Riara University
nurturing innovators

Tamu Virtual Restaurant System

(Food E-commerce)

By

Otieno Vincent

17ZAD102651

Supervisor: Paul Mwaniki

A FINAL PROJECT DOCUMENTATION SUBMITTED IN PARTIAL FULFILMENT OF THE DIPLOMA IN COMPUTER SCIENCE IN THE SCHOOL OF COMPUTING SCIENCES. RIARA UNIVERSITY.

Submitted on 26th May ,2021

Nairobi Kenya.

DECLARATION.

I declare that this project proposal is my original work and has not been presented for a degree in any other university. Therefore, my presentation is non-published and contains no materials of another person or any collaboration whatsoever, except where due reference is made.

OTIENO VINCENT

…………………………………………………………

…………………………………………………………

…………………………………………………………

…………………………………………………………

DEDICATION.

This project is dedicated to my parents who taught me to believe working hard is the only way to make it in life. They also taught me how to do things in a smart way to achieve better things. With that I believe I can make the world a better place.

This is also dedicated to all food enthusiasts out there. People who are passionate about food and food business. They inspired me to be creative enough to make my contribution towards the food industry.

ACKNOWLEDGEMENT

I would like to acknowledge my lecturer Mr. Paul Mwaniki for giving me the opportunity to do this project to the best of my knowledge and learn from it. I also thank the administration for providing me with the resources like proper network connection to do my research in order to do and present this work in due time.

APPROVAL

This project of Otieno Vincent was reviewed by:

Supervisor Name: Mr. Paul Mwaniki.

Signature…………………………….

ABSTRACT

E-commerce has experienced a massive growth after the invention of the internet. The creation of new business models such as online companies whereby the company conducts most of its business on the internet. Indeed, there is very little doubt that the web has created it potential to begin a world on-line business than it ever was to begin a business before. Trading online while fraught with different challenges has been found to pose significant benefits to businesses, consumers and even the society in general.

With the same, Nairobi-Kenya is the leading city across Africa for online food delivery, followed closely by Casablanca-Morocco, Lagos-Nigeria Kampala-Uganda and Abidjan-Ivory Coast. This was attested by the Jumia Kenya Food Index 2020.

In this case, we see that online food business has grown tremendously in the past years. This is the market that can be tapped into with ease to make profits. This will contribute to the reduction of unemployment at an impressive rate. During the pandemic, we have witnessed a rise and boom on online businesses and exceptional transactional activities online. The top benefits were cost reduction for the business startup and running of the business, ability to reach more potential customers, access to regional and/or global markets, easier and faster to serve customers, customers' ability to access the business on a 24/7 basis, better prices to customers, enhanced collection of customer data, higher quality customer service, increased business visibility through search engine marketing and convenience to customers.

Tamu restaurant is where the cooks get to be creative and offer high quality cuisines for less amount. High end cuisines for everyone and anyone. This time people will not have to go to expensive restaurants to get the same meal they would find at tamu. Another wonderful part is that the food gets delivered right on their doorstep.

Tamu is also highly scalable, we will expand it to accommodate all the talented cooks and chefs. This will in turn create more business and employment the country men and women. This is where the cooks can start up their businesses without any capital but only the ingredients. They will be able to reach more customers than a stationary restaurant. Costs of rent and other bills will be cut tremendously since they can make home cooked food and list them on tamu. This will make them more profits for their businesses.

Another wonderful thing is that each and every chef or cook will have their own virtual restaurants. This will be where the business owners are put into clusters, ideally every one gets their own unique virtual space bearing their brand name.

TOOLS AND REQUIREMENTS

This project will require us to use some variety of tools and frame works in order to achieve the end goal and for it to run successfully. Some of the tools are: -

(I) A programming backend language (Python).

(II) A database (MySQL).

(III) GitHub package (version control and the remote repository to back up the project).

(IV) Django Framework and its dependencies.

(V) HTML, JavaScript and CSS.

(VI) Stripe for payments.

And for the system requirements, the following specifications will be required

- A personal computer with standard specifications.
- An internet connection.

LIST OF ACRONYMNS AND ABBREVIATION

HTML – hypertext markup language

CSS – cascading style sheet

JS – JavaScript

JSON – JavaScript object notation

Table of Contents

DEDICATION. ii

ACKNOWLEDGEMENT iii

APPROVAL iv

ABSTRACT v

TOOLS AND REQUIREMENTS vii

LIST OF ACRONYMNS AND ABBREVIATION viii

TABLE OF FIGURES xi

LIST OF TABLES xii

CHAPTER 1: INTRODUCTION 1

1.1 BACKGROUND INFORMATION: 1

1.2 PROBLEM STATEMENT 1

1.3 OBJECTIVES 3

1.3.1 GENERAL OBJECTIVES 3

1.3.2 SPECIFIC OBJECTIVES 3

1.4 JUSTIFICATION 3

1.5 SCOPE 3

CHAPTER 2: LITERATURE REVIEW 4

2.1 INTRODUCTION 4

2.2 WEB APPLICATION 4

2.3 SYSTEM TECHNOLOGIES 5

2.4 CASE STUDY 6

2.5 HOW TAMU WORKS 7

2.6 BENEFITS AND DRAWBACKS 7

CHAPTER 3: METHODOLOGY/ ANALYSIS AND DESIGN 9

3.1 REQUIREMENTS AND ANALYSIS 9

3.2 DESIGN 10

3.2.1 WEB DESIGN 10

3.2.2 DEVELOPMENT MODEL 10

3.2.3 ENTITY RELATION DIAGRAM 12

3.2.4 DATA FLOW DIAGRAM 15

3.2.5 WORK BREAKDOWN 16

CHAPTER 4: IMPLEMENTATION, TESTING AND RESULT 17

4.1 IMPLEMENTATION 17

4.2 TESTING AND ANALYSIS 18

4.2.1 AUTHETICATION 20

4.2.2 AUTHENTICATION ERRORS FOR CLIENTS 22

4.2.3 USER BILLING INFO 24
4.2.4 ADMINISTRATION PANEL 25
CHAPTER 5: CONCLUSION, EVALUATION AND FURTHER WORK 26
5.1 OBJECTIVES 26
5.1.1 A SINGLE NODE (VIRUAL RESTAURANT) FOR INDIVIDUALS OR RESTAURANTS. 26
5.1.2 SIMPLE, EASY AND GOOD USER EXPERIENCE 27
CLIENTS 5.1.3 CHOOSING RESTAURANTS 29
5.2 CONCLUSION 29
5.3 EVALUATION 29
5.4 FURTHER IMPROVEMENTS 30
APPENDICES 31
APPENDIX I: Views Sample Code. 31
APPENDIX II: Model (database schema) Sample Source Code. 32
APPENDIX III: Profile Sample Source Code 32
REFERENCES 33

TABLE OF FIGURES

Figure 1: Agile Methodology.

Figure 2: Entity relational diagram.

Figure 3: Relation between tables.

Figure 4: Dataflow diagram.

Figure 5: Checking the site connection.

Figure 6: Starting the local server.

Figure 7: The home page.

Figure 8: Home page showing the cards.

Figure 9: The details page.

Figure 10: Sign Up page.

Figure 11: Sign In page.

Figure 12: Logout.

Figure 13: Sign up Validation.

Figure 14: Sign up validation 2.

Figure 15: Sign in validation.

Figure 16: Sign in validation for admin.

Figure 17: Checkout form.

Figure 18: Order Summary.

Figure 19: Administration panel.

Figure 20: Adding an item for listing.

LIST OF TABLES

Table 1: The Gantt chart

CHAPTER 1: INTRODUCTION

1.1 BACKGROUND INFORMATION:

The internet has become an important part of the modern-day societies and economies across the world. It has led to massive transformations in the business world, social interactions, communication, research, education, governance, philanthropic endeavors and many other sectors of life (St.Amour, 2012). The Internet World Statistics (2012) estimated that the global internet users to be about 2.4 billion, with well developed nations having tremendously higher internet penetration rates. Internet usage in Kenya started in the early 1990s right until 2009 internet connection was low and expensive as the country relied on expensive satellite connections (Souter & Kerretts-Makau, 2012). Nevertheless, since the landing of submarine cables internet access has impressively improved, the Communications Commission of Kenya [CCK], 2012 estimated that there are 14 million internet users in the country. It is important to note that 99% of internet subscriptions in Kenya are mobile subscriptions which can be attributed to influx of smart phones and advanced feature phones (Souter & Kerretts-Makau, 2012).

The manifestation of the internet saw entrepreneurs all over the world capture ideas and combine technological innovations to create new products, services and business models (Mahadevan,2000; Hasan & Harris, 2009). The internet phenomenon also gave rise to purely internet-based businesses now commonly referred “internet-based companies” or “online company” a name that is applied to a company that conducts ‘most’ of its business online, since such businesses also engage physical logistic systems (Schultz, 2009).

The convenience of online shopping has compelled consumers who value the ease and efficiency of purchasing some services online such as travel packages, tickets, food and hospitality.

This show how potential for e-commerce is available for everyone. Tamu and other restaurants have a huge chance on the huge piece of cake. There is a chance for the entrepreneurial ideas and Tamu is one of them.

1.2 PROBLEM STATEMENT

The technology has become a very big part of our lives. It has become a wonderful tool to create and do business. It has proved that we can make more profits through connectivity and more relationship can be built. In Kenya, there is no a platform that hosts restaurants that make cuisines, wonderfully crafted food. We can only mostly find fast food restaurants offering their services. But now for anyone to wo wants cuisine they really have to do a massive digging in the internet looking for that restaurant they would want to eat from and to make it tiresome. They have to go there physically. The purpose of the proposed system, is to be able to host possibly all the cuisine-oriented food restaurants. The restaurants will get their own full-fledged unit on the platform. Now users can find the food they want in the restaurant they like.

1.3 OBJECTIVES

1.3.1 GENERAL OBJECTIVES

The general objective of this project is to be able to make a platform that can host restaurants with exclusive units. I will be like a market of restaurants, whenever a client clicks on a certain restaurant they are directed into their unit where they will find all the listings of the restaurants and even book early for a special cuisine, if they wish for it to be delivered or go there physically. They will be able to find everything they need from the restaurant on their restaurants.

1.3.2 SPECIFIC OBJECTIVES

The project will be guided by the following objectives:

I. To create a system that allows restaurants and individuals sell their food and services.

II. To create a system where clients will be able to choose the restaurants they like and near them.

III. To design an interface that is good user experience that will be easy and simple to use.

IV. To create units for the restaurants which can be customized according to their requirements.

V. To create a platform where home chefs and cooks can make their virtual restaurants and do business.

1.4 JUSTIFICATION

The development of this platform will be valuable for everyone especially people who understand that they don't have to order fast food to get to enjoy food. Also, people who like enjoying new unique kind of foods. This is a platform where users can order Italian food and get delivered in their doorsteps.

1.5 SCOPE

The project will focus on helping both restaurants and chefs to their potential clients over the internet. The restaurants or chefs get their own unit where they can list what is on their menus. The customers get to choose and order. The project also has room for future improvements. This also a platform that will help customers to be able to order cuisines online to their doorsteps.

CHAPTER 2: LITERATURE REVIEW

2.1 INTRODUCTION

The proposed project will be an E-commerce web application, it will leverage a database to keep records, keep credentials, aid in authentications, the food pictures, items, orders, credit card info, addresses, and all the logistics information. The platform prototype will use stripe as payment method but we will expand to other payment options Like M-pesa and PayPal.

We will create a unit as a prototype to demonstrate how a virtual restaurant will look like.

2.2 WEB APPLICATION

In a web application, front-end and backend development is needed. On Front-end, we deal with creating the user experience using HTML, CSS and JavaScript. This where we list all the product items, the specific product page, the cart, order summery, information form and so much more visible things to the user.

On the back-end, is where we handle all the logic works, this where handle how the front end interacts with the database. How the UI can edit, add, create and delete Items in the database. In simple words how, the user can manipulate the database.

- STATIC WEB APPLICATION

  Any web application that can be delivered directly to an end user's browser without any server-side getting involved to the HTML, CSS, or JavaScript content can be referred to as a static web application.

  The client on the web browser sends a request at a particular address into a web server, the network request is received by the server, which maps the address to a "barebones" HTML file on the server.

  The HTML file is sent from the server to the user agent, which renders it and displays it to the user or client.

  The JavaScript in the HTML page connects to the data services via the browser and retrieves the information it requires to construct the web page content.

  The JavaScript takes the data and manipulates the HTML, updating it with the data retrieved earlier.

- DYNAMIC WEB APPLICATION

  This is a web app that has been built on top of a framework. This kind of software controls the web page construction and aid in easier maintenance. The way such web app is rendered on the user's screen is dynamically influenced the app logic on the back-end or the front-end.

- ONLINE STORE OR E-COMMERCE

  The online store is also a website, but it serves a different purpose in that it allows people to shop and pay online. So, when someone makes a purchase on the site without interacting with a human, we can call it an online store. Examples of Ecommerce stores are Amazon, eBay, Alibaba etc.

