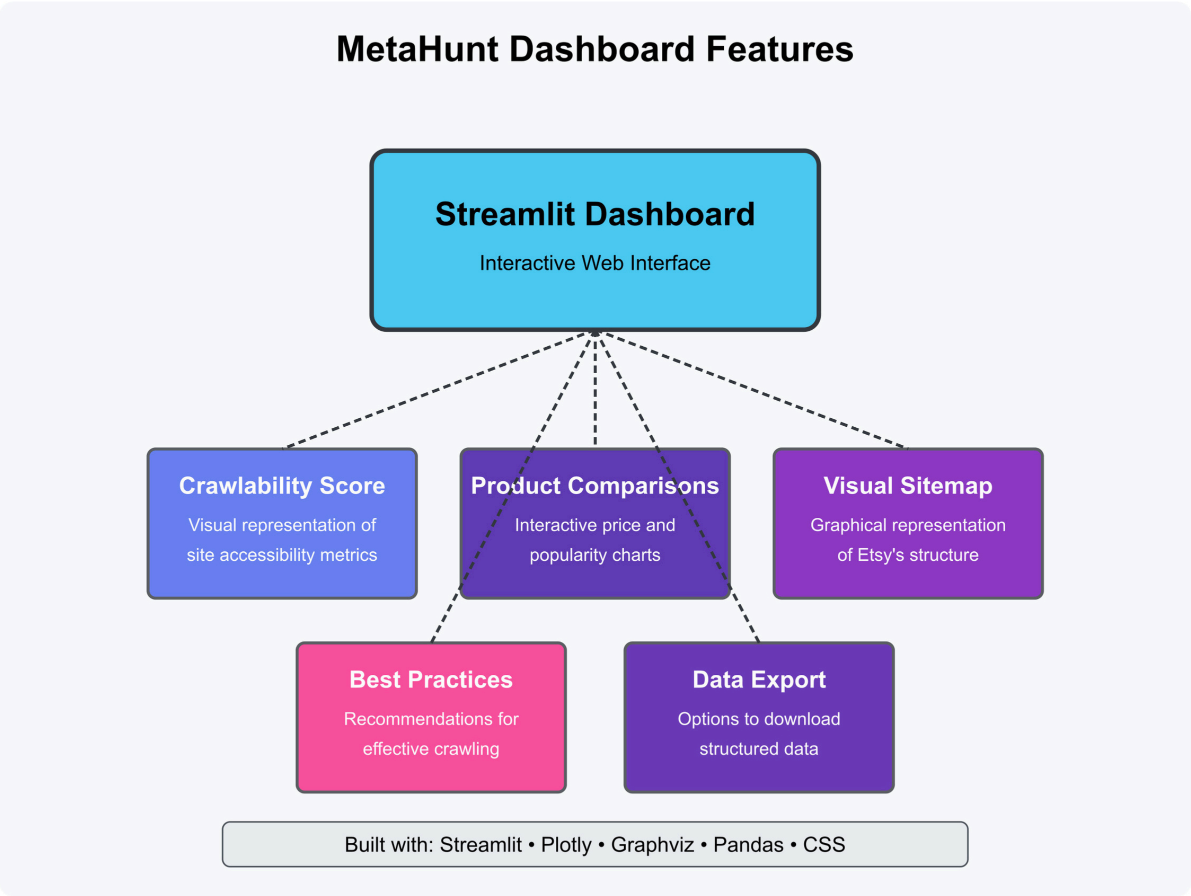
Tools: Streamlit, Plotly, Graphviz

Documentation & Deployment

- Write documentation
- Create README
- Deploy dashboard
- Final testing

Tools: Word, Markdown, Streamlit Cloud

Platform: Etsy | Team Size: 4 | Tools: Selenium, Streamlit, RSS, Public API, Graphviz



