SRI DURGA MALLESWARA
SIDDHARTHA MAHILA KALASALA
An Autonomous College in the Jurisdiction of Krishna University
A College with Potential for Excellence
Sponsors: SIDDHARTHA ACADEMY OF GENERAL AND TECHNICAL EDUCATION

"E-COMMERCE WEB APPLICATION"

Submitted by

B.Bhavani Ishwarya Bai 206802

J.Vasudha 206805

Of

B.Sc (MSDS)

For the academic year 2022-2023

Under the esteemed guidance of

Mrs K. Anitha, M.Tech (CSE)

DEPARTMENT OF COMPUTER SCIENCE

SRI DURGA MALLESHWARA SIDDHARTHA MAHILA KALASALA

(An Autonomous institution)

2022-2023

CERTIFICATE OF APPROVAL

This is to certify that B.Bhavani Ishwarya Bai (206802), J.Vasudha(206805) of Sri Durga Malleshwara Siddhartha Mahila Kalasala has successfully completed the project work titled **E-Commerce Web Application** in particular fulfillment of requirement for the completion of B.Sc Course as predicted by **Sri Durga Malleshwara Siddhartha Mahila Kalasala**. This project report is the record of authentic word carried out by her during the period from_____to_____. She has worked under my guidance.

Signature:

Project guide Name(Internal): Mrs. K. Anitha, M.Tech (CSE)

Date:

Counter signed by

Signature:

Name of the HOD: Mrs. K. Anitha

Date:

Signature:

Name of the External Examiner:

Date:

ACKNOWLEDGEMENT

It is indeed with a great pleasure and immence sence of gratitude that we acknowledgement the help of these individuals. We are highly indeed to our Director for the facilities provided to accomplish this main project.

We whould like to thank our K. Anitha Head of the Department of computer science SRI DURGA MALLESHWARA SIDDHARTHA MAHILA KALASALA and, for this constructive criticism throughout our project.

We feel elated in manifesting our scene of gratitude to our internal project guide K. Anitha, HOD, Department of computer science SRI DURGA MALLESHWARA SIDDHARTHA MAHILA KALASALA. He has been a constant source of inspiration for us and we are very deeply thankful to him for his support and valuble advice.

We extremely greatful to our Department staff members. Labtechnicians and Non-teaching staff members for this extreme help throughout our project.

Finally we express our heartful thanks to all our friends who helped us in successful completion of this project.

Project Associates:

B.Bhavani Ishwarya Bai(206802)

J.Vasudha (206805)

DECLARATION

I, B.Bhavani Ishwarya Bai (206802) and J.Vasudha(206805) hereby declare that this project report is the record of authentic work carried out by me during the period from 2022-2023 and has been submitted to any other university or Institute for the award of any degree/diploma etc.

Signature:

Name of the Student:

Date:

Signature:

Name of the Student:

Date:

INDEX:

SIMPLE E-COMMERCE

SIMPLE E-COMMERCE WEBSITE:

A website that allows people to buy and sell physical goods, services, and digital products over the internet rather than at a brick-and-mortar location. Through an e-commerce website, a business can process orders, accept payments, manage shipping and logistics, and provide customer service.

An e-commerce website is one that allows people to buy and sell physical goods, services, and digital products over the internet rather than at a brick-and-mortar location. Through an e-commerce website, a business can process orders, accept payments, manage shipping and logistics, and provide customer service.

It's tough to imagine daily life without e-commerce. We order food, clothes, and furniture; we register for classes and other online services; we download books, music, and movies; and so much more. E-commerce has taken root and is here to stay.

The term "e-commerce simply means the sale of goods or services on the internet. In its most basic form, e-commerce involves electronically transferring funds and data between 2 or more parties. This form of business has evolved quite a bit since its beginnings in the electronic data incharge of the 1960s and the inception of online shopping in the 1990s.

In recent years, e-commerce has enjoyed a massive boost from the rise of _smart phones, which allow consumers to shop from nearly anywhere.

Much like a traditional physical retail store, e-commerce websites allow consumers and businesses to buy and sell to one another on a designated platform. The main difference between e-commerce and physical commerce, however, is that e-commerce transactions occur entirely over the internet rather than at a brick-and-mortar location.

What is the purpose of e-commerce website?

With the evolution of technology and the wave of digitalization, more and more businesses are adapting to tech evolutions. You will find the digital payment options with the grocery sellers & street-side vendors as well. The ultimate purpose behind this evolution is to make shopping a hassle-free experience for everyone.

If you are also non-techies who want to take their business online, Builderfly can be your savior. Builderfly is an ecommerce platform exclusively designed for individuals & businesses to start selling online, market & grow their business without any technicalities. So take your business to the World at your own pace.

Types of e-commerce sites:

- **Business-to-Consumer (B2C):**

  Business-to-Consumer encompasses transactions made between a business and a consumer. B2C is one of the most popular sales models in the ecommerce context. For example, when you buy shoes from an online retailer, it's a business-to-consumer ecommerce transaction.

- **Business-to-Business (B2B):**

  Unlike B2C, B2B ecommerce encompasses sales made between businesses, such as a manufacturer and a wholesaler or retailer. B2B is not consumer-facing and happens only between businesses.

- **Consumer-to-Consumer (C2C):**

  One of the earliest forms of ecommerce, consumer-to-customer ecommerce relates to the sale of products or services between customers. This includes C2C selling relationships, such as those seen on e Bay or Amazon.

- **Consumer-to-Business (C2B):**

  C2B reverses the traditional model, meaning individual consumers make their products or services available for business buyers.

  One example of a C2B ecommerce business is iStock, an online store where stock photos are available for purchase directly from different photographers.

- **Business-to-Administration (B2A):**

  B2A covers the transactions made between online businesses and administrations. An example would be the products and services related to legal documents, social security, etc.

How do you design an e-commerce system?

The software solution is designed based on the system requirements, the people involved, the content of the operation and the activity to be performed. To increase conversion rates on your ecommerce website, no part of the customer journey can be overlooked. From your landing page to your checkout page, every aspect of your ecommerce website needs to be carefully designed. But cultivating a user journey that successfully balances an enjoyable shopping experience with a clear path to conversion is easier said than done. The design of virtual stores is often the most important factor in the success or failure of online That doesn't simply mean that e-commerce web sites have to look they have to be *usable* (quick and easy to navigate around without irritating or confusing people), *reliable* (customers expect sites to be online 24 hours a day, seven days a week, and for pages to load without delay), and *secure* (because no one is

prepared to type their credit card details into a website that isn't safe).Setting up an online store used to be quite an undertaking. Not only did you have to build a dedicated website from scratch, you also had to develop your own merchant system that could securely process credit card details and ship transactions to and from bank computers. These days, anyone can set up an online store in minutes. Websites like PayPal make it possible to build a store very quickly and, since they have built-in credit card processing features, handling transactions couldn't be simpler. Many people set up virtual storefronts on the auction site eBay and then use PayPal (now a part of eBay too) to process their transactions. Some websites (notably Amazon) allow you to incorporate mini versions of their store inside your own website—so you make a small commission selling their products within your own site. For businesses with lots of products that need to combine an easy-to-use website (for users to order from), secure ordering, and a reliable database "back-end" (to manage stock), there are sophisticated content-management systems like Shopify®, Magento®, and WooCommerce® (built on WordPress) that do most of the work for you

What system does a ecommerce use?

There's lots of money to be made online, but not all of this involves selling goods in the traditional way. Many online businesses try to make money by offering a mixture of free and premium services. Yahoo! (which originally stood for Yet Another Hierarchical Officious Oracle), is probably the best-known example of a website like this. Created as a comprehensive directory of other websites, it mutated into a search engine and then a portal, offering a gateway to all kinds of other premium services. For example, you can get free e-mail through Yahoo!, but you can also pay extra for a more sophisticated e-mail system; you can store your photographs for free on Yahoo's Flickr site, but you can pay an extra sum to have them printed out or processed in various ways. (Yahoo has recently sold Flickr, but that's another story.)

Newspapers, magazines, and book publishers also try to make money through a mixture of free and premium services. While most of them offer their basic content (the horrible, unappealing name that online businesses give to the words and pictures they publish) for free, using advertising to make money, some also offer a proportion of their articles for a one-off fixed fee or subscription). Buying an article involves a transaction similar to the ones you'd make on Amazon or eBay, so this kind of online publishing is also clearly a variety of e-commerce.

How to manage an ecommerce site?

E-commerce, the electronic commerce that has revolutionized the world and the way businesses are carried out, has become the order of the day if people wanted to expand the horizons for their business on the World Wide Web. E-commerce storefronts are present in varying magnitudes in the form of brick and click companies and pure-click companies. But all of these companies have one thing that is common to the e-commerce management**.** Any business should be able to identify, define as well as manage the pertinent eCommerce initiative and have a professional to execute this initiative.

Coming to think of what eCommerce manage is each one has different ideologies as far as this topic is concerned. In earlier times, it was the responsibility of the programmers and IT engineers, and later it was passed on to the web designing professionals. But, a business’s

eCommerce needs in today's world dictate and command a proper eCommerce management and a manager to take care of the entire campaign.

eCommerce manage involves the launching as well as running of a business website wherein the eCommerce manager has the prowess to manage all those personnel who are involved in providing the eCommerce functions. In short it encompasses familiarity with the responsibilities and roles that the fields of IT, marketing and website development faculties need to perform and entail interfacing with them on a regular basis to create as well as promote the eCommerce for the business.

Creation and management of a business's e-commerce initiative is no cakewalk at all. It calls for a special acumen to understand that e-commerce is like any other integral department of a business and hence eCommerce manage involves online branding, promoting the website, maintaining the freshness as well as quality of the eCommerce website, and keep creating fresh plans for the online business initiative of the website.

The chief area where eCommerce manage focuses on is the identification and redefinition of the eCommerce initiative and objectives of the company involved so that the eCommerce website delivers the pertinent message on the internet marketplace. Once the goal of any 'digital strategy' is defined, it becomes easy to sell the concept. So, identifying the prime reason of existence of the eCommerce website is really crucial. Whether the site is for commerce, community or for content…or for a combination of them, getting eCommerce revenue from all types of businesses is become very much possible today on the internet platform with appropriate eCommerce manage.

eCommerce manage therefore stresses more on all the functions of the faculties that have been mentioned earlier. It begins with the creation of a USP and then designing it to suit the web. This management is an integral part of all Business to Business transactions and hence has become the backbone of all commercial sites that exist on the internet platform.

SQL DATABASE:

SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987.SQL is also known as Standard Query Language which is used as a medium for communicating with the database. SQL lets you access and manipulate databases SQL statements are utilized to execute the queries against the database and retrieve data from the database. We can create a new database, table, stored procedure and update, delete and add items to the table. We also can view the data and set permission to the view, procedure, and table. SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in a relational database.

SQL is the standard language for Relational Database System. All the Relational Database Management Systems (RDMS) like MySQL, MS Access, Oracle, Sybase, Informix, Postgres and SQL Server use SQL as their standard database language. Also, they are using different dialects, such as -

• MS SQL Server using T-SQL,

• Oracle using PL/SQL,

• MS Access version of SQL is called JET SQL (native format) etc.

SQL is widely popular because it offers the following advantages

• Allows users to access data in the relational database management systems.

• Allows users to describe the data.

• Allows users to define the data in a database and manipulate that data.

• Allows to embed within other languages using SQL modules, libraries & precompilers.

• Allows users to create and drop databases and tables.

• Allows users to create view, stored procedure, functions in a database.

• Allows users to set permissions on tables, procedures and views.

E-Commerce In Python:

E-Commerce is currently one of the fastest and dynamically evolving industries in the world. Its popularity has been growing rapidly with the ease of digital transactions and quick door-to-door deliveries. A major source of big tech companies revenues comes from the interaction of their underlying proprietary algorithms that are heavily powered by data science, so it's fundamental to understand the methods applied to maintain and grow the number of clients.

In this article, I'll give you some insights about how this amazing industry is applying data science with a Python script to practically show the case. Both the datasets and the script can be found at my GitHub following this link. We will create a e-commerce that will

Project Name	E-Commerce Project In Python and MY SQL with source code
Abstract	E-Commerce Project in Python
Language used	Python(GUI)
Database	MySQL
Type	Desktop Application

Python E-Commerce Project:

In this project, we will develop a ecommerce that will help us in performing various functions like home ,issuing,return cart. We're going to take you step-by-step to build a modern, fully open-source, eCommerce web application using Python. Installation of django is required before starting the project .

Importance of E-Commerce Project in Python:

Python as a programming language makes it easy for e-commerce websites to maintain and regulate the database from time to time**.** This helps the site to easily add the details of their customers, along with adding multiple billing and shipping addresses.

E-Commerce Project in Python Beneficiaries:

- Low costs
- Flexibility and speed
- Speeds up the buying process
- Availability of reviews

The key beneficiaries will be local businesses and individuals with and without business registrations as they will receive e-commerce onboarding training, seller subsidy and sales support. Other local e-commerce players in the ecosystem such as digital marketing, delivery and e-payment services are also benefitting from the spill over effect.

Major Functionalities:

• Log In/Log Out Management – For this functionality the user can log in and log out to the system.

• User Management – For this functionality the user/admin can manage all the users.

Simple e-commerce website using Python with Source Code Installation:

1. You will have to download & install the Python IDLE's, here's the link "https://www.python.org/downloads/".

2. Extract the zip file.

3. Open the extracted folder

4. Locate the py file.

5. Then open the file via python IDLE. 6. Run the py file to launch the program.

To start executing a Project In Python and My sql, we should have Pycharm IDE and XAMMP installed in our computer

SYSTEM ANALYSIS

The purpose of the analysis is to determine the specific functional requirements of a proposed computer-based Integrated E-Commerce. This proposed system is recommended to be selected form the available off the –shelf software packages, a number of techniques have been employed including interviews, document analysis, observation, and literature review.

The Internet related data and communication advancements offer extraordinary open doors for business development. The internet and related advancements Organizations can utilize a global innovation program for e-commerce. Online-stores systems are developed as distributed applications. The architecture of an e-commerce system conceived as a distributed application. The Internet associated information and communication technologies offer unprecedented opportunities for business innovation. Companies can use the Internet and associated technologies extensively as a global and cost effective program for e-commerce. This paper suggests that processes such as project design are personalized, and profiling, control, and security should be implemented in an e-commerce application. Also, search management, content management, payment systems, catalog management, workflow management, event notification, and collaboration and trading. This paper suggests that processes such as project design or content management, payment systems, and search management coordinated effort and exchanging.

5 more user-driven ways to analyze your ecommerce site:

User-driven analysis gives you a better understanding of your web visitors' behavior and helps you identify what's most important to them in their user journey. The data you get from user-driven analysis will help you decide what changes you need to make to improve the user experience, making people more likely to come back to your online store to shop again in the future.

- **Pyschographics and user personas:**

Studying your customers and web visitors based on characteristics and traits like values, desires, goals, interests, and lifestyle choices helps you learn what is most important to them when it comes to their journey across your brand.

- **Market research:**

Using research techniques to gain a better understanding of your target market can help you design better products, improve the overall user experience, reach quality leads, and increase conversion rates.

- **Usability testing:**

Observing web users' behavior to test your site's design and functionality can reveal whether your users understand how to navigate your site or if they can carry out tasks across its pages. You can also see how they react when they encounter pain points in their customer journey.

- **User and customer feedback:**

Collecting feedback from your customers lets you learn directly from them in their own words—and it doesn't have to come just from website surveys, like we showed you above: you can also get on a call with your customers, or read through Customer Support tickets and Customer Success calls notes. You can use what you learn to improve your products, sales funnels, and the overall customer experience.

- **User behavior analytics[UBA]:**

When you combine qualitative and quantitative data about your visitors, you can see *how* and *why* they interact with elements or pages on your site. Understanding the full picture of your visitors' experience will help you pinpoint the changes that need to be made to your site to improve it.

Role of a Systems Analyst:

A systems analyst plays a multifaceted role. He performs as a change agent, monitor, architect, psychologist, salesperson, motivator, and politician.

Change agent:

People inherently resist change and people can even become ineffective in case of excessive changes. However, systems analysis and design is essentially about change and reorientation to a new system.

Architect:

As an architect, the systems analysts create a detailed design of candidate systems. He obtains the abstract ideas and requirements of the users and provides details to build the end product.

Psychologist:

Systems are built around people and therefore, it is very important to understand people. In the role of psychologist the systems analyst reaches out to people, interprets their thoughts, assesses their behavior and draws conclusions from these interactions.

Salesperson:

Selling change can be as crucial as initiating change. Selling the system actually takes place at each step in the system life cycle. Sales skills and persuasiveness are crucial to the success of the system.

Communication:

A system analyst should have the ability to articulate an speak the language of the user, have a flair for mediation and a knack for working with virtually all levels in the organization

Understanding:

A system analyst should be able to identify problems and assessing Systems Analysis 17 its impact and have grasp of the organization goals and objectives and show sensitivity to the impact of the system on people at work.

Problem solving:

He/she should reduce problems to their elemental levels for analysis, develop alternative solutions to a given problem, and define the pros and cons of candidate systems.

Project management:

Scheduling, performing well under time constraints, coordinating team efforts, and managing cots and expenditure form the project management skill required by a system analyst.

Dynamic interface:

As customers go through your attractive banners, they would love to make more of your brand's first purchase. Dynamic flags (articles, products, holiday discounts, etc.) are used to maintain a relationship with the reader and enable them to click on the "go-to site" or "checkout" button.

Personalized contact forms like iframes, sidebar, or pop-ups are a great way to generate leads. These are effective ways of collecting users' contact information and approach them accordingly. This helps to analyze the reader's behavior history and preferences and offers goods and services that correspond to its interests like "buy this product" "similar products" blocks.

The spread of dynamic content and personalization technologies is one of the brightest trends in online marketing in recent years. Shortly, such strategies are likely to spread even more comprehensively and become even more perfect. Therefore, it is worth investing in dynamic content technologies and creating a comprehensive strategy to increase your company's profit with their help.

Dynamic content allows you to immediately give the user the information and opportunities relevant to him at each stage of the user experience. It will enable users to meet their needs and find what they want quickly. It enhances conversion, making the customer experience more relevant and intuitive.

Recognition of Need :

The growth of e-commerce has enabled buyers and sellers to connect in ways that were previously difficult or impossible. E – Commerce is defined as the exchange of goods and services on the internet. A space that was traditionally accessible to only large corporations, e-commerce has enabled entities of all sizes, including freelancers to startups to sell products and services considering the ease of accessibility to customers and suppliers.
E-commerce transactions include everything from buying a t-shirt to hiring a freelancer for your business. It continues to redefine the retail industry and how we shop. However, conducting business via the internet adds complexity to the accounting processes of e-commerce companies.

Analysis:

Ecommerce analytics is the process of accumulating data from all of the areas that have an impact on your store. You should then use this data so that you can comprehend shifts in customer behavior and online shopping trends. Ultimately, you can make more intelligent decisions by basing them on data, which should result in more online sales being made.
Ecommerce analytics give you the power to get a better understanding of how your business is performing now and how it is likely to perform in the future. This forecasting will inform everything from hiring goals and sales goals to making sure that the right products are accessible at the right time so that your customers' expectations are met.

Design:

An ecommerce site offers you the chance to build your brand, connect with more customers, and sell more products but only if you've got the right website design. Web design is critical when creating an ecommerce website. Good ecommerce web design is all about using the right colors, fonts, images, words and graphics to convince visitors to make a purchase. Your ecommerce website design should attract potential customers, provide great user experience and present your shop in the best light. Once you know who you are, you can work it into the branding of your ecommerce site

SOFTWARE REQUIREMENTS

Software Requirements:

Operating System : Windows
User Interface : HTML,CSS
Client-Side Scripting : JavaScript
Programming Language : Python
Database : MYSql
Technologies: HTML,Python

Hardware Requirements :

Processor : Intel
Hard Disk : 50GB
RAM : 300GB

Technologies:

HTML(Hyper Text Markup Language):

HTML stands for Hyper Text Markup Language, which is the most widely used language on Web to develop web pages. HTML was created by Berners-Lee in late 1991 but "HTML 2.0" was the first standard HTML specification which was published in 1995. HTML 4.01 was a major version of HTML and it was published in late 1999. Though HTML 4.01 version is widely used but currently we are having HTML-5 version which is an extension to HTML 4.01, and this version was published in 2012.

Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers. Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.
HTML is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain

Advantages:

Create Web site - You can create a website or customize an existing web template if you know HTML well.
• Become a web designer - If you want to start a carrer as a professional web designer, HTML and CSS designing is a must skill.
• Understand web - If you want to optimize your website, to boost its speed and performance, it is good to know HTML to yield best results.
• Learn other languages - Once you understands the basic of HTML then other related technologies like javascript, php, or angular are become easier to understand.
The main parts are as follows:

1. The opening tag: This consists of the name of the element (in this case, p), wrapped in opening and closing angle brackets. This states where the element begins or starts to take effect — in this case where the paragraph begins.
2. The closing tag: This is the same as the opening tag, except that it includes a forward slash before the element name. This states where the element ends — in this case where the paragraph ends. Failing to add a closing tag is one of the standard beginner errors and can lead to strange results.
3. The content: This is the content of the element, which in this case, is just text.
4. The element: The opening tag, the closing tag, and the content together comprise the element.

Applications of HTML:

As mentioned before, HTML is one of the most widely used language over the web. I'm going to list few of them here:
• Web pages development - HTML is used to create pages which are rendered over the web. Almost every page of web is having html tags in it to render its details in browser.
• Internet Navigation - HTML provides tags which are used to navigate from one page to another and is heavily used in internet navigation.
• Responsive UI - HTML pages now-a-days works well on all platform, mobile, tabs, desktop or laptops owing to responsive design strategy.
• Offline support -HTML pages once loaded can be made available offline on the machine without any need of internet.
• Game development- HTML5 has native support for rich experience and is now useful in gaming developent arena as well.

Python:

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). This tutorial gives enough understanding on Python programming language.Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages. Python is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain.

Advantages:

Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
Python is Interactive − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python is Object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
Python is a Beginner's Language − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

Characteristics of Python:

Following are important characteristics of Python Programming –
• It supports functional and structured programming methods as well as OOP.
• It can be used as a scripting language or can be compiled to byte-code for building large applications.
• It provides very high-level dynamic data types and supports dynamic type checking.
• It supports automatic garbage collection.
• It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java. Hello World using Python. Just to give you a little excitement about Python, I'm going to give you a small conventional Python Hello World program, You can try it using Demo link

```
print ("Hello, Python!");
```

Applications of Python:

As mentioned before, Python is one of the most widely used language over the web. I'm going to list few of them here:

• **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

• **Easy-to-read** − Python code is more clearly defined and visible to the eyes.

• **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.

•**Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

• **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

• **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

• **Databases** − Python provides interfaces to all major commercial database
A software requirements specification (SRS) is a description of a software system to be developed. It is modeled after business requirement specification. The software requirements

specification lays out functional and non-functional requirements, and it may include a set of use cases that describe user interactions that the software must provide to the user for perfect interaction.

Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers on how the software product should function (in a market-driven project, these roles may be played by the marketing and development divisions). Software requirements specification is a rigorous assessment of requirements before the more specific system design stages, and its goal is to reduce later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules. Used appropriately, software requirements specifications can help prevent software project failure.The software requirements specification document lists sufficient and necessary requirements for the project development. To derive the requirements, the developer needs to have clear and thorough understanding of the products under development. This is achieved through detailed and continuous communications with the project team and customer throughout the software development process.

The SRS may be one of a contract's deliverable data item description or have other forms of organizationally-mandated content.

Typically a SRS is written by a technical writer, a system architect, or a software programmer.

The main types of requirements are:

• Functional Requirements.
• Performance Requirements.
• System Technical Requirements.
• Specifications.

User (Customer) :

This document is intended to user and customer to make them ensure that this document is well meeting the need of the users.

Project Manager :

This SRS document is also very important for the project manager to ensure that can estimate the cost easily by referring to the SRS document and that it contains all the information require planning the project.

Project Developer:

The project developer will refer to the SRS document to make sure that they developed exactly hat the customer requires.

Tester:

The tester will read this SRS document and he will ensure that the requirements are understandable from functionality point of view so that he can test the software and validate its working.

Document Writer:

The document writer is reading the SRS document is to ensure that they understand the document well enough to be able to write the users manuals.

Maintenance :

The SRS document helps the maintenance engineers to understand functionality of the system, a clear knowledge of the functionality can help them to understand design and code.

Suggestions:

1. The user can read the whole SRS document but for him Introduction, Overall description and System features is much required the software performance.
2. For the project manager the system features is very important.
3. The developer must go through the whole SRS for understanding the requirement and functioning of software.
4. The designer and coder must see the class and object diagram and state transmission diagram for coding the modules.
5. A tester must be aware of coding language and visit through the code section and check the required output.
6. The document writer should write the qualitative document so that it becomes easy and understand to every one.

Project Scope:

Before we launch into the future and scope of eCommerce in India, let us first understand what is e-commerce. To put it simply, electronic commerce refers to the purchase and sale of goods online or via the internet.Sellers make websites where they display images of their products with price and description. Shoppers who buy the products have multiple payment options like COD, e-wallet, net banking, credit card, and so on.

Online sellers have the responsibility of shipping the product to the buyer and ensuring safe and timely delivery. A list of high-level units and tasks to be executed throughout the project. It's basically a predecessor of a work breakdown structure (WBS) for an online shopping website to be completed at the next step of planning. Companies involved in electronic commerce as either buyers or sellers rely on Internet-based technologies and e-commerce applications and services to accomplish marketing, discovery, transaction processing, and product and customer service processes

Project milestones — the important tasks and achievements that mark a certain stage of the project completion. They are predecessors of a project timeline and represent a clear sequence of events showing how the project advances.

Major known risks — external conditions that may adversely affect a project's progress. The likelihood of each risk occurring and the severity of its risk should be estimated at this point.

