

**Groceries List on The Website**

**Course Title: Programming for Information System**

**Lecturer Name: Paul Laird**

**Module Title: B9IS123**

**Group Member: Mahendra Vajja (10639742)**

**Contents**

[**1.** **Design** 4](#_Toc142607392)

[**1. UI Design:** 4](#_Toc142607393)

[**2.Navigation** 4](#_Toc142607394)

[**3.Search function** 4](#_Toc142607395)

[**4.Product catalogue** 4](#_Toc142607396)

[**5.Product page** 4](#_Toc142607397)

[**6.Basket and payment** 4](#_Toc142607398)

[**7.User Accounts** 4](#_Toc142607399)

[**8.Special offers and promotions** 4](#_Toc142607400)

[**9.Images and media of reaction** 4](#_Toc142607401)

[**10.Accessibility** 4](#_Toc142607402)

[**2.** **Features** 5](#_Toc142607403)

[**1.Registration and user account** 5](#_Toc142607404)

[**2.Search and Filter** 5](#_Toc142607405)

[**3.Product categories and sub-categories** 5](#_Toc142607406)

[**4.Product catalogue** 5](#_Toc142607407)

[**5.Product details page** 5](#_Toc142607408)

[**6.Basket** 5](#_Toc142607409)

[**7.Wishlist** 5](#_Toc142607410)

[**8.Promotions and Discounts** 5](#_Toc142607411)

[**9.Featured products** 5](#_Toc142607412)

[**10.User reviews and ratings** 5](#_Toc142607413)

[**11.Secure Payment** 5](#_Toc142607414)

[**12.Customer's order** 5](#_Toc142607415)

[**13.Order tracking** 5](#_Toc142607416)

[**14.Delivery options** 5](#_Toc142607417)

[**15.Location-based service** 5](#_Toc142607418)

[**16. Responsive Design** 5](#_Toc142607419)

[**17.Personalized recommendations** 5](#_Toc142607420)

[**3. Using Technologies** 6](#_Toc142607421)

[**About HTML** 6](#_Toc142607422)

[**About CSS** 6](#_Toc142607423)

[**About Python Flask** 6](#_Toc142607424)

[**About SQLite3** 6](#_Toc142607425)

[**1. Serverless and Standalone** 6](#_Toc142607426)

[**2. Zero setting** 6](#_Toc142607427)

[**3. ACID Compliant** 6](#_Toc142607428)

[**4. Data types** 6](#_Toc142607429)

[**5. SQL Support** 6](#_Toc142607430)

[**6. Index** 7](#_Toc142607431)

[**7. Activities** 7](#_Toc142607432)

[**8. Small footprints** 7](#_Toc142607433)

[**9. Embeddable** 7](#_Toc142607434)

[**10. Cross-platform** 7](#_Toc142607435)

[**11. Community support** 7](#_Toc142607436)

[**4. References** 11](#_Toc142607437)

[**5. GITHUB Link** 11](#_Toc142607438)

**List of Figures**

[**Figure 1.List of groceries stored in backend Database** 7](#_Toc142607233)

[**Figure 2. Python code for grocery list application.** 8](#_Toc142607234)

[**Figure 3. HTML code for frontend design.** 8](#_Toc142607235)

[**Figure 4. CSS Code for grocery list application.** 9](#_Toc142607236)

[**Figure 5. Adding Items in grocery list** 9](#_Toc142607237)

[**Figure 6. Adding "Coffee" item in grocery items list** 10](#_Toc142607238)

[**Figure 7. Checking an item in grocery list** 10](#_Toc142607239)

[**Figure 8.Checking an item "Apples" in grocery list** 10](#_Toc142607240)

# **Design**

Grocery store website design involves creating user-friendly, visually appealing, and effective interfaces to help customers find and purchase desired products. Here are some key factors and considerations when designing a grocery store website:

**1. UI Design:** Keep the layout clean, organized, and easy to navigate. Use a clear and consistent colour palette that is consistent with the brand and evokes feelings of freshness and nourishment. Ensure responsive design for a seamless experience across different devices and screen sizes.

