International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2024.0081
e ISSN: 2584-2854
Volume: 02
Issue: 04 April 2024
Page No: 581-585

Market Maestro: A Smart E-commerce Application

Kalaivani B[1], Bharathi B[2], Kasu Manaswi[3], Kantabathina Tejaswini[4]
[1,3,4]UG - Computer Science and Engineering, R.M.K. Engineering College, Kavaraipettai, Thiruvallur, India.
[2]Assistant Professor, Computer Science and Engineering, R.M.K. Engineering College, Kavaraipettai, Thiruvallur, India.
Email: *kala20204.cs@rmkec.ac.in[1], bbi.cse@rmkec.ac.in[2], kasu20242.cs@rmkec.ac.in[3], kantle211402.cs@rmkec.ac.in[4]*

Abstract

In an era marked by digital transformation, "Market Maestro" emerges as a pioneering e-commerce application, poised to redefine the very essence of shopping. The application transcends the commonplace, presenting a sophisticated shopping experience that seamlessly melds convenience, sustainability, and innovation. At the heart of the application lies an impeccable user interface, meticulously crafted to provide discerning shoppers with a comprehensive array of products, ranging from fresh produce to everyday essentials. The application is developed using Flask and SQLite which help in providing a better server-side logic. What truly distinguishes "Market Maestro" is its eco-conscious packaging options and strategic partnerships with local farmers and suppliers which embarks on a sustainable journey that harmonizes quality with environmental responsibility. The proposed application further prioritizes security, offering a robust and secure payment gateway rendering a seamless checkout and contactless endeavor. Real-time order tracking provides transparency ensuring customers remain informed about their delivery status. Furthermore, the application also encloses a recommendation feature that provides the users with the possible recipes that can be made from the list of ingredients as specified by the user. "Market Maestro" allows users into a new era of shopping, marked by intelligent design, environmental stewardship, and unwavering customer-centricity.

Keywords—*Online, Convenience, Sustainability, Innovation, Flask, SQLite, Eco-Conscious, Transparency, Customer-Centricity*

1. Introduction

In an era marked by digital transformation, "Market Maestro" emerges as a pioneering e-commerce application, poised to redefine the very essence of shopping. The application transcends the commonplace, presenting a sophisticated shopping experience that seamlessly melds convenience, sustainability, and innovation. At the heart of the application lies an impeccable user interface, meticulously crafted to provide discerning shoppers with a comprehensive array of products, ranging from fresh produce to everyday essentials. The application is developed using Flask and SQLite which help to provide better server-side logic. What truly distinguishes "Market Maestro" is its eco-conscious packaging options and strategic partnerships with local farmers and suppliers which proposed application further prioritizes security, offering a robust and secure payment gateway rendering a seamless checkout and contactless endeavour. Real-time order tracking provides transparency ensuring customers remain informed about their delivery status. Furthermore, the application also encloses a recommendation feature that provides the users with the possible recipes that can be made from the list of ingredients as specified by the user. "Market Maestro" allows users in a new era of shopping, marked by intelligent design, environmental stewardship, and unwavering customer-centricity [1].

2. Experimental Methods or Methodology

The application is developed using Flask and SQLite which helps in defining a strong server-side logic.

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2024.0081
e ISSN: 2584-2854
Volume: 02
Issue: 04 April 2024
Page No: 581-585

