

FRESH CART (Your One-Stop Online Grocery Shop)

V.Vivek, M.Vivek, M.Vyshnavi, Wajeeha Unnisa, M.Yamini, J.Yashwanth

Guide: Prof. Bhavani

Bachelor Of Technology

Computer Science and Engineering (Ai & MI)

Malla Reddy University

1. INTRODUCTION

1.1 ABSTRACT

Our project is to design a website like Big Basket. In this project, we are using python+ (Django + Tkinter). Django is a popular Python-based web framework that is designed for building complex web applications quickly and easily. Some of the features of Django are:User- friendly Mapping (ORM), Admin interface, Template engine, Security features, and multi- language support. Tkinter is a command-line tool for PHP that facilitates package installation and management. Some of the features of Tkinter include Interactive REPL (Read-Evaluate- Print Loop) console that allows you to test and debug your code snippets and applications in real time. Automatic code reloading and debugging for faster development and testing. Branding, Convenience, Cost-effectiveness, and Seamless User friendly are the advantages of our website. Overall, a single website for groceries can simplify the shopping experience for customers while also providing cost and inventory management benefits for the grocery store. Online shopping has become increasingly popular in recent years due to the convenience and flexibility it offers. It allows consumers to browse a wider selection of products, compare prices, and purchase items from the comfort of their own homes or on the go using a mobile device. Online shopping also provides consumers with access to reviews and ratings from other buyers, which can help inform purchasing decisions. Overall, online shopping has transformed the way we shop, making it easier and more accessible for people around the world. In this project, we will develop a web application that allows users to purchase groceries online from multiple web sources. Online shopping is a form of ecommerce that allows consumers to buy goods and services directly from a seller over the internet, without the need for physical interaction.

1.2 LIMITATIONS OF THE PROJECT

There are several limitations of grocery stores that can impact their customers and operations, including:

1. **Limited Selection**: Grocery stores have a limited selection of products compared to specialty stores, which may have a narrower focus on particular types of food or brands.
2. **Inconvenient Location**: Grocery stores may be located far away from some customers, making it difficult or costly to get there.
3. **Limited Hours**: Many grocery stores have limited hours, which can be inconvenient for people who work long hours or have other commitments.
4. **Crowded Aisles**: Grocery stores can be crowded, especially during peak times, which can make it difficult to move around and find the products you need.
5. **Out-of-Stock Items**: Grocery stores may run out of popular items, which can be frustrating for customers who have made a special trip to purchase them.
6. **Checkout Lines**: Long checkout lines can be frustrating, especially during peak hours or when there are not enough cashiers available.
7. **Pricing**: Some customers may find grocery store prices to be higher than those of discount stores or online retailers.

Overall, grocery stores are an essential part of our daily lives, but they do have limitations that can impact the shopping experience.

2.ANALYSIS

2.1 SOFTWARE REQUIREMENT SPECIFICATION

2.1.1 SOFTWARE REQUIREMENTS

- Python
- Django:

Django is a widely used framework with python for web-application and it is high-level pythonweb framework that enables rapid development of secure and maintainable websites. Built by Experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel.

It is such a powerful framework that companies like Instagram, the Washington Post, and Dropbox have all used Django in their applications. Django is used for full-stack web development. This is because Django lets you serve up HTML and process the back-end of a website.

- TKINTER:

Tkinter is the python interface to the TK GUI toolkit that is shipped with python.Flask is a web application framework written in python.You can develop a web application with the help of it. Tkinter is the inbuilt python module that is used to create GUI applications. It is one of the most commonly used modules for creating GUI applications in Python as it is simple and easy to work with. You don't need to worry about the installation of the Tkinter module separately as it comes with Python already.

2.1.2 HARDWARE REQUIREMENTS

- Operating system: Microsoft Windows 7 or newer
- RAM: 2GB RAM (4 GB recommended)

- Hard disk space: 64 GB (minimum)
- Network:10/100 Network Interface Card.

2.2 EXISTING SYSTEM

The existing system for purchasing groceries involves physically going to a supermarket or store and selecting the items you need. This can be time-consuming and inconvenient, especially for people who have busy schedules. Moreover, not all stores have all the items one might need, and the availability of the items can vary from store to store. User interaction is less and no services are offered.