Constraints — restrictions defining a project's limits (e.g., budget, schedule, dependency on third-party software releases, etc.). These may also include the external dependencies — activities and tasks outside the team's control — that may affect the project implementation.

System Features :

The users of the system should be provided the surety that their account is secure. This is possible by providing:-

- An eCommerce business is a business model which let the companies set up their business on an electronic network. Rather than displaying products in a physical store, a businessman put the products or services he wants to sell on a website on the internet.
- As eCommerce business is easy to set up and does not require too much investment to set up companies of all sizes can set up an eCommerce business.
-

User Requirement:

The first step in developing any eCommerce application is to interview the user base to generate a list of features to be included in the application. This comprises the important input for defining the capabilities of the application.

There are two sets of users of shopping cart applications: site administrators and end users who purchase items using their Web browsers. After interviewing end users and administrators, application requirements such as the following may be generated.

Backup and Recovery :

Forgot Password –

Data migration i.e. whenever user registers for the first time then the data is stored in the server

Data replication i.e. if the data is lost in one branch, it is still stored with the server

Auto Recovery i.e. frequently auto saving the information Maintaining files i.e. File Organization The server must be maintained regularly and it has to be updated from time to time

Data and Category Requirement:

The following are the list of conventions and acronyms used in this document and the project as well:

• **User:** A general login id assigned to most users

• **Client:** Intended users for the software

• **SQL:** Structured Query Language; used to retrieve information from a database

• **SQL Server:** A server used to store data in an organized format

• **Layer:** Represents a section of the project

• **User Interface Layer:** The section of the assignment referring to what the user interacts with directly

• **Application Logic Layer:** The section of the assignment referring to the Web Server. This is where all computations are completed

• **Data Storage Layer:** The section of the assignment referring to where all data is recorded

• **Interface:** Something used to communicate across different mediums

General Constraints:

- The need for online identity verification ...
- Delivering an omnichannel customer experience ...
- Outshining the vast competition ...
- The need to revamp selling tactics ...
- Shopping cart abandonment ...
- Maintaining customer loyalty ...
- The headaches of product return and refund ...
- The struggle of competing on price and shipping.

Assumptions and Dependencies:

The success of this system depends on

- Existence of an Internet service to all people in Gaza Strip.
- Website interface must be friendly and easy-to-use.
- The search mechanism should be simple and fast.

Functional Requirements:

- **Speed** — Shoppers expect lightning-speed load times, and Google does too.
- **Mobile Friendliness** — Make sure your responsive implementation is professionally executed, as clunky mobile experience discounts the quality of the business for the fast-paced shopper.
- **Ease of use in the checkout flow** — Focus on ease of use; don't introduce unnecessary steps to complete the purchase. Strive for a one-click experience.
- **Personalization** — A third-party personalization tool integrated into your ecommerce platform can influence your sales by up to 59% percent.
- **Accessibility** — Follow the universal guidelines and implement your accessibility standard to include all shoppers.

It is all about the functions and core operations of your e-store that enable a user to take action on the website. They can be implemented as a single website feature and form the basis of the whole software development process.

Nonfunctional Requirements:

❖ Performance Requirements:

As corporations turn to web-based architecture for mission-critical systems, such as e-commerce, customer relationship management, field sales management systems, etc., they are faced with the challenge of optimizing said systems to make the most of their on-prem or cloud infrastructure. Defining the non-functional requirements for a load test is perhaps the most important step in ensuring that mission-critical systems can be tested to scale and meet customer load. They can make or break the success of a product or system. Responses to view information shall take no longer than 5 seconds to appear on the screen.

Safety Requirements:

• System use shall not cause any harm to human users.

Security Requirements:

• System will use secured database

• Normal users can just read information but they cannot edit or modify anything except their personal and some other information.

• System will have different types of users and every user has access constraints.

Usability:

- System should be very much convenient to be used by both managers and accountant so that they do not require any extra manual for it.

- The system should be simple in such a way that it can deliver all the required data and relationship.

Error Handling:

- Lack of Payment Options
- Slow-Moving Checkout Process
- Absence of Relevant Information
- Sluggish Customer Support Process
- Unimpressive User-Interface Design
- Inconsistency Between Channels
- Unavailability of Real-time Assistance
- Products Are “Out of Stock”
- The Missing Case of ‘Personalized Experience’
- Improper Packaging & Handling
- Poor Filtering Options

Product Perspective:

E-commerce can be defined from five perspectives. First, business process. E-commerce is doing business electronically by completing business process over electronic network, thereby substituting information for physical business process (Weill and Vitale 2001). Second, services. It is a tool that addresses the desire of governments, firms, consumers, and management to cut service costs while improving the quality of customer service and increasing the speed of service delivery. Third, learning. It enables online training and education schools to the customers. Fourth is collaborative which is framework for inter- and intra organizational collaboration. Final is community, which provides a gathering place for community members to learn, transact, and collaborate.

Product Functions:

This section provides the graspable functional overview of the end product. The product is expected to be providing following functionalities

1. Login
2. Logout
3. Registration
4. Add to cart
5. Payment

Operating Environment:

Operating system: Any Windows Operating System

Language: python

User Documentation: It will provide specific guidelines to a user for using the E-Commerce.

External Interface Requirements:

User Interfaces:

The system provides a sophisticated platform for the user to enter their account information as to login to their account..When it comes to e-commerce, User Interface design makes a difference between customers who abandon their carts and the ones that place an order. Elements like high quality product images are useful to persuade users to make a purchase. Clear call to action buttons and Seamless checkouts demonstrate that graphic and intuitive design must combine to make user navigation easy to use. Attention to user interface design details like the shopping cart, images, and call to actions will increase purchases and boost customer satisfaction.

Hardware Interfaces:

Add Cart:

The **add-to-cart button** is a feature of ecommerce stores that allows customers to choose items to purchase without actually completing the payment.

Search for Categories:

I have a category function on my blog. If I want 'fruits' I can just go to the grocey category and find fruits there in the same way we have different categories like “grocery”, “women”, “men”, “kids” where we can find the items related to them.

User Panel:

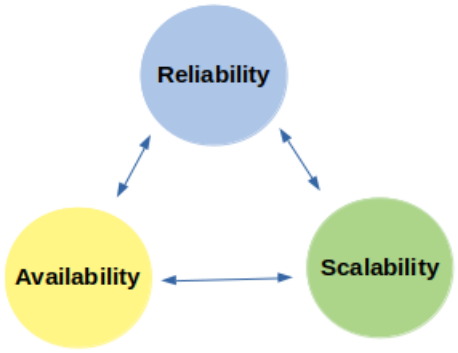
A window which shows all the functionalities of the user in the system.

Logout:

This functionality allows the customer to logout from the account.

SYSTEM DESIGN

System Design is the process of designing the architecture, components, and interfaces for a system so that it meets the end-user requirements. System Design for tech interviews is something that can't be ignored! Almost every IT giant whether it be Facebook, Amazon, Google, or any other ask various questions based on System Design concepts such as scalability, load-balancing, caching, etc



Reliability in System Design – A system is Reliable when it can meet the end-user requirement. When you are designing a system you should have planned to implement a set of features and services in your system. If your system can serve all those features without wearing out then your System can be considered to be Reliable. A Fault Tolerant system can be one that can continue to be functioning reliably even in the presence of faults. Faults are the errors that arise in a particular component of the system. An occurrence of fault doesn't guarantee Failure of the System. Failure is the state when the system is not able to perform as expected. It is no longer able to provide certain services to the end-users.

Availability in System Design:

Availability is a characteristic of a System which aims to ensure an agreed level of Operational Performance, also known as uptime. It is essential for a system to ensure high availability in order to serve the user's requests. The extent of Availability varies from system to system. Suppose you are designing a Social Media Application then high availability is not much of a need. A delay of a few seconds can be tolerated. Getting to view the post of your favorite celebrity on Instagram with a delay of 5 to 10 seconds will not be much of an issue. But if you are designing a system for hospitals, Data Centers, or Banking, then you should ensure that your system is highly available. Because a delay in the service can lead to a huge loss.

There are various principles you should follow in order to ensure the availability of your system :

• Your System should not have a Single Point of Failure. Basically, your system should not be dependent on a single service in order to process all of its requests. Because when that service fails then your entire system can be jeopardized and end up becoming unavailable.

• Detecting the Failure and resolving it at that point.

Scalability in System Design :

Scalability refers to the ability of the System to cope up with the increasing load. While designing the system you should keep in mind the load experienced by it. It's said that if you have to design a system for load X then you should plan to design it for 10X and Test it for 100X. There can be a situation where your system can experience an increasing load. Suppose you are designing an Ecommerce application then you can expect a spike in the load during a Flash Sale or when a new Product is Launched for sale. In that case, your system should be smart enough to handle the increasing load efficiently and that makes it Scalable. In order to ensure scalability you should be able to compute the load that your system will experience. There are various factors that describe the Load on the System:

• Number of requests coming to your system for getting processed per day
• Number of Database calls made from your system
• Amount of Cache Hit or Miss requests to your system
• Users currently active on your system.

System Design Strategy:

A good system design is to organize the program modules in such a way that are easy to develop and change. Structured design techniques help developers to deal with the size and complexity of programs. Analysts create instructions for the developers about how code should be written and how pieces of code should fit together to form a program.

Importance :

1. If any pre-existing code needs to be understood, organized, and pieced together.
2. It is common for the project team to have to write some code and produce original programs that support the application logic of the system.

There are many strategies or techniques for performing system design. They are:

• Bottom-up approach:

The design starts with the lowest level components and subsystems. By using these components, the next immediate higher-level components and subsystems are created or composed. The process is continued till all the components and subsystems are composed into a single component, which is considered as the complete system. The amount of abstraction grows high as the design moves to more high levels. By using the basic information existing system, when a new system needs to be created, the bottom-up strategy suits the purpose.

Advantages:

• The economics can result when general solutions can be reused.
• It can be used to hide the low-level details of implementation and be merged with the top-down technique.

Disadvantages:

• It is not so closely related to the structure of the problem.
• High-quality bottom-up solutions are very hard to construct.
• It leads to the proliferation of 'potentially useful' functions rather than the most appropriate ones.

Top-down approach:

Each system is divided into several subsystems and components. Each of the subsystems is further divided into a set of subsystems and components. This process of division facilitates in forming a system hierarchy structure. The complete software system is considered as a single entity and in relation to the characteristics, the system is split into sub-system and component. The same is done with each of the sub-systems. This process is continued until the lowest level of the system is reached. The design is started initially by defining the system as a whole and then keeps on adding definitions of the subsystems and components. When all the definitions are combined together, it turns out to be a complete system.

Advantages:

• The main advantage of the top-down approach is that its strong focus on requirements helps to make a design responsive according to its requirements.

Disadvantages:

• Project and system boundaries tend to be application specification-oriented. Thus it is more likely that advantages of component reuse will be missed.
•The system is likely to miss, the benefits of a well-structured, simple architecture. System Design is one of the tasking sections of the Programming. In this section of the project many previews are going to be seen and we are gradually getting close to the new system. System design is a transition from a user-oriented document to a document oriented to programmers or ⎪ Output design ⎪database personnel. The system design is structured into the following parts: System Flowchart⎪ Database design ⎪Input design.
Output Design In a very competitive world that we are, a good and attractive GUI is needed to make customers and administrators enjoy the services of a system, which would serve as a system to increase productivity in supermarket business below are previews of the output designs.
Database Design Database is a file composed of records, each containing fields together with a set of operations it helps in organizing data in a logical order for references. Database contains related data which are organized together in a group of object, table, and file. It can be in form of node. In this project a relational database concept will be used in this appraisal, related data will be store or organize in different table.
System Implementation and Documentation System Implementation This section is the part that puts a planned system into action and examine in details the analysis and design of the Skillmid supermarket system. The present section discusses the implementation of the system, highlighting the testing exercise and describing some of the main components of the system's Graphical User Interface.

System Maintenance:

The program may be maintained on the ground that the system requires an upgrade. When there is a new field to be added or a new form to be added in other to serve users well. Though it is compiled as standalone software the database can be tempered with but it's advisable that the admin put a password on the file to secure the database from intrusion. The following precaution should be done

• Ensure that the computer is kept in clean areas.
• Backup of data is important

OPERATING ENVIRONMENT:

This system will be operated on any computer with the following minimum specifications:
1.Windows XP Service Pack
2.OR higher versions of windows.
3.Computer hardware should be build on INTEL chipset.
4. Minimum free RAM of 128 MB.
5. Internet connectivity required.

DESIGN AND IMPLEMENTATION CONSTRAINTS:

The design & implementation constraints are:
The system database used should be an open source technology.
This system software size should not exceed 1GB.

USER DOCUMENTATION:

The user manual and help will be available online and can be accessed any time by any user. The manual will be updated on a regular basis so as to keep the contents up to date.

ASSUMPTIONS AND DEPENDENCIES:

It is assumed that the optimum internet connectivity speed will be more than 512Kbps. If the bandwidth is less than this then the transaction completion will take more time to processed and to be complete.

EXTERNAL INTERFACE REQUIREMENTS:

- **User Interface:**

Various GUI elements like forms, images and standard buttons will be included in the User Interface.

- **Software Interface:**

Software will work on Windows OS. The Database used will be an open source database like MySql. And the system will run on Java Virtual Machine.

- **Communication Interface:**

This system will require web browser, internet connection which supports HTTP and server.

FUNCTIONAL REQUIREMENTS:

- Speed
- Mobile Friendliness
- Easy of use in the checkout flow
- Personalization
- Accessibilty

OTHER NON-FUNCTIONAL REQUIREMENTS

PERFORMANCE REQUIREMENTS:

Login/Registration will not take more than 10 seconds. Any financial transactions will not take more than 15 seconds.

SAFETY AND SECURITY REQUIREMENTS:

1. Database will be secured by authentication process.
2. Unauthorized access will be avoided and will be tracked.
3. Database backup will be maintained.

SOFTWARE QUALITY ATTRIBUTES:

System will be reliable. System can be maintained easily. Testing is the process of executing a program with the aim of finding errors. To make our software perform well it should be error-free. If testing is done successfully it will remove all the errors from the software. A systemic approach is required for a coherent and well-running system. Bottom-Up or TopDown approach is required to take into account all related variables of the system. A designer uses the modelling languages to express the information and knowledge in a structure of system that is defined by a consistent set of rules and definitions. The designs can be defined in graphical or textual modelling languages.
Some of the examples of graphical modelling languages are

a. Unified Modelling Language (UML): To describe software both structurally and behaviourally with graphical notation.

b. Flowchart : A schematic or stepwise representation of an algorithm.
c. Business Process Modelling Notation (BPMN): Used for Process Modelling language.
d. Systems Modelling Language (SysML): Used for systems engineering.

Types of System Design

- **Architectural design:**

The architectural design of a system emphasizes the design of the system architecture that describes the structure, behavior and more views of that system and analysis. It is also known as high level design that focuses on the design of system architecture. It describes the structure and behavior of the system. It defines the structure and relationship between various modules of system development process.

- **Logical design:**

The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system. This is often conducted via modelling, using an over-abstract (and sometimes graphical) model of the actual system. In the context of systems, designs are included. Logical design includes entity-relationship diagrams (ER diagrams). Logical design pertains to an abstract representation of the data flow, inputs, and outputs of the system. It describes the inputs (sources), outputs (destinations), databases (data stores), procedures (data flows) all in a format that meets the user requirements.

- **Physical design:**

Physical design relates to the actual input and output processes of the system. It focuses on how data is entered into a system, verified, processed, and displayed as output. It produces the working system by defining the design specification that specifies exactly what the candidate system does. It is concerned with user interface design, process design, and data design. It consists of the following steps

• Specifying the input/output media, designing the database, and specifying backup procedures.
• Planning system implementation.
• Devising a test and implementation plan, and specifying any new hardware and software.
• Updating costs, benefits, conversion dates, and system constraints.
Put another way, the physical portion of system design can generally be broken down into three subtasks:
1. User Interface Design
2. Data Design
3. Process Design
User Interface Design is concerned with how users add information to the system and with how the system presents information back to them. Data Design is concerned with how the data is represented and stored within the system. Finally, Process Design is concerned with how data

moves through the system, and with how and where it is validated, secured and/or transformed as it flows into, through and out of the system.

- **Detailed Design**

It follows Architectural design and focuses on development of each module.

Conceptual Data Modeling:

It is representation of organizational data which includes all the major entities and relationship. System analysts develop a conceptual data model for the current system that supports the scope and requirement for the proposed system.

The main aim of conceptual data modeling is to capture as much meaning of data as possible. Most organization today use conceptual data modeling using E-R model which uses special notation to represent as much meaning about data as possible.

Entity Relationship Model:

It is a technique used in database design that helps describe the relationship between various entities of an organization.
Terms used in E-R model

ENTITY − It specifies distinct real world items in an application.
RELATIONSHIP − They are the meaningful dependencies between entities.
ATTRIBUTES − It specifies the properties of relationships.

UML DIAGRAMS

UML:

A UML diagram is a diagram based on the UML (Unified Modeling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain, or document information about the system.UML is an acronym that stands for Unified Modeling Language. Simply put, UML is a modern approach to modeling and documenting software. In fact, it's one of the most popular business process modeling techniques.It is based on diagrammatic representations of software components. As the old proverb says: "a picture is worth a thousand words". By using visual representations, we are able to better understand possible flaws or errors in software or business processes.

UML was created as a result of the chaos revolving around software development and documentation. In the 1990s, there were several different ways to represent and document software systems. The need arose for a more unified way to visually represent those systems and as a result, in 1994-1996, the UML was developed by three software engineers working at Rational Software. It was later adopted as the standard in 1997 and has remained the standard ever since, receiving only a few updates. Mainly, UML has been used as a general-purpose modeling language in the field of software engineering. However, it has now found its way into the documentation of several business processes or workflows. For example, activity diagrams, a type of UML diagram, can be used as a replacement for flowcharts. They provide both a more standardized way of modeling workflows as well as a wider range of features to improve readability and efficacy.

You can also create your own set of diagrams to meet your requirements. Diagrams are generally made in an incremental and iterative way.

There are two broad categories of diagrams and they are again divided into subcategories

- Structural Diagrams
- Behavioral Diagrams

❖ Structural Diagrams:

The structural diagrams represent the static aspect of the system. These static aspects represent those parts of a diagram, which forms the main structure and are therefore stable.These static parts are represented by classes, interfaces, objects, components, and nodes. The four structural diagrams are

- Class diagram
- Object diagram
- Component diagram
- Deployment diagram

- **Class Diagram:**

Class diagrams are the most common diagrams used in UML. Class diagram consists of classes, interfaces, associations, and collaboration. Class diagrams basically represent the object-oriented view of a system, which is static in nature .

Active class is used in a class diagram to represent the concurrency of the system. Class diagram represents the object orientation of a system. Hence, it is generally used for development purpose. This is the most widely used diagram at the time of system construction.

- **Object Diagram:**

Object diagrams can be described as an instance of class diagram. Thus, these diagrams are more close to real-life scenarios where we implement a system.Object diagrams are a set of objects and their relationship is just like class diagrams. They also represent the static view of the system. The usage of object diagrams is similar to class diagrams but they are used to build prototype of a system from a practical perspective.

- **Component Diagram:**

Component diagrams represent a set of components and their relationships. These components consist of classes, interfaces, or collaborations. Component diagrams represent the implementation view of a system. During the design phase, software artifacts (classes, interfaces, etc.) of a system are arranged in different groups depending upon their relationship. Now, these groups are known as components. Finally, it can be said component diagrams are used to visualize the implementation.

- **Deployment Diagram**:

Deployment diagrams are a set of nodes and their relationships. These nodes are physical entities where the components are deployed. Deployment diagrams are used for visualizing the deployment view of a system. This is generally used by the deployment team.

❖ Behavioral Diagrams:

Any system can have two aspects, static and dynamic. So, a model is considered as complete when both the aspects are fully covered. Behavioral diagrams basically capture the dynamic aspect of a system. Dynamic aspect can be further described as the changing/moving parts of a system. UML has the following five types of behavioral diagrams

• Use case diagram
• Sequence diagram
• Collaboration diagram
• State chart diagram
• Activity diagram

- **Use Case Diagram:**

Use case diagrams are a set of use cases, actors, and their relationships. They represent the use case view of a system. A use case represents a particular functionality of a system. Hence, use case diagram is used to describe the relationships among the functionalities and their internal/external controllers. These controllers are known as actors.

- **Sequence Diagram:**

A sequence diagram is an interaction diagram. From the name, it is clear that the diagram deals with some sequences, which are the sequence of messages flowing from one object to another.

- **Collaboration Diagram:**

Collaboration diagram is another form of interaction diagram. It represents the structural organization of a system and the messages sent/received. Structural organization consists of objects and links. The purpose of collaboration diagram is similar to sequence diagram. However, the specific purpose of collaboration diagram is to visualize the organization of objects and their interaction.

- **State chart Diagram:**

Any real-time system is expected to be reacted by some kind of internal/external events. These events are responsible for state change of the system. State chart diagram is used to represent the event driven state change of a system. It basically describes the state change of a class, interface, etc.State chart diagram is used to visualize the reaction of a system by internal/external factors.

- **Activity Diagram:**

Activity diagram describes the flow of control in a system. It consists of activities and links. The flow can be sequential, concurrent, or branched.Activities are nothing but the functions of a system. Numbers of activity diagrams are prepared to capture the entire flow in a system.Activity diagrams are used to visualize the flow of controls in a system. This is prepared to have an idea of how the system will work when executed.

Usecase diagrams:

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

Purpose of UseCase Diagrams:

The purpose of use case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose, as other four diagrams (activity, sequence, collaboration, and Statechart) also have the same purpose. We will look into some specific purpose, which will distinguish it from other four diagrams. Use case diagrams are used to gather the requirements of a system including internal and external influences. In brief, the purposes of use case diagrams can be said to be as follows

- Used to gather the requirements of a system.
- Used to get an outside view of a system.

Usecase diagram components:

Actors:

The users that interact with a system. An actor can be a person, an organization, or an outside system that interacts with your application or system. They must be external objects that produce or consume data.

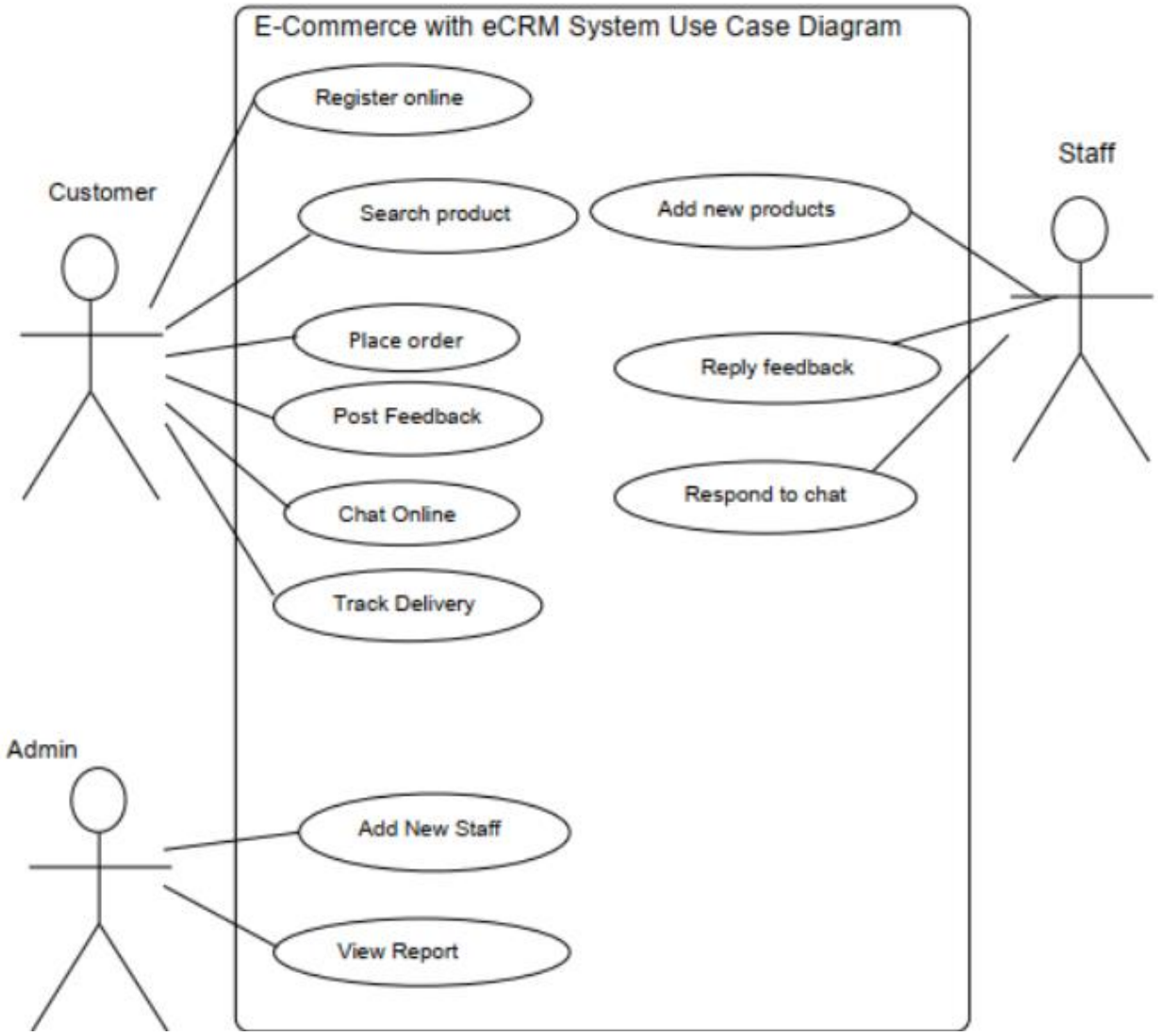
System:

A specific sequence of actions and interactions between actors and the system. A system may also be referred to as a scenario.

