E-COMMERCE WEBSITE USING DJANGO FRAMEWORK

A Project

Presented to the

Faculty of

California State Polytechnic University, Pomona

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science

In

Computer Science

By

Ashok Chakravarthi Suryadevara

2022

SIGNATURE PAGE

PROJECT:	E-COMMERCE WEBSITE USING DJANGO FRAMEWORK
AUTHOR:	Ashok Chakravarthi Suryadevara
DATE SUBMITTED:	Fall 2022
	Computer Science Department

Dr. Gilbert S. Young
Project Committee Chair
Computer Science

_____

Dr. Yu Sun
Computer Science

_____

ACKNOWLEDGEMENTS

I want to thank God first and foremost for gifting me with such wonderful friends who have helped me out in everything I have done. I love you, to my parents, who have given me so much love and joy in my life. I'd like to express my gratitude to my family and friends for helping me develop into the person I am today. I want to thank all my former Cal Poly Pomona professors, especially Dr. Young, for their support and encouragement in getting me through this project. I want to express my sincere gratitude to you all for supporting me as I achieve yet another life goal.

ABSTRACT

Electronic commerce, or e-commerce, is a sector of the economy where the purchasing and selling of goods is done through electronic channels like the internet. To give consumers immediate access to our products in today's quickly evolving business climate, we must be more efficient and quicker in our response times. This can be achieved by creating an online shopping E-commerce web application that allows clients to purchase a variety of clothing anditems and make payments instantly or after delivery. Websites are used by many corporate organizations to do business. This makes online shopping convenient and e-commerce a widely accepted paradigm. A virtual online store that enables clients to look for products andchoose them from a catalog is required to implement online shopping. To order a certain product, the customer must fill out various forms. The creation and operation of an online website is the goal of this project. It is necessary to study and comprehend server and client approaches, the relational databases, and numerous programming languages such as HTML, CSS, JAVASCRIPT, and PYTHON to create and develop this e-commerce retail website.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

E-commerce is quickly becoming a recognized and utilized business paradigm. More and more companies are putting in place websites with capability for conducting business transactions online. It is safe to argue that shopping online has become routine. The goal of this project is to create a general-purpose online store where people may shop for items like clothing, Shoes, Electronics etc. while lounging at home. Logically, a consumer finds an item more intriguing and appealing when they can view the item's details online and find it on the retailer's website. Customers of today are drawn to online shopping not only because it is very easy, but also because there are more options, prices are very competitive, there is greater product information, and the navigation for searching for products is very straightforward.

In addition, business owners frequently provide low-cost online purchasing choices due to the higher overhead costs associated with creating and operating a physical store. Additionally, the products have access to a global market thanks to online purchasing, which boosts the number of clients from various ethnic backgrounds, increases customer value, and makes marketing more generally viable [4]. A virtual store on the Internet where clients can browse the inventory and choose items of interest is known as an online store. The chosen items could be put in a shopping basket. The goods in the shopping cart will be shown as an order when it is time to check out. At that point, more details will be required to finalize the purchase. The customer will typically be required to provide or choose a billing address, a mailing address, a delivery option, and payment details like a credit card number or payment gateway.

CHAPTER 2

LITERATURE SURVEY – WEB DEVELOPMENT

The process of creating a website for the Internet (World Wide Web) or an intranet is referred to as web development (a private network). The smallest static single-page of plain text to the most complex web-based internet apps, electronic enterprises, and social network services can all be created through the process of web development. Web engineering, web design, web content creation, client liaison, client-side/side scripting, web server and network security settings, and e-commerce development may all be included in a more extensive list of duties that are often referred to as web development. "Web development" is a term used by web experts to describe the two main non-design components of creating websites: writing markup and coding. The term "web development" has most recently come to refer to the development of content management systems, or CMS. These CMSs can be custom-built, closed source, or proprietary. In general, the CMS serves as a middleman between the user and the database via the browser. A key advantage of a CMS is that it enables non-technical users to edit their website without any prior technical knowledge [3].