- PORTAL WEB APP

  This is a web-based platform that gathers data from various sources into a single user interface and displays it to users in the most appropriate way for their situation. Simple web portals evolved into portal platforms that support digital customer experience initiatives over time. Only a particular user or group of users can access the data in the web portal.

- PROGRESSIVE WEB APP

  Also known as PWA is a type of web-based application that is designed with popular web technologies such as HTML, CSS, and JavaScript. It should run on any computer with a standards-compliant browser, which includes both desktop and mobile devices.

2.3 SYSTEM TECHNOLOGIES

- DATABASE

  A database is a collection of information that is organized so that it can be easily accessed, managed and updated. It typically contains aggregations of data records or files, containing information about transactions and interactions.

  There are different kinds of databases MySQL(Open source), MariaDB, MongoDB, Redis, PostgreSQL etc. On this project we will use MySQL. This is because it is open source and it has enormous amount of support and documentation, this is due to its popularity.

- ORM – OBJECT RELATIONAL MAPPING

  Object–relational mapping is a programming technique that uses object-oriented programming languages to translate data between incompatible type systems. This effectively produces a "virtual object database" that can be accessed from the programming language itself. This helps the web app to be able to map the virtual data into the database.

- PYTHON LAGUAGE

  Python is a general-purpose scripting language, which means that it can be used for other forms of programming and software creation apart from web development, unlike HTML, CSS, and JavaScript.

- DJANGO ALL-AUTH

  It is a set of Django integrations addressing authentication, registration, account management, and 3rd party (social) account authentication.

2.4 CASE STUDY

Ecommerce has been given a much-needed boost in customer interest for them to convert from the normal stores. Kenya has always been a better target market for e-commerce, majorly due to mobile money penetration, increasing internet adoption and willingness of the population adopting to the fast-ever-changing technology.

According to Google statistics, foot traffic into retail and to grocery stores, in Nairobi has gone down by an average of 52% and 40% respectively by march 2020.

The great digital revolution, burgeoning e-commerce, and the advantage of mobile money, which enables shoppers to pay for their deliveries right from their phones, has sparked a whole new sector in the food delivery market, which is now a multibillion-dollar industry.

And Kenyans have flocked to the concept, enticed by the convenience of ordering food from their favorite local restaurants online, getting it delivered to their door, and being able to pay in cash or via mobile money. The food delivery companies have established alliances and agreements with restaurants to deliver food to city dwellers at their doorstep, relying on Internet connectivity.

The pattern began in Nairobi and has since spread to cities such as Mombasa, Nakuru, Eldoret, and Kisumu. According to several sources familiar with the situation, the food distribution industry is only getting started in terms of order volume.

According to Jumia's Food forecast, on-demand delivery growth is expected to accelerate by more than 50% by 2020.

2.5 HOW TAMU WORKS

The web ecommerce will have a landing page where the different units of restaurants the client will click one of them. After clicking a restaurant, the are redirected to the restaurant's main portfolio. This where they will find all the food in the menu and can request on them to be customized. They click on a cuisine, beverage or drinks and they get redirected to the product page where they can find more information about the product and they can order it at this point. After adding the item to cart they can either choose to go back to continue shopping or they can move to checkout. If they click checkout they are redirected to an order summary where items in the cart are listed. The price of individual items, the total and the discount if it exists. If everything is okay the way the expect, they click proceed where their information is collected then finally move to payment and pay. The order will be confirmed and marked for delivery.

Whenever a refund case arises, they can request for it and they can be handled well by the system.

The authentication of the system will be using allauth. The user will need login or sign up in order for any kind of transaction to take place.

The admin will also have a centralized admin panel where they can manage their system from. They can handle and manipulate the date in the system but only if the admin has super user privileges. This means they can change any kind of data in the system.

2.6 BENEFITS AND DRAWBACKS

BENEFITS: -

- Less startup costs for starting their online businesses. They will not need to have a physical restaurant.
- No fast foods will be sold on the platform.

- All restaurants will have a centralized administration management system for them to track their logistics.
- Clients can find their favorite restaurants easily near them as well.
- It has payment functionality.
- Restaurants can list their signature dishes.
- Anyone who can cook good food can join.

DRAWBACKS: -

- It requires an internet connection since it is web based.
- Some people might find it hard to use especially the illiterate.

CHAPTER 3: METHODOLOGY/ ANALYSIS AND DESIGN

In this chapter, we will depict design and methodology that we will use to make the Tamu Ecommerce. There are most ecommerce which have already implemented the design along with an extensive research.

This platform will be a web application, which means it will have both front-end and back-end functionality. On the front-end side, we will use HTML, CSS, Bootstrap and JavaScript and for the back-end side, we will use Python language, Django framework, MySQL database, some python dependencies and jinja 2(this is used to connect the back-end logic with to the front end).

3.1 REQUIREMENTS AND ANALYSIS

Everybody will be using the system but in order for exclusivity and privacy, users will need to log in or sign up before any transactions or activity is carried out. The Django framework has a module that can handle this feature that way, we will not have to re-invent the wheel. A database is needed to store everything that needs to be stored for the logic processing. With a database we will have to use MySQL. Reason is, MySQL is the world's most popular open source database and it provides comprehensive support for each and every single application development need. It is the most preferred data management server.

For the UI(user interface), will needs to be accessible and as simple as possible. We will use HTML to create the structure of the web document. This will create the skeleton for the whole user interface of our web application. Next, we will use the CSS to prettify the user interface, give it a good layout to make the web application look good.

We will use JavaScript to add interactivity to our application like what the button does when clicked, auto filling forms and saving data. Bootstrap will be used for specific styling like creating grid of cards.

Django is a web application framework written in python. It is robust and can be used to create small to large web applications fast. Another thing that makes Django to stand out is, it provides an administration dashboard that you can also customize. With Django, we do not have to write the actual SQL in order to connect with the MySQL database. Rather, we will use ORM (Object Relational Mapper). This where we use python-oriented language to map and connect to the database and also get to enjoy more features.

Django also has its own inbuilt local host server to serve the project while in development mode. The server runs in port 3000.

3.2 DESIGN

3.2.1 WEB DESIGN

The web application will consist of web pages and functionalities for front-end where clients will have to interact with on the system. The system will be both responsive and dynamic. Responsive in the sense that it can be comfortably used on mobile phones and tablets. The interface will include several features like:- navbar, hyperlinks, breadcrumbs for sliders, buttons, forms, card images, product page, summary page, info and checkout page.

For the system to become dynamic, the frontend will have to constantly and seamlessly interact with the database to retrieve data. With python, MySQL database and the ORM for backend the web application will have the capability to interact with the database wholesomely. Activities that are involved in the backend consist of creating, Reading, Updating and Deleting of data from the database.

The information that will be collected and save are user data, restaurants data, images, titles, descriptions, prices and so much more.

For security, the system will consider top notch user data protection like good and secure login using a valid email and password. For credit card info we will use the stripe API, this means we will not store the credentials ourselves but stripe itself will store the data securely. Instead it will offer the system with a unique token during transactions.

Django offers comprehensive security using a feature called SRF_TOKEN. This prevents hackers from creating scripts that can compromise the system.

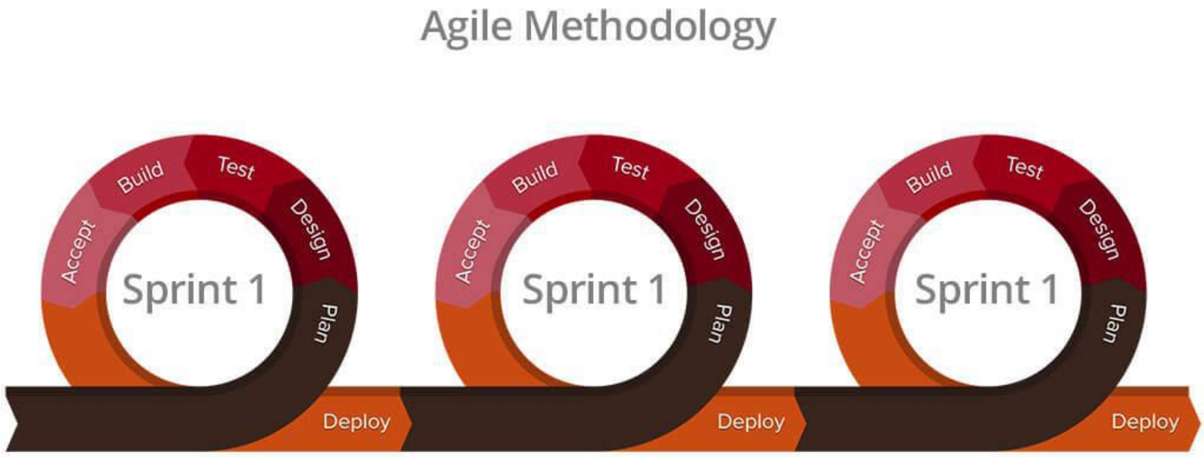
3.2.2 DEVELOPMENT MODEL

For a project to be successful, ta development model should be used to aid the entire life cycle of the project development. There are a couple of development models like waterfall model, agile model, prototyping, incremental development model and many others.

The development methodology for this project will be agile development. This methodology was chosen because it gives capability for multiple iteration during the development cycle due to its flexibility. Agile methodology is defined as a group of software development models based on the incremental and iterative approach to software design and development. The increments are small and normally new version releases of the system which are created and made available after some time.

Due to the fast movement of technology, agile development method is the the best approach and much easier to deploy new updates or completely revamp some feature in the project.

What really aids this project during its development is version control using git. In git, you can make changes to a project in chunks without messing up the main project. The project is then hosted on a remote repository on a remote server called GitHub. During the minor changes and updates branches are created using git, then after the changes have been done it is merged to the main project code for it to be consumed by users of the system.



Thandi Guiliherme (Author) Nov 2018.

Figure 1: Agile methodology

3.2.3 ENTITY RELATION DIAGRAM

The web application will be making frequent request to the database, this means it will contain different tables of normalized databases. All the tables will be real-time so whenever any request is made from frontend (like login, signup, listing of menus, product page request, order summary, checkout and ordering). The database table will have:-

- Items
  - This table will have all the contents of the food items, images, prices, descriptions, discount price, category and slug for the url. This are items that a user can click on on the frontend and it will make queries from the database.
- Item
  - This will be a normalized table for the Items table. This represents a single component in the items list.
- Order Item
  - This part of the table will store the items and quantity if the items that a user wants to order the part of the table is the part that will be displayed on the order summary. Items that are waiting to be paid for.
- Order
  - This is the part of the database that contains the user details as foreign key, items ordered, order date, reference code, shipping address, billing address, payment information and all the other logistics information after payment.

The following figure depicts an entity relation diagram of the system. It contains multiple tables that are intertwined with each other with different types relationship process.

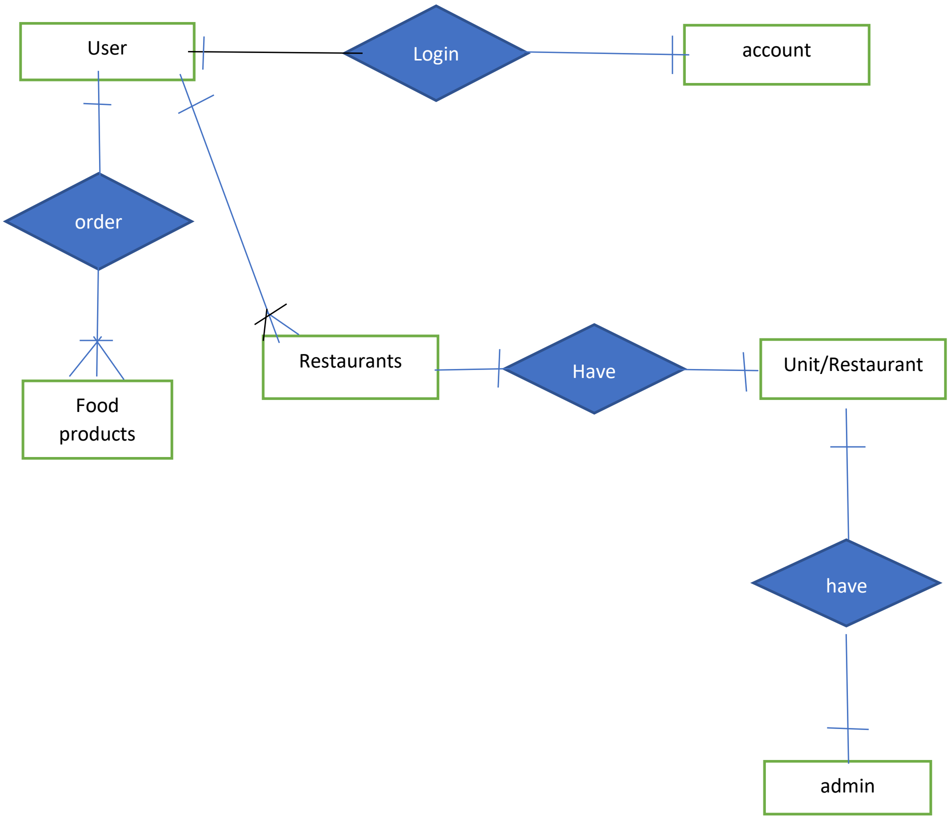


Figure 2: Entity relation diagram

Figure 3 is showing the relationship of all the tables in the database and how the primary key is used as foreign key in other tables.

