**PROJECT TITLE**

‘’E-COMMERCE PROTAL’’

An E-Commerce portal which will allow formal and informal merchants in developing countries to advertise and sell their goods on the internet. This would permit rural communities to make their wares available to the rest of the world via the World Wide Web. The objective of this project is to create an e-commerce web portal with a content management system which would allow product information to be updated securely using a mobile device. The web portal will have an online interface in the form of an e-commerce website that will allow users to buy goods from the merchants

E-commerce businesses may also employ some or all of the followings:

Online shopping for retail sales direct to consumers via Web sites and mobile apps, and conversational commerce via live chat, chatbots, and voice assistants.

Providing or participating in online marketplaces, which process third-party business-to-consumer or consumer-to-consumer sales

Business-to-business buying and selling;

Gathering and using demographic data through web contacts and social media Business-to-business (B2B) electronic data interchange

Marketing to prospective and established customers by e-mail or fax (for example, with newsletters) Engaging in [pretail](https://en.wikipedia.org/wiki/Pretail) for launching new products and services

Online financial exchanges for currency exchanges or trading purposes.

This project will be divided into following separate components: The e-commerce website/portal

The product, merchant and customer database Reporting of the sales, orders, shipments etc The online transaction security system

The data security system

The e-commerce portal will have the following key features:

An online shop that will allow online shoppers to buy wares from formal and informal merchants. A search engine on the website to allow customers to find specific types of merchandise.

A secure online transaction system that will allow shoppers to purchase goods safely using their credit cards.

A database of merchandise with photos, product descriptions and stock information. This database will also contain all relevant merchant and customer information.

A data security system that will ensure that all data that is transmitted between the various system

#### OBJECTIVE

The objective of the project is to create an e-commerce we portal with content management

System which would allow product information to be updated securely using mobile device . the web Portal will have an online interface in the form of an e-commerce website that allow user to buy Goods from the merchants.

This project will be divided into following separate components : The e-commerce website/portal

The product , merchant and customer database The reporting of the sales ,orders , shipment etc The online transaction securities system.

The data security system

#### Project Category:

Relational Database Management System (RDBMS)

This is an RDBMS based project which is currently using MySQL for all the transaction statements.

MySQL is an opensource RDBMS System. A RDBMS is a database that has extremely powerful relational capabilities. By terms

“ relation ” we mean a database which can deal with a set of multiple database tables at the same instance of time and can perform all the database operation very efficiently such as appending, modifying , deleting sorting searching , query handling, record blocking, replication facilities. Which is an extremely powerful RDBMS as the back end, is that it offers powerful backup and recovery mechanisms.

In this software the information will be handled by using SQL server for sorting records in a database

,as a back-end

Brief Introduction about RDBSM : A relational database management system (RDBMS) is a database management system (DBMS) that is based on the relational model as invented by E. F. Codd, of IBM's San Jose Research Laboratory. Many popular databases currently in use are based on the relational database model. RDBMSs have become a predominant choice for the storage of information in new databases used for financial records, manufacturing and logistical information, personnel data, and much more since the 1980s. Relational databases have often replaced legacy hierarchical databases and network databases because they are easier to understand and use. However, relational databases have been challenged by object databases, which were introduced in an attempt to address the object-relational impedance mismatch in relational database, and XML databases.

## Tools/Platform

Hardware and Software Requirement specifications: For setting this ecommerce portal, it requires certain technical requirements to be met for the store to operate properly. First, a web server must be created to make the ecommerce store publicly available on the web. Domain names and hosting services can easily be purchased for an affordable price. When selecting a hosting service, you should check to see that these server requirements are provided and installed on their web servers:

Software Details for running E-Commerce Portal

#### Software Requirement

|  |  |
| --- | --- |
| FRONT END | Html , CSS, JAVASCRIPT |
| OPERATING SYSTEM | WINDOWS 7 |
| WEB SERVER | HTTP,URLs |
| BACK END | Database |
| LANGUAGE | Python |
| FRAMEWORK | DJANGO |
|  |  |

**HARDWARE REQUIREMENT**

|  |  |
| --- | --- |
| RAM | 2 GB |
| HARDDISK | 500GB |
| PROCESSOR | DUAL CORE AT LEAST |

**System Analysis**

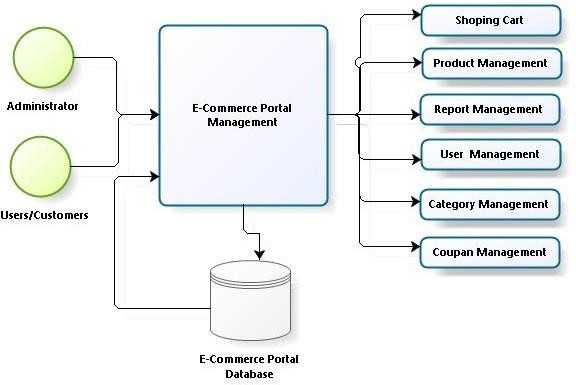
The development of a computer-based information system includes a system analysis phase. This helps produce the data model, a precursor to creating or enhancing a database. There are a number of different approaches to system analysis. When a computer-based information system is developed, system analysis (according to the Waterfall model) would constitute the following steps:

* The development of a feasibility study: determining whether a project is economically, socially, technologically and organizationally feasible
* Fact-finding measures, designed to ascertain the requirements of the system's end- users (typically involving interviews, questionnaires, or visual observations of work on the existing system)
* Gauging how the end-users would operate the system (in terms of general experience in using computer hardware or software), what the system would be used for and so on

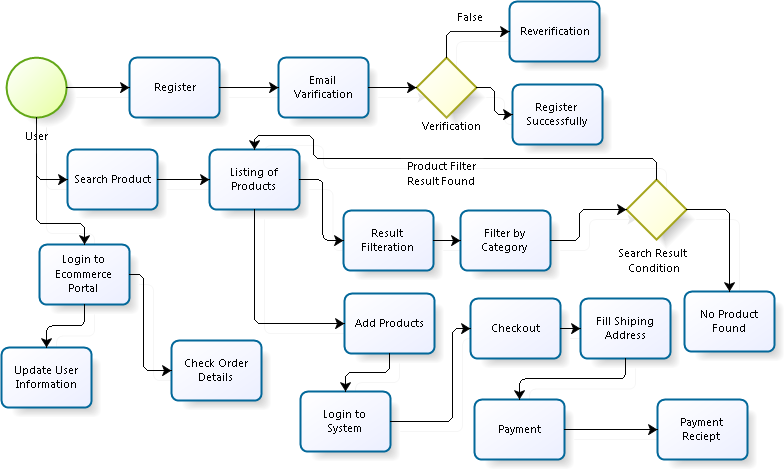
Another view outlines a phased approach to the process. This approach breaks system analysis into 5 phases:

* Scope Definition: Clearly defined objectives and requirements necessary to meet a project's requirements as defined by its stakeholders
* Problem analysis: the process of understanding problems and needs and arriving at solutions that meet them
* Requirements analysis: determining the conditions that need to be met
* Logical design: looking at the logical relationship among the objects
* Decision analysis: making a final decision

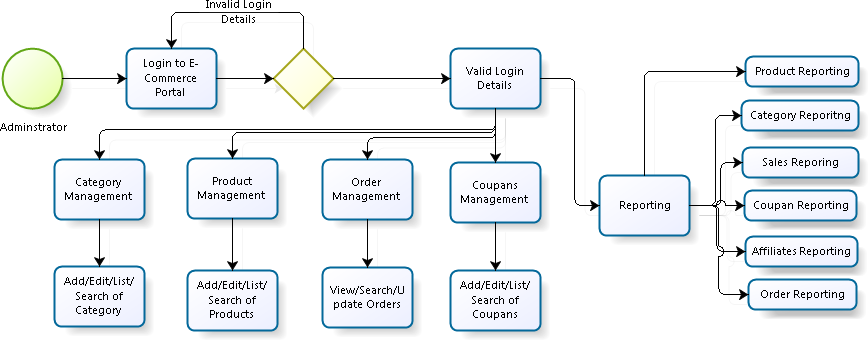
#### DATAFLOWDIAGRAM LEVEL= 0



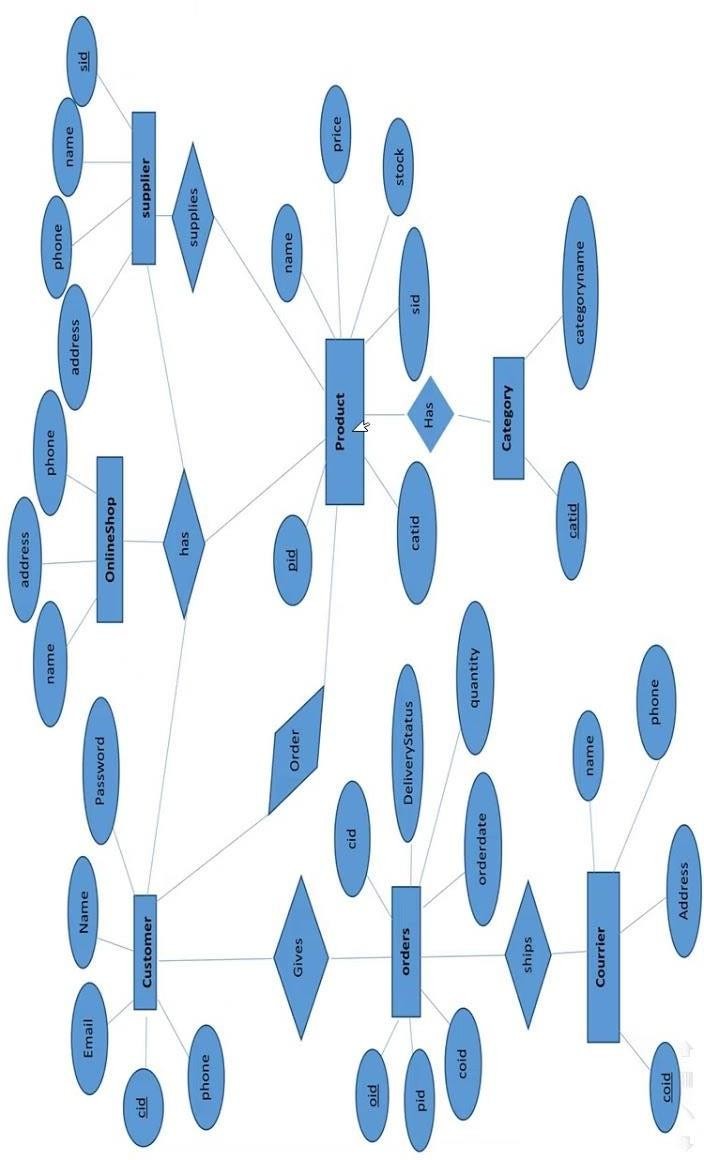
**Dataflow Diagram Level 1**



**Dataflow Diagram Level 2**



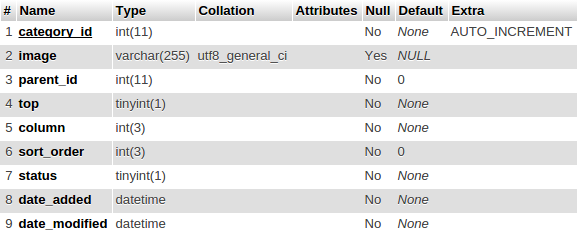
**ERD\_DIAGRAM**

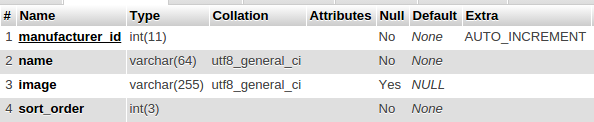


**Complete Structure**

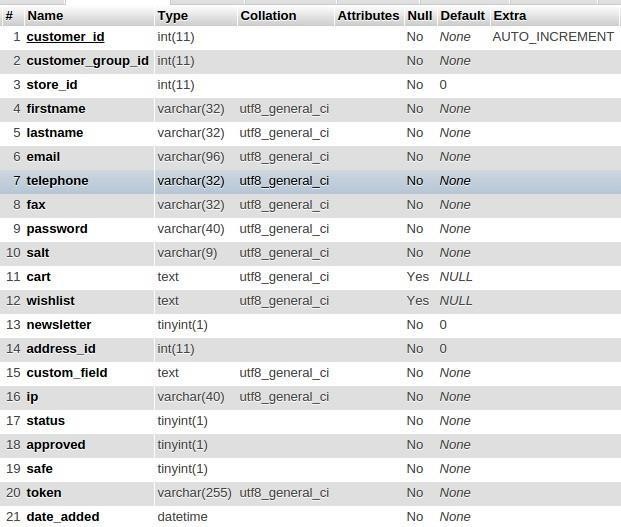
**Database structure and table details**