2.3 PROPOSED SYSTEM

The proposed system is an online grocery shopping platform that allows users to purchase groceries from multiple web sources. The system provides a wide range of items to choose from, and users can filter and sort the items based on their preferences. It also provides a seamless and hassle-free checkout process and offers various payment options. In this system, the best feature is that whenever a user purchases some vegetables or fruits the dishes that are possible to make will be recommended, it is up to the user to use it or not. If the customer gets satisfied they will be ranking our recommendations, which will be helping the other users. Notonly vegetable but also fruit recipes will be recommended.

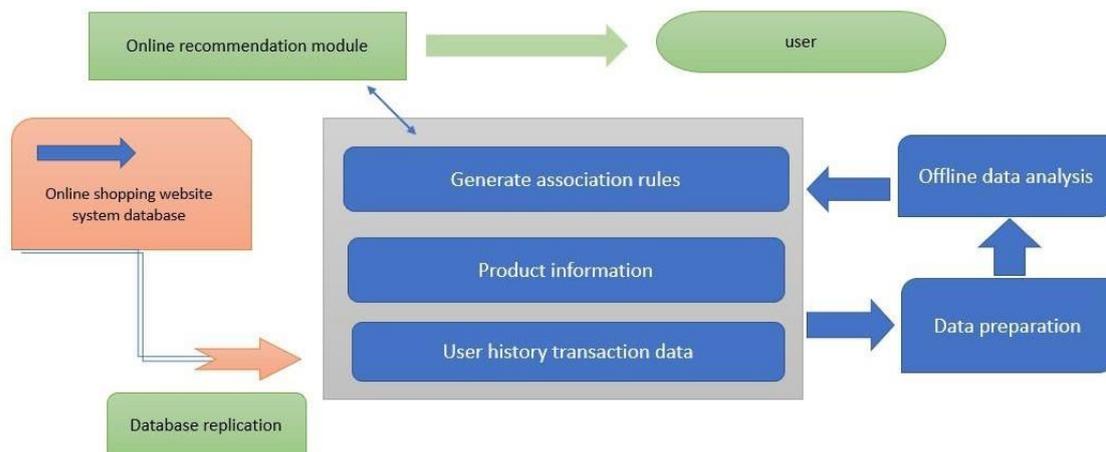
2.4 MODULES

The project can be divided into several modules that perform different functions. Here are some of the modules for the proposed system:

- 1. User Authentication:** This module handles the login and signup process for users. Product Catalog: This module provides a list of all the available products for users to purchase.
- 2. Cart Management:** This module allows users to add or remove items from their cart and displays the total amount to be paid.
- 3. Payment Gateway Integration:** This module integrates with third-party payment gateways to facilitate payment for purchased items.