Goals:

The end result of most use cases. A successful diagram should describe the activities and variants used to reach the goal.

Usecase diagram for ecommerce:



Sequence diagrams:

The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time. In UML, the lifeline is represented by a vertical bar, whereas the message flow is represented by a vertical dotted line that extends across the bottom of the page. It incorporates the iterations as well as branching.

Purpose of a Sequence Diagram:

1. To model high-level interaction among active objects within a system.
2. To model interaction among objects inside a collaboration realizing a use case.
3. It either models generic interactions or some certain instances of interaction.

Sequence diagram components:

Lifeline:

An individual participant in the sequence diagram is represented by a lifeline. It is positioned at the top of the diagram.

Actor:

A role played by an entity that interacts with the subject is called as an actor. It is out of the scope of the system. It represents the role, which involves human users and external hardware or subjects. An actor may or may not represent a physical entity, but it purely depicts the role of an entity. Several distinct roles can be played by an actor or vice versa.

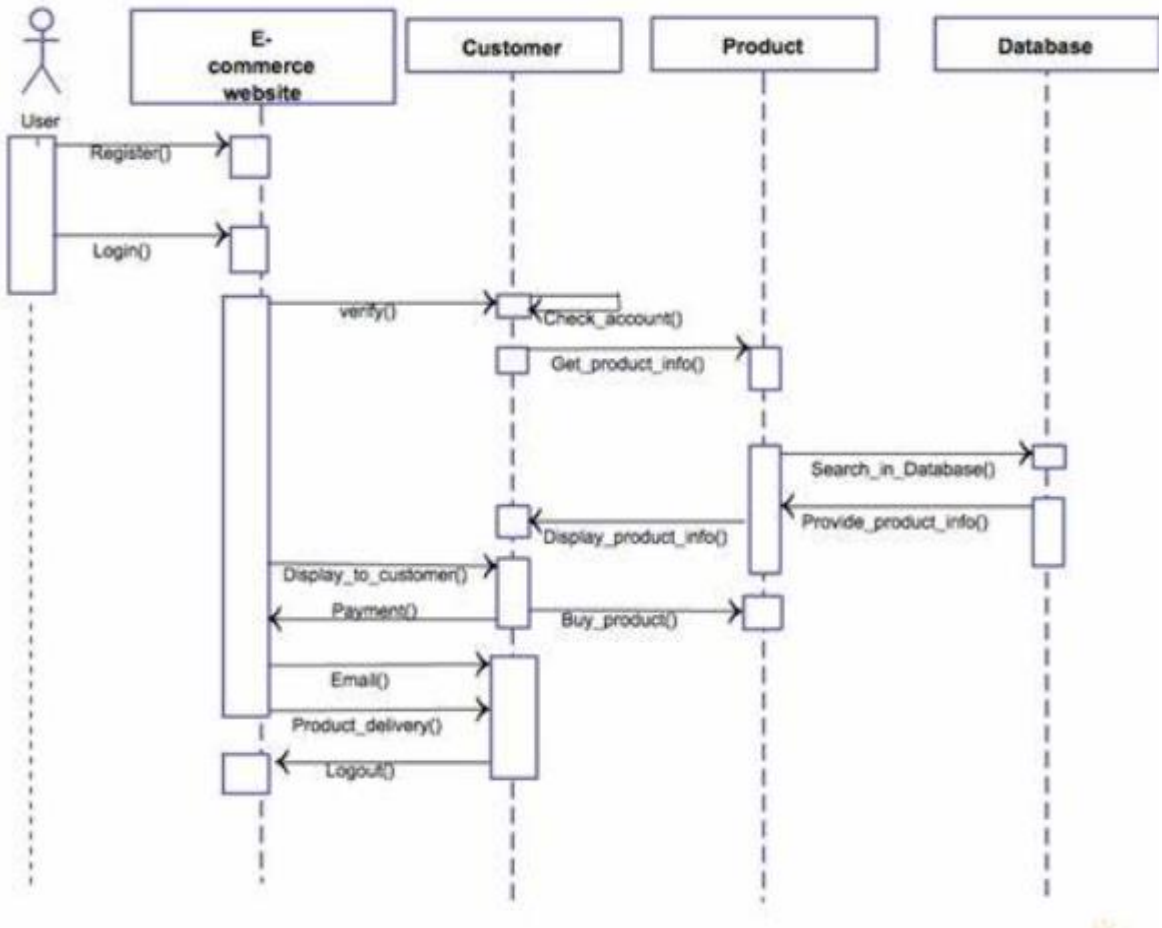
Activation:

It is represented by a thin rectangle on the lifeline. It describes that time period in which an operation is performed by an element, such that the top and the bottom of the rectangle is associated with the initiation and the completion time, each respectively.

Messages:

The messages depict the interaction between the objects and are represented by arrows. They are in the sequential order on the lifeline. The core of the sequence diagram is formed by messages and lifelines.

Sequence diagram for ecommerce:



Collaboration diagram:

The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features. Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system.

Components of a component diagram

- **Objects:**

The representation of an object is done by an object symbol with its name and class underlined separated by colon. In the collaboration diagram, objects are utilized in the following ways: The object is represented by specifying their name and class. It is not mandatory for every class to appear. A class may constitute more than one object. In the collaboration diagram, firstly, the object is created, and then its class is specified. To differentiate one object from another object, it is necessary to name them.

- **Actors:**

In the collaboration diagram, the actor plays the main role as it invokes the interaction. Each actor has its respective role and name. In this, one actor initiates the use case.

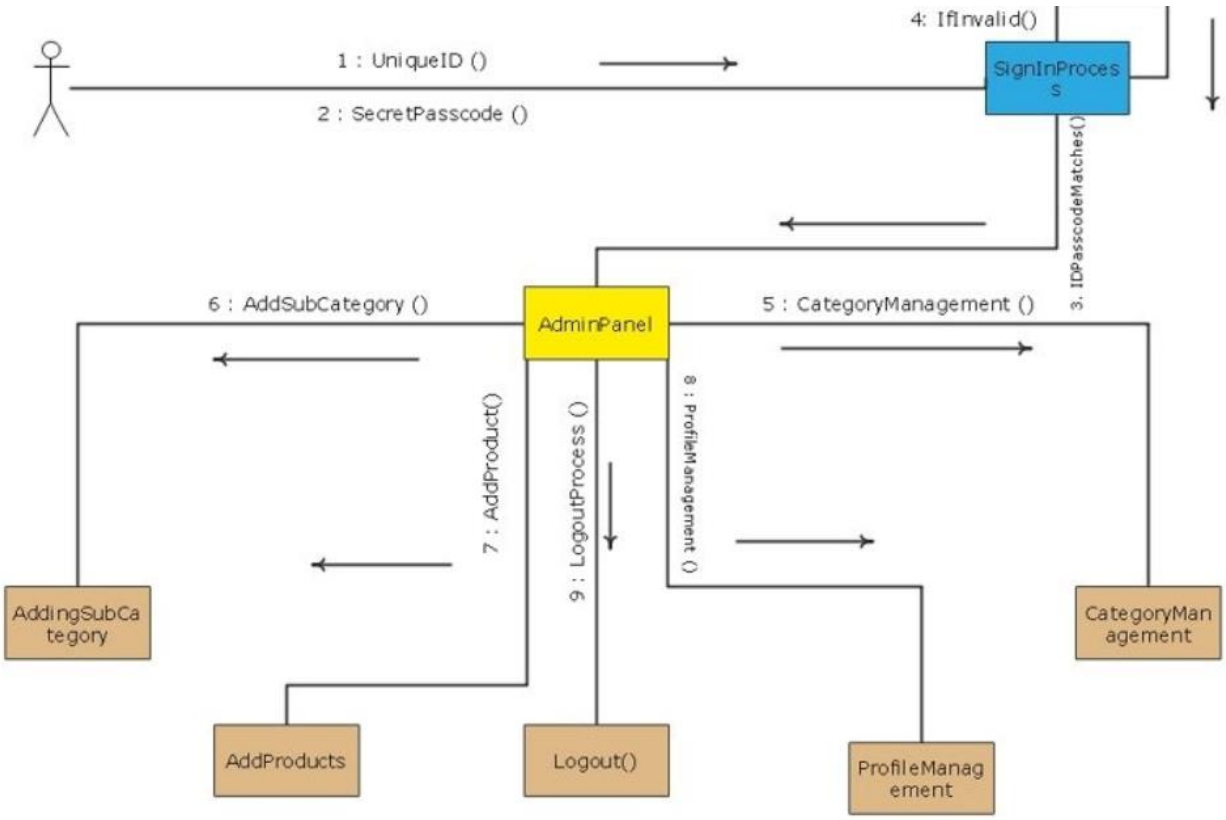
- **Links:**

The link is an instance of association, which associates the objects and actors. It portrays a relationship between the objects through which the messages are sent. It is represented by a solid line. The link helps an object to connect with or navigate to another object, such that the message flows are attached to links.

- **Messages:**

It is a communication between objects which carries information and includes a sequence number, so that the activity may take place. It is represented by a labeled arrow, which is placed near a link.The messages are sent from the sender to the receiver, and the direction must be navigable in that particular direction. The receiver must understand the message.

Collaboration diagram for ecommerce:



Activity Diagram:

In UML, the activity diagram is used to demonstrate the flow of control within the system rather than the implementation. It models the concurrent and sequential activities. The activity diagram helps in envisioning the workflow from one activity to another. It put emphasis on the condition of flow and the order in which it occurs. The flow can be sequential, branched, or concurrent, and to deal with such kinds of flows, the activity diagram has come up with a fork, join, etc.It is also termed as an object-oriented flowchart. It encompasses activities composed of a set of actions or operations that are applied to model the behavioral diagram.

Components of an Activity Diagram

Activities:

The categorization of behavior into one or more actions is termed as an activity. In other words, it can be said that an activity is a network of nodes that are connected by edges. The edges depict the flow of execution. It may contain action nodes, control nodes, or object nodes.

Activity partition /swimlane:

The swimlane is used to cluster all the related activities in one column or one row.It can be either vertical or horizontal. It used to add modularity to the activity diagram. It is not necessary to incorporate swimlane in the activity diagram.

Forks:

Forks and join nodes generate the concurrent flow inside the activity. A fork node consists of one inward edge and several outward edges. It is the same as that of various decision parameters.Whenever a data is received at an inward edge, it gets copied and split crossways various outward edges. It split a single inward flow into multiple parallel flows.

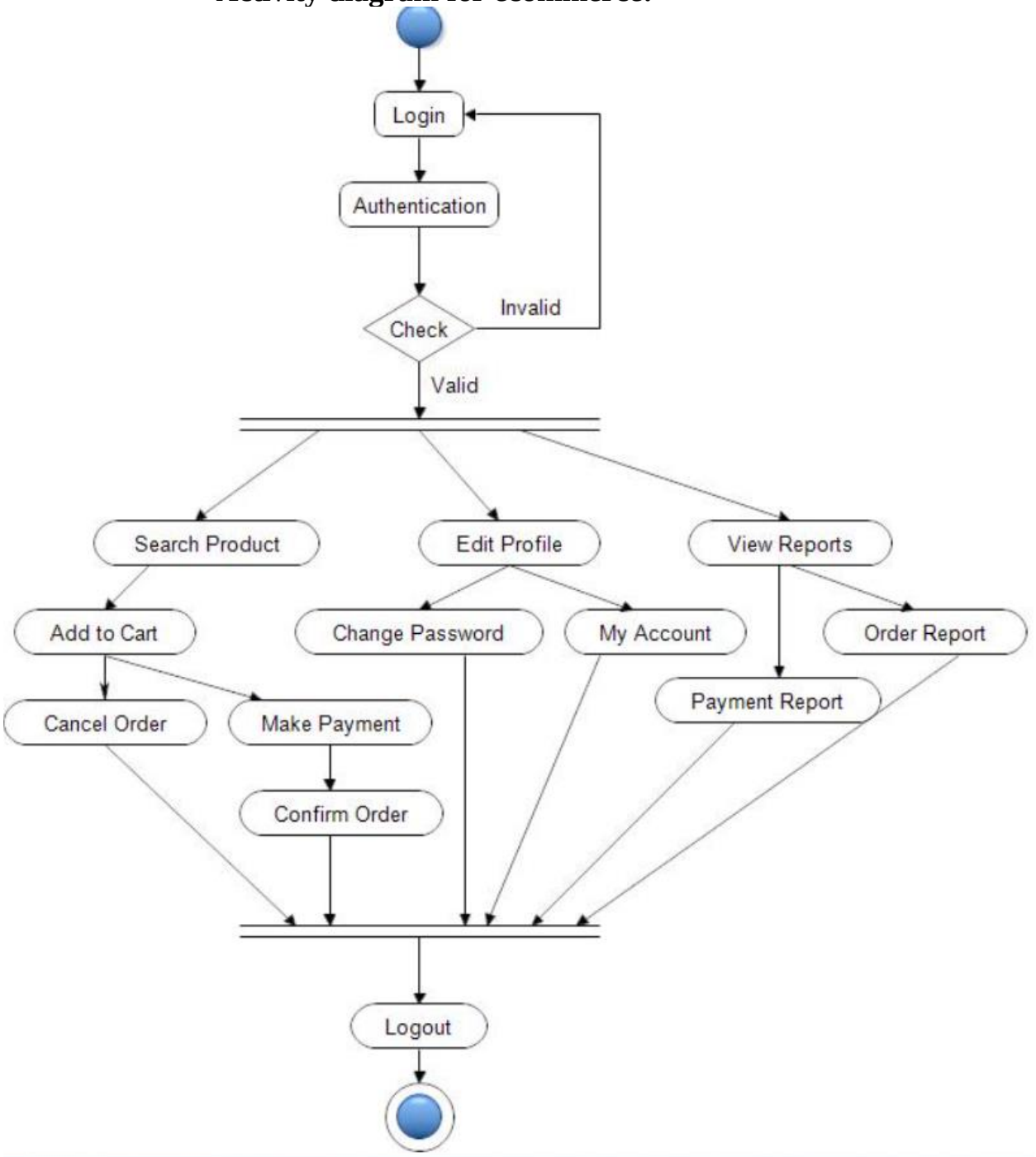
Join Nodes:

Join nodes are the opposite of fork nodes. A Logical AND operation is performed on all of the inward edges as it synchronizes the flow of input across one single output (outward) edge.

Rules:

1. A meaningful name should be given to each and every activity.
2. Identify all of the constraints.
3. Acknowledge the activity associations.

Activity diagram for ecommerce:



Class diagrams:

The class diagram depicts a static view of an application. It represents the types of objects residing in the system and the relationships between them. A class consists of its objects, and also it may inherit from other classes. A class diagram is used to visualize, describe,document various different aspects of the system, and also construct executable software code.

Purpose of Class Diagrams:

The main purpose of class diagrams is to build a static view of an application. It is the only diagram, and it can be mapped with object-oriented languages. It is one of the most popular UML diagrams. Following are the purpose of class diagrams given below:
1. It analyses and designs a static view of an application.
2. It describes the major responsibilities of a system.

Components of a Class Diagram:

Upper Section:

The upper section encompasses the name of the class. A class is a representation of similar objects that shares the same relationships, attributes, operations, and semantics. Some of the following rules that should be taken into account while representing a class are given below:
1.Capitalize the initial letter of the class name.
2. Place the class name in the center of the upper section.
3. A class name must be written in bold format.
.

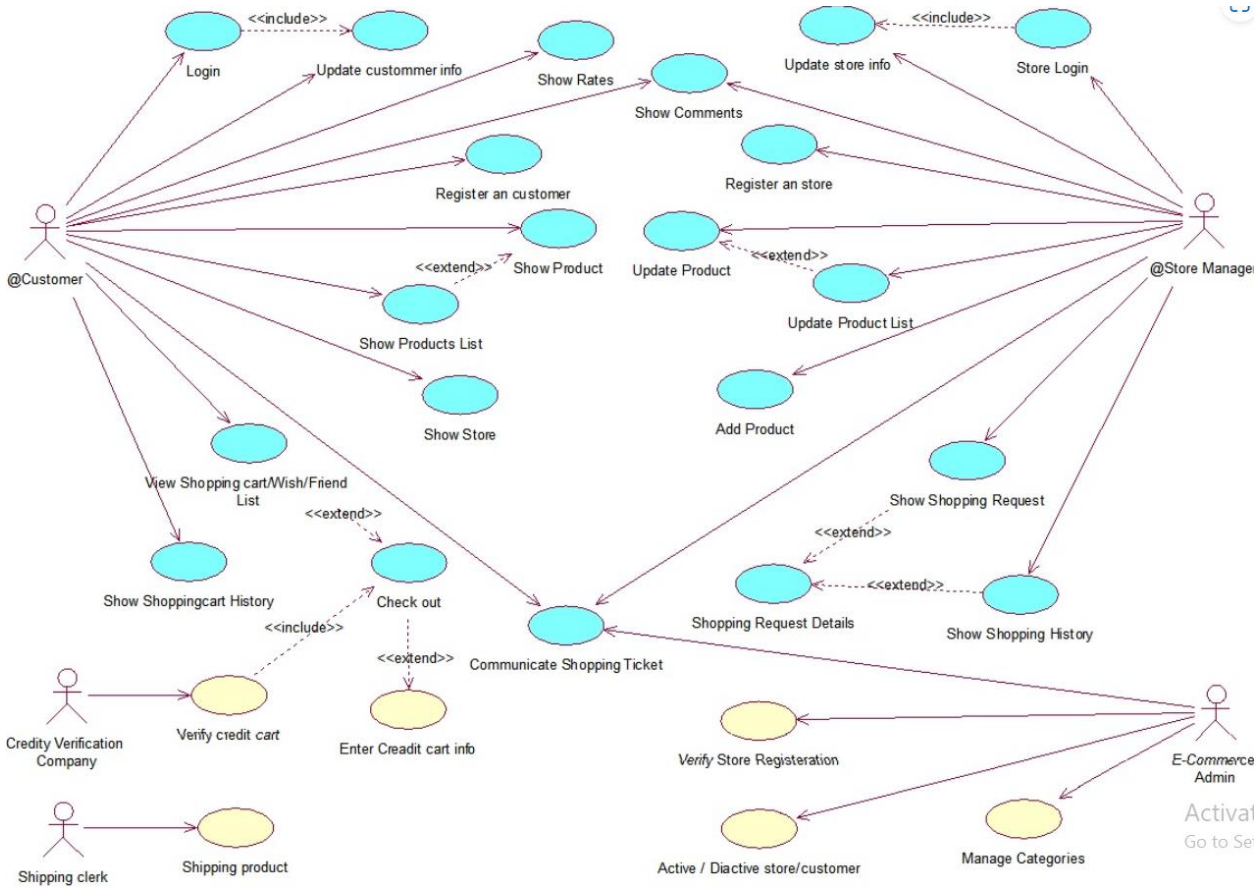
Middle Section:

The middle section constitutes the attributes, which describe the quality of the class. The attributes have the following characteristics:
4. The attributes are written along with its visibility factors, which are public (+), private (-), protected (#), and package (~).
5. The accessibility of an attribute class is illustrated by the visibility factors.
6. A meaningful name should be assigned to the attribute, which will explain its usage inside the class.

Lower Section:

The lower section contain methods or operations. The methods are represented in the form of a list, where each method is written in a single line.

Class diagram for ecommerce:



SYSTEM TESTING

System Testing :

This is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements. In system testing, integration testing passed components are taken as input. The goal of integration testing is to detect any irregularity between the units that are integrated together. System testing detects defects within both the integrated units and the whole system. The result of system testing is the observed behavior of a component or a system when it is tested. E-Commerce and System Design When starting up an e-commerce website, the first thing to decide upon is the type of business model that will be used for the foundation of the website. The internet business model that best applies to Steve's Used Appliances is affiliate marketing. In this model, a business pays off one or more affiliates for their marketing efforts to broaden the scope of customers. (Boris, 2011) This model will be the most helpful for a used appliance store because of the fact that appliances are not easy to sell. Having an affiliate model opens the door for other larger businesses to spread the word on this smaller business. Steve's Used Appliances can definitely benefit from the creation of an e-commerce website. Having a website will make it possible for the company to get out information about their products and services without wasting money on postage, publicizing, or couriers.
System Testing is carried out on the whole system in the context of either system requirement specifications or functional requirement specifications or in the context of both. System testing tests the design and behavior of the system and also the expectations of the customer. It is performed to test the system beyond the bounds mentioned in the software requirements specification (SRS).System Testing is basically performed by a testing team that is independent of the development team that helps to test the quality of the system impartial. It has both functional and non-functional testing. System Testing includes the following steps.

- Verification of input functions of the application to test whether it is producing the expected output or not.
- Testing of integrated software by including external peripherals to check the interaction of various components with each other.
- Testing of the whole system for End to End testing.
- Behavior testing of the application via auser's experience.

Types of System testing

- ## Usability Testing

  Primarily focused on the application's usability, its flexibility in handling controls, and the ability of the system to achieve its goals.

- ## Scalability Testing:

Scalability testing is a kind of software program testing done to validate the overall performance of a software program software or machine in phrases of its capacity to boom or lower the number of personal requests.

- **Hardware/Software Testing:**

Here, the tester concentrates on the interaction between hardware and software during system testing.

- **Recovery Testing :**

Conducted to demonstrate that the software solution is reliable, dependable, and can successfully recover from potential crashes. Migration Testing – Performed to ensure that the software can be easily migrated from the legacy system infrastructure to the current system infrastructure.

- **Stress testing:**

Stress testing is software testing performed to verify the robustness of a system under various loads.

- **Regression Testing:**

Includes tests performed to ensure that changes made during the development process have not introduced new bugs. It also prevents old bugs from appearing as new software modules are added over time.

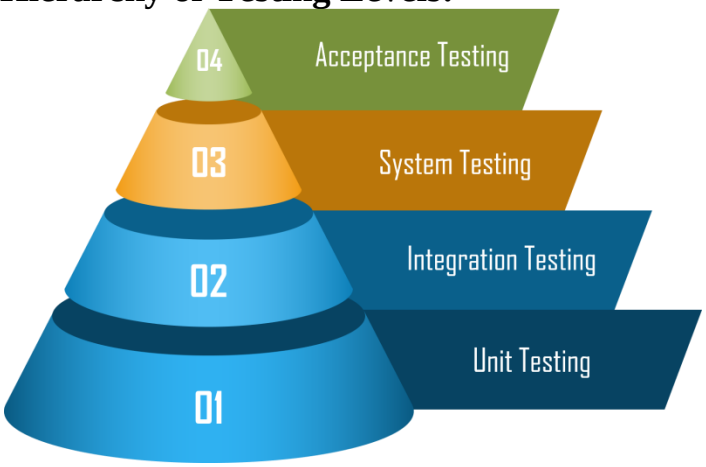
- **Functional Testing:**

Functional testing, also known as functional integrity testing, involves considering possible missing functionality. Testers can create a list of additional features needed to improve the product during functional testing.

- **Performance Testing:**

Performance testing is a type of software testing conducted to test the speed, scalability, stability, and reliability of software products or applications.

Hierarchy of Testing Levels:



- **UNIT TESTING:**

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process. It forms the basis for component testing. Ideally, each test case is independent from the others. Substitutes such as method stubs, mock objects, fakes, and test harnesses can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intend. Unit tests run against each module or block of code during development. Unit testing is usually done by programmers writing code.

- **INTEGRATION TESTING:**

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing. Integration testing before, during, and after the integration of new modules into the main software package. This includes testing individual code modules. The software can often contain multiple modules written by multiple different programmers. It’s important to test the impact of each module on the overall program model.

- **System testing:**

It is done before the finished software product is released to the market. It checks the overall functionality of a product.It is a kind of black box testing.

- **Acceptance Testing:**

It determines whether a system meets acceptance criteria according to user needs, requirements, and business processes, allowing users, customers, or other approved authorities to decide whether to accept the system.

Benefits :

The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict, written contract that the piece of code must satisfy. As a result, it affords several benefits.

- **Find problems early:**

Unit testing finds problems early in the development cycle. In test-driven development (TDD), which is frequently used in both extreme programming and scrum, unit tests are created before the code itself is written. When the tests pass, that code is considered complete. The same unit tests are run against that function frequently as the larger code base is developed either as the code is changed or via an automated process with the build. If the unit tests fail, it is considered to be a bug either in the changed code or the tests themselves. The unit tests then allow the location of the fault or failure to be easily traced. Since the unit tests alert the development team of the problem before handing the code off to testers or clients, it is still early in the development process.

- **Facilitates Change:**

Unit testing allows the programmer to refactor code or upgrade system libraries at a later date, and make sure the module still works correctly (e.g., in regression testing). The procedure is to write test cases for all functions and methods so that whenever a change causes a fault, it can be quickly identified. Unit tests detect changes which may break a design contract.

- **Simplifies Integration:**

Unit testing may reduce uncertainty in the units themselves and can be used in a bottom-up testing style approach. By testing the parts of a program first and then testing the sum of its parts, integration testing becomes much easier.

- **Documentation:**

Unit testing provides a sort of living documentation of the system. Developers looking to learn what functionality is provided by a unit, and how to use it, can look at the unit tests to gain a basic understanding of the unit's interface (API).Unit test cases embody characteristics that are critical to the success of the unit. These characteristics can indicate appropriate/inappropriate use of a unit as well as negative behaviors that are to be trapped by the unit. A unit test case, in and of itself, documents these critical characteristics, although many software development environments do not rely solely upon code to document the product in development.

Purpose:

The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items. These "design items", i.e., assemblages (or groups of units), are exercised through their interfaces using black-box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test whether all the components within assemblages interact correctly, for example across procedure calls or process activations, and this is done after testing individual modules, i.e., unit testing. The overall idea is a "building block" approach, in which verified assemblages are added to a verified base which is then used to support the integration testing of further assemblages.