**Category Table**

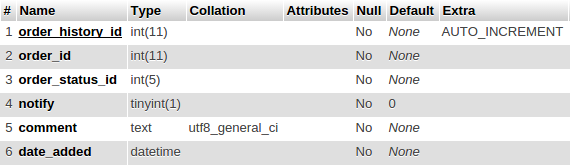


**Manufacturer Table**

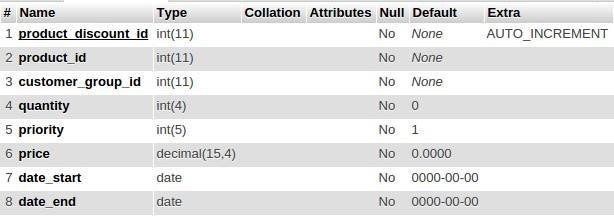
**customer table Description**



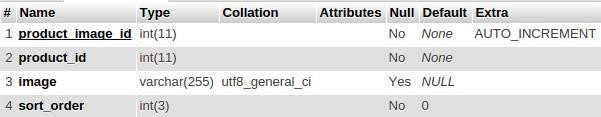
**Order History Table Description**



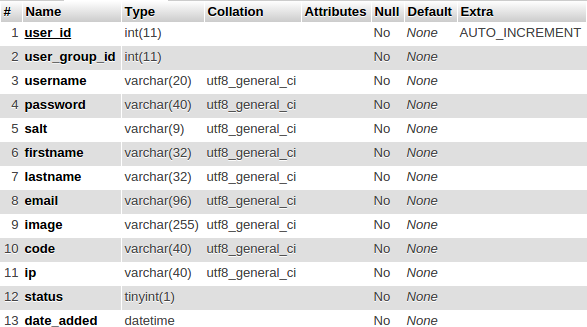
**Product *Discount* Table Description**



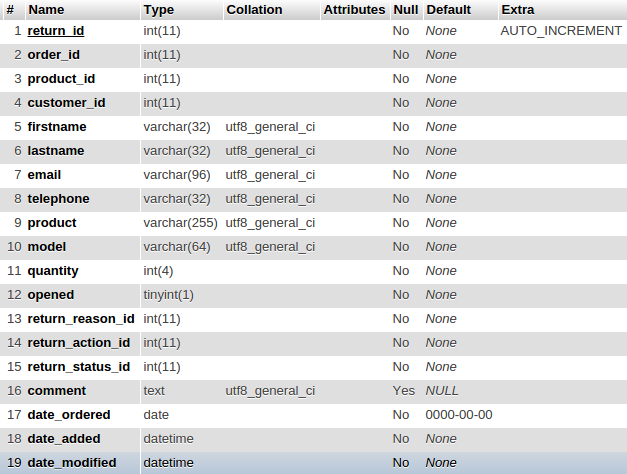
**Product Image *Table* Description**



**User *Table* Description**



#### Return Table Description



**Module Description**

**Orders Module:** When a customer goes through checkout, the information on their order is automatically transferred to the Orders section for you to keep track of. In the administration, you can view all of the orders made on their site, manually add orders, or edit the details of existing orders. The Orders section is located under Sales > Orders. On this page, every order ever made from the store is listed in detail.

**Customer Module:** Shop owners should know who their customers are and how to manage their information. In the administration, customer information will need to be stored efficiently to remember any transactions made with their account. To access customer information, you can log into the administration panel of the OpenCart store. The Customer management sections are located under Sales > Customers. There are three sections used to manage customer information: Customers, Customer Groups, and IP Blacklist.

**Mail Module:** The Mail system in OpenCart lets you send emails to specific customer groups. You can use this feature to send newsletters, information on specials, or to communicate any type of store information to select groups of customers. Customer groups are especially useful in emailing specific

**Profiles Module:** Profiles are a new, optional feature in OpenCart 1.5.6. They allow you to set up recurring payments for your customers. Profiles are useful if you plan to sell products by subscription, if you are promoting a discount or member's club, or if you are simply breaking up payments into smaller portions. This feature gives the store owner more control over how customers are charged for products or services.

**Manufacturers Module:** The Manufacturer section is used to categorize products by manufacturer. This section can be accessed under Catalog > Manufacturers. The Manufacturer names below came with the install. They can be deleted, and new manufacturer information can be added in its place. The manufacturer information should be created before products are added, so the manufacturer category can be selected when adding the product. Saving a product without including the manufacturer isn't advised, because customers can search for products in the store front through their manufacturer page. With the manufacturer missing from the product information, the product will not be available on those pages.

**Categories Module:** In the default theme of the store front, parent categories are listed in the top menu of the home page, and on the left side of product pages. This navigational feature is used to guide customers to similar products within the same category. Exposing customers to different products within a category lets the customer compare the similarities and differences between products to make the most informed purchase. When adding products to the store, you will be asked for a product category to sort them in. It is a good idea to establish these categories before adding products, to save yourself the trouble of adding the category name to the products later.

Process logic of the system

The use can visit our website to just enter our website name on browser.

Then user can see every product but can’t order without sign up in our website User can sign up in our website for order the product.

After select the product user can do modification in their order details Then user select the payment mode like cod or credit card

Then user get the receipt and date of product arrival

List of Reports

Customer reports: The Customer Reports section can be accessed under Reports>Customers. These reports display specific information to track orders made, reward points, and store credit for each customer using the store.

Order reports : The Order Report section can be accessed under Reports>Customers>Orders. The following information will be displayed about customer orders:

Customer Name E-Mail

Customer Group: The customer group the customer is categorized in.

Status: Shows if the customer is currently allowed to login to their store account. No. Orders: The total number of orders made by this customer.

No. Products: The total number of products purchased by this customer.

Total: The total amount purchased from this customer. Customers are arranged on the list from the highest amount to the lowest

Customer credit report : The Order Report section can be accessed under Reports>Customers>Credit. The following information will be displayed about customer store credit:

Customer Name E-Mail

Customer Group: The customer group the customer is categorized in.

Status: Shows if the customer is currently allowed to login to their store account. Total: The total amount purchased from this customer.

Products viewed report : The Products Viewed Report gives you an idea of what products are being viewed the most, or the least, in the store front. You can access this section under Reports > Products

* Viewed in the administration side. The report will position the top viewed product first. Along with the product name and model, the report will display the number of customers who viewed that product in the store front, and the percentage of views for this specific product out of all the views for products on the site. Clicking "Reset" will clear the information below and reset the report to only collect information from that moment on.

Sales reports : The reports under Sales Reports can be accessed from Reports Sales. Orders, taxes, shipping, returns, and coup

on sales information can be tracked in list form.

Tax Report : The Tax Report can be accessed through Reports > Sales > Tax. The Tax report displays which type of tax is being paid the most by customers in the store front. Tax information is displayed in

either daily, weekly, monthly, or yearly increments depending on which option you select in the "Group By" drop box.

The Tax Report displays the following information:

Date Start: The date of the first day of the week tracked (if set to week in Group By). Date End: The date of the last day of the week tracked (if set to week in Group By). Tax Title: The title of the tax type used.

No. of Orders: The number of orders made through the store between the date start and end. Total: The total amount of money paid for taxes by customers between the date start and end.

Shipping report : The Shipping Report can be accessed through Reports > Sales > Shipping. The Shipping report displays which shipping method is being used the most by customers in the store

front. Shipping information is displayed in either daily, weekly, monthly, or yearly increments depending on which option you select in the "Group By" drop box.

The Shipping Report displays the following information:

Date Start: The date of the first day of the week tracked (if set to week in Group By). Date End: The date of the last day of the week tracked (if set to week in Group By).

Shipping Title: The shipping method used. The shipping title with the highest total will be displayed at the top. The lowest total will be at the bottom.1

No. of Orders: The number of orders made through the store between the date start and end. Total: The total amount of money paid for shipping between the date start and end.

Returns report : The Returns Report can be accessed through Reports>Sales>Returns. The returns report shows how many returns were requested within a given time frame. The returns information is displayed in either daily, weekly, monthly, or yearly increments depending on which option you select in the "Group By" drop box.

The Returns Report displays the following information:

Date Start: The date of the first day of the week tracked (if set to week in Group By) Date End: The date of the last day of the week tracked (if set to week in Group By)

No. Returns: The number of return requests submitted or created between the date start and date end.

\

SECURITY IMPLEMENTATION

Online Transaction Security System : The online transaction security system will provide a safe and secure method for online shoppers to make credit card purchases on the e-commerce website. When a credit purchase is made using the website, the credit card information will be encrypted using Secure Socket Layer (SSL) and transmitted to the bank for processing. This security system will also provide access control for website visitors, which will allow only registered users to make purchases. The registered customers account information will be stored in the customer information database mentioned earlier.

Data Security System: The data security system will allow data to be securely transmitted between the various components of the e-commerce portal. This includes transmission of product, merchant and customer information from the content management system to the website, and also the transmission of data from the website to the content management system.

Folder Security using .htaccess: The .htaccess (short for â€œHypertextAccessâ€•) file in your siteâ€™s directory is a configuration file you can use to override the settings on your web server. With the right commands, you can enable/disable extra functionality and features to protect your site from spammers, hackers and other threats.

Prevent Directory Browsing: Protecting directories from being listed is, at best, security by obscurity. That is, its hiding your stuff from view, preventing meddling visitors from browsing through your directories. Really, its the web equivalent of hiding your cash under your mattress.Still, its good practice to prevent directory browsing, along with implementing other measures to secure your site.

To disable browsing of your directories, add this to your .htaccess file: Options All –Indexes

Restrict Access to Your Admin Area : A simple way to restrict access if your internet has a fixed IP address and you always access your site form the same location is by creating a new .htaccess file with the following snippet:

order deny,allow

allow from 192.168.5.1 deny from all

LIMITATION

First of all, if someone new things have to be added then the new designer should have the thorough knowledge of each and every year thing.

The database design is somewhat complex as every table is related with one another so the adding of some new table will be difficult.

This project has limited facility comparatively to market standard. It may many new features to suite the current market trend requirement.

Future scope and further enhancement of the project:

Today, the market place is flooded with several e-commerce options for shoppers to choose from. A

variety of innovative products and services are being offered spoiling customers for choice. Online shopping is no more a privilege enjoyed by your friends and family living in the US or UK. Today, it is a reality in India. In the last couple of years, the growth of e-commerce industry in India has been phenomenal as more shoppers have started discovering the benefits of using this platform. There is enough scope for online businesses in the future if they understand the Indian shoppers psyche and cater to their needs.

# TITLE OF PROJECT

**“E-COMMERCE =PROTAL”**

## ACKNOWLEDGEMENT

I have made a great effort for making this project .I would thanks to

IGNOU for giving a golden opportunity to work on project. It a my great pleasure to take this opportunity to acknowledge the contribution of a number of people who helped me in completing this project .

First I would like express my heartfelt gratitude and sincere thanks to **advity abrol** my project guide for giving me an opportunity to work inder him and allotting me this project.

I offer my sincere regards to the faculties members who taught me their continuous guidelines.

## (PRINCE )

#### INTRODUCTION

An eCommerce website is an information technology method in which trader, businesses/distributor/marketers can sell products/services and the customer can purchase on that website electronically by using internet on the mobile and computer. It means an e- commerce website is an online shop. e means electronic. Commerce mean business. Website means the group of HTML web pages and that is created to market/sell information/product/services.

In a bigger perspective, every website on the internet is the eCommerce Website. It can be the platform, it can be a marketplace, it can be portal, it can be apps, it can be an entertainment website, shopping website, online courses website and online degree college.