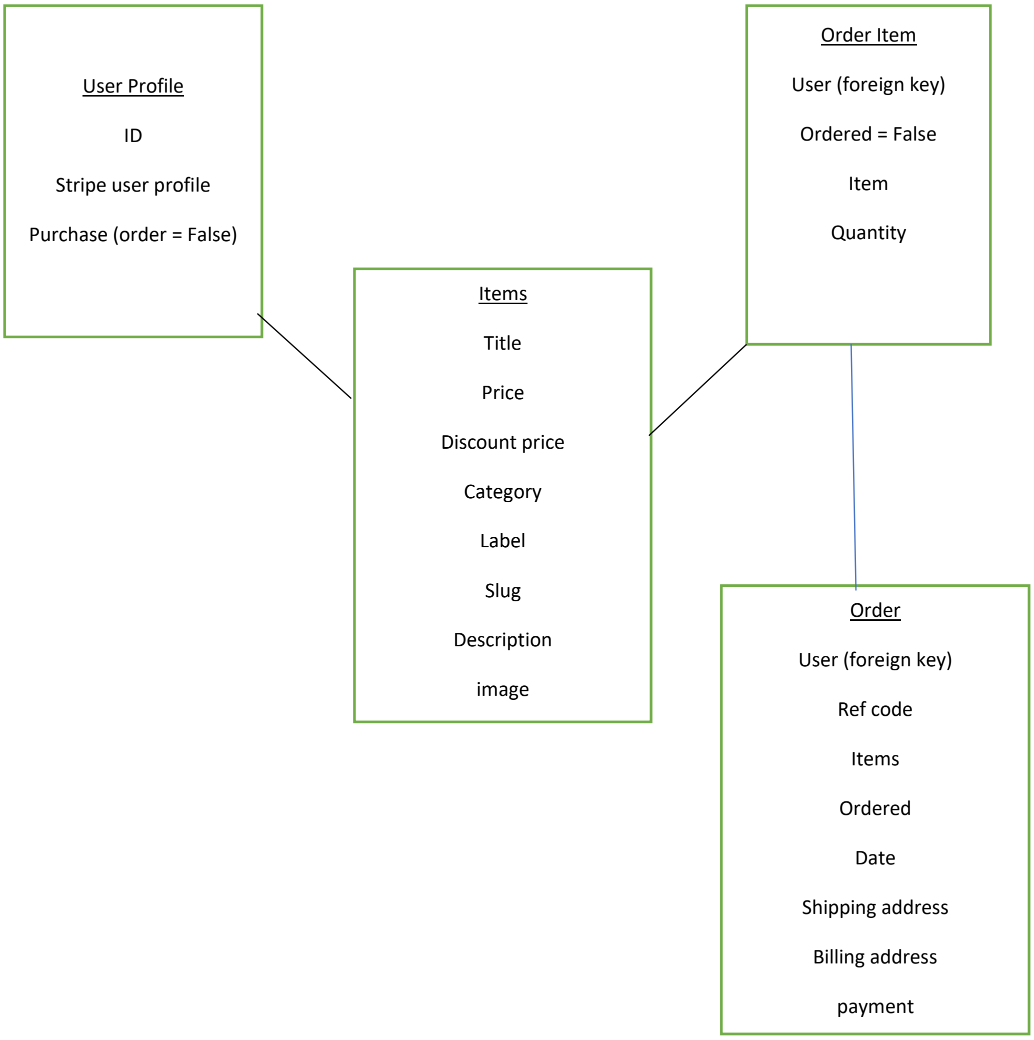


Figure 3: Relationship between tables

3.2.4 DATA FLOW DIAGRAM

The following shows the execution flow. The system starts by login in in 2 ways regular and admin login.

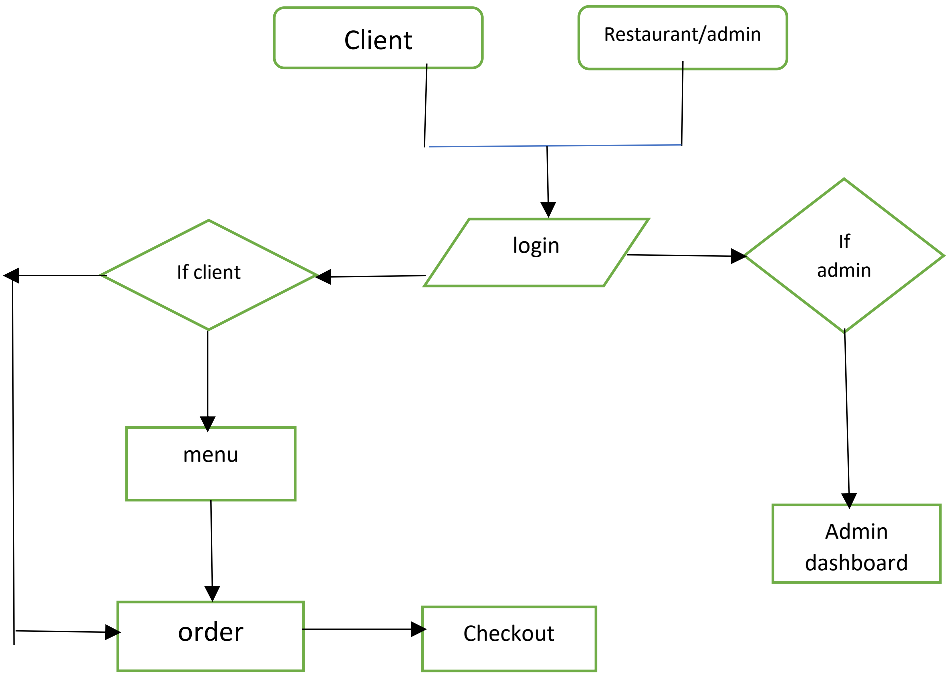


Figure 4: Dataflow diagram

3.2.5 WORK BREAKDOWN

WEEKS	1	2	3	4	5	6	7	8	9
Research	⬛	🟦							
Survey Analysis			🟩						
Report and Code				🟨	🟨	🟨	🟨		
Evaluation								🟦	
Final re									🟩

Table 1: The Gantt chart

CHAPTER 4: IMPLEMENTATION, TESTING AND RESULT

In this chapter, we will see the implementation of the design, the testing process and the results that was carried out during the development of this project.

4.1 IMPLEMENTATION

The Tamu restaurant system consists of various features. The features include the home page, products list page, product detail page, products summary page, checkout page, payment page and admin dashboard. Database used is MySQL with ORM as the database mapper to map the data into the database.

The web application is leveraging python 3.7 specifically Django framework 2.2 on the backend. Different kinds of dependencies like Pillow for image uploading, Django-allauth for authentications, pytz for time zone, urllib3 for URL routing and so many more.

For the frontend, it is leveraging HTML5, JavaScript, CSS and Bootstrap4. The home layout is in grid formation. An item card is clicked and the product page is displayed. This is due to the unique slug(id) for every single product. The slug also aids in the purchase process as the system knows which product is the client dealing with.

Visual studio code was the main editor tool of choice. It has so many useful features that aids in easier and seamless development stage. It has full-fledged support for git and GitHub. I could push my code with literally a click of a button. That is after onetime configuration.

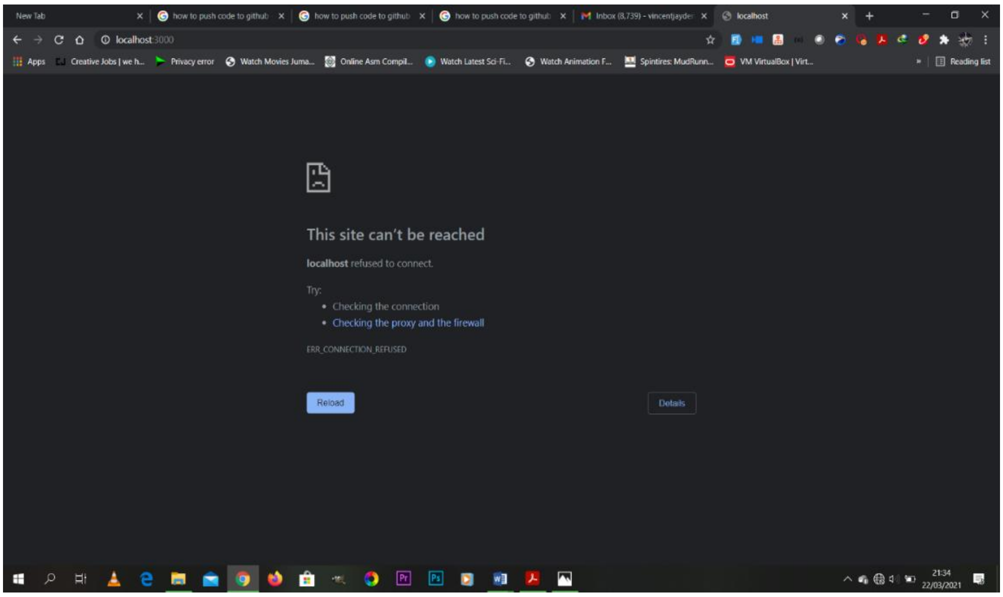
The project leveraged the use of git for version control. This enabled the process to be worked on in a agile movement this enabled me to work on small parts slowly while merging after that particular feature to the main codebase.

In the case of authentication, I was able to use a powerful Django-allauth library. The library has so many feature functionalities like helping the system remember the user.

The system has an admin dashboard that enables the restaurants admin or the chefs track everything in their businesses from users to orders. The also upload content through the admin dashboard.

4.2 TESTING AND ANALYSIS

The main aim of testing and analysis is to ensure that the system can handle all errors well with minimal details to the user. The errors can be caused by human faults or system fault.



This site can't be reached

localhost refused to connect.

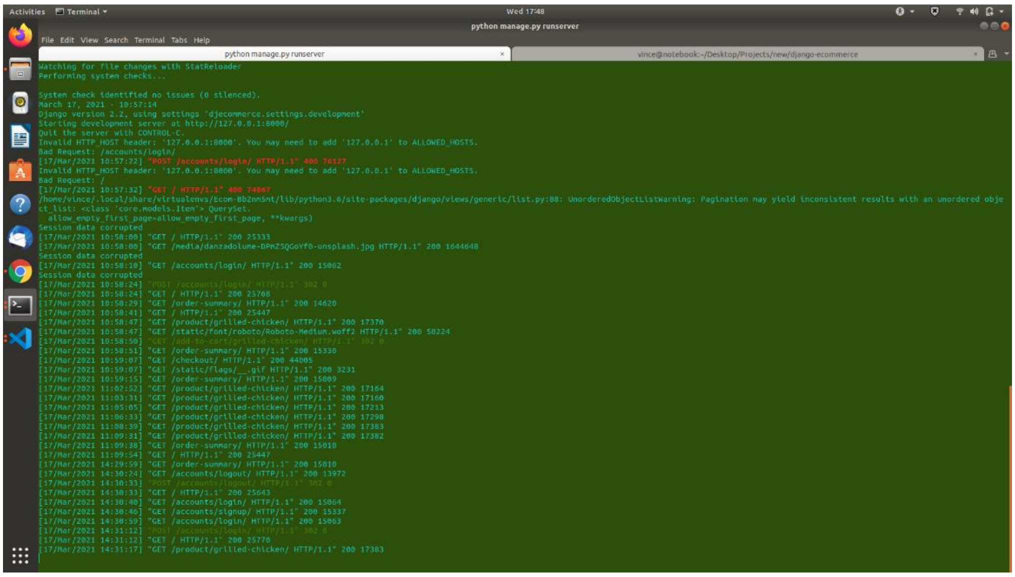
Try:

- Checking the connection
- Checking the proxy and the firewall

ERR_CONNECTION_REFUSED

Figure 5: Checking site connection (Failed)

This figure 5 shows the error you get in case the internet connection fails. It doesn't show the user much but rather the typical connection error.

