GroceryGo, Ordering groceries made easy!

Chandana(10539269)
Department of Information systems with computing
Dublin Business School

CONTENTS

LIST OF FIGURES

GroceryGo, Ordering groceries made easy!

***Abstract*—GroceryGo is an online grocery shopping website that allows its customers to order and schedule delivery of daily essentials through a subscription plan. To store user-related data and order information, MySQL as the back-end database, Python Flask as an API computing interface between a web browser and Database, and HTML, CSS, Bootstrap, Jquery, Jinja to develop a responsive web application. The web application hosted on an AWS cloud service called EC2. The platform enables users to see the list of products based on their categories. User will be able to choose the product and add to cart. User will be able to check out the cart which takes him to address form where google API used to display the address based on the city entered and confirm the order. Admin access is provided to a specific user to post their offerings on the site, delete an item, edit details for the items listed, also to analyse user and order statistics.**

I. List of Abbreviations

API	Application Programming Interface
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
EC2	Amazon Elastic Compute Cloud
AWS	Amazon Web Services
SQL	Structured Query Language
JSON	JavaScript Object Notation
AJAX	Asynchronous JavaScript
UI	User Interface

II. Introduction

The aim is to develop an information system detailing the requirements including user requirements, data requirements, operations on the data and validations; using a database for organising the data, a responsive web application for the front end and API computing interface between web application and Database for comprehensive user interactions.

The application developed should be in a well-defined and structured sequence; through the requirement engineering process where the initial step is to go through a feasibility study to verify GroceryGo is operationally and technically feasible. The next step is to gather the requirements of the users, which are system requirements. The concept of the observer design pattern is utilised as there is many to many dependencies between the objects. The website is developed and tested against system requirements to verify that user requirements.

III. Background

Here are some of the functional system requirements for GroceryGo:

- Customer shall be able to log in/sign-up.
- Customer shall be able to search the list of products available.
- System must allow customers to select the product, add to cart, and to order cart items.
- Admin shall have the ability to add, update or delete product details.
- Admin shall be able to view all the customer details and order details.

Interaction between the GroceryGo system and the users are analysed using the Use case diagram as shown in the Fig 1

The figure shows how different participants like admin and the user shares the same functionality like login, logout. User actor may be a registered user or new user. A new user will have to register to view and order items. A user can select, filter and add items to the cart. Admin has the authorised functionality to add, update and delete product details. Also, analyse users and ordered items.

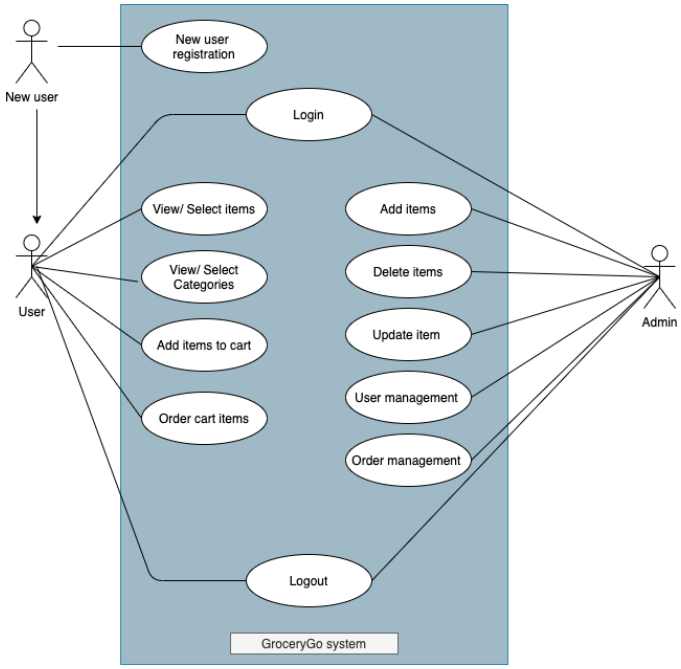

Fig. 1. Use case diagram

IV. Technologies Implemented

In order to develop the product is divided into three parts database as backend, python API interface and responsive web app as a front end.

A. FRONT END

A website is designed and developed using the below technologies for responsive and interactive UI:

1) HTML5: Hypertext markup language assists in designing and transforming the content of the webpage. The latest version HTML5 uses semantic tags to maintain the structure of the page. It also supports mobile-applications cross-platform keeping low-powered devices in mind.