4. Shipping Management: This module handles the shipping of the items and provides tracking information to users.

2.5 ARCHITECTURE

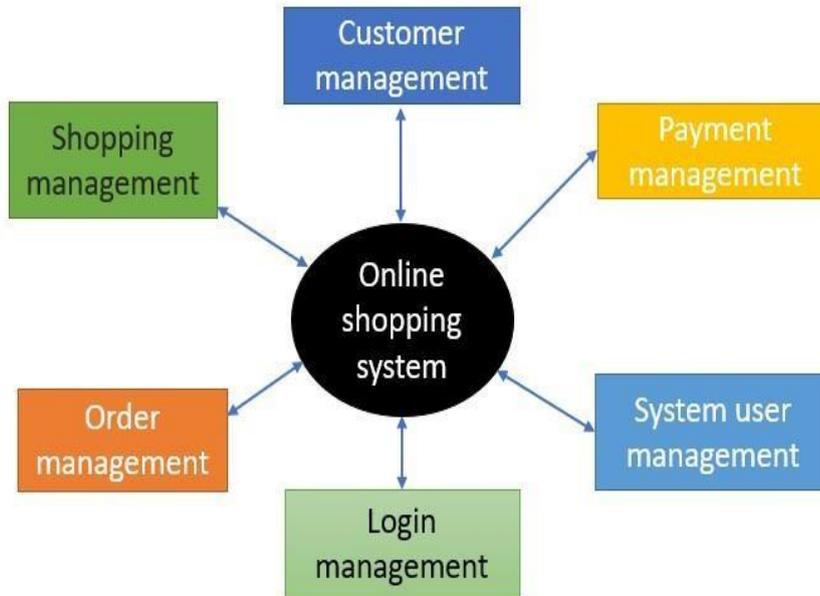


3. DESIGN

3.1 DFD/ER/UML DIAGRAM

A. DFD DIAGRAM

LEVEL 0



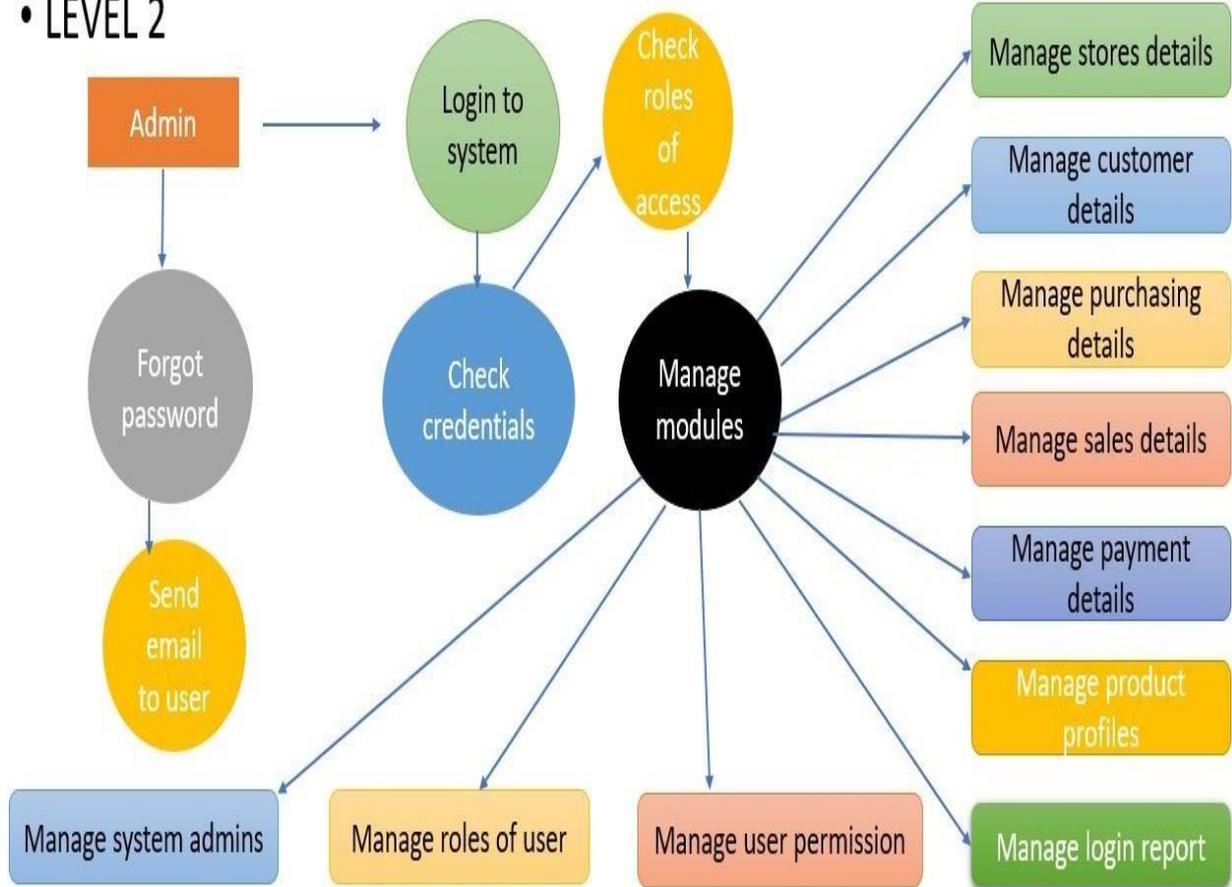
Zero level DFD – online shopping system

• LEVEL 1

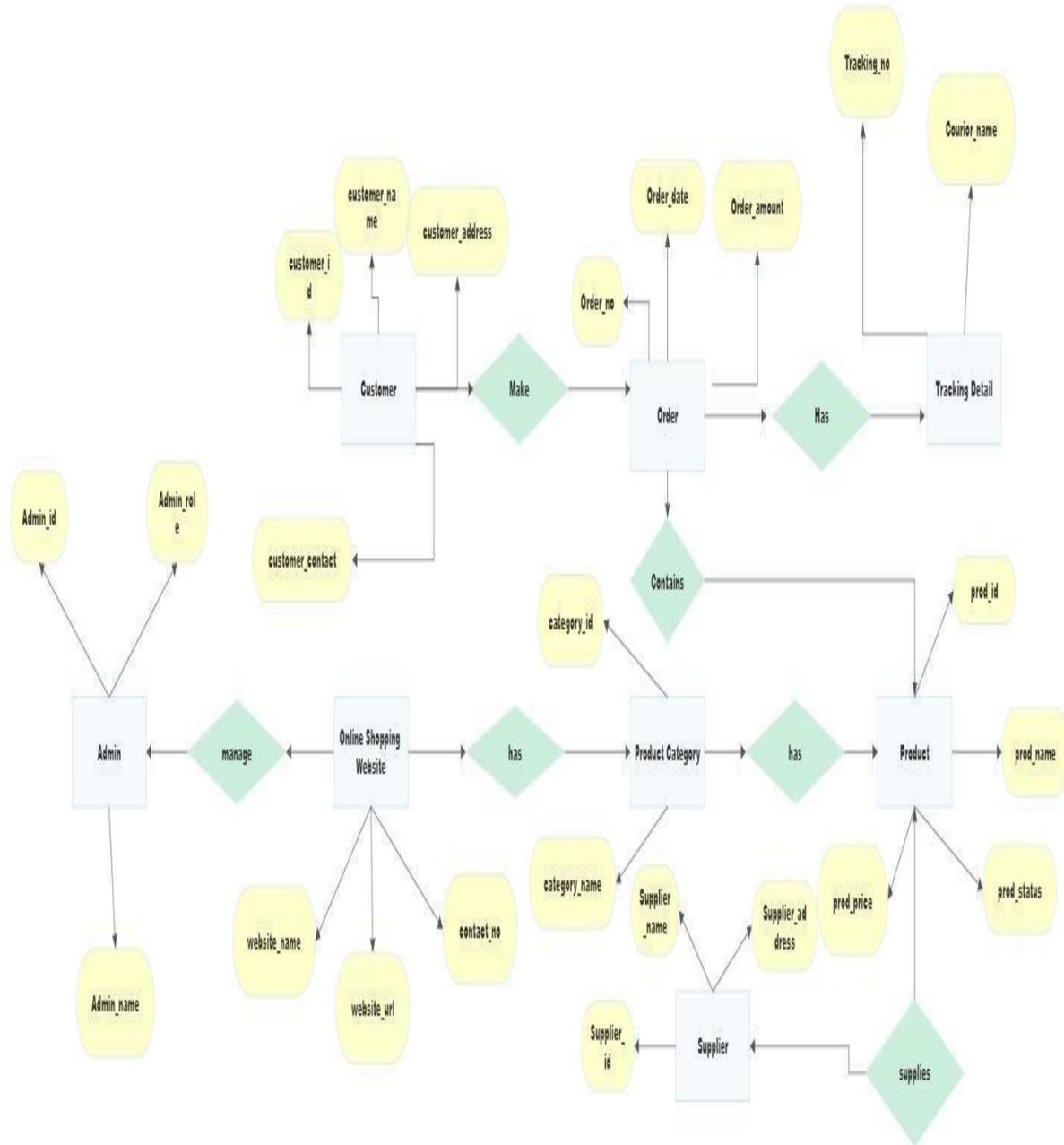


First level DFD – online shopping system

• LEVEL 2



B . ER DIAGRAM



4. DEPLOYMENT AND RESULTS

4.1 Source code:

Separates the different sections:

```
#!/usr/bin/env pythonimport os
import sys

if __name__ == '__main__': os.environ.setdefault('DJANGO_SETTINGS_MODULE',
'djecommerce.settings.development')try:
    from django.core.management import execute_from_command_lineexcept ImportError as exc:
    raise ImportError(
        "Couldn't import Django. Are you sure it's installed and "
        "available on your PYTHONPATH environment variable? Did you ""forget to activate a virtual
        environment?"
    ) from exc execute_from_command_line(sys.argv)
```

URL's of all the buttons used in the app:

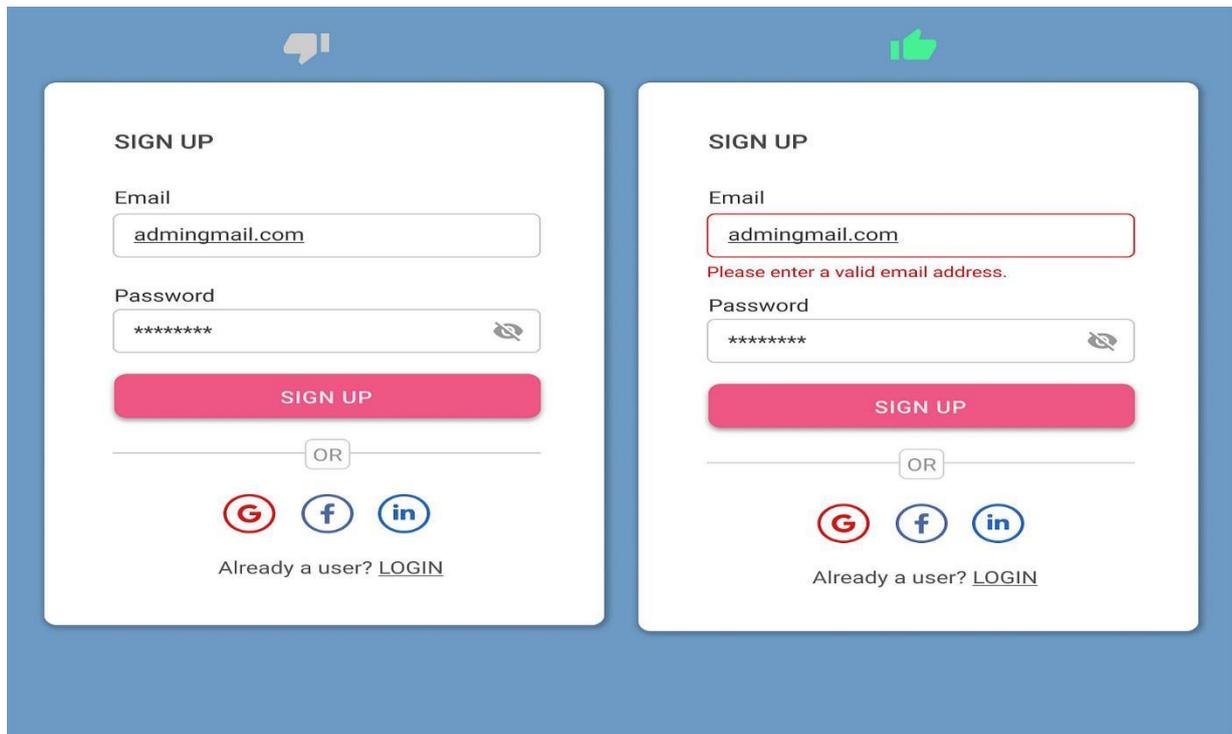
```
from django.conf import settings
from django.conf.urls.static import staticfrom django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls), path('accounts/', include('allauth.urls')), path("", include('core.urls',
    namespace='core'))
```

]

```
if settings.DEBUG: import debug_toolbar
    urlpatterns += [path('_debug_/', include(debug_toolbar.urls))]urlpatterns +=
    static(settings.MEDIA_URL,
            document_root=settings.MEDIA_ROOT)urlpatterns += static(settings.STATIC_URL,
            document_root=settings.STATIC_ROOT)
```

4.2 Final Result:





5. CONCLUSION

5.1 Project Conclusion

The Fresh Cart project is a great example of how Python can be used to create a functional and user-friendly grocery store application. Throughout the project, various Python libraries and tools were utilized to implement the different features of the application, such as the GUI, database management, and data analysis. Additionally, the integration of data visualization tools allowed for the generation of useful insights that can help optimize the store's inventory management and sales.

By using Python's object-oriented programming paradigm, the project was able to efficiently organize and manipulate data structures such as product catalogs and shopping carts.

Overall, the Fresh Cart project serves as a great demonstration of how Python can be used to create practical and valuable applications that can benefit businesses and consumers alike. By leveraging the power of Python, developers can create innovative solutions that can improve the efficiency, productivity, and profitability of various industries.

5.2 Future Enhancement :

1. **Search Functionality:** Adding a search bar to the product catalog would make it easier for customers to find specific products they are looking for.
2. **Recommendation Engine:** The use of a recommendation engine would allow the store to suggest additional products to customers based on their purchase history, increasing the chances of repeat business.
3. **User Reviews:** Adding a user review section for products would allow customers to leave feedback on the products they have purchased, improving the store's credibility and helping customers make informed purchasing decisions.
4. **Payment integration:** To make the shopping experience more seamless for customers, you could integrate payment methods such as credit cards, PayPal, or other online payment gateways. This could reduce the need for customers to physically visit the store to make a purchase.