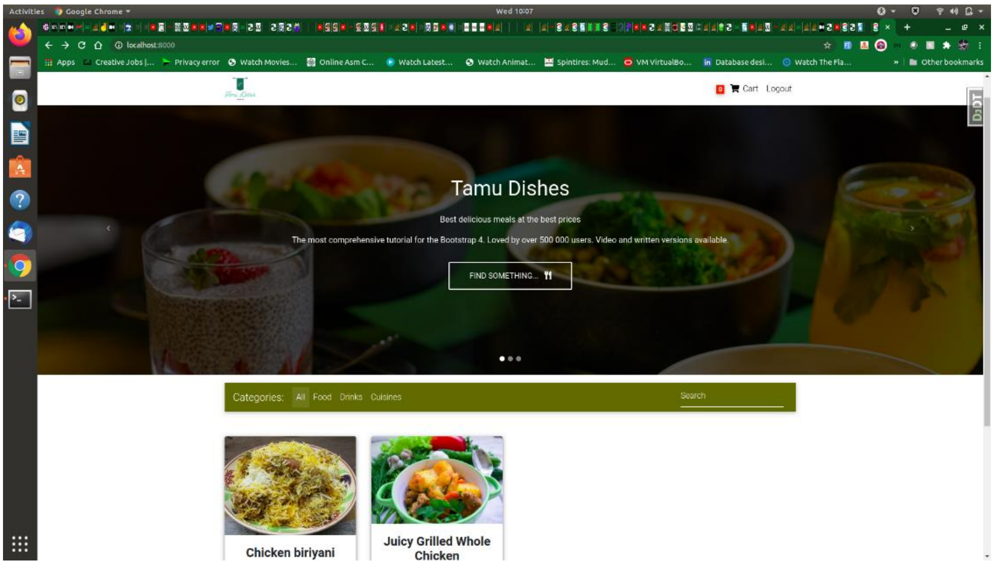


```
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
March 17, 2021 - 10:57:14
Django version 2.2, using settings 'djecommerce.settings.development'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
Invalid HTTP_HOST header: '127.0.0.1:8000'. You may need to add '127.0.0.1' to ALLOWED_HOSTS.
Bad Request: /accounts/login/
[17/Mar/2021 10:57:22] "POST /accounts/login/ HTTP/1.1" 400 76127
Invalid HTTP_HOST header: '127.0.0.1:8000'. You may need to add '127.0.0.1' to ALLOWED_HOSTS.
Bad Request: /
[17/Mar/2021 10:57:32] "GET / HTTP/1.1" 400 74867
/home/vince/.local/share/virtualenvs/Ecom-Bb2nm5mt/lib/python3.6/site-packages/django/views/generic/list.py:88: UnorderedObjectListWarning: Pagination may yield inconsistent results with an unordered obje
ct_list: <class 'core.models.Item'> QuerySet.
  allow_empty_first_page=allow_empty_first_page, **kwargs)
Session data corrupted
[17/Mar/2021 10:58:00] "GET / HTTP/1.1" 200 25333
[17/Mar/2021 10:58:00] "GET /media/danzadolume-DPmZSQGoYf0-unsplash.jpg HTTP/1.1" 200 1644648
Session data corrupted
[17/Mar/2021 10:58:10] "GET /accounts/login/ HTTP/1.1" 200 15062
Session data corrupted
[17/Mar/2021 10:58:24] "POST /accounts/login/ HTTP/1.1" 302 0
[17/Mar/2021 10:58:24] "GET / HTTP/1.1" 200 25768
[17/Mar/2021 10:58:29] "GET /order-summary/ HTTP/1.1" 200 14620
[17/Mar/2021 10:58:41] "GET / HTTP/1.1" 200 25447
[17/Mar/2021 10:58:47] "GET /product/grilled-chicken/ HTTP/1.1" 200 17370
[17/Mar/2021 10:58:47] "GET /static/font/roboto/Roboto-Medium.woff2 HTTP/1.1" 200 50224
[17/Mar/2021 10:58:50] "GET /add-to-cart/grilled-chicken/ HTTP/1.1" 302 0
[17/Mar/2021 10:58:51] "GET /order-summary/ HTTP/1.1" 200 15330
[17/Mar/2021 10:59:07] "GET /checkout/ HTTP/1.1" 200 44005
[17/Mar/2021 10:59:07] "GET /static/flags/__.gif HTTP/1.1" 200 3231
[17/Mar/2021 10:59:15] "GET /order-summary/ HTTP/1.1" 200 15009
[17/Mar/2021 11:02:52] "GET /product/grilled-chicken/ HTTP/1.1" 200 17164
[17/Mar/2021 11:03:31] "GET /product/grilled-chicken/ HTTP/1.1" 200 17160
[17/Mar/2021 11:05:05] "GET /product/grilled-chicken/ HTTP/1.1" 200 17213
[17/Mar/2021 11:06:33] "GET /product/grilled-chicken/ HTTP/1.1" 200 17298
[17/Mar/2021 11:08:39] "GET /product/grilled-chicken/ HTTP/1.1" 200 17383
[17/Mar/2021 11:09:31] "GET /product/grilled-chicken/ HTTP/1.1" 200 17382
[17/Mar/2021 11:09:38] "GET /order-summary/ HTTP/1.1" 200 15010
[17/Mar/2021 11:09:54] "GET / HTTP/1.1" 200 25447
[17/Mar/2021 14:29:59] "GET /order-summary/ HTTP/1.1" 200 15010
[17/Mar/2021 14:30:24] "GET /accounts/logout/ HTTP/1.1" 200 13972
[17/Mar/2021 14:30:33] "POST /accounts/logout/ HTTP/1.1" 302 0
[17/Mar/2021 14:30:33] "GET / HTTP/1.1" 200 25643
[17/Mar/2021 14:30:40] "GET /accounts/login/ HTTP/1.1" 200 15064
[17/Mar/2021 14:30:46] "GET /accounts/signup/ HTTP/1.1" 200 15337
[17/Mar/2021 14:30:59] "GET /accounts/login/ HTTP/1.1" 200 15063
[17/Mar/2021 14:31:12] "POST /accounts/login/ HTTP/1.1" 302 0
[17/Mar/2021 14:31:12] "GET / HTTP/1.1" 200 25770
[17/Mar/2021 14:31:17] "GET /product/grilled-chicken/ HTTP/1.1" 200 17383
```

Activities
Terminal
Wed 17:48
python manage.py runserver
File Edit View Search Terminal Tabs Help
python manage.py runserver
vince@notebook:~/Desktop/Projects/new/django-ecommerce

Figure 6: Starting the local server (server running)

On figure 6 we start the local sever by running "python manage.py runserver" on the terminal and it will start the local server on port 8000. This will enable the web application to run and be accessible on our local machine.



Tamu Dishes

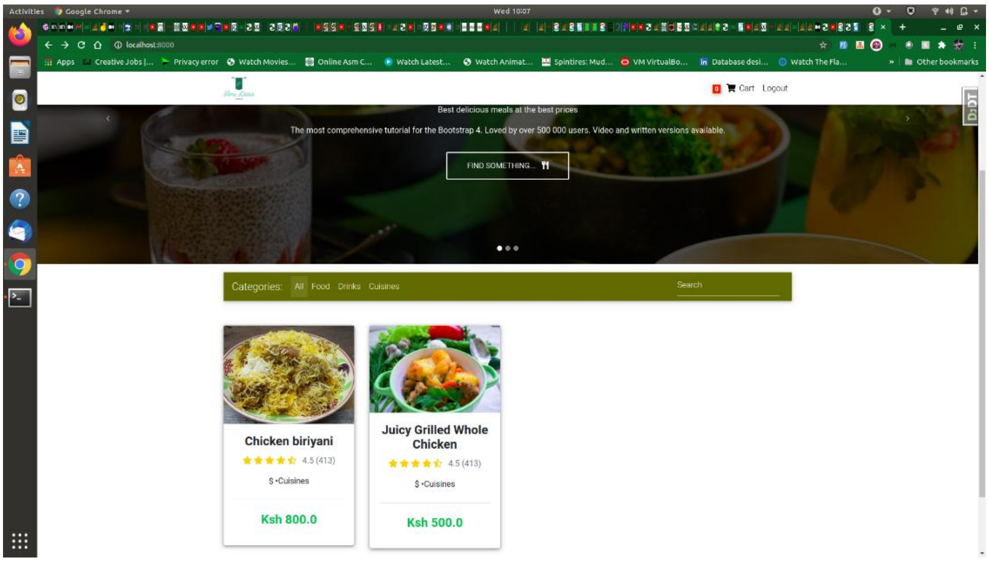
Best delicious meals at the best prices

The most comprehensive tutorial for the Bootstrap 4. Loved by over 500 000 users. Video and written versions available.

Chicken biriyani

Juicy Grilled Whole Chicken

Figure 7: The home page (server running)



Best delicious meals at the best prices

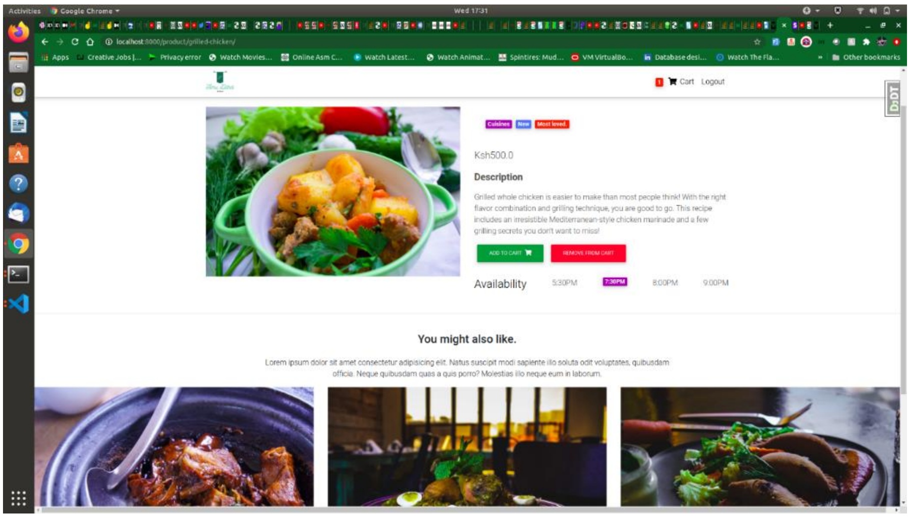
The most comprehensive tutorial for the Bootstrap 4. Loved by over 500 000 users. Video and written versions available.

Chicken biriyani
4.5 (413)
$ •Cuisines
Ksh 800.0

Juicy Grilled Whole Chicken
4.5 (413)
$ •Cuisines
Ksh 500.0

Figure 8: Home page showing the cards (server running)

Both figure 7 and 8 are the welcoming home page of a restaurant. This is where the user gets to scroll on to find whatever cuisine they want to buy. When they want to buy a certain cuisine, they just click on it and they are taken to the details page for information and adding to the cart.



Ksh500.0

Description

Grilled whole chicken is easier to make than most people think! With the right flavor combination and grilling technique, you are good to go. This recipe includes an irresistible Mediterranean-style chicken marinade and a few grilling secrets you don't want to miss!

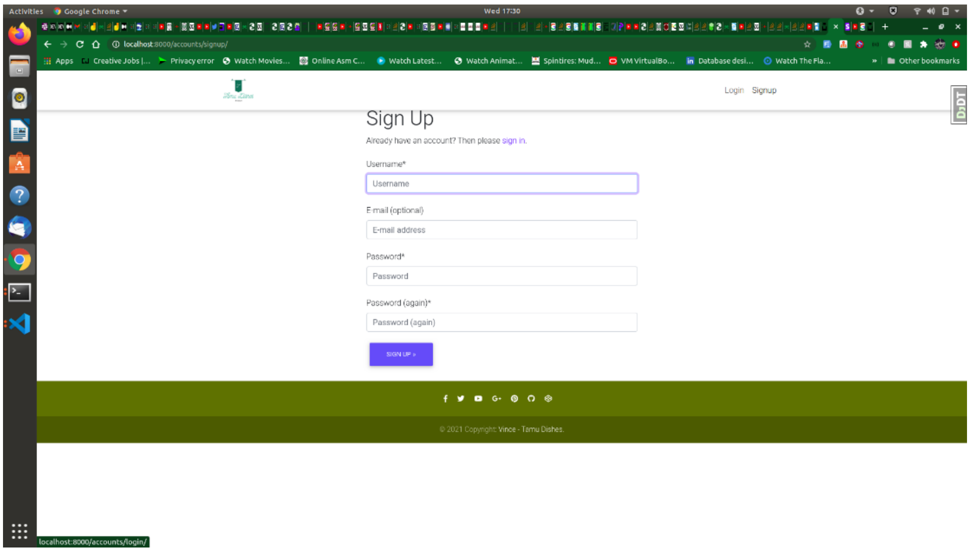
You might also like.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Natus suscipit modi sapiente illo soluta odit voluptates, quibusdam officia. Neque quibusdam quas a quis porro? Molestias illo neque eum in laborum.

Figure 9: The details page (server running)

When item exits in the cart, add to cart button and remove from cart buttons are displayed. But when that particular item doesn’t exist in the cart only the add to cart button is displayed. When the user adds an item to cart, it add the item into cart or if the item existed in the cart the quantity will be added again.

4.2.1 AUTHETICATION


Wed 17:30
localhost:8000/accounts/signup/
Login
Signup
SIGN UP »
localhost:8000/accounts/login/

Sign Up

Already have an account? Then please sign in.