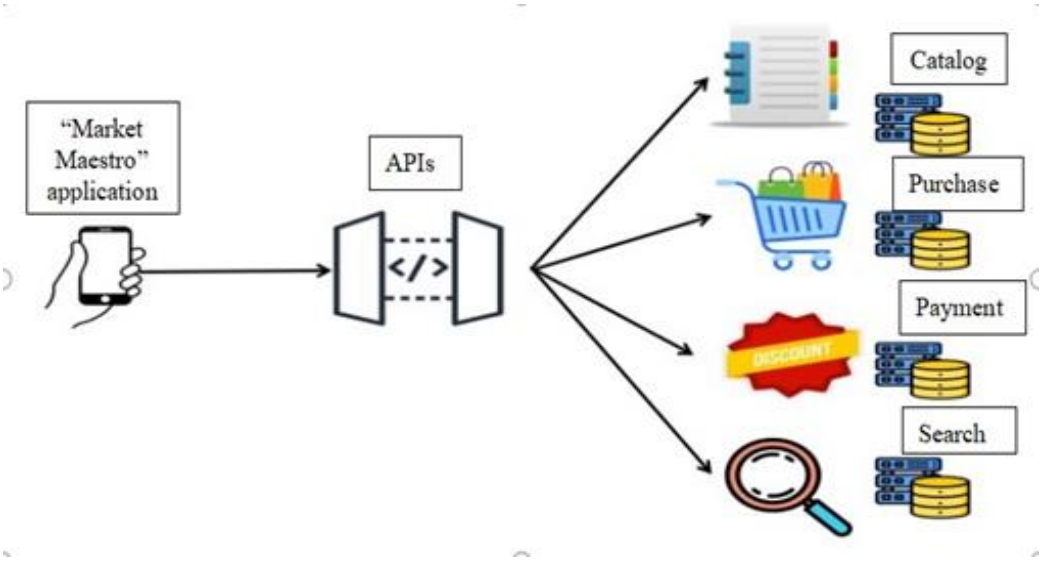

Figure 1 System Architecture

The reason for the implementation of the application using Flask is because of its flexibility, extensibility, and community support. Flask is also identified as a light weighted and dynamic framework which facilitates easy integration of the components as and how the user requires it to be SQLite, on the other hand, is best known for its simplicity, ease of integration, ACID (Atomicity, Consistency, Isolation, Durability) compliance and cross-platform compatibility [2]. It provides a suitable medium for prototyping smaller to medium-scale applications. The application structure consists of routes, models, forms, and unit snippets of code. The routes ensure providing dynamic routing to the different pages in the application upon an action say, click or hover through a button. The models consist of the database details of the corresponding data with all the attribute specifications, primary keys, foreign keys, relationships, etc. The forms are a classic feature of Flask that is directly mapped with the models so that the values entered in the form will be directly stored in the database after validating carefully. The suggested work makes use of Flask, which allows users to request on-demand database access to the product, and SQLite to store all user data. For each of the routing commands, the corresponding GET and POST requests are mentioned to ensure the proper functioning of the application [3]. There can be as many numbers of routes, models, and forms. However, the control lies in the main file that runs the entire application on the whole. It generates a hyperlink that can be accessed using any of the compatible browsers. The proposed system has a very simple system architecture that is depicted in (Figure 1). The following are the different modules available in the application:

- Multiple login
- Category Management
- Product Management
- Secure Payment
- Search options
- Effective query and response

2.1 Multiple logins

Separate login for customer, admin, and manager is created incorporating Role Based Access Control (RBAC). The individual login has got different function access along with proper validations. Utilizing the Flask-login, Flask-WTF, and Flask-bcrypt packages, the suggested work integrates Role Based Access Control (RBAC) to provide the multiple login capability which is explained in (Figure 2). The user login is exclusively for the customers where they can view, search, and purchase products as per their requirements. They can also interact with the bot and get recommendations. The manager login is for use by

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2024.0081
e ISSN: 2584-2854
Volume: 02
Issue: 04 April 2024
Page No: 581-585

the store manager where categories and products can be added, updated, or deleted only that the tasks performed by the manager need to be approved by the admin first. The admin has control over all the operations of the manager and also performs independently [4]. Since it is the admin who is responsible for approving or rejecting the requests, no other user has access to modify the contents. Also, the entries in the database are encrypted ensuring that it will be difficult to obtain the original plain text of the password. Moreover, each user login session or other activities are accompanied by the use of session tokens which remain valid only for a stipulated period after which the user session expires. This way, the user has to log in using their unique credentials to prove authenticity.