**2.Navigation:** Implement a logical menu structure to categorize products into relevant sections such as Fresh Products, Staples, Milk, etc. Use drop-down menus or expandable categories for subcategories.

**3.Search function:** Combines a powerful search bar with automatic suggestions and filters to help users quickly find specific products. - Provides advanced search options, such as sorting by price, brand, food preferences, and more.

**4.Product catalogue:** Display product image, name, price, and brief description. Use clear call-to-action buttons to add items to your cart or wish list.

**5.Product page:** Includes detailed product descriptions, nutritional information, and customer reviews. Presenting multiple product images from different angles. Make relevant product recommendations to encourage more sales.

**6.Basket and payment:** Always show the cart icon and show the number of items in it. - Provide a streamlined checkout process with minimal payment options and steps for guests Includes secure payment options and prominently displays trust badges for data security.

**7.User Accounts:** Allows users to create accounts to save shopping preferences, view order history, manage addresses and payment methods.

**8.Special offers and promotions:** Highlight current sales, discounts, and special offers on the homepage. Create a separate category for featured products or weekly promotions.

**9.Images and media of reaction:** Optimize images and videos for fast loading times and smooth browsing experience.

**10.Accessibility:** Ensure that the site is accessible to persons with disabilities, in accordance with relevant guidelines.

Remember that the design must reflect the brand identity and create an enjoyable shopping experience for the customer. You should also continuously collect user feedback and make improvements based on their needs and preferences. If you're looking for specific examples or inspiration, you can search existing grocery store websites to see how they've implemented these design principles.

# **Features**

A well-designed grocery website offers a wide range of features that enhance the user experience and streamline the shopping process. Here are some key features to consider:

**1.Registration and user account:** Allows users to create accounts to save preferences, track orders, manage shopping lists.

**2.Search and Filter:** Provides a powerful search bar with automatic suggestions and advanced filtering options to help users find products quickly.

**3.Product categories and sub-categories:** Organize products into clear and logical categories, with subcategories for easy navigation.

**4.Product catalogue:** Display product images, names, prices, and brief descriptions in an organized grid or as a list.

**5.Product details page:** Provide detailed information about each product, including description, nutritional information, ingredients, and customer reviews.

**6.Basket:** Allow users to add products to cart, view cart contents and easily adjust quantity.

**7.Wishlist:** Allows users to create and manage wish lists for products they plan to purchase in the future.

**8.Promotions and Discounts:** Highlight current promotions, discounts and special offers on the homepage.

**9.Featured products:** Highlight featured or trending products to attract users' attention.

**10.User reviews and ratings:** Allow customers to leave product reviews and ratings, helping others make informed decisions.

**11.Secure Payment:** Provides a secure and user-friendly payment process with multiple payment options.

**12.Customer's order:** Provide an option for users to make purchases without creating an account.

**13.Order tracking:** Allows users to track order status and view their order history.

**14.Delivery options:** Provide options for delivery methods, time, and shipping costs.

**15.Location-based service:** Provides options to find the nearest store, check product availability, or choose door-to-door delivery.

**16. Responsive Design:** Make sure the website is fully functional and looks good on a variety of devices including desktops, tablets, and smartphones.

**17.Personalized recommendations:** Implement algorithm to recommend products based on user's browsing and purchase history.

# **3. Using Technologies**

**About HTML:**

This is a simple sample HTML template that you can use for your shopping list web application. This template includes a form to add items to a shopping list and an item list to display shopping items. Please note that this is a basic example, and you can customize and improve it to your liking.

**About CSS:**

Creating a visually appealing and user-friendly look and feel for your grocery store website involves using CSS (Cascading Style Sheets) to style your HTML elements. Here's a simple sample CSS template that you can use as a starting point. You can customize and extend this template to fit the design aesthetic you have in mind.

**About Python Flask:**

Flask is a popular web framework for building web applications using the Python programming language. It is designed to be lightweight, flexible, and easy to use, making it a great choice for both new and experienced developers. Flask provides the necessary tools and libraries to handle various aspects of web development, including routing, templates, forms, databases, and more.