Username*

Username

E-mail (optional)

E-mail address

Password*

Password

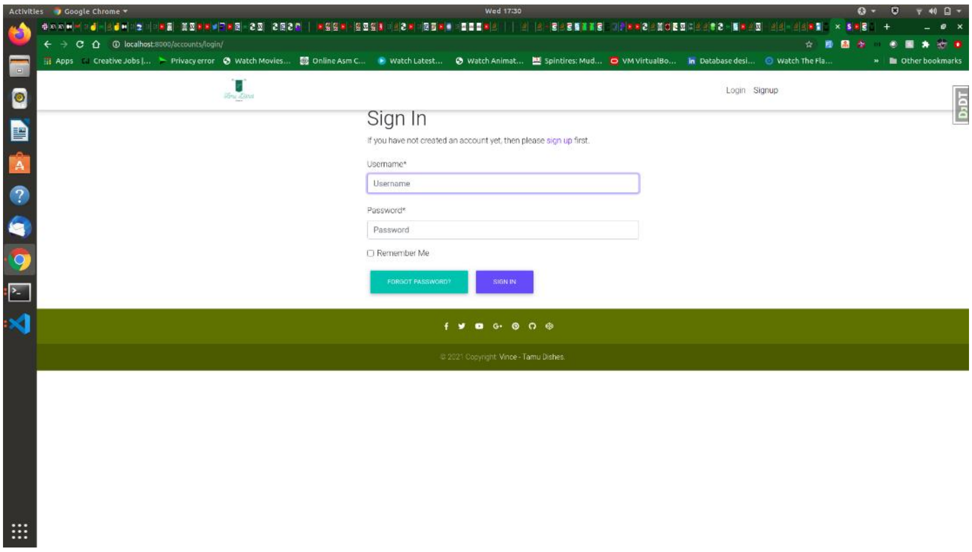
Password (again)*

Password (again)

© 2021 Copyright: Vince - Tamu Dishes.

Figure 10: Sign Up page (server running)

A user can create an account if they don’t have by going to the sign up page this will enable them get access into the system and make transactions. Django-allauth is handling this process securely.



Sign In

If you have not created an account yet, then please sign up first.

Username*

Username

Password*

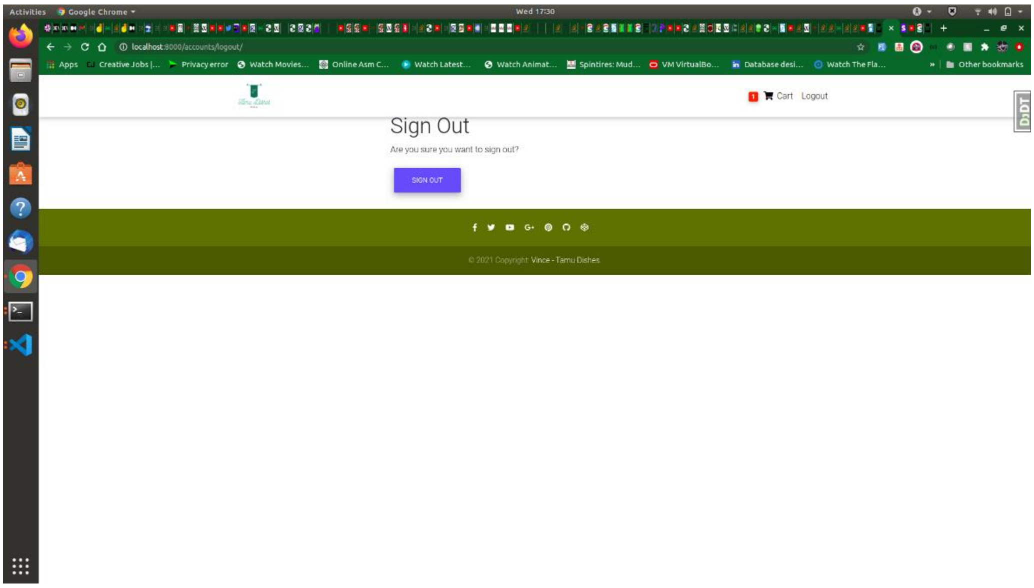
Password

□ Remember Me

© 2021 Copyright: Vince - Tamu Dishes

Figure 11: Sign In page (server running)

This where the user signs in before ordering their food from the system. The only action that is not working right now is the forgot password feature which needs the web application to be hosted. But it is a feature that will work once the site has been hosted.



Sign Out

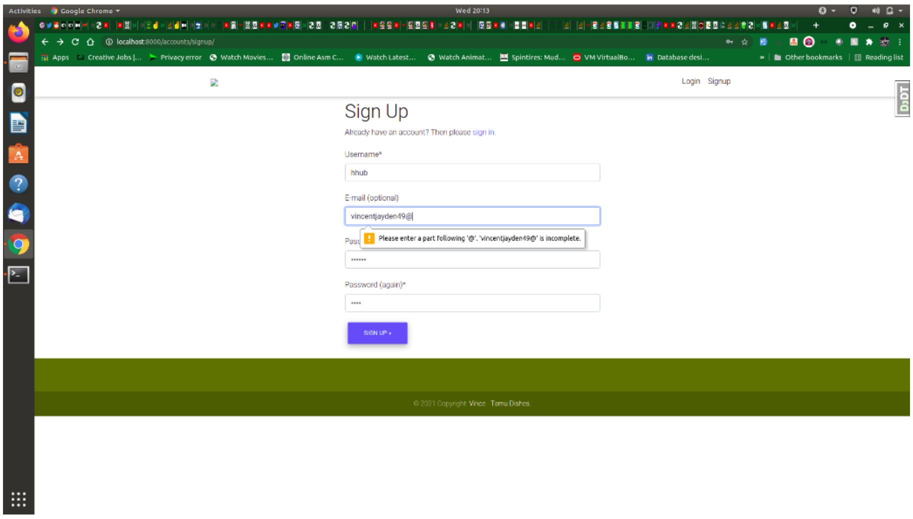
Are you sure you want to sign out?

© 2021 Copyright: Vince - Tamu Dishes

Figure 12: Logout (server running)

A user can successfully log out from the system.

4.2.2 AUTHENTICATION ERRORS FOR CLIENTS



Sign Up

Already have an account? Then please sign in.

Username*

hhub

E-mail (optional)

vincentjayden49@

Please enter a part following '@'. 'vincentjayden49@' is incomplete.

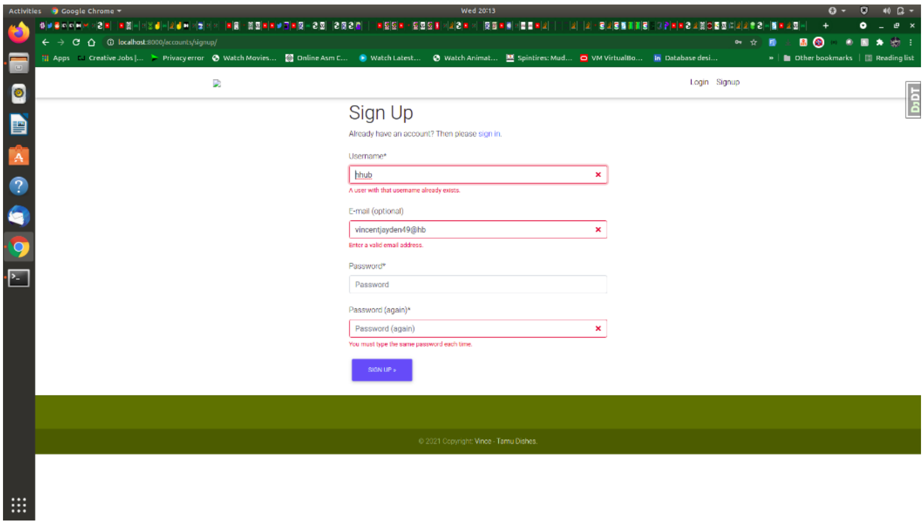
******

Password (again)*

****

SIGN UP »

© 2021 Copyright: Vince - Tamu Dishes.



Sign Up

Already have an account? Then please sign in.

Username*

hhub

A user with that username already exists.

E-mail (optional)

vincentjayden49@hb

Enter a valid email address.

Password*

Password

Password (again)*

Password (again)

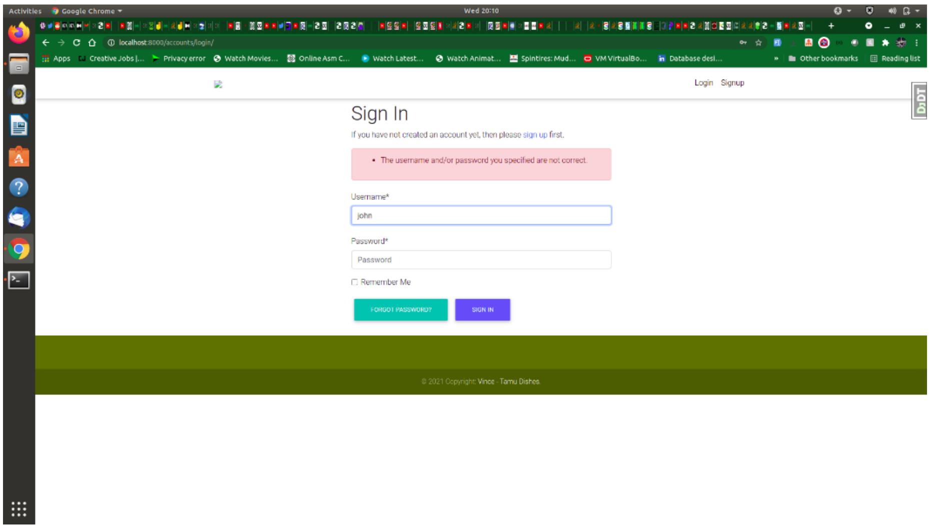
You must type the same password each time.

SIGN UP »

© 2021 Copyright: Vince - Tamu Dishes.

Figure 13 & 14: Sign up validation (server running)

During sign up, the data that the user provides gets validated. In the case the user enters invalid data, the system throws an error stressing where and what the problem is. So, the system ensures security.



Sign In

If you have not created an account yet, then please sign up first.

- The username and/or password you specified are not correct.

Username*

john

Password*

Password

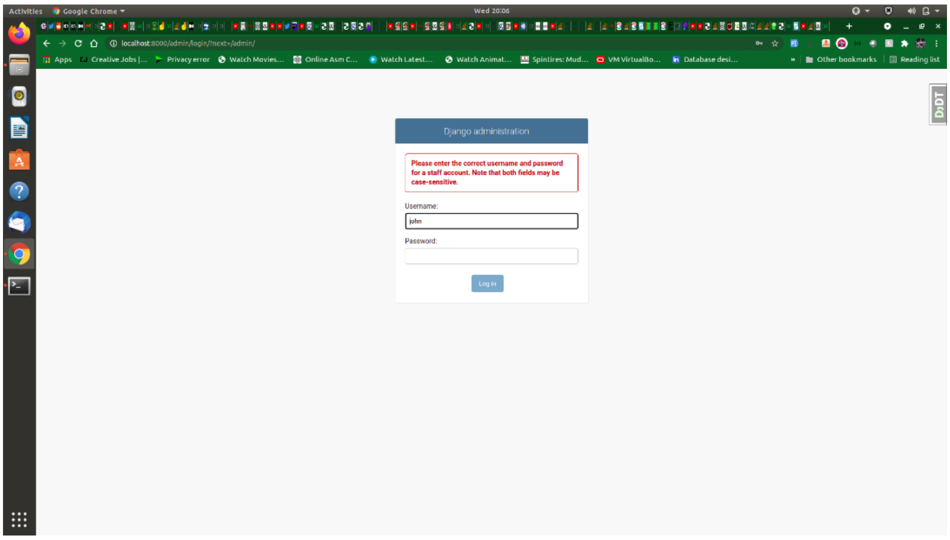
□ Remember Me

FORGOT PASSWORD? SIGN IN

© 2021 Copyright: Vince - Tamu Dishes.

Figure 15: Sign in validation (server running)

There is a login validation as well. When the user enters an invalid password or username the program throws an error and prompts the user to input valid details.



Django administration

Please enter the correct username and password for a staff account. Note that both fields may be case-sensitive.

Username:

john

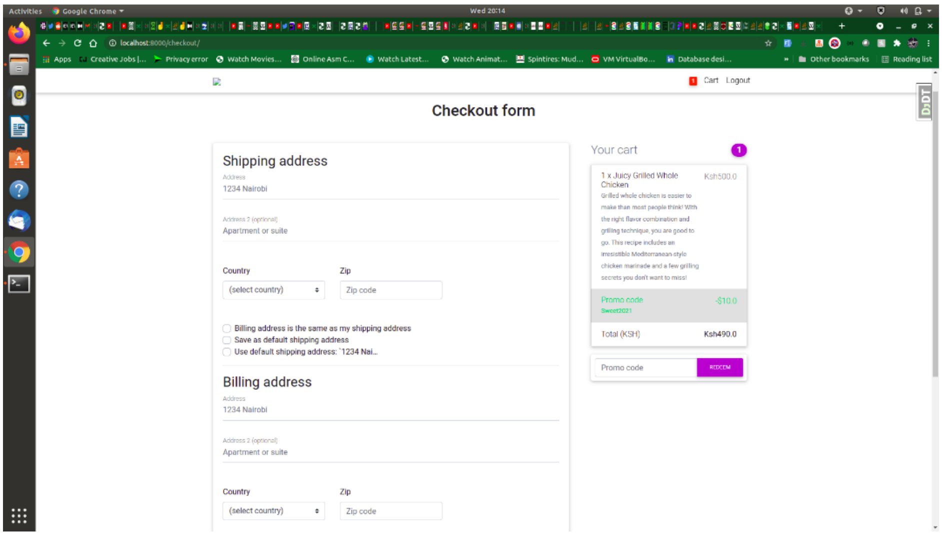
Password:

Log in

Figure 16: Sign in validation for the admin (server running)

The admin dashboard also has a solid validation only the staff can sign in.