Examples of Ecommerce Website and business:

When you purchase a mobile phone /shoes/software/ flowers on any website such as Amazon, Flipkart etc. and pay through credit/debit card and then seller deliver the product through courier or post mail on your location then it’s called e-commerce. In this case, Flipkart is an online store website or an e- commerce website.

When you subscribe to watch a cricket match, movies, and shows on any website such as hotstar through debit card and credit card it is called e-commerce. And in this way, hotstar is a digital and mobile entertainment e-commerce website.

When you rent or buy movies on YouTube and pay to watch by using the mobile/computer and internet it’s called e-commerce. In this method, you have used computer/mobile and internet through electricity and visited youtube website to watch/buy/rent the movie and paid through debit card/credit card/ net banking/payment wallet etc. It means youtube is an eCommerce website in which you can buy/watch/rent the latest movies and shows.

When you use Google Ads or Facebook Advertising etc. to promote and advertise your products/services online and pay Google and Facebook to use the services and platforms then it’s e-commerce. In this case Google / Facebook etc. are e-commerce companies that provide you platform and tools to advertise and promote your business/products/services online.

When you launch your apps on the Google Play store, you pay Google to use their platform to connect with your customers/target audience then it’s e-commerce and google play store is an eCommerce platform in which apps developer or apps launcher or owners have to pay Google.

And all these

transactions are online. It’s e-commerce.

When you recharge your mobile phone/dish tv/internet data pack by using the internet and the website such as paytm, mobiwiki, JIO etc. and pay through debit/credit card, wallet, and

net banking then it’s e- commerce and PayTM, Mobiwiki, Jio apps or websites are e-commerce websites. In which they are doing customers recharge online and getting paid directly to their bank account.

When you purchase software as a service, platform as a service, infrastructure as service for your business from cloud computing service providers such as Alibaba, Amazon web service, Microsoft, google cloud etc. on their website then these are the e-commerce websites.

When you use internet banking then it is e-commerce. You pay bills, transfer money, open RD/FD account, pay installments online, pay for offline products from payment wallets etc. are e-commerce.

When political parties, government or non-government organizations received funds or donations online then it is e-commerce.

#### OBJECTIVE

E-commerce stands for electronic commerce means to transfer the conversation or information across the internet. E-Commerce allows people to sell and buys goods or services over an electronic medium, like the internet. We all know very well e-commerce is online shopping.

If you are managing any business you always focus on the extra customer. A merchant always thinks about it that how can he get more customer. The things behind the e-commerce that a merchant doesn't have to promote his business as an offline market.

The motto of e-commerce activity is to reach millions of customers easily and to increase sales in business. It generates a high revenue in the online industry as the viewers are turning into the audience every day.

To spread the small scale retails into large level business, e-commerce is developed with best- in-class features. In simple, it is a big opportunity for small vendors to huge traffic of customers.

By opening an e-commerce business a merchant can grab many benefits like sell worldwide, get more traffic, promote business etc.

Let’s look at some goals of e-commerce -

Catch more customer. Increase the traffic.

Make more sale.

Build a good will.

Best customer service. Minimum Shipping time. Increase the number of reviews. Get Positive feedback.

Reduce the number of refund item. Customer follow up

Here are some advantages of E-Commerce -

1. Cost-effective
2. Secure
3. Easy to find products
4. Faster buying & selling system
5. Doesn't require any physical shop.
6. Represent your business in the world
7. A huge amount of profit
8. Gain your new customer
9. Boost conversation rate

#### Identification of the need

1. Build a customer email list and market directly to them

One of the most important aspects of having your own website is that you can market directly to website visitors and customers. Unlike marketplaces, where people who buy your product are the customers of the marketplace, selling directly to consumers on your website means you get to collect their contact information. When you have your customers’ email addresses, you can send them promotions, offer discounts, and announce new products.

1. Establish and strengthen your brand

When you sell your products on a marketplace, they are listed in a generic way. From character limits or word count restrictions to logo usage, there is little to no room for customization or branding. In fact, it is the marketplace’s brand that is in the forefront – not yours. This makes it nearly impossible to build brand awareness and recognition.

1. Learn more about your audience

Having your own ecommerce website helps you better understand your customers. This includes demographics such as their location, as well as how they found your website and heard about you. You can analyze their behavior on your website, like what they looked at and the path they took to buy from you.

5. It’s never been easier with plug-and-play ecommerce platforms

With the rise of out-of-the-box ecommerce platforms like Shopify, Magento, and WooCommerce, creating a website can be a pretty straightforward and fast process. You can enter your details, choose from free or cheap templates, and create the shopping experience you want for your customers.

## Feasibility study

feasibility study includes an estimate of the level of expertise required for a project and who can provide it, quantitative and qualitative assessments of other essential resources, identification of critical points, a general timetable, and a general cost estimate.

Whether a project is viable or not, i.e. whether it can generate an equal or a higher rate of return during its lifetime requires a thorough investigation of the investment per se as well as the level of current expenditure. The preliminary design is the simple description of the conceived idea with an indication of the main factors to be considered in the study.

“feasibility is the determination of whether or not a project is worth doing . The process followed in making this determination I s called a feasibility study . This type of study determines if a project can and should ne taker . Once it has been determined that project of a project can and should be taken . Once it has been determined that project is feasible, the analyst can go ahead and prepare the project specification which finalize project requirements

. Generally , feasibility studies are undertaken within tight report. The contest and recommendation of such a study will be used as a sound basis for deciding to whether to proceed , postpone or cancel the project Thus since the feasibility study may lead to the commitment of large resources it becomes necessary that it should be conducted

competently and no fundamental error of judgment are made” Feasibility study comprises the following steps.

* + Information assessment: Identifies information about whether the system helps in achieving the objectives of the organization. It also verifies that the system can be implemented using new technology and within the budget and whether the system can be integrated with the existing system.
  + Information collection: Specifies the sources from where information about software can be obtained. Generally, these sources include users (who will operate the software), organization (where the software will be used), and the software development team (which understands user requirements and knows how to fulfill them in software).
  + Report writing: Uses a feasibility report, which is the conclusion of the feasibility study by the software development team. It includes the recommendations whether the software development should continue. This report may also include information about changes in the software scope, budget, and schedule and suggestions of any requirements in the system.
  + General information: Describes the purpose and scope of feasibility study. It also describes system overview, project references, acronyms and abbreviations, and points of contact to be used. System overview provides description about the name of the organization responsible for the software development, system name or title, system category, operational status, and so on. Project references provide a list of the references used to prepare this document such as documents relating to the project or previously developed documents that are related to the project. Acronyms and abbreviations provide a list of the terms that are used in this document along with their meanings. Points of contact provide a list of points of organizational contact with users for information and coordination. For example, users require assistance to solve problems (such as troubleshooting) and collect information such as contact number, e-mail address, and so on.
  + Environment: Identifies the individuals responsible for software development. It provides information about input and output requirements, processing requirements of the software and the interaction of the software with other software. It also identifies system security requirements and the system's processing requirements
  + Current functional procedures: Describes the current functional procedures of the existing system, whether automated or manual. It also includes the data-flow of the current system and the number of team members required to operate and maintain the software.
  + Functional objective: Provides information about functions of the system such as new services, increased capacity, and so on.
  + Performance objective: Provides information about performance objectives such as reduced staff and equipment costs, increased processing speeds of software, and improvedcontrols.
  + Assumptions and constraints: Provides information about assumptions and constraints such as operational life of the proposed software, financial constraints, changing hardware, software and operating environment, and availability of information and sources.
  + Methodology: Describes the methods that are applied to evaluate the proposed software in order to reach a feasible alternative. These methods include survey, modeling,

benchmarking,etc.

* + Evaluation criteria: Identifies criteria such as cost, priority, development time, and ease of system use, which are applicable for the development process to determine the most suitable systemoption.
  + Recommendation: Describes a recommendation for the proposed system. This includes the delays and acceptable risks.
  + Proposed software: Describes the overall concept of the system as well as the procedure to be used to meet user requirements. In addition, it provides information about improvements, time and resource costs, and impacts. Improvements are performed to enhance the functionality and performance of the existing software. Time and resource costs include the costs associated with software development from its requirements to its maintenance and staff training. Impacts describe the possibility of future happenings and include various types of impacts as listed below.
  + Equipment impacts: Determine new equipment requirements and changes to be made in the currently available equipment requirements.
  + Software impacts: Specify any additions or modifications required in the existing software and supporting software to adapt to the proposed software.
  + Organizational impacts: Describe any changes in organization, staff and skills requirement.
  + Operational impacts: Describe effects on operations such as user-operating procedures,data processing, data entry procedures, and so on.
  + Developmental impacts: Specify developmental impacts such as resources required to develop databases, resources required to develop and test the software, and specific activities to be performed by users during software development.
  + Security impacts: Describe security factors that may influence the development, design, and continued operation of the proposed software.
  + Alternative systems: Provide description of alternative systems, which are considered in a feasibility study. This also describes the reasons for choosing a particular alternative system to develop the proposed software and the reason for rejecting alternative systems.

#### Technical feasibility

Technical feasibility assesses the current resources (such as hardware and software) and technology, which are required to accomplish user requirements in the software within the allocated time and budget. For this, the software development team ascertains whether the current resources and technology can be upgraded or added in the software to accomplish specified user requirements.

Technical feasibility also performs the following tasks.

* + Analyzes the technical skills and capabilities of the software development team members.
  + Determines whether the relevant technology is stable and established.
  + Ascertains that the technology chosen for software development has a large number of users so that they can be consulted when problems arise or improvements are required.

Operational feasibility assesses the extent to which the required software performs a series of steps to solve business problems and user requirements. This feasibility is dependent on human resources (software development team) and involves visualizing whether the software will operate after it is developed and be operative once it is installed. Operational feasibility also performs the following tasks.

* + Determines whether the problems anticipated in user requirements are of high priority.
  + Determines whether the solution suggested by the software development team is acceptable.
  + Analyzes whether users will adapt to a new software.
  + Determines whether the organization is satisfied by the alternative solutions proposed by the software development team.

#### operational feasibility

Operational feasibility is mainly concerned with issues like whether the system will be used if it is developed and implemented. Whether there will be resistance from users that will effect the possible application benefits? The essential questions that help in testing the operational feasibility of a system are following.

Does management support the project?

Are the users not happy with current business practices? Will it reduce the time (operation) considerably? If yes, then they will welcome the change and the new system.

Have the users been involved in the planning and development of the project? Early involvement reduces the probability of resistance towards the new system.

Will the proposed system really benefit the organization? Does the overall response increase? Will accessibility of information be lost? Will the system effect the customers in considerable way?

#### Economic Feasibility

For any system if the expected benefits equal or exceed the expected costs, the system can be judged to be economically feasible. In economic feasibility, cost benefit analysis is done in which expected costs and benefits are evaluated. Economic analysis is used for evaluating the effectiveness of the proposed system.

In economic feasibility, the most important is cost-benefit analysis. As the name suggests, it is an analysis of the costs to be incurred in the system and benefits derivable out of the system. Click on the link below which will get you to the page that explains what cost benefit analysis is and how you can perform a cost benefit analysis.

Cost Benefit Analysis

#### SOFTWARE REQUIREMENT SPECIFICATION

A software requirements specification (SRS) is a document that captures complete description about how the system is expected to perform. It is usually signed off at the end of requirements engineering phase.

Qualities of SRS:

* + - Correct
    - Unambiguous
    - Complete
    - Consistent
    - Ranked for importance and/or stability
    - Verifiable
    - Modifiable
    - Traceable

A **software requirements specification** (SRS) is a detailed description of a software system to be developed with its functional and non-functional requirements. The SRS is developed based the agreement between customer and contractors. It may include the use cases of how user is going to interact with software system. The software requirement specification document consistent of all necessary requirements required for project development. To develop the software system we should have clear understanding of Software system. To achieve this we need to continuous communication with customers to gather all requirements.