Teams of web developers for larger enterprises and organizations may number in the hundreds and create websites using standardized techniques like Agile processes. Smaller businesses could simply need a single permanent or contract developer, or they might assign additional personnel to similar roles like graphic designer or information systems technician. Instead of being the purview of a single department, web development may involve departmental collaboration.

2.1 WEB-SITE

A website is a collection of linked online pages, often with multimedia material, that are published on at least one web server and often have a shared domain name. By referring to a unified resource locator (URL) that identifies the site, a website may be accessed over a private local area network (LAN) or a public Internet Protocol (IP) network, such as the Internet. Websites can be used in many ways and serve a variety of purposes; they can be personal websites, business websites for corporations, governmental websites, or websites for nonprofit organizations. Websites are often focused on a single subject or objective, ranging from social networking and entertainment to news and education. The World Wide Web is made up of all publicly accessible websites, whereas private websites, including those for a company's employees, are often a component of an intranet.

Web pages are documents that make up websites. They are commonly written in plain text with formatting directives from HTML interspersed (HTML, XHTML). They may use ten appropriate markup anchors to include content from other websites. The Hypertext Transfer Protocol (HTTP) is used to view and transfer web pages, and it can optionally use encryption (HTTP Secure, HTTPS) to protect user security and privacy. The user's application, which is frequently a web browser, displays the page content on a display terminal in accordance with the HTML markup instructions it receives [3].

Links between online pages help the reader understand the site's structure and orient their navigation, which frequently begins with the home page, which typically includes a list of the site's web content. On some websites, accessing material requires user registration or a membership. Many business websites, news websites, academic

journal websites, gaming websites, file-sharing websites, message boards, web-based email, social networking websites, websites offering real-time stock market data, as well as websites offering various other services, are examples of subscription websites.

2.2 WEB-PAGE

A document that is appropriate for the World Wide Web and web browsers is known as a web page, or webpage. An online page is shown via a web browser on a computer or mobile device. The phrase "web page" also refers to a computer file, typically created in HTML or another markup language. To render the written web page, web browsers coordinate the many online resource pieces, such as style sheets, scripts, and graphics.

Typical web sites link to other web pages via hypertext, often known as links, which includes a navigation bar or sidebar menu. A web browser on a network can get a web page from a distant web server. The web server may, at a higher level, limit access to only a private network, such as an intranet for a company, or it may offer access to the World Wide Web. Such queries are made by the web browser using HTTP, a more fundamental protocol. A dynamic web page is produced by a web application that is powered by server-side code or client-side scripting, whereas a static web page is delivered exactly as stored, as web content in the file system of the web server [1]. Using user input sent to the server, dynamic website pages enable the browser (the client) to improve the web page.

CHAPTER 3

PROJECT GOAL

The project involves creating an online store for customers to buy for various products. Additionally, it handles and manages resources that were previously handled and controlled by labor. The project's primary goal is to unite several shop areas in a unified way so that complicated tasks may be completed without difficulty online using the website. The following are the project's goals:

- Choose the methodology that will serve as the project's direction by understanding the advantages and disadvantages of the various website development techniques.
- Develop a completely functional website where customers may make purchases using Django Framework.

CHAPTER 4

METHODOLOGY

There are several steps involved in creating a website, including the ones listed below:

- Building a UI (User interface)
- Scripting (Both at server end and client end)
- Building a database or a backend

4.1 UI Development

Technologies that are mostly used to develop a User Interface are:

- HTML
- CSS
- Bootstrap

HTML:

The markup language used to create web pages and web applications is called Hypertext Markup Language (HTML). Together with JavaScript and Cascading Style Sheets (CSS), it makes up the trio of foundational technologies for the World Wide Web. Web browsers transform HTML documents into multimedia web pages after receiving them from a webserver or local storage. HTML originally featured cues for the document's design and semantically explains the structure of a web page. The foundation of HTML pages are HTML components. Images and other objects, including interactive forms, can be embedded into the produced page using HTML structures. By indicating structural semantics for text elements including headers, paragraphs, lists, links, quotes, and other objects, it offers a way to produce structured documents [6]. Tags, which are written in

angle brackets, are used to distinguish HTML elements. Input and image tags, for example, bring content into the page immediately. They may also contain other tags as sub- elements. Browsers employ the HTML tags to decipher the page's content rather than displaying them.