4.2.3 USER BILLING INFO



Checkout form

Shipping address

Address
1234 Nairobi

Address 2 (optional)
Apartment or suite

Country
(select country)

Zip
Zip code

□ Billing address is the same as my shipping address
□ Save as default shipping address
□ Use default shipping address: `1234 Nai...

Billing address

Address
1234 Nairobi

Address 2 (optional)
Apartment or suite

Country
(select country)

Zip
Zip code

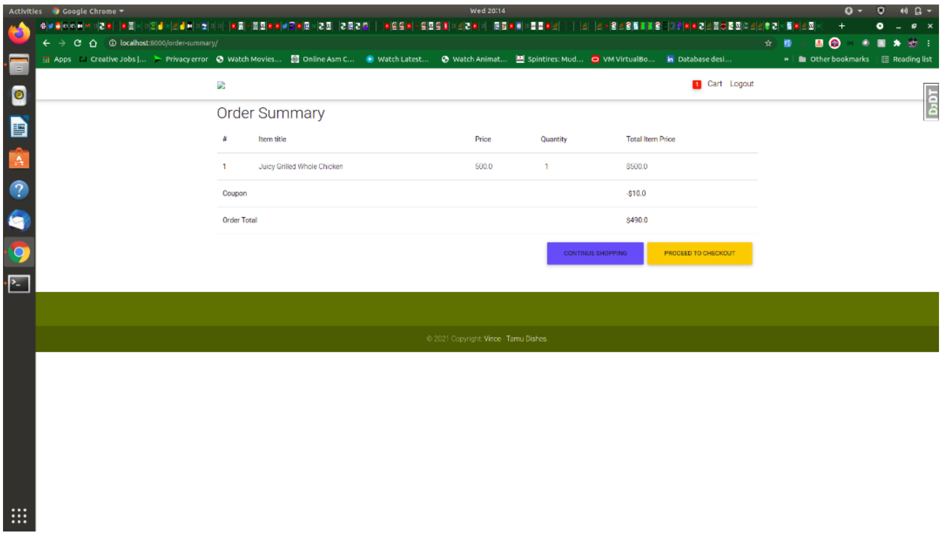
Your cart 1

1 x Juicy Grilled Whole Chicken Grilled whole chicken is easier to make than most people think! With the right flavor combination and grilling technique, you are good to go. This recipe includes an irresistible Mediterranean-style chicken marinade and a few grilling secrets you don't want to miss!	Ksh500.0
Promo code Sweet2021	-$10.0
Total (KSH)	Ksh490.0

Promo code

Figure 17: Checkout form (server running)

This is the checkout form where clients input their information for logistics purposes. It takes in the billing address and shipping address and they get stored into the database.



Order Summary

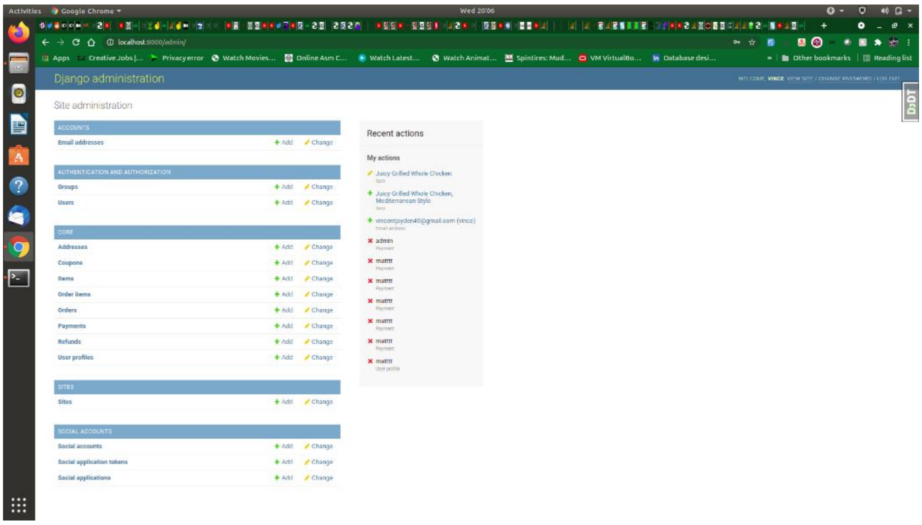
#	Item title	Price	Quantity	Total Item Price
1	Juicy Grilled Whole Chicken	500.0	1	$500.0
Coupon				-$10.0
Order Total				$490.0

© 2021 Copyright: Vince · Tamu Dishes.

Figure 18: Order Summary (server running)

The order summery contains the quantity, price and total price of the items in the user's cart for purchase. When the client is ready to order he/she proceeds to checkout.

4.2.4 ADMINISTRATION PANEL



Django administration

Site administration

ACCOUNTS		
Email addresses	+ Add	Change

AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
Users	+ Add	Change

CORE		
Addresses	+ Add	Change
Coupons	+ Add	Change
Items	+ Add	Change
Order items	+ Add	Change
Orders	+ Add	Change
Payments	+ Add	Change
Refunds	+ Add	Change
User profiles	+ Add	Change

SITES		
Sites	+ Add	Change

SOCIAL ACCOUNTS		
Social accounts	+ Add	Change
Social application tokens	+ Add	Change
Social applications	+ Add	Change

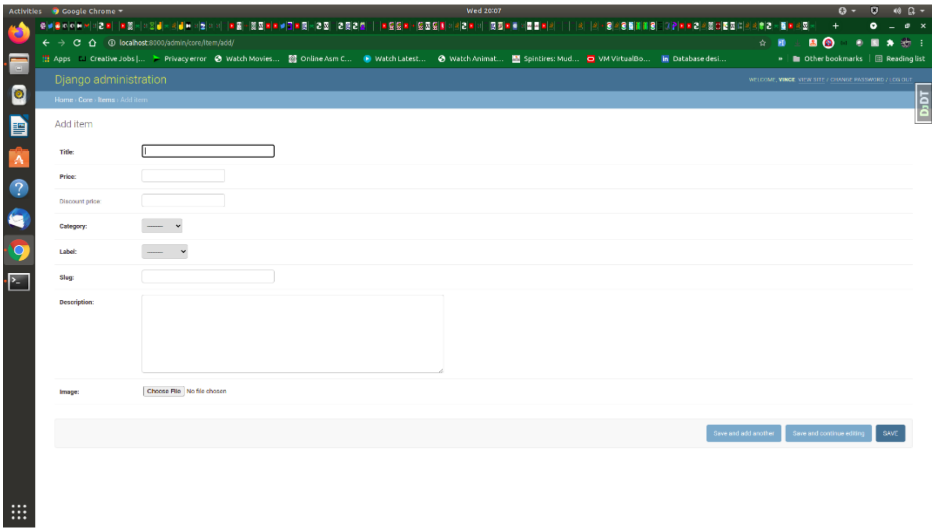
Recent actions

My actions

- Juicy Grilled Whole Chicken
  Item
- Juicy Grilled Whole Chicken, Mediterranean Style
  Item
- vincentjoydon46@gmail.com (vince)
  Email address
- admin
  Payment
- matttt
  Payment
- matttt
  Payment
- matttt
  Payment
- matttt
  Payment
- matttt
  Payment
- matttt
  User profile

Figure 19: Administration panel (server running)

This where the seller controls and carries most activities on regarding the system. The seller can create, update, read or delete information from the system.



Django administration

Home › Core › Items › Add item

Add item

Title:	
Price:	
Discount price:	
Category:	---------
Label:	---------
Slug:	
Description:	
Image:	Choose File No file chosen

Save and add another | Save and continue editing | SAVE

Figure 20: Adding an item for listing (server running)

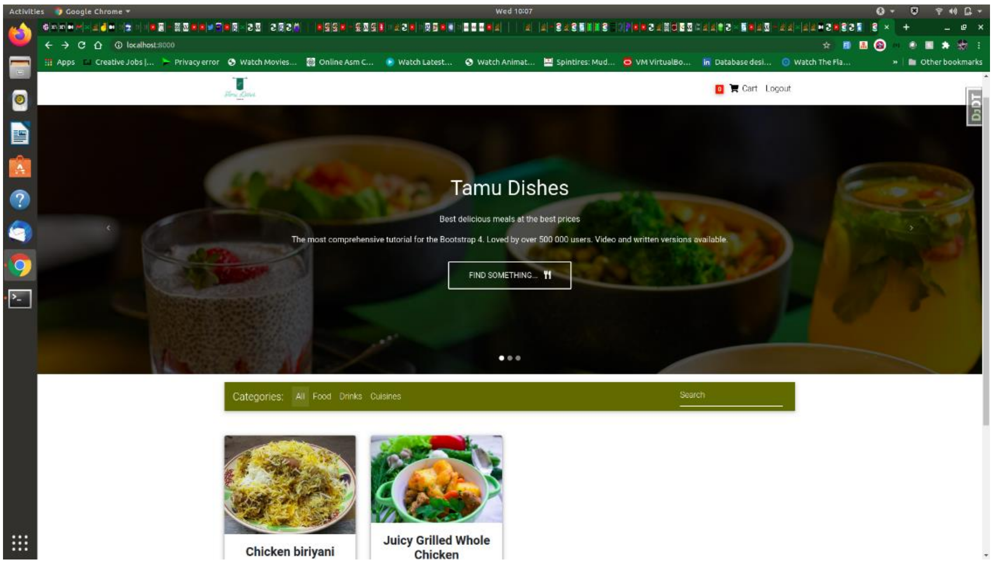
This is the panel where the seller adds information for listing on the front end. Once the item is added is appears to the client side.

CHAPTER 5: CONCLUSION, EVALUATION AND FURTHER WORK

5.1 OBJECTIVES

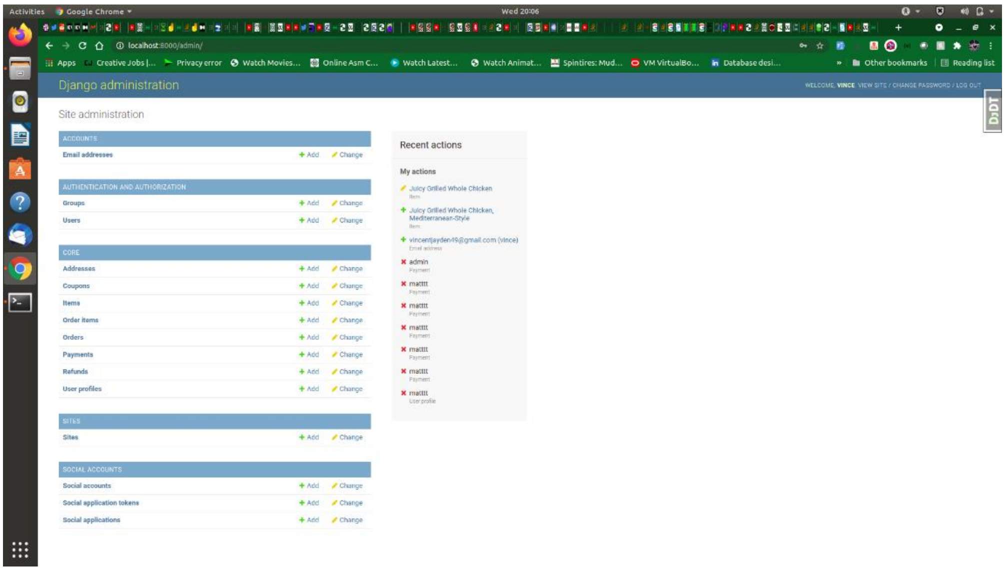
5.1.1 A SINGLE NODE (VIRUAL RESTAURANT) FOR INDIVIDUALS OR RESTAURANTS.

The completed product allows the individual chefs or restaurants to sell their cuisines to customers through the platform. They have a place to render their menu along with an administration dashboard to manage everything they have in the system.



Best delicious meals at the best prices

The most comprehensive tutorial for the Bootstrap 4. Loved by over 500 000 users. Video and written versions available.