2) CSS: A cascading style sheet is a language for interpreting the appearance of the website; it comprises many things which consist of colours, layout, and fonts. Bootstrap, a CSS framework, adds a deliberate look to the website and makes the website more responsiveness by designing templates for typography, forms, buttons, tables and many other in a faster and easier way.

3) JQuery: Javascript control the functioning of the webpage by making the user-friendly website interactions. Jquery is widely used, developer's favourite lightweight Javascript library performs more things in a few lines of code. JSON is a key-value pair text used for storing and exchanging data between the server and the client that reduces complications in parsing and translations.

4) Jinja: Jinja2 is a text-based templating language which is safe and secure, used by designers and developers that stick to Python's principle and adds functionality that is useful for templating environments(Jinja, 2020).

5) AJAX: JQuery Ajax, an asynchronous call is used to exchange data with the server for updating a webpage without reloading the whole page(flask, Agrahari and Killian, 2020)(AJAX, 2020).

B. API

An API, is an interface between the components of the software product. Below are the technologies used to achieve the computing interface between the webpage and the database.

1) Python Flask: Flask is a microframework that helps to decide what templating engine to use by supporting extensions to add functionality to the application. Flask provides many extensions like database integration, validations, authentication and many more(Foreword — Flask Documentation (1.1.x), 2020). A Blueprint from flask helps to organize a group of code by registering through them. The blueprints are then registered with the application in the main module(Blueprints, 2020). Many modules and extensions are used like to maintain the inactive session, timedelta module has been used. Flask session extension is used to support server-side session of the application. Redirect is used which returns to location header as the URL for the function.Render template is used to return the content of the URL provided. The flashing system is used using flash at the end of the request to record a feedback message to the user. Flask Jsonify is used to wrap dumps into a response object with the application. It also converts multiple arguments as an array and multiple keyword arguments into a dictionary.

C. DATABASE

Using Relational Database management System(RDBMS), data is repressented in a structured and language consistent way.

1) MySQL: The primary and advance concepts like create, sort, update, insert, delete, alter are used in order to design a data structure as per the GroceryGo requirement; parameterized queries are written in SQL and accessed using python.

2) SSH Sessionhandler: Flask-session is an extension for a flask that supports server-side session. A sshtunnel is used for port forwarding for a MySQL remote server port. API allows starting and stopping the tunnel using wit the context(tunnel, 2020). Also, we used the SSH Sessionhandler to connect to AWS EC2 SQL server.

D. Cloud Services

Cloud Computing provides various network resources like servers and storage that can be handled according to the requirement remotely. Amazon web services is one of the platform that provides these services.

1) AWS EC2: Amazon EC2 network service provides a secure compute module that allows complete control of the resources and lets us run servers on Amazon's proven computing environment. The website has been hosted on AWS EC2 instance using the assigned AWS domain.

V. Website Features

Over here, the implementation of the GroceryGo is to allow users to order daily groceries online(NS, 2020). The design of the page is as follows when the user hits the webpage using the web address ec2-54-72-85-207.eu-west-1.compute.amazonaws.com; a home page is loaded which gives a brief idea about the webpage(Fig 2). On click of Login/Signup(Fig 3), user is taken to a login page where he has the option to register if he does not have an account(Fig 4), and also user can update the password(Fig 5) using his username. Once the user is logged in a session is created for the user. Now on click of shop(Fig 6), a complete list of products with category filter(Fig 7) is displayed which filters items according to its categories. User can add the products to the cart, and cart items are displayed on click of the cart(Fig 8), where there is an option to add more number of items or delete items from the cart. On click of order button, address page(Fig 9) is displayed where the customer can enter the city and search for the address that takes to confirmation page(Fig 10).

On the contrary, an authorized privilege is given to an admin(Fig 12) where he has the option to add new products to the shop(Fig 11), update or delete the existing items(Fig 13).Also, he can analyse user details(Fig 14) and order statistics(Fig 15).