Software integration testing is performed according to the software development life cycle (SDLC) after module and functional tests. The cross-dependencies for software integration testing are: schedule for integration testing, strategy and selection of the tools used for integration, define the cyclomatical complexity of the software and software architecture, reusability of modules and life-cycle and versioning management.Some different types of integration testing are big-bang, top-down, and bottom-up, mixed (sandwich) and riskyhardest. Other Integration Patterns[2] are: collaboration integration, backbone integration, layer integration, client-server integration, distributed services integration and high-frequency integration.

Big Bang:

In the big-bang approach, most of the developed modules are coupled together to form a complete software system or major part of the system and then used for integration testing. This method is very effective for saving time in the integration testing process. However, if the test cases and their results are not recorded properly, the entire integration process will be more complicated and may prevent the testing team from achieving the goal of integration testing.A type of big-bang integration testing is called "usage model testing" which can be used in both software and hardware integration testing. The basis behind this type of integration testing is to run user-like workloads in integrated user-like environments. In doing the testing in this manner, the environment is proofed, while the individual components are proofed indirectly through their use. Usage Model testing takes an optimistic approach to testing, because it expects to have few

problems with the individual components. The strategy relies heavily on the component developers to do the isolated unit testing for their product. The goal of the strategy is to avoid redoing the testing done by the developers, and instead flesh-out problems caused by the interaction of the components in the environment. For integration testing, Usage Model testing can be more efficient and provides better test coverage than traditional focused functional integration testing. To be more efficient and accurate, care must be used in defining the user-like workloads for creating realistic scenarios in exercising the environment. This gives confidence that the integrated environment will work as expected for the target customers.

Top-down And Bottom-up:

Bottom-up testing is an approach to integrated testing where the lowest level components are tested first, then used to facilitate the testing of higher level components. The process is repeated until the component at the top of the hierarchy is tested.All the bottom or low-level modules, procedures or functions are integrated and then tested. After the integration testing of lower level integrated modules, the next level of modules will be formed and can be used for integration testing. This approach is helpful only when all or most of the modules of the same development level are ready. This method also helps to determine the levels of software developed and makes it easier to report testing progress in the form of a percentage.Top-down 60 testing is an approach to integrated testing where the top integrated modules are tested and the branch of the module is tested step by step until the end of the related module.Sandwich testing is an approach to combine top down testing with bottom up testing.

From Testing Perspective

• Fault – wrong or missing function in the code.
• Failure – the manifestation of a fault during execution.
• Malfunction – according to its specification the system does not meet its specified functionality.

Test Cases:

A test case is a tool used in the process. Test cases may be prepared for software verification and software validation to determine if the product was built according to the requirements of the user. Other methods, such as reviews, may be used early in the life cycle to provide for software validation. Black-Box Testing

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: unit, integration, system and acceptance. It typically comprises most if not all higher level testing, but can also dominate unit testing as well.

Test Procedures :

Specific knowledge of the application's code/internal structure and programming knowledge in general is not required. The tester is aware of what the software is supposed to do but is not

aware of how it does it. For instance, the tester is aware that a particular input returns a certain, invariable output but is not aware of how the software produces the output in the first place.

Test Cases:

Test cases are built around specifications and requirements, i.e., what the application is supposed to do. Test cases are generally derived from external descriptions of the software, including specifications, requirements and design parameters. Although the tests used are primarily functional in nature, non-functional tests may also be used. The test designer selects both valid and invalid inputs and determines the correct output, often with the help of an oracle or a previous result that is known to be good, without any knowledge of the test object's internal structure.

White-Box Testing:

White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g. in-circuit testing (ICT). White-box testing can be applied at the unit, integration and system levels of the software testing process. Although traditional testers tended to think of white-box testing as being done at the unit level, it is used for integration and system testing more frequently today. It can test paths within a unit, paths between units during integration, and between subsystems during a system–level test. Though this method of test design can uncover many errors or problems, it has the potential to miss unimplemented parts of the specification or missing requirements.

Black-Box Testing :

Black box testing is a technique of software testing which examines the functionality of software without peering into its internal structure or coding. The primary source of black box testing is a specification of requirements that is stated by the customer.

In this method, tester selects a function and gives input value to examine its functionality, and checks whether the function is giving expected output or not. If the function produces correct output, then it is passed in testing, otherwise failed. The test team reports the result to the development team and then tests the next function. After completing testing of all functions if there are severe problems, then it is given back to the development team for correction.

Procedures:

White-box testing's basic procedures involves the tester having a deep level of understanding of the source code being tested. The programmer must have a deep understanding of the application to know what kinds of test cases to create so that every visible path is exercised for testing. Once

the source code is understood then the source code can be analyzed for test cases to be created. These are the three basic steps that white-box testing takes in order to create test cases:

• Input involves different types of requirements, functional specifications, detailed designing of documents, proper source code, security specifications. This is the preparation stage of white-box testing to layout all of the basic information.
• Processing involves performing risk analysis to guide whole testing process, proper test plan, execute test cases and communicate results. This is the phase of building test cases to make sure they thoroughly test the application the given results are recorded accordingly.

Real-life example of System testing:

- Suppose a car-making company has already made systems like the ignition system, braking system, engine, fuel system, accelerator, GPS, air conditioner, etc., which were tested individually(Unit testing). The collaboration of these systems is tested(integration testing) for example the fuel system is checked in collaboration with the car engine. But when all systems are ready, the company will check all the systems working by actually driving the car to check if all the systems are working fine with each other. That will be system testing.
- Suppose an e-commerce website has different modules like a registration page, Login page, card, dashboard, payment page, etc. These modules are tested by the developer first, which will be unit testing. The two or more modules will be tested in collaboration with each other. That will be integration testing. But after the completion of all the modules, when the overall website functioning is tested, then that is called system testing.

Advantages of System testing:

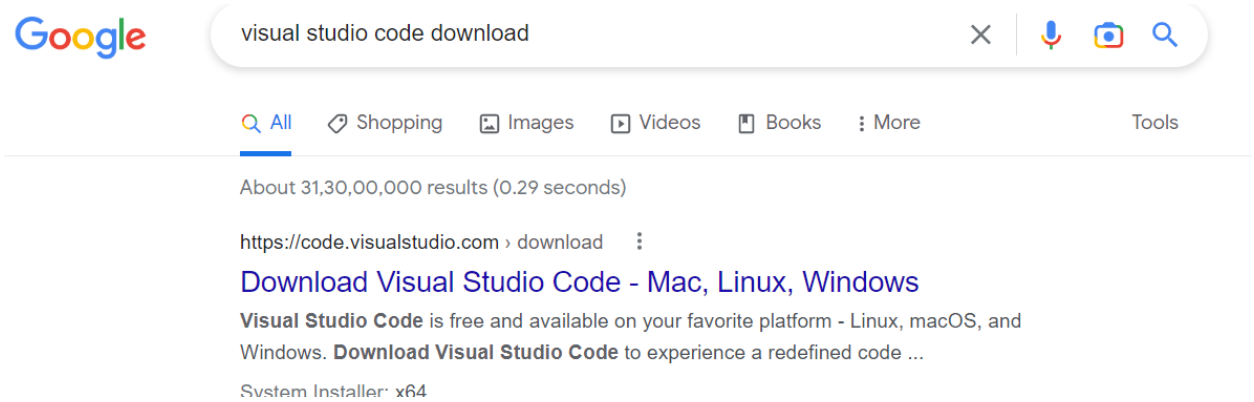
- It includes all software through end-to-end software testing. As a result, you can be sure that the integration of your business requirements and your software architecture is working well.
- Detects errors that occur when your application is released to customers. Build your team's confidence in your product before acceptance testing.
- Since this is a form of black box testing, testers do not need to know programming to complete these tests.
- Various test scripts are used to validate the entire system's functionality and meet the customer's technical and business needs.

Disadvantages of System testing:

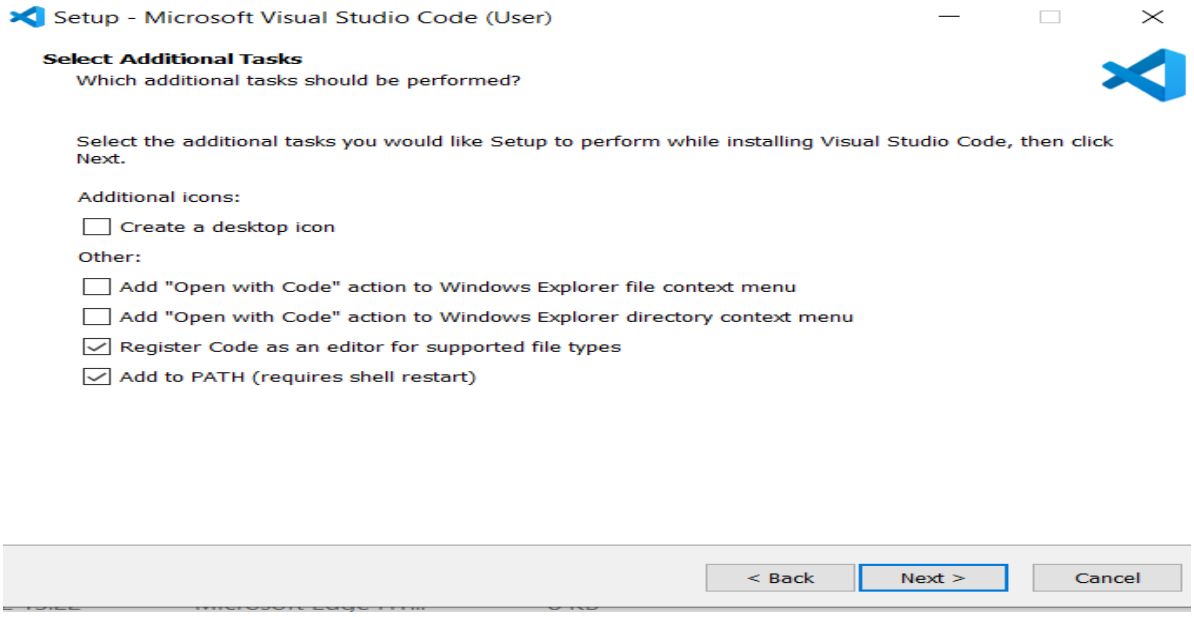
- The effort involved in this testing process is also greater, as business requirements and software architecture must be considered when conducting tests.
- It takesmore time than other types of software testing because it covers the entire software framework.

SCREENSHOTS

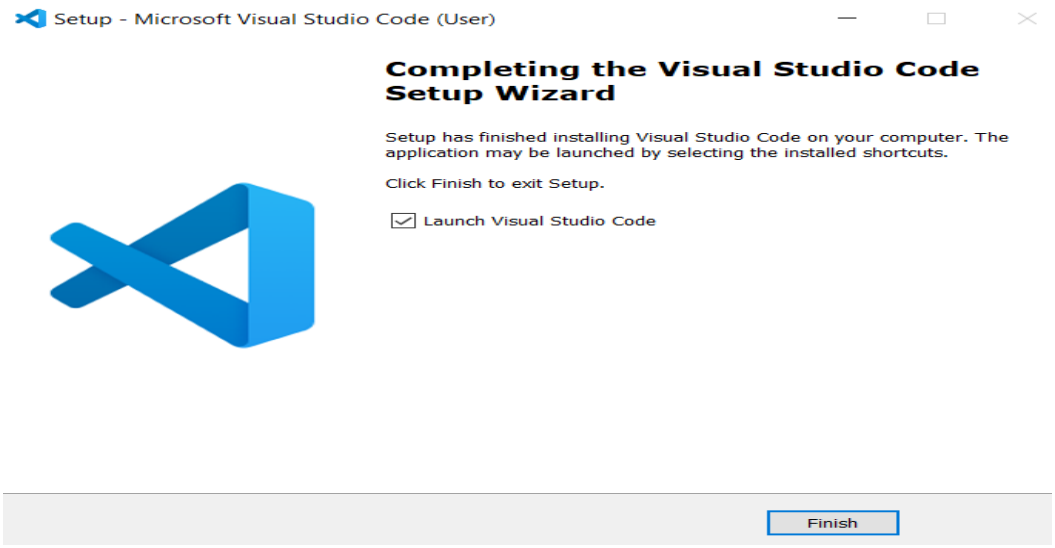
• We can directly download Visual Studio Code to view our code from online as it is a open source



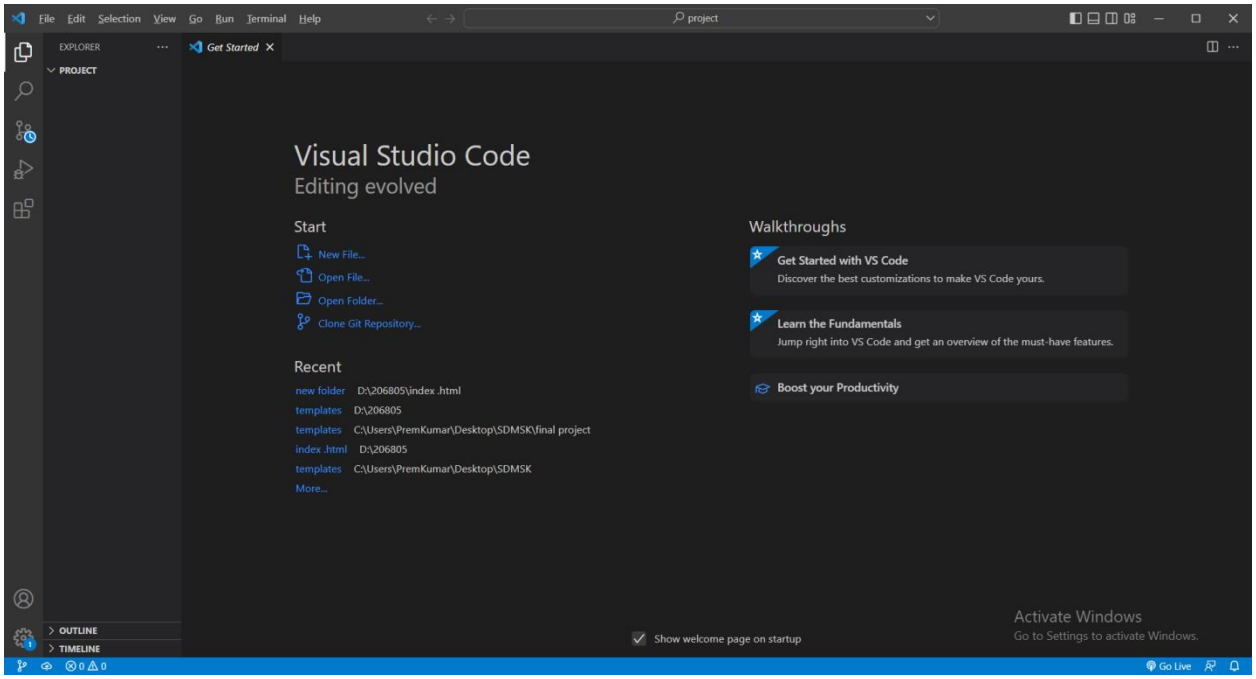
•It shows the destined path of the installed Visual Studio Code



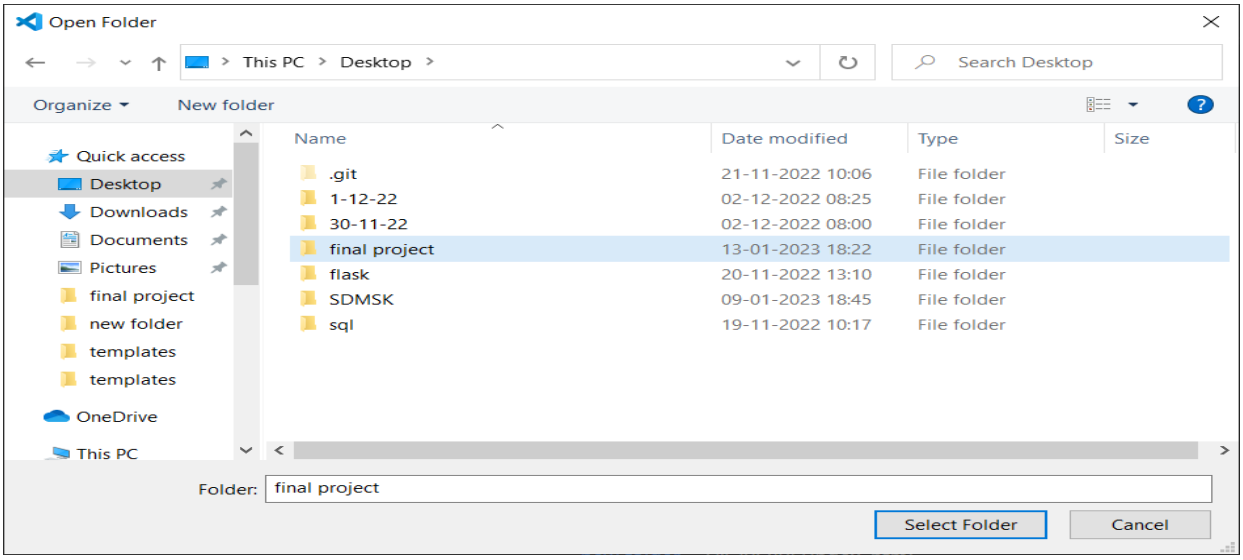
• After the location is set click on finish



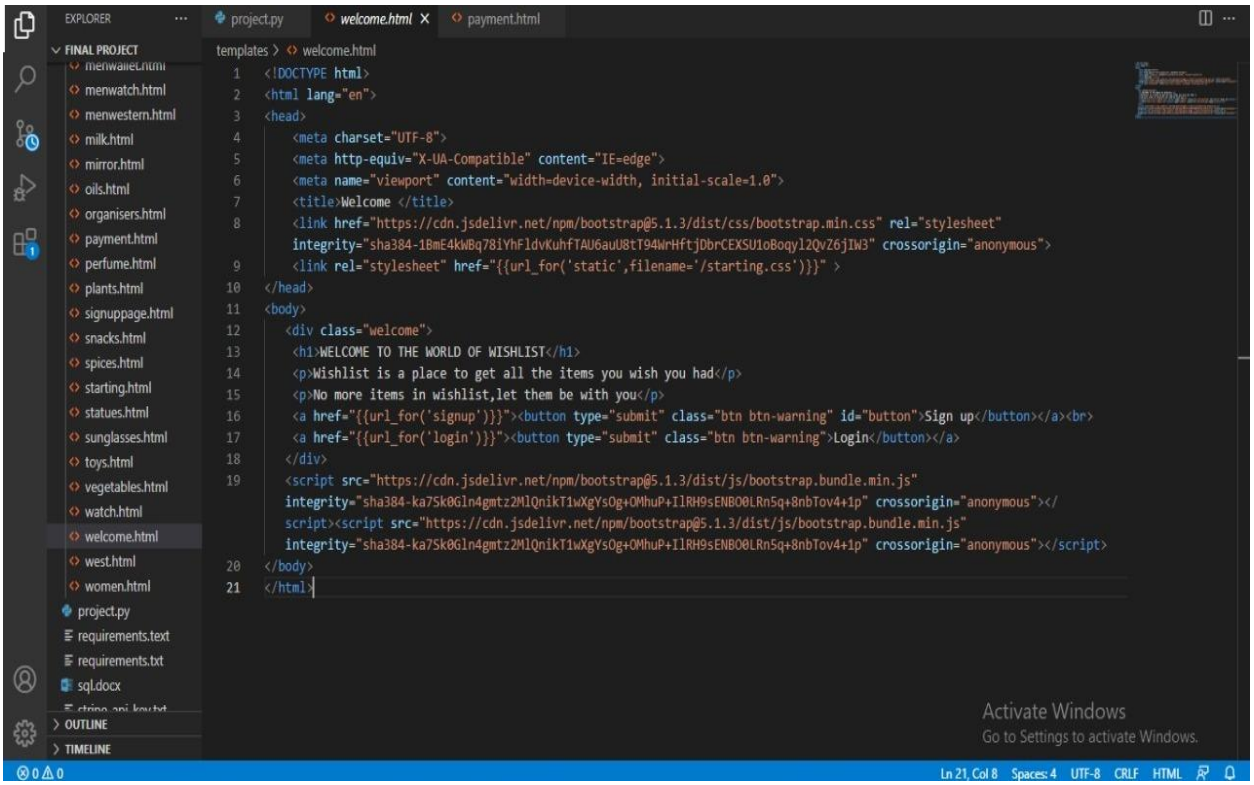
• Open Visual Studio Code

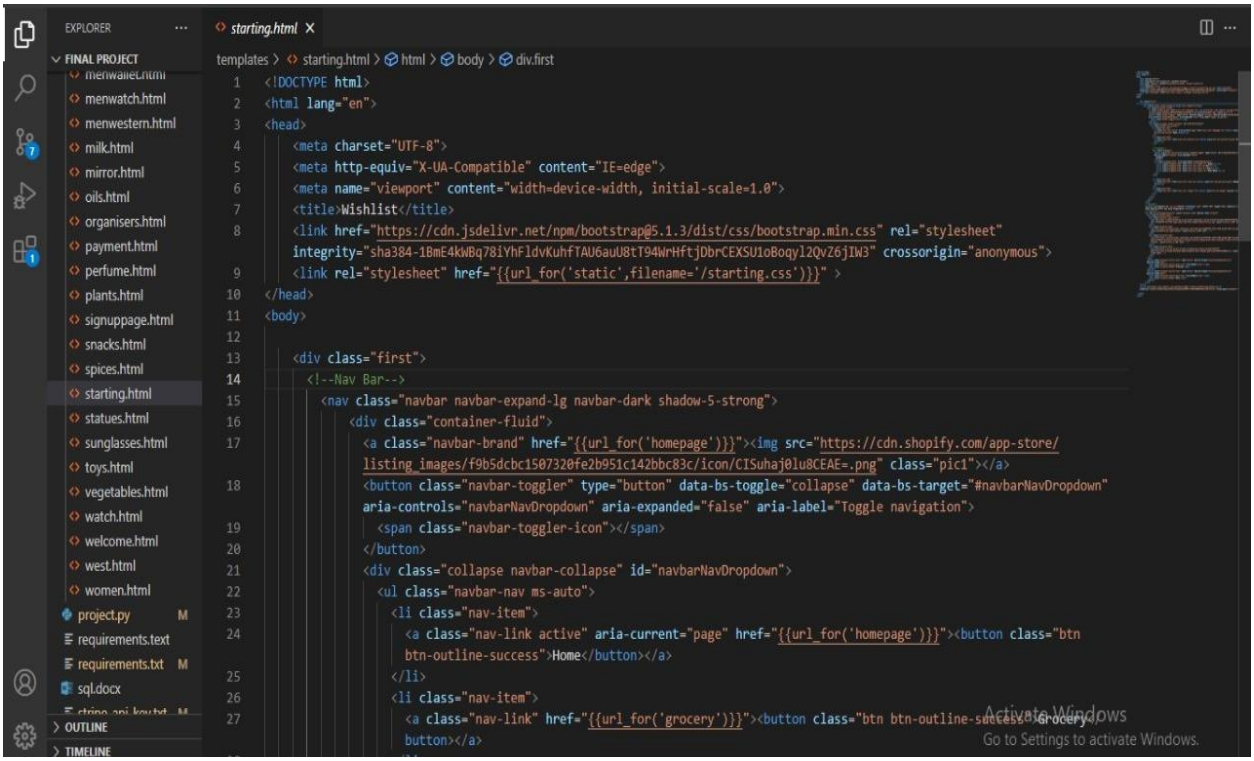


• Browse the folder and select the project



• The Html code is opened in the Visual Studio Code




EXPLORER
starting.html
templates > starting.html > html > body > div.first
FINAL PROJECT
menwatch.html
menwestern.html
milk.html
mirror.html
oils.html
organisers.html
payment.html
perfume.html
plants.html
signuppage.html
snacks.html
spices.html
starting.html
statues.html
sunglasses.html
toys.html
vegetables.html
watch.html
welcome.html
west.html
women.html
project.py
requirements.text
requirements.txt
sql.docx
OUTLINE
TIMELINE
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Wishlist</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
<link rel="stylesheet" href="{{url_for('static',filename='/starting.css')}}" >
</head>
<body>

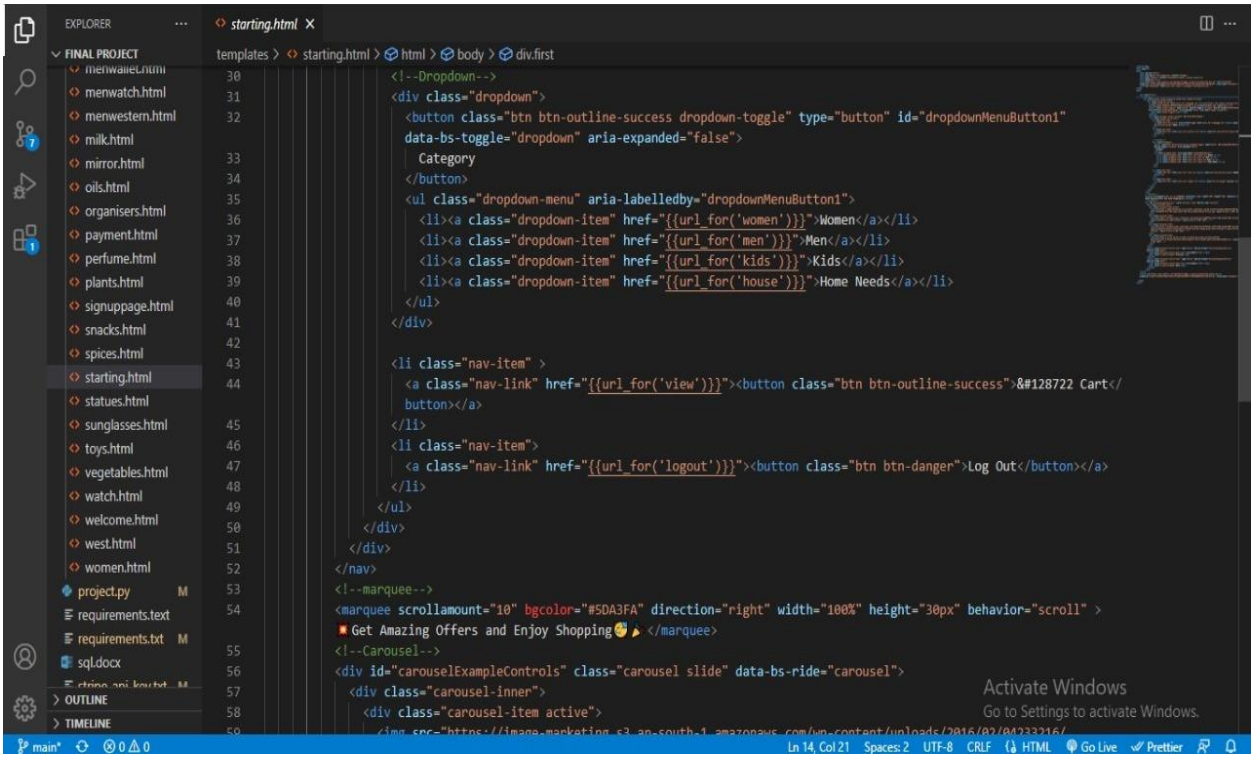
<!--Nav Bar-->
<nav class="navbar navbar-expand-lg navbar-dark shadow-5-strong">

<img src="https://cdn.shopify.com/app-store/
listing_images/f9b5dcbc1507320fe2b951c142bbc83c/icon/CISuhaj0lu8CEAE=.png" class="pic1">
<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNavDropdown"
aria-controls="navbarNavDropdown" aria-expanded="false" aria-label="Toggle navigation">

</button>

<ul class="navbar-nav ms-auto">
<li class="nav-item">
<button class="btn
btn-outline-success">Home</button>

<li class="nav-item">
<button class="btn btn-outline-success">Grocery
button>
Activate Windows
Go to Settings to activate Windows.