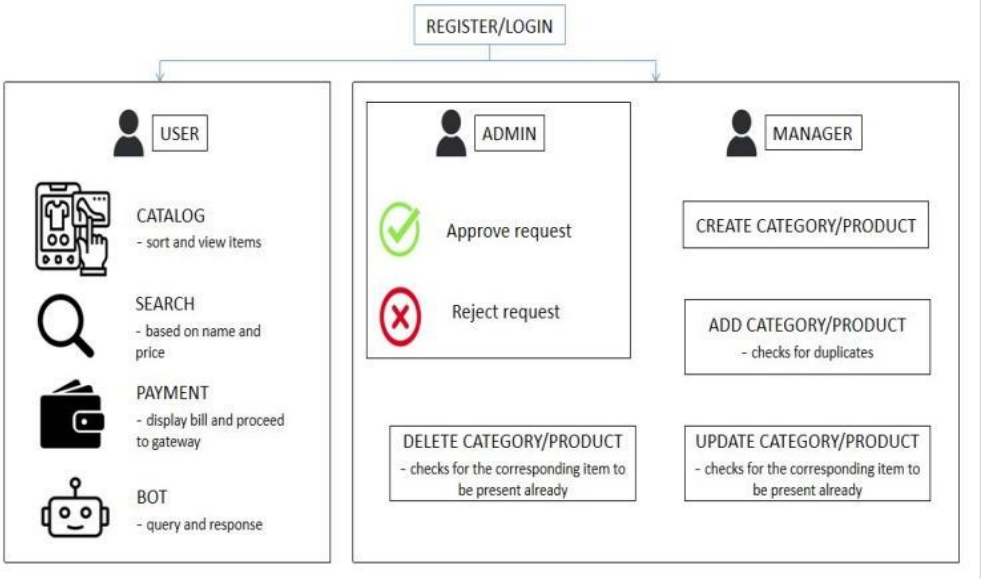

Figure 2 Role-Based Access Control

2.2 Category Management

Different products are grouped under similar categories. There may be several categories available under a shopping application. The categories can be created, edited, and deleted as per the requirements. This category management can be done by both the admin and the manager. By utilizing SQLite features, when a manager makes any changes to any category, a request will be sent to the admin that will be done on being granted by the admin, else the request will be rejected. Categories are created in such a way that no duplicate entries can be made. The newly created category is compared with the pre-historic entries in the dataset. The attribute that uniquely identifies the item (say, the category name) is compared with the currently entered value. If a match is identified, then an alert message flash indicating duplication attempts. However, if a match is not found, it is identified as a new category and hence a new entry is added to the database [5].

2.3 Product Management

Each category has a variety of different products with basic details like quantity, cost, and stock which can also be created, edited, and deleted like the categories. Like the categories, the products can be created, edited, and deleted as per the requirements. They also do not support duplicate entries like categories. Like category management, product management can also be done by both admin and manager. The only difference is, that when a manager makes any changes to any product, a request will be sent to the admin that will be done on being granted by the admin, or else the request will be rejected [6].

2.4 Secure Payment

Once things are finalized to be added to the cart, the amount of money for all the products is calculated

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2024.0081
e ISSN: 2584-2854
Volume: 02
Issue: 04 April 2024
Page No: 581-585

and displayed for further payment procedures. The values of the added products can still be modified even after the receipt is generated. When the quantity is modified to the products that are already added to the cart the changes are reflected in the cost and stock features also ensuring smooth inventory management [7].

2.5 Search options

The user can make use of the search option to immediately land on the corresponding product that they are looking for. Rather than going through the entire product list for a random product that may or may not be available currently, this option saves time. The search option can support a variety of different elemental searches like searching using the category name, the product name, the cost value, the stock value, the starting letter of the product, etc [8].

2.6 Effective query and response

At any point in time, if the user finds it difficult to navigate through the different available features of the application, queries can be asked using the chatbot which provides corresponding responses. The chatbot is fed with a predefined set of inputs and the relevant outputs which are stored in the IBM cloud DB2 and PostgreSQL databases. The set of questions and answers are connected into a dialog that establishes a meaningful interaction like speaking to a human being [9].