MetaHunt Dashboard Features

Streamlit Dashboard

Interactive Web Interface

Crawlability Score

Visual representation of site accessibility metrics

Product Comparisons

Interactive price and popularity charts

Visual Sitemap

Graphical representation of Etsy's structure

Best Practices

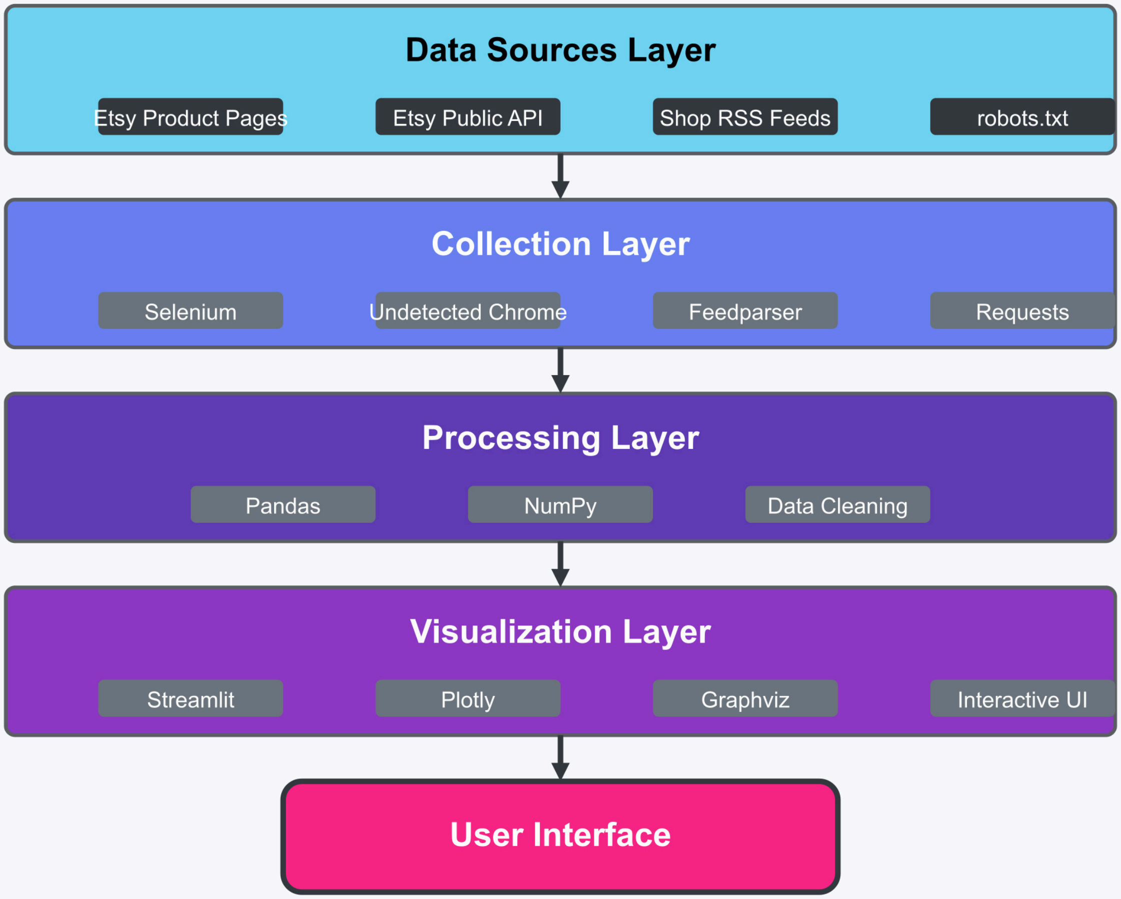
Recommendations for effective crawling

Data Export

Options to download structured data

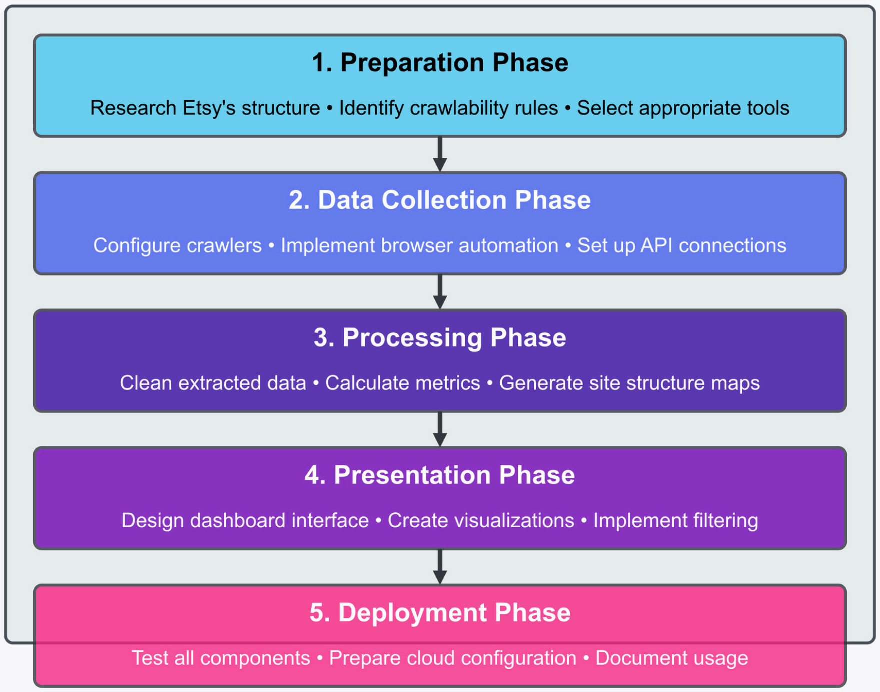
Built with: Streamlit • Plotly • Graphviz • Pandas • CSS

MetaHunt Technical Architecture



Five-Layer Architecture for the MetaHunt Etsy Crawling Project

MetaHunt Implementation Process


1. Preparation Phase
Research Etsy's structure • Identify crawlability rules • Select appropriate tools
2. Data Collection Phase
Configure crawlers • Implement browser automation • Set up API connections
3. Processing Phase
Clean extracted data • Calculate metrics • Generate site structure maps
4. Presentation Phase
Design dashboard interface • Create visualizations • Implement filtering
5. Deployment Phase
Test all components • Prepare cloud configuration • Document usage

Five-Stage Implementation Process for the MetaHunt Etsy Crawling Project