A good SRS defines the how Software System will interact with all internal modules, hardware, communication with other programs and human user interactions with wide range of real life scenarios. Using the *Software requirements specification* (SRS) document on QA lead, managers creates test plan. It is very important that testers must be cleared with every detail specified in this document in order to avoid faults in test cases and its expected results. It is highly recommended to review or test SRS documents before start writing t est cases and making any plan for testing. Let’s see how to test SRS and the important point to keep in mind while testing it

1. Correctness of SRS should be checked. Since the whole testing phase is dependent on SRS, it is very important to check its correctness. There are some standards with which we can compare and verify.
2. Ambiguity should be avoided. Sometimes in SRS, some words have more than one meaning and this might confused testers making it difficult to get the exact reference. It is advisable to check for such ambiguous words and make the meaning clear for better understanding.
3. Requirements should be complete. When tester writes test cases, what exactly is required fromthe application, is the first thing which needs to be clear. For e.g. if application needs to send the specific data of some specific size then it should be clearly mentioned in SRS that how much data and what is the size limit to send.
4. Consistent requirements.The SRS should be consistent within itself and consistent to its reference documents. If you call an input “Start and Stop” in one place, don’t call it “Start/Stop” in another. This sets the standard and should be followed throughout the testing phase.
5. Verification of expected result: SRS should not have statements like “Work as expected”, it should be clearly stated that what is expected since different testers would have different thinking aspects and may draw different results from this statement.
6. Testing environment: some applications need specific conditions to test and also a particular environment for accurate result. SRS should have clear documentation on what type of environment is needed to set up.
7. Pre-conditions defined clearly: one of the most important part of test cases is pre-conditions. If they are not met properly then actual result will always be different expected result. Verify that in SRS, all the pre-conditions are mentioned clearly.
8. Requirements ID: these are the base of test case template. Based on requirement Ids, test case ids are written. Also, requirements ids make it easy to categorize modules so just by looking at them, tester will know which module to refer. SRS must have them such as id defines a particular module.
9. Security and Performance criteria: security is priority when a software is tested especially when it is built in such a way that it contains some crucial information when leaked can cause harm to business. Tester should check that all the security related requirements are properly defined and are clear tohim. Also, when we talk about performance of a software, it plays a very important role in business so all the requirements related to performance must be clear to the tester and he must also know when and how much stress or load testing should be done to test the performance.
10. Assumption should be avoided: sometimes when requirement is not cleared to tester, he tends to make some assumptions related to it, which is not a right way to do testing as assumptions could go wrong and hence, test results may vary. It is better to avoid

assumptions and ask clients about all the “missing requirements” to have a better understanding of expected results.

1. Deletion of irrelevant requirements: there are more than one team who work on SRS so it

might be possible that some irrelevant requirements are included in SRS. Based on the understanding of the software, tester can find out which are these requirements and remove them to avoid confusions and reduce work load.

1. Freeze requirements: when an ambiguous or incomplete requirement is sent to client to analyze and tester gets a reply, that requirement result will be updated in the next SRS version and client will freeze that requirement. Freezing here means that result will not change again until and unless some major addition or modification is introduced in the software.

Most of the defects which we find during testing are because of either incomplete requirements or ambiguity in SRS. To avoid such defects it is very important to test software requirements specification before writing the test cases. Keep the latest version of SRS with you for reference and keep yourself updated with the latest change made to the SRS. Best practice is to go through the document very carefully and note down all the confusions, assumptions and incomplete requirements and then have a meeting with the client to get them clear before development phase starts as it becomes costly to fix the bugs after the software is developed. After all the requirements are cleared to a tester, it becomes easy for him to write effective test cases and accurate expected results.

Over to you: I think I have addressed all major points of Software requirements specification. Have you ever worked on “Testing of software requirements specification (SRS) document”. If yes then please make sure that you share it with your QA friends. Please leave your

questions/tips/suggestions in the comment section below and I’ll try to answer as many as I can.

Here is the detailed description given which is being used in the EMIS to manage the entire work .

1. Information about customer
2. Information about product
3. Information about seller
4. Information about payment gateway
5. Information about delivery

#### Tools and Requirement Software requirement

|  |  |
| --- | --- |
| FRONT END | Html , CSS, JAVASCRIPT |
| OPERATING SYSTEM | WINDOWS 7 |
| WEB SERVER | HTTP,URLs |
| BACK END | Database |
| LANGUAGE | Python |
| PROCESSOR | DUAL CORE AT LEAST |
| FRAMEWORK | DJANGO |

**Hardware Requirement**

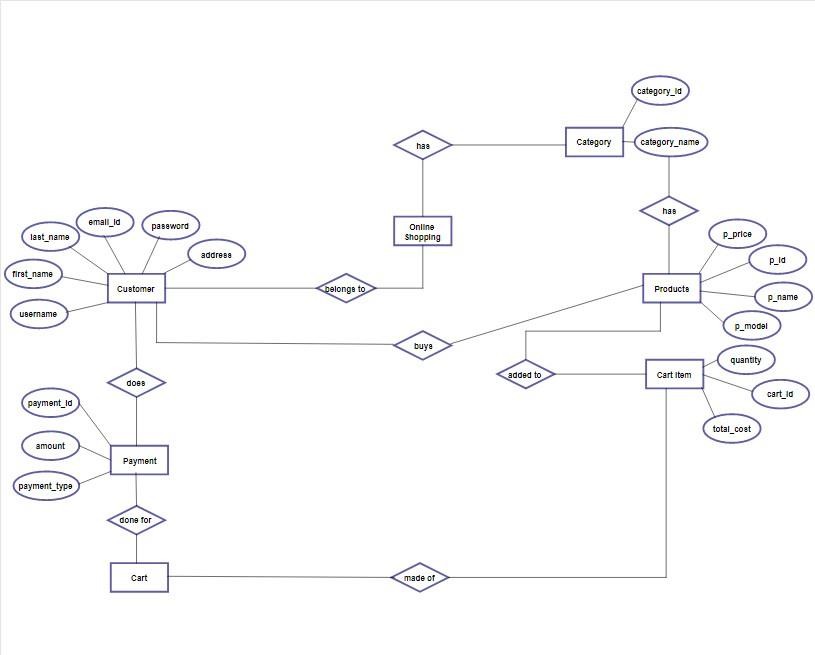
|  |  |
| --- | --- |
|  |  |
| RAM | 2 gb |
| Hard disk | 250 mb |
| Internet | 10 mbps |
|  |  |
| PROCESSOR | DUAL CORE AT LEAST |
|  |  |

**System specification**

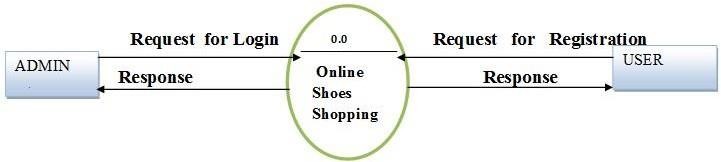
Client Pc : 1 no each

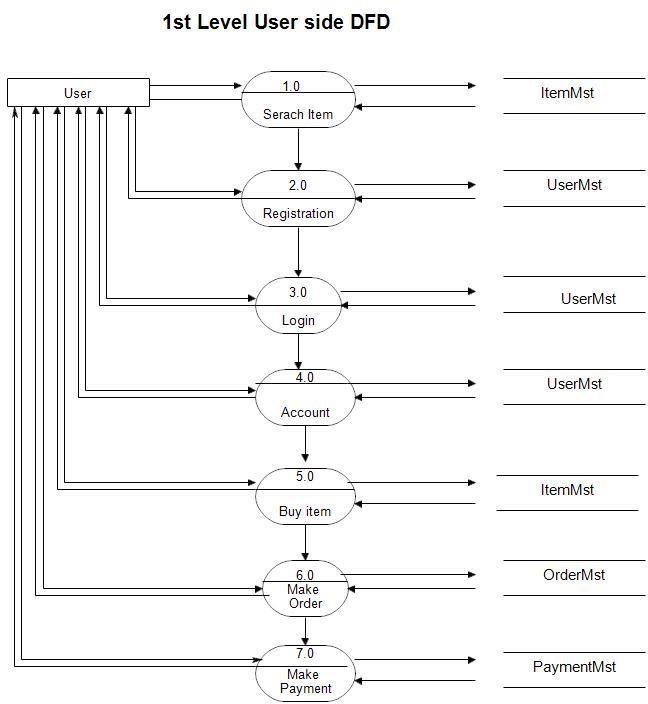
Intel dual core processer minimum or android /ios mobile Ram 1 gb Internet connection

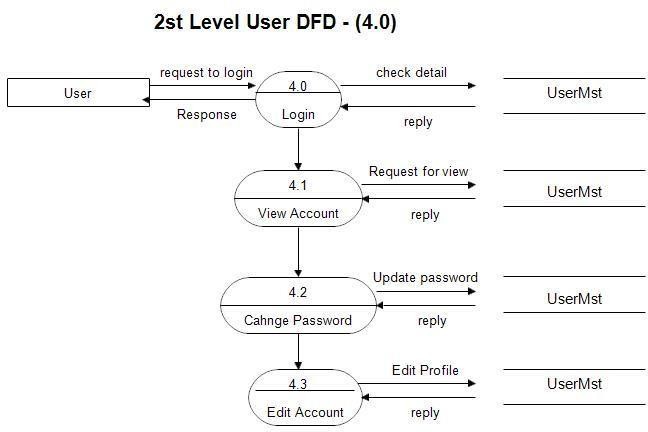
#### E R DIAGRAM

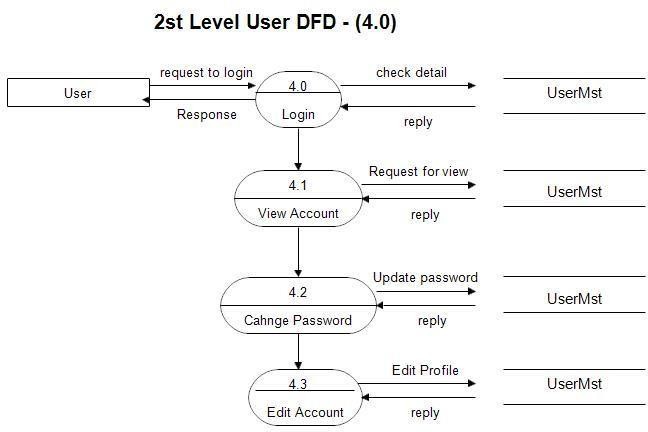


**DFD 0 LEVEL**

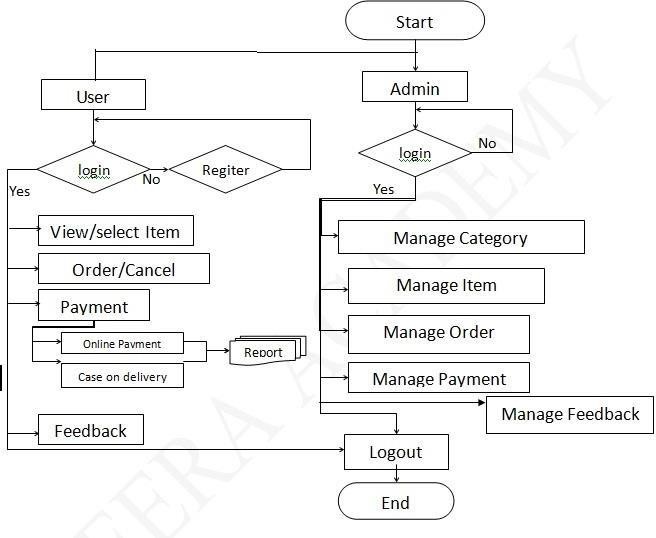








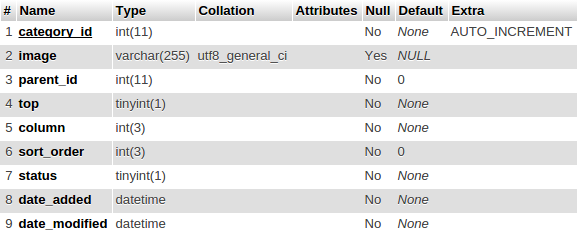
**System flow digram**



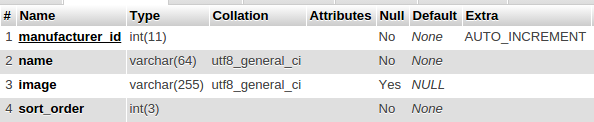
**Process Logic of system**

* + search website on Google “website name shop-online”
  + login on website with email id
  + search product and add to cart
  + enter address , mobile number all necessary credentials
  + next choose payment mode “ cod ,paytm , paypal etc”
  + get estimate date