3. Results and Discussion

"Market Maestro" allows the users to enjoy the features of a typical e-commerce application bordered with major backend processing like RBAC and management of categories and products smartly and efficiently. It also aids the user by providing recommendations and virtual try-on features that upgrade the online shopping process to the next level. The application is well developed with advanced routing and navigation that provides a friendly user interface to the end user. Online shopping has grown incredibly over the past few years, and this growth does not appear to be slowing down anytime soon. Convenience and affordable prices are two reasons why people shop online. In addition to that, online businesses are mapped with the models so that the values entered in the form will be directly stored in the database [10].

Conclusion

For further improvements, a Virtual try-on option can be enabled which is a smart way of trying on clothing and fashion wear, so that the user might get a better understanding of whether to purchase the product or not. Like the "Recommendations" feature for eatables, this Virtual try-on option will be an initiative for developing a smart application. Furthermore, day-to-day updates on current trends in the market can be visualized on the news/blog page. The introduction of audio options in the chatbot (voice bot) and search options will be an added improvement to the application making it more user-friendly.

References

[1]. A. Momenikorbekandi and M. F. Abbod, "A Novel Metaheuristic Hybrid Parthenogenetic Algorithm for Job Shop Scheduling Problems: Applying an Optimization Model," in IEEE Access, vol. 11, pp. 56027-56045, 2023, doi: 10.1109/ACCESS.2023.3278372.

[2]. Zhao F, He X, Wang L. a Two-Stage Cooperative Evolutionary Algorithm with Problem-Specific Knowledge for Energy-Efficient Scheduling of No-Wait Flow-Shop Problem. IEEE Trans Cybern. 2021 Nov;51(11):5291-5303. doi: 10.1109/TCYB.2020.3025662. Epub 2021 Nov 9. PMID: 33095728.

[3]. B. Sainz-De-Abajo, J. M. García-Alonso, J. J. Berrocal-Olmeda, S. Laso-Mangas and I. De La Torre-Díez, "FoodScan: Food Monitoring App by Scanning the Groceries Receipts," in IEEE Access, vol. 8, pp. 227915-227924, 2020, doi: 10.1109/ACCESS.2020.3046031.

[4]. S. Wu and L. Liu, "Green Hybrid Flow Shop Scheduling Problem Considering Sequence Dependent Setup Times and Transportation Times," in IEEE Access, vol. 11, pp. 39726-39737, 2023, doi: 10.1109/ACCESS.2023.3269293.

[5]. Y. Pan, K. Gao, Z. Li, and N. Wu, "Improved Meta-Heuristics for Solving Distributed Lot-Streaming Permutation Flow Shop Scheduling Problems," in IEEE Transactions on Automation Science and Engineering, vol. 20,

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2024.0081
e ISSN: 2584-2854
Volume: 02
Issue: 04 April 2024
Page No: 581-585

no. 1, pp. 361-371, Jan. 2023, doi: 10.1109/TASE.2022.3151648.

[6]. Zhong, Hongyang, Jianjun Liu, Qing-xin Chen, Ning Mao and Xiaojia Yang. “Performance Assessment of Dynamic Flexible Assembly Job Shop Control Methods.” IEEE Access 8 (2020): 226042-226058.

[7]. T. Zhang, X. Guo, and G. Ji, "Permutation Flow Shop Scheduling Optimization Method Based on Cooperative Games," in IEEE Access, vol. 11, pp. 47377-47389, 2023, doi: 10.1109/ACCESS.2023.3275533.

[8]. S. Luo, L. Zhang, and Y. Fan, "Real-Time Scheduling for Dynamic Partial-No-Wait Multiobjective Flexible Job Shop by Deep Reinforcement Learning," in IEEE Transactions on Automation Science and Engineering, vol. 19, no. 4, pp. 3020-3038, Oct. 2022, doi: 10.1109/TASE.2021.3104716.

[9]. Rao, Sharath, and Lily Zhang. “The Algorithms That Make Instacart Roll: How Machine Learning and Other Tech Tools Guide Your Groceries from Store to Doorstep.” IEEE Spectrum 58 (2021): 36-42.

[10]. C. -H. Chen and W. Zhai, "The Effects of Information Layout, Display Mode, and Gender Difference on the User Interface Design of Mobile Shopping Applications," in IEEE Access, vol. 11, pp. 47024-47039, 2023, doi: 10.1109/ACCESS.2023.3274575.