Site administration

ACCOUNTS		
Email addresses	+ Add	Change
AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
Users	+ Add	Change
CORE		
Addresses	+ Add	Change
Coupons	+ Add	Change
Items	+ Add	Change
Order items	+ Add	Change
Orders	+ Add	Change
Payments	+ Add	Change
Refunds	+ Add	Change
User profiles	+ Add	Change
SITES		
Sites	+ Add	Change
SOCIAL ACCOUNTS		
Social accounts	+ Add	Change
Social application tokens	+ Add	Change
Social applications	+ Add	Change

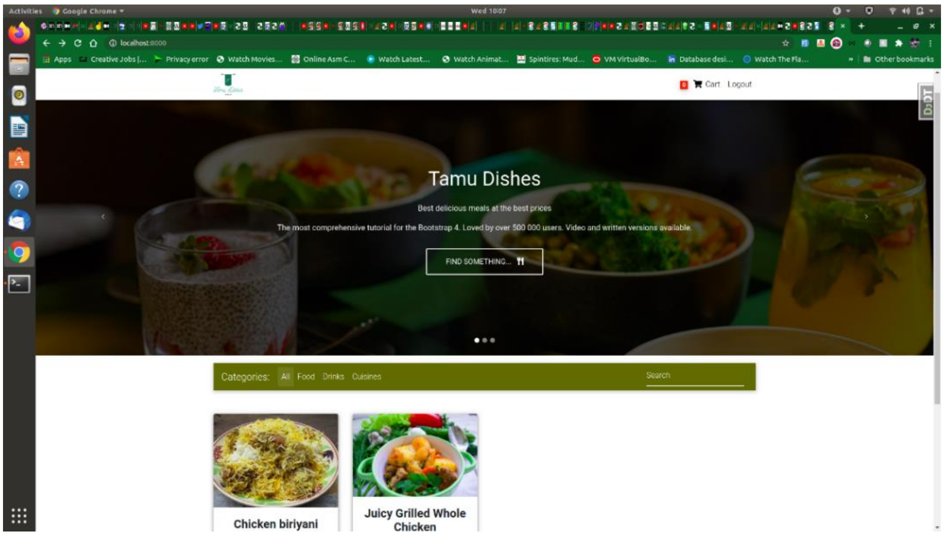
Recent actions

My actions

- Juicy Grilled Whole Chicken — Item
- Juicy Grilled Whole Chicken, Mediterranean-Style — Item
- vincentjayden49@gmail.com (vince) — Email address
- admin — Payment
- mattttt — Payment
- mattttt — Payment
- mattttt — Payment
- mattttt — Payment
- mattttt — Payment
- mattttt — User profile

5.1.2 SIMPLE, EASY AND GOOD USER EXPERIENCE

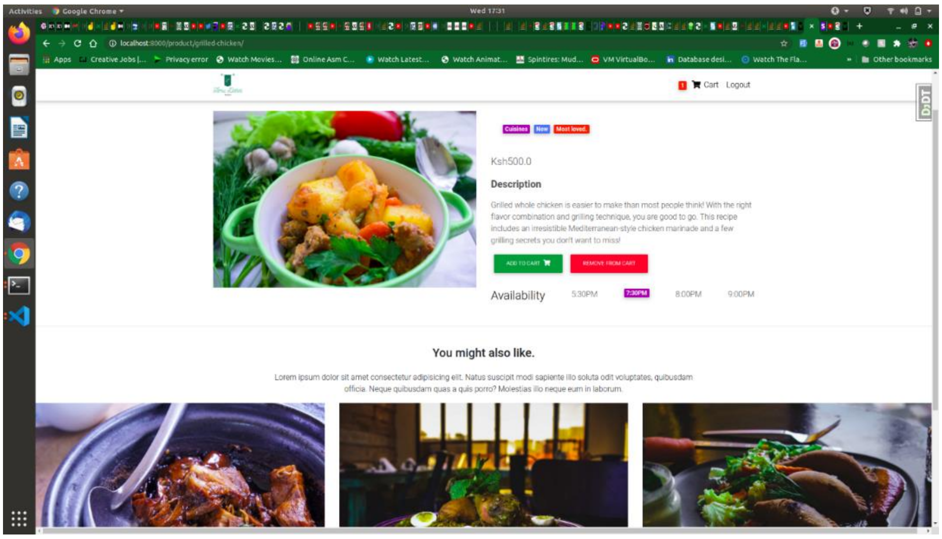
The system is easy to use as anybody with a computer or a mobile phone and a decent literacy can use the platform. The users can post, delete, order, pay and so much more in the system.



Tamu Dishes

Best delicious meals at the best prices

The most comprehensive tutorial for the Bootstrap 4. Loved by over 500 000 users. Video and written versions available.



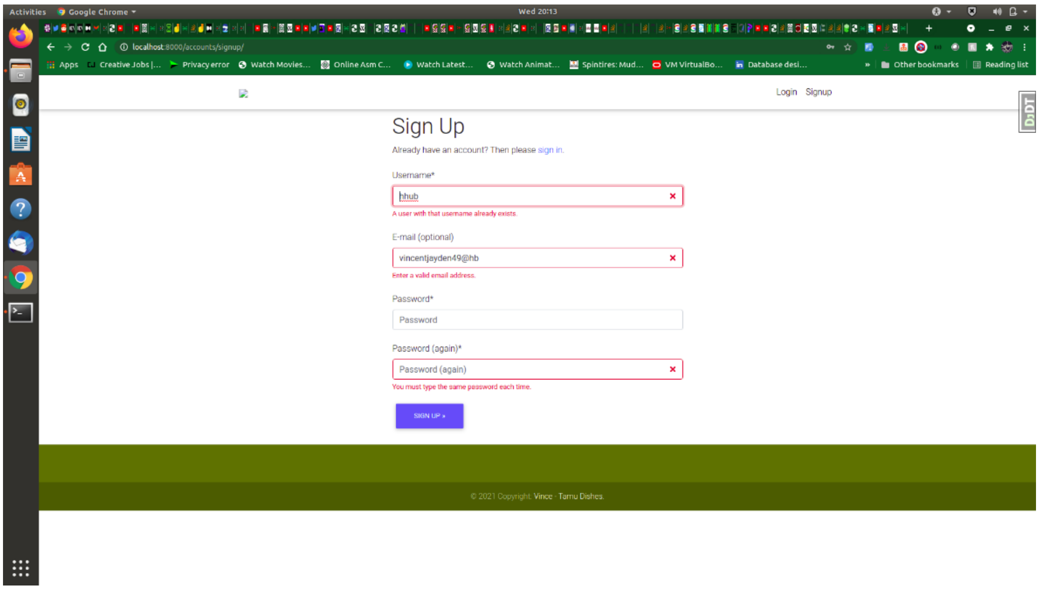
Ksh500.0

Description

Grilled whole chicken is easier to make than most people think! With the right flavor combination and grilling technique, you are good to go. This recipe includes an irresistible Mediterranean-style chicken marinade and a few grilling secrets you don't want to miss!

You might also like.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Natus suscipit modi sapiente illo soluta odit voluptates, quibusdam officia. Neque quibusdam quas a quis porro? Molestias illo neque eum in laborum.



Sign Up

Already have an account? Then please sign in.

Username*

hhub

A user with that username already exists.

E-mail (optional)

vincentjayden49@hb

Enter a valid email address.

Password*

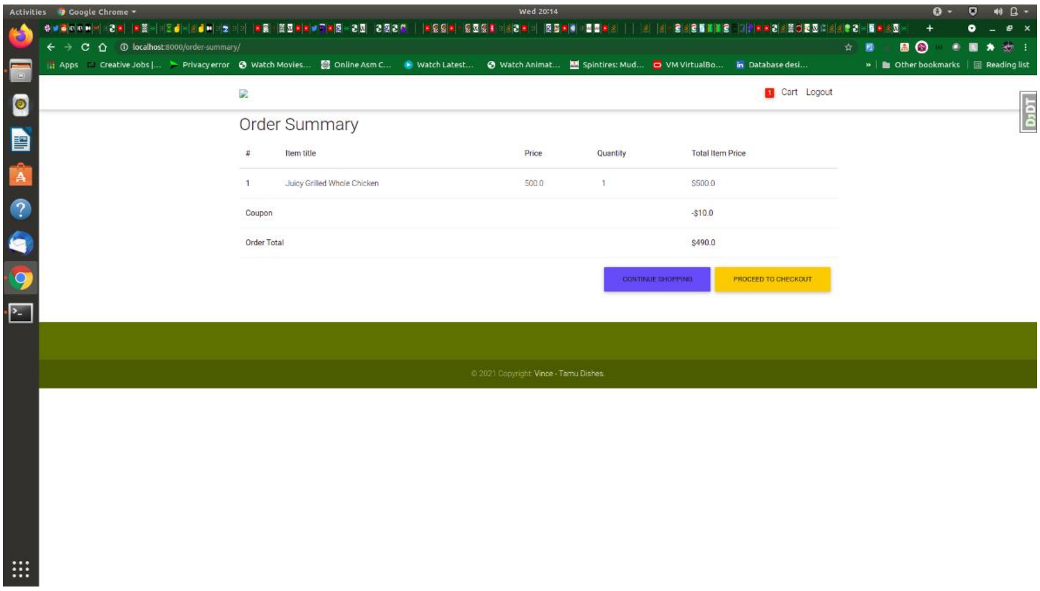
Password

Password (again)*

Password (again)

You must type the same password each time.

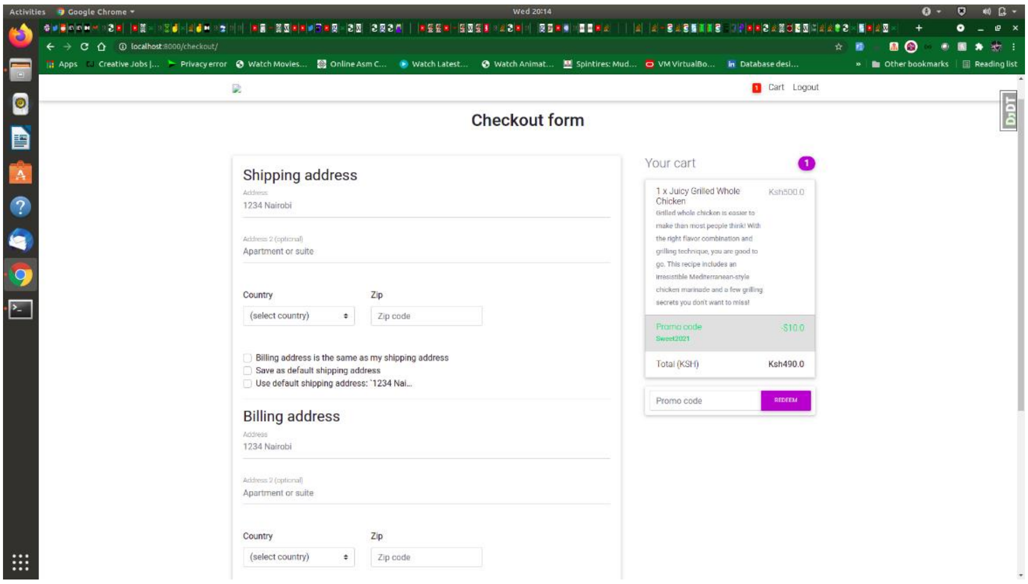
© 2021 Copyright: Vince - Tamu Dishes.



Order Summary

#	Item title	Price	Quantity	Total Item Price
1	Juicy Grilled Whole Chicken	500.0	1	$500.0
Coupon				-$10.0
Order Total				$490.0

© 2021 Copyright: Vince - Tamu Dishes.



Checkout form

Shipping address

Address
1234 Nairobi

Address 2 (optional)
Apartment or suite

Country
(select country)

Zip
Zip code

□ Billing address is the same as my shipping address
□ Save as default shipping address
□ Use default shipping address: `1234 Nai...

Billing address

Address
1234 Nairobi

Address 2 (optional)
Apartment or suite

Country
(select country)

Zip
Zip code

Your cart 1

1 x Juicy Grilled Whole Chicken Ksh500.0
Grilled whole chicken is easier to make than most people think! With the right flavor combination and grilling technique, you are good to go. This recipe includes an irresistible Mediterranean-style chicken marinade and a few grilling secrets you don't want to miss!

Promo code -$10.0
Sweet2021

Total (KSH) Ksh490.0

Promo code REDEEM

CLIENTS 5.1.3 CHOOSING RESTAURANTS

This part of the system was un-attainable due to its complexity and much development time. This is also because I started with the node (single restaurant) prototype which was the core of the project and its full functionality was crucial for the project. The building of the restaurant choosing page is in the future project expansion plan as it is the secondary and equally very important feature of the platform.

5.2 CONCLUSION

In this chapter, the synopsis of the entire project will be drafted with evaluation and further improvements and feedbacks that should be used to enhance the system.

The project was to develop an Ecommerce system that lets chefs and restaurants sell cuisines and their signature dishes to clients. The system will not allow fast foods to be sold. The system allows the clients have access to best dishes that they can get right on their doorsteps. The system allows anybody who can cook good food or cuisine start a food business from their homes or restaurants. This means the system will create so many businesses for people and more jobs.

5.3 EVALUATION

This project was to serve restaurants, private food businesses and chefs be able to sell their cuisines, foods, drinks and beverages to clients. Based on its functionalities and features. This project will be a success and it will help so many people have an income and also in turn, clients will have a central place for cuisines and better food.