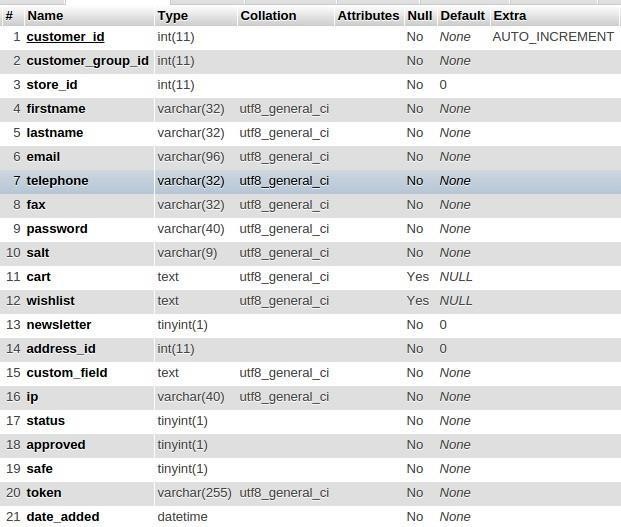
#### Database Structure and Table Details Category Table



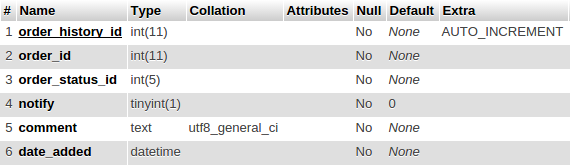
**Manufacturer Table**



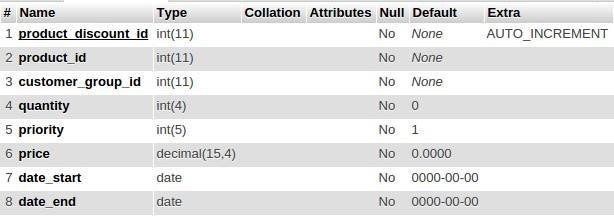
**Customer Table Description**



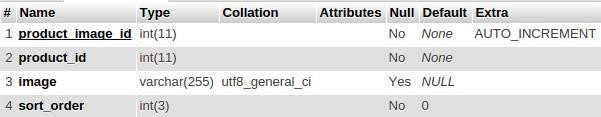
**Order History Table Description**



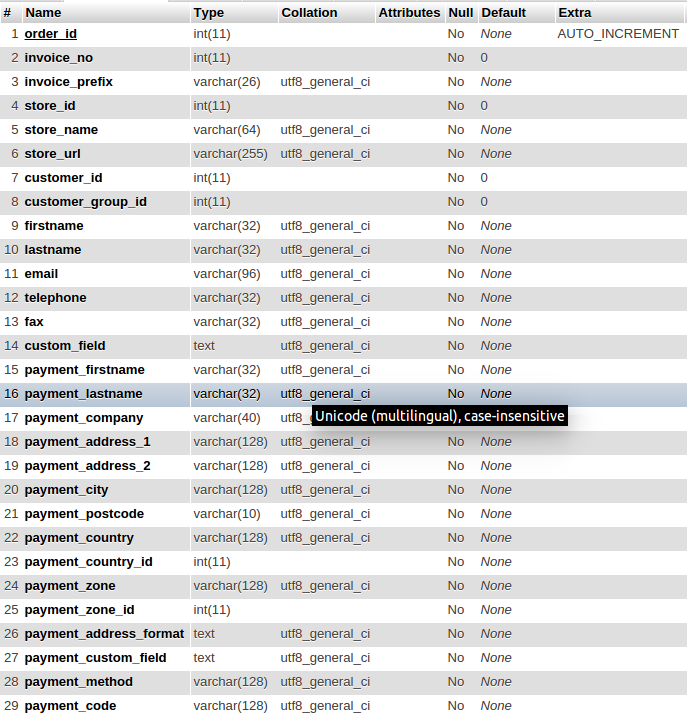
**Product Discount Table Description**



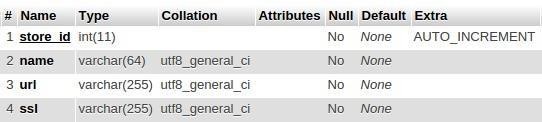
**Product Image Table Description**



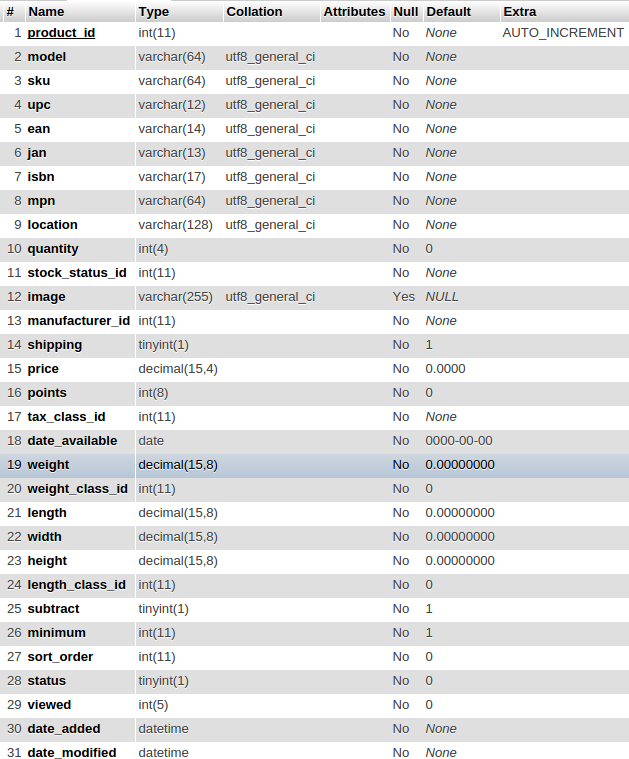
**Order Table Description**



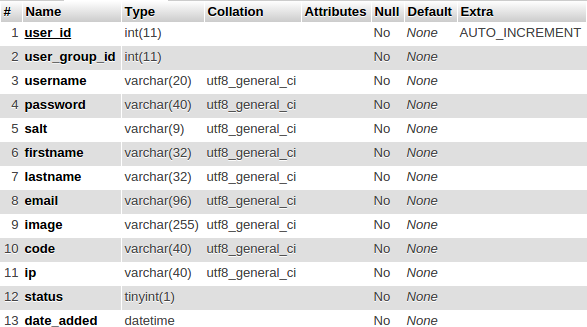
**Store Table Description**



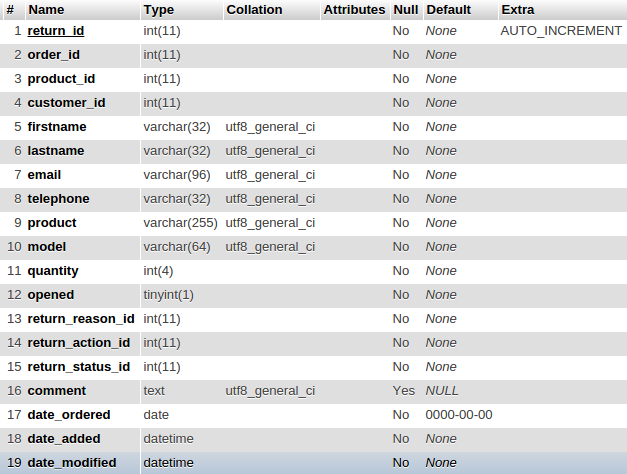
**Product Table Description**

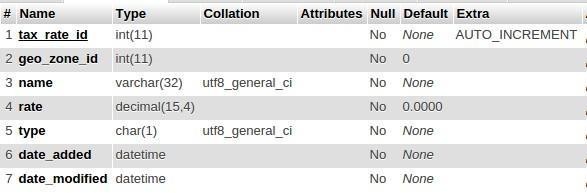


**User Table Description**



**Return Table Description**

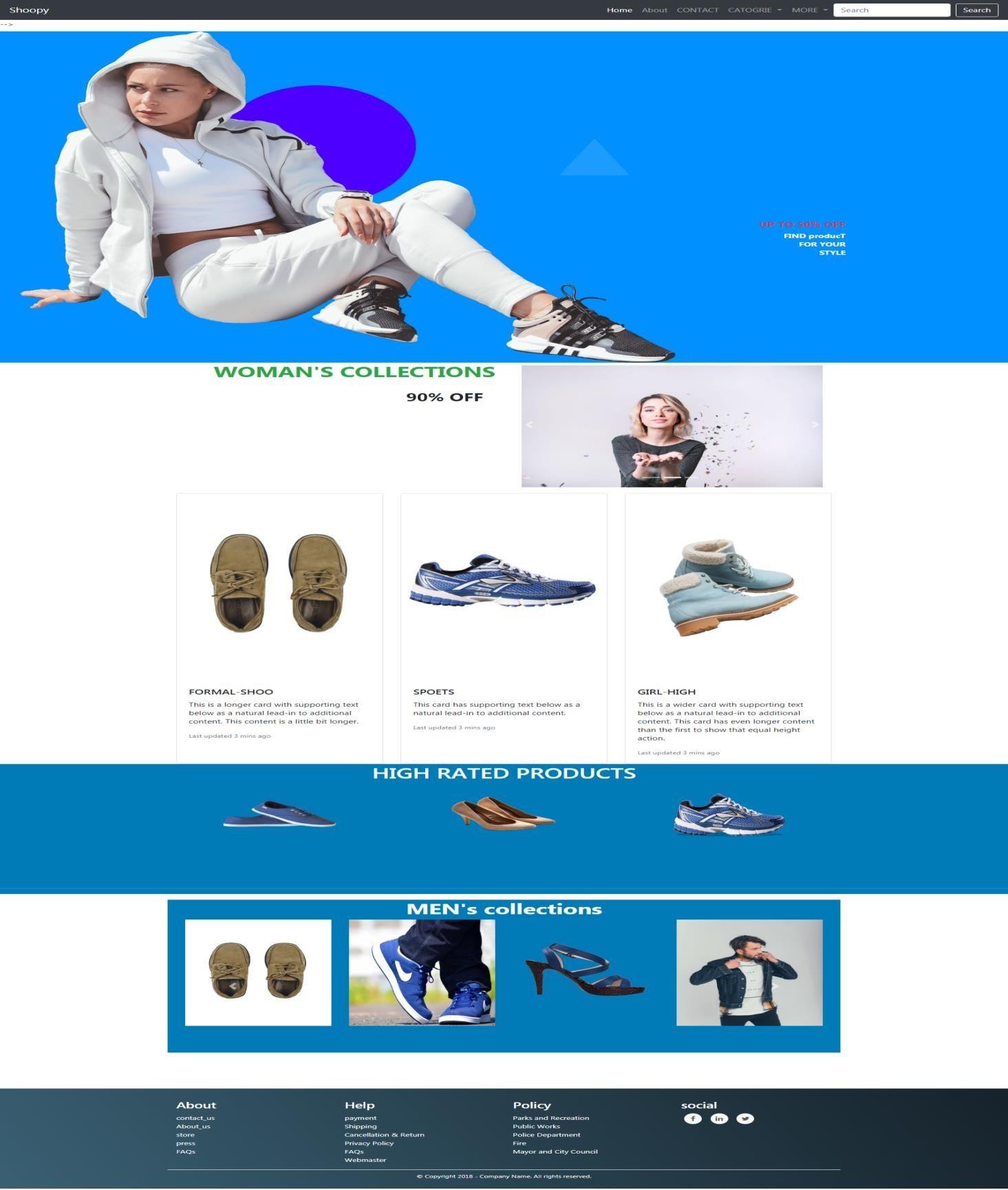




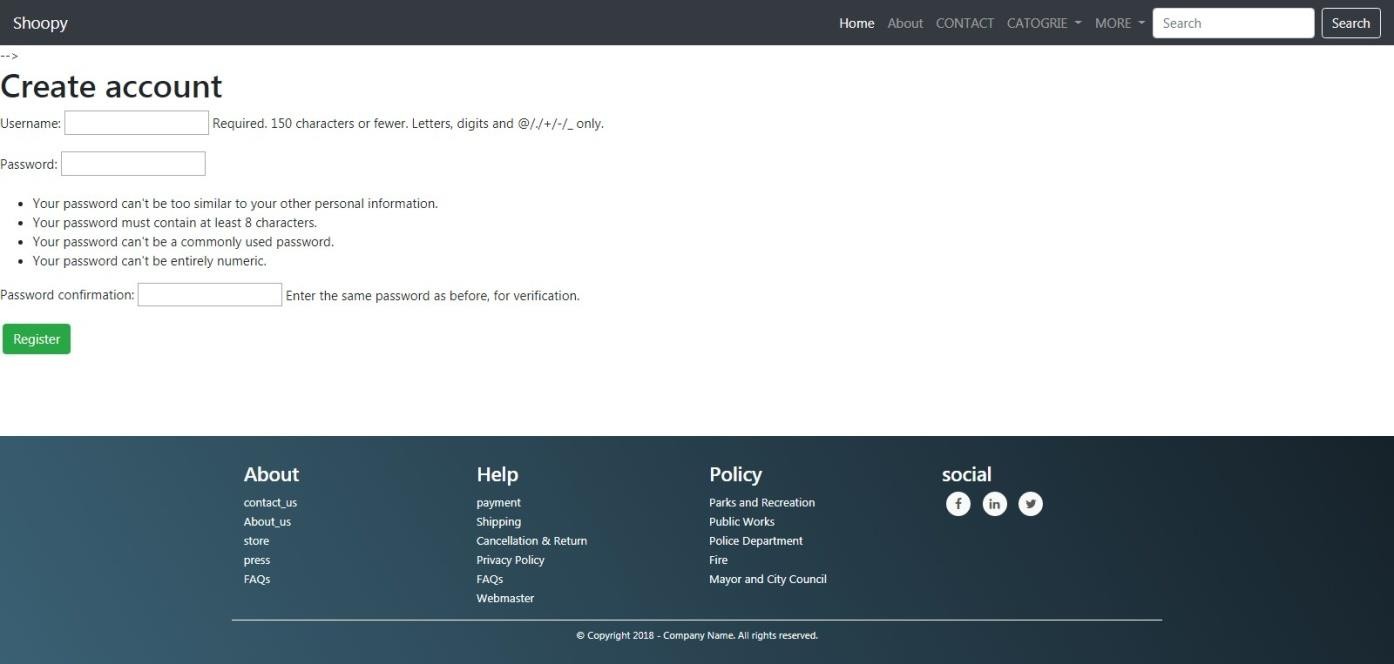
**Tax Rate Table Description**

**USER INTERFACE**

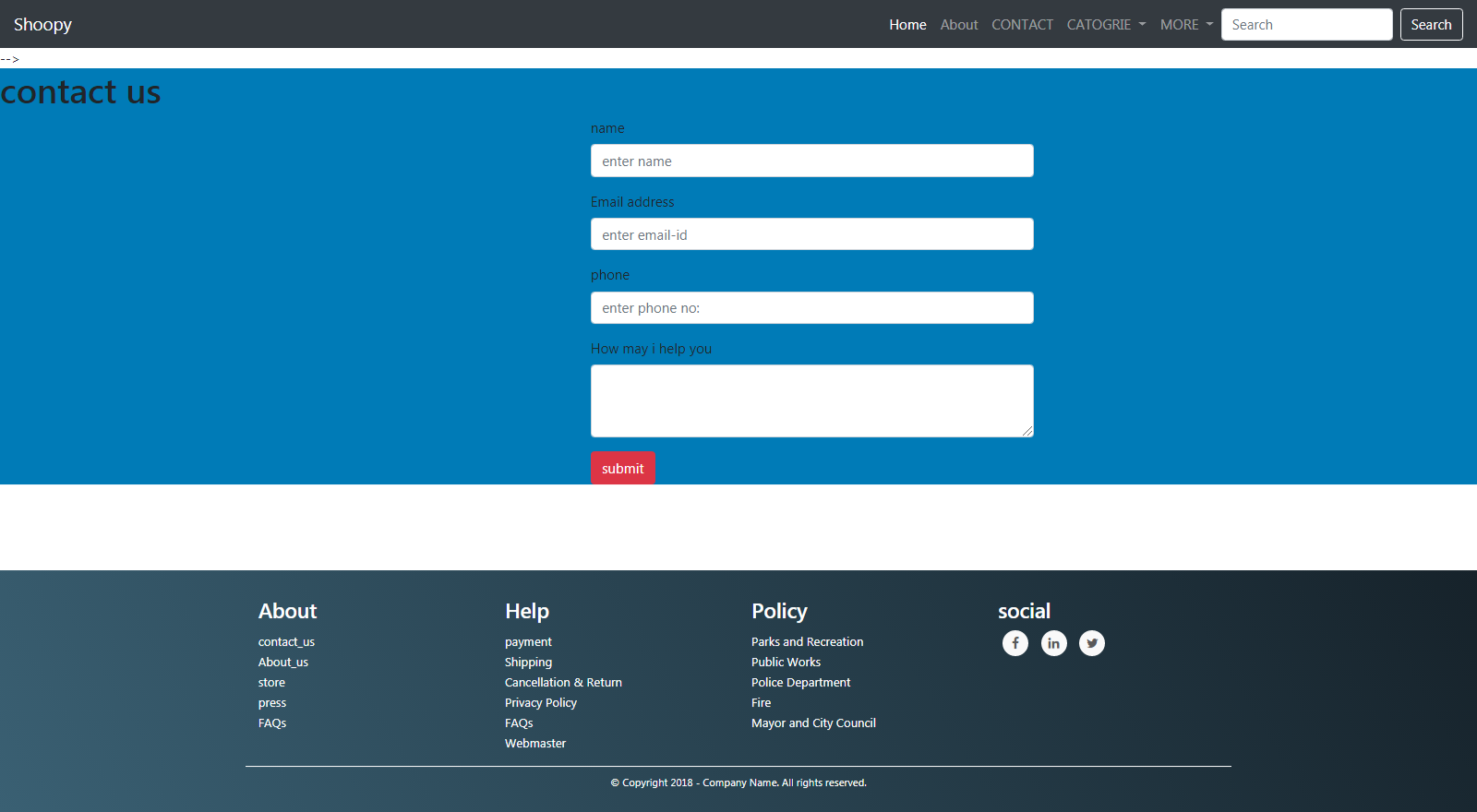
**HOME PAGE DESIGN**



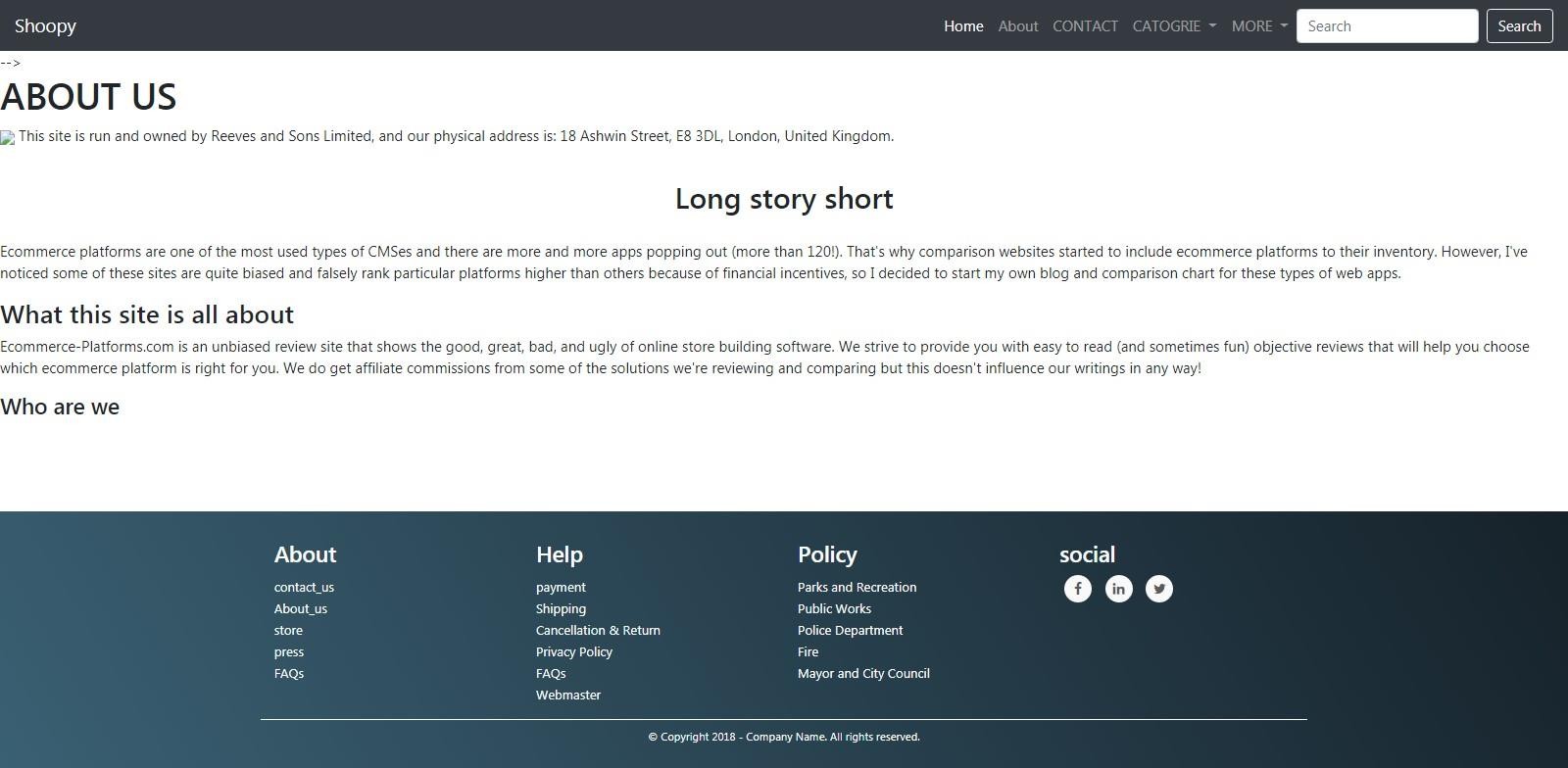
**REGISTRATION PAGE**



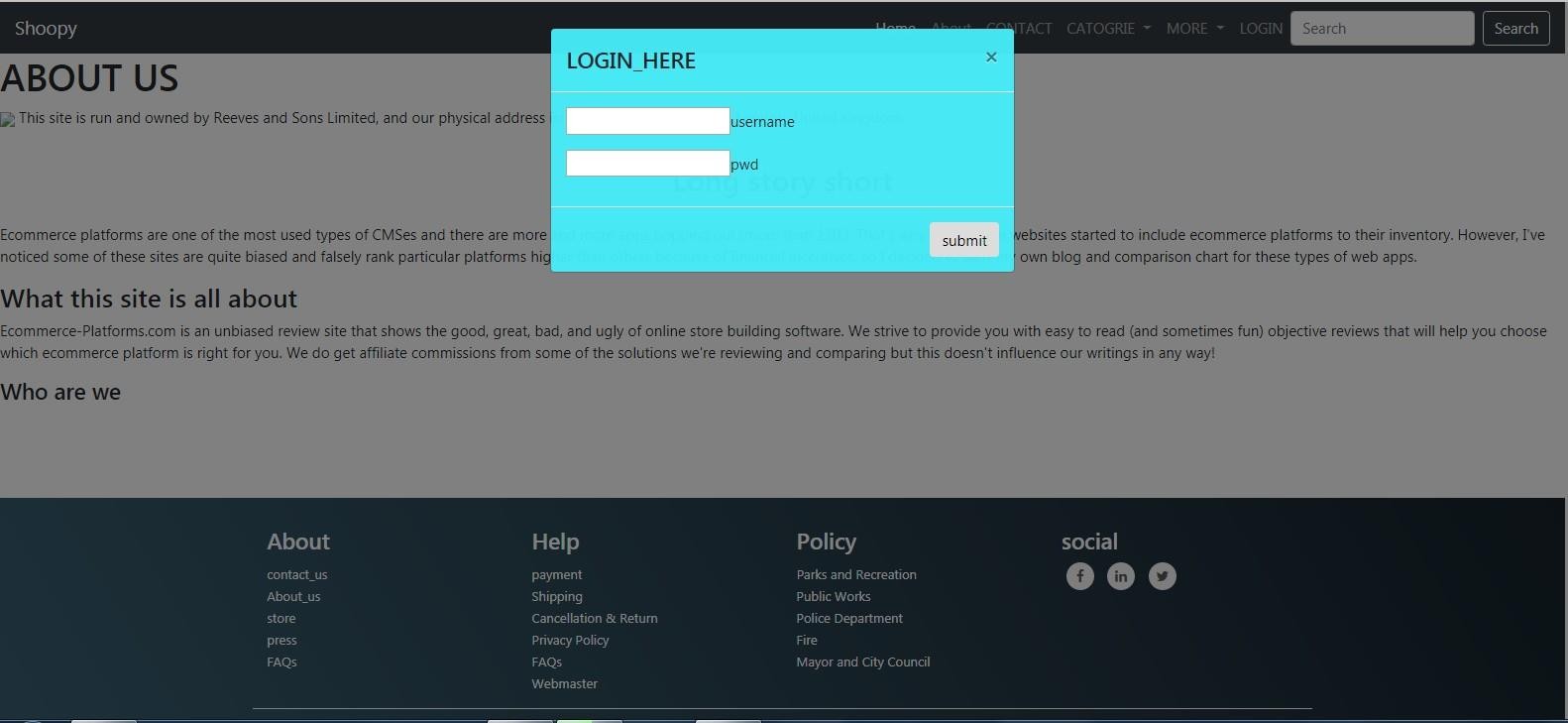
**CONTACT US PAGE**



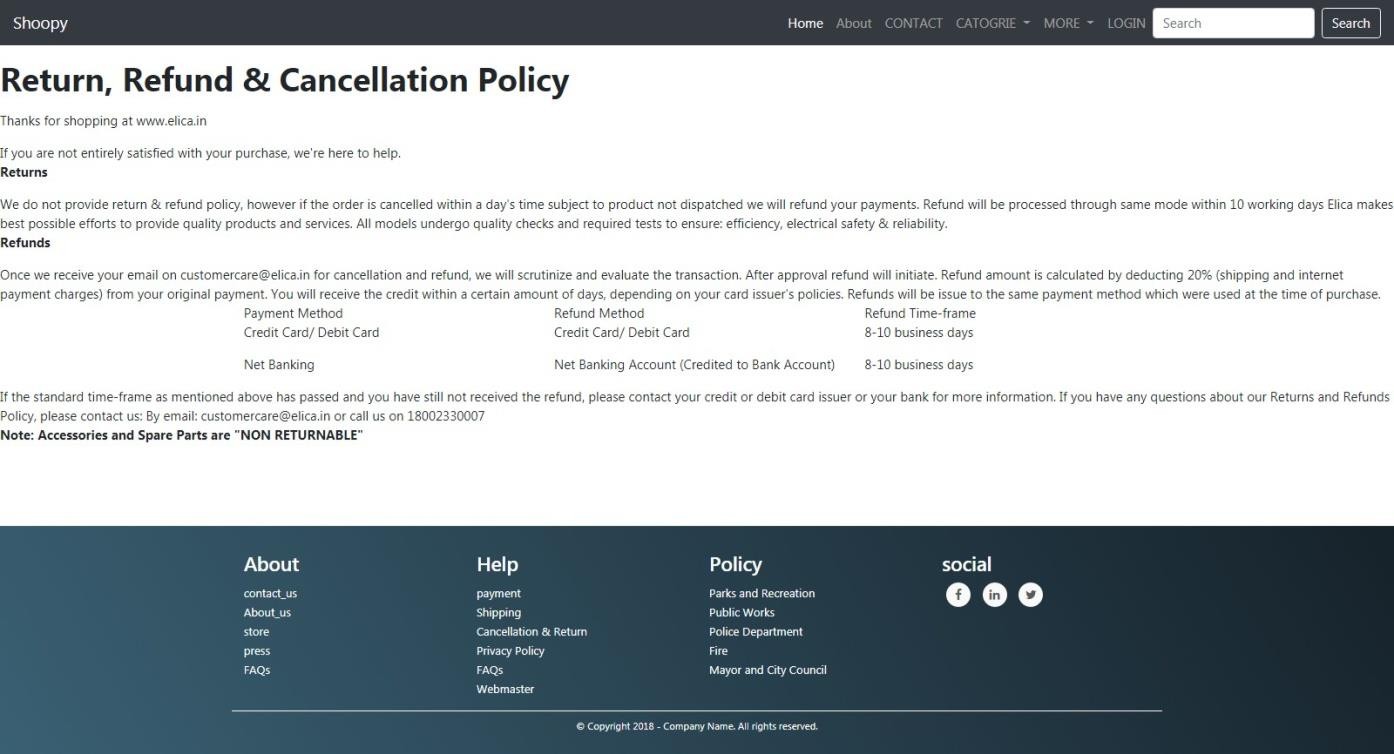
**ABOUT US PAGE**



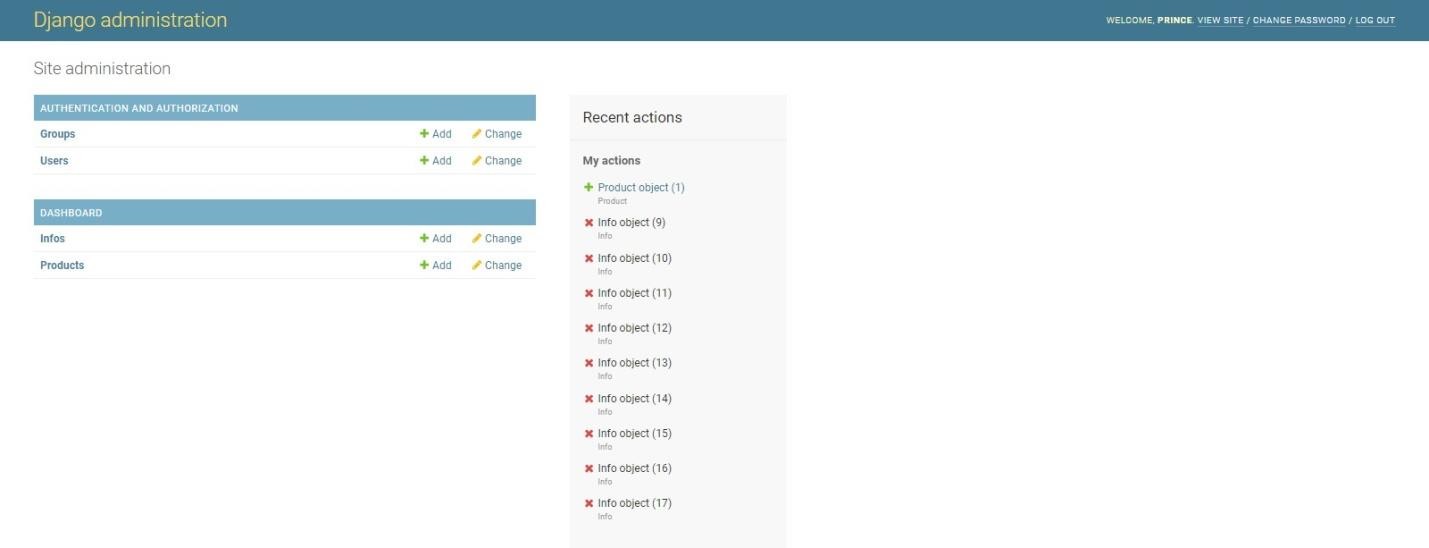
**LOGIN PAGE**



**RETURN AND CANCELLATION**



**ADMINSTRATION PAGE**



**CODING**

Templates coding in html

{% extends 'dashboard/basic.html' %} {% block body%}

<section>

<img src="static/dashboard/image/choose-2.jpg" class="img-fluid mt-2" alt="Responsive image">

<div class="carousel-caption d-l d-md-block">

<h5 class="text-right font-weight-bold text-danger">UP TO 50% OFF </h5>

<p class="text-right font-weight-bold text-WHITE"style="padding-left: 90%">FIND producT FOR YOUR STYLE</div>

</div>

</section>

<section ><div class="container">

<div class="row">

<div class="col">

<h1 class="text-right font-weight-bold text-success">WOMAN'S COLLECTIONS </h1>

<h2 class="text-right font-weight-bold ">90% OFF</h2>

</div>