CSS:

The display of a text expressed in a markup language can be described using Cascading Style Sheets (CSS), a style sheet language. The language can be used to set the visual style of any XML document, including plain XML, SVG, and XUL, and is adaptable to rendering in voice or on other media, while being most frequently used to set the visual style of web pages and user interfaces written in HTML and XHTML. The majority of websites employ CSS, together with HTML and JavaScript, as a foundational technology to design visually appealing webpages, user interfaces for web apps, and user interfaces for many mobile applications.

The main purpose of CSS is to make it possible to separate presentation from content, including elements like fonts, colors, and layout. By specifying the pertinent CSS in a separate.css file, this separation can make the content more accessible, give the specification of presentation characteristics more flexibility and control, allow multiple HTML pages to share formatting, and reduce complexity and repetition in the structural content [6]. The presentation of the same HTML page in many styles for various rendering techniques, such as on-screen, in print, via voice (through speech-based browser or screen reader), and on Braille-based tactile devices, is made feasible by the separation of formatting and content. In addition, depending on the screen size or viewing device, it can show the web page differently.

To replace the one the author gave, readers can specify a new style sheet, such as one stored on their own computer. Instead of updating the markup in the documents, changes to the graphic design of a document (or hundreds of pages) can be made fast and easily by editing a few lines in the CSS file they utilize. If more than one style rule matches against a certain element, the CSS specification defines a priority method to decide which style rules should take precedence. Priorities (or weights) are determined and assigned to rules in this so-called cascade, making the outcomes predictable.

4.2 Server-side Scripting

To create responses that are unique for each user's (client's) request to the website, server-side scripting is a technique used in web development. The alternative is for a static web page to be delivered directly by the web server. Any of the available server-side scripting languages can be used to create scripts (see below). Client-side scripting, where embedded scripts like JavaScript are run client-side in a web browser, is distinct from server-side scripting, though both approaches are frequently combined [2].

To give users a personalized interface, server-side scripting is frequently employed. In order to tailor the response based on the client's characteristics, the user's requirements, access rights, etc., these scripts may compile the client's characteristics. In contrast to client-side scripting, which gives the user access to all the code received by the client, server-side scripting allows the website owner to conceal the source code that creates the interface. The usage of server- side scripting has the drawback of requiring further network requests from the client to the server for fresh information to be displayed to the user through the web browser. These queries can make the user's experience slower, put

additional strain on the server, and stop the application from working when the user loses connection to the server [2].

Users may have a variety of client programs to choose from when the server provides data in a widely used method, such as in accordance with the HTTP or FTP protocols (most modern web browsers can request and receive data using both of those protocols). Programmers may create their own server, client, and communications protocol for more specialized applications such that they can only be used in conjunction with one another. Programs that run entirely locally on a user's computer, never sending, or receiving data over a network, are not regarded as clients, and their operations are not regarded as client-side operations.

4.3 Client-Side Scripting

Client-side scripting is the process of altering the interface behaviors on a particular web page in response to mouse or keyboard inputs or at predetermined timing events. In this instance, the presentation itself exhibits the dynamic nature. The user's local computer system is where the client-side content is created. Rich interfaced pages, a presentation technique, are used on these websites. To choreograph the presentation's media kinds (sound, animations, changing text, etc.), client-side scripting languages like JavaScript or ActionScript, used for Dynamic HTML (DHTML) and Flash technologies respectively, are widely utilized. The usage of remote scripting, which involves the DHTML page requesting more data from a server using a hidden frame, XML Http Requests, or a Web service, is also made possible by client-side scripting. In 1997, after becoming standardized as ECMAScript and being integrated into Netscape 3, JavaScript saw its first significant application.