**Here are some key Flask concepts and features:**

**1. Routing:** Flask uses decorators to define routes that map URLs to functions. These functions generate the response that the user sees when visiting a particular URL.

**2. Model:** Flask supports rendering HTML templates using prototyping engines like Jinja2. Templates allow you to dynamically generate HTML code by inserting data into placeholders.

**3. Request and Response Processing:** Flask provides tools to handle incoming HTTP requests and generate appropriate HTTP responses. You can access request data, cookies, headers, etc.

**About SQLite3**

SQLite is a lightweight, standalone relational database management system (RDBMS) commonly used for local storage in applications. It's designed to be simple, fast, and reliable, making it a popular choice for embedded systems, mobile apps, and small-scale projects. Here are some key features and concepts of SQLite:

**1. Serverless and Standalone:** SQLite is serverless, meaning it doesn't require a separate database server process to run. The entire database is stored in a single file on disk.

**2. Zero setting:** No complicated installation or configuration is required to use SQLite. You can create a database file and start using it right away.

**3. ACID Compliant:** SQLite supports ACID attributes (atomicity, consistency, isolation, durability), which ensures data integrity and reliability.

**4. Data types:** SQLite supports many data types, including INTEGER, REAL, TEXT, BLOB, and NULL.

**5. SQL Support:** SQLite supports standard SQL syntax, making it compatible with other relational databases. It includes features like SELECT, SELECT, UPDATE, DELETE and JOIN operations.

**6. Index:** SQLite supports indexes to improve query performance by enabling faster data retrieval.

**7. Activities:** SQLite supports transactions, allowing you to group multiple SQL statements into a single unit of work that can be committed or rolled back.

**8. Small footprints:** The SQLite library is compact and has a small memory footprint, making it suitable for resource-constrained environments.

**9. Embeddable:** Because SQLite is standalone and does not require a separate server, it is easy to integrate into applications.

**10. Cross-platform:** SQLite is cross-platform and available on a variety of operating systems including Windows, macOS, Linux, iOS, and Android.

**11. Community support:** SQLite has a large and active community, as well as rich documentation and resources.

**My database Groceries List:**

A black screen with a black border

Description automatically generated

**Figure 1.List of groceries stored in backend Database**

**My Python Code Screen shot:**

A screen shot of a computer program

Description automatically generated

**Figure 2. Python code for grocery list application.**

**My HTML Code:**

**A screen shot of a computer

Description automatically generated**

**Figure 3. HTML code for frontend design.**

**My CSS Code:**

A screenshot of a computer

Description automatically generated

**Figure 4. CSS Code for grocery list application.**

**My website After open:**

**Add item:**

A close-up of a computer screen

Description automatically generated

**Figure 5. Adding Items in grocery list**

A screenshot of a computer

Description automatically generated

**Figure 6. Adding "Coffee" item in grocery items list**

**Check Item:**

A screenshot of a checklist

Description automatically generated

**Figure 7. Checking an item in grocery list**

A screenshot of a computer

Description automatically generated

**Figure 8.Checking an item "Apples" in grocery list**

# **4. References**

Jones, M., Campbell, B. and Mortimore, C., 2015. JSON Web Token (JWT) profile for OAuth 2.0 client authentication and authorization Grants (No. rfc7523).

[Welcome to Flask — Flask Documentation (2.3.x)](https://flask.palletsprojects.com/): This is the official documentation of Flask, where you can find a QuickStart guide, a tutorial, a user’s guide, and an API reference. It covers all the aspects of Flask, from installation and configuration to routing, templating, testing, and deployment. You can also find information about common patterns, signals, errors, logging, and extensions.

How To Use MySQL Database with Flask.2021.

https://www.youtube.com/watch?v=Ohj-CqALrwk&ab\_channel=PythonSimplified

**Total website reference YouTube. https://www.youtube.com/watch?v=v3CSQkPJtAc&ab\_channel=PythonSimplified**

**5. GITHUB Link:**

**GitHub link: https://github.com/vajjamahendra/python-ca2-1**