

<https://chat.openai.com/c/3c28c9a9-3024-4d4b-bb42-989bf4bbdc8e>



write a software specification for a self-hosted web-UI "universal wishlist" program, that acts like the Amazon wishlist does for products on Amazon, but this software will be for any product on the internet. the "killer feature" is being able to add a link and the program will parse the webpage for information like product name , price, rating (on that website), and other details about the product.



# Software Specification: Universal Wishlist Web-UI Program



## 1. Introduction

The Universal Wishlist Web-UI program is a self-hosted web application that allows users to create and manage wishlists for products from any website on the internet. Inspired by the Amazon wishlist functionality, this software provides a "killer feature" that automatically parses webpages to extract relevant information about products, including the product name, price, rating (on that website), and other details. This specification outlines the features, functionality, and system requirements of the Universal Wishlist Web-UI program.

## 2. Features

### 2.1 User Management

- User Registration: Users can create an account by providing their email address and setting a password.
- User Authentication: Registered users can securely log in to their accounts using their email address and password.
- User Profile: Each user will have a profile that displays their basic information and preferences.

### 2.2 Wishlist Management

- Create Wishlist: Users can create multiple wishlists, each with a unique name and optional description.
- Add Product: Users can add products to their wishlists by providing a URL/link to the product webpage.
- Parse Product Information: The program will automatically extract relevant product information from the provided URL, including product name, price, rating (on the website), and other details.
- Edit Product Details: Users can manually edit the parsed product information or add additional details.
- Categorize Products: Users can assign categories or tags to the products to organize their wishlists efficiently.
- Mark as Purchased: Users can mark products as purchased or indicate their status.
- Share Wishlist: Users can generate a unique URL to share their wishlists with others.

### 2.3 Search and Filtering

- Search Wishlist: Users can search for specific products within their wishlists by name, category, or tags.
- Filter Wishlist: Users can filter their wishlists based on various criteria such as category, price range, rating, etc.

### 2.4 Notification and Reminders

- Wishlist Updates: Users will receive notifications when new products are added or changes are made to their wishlists.
- Price Drop Alerts: Users can set price thresholds for products and receive notifications when the price drops below the specified limit.

### 2.5 User Interaction

- Comment and Discussion: Users can leave comments and initiate discussions on specific products or wishlists.
- Like and Share: Users can like or share products or wishlists on popular social media platforms.
- Recommendations: Users can receive personalized product recommendations based on their wishlists and browsing history.

### 2.6 System Administration

- Administrator Account: The system will have an administrator account with elevated privileges to manage user accounts and system settings.
- System Logs: Detailed logs will be maintained to track user activities and system events for security and troubleshooting purposes.

## 3. System Requirements

### 3.1 Technology Stack

The Universal Wishlist Web-UI program will be developed using the following technologies:

- Backend Framework: Node.js
- Web Framework: Express.js
- Database: MySQL or MongoDB
- Frontend Framework: React.js
- HTML, CSS, and JavaScript
- Web Scraping Library: BeautifulSoup or equivalent
- Authentication and Security: JWT (JSON Web Tokens)
- Deployment: Docker, Nginx

### 3.2 Functional Requirements

- The program should accurately parse the product information from a given URL and store it in the database.
- User authentication and authorization should be implemented to ensure secure access to the application.
- User inputs, including product details and comments, should be properly validated to prevent malicious activities.
- The application should be responsive, providing an optimal user experience across different devices and screen sizes.
- The search functionality should efficiently retrieve and display relevant products based on user queries.
- Notifications and alerts should be sent in real-time or near real-time to ensure timely updates.
- The system should handle concurrent user requests and provide acceptable performance.

### 3.3 Non-Functional Requirements

- The application should have an intuitive and user-friendly interface to ensure ease of use.
- The program should be scalable and able to handle a growing number of users and wishlists.
- The system should be robust and handle exceptions gracefully to avoid downtime or data loss.
- Security measures should be implemented to protect user data and prevent unauthorized access.
- Regular backups of the database should be performed to prevent data loss in case of failures.
- The software should be well-documented, including installation instructions, user guides, and API documentation.

## 4. Conclusion

The Universal Wishlist Web-UI program aims to provide users with a versatile platform for managing wishlists for products from any website on the internet. By leveraging web scraping techniques, the software automates the extraction of product details, saving users time and effort. This software specification outlines the core features, system requirements, and technology stack required to develop the Universal Wishlist Web-UI program.