EXPLORER
starting.html
templates > starting.html > html > body > div.first
FINAL PROJECT
menwatch.html
menwestern.html
milk.html
mirror.html
oils.html
organisers.html
payment.html
perfume.html
plants.html
signuppage.html
snacks.html
spices.html
starting.html
statues.html
sunglasses.html
toys.html
vegetables.html
watch.html
welcome.html
west.html
women.html
project.py
requirements.text
requirements.txt
sql.docx
OUTLINE
TIMELINE
<!--Dropdown-->

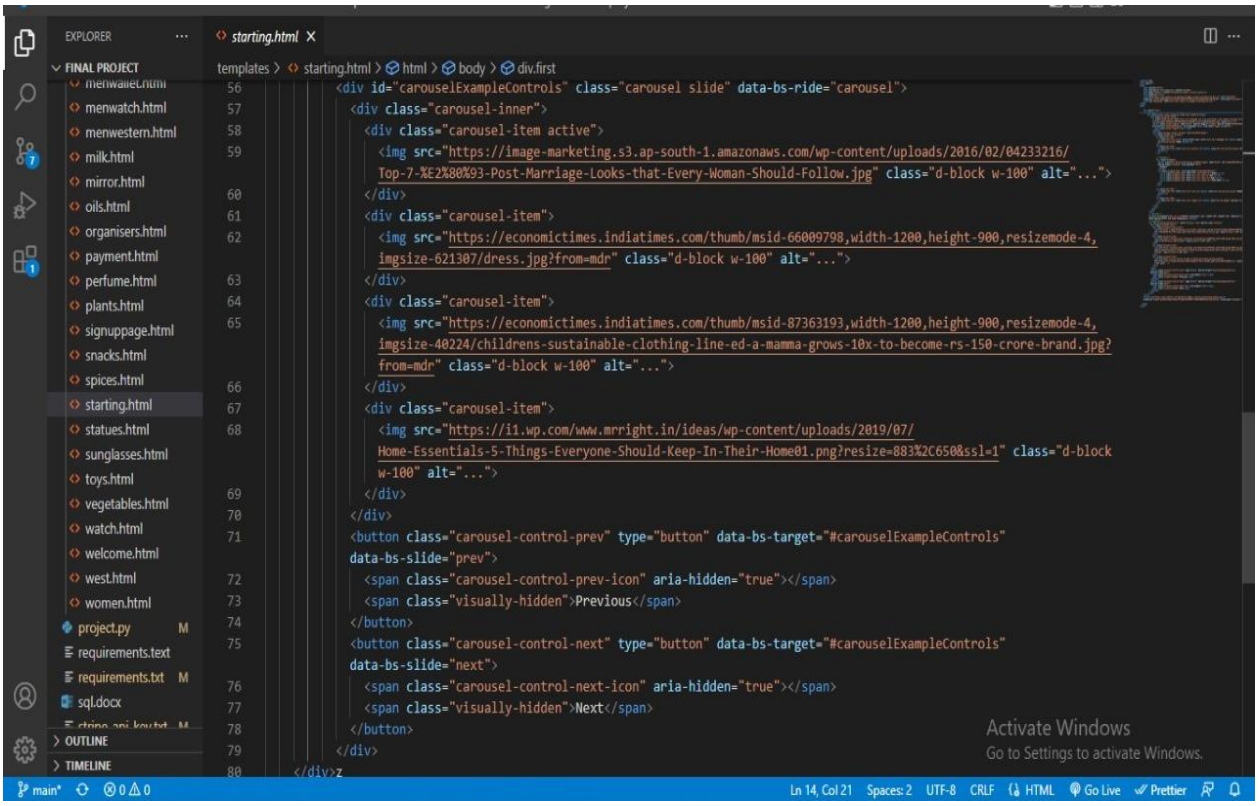
<button class="btn btn-outline-success dropdown-toggle" type="button" id="dropdownMenuButton1"
data-bs-toggle="dropdown" aria-expanded="false">
Category
</button>
<ul class="dropdown-menu" aria-labelledby="dropdownMenuButton1">
Women
Men
Kids
Home Needs

<li class="nav-item" >
<button class="btn btn-outline-success">&#128722 Cart</
button>

<li class="nav-item">
<button class="btn btn-danger">Log Out</button>

</nav>
<!--marquee-->
<marquee scrollamount="10" bgcolor="#5DA3FA" direction="right" width="100%" height="30px" behavior="scroll" >
Get Amazing Offers and Enjoy Shopping
</marquee>
<!--Carousel-->

Activate Windows
Go to Settings to activate Windows.
main*
Ln 14, Col 21
Spaces: 2
UTF-8
CRLF
HTML
Go Live
Prettier


EXPLORER
starting.html
FINAL PROJECT
templates > starting.html > html > body > div.first
menwatch.html
menwestern.html
milk.html
mirror.html
oils.html
organisers.html
payment.html
perfume.html
plants.html
signuppage.html
snacks.html
spices.html
starting.html
statues.html
sunglasses.html
toys.html
vegetables.html
watch.html
welcome.html
west.html
women.html
project.py
requirements.text
requirements.txt
sql.docx
OUTLINE
TIMELINE

<img src="https://image-marketing.s3.ap-south-1.amazonaws.com/wp-content/uploads/2016/02/04233216/Top-7-%E2%80%93-Post-Marriage-Looks-that-Every-Woman-Should-Follow.jpg" class="d-block w-100" alt="...">

<img src="https://economictimes.indiatimes.com/thumb/msid-66009798,width-1200,height-900,resizemode-4,imgsize-621307/dress.jpg?from=mdr" class="d-block w-100" alt="...">

<img src="https://economictimes.indiatimes.com/thumb/msid-87363193,width-1200,height-900,resizemode-4,imgsize-40224/childrens-sustainable-clothing-line-ed-a-mamma-grows-10x-to-become-rs-150-crore-brand.jpg?from=mdr" class="d-block w-100" alt="...">

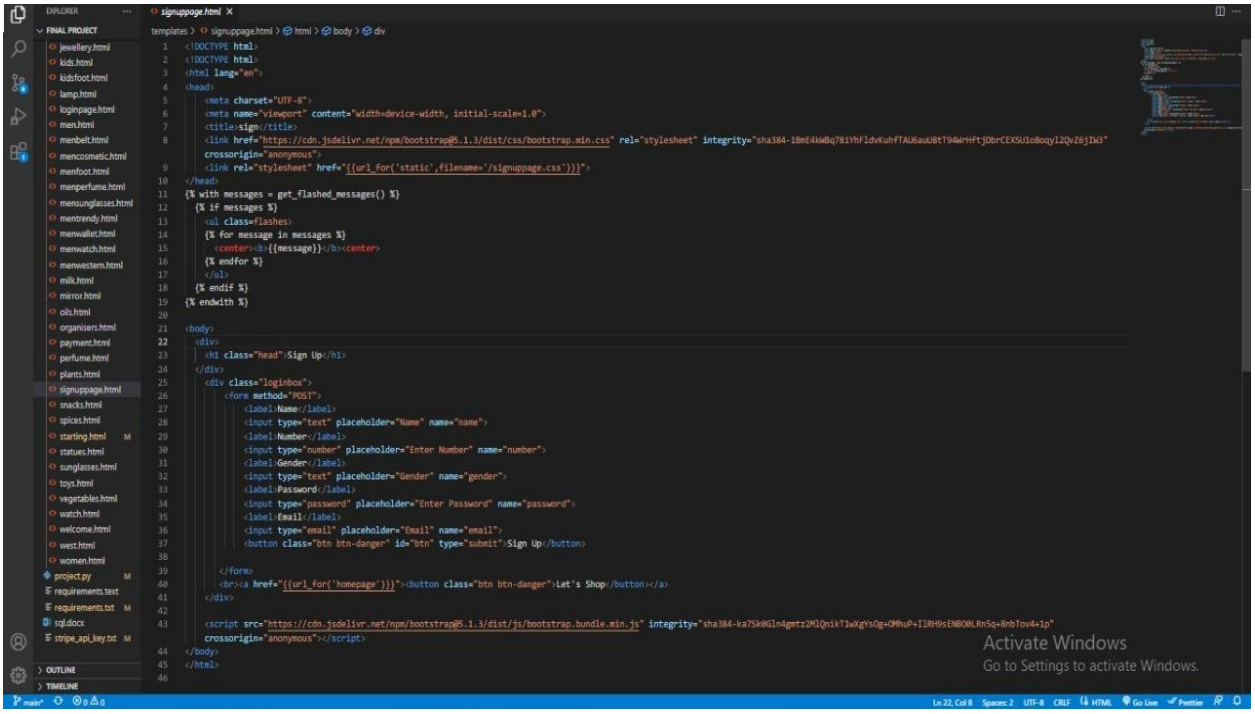
<img src="https://i1.wp.com/www.mrright.in/ideas/wp-content/uploads/2019/07/Home-Essentials-5-Things-Everyone-Should-Keep-In-Their-Home01.png?resize=883%2C650&ssl=1" class="d-block w-100" alt="...">

<button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleControls" data-bs-slide="prev">

Previous
</button>
<button class="carousel-control-next" type="button" data-bs-target="#carouselExampleControls" data-bs-slide="next">

Next
</button>

z
Activate Windows
Go to Settings to activate Windows.
main*
Ln 14, Col 21
Spaces: 2
UTF-8
CRLF
HTML
Go Live
Prettier


signuppage.html
templates > signuppage.html > html > body > div
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>sign</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
<link rel="stylesheet" href="{{url_for('static',filename='/signuppage.css')}}">
</head>
{% with messages = get_flashed_messages() %}
{% if messages %}
<ul class=flashes>
{% for message in messages %}
{{message}}
{% endfor %}

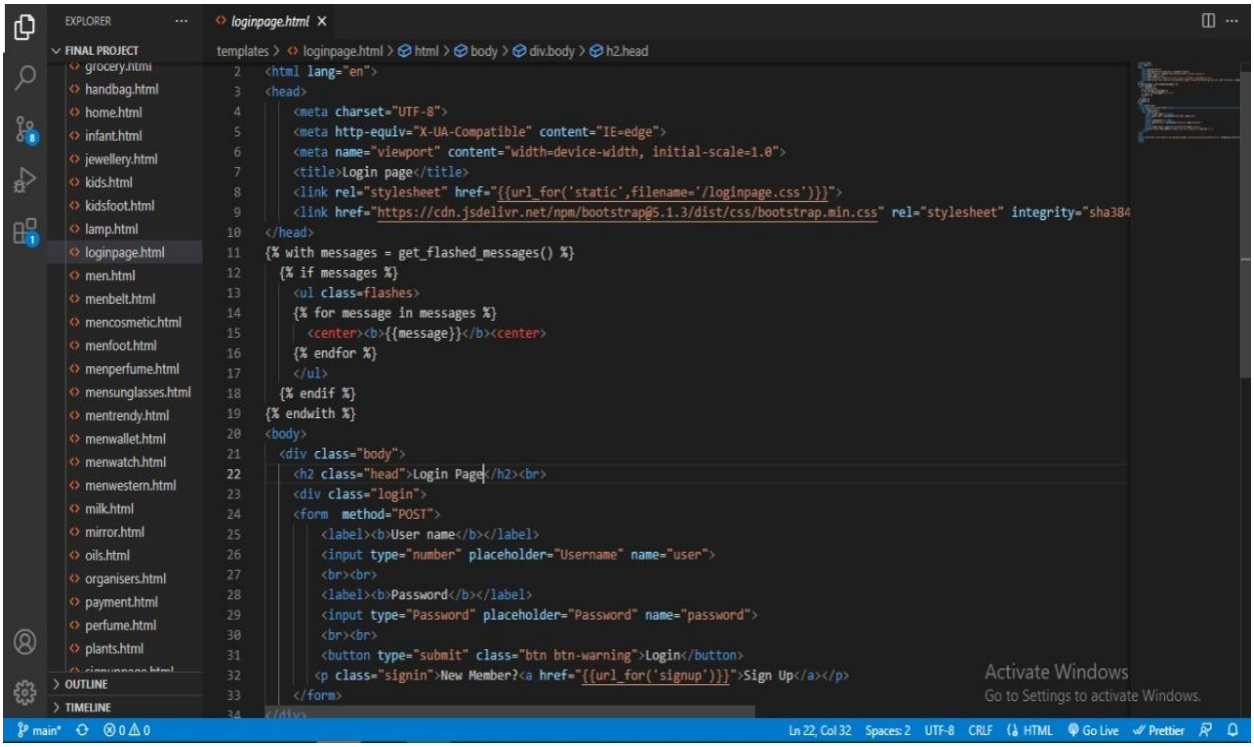
{% endif %}
{% endwith %}
<body>

<h1 class="head">Sign Up</h1>

<form method="POST">
<label>Name</label>
<input type="text" placeholder="Name" name="name">
<label>Number</label>
<input type="number" placeholder="Enter Number" name="number">
<label>Gender</label>
<input type="text" placeholder="Gender" name="gender">
<label>Password</label>
<input type="password" placeholder="Enter Password" name="password">
<label>Email</label>
<input type="email" placeholder="Email" name="email">
<button class="btn btn-danger" id="btn" type="submit">Sign Up</button>
</form>

<button class="btn btn-danger">Let's Shop</button>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p" crossorigin="anonymous"></script>
</body>
</html>
Activate Windows
Go to Settings to activate Windows.


loginpage.html
templates > loginpage.html > html > body > div.body > h2.head
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Login page</title>
<link rel="stylesheet" href="{{url_for('static',filename='/loginpage.css')}}">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384
</head>
{% with messages = get_flashed_messages() %}
{% if messages %}
<ul class=flashes>
{% for message in messages %}
{{message}}
{% endfor %}

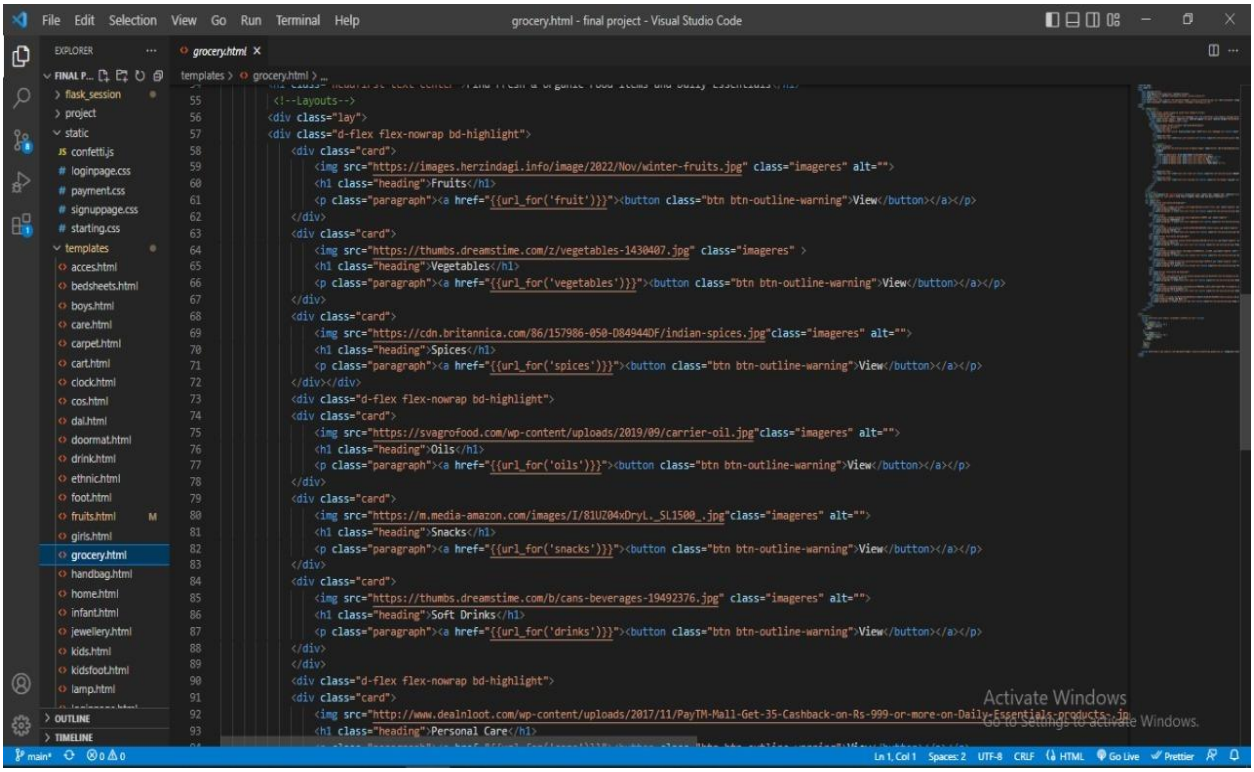
{% endif %}
{% endwith %}
<body>

<h2 class="head">Login Page</h2>

<form method="POST">
<label>User name</label>
<input type="number" placeholder="Username" name="user">

<label>Password</label>
<input type="Password" placeholder="Password" name="password">

<button type="submit" class="btn btn-warning">Login</button>
New Member?Sign Up
</form>
Activate Windows
Go to Settings to activate Windows.


grocery.html - final project - Visual Studio Code
<!--Layouts-->

<img src="https://images.herzindagi.info/image/2022/Nov/winter-fruits.jpg" class="imageres" alt="">
<h1 class="heading">Fruits</h1>
<button class="btn btn-outline-warning">View</button>

<img src="https://thumbs.dreamstime.com/z/vegetables-1430407.jpg" class="imageres" >
<h1 class="heading">Vegetables</h1>
<button class="btn btn-outline-warning">View</button>

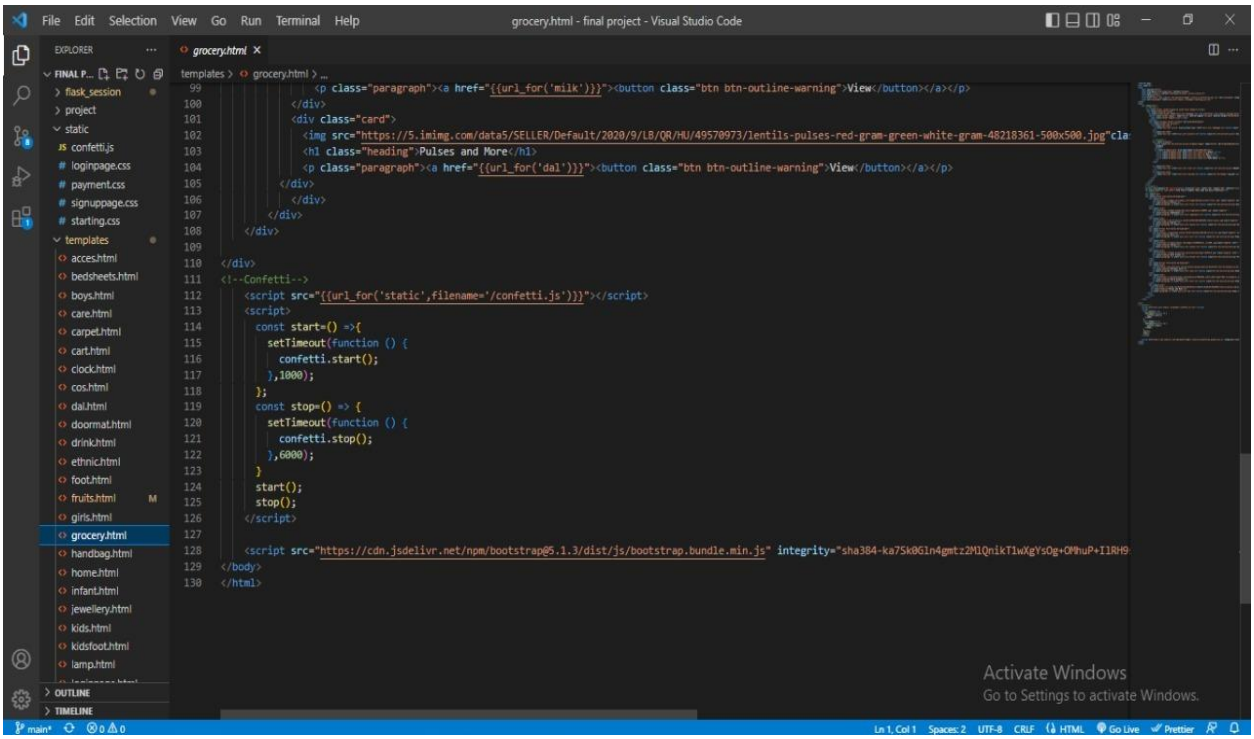
<img src="https://cdn.britannica.com/86/157986-050-D84944DF/indian-spices.jpg"class="imageres" alt="">
<h1 class="heading">Spices</h1>
<button class="btn btn-outline-warning">View</button>

<img src="https://svagrofood.com/wp-content/uploads/2019/09/carrier-oil.jpg"class="imageres" alt="">
<h1 class="heading">Oils</h1>
<button class="btn btn-outline-warning">View</button>

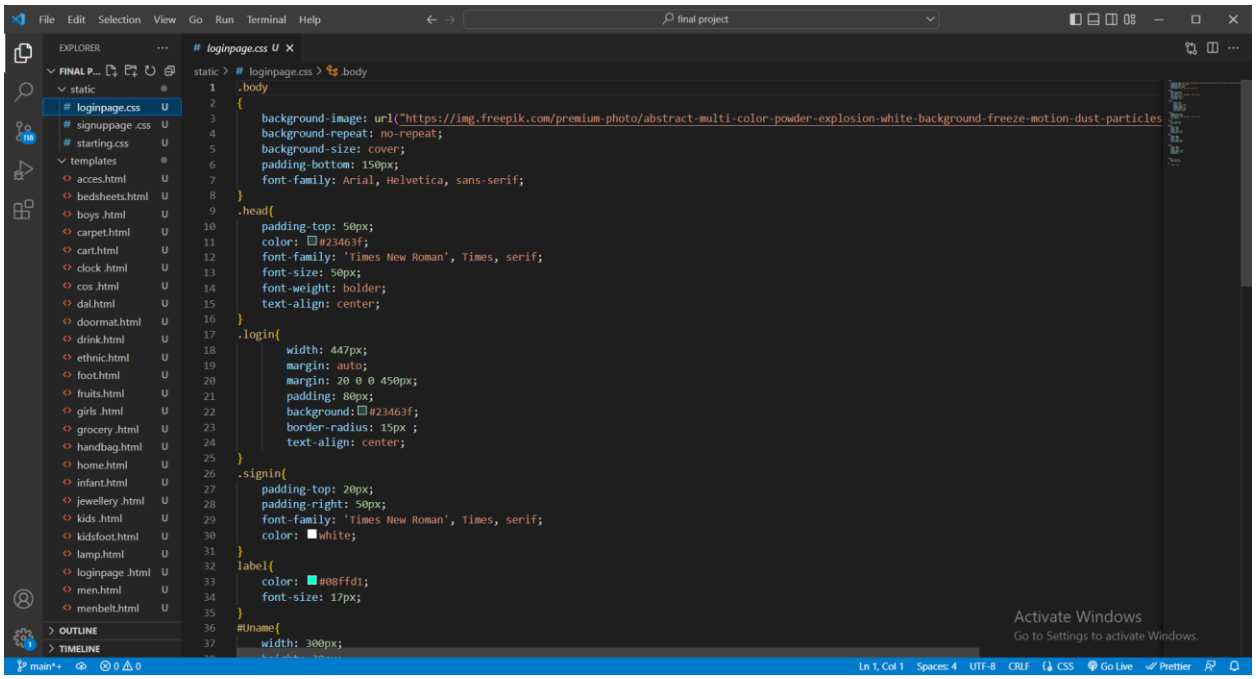
<img src="https://m.media-amazon.com/images/I/81U704xDryL._SL1500_.jpg"class="imageres" alt="">
<h1 class="heading">Snacks</h1>
<button class="btn btn-outline-warning">View</button>

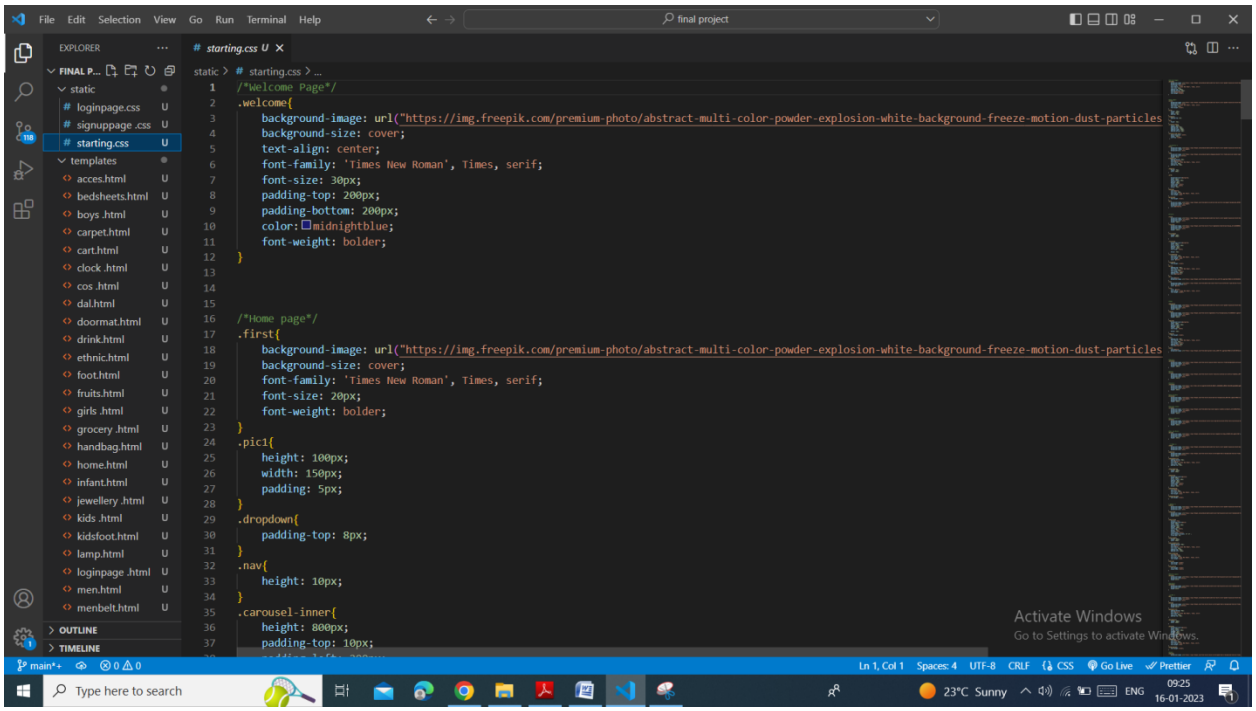
<img src="https://thumbs.dreamstime.com/b/cans-beverages-19492376.jpg" class="imageres" alt="">
<h1 class="heading">Soft Drinks</h1>
<button class="btn btn-outline-warning">View</button>

<img src="http://www.dealnloot.com/wp-content/uploads/2017/11/PayTM-Mall-Get-35-Cashback-on-Rs-999-or-more-on-Daily-Essentials-products-.jp
<h1 class="heading">Personal Care</h1>
Activate Windows
Go to Settings to activate Windows.