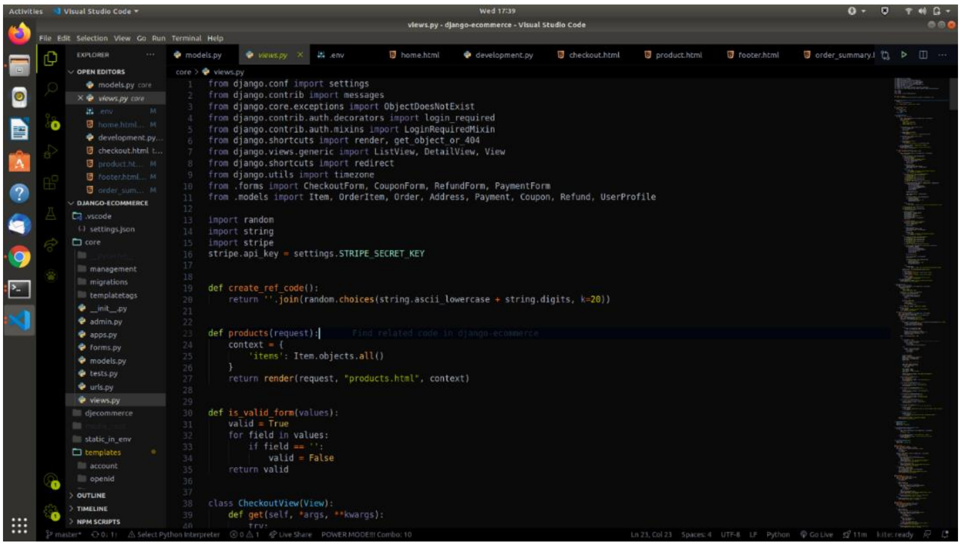
5.4 FURTHER IMPROVEMENTS

The project uses agile methodology during its development phase. This kind of development life cycle which will allow changes to be carried out progressively. For that reason, the system will be improved constantly in order to provide better and easy user experience. More useful features will also be added like for example adding a centralized place for all the restaurants where clients can choose which restaurant they want to buy from. Some of improvements that I'm looking at are: -

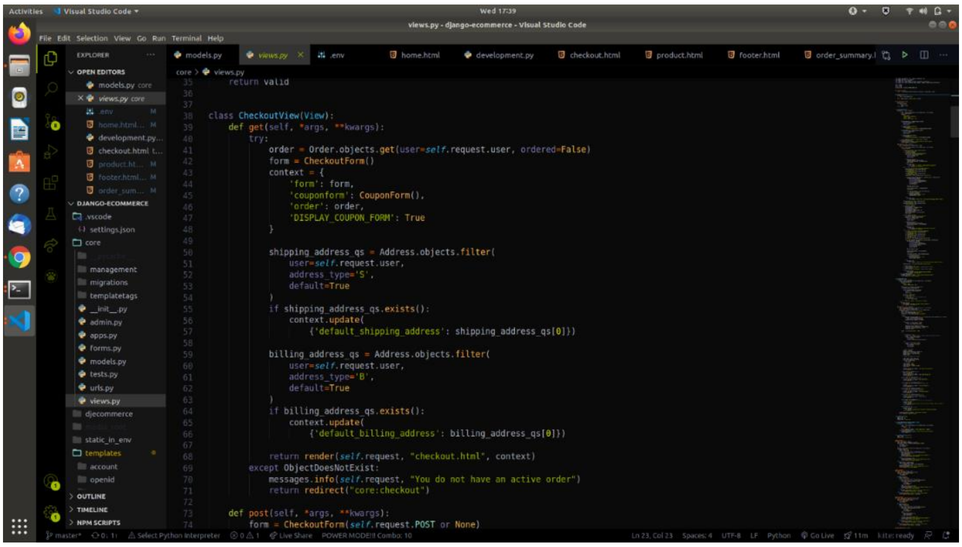
- Adding more payment options.
- Adding a centralized platform to list all the unit restaurants where users can choose the restaurant to buy from.
- Adding an account generator for unit both admins and units.
- Adding more sign in with other accounts with google.

APPENDICES

APPENDIX I: Views Sample Code.

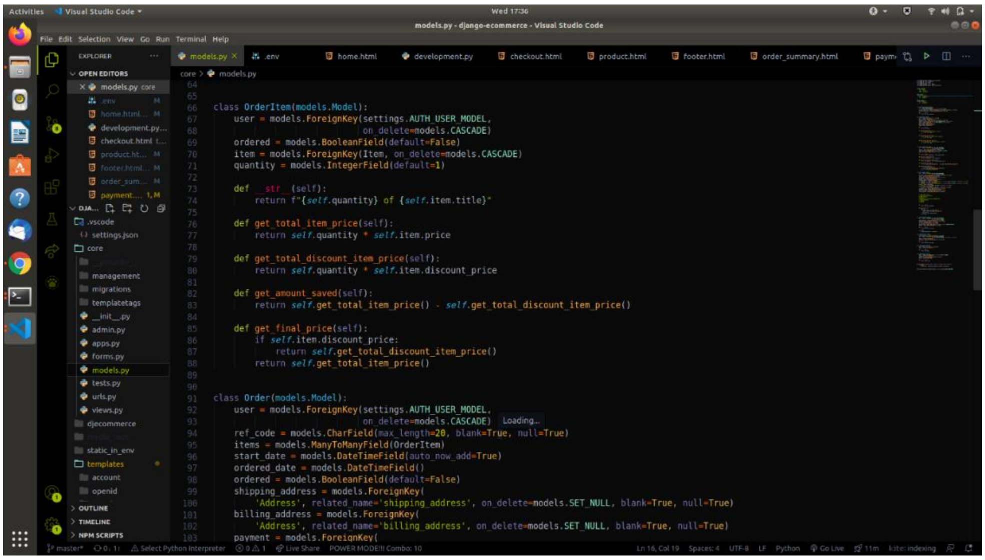


```
from django.conf import settings
from django.contrib import messages
from django.core.exceptions import ObjectDoesNotExist
from django.contrib.auth.decorators import login_required
from django.contrib.auth.mixins import LoginRequiredMixin
from django.shortcuts import render, get_object_or_404
from django.views.generic import ListView, DetailView, View
from django.shortcuts import redirect
from django.utils import timezone
from .forms import CheckoutForm, CouponForm, RefundForm, PaymentForm
from .models import Item, OrderItem, Order, Address, Payment, Coupon, Refund, UserProfile

import random
import string
import stripe
stripe.api_key = settings.STRIPE_SECRET_KEY


def create_ref_code():
    return ''.join(random.choices(string.ascii_lowercase + string.digits, k=20))


def products(request):
    context = {
        'items': Item.objects.all()
    }
    return render(request, "products.html", context)


def is_valid_form(values):
    valid = True
    for field in values:
        if field == '':
            valid = False
    return valid


class CheckoutView(View):
    def get(self, *args, **kwargs):
        try:
```



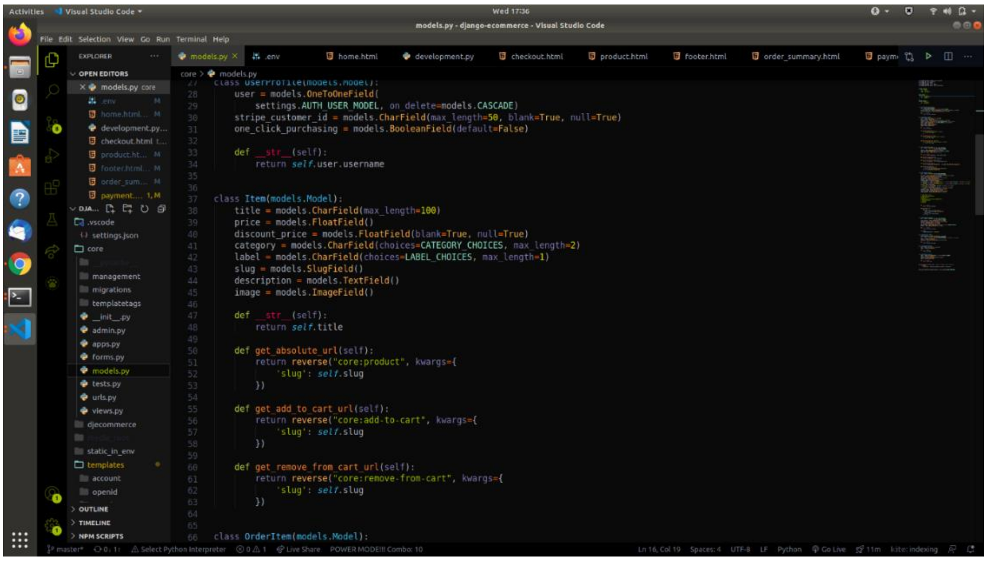
```
    return valid


class CheckoutView(View):
    def get(self, *args, **kwargs):
        try:
            order = Order.objects.get(user=self.request.user, ordered=False)
            form = CheckoutForm()
            context = {
                'form': form,
                'couponform': CouponForm(),
                'order': order,
                'DISPLAY_COUPON_FORM': True
            }

            shipping_address_qs = Address.objects.filter(
                user=self.request.user,
                address_type='S',
                default=True
            )
            if shipping_address_qs.exists():
                context.update(
                    {'default_shipping_address': shipping_address_qs[0]})

            billing_address_qs = Address.objects.filter(
                user=self.request.user,
                address_type='B',
                default=True
            )
            if billing_address_qs.exists():
                context.update(
                    {'default_billing_address': billing_address_qs[0]})

            return render(self.request, "checkout.html", context)
        except ObjectDoesNotExist:
            messages.info(self.request, "You do not have an active order")
            return redirect("core:checkout")

    def post(self, *args, **kwargs):
        form = CheckoutForm(self.request.POST or None)
```

APPENDIX II: Model (database schema) Sample Source Code.



```python
class OrderItem(models.Model):
    user = models.ForeignKey(settings.AUTH_USER_MODEL,
                             on_delete=models.CASCADE)
    ordered = models.BooleanField(default=False)
    item = models.ForeignKey(Item, on_delete=models.CASCADE)
    quantity = models.IntegerField(default=1)

    def __str__(self):
        return f"{self.quantity} of {self.item.title}"

    def get_total_item_price(self):
        return self.quantity * self.item.price

    def get_total_discount_item_price(self):
        return self.quantity * self.item.discount_price

    def get_amount_saved(self):
        return self.get_total_item_price() - self.get_total_discount_item_price()

    def get_final_price(self):
        if self.item.discount_price:
            return self.get_total_discount_item_price()
        return self.get_total_item_price()


class Order(models.Model):
    user = models.ForeignKey(settings.AUTH_USER_MODEL,
                             on_delete=models.CASCADE)
    ref_code = models.CharField(max_length=20, blank=True, null=True)
    items = models.ManyToManyField(OrderItem)
    start_date = models.DateTimeField(auto_now_add=True)
    ordered_date = models.DateTimeField()
    ordered = models.BooleanField(default=False)
    shipping_address = models.ForeignKey(
        'Address', related_name='shipping_address', on_delete=models.SET_NULL, blank=True, null=True)
    billing_address = models.ForeignKey(
        'Address', related_name='billing_address', on_delete=models.SET_NULL, blank=True, null=True)
    payment = models.ForeignKey(
```

APPENDIX III: Profile Sample Source Code



```python
class UserProfile(models.Model):
    user = models.OneToOneField(
        settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    stripe_customer_id = models.CharField(max_length=50, blank=True, null=True)
    one_click_purchasing = models.BooleanField(default=False)

    def __str__(self):
        return self.user.username


class Item(models.Model):
    title = models.CharField(max_length=100)
    price = models.FloatField()
    discount_price = models.FloatField(blank=True, null=True)
    category = models.CharField(choices=CATEGORY_CHOICES, max_length=2)
    label = models.CharField(choices=LABEL_CHOICES, max_length=1)
    slug = models.SlugField()
    description = models.TextField()
    image = models.ImageField()

    def __str__(self):
        return self.title

    def get_absolute_url(self):
        return reverse("core:product", kwargs={
            'slug': self.slug
        })

    def get_add_to_cart_url(self):
        return reverse("core:add-to-cart", kwargs={
            'slug': self.slug
        })

    def get_remove_from_cart_url(self):
        return reverse("core:remove-from-cart", kwargs={
            'slug': self.slug
        })


class OrderItem(models.Model):
```

REFERENCES

P.K Kabuba. (2014). Ecommerce and Internet: E-commerce and performance of online businesses in Kenya. Retrieved (November, 2014), from: http://erepository.uonbi.ac.ke/handle/11295/77075

Ben Maina. (2020). Retail and E-commerce statistics: State of Retail & Grocery E-commerce in Kenya. Retrieved (April, 2020), from: https://www.linkedin.com/pulse/state-retail-grocery-e-commerce-kenya-ben-maina/

Abdullah Hasan. (2021). Entity Relationship Diagram: e-Commerce Database. Retrieved (May, 2021), from: https://creately.com/diagram/example/he7cxejx1/e-Commerce%20Database

Thandi Guilherme. (2018). Agile Process: I bet your Agile process is broken. Retrieved (November, 2018), from:

https://medium.com/@thandi.guilherme/i-bet-your-agile-process-is-broken-32985716233f#:~:text=The%20reason%20Agile%20is%20so,you're%20building%E2%80%9D%20fluid.