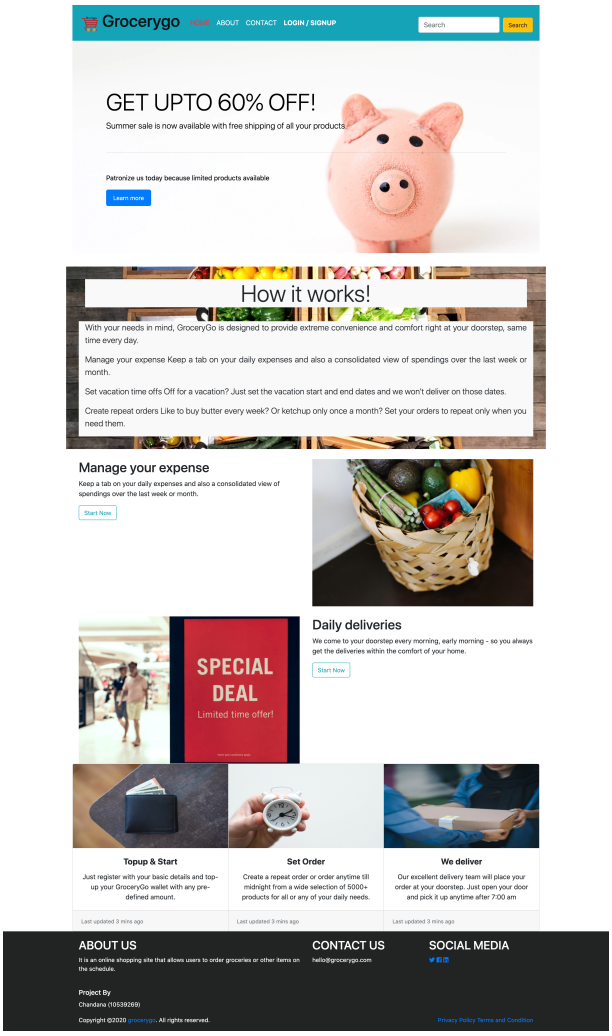

Fig. 2. Home page

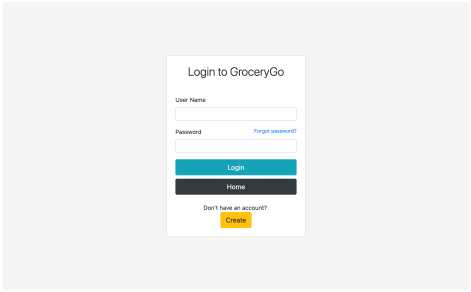

Fig. 3. Login

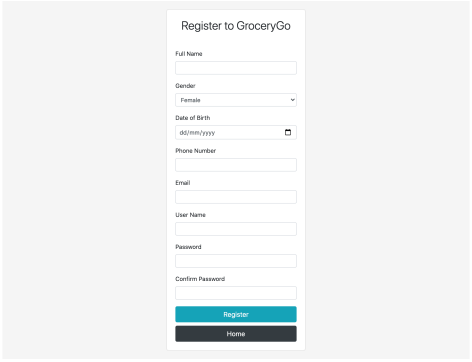

Fig. 4. Create user

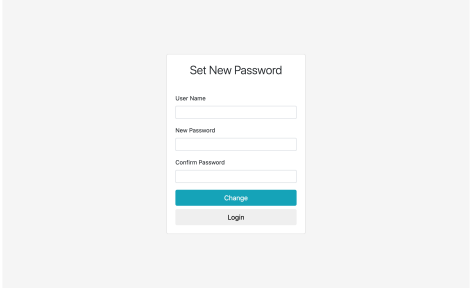

Fig. 5. Forgot password

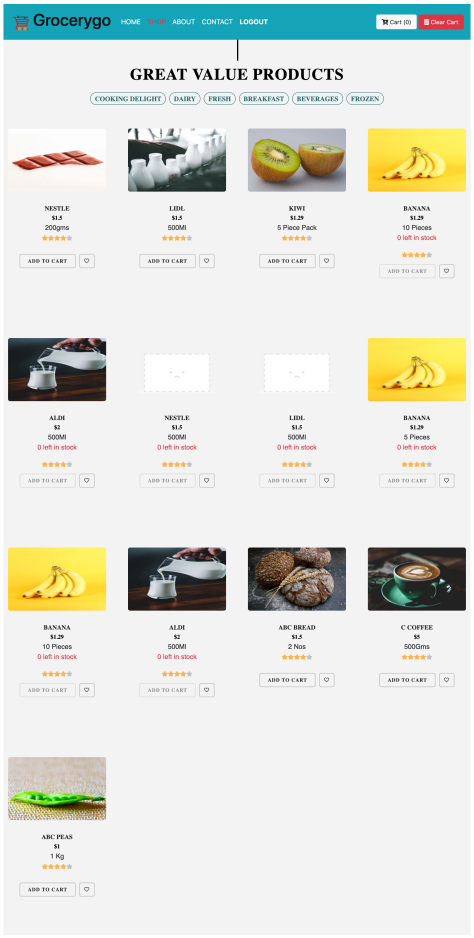

Fig. 6. User shop

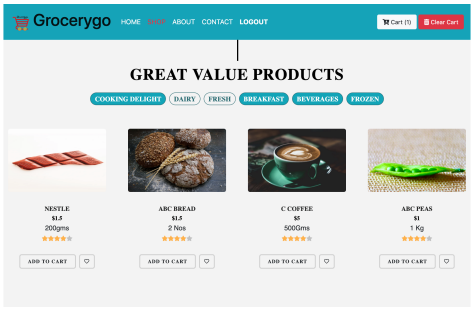

Fig. 7. Filter categories

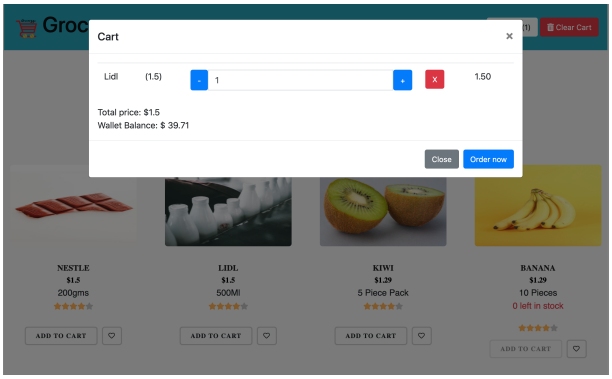

Fig. 8. Cart details

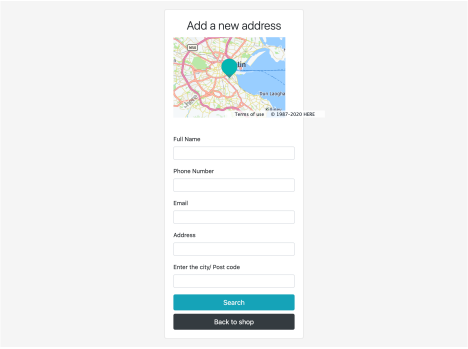

Fig. 9. Search address

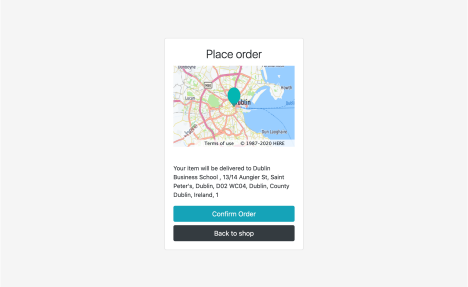

Fig. 10. Confirm order

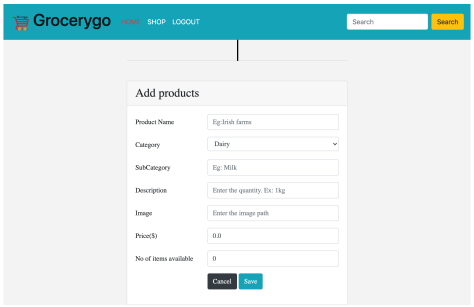

Fig. 11. Add item

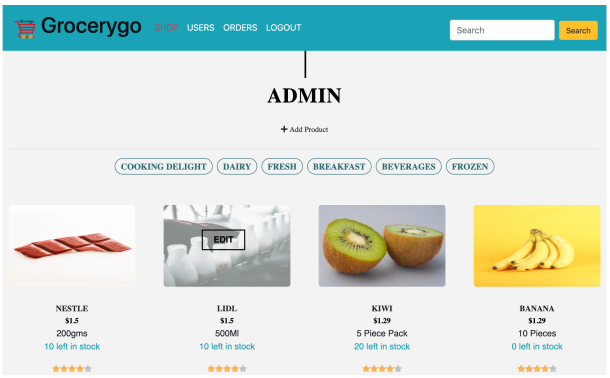

Fig. 12. Admin home

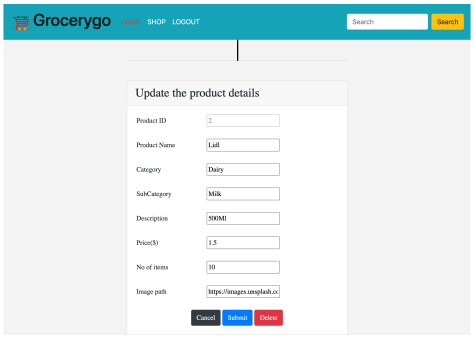

Fig. 13. Edit item

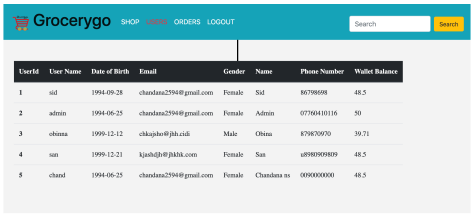

Fig. 14. User statistics

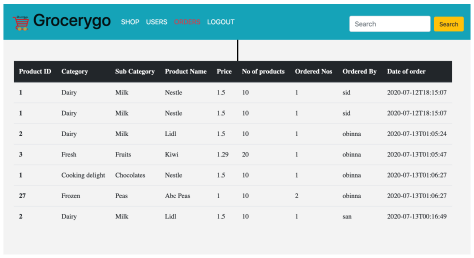

Fig. 15. Order statistics

VI. Project Management

Manual testing is done to test the complete functionalities of the user requirement for GroceryGo system. GitHub(Go, 2020) management tool is used to track, and update the work in one place so projects stay transparent. Git version control(NS, 2020) is used to track and manage source periodically.

VII. Scope of improvement