4.4 Database

An ordered collection of data is called a database. It is a collection of items, including tables, queries, reports, and views. The data are often set up to model features of reality in a way that helps information-intensive activities, such as modeling hotel room availability in a way that facilitates finding a hotel with open rooms.

In order to collect and process data, a database management system (DBMS) is a piece of computer software that communicates with the user, other programs, and the database itself. A general-purpose DBMS is made to make it possible to define, create, query, update, and manage databases. DBMSs that are well-known in the industry include MySQL, PostgreSQL, MongoDB, MariaDB, Microsoft SQL Server, Oracle, Sybase, SAP HANA, MySQL, and IBM DB2. A 21 database is often not transferable between DBMSs, but DBMSs can cooperate by leveraging standards like SQL, ODBC, or JDBC to enable a single application to operate with many DBMSs. The relational model as described by the SQL language has been supported by all the most widely used database management systems since the 1980s. Database management systems are frequently categorized according to the database model that they support. A DBMS is referred to as a "database" informally on occasion [5].

SQL:

SQL is made up of a data definition language, a data manipulation language, and a data control language. It was initially based on relational algebra and tuple relational calculus. Data access control, schema construction and change, data entry, query, update, and deletion are all included in the scope of SQL. Although SQL is sometimes referred to as a declarative language, which it largely is (4GL), it also contains procedural

components. In his influential 1970 work, "A Relational Model of Data for Large Shared Data Banks," Edgar F. Codd described one of the first commercial languages for his relational model, which was SQL. It became the most used database language despite not entirely adhering to Codd's relational paradigm [5].

In 1986, the American National Standards Institute (ANSI) and the International Organization for Standardization (ISO) recognized SQL as a standard. Since then, a wider range of functionality have been added to the standard. Despite the existence of such standards, the majority of SQL code requires modifications in order to function properly across various database systems.

4.5 Django Framework

A high-level Python web framework called Django enables the quick creation of safe and dependable websites. Django, which was created by seasoned programmers, handles a lot of the hassle associated with web development, allowing you to concentrate on developing your app without having to invent the wheel. It is open source and free, has a strong community, excellent documentation, and a variety of free and paid support options. The following are some of Django's features:

- Django adheres to the "batteries included" philosophy and provides nearly everything a developer might need "out of the box." Because everything you require is included in a single "product," it all works in unison and adheres to consistent design principles.
- Django can (and has been) used to create nearly any type of website, ranging from content management systems and wikis to social networks and news sites. It is compatible with any client-side framework and can deliver content in nearly any

format (including HTML, RSS feeds, JSON, and XML). Internally, while it offers options for almost any functionality you might require (e.g., several popular databases, templating engines, and so on), it can also be extended to use other components if necessary.

- Django assists developers in avoiding many common security mistakes by providing a framework that has been engineered to "do the right things" to automatically protect the website. Django, for example, provides a secure way to manage user accounts and passwords, avoiding common mistakes such as storing session information in cookies (instead, cookies only contain a key, and the actual data is stored in the database) or directly storing passwords rather than a password hash. A password hash is a fixed-length value that is generated by passing the password through a cryptographic hash function. Django can validate a password by running it through the hash function and comparing the output to the previously stored hash value.
- Django employs a "shared-nothing" component-based architecture. Because the different parts are clearly separated, it can scale for increased traffic by adding hardware at any level: caching servers, database servers, or application servers. Some of the most popular websites have successfully scaled Django to meet their needs.
- Django code is written in accordance with design principles and patterns that promote the creation of maintainable and reusable code. It specifically employs the Don't Repeat Yourself (DRY) principle to eliminate unnecessary duplication, thereby reducing the amount of code. Django also encourages the organization of related functionality into reusable "applications" and, at a lower level, the organization of related code into modules (along the Model View Controller (MVC) pattern).