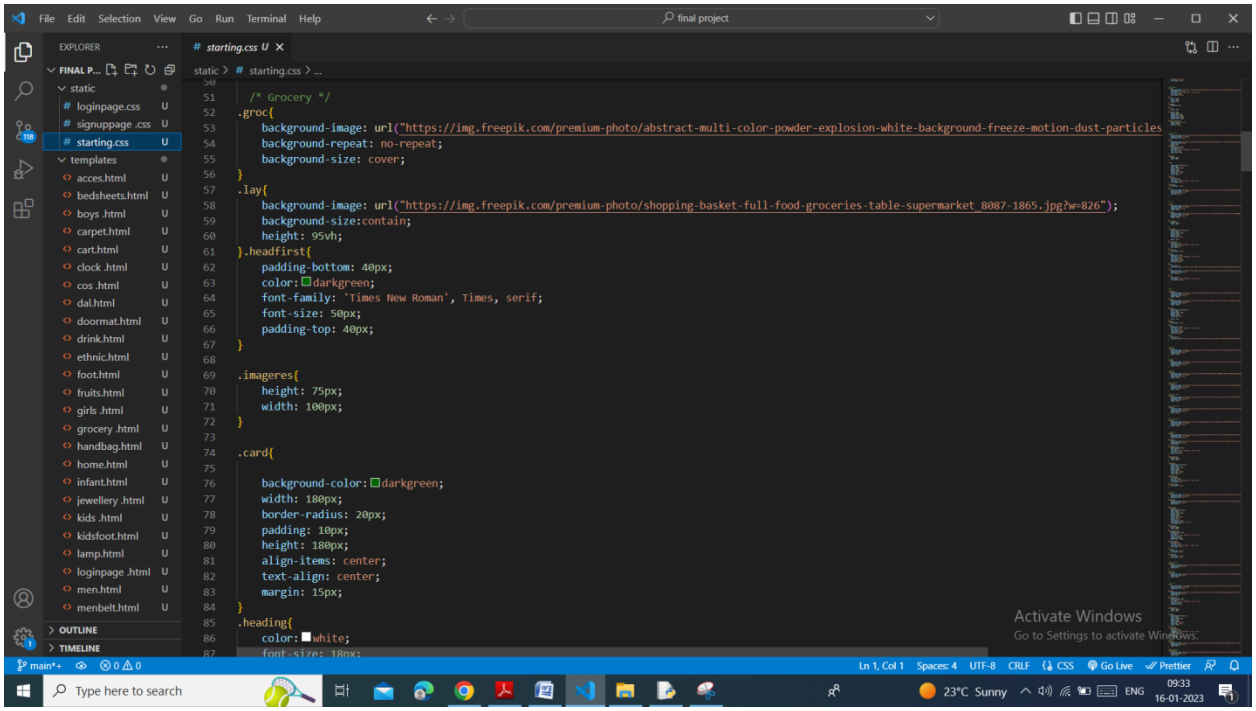


• The Css code is opened in the Visual Studio Code



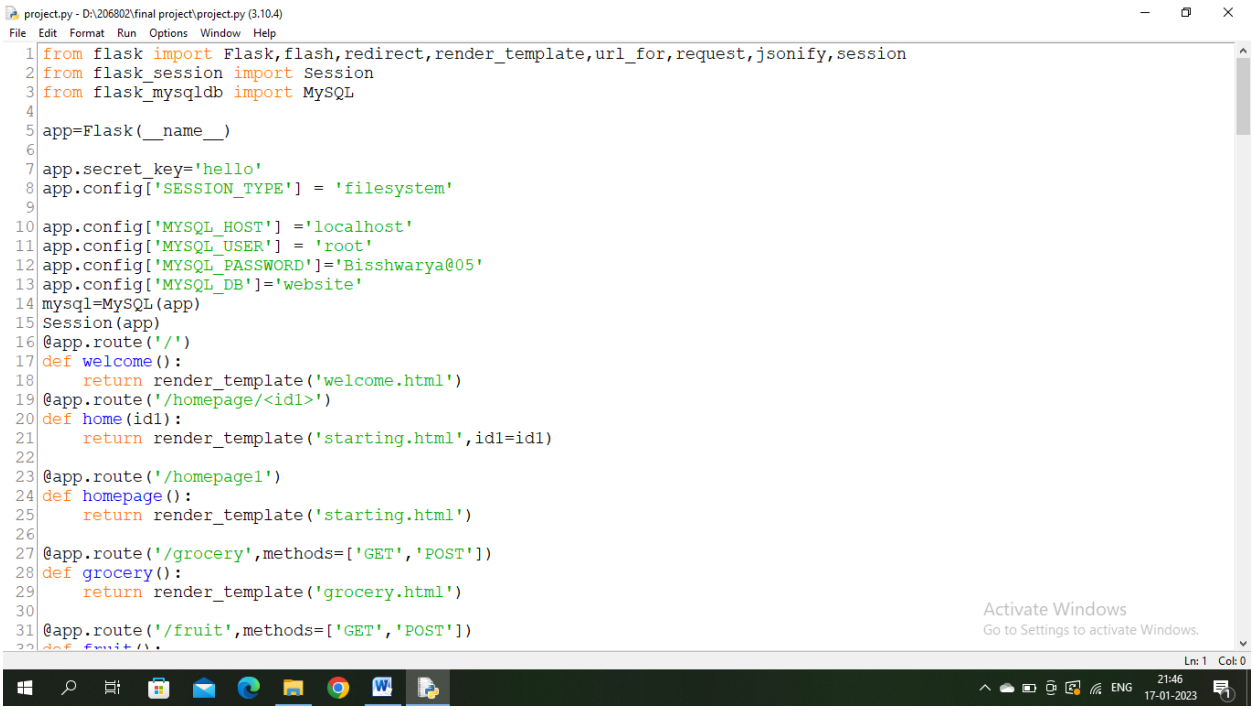


```
/*Welcome Page*/
.welcome{
    background-image: url("https://img.freepik.com/premium-photo/abstract-multi-color-powder-explosion-white-background-freeze-motion-dust-particles
    background-size: cover;
    text-align: center;
    font-family: 'Times New Roman', Times, serif;
    font-size: 30px;
    padding-top: 200px;
    padding-bottom: 200px;
    color: midnightblue;
    font-weight: bolder;
}


/*Home page*/
.first{
    background-image: url("https://img.freepik.com/premium-photo/abstract-multi-color-powder-explosion-white-background-freeze-motion-dust-particles
    background-size: cover;
    font-family: 'Times New Roman', Times, serif;
    font-size: 20px;
    font-weight: bolder;
}
.pic1{
    height: 100px;
    width: 150px;
    padding: 5px;
}
.dropdown{
    padding-top: 8px;
}
.nav{
    height: 10px;
}
.carousel-inner{
    height: 800px;
    padding-top: 10px;
```

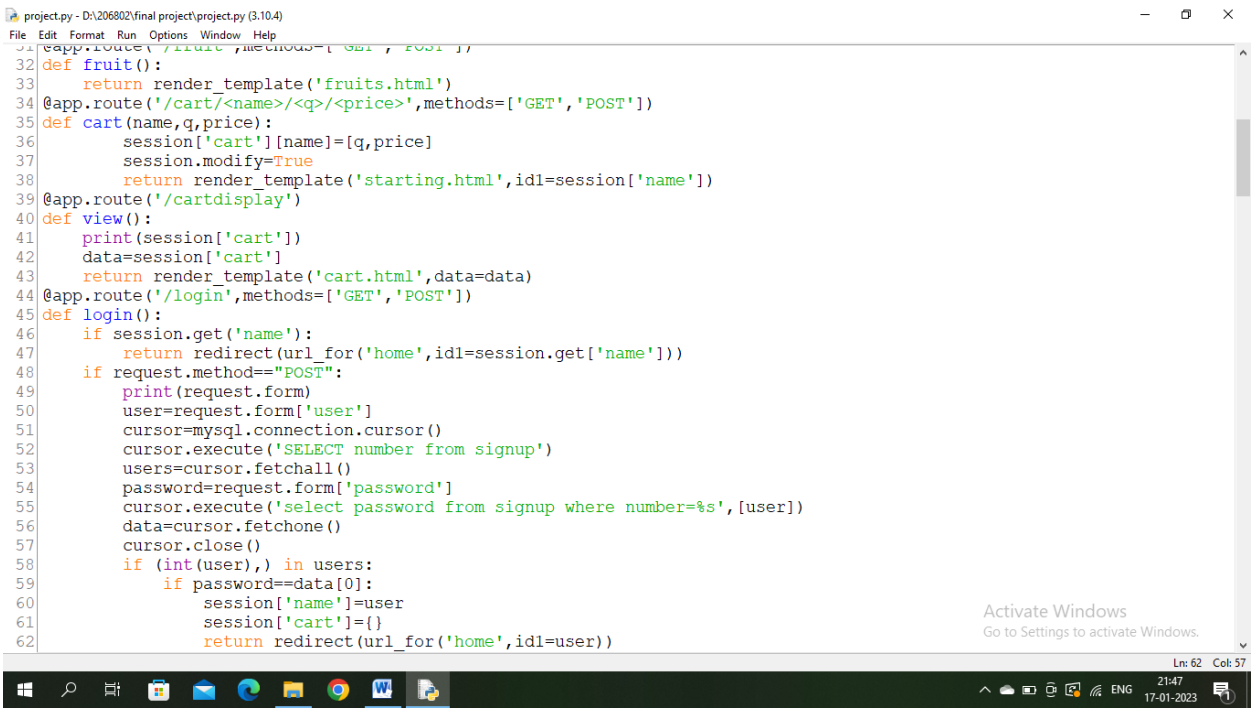


```
  /* Grocery */
.groc{
    background-image: url("https://img.freepik.com/premium-photo/abstract-multi-color-powder-explosion-white-background-freeze-motion-dust-particles
    background-repeat: no-repeat;
    background-size: cover;
}
.lay{
    background-image: url("https://img.freepik.com/premium-photo/shopping-basket-full-food-groceries-table-supermarket_8087-1865.jpg?w=826");
    background-size:contain;
    height: 95vh;
}.headfirst{
    padding-bottom: 40px;
    color: darkgreen;
    font-family: 'Times New Roman', Times, serif;
    font-size: 50px;
    padding-top: 40px;
}

.imageres{
    height: 75px;
    width: 100px;
}

.card{

    background-color: darkgreen;
    width: 180px;
    border-radius: 20px;
    padding: 10px;
    height: 180px;
    align-items: center;
    text-align: center;
    margin: 15px;
}
.heading{
    color: white;
    font-size: 18px;
```

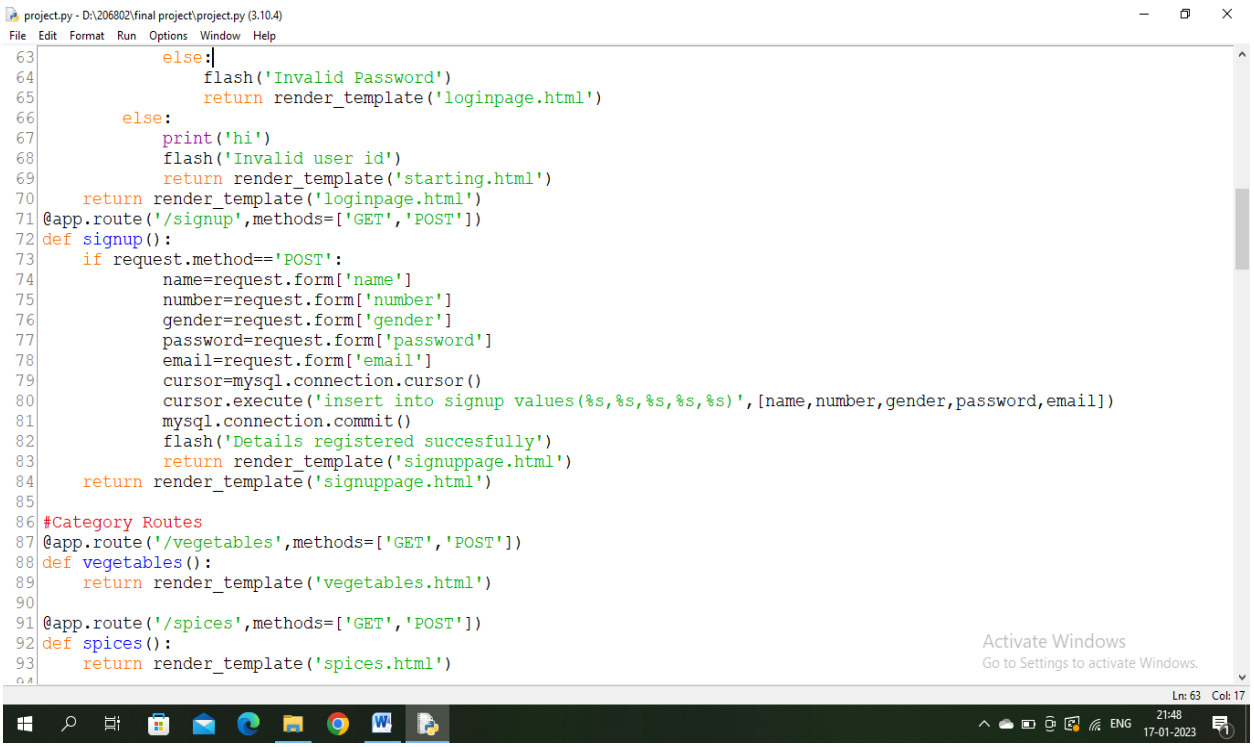
• The Python code is opened in the IDLE



```
project.py - D:\206802\final project\project.py (3.10.4)
File  Edit  Format  Run  Options  Window  Help
from flask import Flask,flash,redirect,render_template,url_for,request,jsonify,session
from flask_session import Session
from flask_mysqldb import MySQL

app=Flask(__name__)

app.secret_key='hello'
app.config['SESSION_TYPE'] = 'filesystem'

app.config['MYSQL_HOST'] ='localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD']='Bisshwarya@05'
app.config['MYSQL_DB']='website'
mysql=MySQL(app)
Session(app)
@app.route('/')
def welcome():
    return render_template('welcome.html')
@app.route('/homepage/<id1>')
def home(id1):
    return render_template('starting.html',id1=id1)

@app.route('/homepage1')
def homepage():
    return render_template('starting.html')

@app.route('/grocery',methods=['GET','POST'])
def grocery():
    return render_template('grocery.html')

@app.route('/fruit',methods=['GET','POST'])
```

```
project.py - D:\206802\final project\project.py (3.10.4)
File  Edit  Format  Run  Options  Window  Help
def fruit():
    return render_template('fruits.html')
@app.route('/cart/<name>/<q>/<price>',methods=['GET','POST'])
def cart(name,q,price):
        session['cart'][name]=[q,price]
        session.modify=True
        return render_template('starting.html',id1=session['name'])
@app.route('/cartdisplay')
def view():
    print(session['cart'])
    data=session['cart']
    return render_template('cart.html',data=data)
@app.route('/login',methods=['GET','POST'])
def login():
    if session.get('name'):
        return redirect(url_for('home',id1=session.get['name']))
    if request.method=="POST":
        print(request.form)
        user=request.form['user']
        cursor=mysql.connection.cursor()
        cursor.execute('SELECT number from signup')
        users=cursor.fetchall()
        password=request.form['password']
        cursor.execute('select password from signup where number=%s',[user])
        data=cursor.fetchone()
        cursor.close()
        if (int(user),) in users:
            if password==data[0]:
                session['name']=user
                session['cart']={}
                return redirect(url_for('home',id1=user))
```



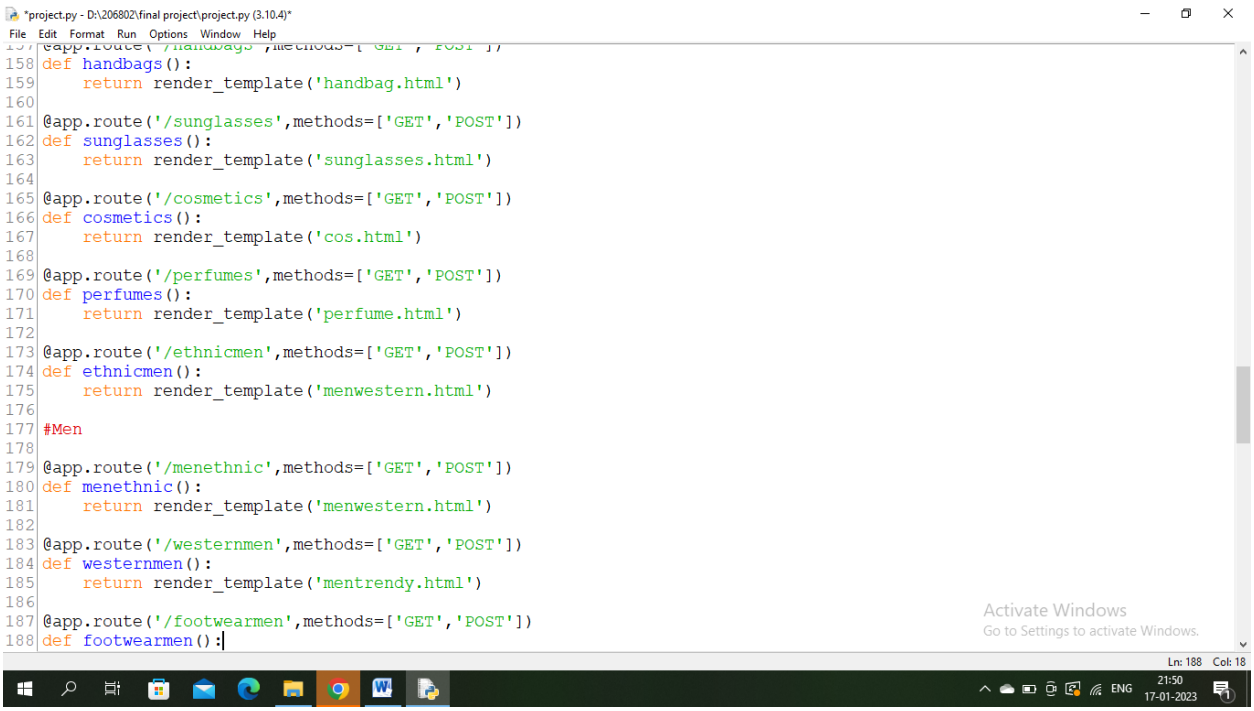
```
project.py - D:\206802\final project\project.py (3.10.4)
File  Edit  Format  Run  Options  Window  Help
            else:
                flash('Invalid Password')
                return render_template('loginpage.html')
        else:
            print('hi')
            flash('Invalid user id')
            return render_template('starting.html')
    return render_template('loginpage.html')
@app.route('/signup',methods=['GET','POST'])
def signup():
    if request.method=='POST':
            name=request.form['name']
            number=request.form['number']
            gender=request.form['gender']
            password=request.form['password']
            email=request.form['email']
            cursor=mysql.connection.cursor()
            cursor.execute('insert into signup values(%s,%s,%s,%s,%s)',[name,number,gender,password,email])
            mysql.connection.commit()
            flash('Details registered succesfully')
            return render_template('signuppage.html')
    return render_template('signuppage.html')

#Category Routes
@app.route('/vegetables',methods=['GET','POST'])
def vegetables():
    return render_template('vegetables.html')

@app.route('/spices',methods=['GET','POST'])
def spices():
    return render_template('spices.html')
```


Ln: 63 Col: 17

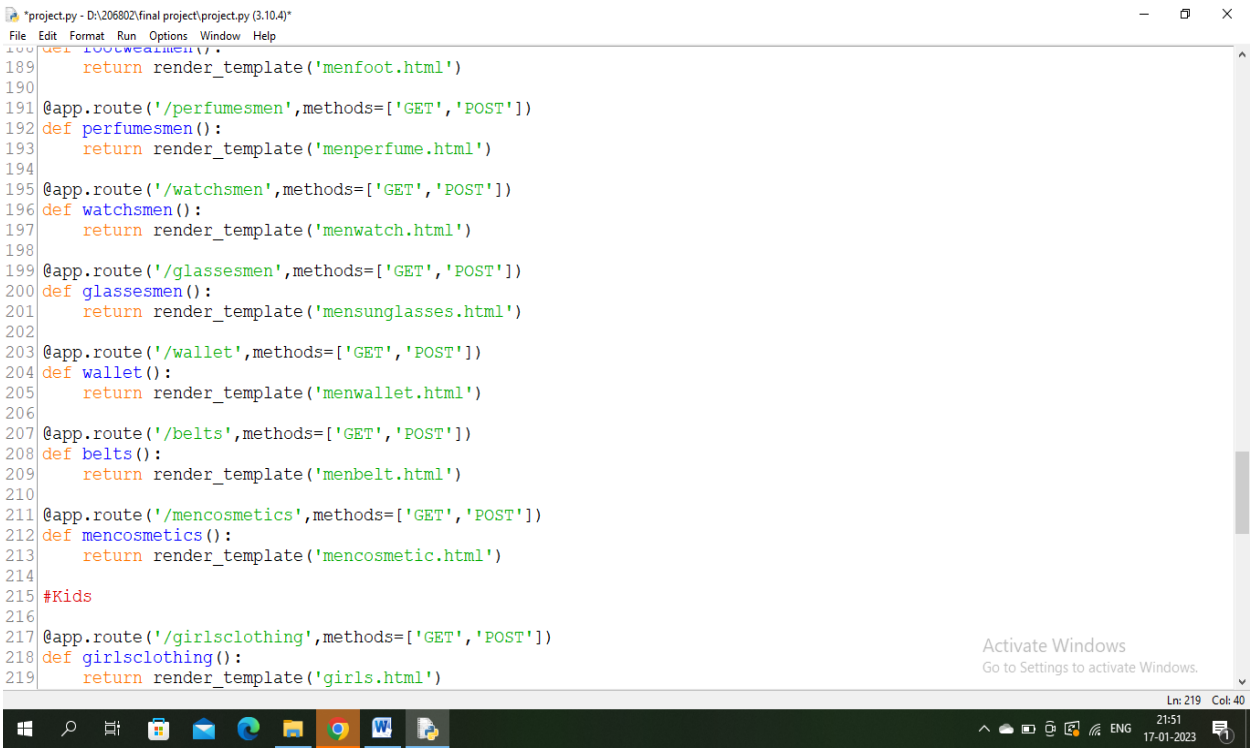
```
*project.py - D:\206802\final project\project.py (3.10.4)*
File  Edit  Format  Run  Options  Window  Help
@app.route('/oils',methods=['GET','POST'])
def oils():
    return render_template('oils.html')

@app.route('/snacks',methods=['GET','POST'])
def snacks():
    return render_template('snacks.html')

@app.route('/drinks',methods=['GET','POST'])
def drinks():
    return render_template('drink.html')

@app.route('/care',methods=['GET','POST'])
def care():
    return render_template('care.html')

@app.route('/milk',methods=['GET','POST'])
def milk():
    return render_template('milk.html')

@app.route('/dal',methods=['GET','POST'])
def dal():
    return render_template('dal.html')

@app.route('/women',methods=['GET','POST'])
def women():
    return render_template('women.html')

@app.route('/men',methods=['GET','POST'])
def men():
    return render_template('men.html')
```