There is a broad scope of improvement for the website in the more extended run. Third-party payment services could be used to take up the payment from the customers to add money to the wallet and send notifications. Providing the option for the users to view their profiles and edit their details. An option for the user to see order history and repeat orders. Uploading a image from devices by the user instead of link.

VIII. Conclusions

To conclude, this project helped to throughput the knowledge of front end technologies, python and database, cloud services to make a web application for a GroceryGo information system that helps the customer to order an item online over the internet.

References

[1] AJAX, J., 2020. Jquery.Ajax() — Jquery API Documentation. [online] Api.jquery.com. Available at: https://api.jquery.com/jquery.ajax/ [Accessed 9 July 2020].

[2] API, F., 2020. API — Flask Documentation (1.1.X). [online] Flask.palletsprojects.com. Available at: https://flask.palletsprojects.com/en/1.1.x/api/ [Accessed 9 July 2020].

[3] Barry, P., n.d. Head First Python. 2nd ed. Sebastopol, CA O Reilly Media 2017, pp.47-94, 281-305,363-460,.

[4] Blueprints, F., 2020. Modular Applications With Blueprints — Flask Documentation (1.1.X). [online] Flask.palletsprojects.com. Available at: https://flask.palletsprojects.com/en/1.1.x/blueprints/ [Accessed 9 July 2020].

[5] Jinja, 2., 2020. Introduction — Jinja Documentation (2.11.X). [online] Jinja.palletsprojects.com. Available at: https://jinja.palletsprojects.com/en/2.11.x/intro/ [Accessed 9 July 2020].