<div class="col">

<div class="container-fluid mt-2" >

<div id="myCarousel" class="carousel slide" data-ride="carousel" style="width: 100%; background-color: black">

<!-- Indicators -->

<ul class="carousel-indicators">

<li data-target="#myCarousel" data-slide-to="0" class="active"></li>

<li data-target="#myCarousel" data-slide-to="1"></li>

<li data-target="#myCarousel" data-slide-to="2"></li>

</ul>

<!-- The slideshow -->

<div class="carousel-inner" style="background-color: black; width: 100%;">

<div class="carousel-item active" >

<img src="static/dashboard/image/image\_1.jpg" alt="...">

</div>

<div class="carousel-item">

<img src="static/dashboard/image/image\_2.jpg" alt="...">

</div>

</div>

<div class="carousel-item">

<img src="static/dashboard/image/image\_3.jpg" alt="...">

</div>

</div>

<!-- Left and right controls -->

<a class="carousel-control-prev" href="#myCarousel" data-slide="prev">

<span class="carousel-control-prev-icon"></span>

</a>

<a class="carousel-control-next" href="#myCarousel" data-slide="next">

<span class="carousel-control-next-icon"></span>

</a>

</div>

</div></div>

</div>

</div>

</section>

<style>/\* Make the image fully responsive \*/

.carousel-inner img { width: 100%;

height: 30%;

;

}</style>

<section>

<div class="container mt-3">

<div class="card-deck">

<div class="card">

<img class="card-img-top" src="static/dashboard/image/product-2.jpg" alt="Card image

cap">

<div class="card-body">

<h5 class="card-title">FORMAL-SHOO</h5>

<p class="card-text">This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.</p>

<p class="card-text"><small class="text-muted">Last updated 3 mins ago</small></p>

</div>

</div>

<div class="card">

<img class="card-img-top" src="static/dashboard/image/product-3.jpg" alt="Card image

cap">

<div class="card-body">

<h5 class="card-title">SPOETS</h5>

<p class="card-text">This card has supporting text below as a natural lead-in to additional content.</p>