- Django is written in Python and runs on a variety of platforms. This means you are not restricted to a single server platform and can run your applications on a variety of Linux, Windows, and macOS flavors. Furthermore, Django is well-supported by many web hosting providers, who frequently provide Django-specific infrastructure and documentation.

A web application in a traditional data-driven website waits for HTTP requests from the web browser (or another client). When a request is received, the application determines what is required based on the URL and possibly data in POST or GET data. Depending on what is needed, it may then read or write data from a database or perform other tasks to fulfill the request. The application will then respond to the web browser by inserting the retrieved data into placeholders in an HTML template and dynamically creating an HTML page for the browser to display.

Django web applications usually separate the code for each of these steps into separate files:

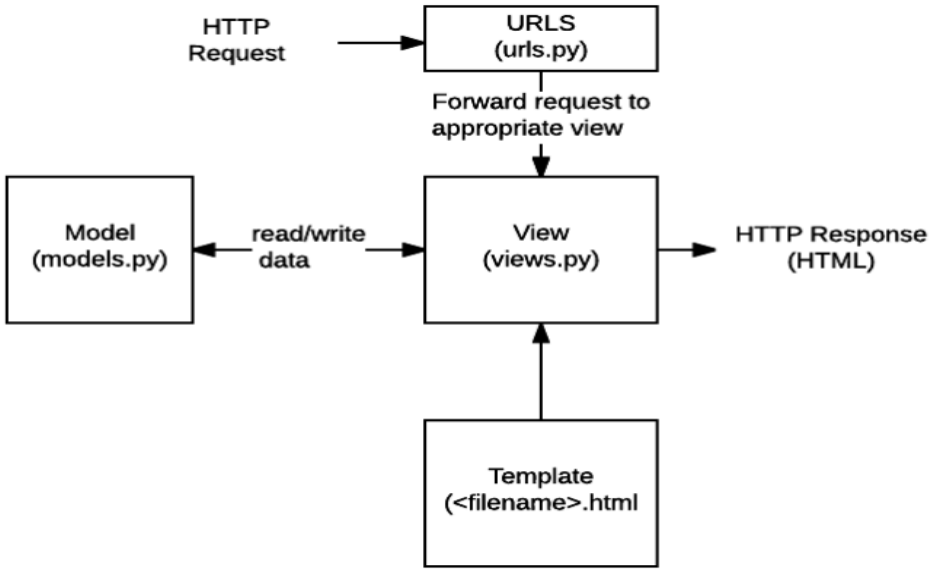

Figure 1: Django Network Architecture

- While it is possible to process requests from all URLs using a single function, it is far more maintainable to write a separate view function for each resource. A URL mapper is used to direct HTTP requests to the correct view based on the URL of the request. The URL mapper can also match specific patterns of strings or digits in a URL and pass them as data to a view function.
- A view is a request handler function that receives HTTP requests and responds to them. Views use models to access the data needed to satisfy requests and delegate response formatting to templates.
- Models are Python objects that define the structure of an application's data and provide mechanisms for managing (adding, modifying, deleting) and querying database records.
- Templates are text files that define the structure or layout of a file (such as an HTML page), with placeholders to represent actual content. A view can generate an HTML page dynamically by using an HTML template and populating it with data from a model. A template can be used to define the structure of any type of file; HTML is not required!

Models, which are Python objects, are used to manage and query data in Django web applications. Models define the structure of stored data, such as field types and, if applicable, their maximum size, default values, selection list options, help text for documentation, label text for forms, and so on. The model definition is independent of the underlying database — you can select one of several options as part of your project settings. Once you've decided on a database, you don't need to communicate with it directly — you just write your model structure and other code, and Django handles all the

"dirty work" of communicating with the database for you.

For searching the associated database, the Django model provides a simple query API. This can match against multiple fields at once using various criteria (e.g., exact, case-insensitive, greater than, and so on), and it can support complex statements (for example, you can specify a search on U11 teams that have a team name that starts with "Fr" or ends with "al"). Template systems let you define the structure of an output document by inserting placeholders for data that will be filled in when a page is generated. Templates are commonly used to generate HTML, but they can also generate other types of documents. Django comes with support for both its native templating system and another popular Python library called Jinja2.