[6] FinanceTrainingCourse.com. 2020. From Http To Https. AWS SSL Install Guide. Apache Ubuntu 14.04 — Financetrainingcourse.Com. [online] Available at: https://financetrainingcourse.com/education/2016/06/ssl-certificate-install-aws-guide-apache-ubuntu-14-04/ [Accessed 9 July 2020].

[7] Flask.palletsprojects.com. 2020. Foreword — Flask Documentation (1.1.X). [online] Available at: https://flask.palletsprojects.com/en/1.1.x/foreword/ [Accessed 9 July 2020].

[8] flask, P., Agrahari, S. and Killian, S., 2020. Process Ajax Request In Flask. [online] Stack Overflow. Available at: https://stackoverflow.com/questions/48595068/process-ajax-request-in-flask [Accessed 9 July 2020].

[9] Go, G., 2020. Project Management. [online] Github. Available at: https://github.com/users/ChandanaNS/projects/2 [Accessed 12 July 2020].

[10] Grinberg, M., 2020. Running Your Flask Application Over HTTPS. [online] Blog.miguelgrinberg.com. Available at: https://blog.miguelgrinberg.com/post/running-your-flask-application-over-https [Accessed 9 July 2020].

[11] NS, C., 2020. Grocerygo. [online] http://54.72.85.207/. Available at: https://github.com/ChandanaNS/OnlineGroceryWebsite[Accessed 12 July 2020].

[12] JournalDev. 2020. Python Ordereddict - Journaldev. [online] Available at: https://www.journaldev.com/21807/python-ordereddict [Accessed 9 July 2020].

[13] tunnel, S., 2020. Sshtunnel. [online] PyPI. Available at: https://pypi.org/project/sshtunnel/ [Accessed 9 July 2020].