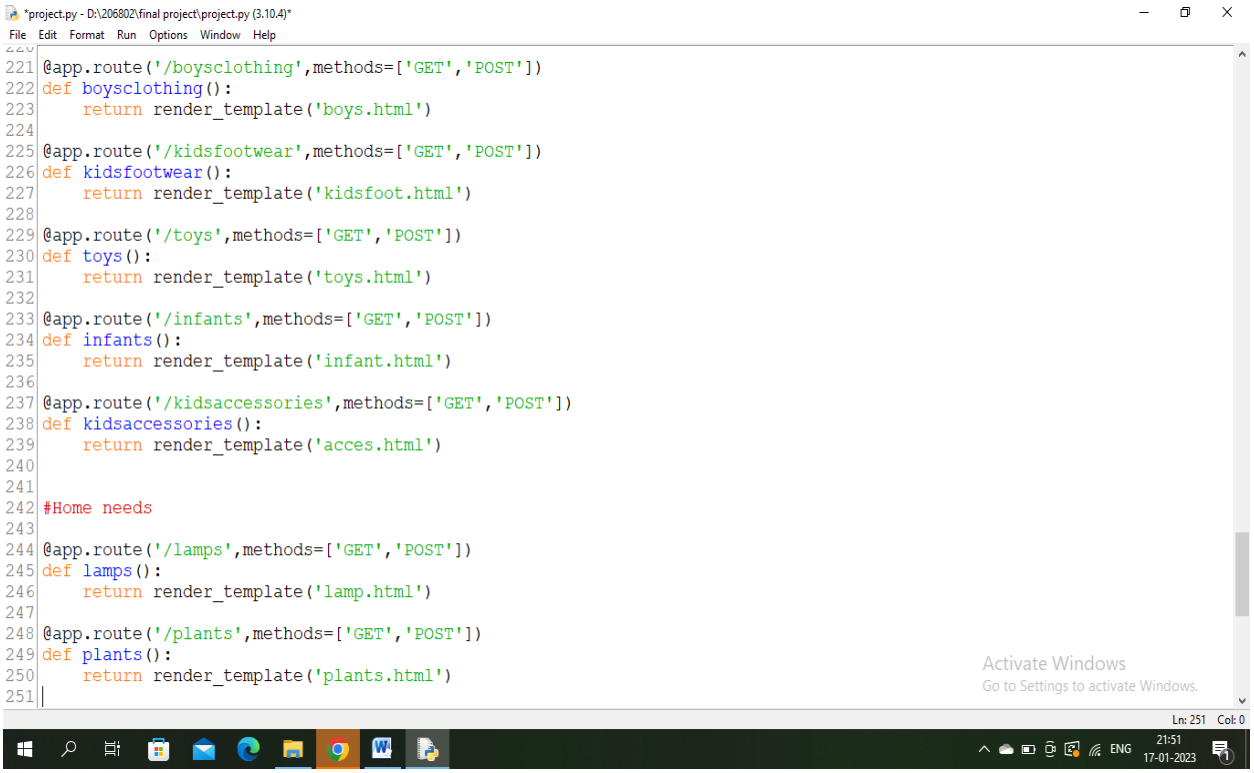

Ln: 125 Col: 38

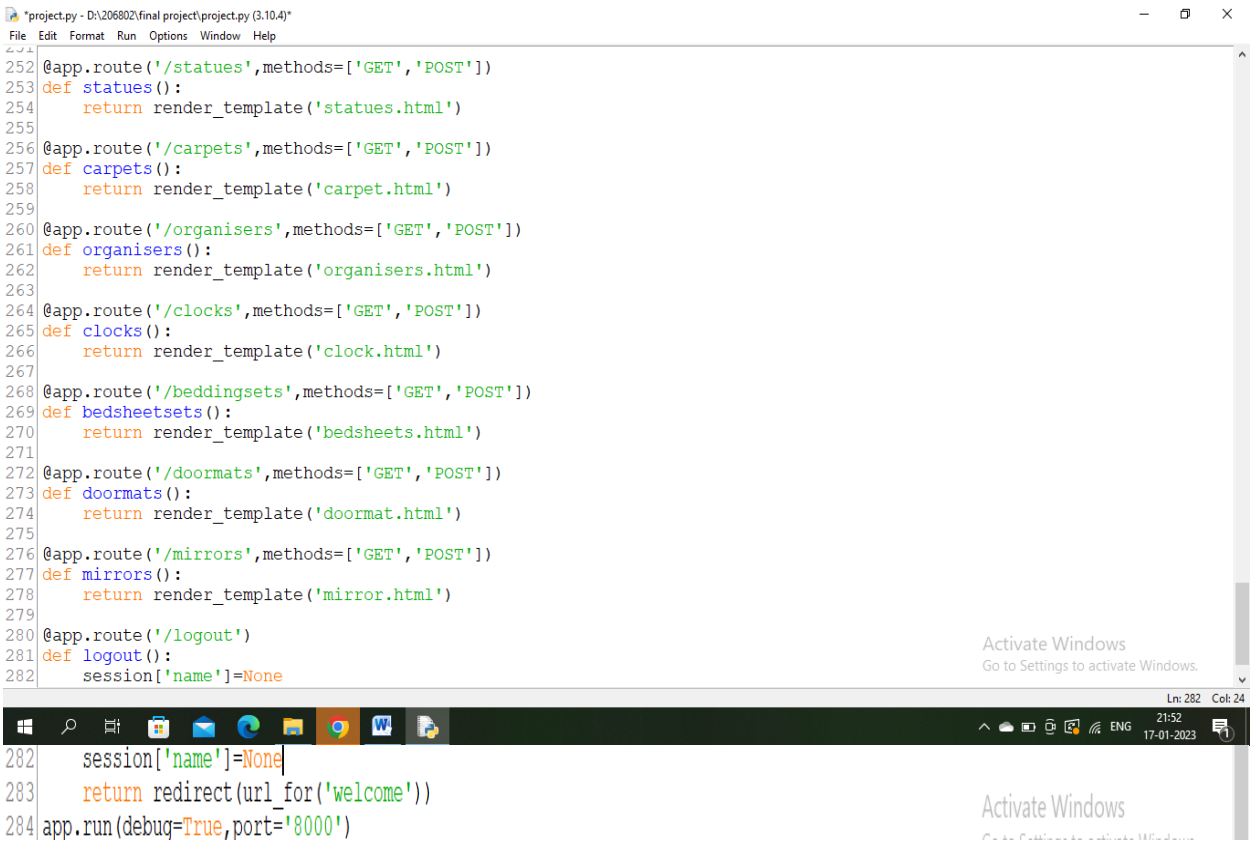
```
*project.py - D:\206802\final project\project.py (3.10.4)*
File Edit Format Run Options Window Help
@app.route('/kids',methods=['GET','POST'])
def kids():
    return render_template('kids.html')

@app.route('/house',methods=['GET','POST'])
def house():
    return render_template('home.html')

#Women

@app.route('/ethnicwomen',methods=['GET','POST'])
def ethnicwomen():
    return render_template('ethnic.html')

@app.route('/westernwomen',methods=['GET','POST'])
def westernwomen():
    return render_template('west.html')

@app.route('/watch',methods=['GET','POST'])
def watch():
    return render_template('watch.html')

@app.route('/footwearwomen',methods=['GET','POST'])
def footwearwomen():
    return render_template('foot.html')

@app.route('/jewellery',methods=['GET','POST'])
def jewellery():
    return render_template('jewellery.html')

@app.route('/handbags',methods=['GET','POST'])
```

```
*project.py - D:\206802\final project\project.py (3.10.4)*
File Edit Format Run Options Window Help
def handbags():
    return render_template('handbag.html')

@app.route('/sunglasses',methods=['GET','POST'])
def sunglasses():
    return render_template('sunglasses.html')

@app.route('/cosmetics',methods=['GET','POST'])
def cosmetics():
    return render_template('cos.html')

@app.route('/perfumes',methods=['GET','POST'])
def perfumes():
    return render_template('perfume.html')

@app.route('/ethnicmen',methods=['GET','POST'])
def ethnicmen():
    return render_template('menwestern.html')

#Men

@app.route('/menethnic',methods=['GET','POST'])
def menethnic():
    return render_template('menwestern.html')

@app.route('/westernmen',methods=['GET','POST'])
def westernmen():
    return render_template('mentrendy.html')

@app.route('/footwearmen',methods=['GET','POST'])
def footwearmen():
```


project.py - D:\206802\final project\project.py (3.10.4)
File Edit Format Run Options Window Help
def footwearmen():
return render_template('menfoot.html')
@app.route('/perfumesmen',methods=['GET','POST'])
def perfumesmen():
return render_template('menperfume.html')
@app.route('/watchsmen',methods=['GET','POST'])
def watchsmen():
return render_template('menwatch.html')
@app.route('/glassesmen',methods=['GET','POST'])
def glassesmen():
return render_template('mensunglasses.html')
@app.route('/wallet',methods=['GET','POST'])
def wallet():
return render_template('menwallet.html')
@app.route('/belts',methods=['GET','POST'])
def belts():
return render_template('menbelt.html')
@app.route('/mencosmetics',methods=['GET','POST'])
def mencosmetics():
return render_template('mencosmetic.html')
#Kids
@app.route('/girlsclothing',methods=['GET','POST'])
def girlsclothing():
return render_template('girls.html')
Activate Windows
Go to Settings to activate Windows.
Ln: 219 Col: 40
ENG 21:51 17-01-2023


project.py - D:\206802\final project\project.py (3.10.4)
File Edit Format Run Options Window Help
@app.route('/boysclothing',methods=['GET','POST'])
def boysclothing():
return render_template('boys.html')
@app.route('/kidsfootwear',methods=['GET','POST'])
def kidsfootwear():
return render_template('kidsfoot.html')
@app.route('/toys',methods=['GET','POST'])
def toys():
return render_template('toys.html')
@app.route('/infants',methods=['GET','POST'])
def infants():
return render_template('infant.html')
@app.route('/kidsaccessories',methods=['GET','POST'])
def kidsaccessories():
return render_template('acces.html')
#Home needs
@app.route('/lamps',methods=['GET','POST'])
def lamps():
return render_template('lamp.html')
@app.route('/plants',methods=['GET','POST'])
def plants():
return render_template('plants.html')
Activate Windows
Go to Settings to activate Windows.
Ln: 251 Col: 0
ENG 21:51 17-01-2023



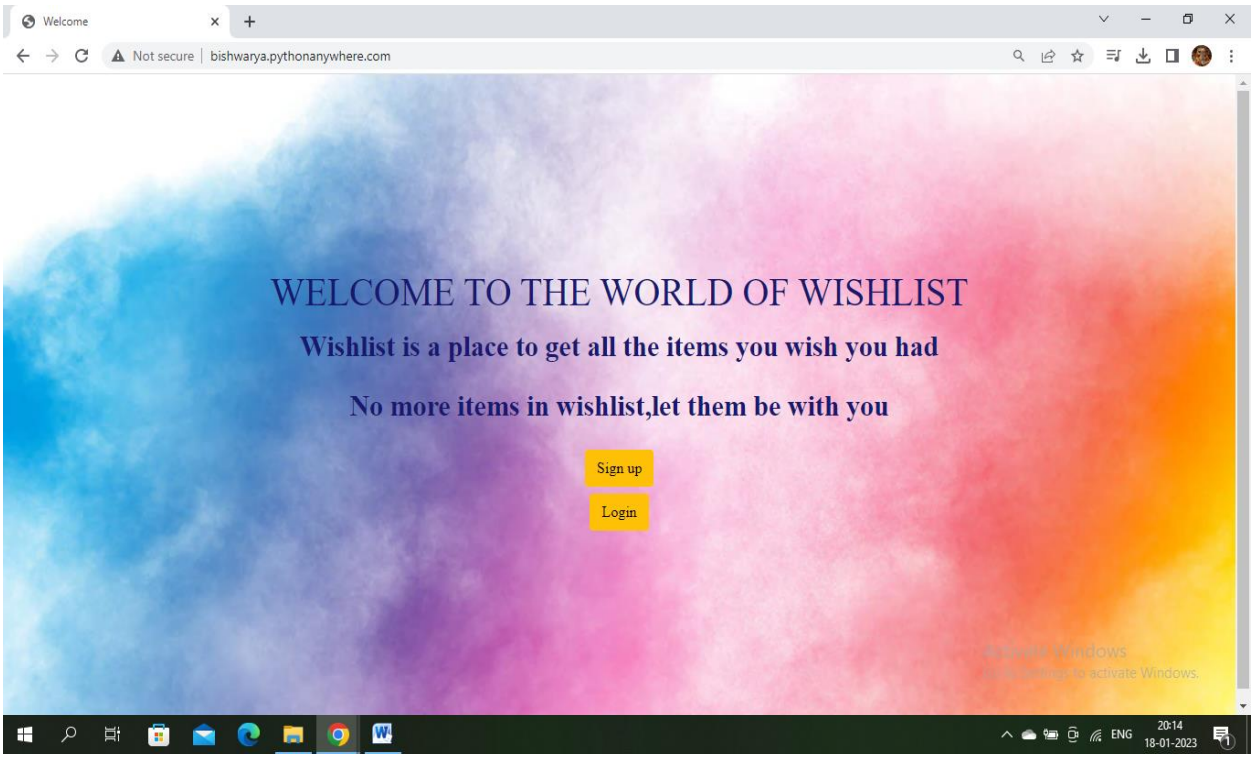
```
*project.py - D:\206802\final project\project.py (3.10.4)*
File Edit Format Run Options Window Help
@app.route('/statues',methods=['GET','POST'])
def statues():
    return render_template('statues.html')

@app.route('/carpets',methods=['GET','POST'])
def carpets():
    return render_template('carpet.html')

@app.route('/organisers',methods=['GET','POST'])
def organisers():
    return render_template('organisers.html')

@app.route('/clocks',methods=['GET','POST'])
def clocks():
    return render_template('clock.html')

@app.route('/beddingsets',methods=['GET','POST'])
def bedsheetsets():
    return render_template('bedsheets.html')

@app.route('/doormats',methods=['GET','POST'])
def doormats():
    return render_template('doormat.html')

@app.route('/mirrors',methods=['GET','POST'])
def mirrors():
    return render_template('mirror.html')

@app.route('/logout')
def logout():
    session['name']=None
Activate Windows
Go to Settings to activate Windows.
Ln: 282 Col: 24
ENG 21:52 17-01-2023
    session['name']=None
    return redirect(url_for('welcome'))
app.run(debug=True,port='8000')
Activate Windows
```

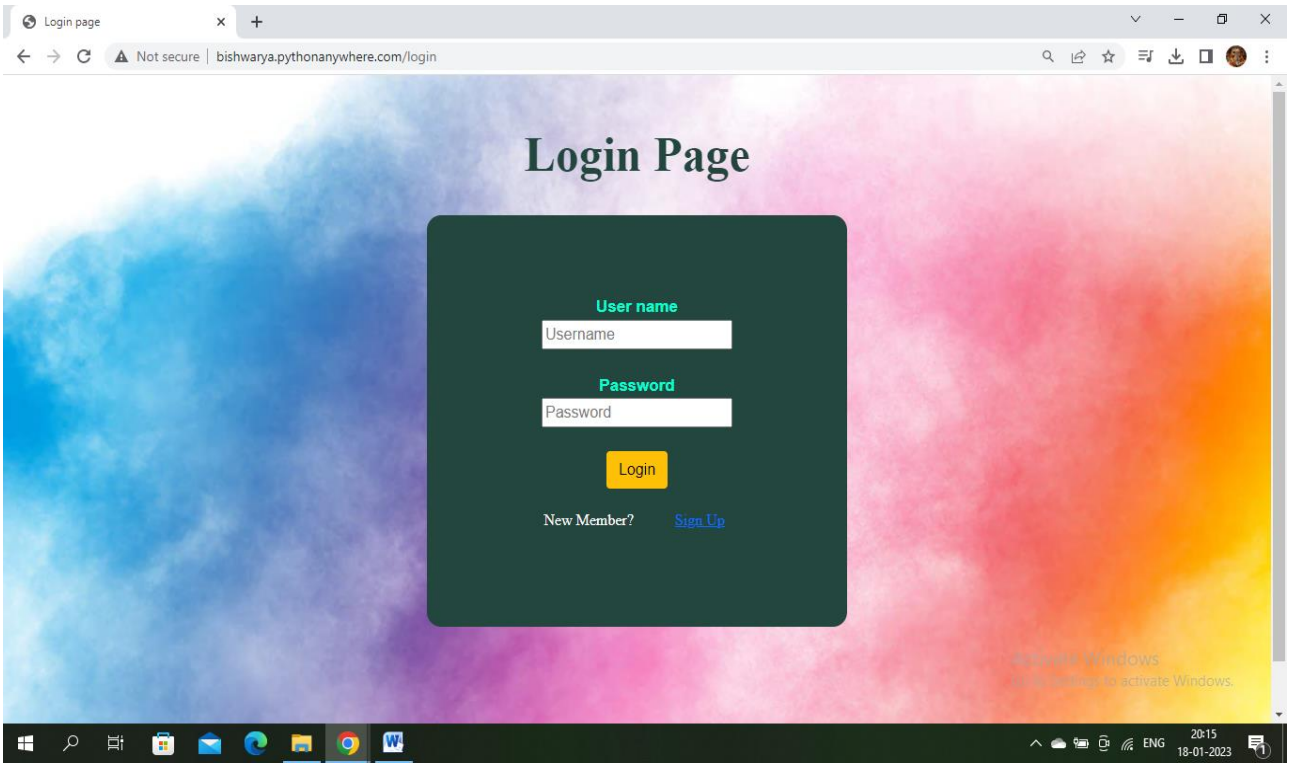
• Create a tables in MYSQL

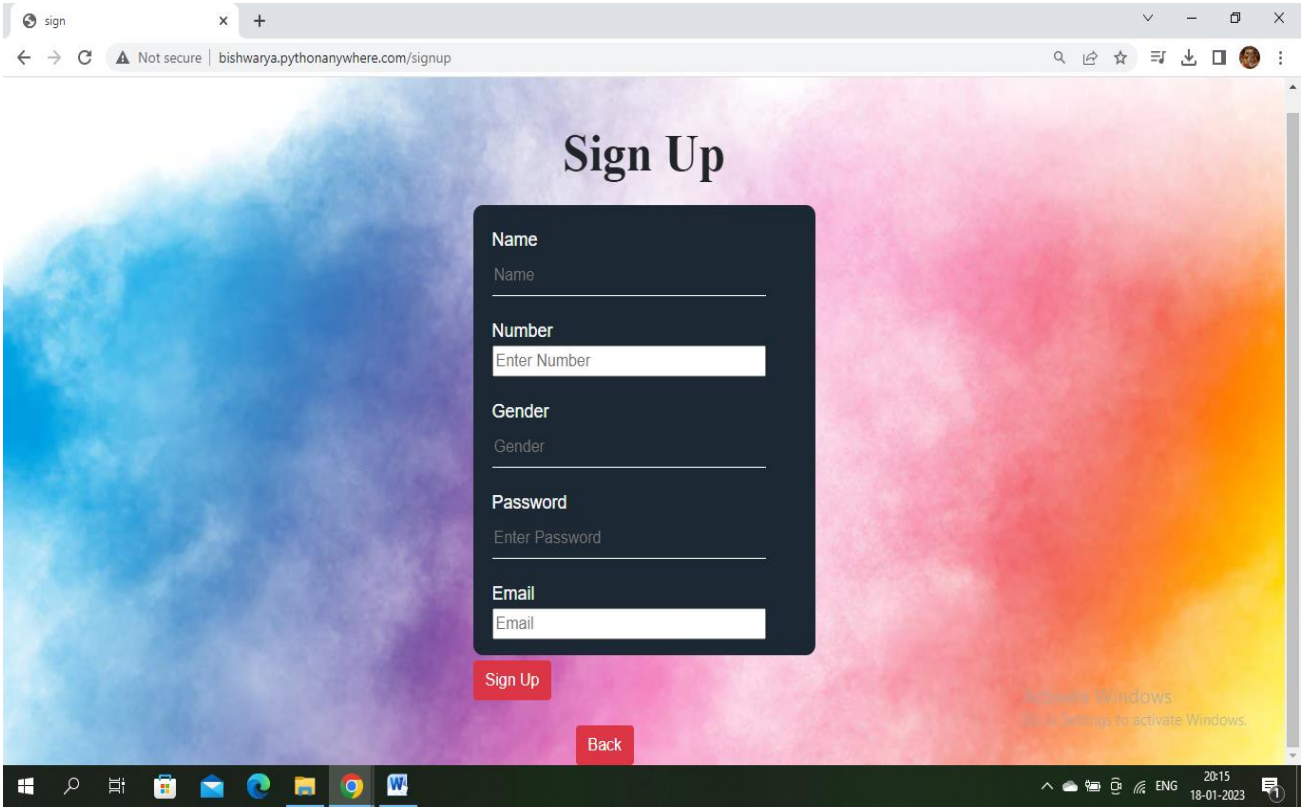
```
mysql> desc signup;
+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| name     | varchar(100) | YES  |     | NULL    |       |
| number   | bigint       | NO   | PRI | NULL    |       |
| gender   | varchar(10)  | YES  |     | NULL    |       |
| password | varchar(50)  | YES  |     | NULL    |       |
| email    | varchar(50)  | YES  |     | NULL    |       |
+----------+--------------+------+-----+---------+-------+
5 rows in set (0.44 sec)
```

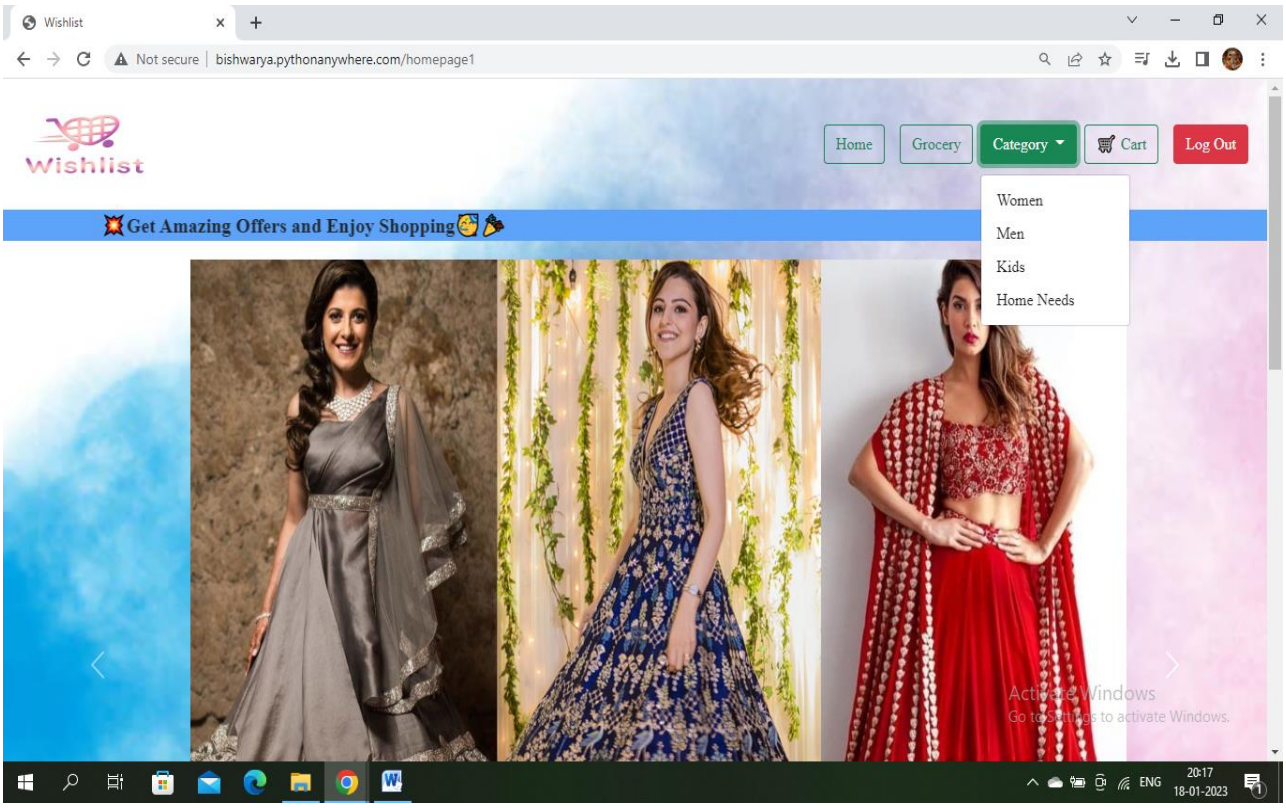
```
mysql> desc cart;
+-------+--------------+------+-----+---------+-------+
| Field | Type         | Null | Key | Default | Extra |
+-------+--------------+------+-----+---------+-------+
| item  | varchar(100) | YES  |     | NULL    |       |
| qty   | int          | YES  |     | NULL    |       |
| price | int          | YES  |     | NULL    |       |
+-------+--------------+------+-----+---------+-------+
3 rows in set (0.11 sec)
```