<p class="card-text"><small class="text-muted">Last updated 3 mins ago</small></p>

</div>

</div>

<div class="card">

<img class="card-img-top" src="static/dashboard/image/product-1.png" alt="Card image

cap">

<div class="card-body">

<h5 class="card-title">GIRL-HIGH</h5>

<p class="card-text">This is a wider card with supporting text below as a natural lead-in to additional content. This card has even longer content than the first to show that equal height action.</p>

<p class="card-text"><small class="text-muted">Last updated 3 mins ago</small></p>

</div>

</div>

</div>

</div>

</div></section>

<section class='im'>

<h1 class='text-center text-white'>HIGH RATED PRODUCTS

<div class="container">

<div class="row">

<div class="col">

<img src="static/dashboard/image/product-4.png" style="width: 60%; height: 60%" alt="">

</div><div class="col"><img src="static/dashboard/image/product-5.png" style="width: 60%; height: 60%" alt=""></div>

<div class="col"><img src="static/dashboard/image/prod-1.png" style="width: 60%; height: 60%" alt=""> </div>

</section>

<!-- /////////////////////// -->

<section class="container mt-3" style="background-color: #007bb7">

<div class="container mt-3" >

<h1 class="text-center font-weight-bold text-white">MEN's collections</h1>

<div id="carouselExampleControls" class="carousel slide" data-ride="carousel">

<div class="carousel-inner">

<div class="carousel-item active">

<div class="row">

<div class="col-3"><img class="d-block w-100" style="height: 80%" src="static/dashboard/image/product-2.jpg" alt="First slide"></div>

<div class="col-3"><img class="d-block w-100" style="height: 80%" src="static/dashboard/image/gallery-6.jpg" alt="First slide"></div>

<div class="col-3"><img class="d-block w-100" style="height: 80%" src="static/dashboard/image/product-7.png" alt="First slide"></div>

<div class="col-3"><img class="d-block w-100" style="height: 80%" src="static/dashboard/image/bg\_1.jpg" alt="First slide"></div>