CHAPTER 5

PROJECT MODEL VIEW

5.1 Project Settings

requirements.txt

```
asgiref==3.4.1
certifi==2021.5.30
charset-normalizer==2.0.4
Django==3.2.11
idna==3.2
Pillow==9.0.0
pytz==2021.1
sqlparse==0.4.2
urllib3==1.26.6
```

Figure 2 : Requirements

```
# Database
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}
```

Figure 3 : DB Connection

```
# Password validation

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]
```

Figure 4 : Authentication

5.2 Models: Cart, Address, and Item

```python
class Order(models.Model):
    customer = models.ForeignKey(Customer, on_delete=models.SET_NULL, null=True, blank=True)
    date_order = models.DateTimeField(auto_now_add=True)
    complete = models.BooleanField(default=False, null=True, blank=True)
    transaction_id = models.CharField(max_length=200, null=True)

    def __str__(self):
        return str(self.id)
```

Figure 5 : Order Class

```python
class OrderItem(models.Model):
    product = models.ForeignKey(Product, on_delete=models.SET_NULL, null=True)
    order = models.ForeignKey(Order, on_delete=models.SET_NULL, null=True)
    quantity = models.IntegerField(default=0, null=True, blank=True)
    date_added = models.DateTimeField(auto_now_add=True)

    @property
    def get_total(self):
        total = self.product.price * self.quantity
        return total
```

Figure 6 : Item Class

```python
class ShippingAddress(models.Model):
    customer = models.ForeignKey(Customer, on_delete=models.SET_NULL, null=True, blank=True)
    order = models.ForeignKey(Order, on_delete=models.SET_NULL, null=True)
    address = models.CharField(max_length=200, null=True)
    city = models.CharField(max_length=200, null=True)
    state = models.CharField(max_length=200, null=True)
    zipcode = models.CharField(max_length=200, null=True)
    date_added = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return self.address
```

Figure 7 : Address Class

CHAPTER 6

CONCLUSION AND FUTURE WORKS

This undertaking is merely a modest effort to meet shop demands. This website offers a computerized store manipulation system that will be useful to both users and shop customers. Users can now browse for and purchase a variety of products entirely online. It also has a feature for regular users to log in and see the status of their orders, as well as to request additional items or make suggestions. It offers the option for administrators to log in so they can add various items, check on users' activity, occasionally offer discounts, and add details about various events for the customers. Future potential for the project is enormous. The project may one day be executed on an intranet. As and when the need arises, the project can be upgraded in the near future because it is very expandable. There are numerous ways the project might be updated and enhanced in the future. It is possible to add an inventory management system, numerous branches, multilingual content, and a variety of other features to this project to make it stronger.

REFERENCES

[1] “Web Development”, Wikipedia. Accessed May 2, 2022 [Online]. Available: https://en.wikipedia.org/wiki/Web_development

[2] “Python”, W3 Schools. Accessed May 25, 2022 [Online]. Available: https://www.w3schools.com/python/

[3] “Web Site”, Wikipedia. Accessed May 29, 2022 [Online]. Available: https://en.wikipedia.org/wiki/Website

[4] “E-Commerce”, Wikipedia. Accessed June 5, 2022 [Online]. Available: https://en.wikipedia.org/wiki/E-commerce

[5] Bryan Syverson and Joel Murach, Murach’s SQL Server 2012 for developers. Austin, Texas, USA: Campbell, 2012

[6] “Database”, Stack overflow. Accessed June 20, 2022 [Online]. Available: https://stackoverflow.com/questions/tagged/database

[7] “Learn HTML and CSS faster”, Mark Myers. Accessed July 10, 2022 [Online]. Available: http://www.asmarterwaytolearn.com/htmlcss/h.html

[8] “Django”, Wikipedia. Accessed June 15, 2022 [Online]. Available: https://en.wikipedia.org/wiki/Django