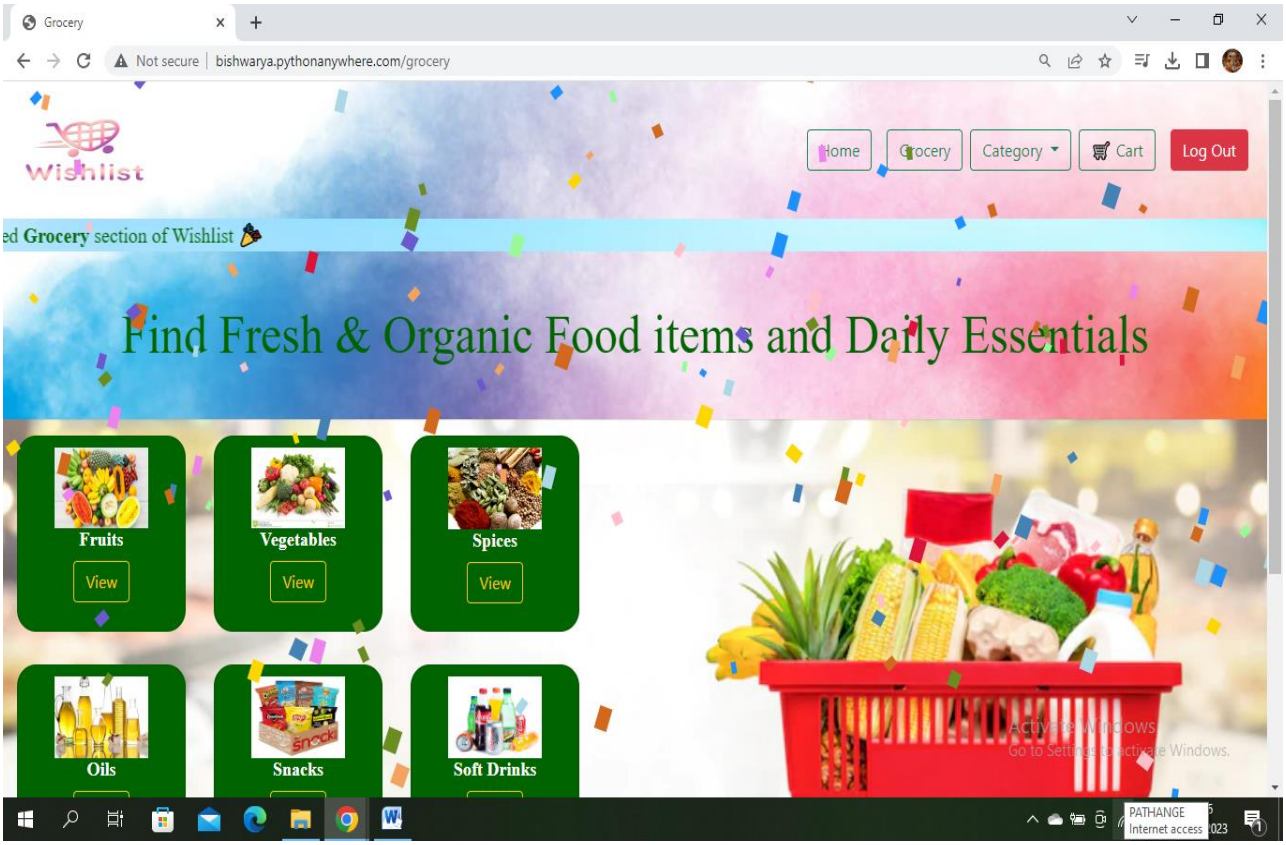
- Output:

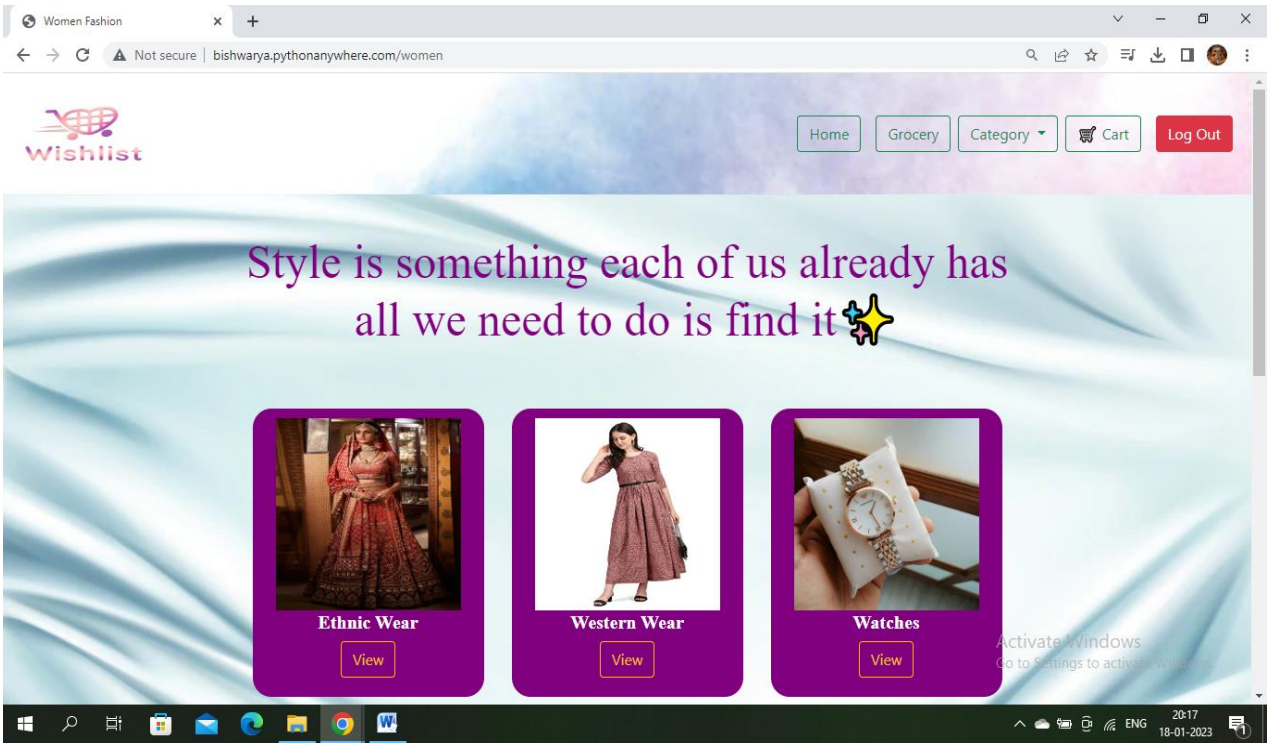


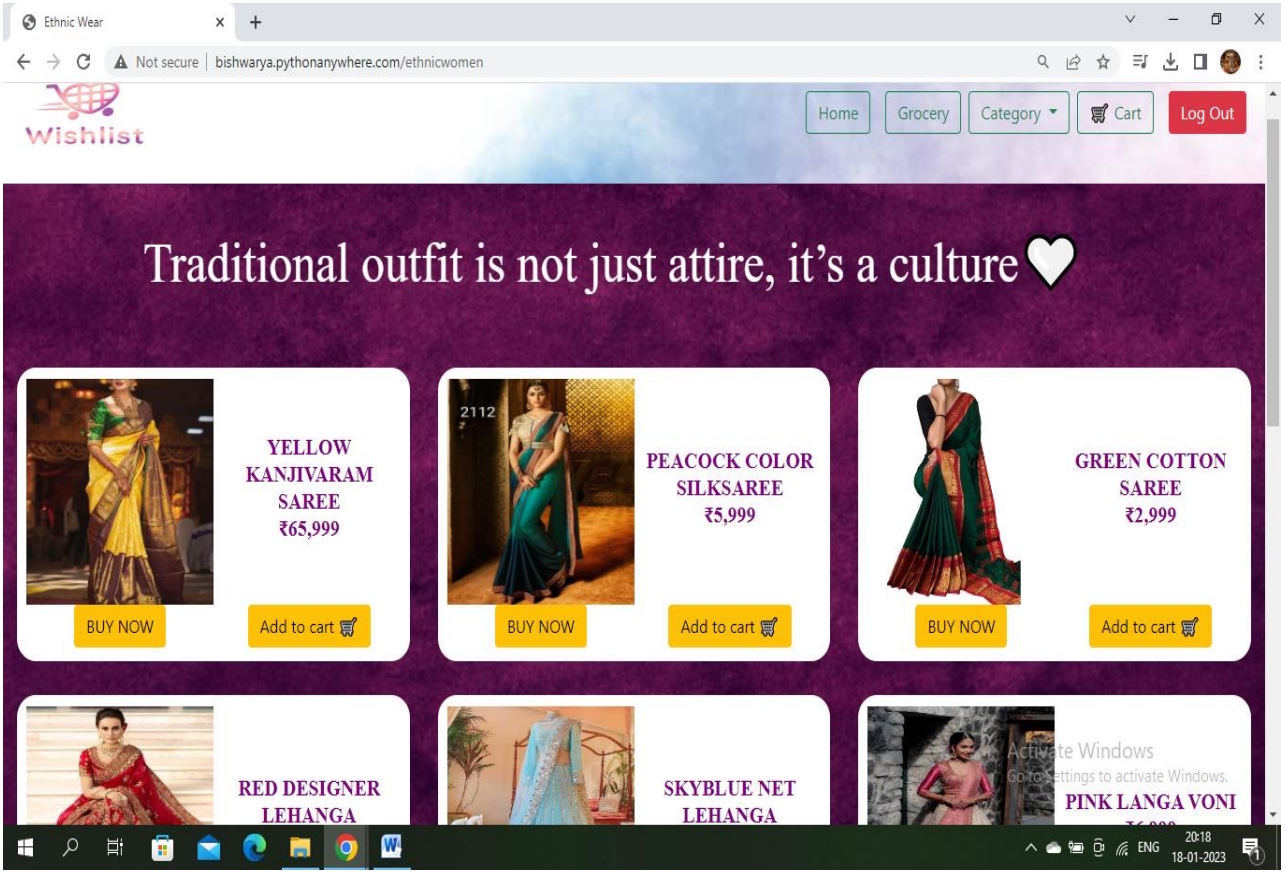

Login page
Not secure | bishwarya.pythonanywhere.com/login
Login Page
User name
Username
Password
Password
Login
New Member?
Sign Up
ENG
20:15
18-01-2023

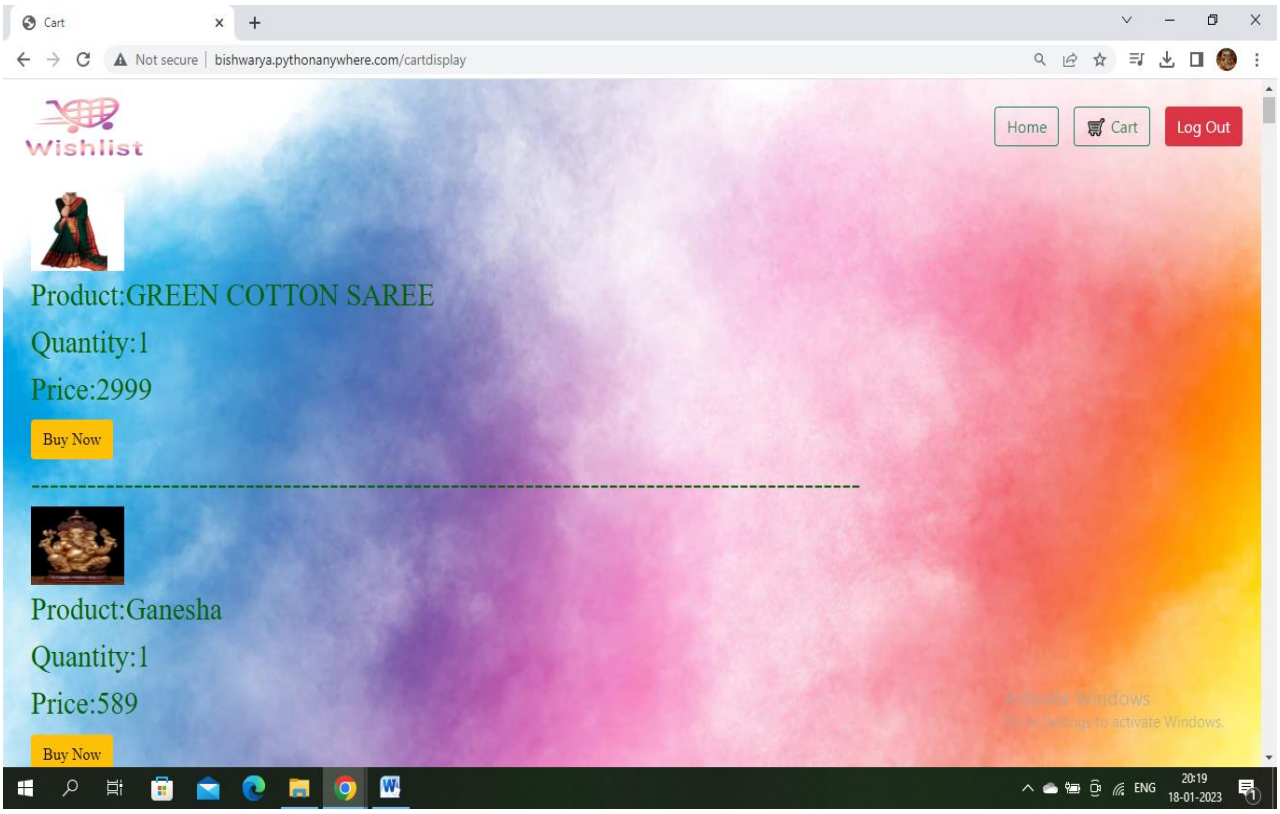

sign
Not secure | bishwarya.pythonanywhere.com/signup
Sign Up
Name
Name
Number
Enter Number
Gender
Gender
Password
Enter Password
Email
Email
Sign Up
Back
ENG
20:15
18-01-2023


Wishlist
Not secure | bishwarya.pythonanywhere.com/homepage1
Wishlist
Home
Grocery
Category
Cart
Log Out
Women
Men
Kids
Home Needs
Get Amazing Offers and Enjoy Shopping


Grocery
Not secure | bishwarya.pythonanywhere.com/grocery
Wishlist
Home
Grocery
Category
Cart
Log Out
ed Grocery section of Wishlist
Find Fresh & Organic Food items and Daily Essentials
Fruits
View
Vegetables
View
Spices
View
Oils
Snacks
Soft Drinks


Women Fashion
Not secure | bishwarya.pythonanywhere.com/women
Wishlist
Home
Grocery
Category
Cart
Log Out
Style is something each of us already has
all we need to do is find it
Ethnic Wear
View
Western Wear
View
Watches
View
Activate Windows
Go to Settings to activate Windows.
ENG
20:17
18-01-2023


Ethnic Wear
Not secure | bishwarya.pythonanywhere.com/ethnicwomen
Wishlist
Home
Grocery
Category
Cart
Log Out
Traditional outfit is not just attire, it's a culture
YELLOW KANJIVARAM SAREE
₹65,999
BUY NOW
Add to cart
PEACOCK COLOR SILKSAREE
₹5,999
BUY NOW
Add to cart
GREEN COTTON SAREE
₹2,999
BUY NOW
Add to cart
RED DESIGNER LEHANGA
SKYBLUE NET LEHANGA
PINK LANGA VONI
Activate Windows
Go to Settings to activate Windows.
ENG
20:18
18-01-2023



• It goes to Exit and we exit and the file is closed

Conclusion

Conclusion:

In conclusion we can say that there is no end to product improvement.Even when there seems to room for improvement there always is.just like google was once just a search engine,but provided additional related features.similarly our goal is other features to our ecommerce system.

The objective from this course was to learn to buld a product using formal methods such as information gathering,benchmarking,various diagrams DFD,use case,activity etc.We applied these tools we learned into our own project.These formal methods has given a structure to our project.These formal and necessary documentation for future upgrade.
We want to continue build our project as we learn more formal methods.

These below are Some of the future directions that we want to see in our project.Al based chatting system with customer for statistically similar questions.Personalized product suggestion based on customer purchase and browsing record.Using sslcommerz api to provide online transations.

Creating an algorithms to provide delivary routine to minimize cost.if multiple products are being delivered to same area in close time slot then cost can be minimized.A product request system for customers to add new products to inventory.Doing sentimental analysis on product review to get true customer satisfaction of product .An admin panel with complete customization and controlover website.

We have successfully implemented the site ‘Bookz Kart’.with the help of various links and tools,we have been able to provide a site which will be live soon and running on theweb.We have been successful in our attempt to take care of the needs of both the user as well as the administrator.Finally we hope that this will go a long way in popularizing.

Modules:

Modern e-commerce stores require a feature-rich set of components to provide customers with the best possible shopping experience. The traditional e-commerce architecture tightly couples all these services into a single system. Combining all parts of the e-commerce system into a single platform like this limits what you are able to get out of each service. It is also difficult to scale.

In this post, we’ll examine the three e-commerce modules most integral to an online store: product information management, content management, cart and checkout.Together, these modules allow you to store your e-commerce data, present it to customers, and capture their orders.

Modular ecommerce divides the backend components into individual modules. Each is responsible for a single service. Breaking each area of functionality out into its own module provides more flexibility. It cleans up the codebase and makes it easier to deploy new features.

Some of the most common e-commerce modules include:

- Cart and checkout
- Product information management
- Order management
- Pricing and promotions engines
- Content management

Cart and checkout:

The cart page of an e-Commerce website shows all items that a customer has added to the cart. The cart page is built by using the cart module. The cart module is a container that hosts all the modules that are required to showcase items in the cart. The checkout page is where customers enter the information that is required to place an order.

Product information management:

Product information management (**PIM**) is the process of managing all the information required to market and sell products through distribution channels. This product data is created by an internal organization to support a multichannel marketing strategy.

Order management:

Order management is the process of order capturing, tracking, and fulfilling customer orders. An order management system (OMS) is a computer software system used in several industries for order entry and processing. A perfect order means fulfilling a sales order to the customer's specifications, delivering goods as promised at the time of sale.

Pricing and promotions engines:

Price may seem like a static element that is consistent across users and channels, however, adjusting prices based on buying intent and individual characteristics can lead to increased conversions. Promotions are special offers that allow customers to buy a product below its normal selling price.

Content management:

Content management (CM) is the process for collection, delivery, retrieval, governance and overall management of information in any format. The term is typically used in reference to administration of the digital content lifecycle, from creation to permanent storage or deletion.

Purpose of Ecommerce:

With the evolution of technology and the wave of digitalization, more and more businesses are adapting to tech evolutions. You will find the digital payment options

with the grocery sellers & street-side vendors as well. If you are also non-techies who want to take their business online, Builderfly can be your savior. Builderfly is an ecommerce platform exclusively designed for individuals & businesses to start selling online, market & grow their business without any technicalities. So take your business to the World at your own pace.

- Run discount sales for past customers to get them to shop again
- Offer good and timely customer support in your store: the live chat feature available on Instamojo helps you communicate with customers and provide them with the right level of support.
- Send out emails to announce a new product launch or when you run a sale

Functions of E-commerce

- **To save time and money:**

E Commerce is a great way to save time and money for all of the customers surfing on Internet. It's convenient and you can find great deals on the items you need. You can also find a wide variety of products and brands online which makes it easy to find the perfect product for you.

- **To get the best deals:**

Ecommerce is the process of buying and selling products or services online. The goal of ecommerce is to provide customers with a variety of choices, competitive prices, and quick delivery times. Ecommerce platforms also allow businesses to track customer behaviour and improve customer service.

- **To have a personal shopper:**

The customers like ecommerce because it provides a great shopping and shipping experience of physical and digital products. Online Customers can browse through products, add items to their cart, and checkout all in one place.

- **To find new and unique products:**

Additionally, ecommerce platforms often allow customers to compare prices and read reviews before making a purchase. This helps shoppers make informed decisions about what they buy and saves them time and money.

Looking at some technical benefits:

• No need of any additional hardware and software installations
• Runs on the cloud based
• User friendly interface
• Smart, customizable and robust design

Features of an Effective E-commerce:

Business in the e-commerce sector is growing at an accelerated pace. If you're not aware, some e-commerce websites barely make a profit while others are wildly successful. This article outline nine important e-commerce website features that can improve sales.

- **An Easy-To-Use Navigation System:**

A survey shows that 76% of consumers list website ease of use as the most significant characteristic. Developing user-friendly e-commerce web design improves the search functionality of products.

- **Mobile-Friendly Design:**

The website will respond to the size of the screen displayed. With your mobile-friendly design, your laptops and smartphones will all display correctly.

- **Customer Reviews & Ratings:**

eCommerce websites offer numerous choices, so customers remain cautious in purchase decisions. Customers check the review and rating of the product purchased by other customers. More than 92% of users verify the review before placing an order.

- **Secured Payment Options:**

A trustworthy e-commerce store should accept secure payment methods such as PayPal, Stripe, Apple Pay, and Braintree. E-commerce websites with various Payment gateway enable customers across the globe to purchase the products.

- **Buyer Friendly Shopping Cart:**

Shopping carts must allow the user to modify the quantities, apply promo codes, add or remove products from the card, and shipping options. Design the shopping cart with simple visuals displaying brand colors and check-out buttons.

SQL DATABASE:

SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987.SQL is also known as Standard Query Language which is used as a medium for communicating with the database. SQL lets you access and manipulate databases SQL statements are utilized to execute the queries against the database and retrieve data from the database. We can create a new database, table, stored

procedure and update, delete and add items to the table. We also can view the data and set permission to the view, procedure, and table. SQL is Structured Query Language, which is a computer language for storing,manipulating and retrieving data stored in a relational database.

Analysis:

A good marketing analytics software keeps all your data in one place. You can keep tabs on all your campaigns, from social ads to emails to marketing automations. You can also see real-time stats, so you can know what's working quickly and make better decisions about where to put your marketing dollars. Data points can tell you a story about the total number of visitors to your site in a given week. For example, maybe only 50% enjoyed your website enough to even spend more than a few seconds on it. And maybe only half of those who stayed actually made a purchase, while another 10% got stuck at the checkout page, became frustrated, and left.

Technologies:

HTML(Hyper Text Markup Language):

HTML stands for **Hyper Text Markup Language**, which is the most widely used language on Web to develop web pages. **HTML** was created by Berners-Lee in late 1991 but "HTML 2.0" was the first standard HTML specification which was published in 1995. HTML 4.01 was a major version of HTML and it was published in late 1999. Though HTML 4.01 version is widely used but currently we are having HTML-5 version which is an extension to HTML 4.01, and this version was published in 2012.

Originally, **HTML** was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers. Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

HTML is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain.

Python:

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). This **tutorial** gives enough understanding on **Python programming** language.**Python** is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than
other languages.

Python is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain.

SRS:

A **software requirements specification** (**SRS**) is a description of a software system to be developed. It is modeled after business requirement specification. The software requirements specification lays out functional and non-functional requirements, and it may include a set of use cases that describe user interactions that the software must provide to the user for perfect interaction.

Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers on how the software product should function (in a market driven project, these roles may be played by the marketing and development divisions). Software requirements specification is a rigorous assessment of requirements before the more specific system design stages, and its goal is to reduce later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules.

Project Scope:

E-commerce has bloomed over the years and is one of the fastest-growing domains in the online world. Though it took some time for this to be accepted by the end-users, today we are at a point where the majority of the people love to shop online. There were numerous concerns revolving around online shopping at its launch, but over years people tend to have started trusting E-commerce for all their shopping needs. In India, people prefer shopping online these days rather than having to visit the physical store. The payment features that are smart and secure as well as the cash on delivery (COD), which makes the payment, even more, safer with hassle-free shipping, easy returns and reach out.

- Different Ecommerce Marketplaces
- Offers and Discounts
- Fast Shipping
- Customer Service
- Reach to More Audience
- Brand Exposure
- Advertising

Product Functions:

This section provides the graspable functional overview of the end product. The product is expected to be providing following functionalities

- Login
- Logout
- Add to cart
- Search for categories

FUNCTIONAL REQUIREMENTS:

- The mobile-first approach.
- The power of personalisation.
- Provide seamless customer journeys

Uml:

A UML diagram is a diagram based on the UML (Unified Modeling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain, or document information about the system.UML is an acronym that stands for Unified Modeling Language. Simply put, UML is a modern approach to modeling and documenting software. In fact, it's one of the most popular business process modeling techniques.It is based on diagrammatic representations of software components. As the old proverb says: "a picture is worth a thousand words". By using visual representations, we are able to better understand possible flaws or errors in software or business processes.

Use Case Diagram:

Use case diagrams are a set of use cases, actors, and their relationships. They represent the use case view of a system. A use case represents a particular functionality of a system. Hence, use case diagram is used to describe the relationships among the functionalities and their internal/external controllers. These controllers are known as actors.

Sequence Diagram:

A sequence diagram is an interaction diagram. From the name, it is clear that thediagram deals with some sequences, which are the sequence of messages flowing from one object to another. Interaction among the components of a system is very important from implementation and execution perspective. Sequence diagram is used to visualize the sequence of calls in a system to perform a specific functionality.

Collaboration Diagram:

Collaboration diagram is another form of interaction diagram. It represents the structural organization of a system and the messages sent/received. Structural organization consists of objects and links.

Activity Diagram:

Activity diagram describes the flow of control in a system. It consists of activities and links. The flow can be sequential, concurrent, or branched.Activities are nothing but the functions of a system. Numbers of activity diagrams are prepared to capture the entire flow in a system.Activity diagrams are used to visualize the flow of controls in a system. This is prepared to have an idea of how the system will work when executed.

Class diagrams:

The class diagram depicts a static view of an application. It represents the types of objects residing in the system and the relationships between them. A class consists of its objects, and also it may inherit from other classes. A class diagram is used to visualize, describe, document various different aspects of the system, and also construct executable software code. It shows the attributes, classes, functions, and relationships to give an overview of the software system. It constitutes class names, attributes, and functions in a separate compartment that helps in software development. Since it is a collection of classes, interfaces, associations, collaborations, and constraints, it is termed as a structural diagram.

Advantages Of E-Commerce:

- E-commerce involves selling products online, building a brand, and online advertising.
- Businesses find many advantages to e-commerce such as cost reduction and customer data.
- Consumers enjoy the advantages of 24/7 shopping from their homes.

THANK YOU