</div>

</div>

<div class="carousel-item">

<div class="row">

<div class="col-3"><img class="d-block w-100" style="height: 80%" src="static/dashboard/image/product-6.png" alt="First slide"></div>

<div class="col-3"><img class="d-block w-100" style="height: 80%" src="static/dashboard/image/product-4.jpg" alt="First slide"></div>

<div class="col-3"><img class="d-block w-100"style="height: 80%" src="static/dashboard/image/product-3.png" alt="First slide"></div>

<div class="col-3"><img class="d-block w-100"style="height: 80%" src="static/dashboard/image/product-2.png" alt="First slide"></div>

</div>

</div>

<div class="carousel-item">

<div class="row">

<div class="col-3"><img class="d-block w-100" style="height: 80%" src="static/dashboard/image/product-6.png" alt="First slide"><button type="button" class="btn btn-info">cart</button></div>

<div class="col-3"><img class="d-block w-100" style="height: 80%" src="static/dashboard/image/product-4.jpg" alt="First slide"><button type="button" class="btn btn-warning">cart</button></div>

<div class="col-3"><img class="d-block w-100"style="height: 80%" src="static/dashboard/image/product-3.png" alt="First slide"><button type="button" class="btn btn-light">cart</button></div>

<div class="col-3"><img class="d-block w-100"style="height: 80%" src="static/dashboard/image/product-2.png" alt="First slide"><button type="button" class="btn btn-success">cart</button></div>

</div>

</div>

</div>

<a class="carousel-control-prev" href="#carouselExampleControls" role="button" data- slide="prev">

<span class="carousel-control-prev-icon" aria-hidden="true"></span>

<span class="sr-only">Previous</span>

</a>

<a class="carousel-control-next" href="#carouselExampleControls" role="button" data- slide="next">

<span class="carousel-control-next-icon" aria-hidden="true"></span>

<span class="sr-only">Next</span>

</a>

</div></div></section>

<script>

</script>

{% endblock %}

#### CONTACT PAGE HTML

{% extends 'dashboard/basic.html' %} {% block body%} <section style="background-color: #007bb7;width: 100%; ">

<h1>contact us </h1>

<form method="GET" style=" width:70%; padding-left: 40% ">{% csrf\_token %}

<div class="form-group">

<label for="name">name</label>

<input type="text" class="form-control" id="name" name="name" placeholder="enter name">

</div>

<div class="form-group">

<label for="email">Email address</label>

<input type="email" class="form-control" id="email" name="email" placeholder="enter email-id">

</div>

<div class="form-group">

<label for="name">phone</label>

<input type="tel" class="form-control" id="phone" name="phone" placeholder="enter phone no:">

</div>

<div class="form-group">

<label for="desc">How may i help you</label>

<textarea class="form-control" id="desc" name="desc" rows="3"></textarea>

</div>

<button type="submit" class="btn btn-danger">submit</button>

</form>

</section> {% endblock %}

#### REGISTRATION PAGE HTML CODING

{% extends 'dashboard/basic.html' %} {% block body%}

<div class="login">

{% if messages %}

<ul>

{% for message in messages %}

<li>{{ message }}</li>

{% endfor %}

</ul>

{% endif %}

<h1>Create account </h1>

<form method="post" >

{% csrf\_token %}

<table>

{{ form.as\_p }}

<tr>

<td></td>

<td><input type="submit" name="submit" value="Register" class="btn btn-success"

/></td>

</tr>

</table>

</form>

</div>

{% endblock %}

#### RETURN POLICY PAGE CODING IN HTML

{% extends 'dashboard/basic.html' %} {% block body%}

<p><h1><strong>Return, Refund & Cancellation Policy</strong></h1></p>

<p>Thanks for shopping at [www.elica.in](http://www.elica.in/)</p>

If you are not entirely satisfied with your purchase, we're here to help.

<p><strong>Returns</strong></p>

We do not provide return & refund policy, however if the order is cancelled within a day's time subject to product not dispatched we will refund your payments. Refund will be processed through same mode within 10 working days

Elica makes best possible efforts to provide quality products and services. All models undergo quality checks and required tests to ensure: efficiency, electrical safety & reliability.

<p><strong>Refunds</strong></p>

Once we receive your email on [customercare@elica.in](mailto:customercare@elica.in) for cancellation and refund, we will scrutinize and evaluate the transaction. After approval refund will initiate. Refund amount is calculated by deducting 20% (shipping and internet payment charges) from your original payment.

You will receive the credit within a certain amount of days, depending on your card issuer's policies.

Refunds will be issue to the same payment method which were used at the time of purchase.

<div class="container">

<div class="row">

<div class="col">Payment Method <p>Credit Card/ Debit Card</p><p>Net Banking</p></div>

<div class="col">Refund Method<p>Credit Card/ Debit Card</p><p>Net Banking Account (Credited to Bank Account)</p></div>

<div class="col">Refund Time-frame<p>8-10 business days</p><p>8-10 business days

</p></div>

</div>

</div>

#### About us page coding

{% extends 'dashboard/basic.html' %}{% block title%} ABOUT{% endblock %} {% block body%}

<h1>ABOUT US</h1>

<p><img src="static/dashboard/image/product-8.png" >

This site is run and owned by Reeves and Sons Limited, and our physical address is: 18 Ashwin Street, E8 3DL, London, United Kingdom.

<h2>Long story short</h2><p>

Ecommerce platforms are one of the most used types of CMSes and there are more and more apps popping out (more than 120!). That's why comparison websites started to include ecommerce platforms to their inventory. However, I've noticed some of these sites are quite biased and falsely rank particular platforms higher than others because of financial incentives, so I decided to start my own blog and comparison chart for these types of web apps.</p>

<h3> What this site is all about</h3>

<p>Ecommerce-Platforms.com is an unbiased review site that shows the good, great, bad, and ugly of online store building software. We strive to provide you with easy to read (and sometimes fun) objective reviews that will help you choose which ecommerce platform is right for you. We do get affiliate commissions from some of the solutions we're reviewing and comparing but this doesn't influence our writings in any way!</p>

<h4> Who are we</h4>

{% endblock %}

#### Basic page html

<!DOCTYPE html> {%load static%}

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title> {% block title %}{% endblock %} </title>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384- Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

<link rel="stylesheet" href="{% static 'assets/dashboard/image' %}">

<link rel="stylesheet" href="{% static 'dashboard/css/index.css' %}">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font- awesome/4.7.0/css/font-awesome.min.css">

<!-- Optional JavaScript -->

<!-- jQuery first, then Popper.js, then Bootstrap JS -->

<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384- KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384- ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384- JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl" crossorigin="anonymous"></script>

</head>

<body><nav class="navbar navbar-expand-lg navbar-dark bg-dark">

<a class="navbar-brand" href="#">Shoopy</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data- target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria- expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav ml-auto">

<li class="nav-item active">

<a class="nav-link" href="{% url 'index' %}">Home <span class="sr- only">(current)</span></a>

</li>

<li class="nav-item">

<a class="nav-link" href="{% url 'about' %}">About</a>

</li>

</li>

<li class="nav-item">

<a class="nav-link" href="{% url 'contact' %}">CONTACT</a>

</li>

<li class="nav-item">

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data- toggle="dropdown" aria-haspopup="true" aria-expanded="false">

CATOGRIE

</a>

<div class="dropdown-menu" aria-labelledby="navbarDropdown">

<a class="dropdown-item" class="nav-link dropdown" data-toggle="dropdown" id="navbarDropdown" href="#">MEN'S</a>

<div class="dropdown-menu" aria-labelledby="navbarDropdown">

<a class="dropdown-item" href="#">cupon</a>

<a class="dropdown-item" href="{% url 'login' %}">log</a>

<div class="dropdown-divider">Order</div>

<a class="dropdown-item" href="{% url 'register' %}">register</a>

</div>

<a class="dropdown-item" href="#">WOMAN'S</a>

<div class="dropdown-divider">KID'S</div>

</div>

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data- toggle="dropdown" aria-haspopup="true" aria-expanded="false">

MORE

</a>

<div class="dropdown-menu" aria-labelledby="navbarDropdown">

<a class="dropdown-item" href="{% url 'cart' %}">cart</a>

<a class="dropdown-item" href="#">SIGN-UP</a>

<div class="dropdown-divider">Order</div>

<a class="dropdown-item" href="{% url 'register' %}">register</a>

</div>

</li>

<li class="nav-item">

<a class="nav-link " data-toggle="modal" data-target="#myModal" onclick="myFunction()" href="">LOGIN</a>

</li>

</ul>

<form class="form-inline my-2 my-lg-0">

<input class="form-control mr-sm-2" type="search" placeholder="Search" aria- label="Search">

<button class="btn btn-outline-light my-2 my-sm-0" type="submit">Search</button>

</form>

</div>

</nav>

<form action="login" method="POST">

<div class="container" id="test">

<!-- Trigger the modal with a button

<!-- Modal -->

<div class="modal fade" id="myModal" role="dialog">

<div class="modal-dialog">

<!-- Modal content-->

<div class="modal-content" style="background-color: rgba(70, 236, 248, 0.966)">

<div class="modal-header">

<h4 class="modal-title">LOGIN\_HERE</h4>

<!-- <img class="img1" src="static/appor/css/avtar.jpg" padding> -->

<button type="button" class="close" data-dismiss="modal">&times;</button>

</div>

<div class="modal-body">

<p> <input type="text" name="username" id="username" >username</p>

<p> <input type="password" name="password" id="password">pwd</p>

</div>

<div class="modal-footer ">

<button type="submit " class="btn btn-default " value="submit">submit</button>

</div>

</div>

</div>

</div>

</div>

</form

{% block body%}

{% endblock %}

<footer class="mainfooter" role="contentinfo">

<div class="footer-middle">

<div class="container">

<div class="row">

<div class="col-md-3 col-sm-6">

<!--Column1-->

<div class="footer-pad">

<h4>About</h4>

<ul class="list-unstyled">

<li><a href="#"></a></li>

<li><a href="{% url 'contact' %}">contact\_us</a></li>

<li><a href="{% url 'about' %}">About\_us</a></li>

<li><a href="{% url 'index' %}">store</a></li>

<li><a href="{% url 'about' %}">press</a></li>

<li><a href="{% url 'about' %}">FAQs</a></li>

</ul>

</div>

</div>

<div class="col-md-3 col-sm-6">

<!--Column1-->

<div class="footer-pad">

<h4>Help</h4>

<ul class="list-unstyled">

<li><a href="#">payment</a></li>

<li><a href="#">Shipping</a></li>

<li><a href="{% url 'returnpage' %}">Cancellation & Return</a></li>

<li><a href="#">Privacy Policy</a></li>

<li><a href="#">FAQs</a></li>

<li><a href="#">Webmaster</a></li>

</ul>

</div>

</div>

<div class="col-md-3 col-sm-6">

<!--Column1-->

<div class="footer-pad">

<h4>Policy</h4>

<ul class="list-unstyled">

<li><a href="#">Parks and Recreation</a></li>

<li><a href="#">Public Works</a></li>

<li><a href="#">Police Department</a></li>

<li><a href="#">Fire</a></li>

<li><a href="#">Mayor and City Council</a></li>

<li>

<a href="#"></a>

</li>

</ul>

</div>

</div>

<div class="col-md-3">

<h4>social</h4>

<ul class="social-network social-circle">

<li><a href="#" class="icoFacebook" title="Facebook"><i class="fa fa- facebook"></i></a></li>

<li><a href="#" class="icoLinkedin" title="Linkedin"><i class="fa fa- linkedin"></i></a></li>

<li><a href="#" class="icotwitter" title="twitter"><i class="fa fa-twitter"></i></a></li>

</ul>

</div>

</div>

<div class="row">

<div class="col-md-12 copy">

<p class="text-center">&copy; Copyright 2018 - Company Name. All

rights reserved.</p>

</div>

</div

</div>

</footer>

</body></html>

{% block html%}

{% endblock %} VIEW PAGE CODING

from django.shortcuts import render,render\_to\_response,redirect,HttpResponseRedirect,reverse

from django.http import HttpResponse from .models import Info,Product

from django.contrib.auth.models import User,auth from django.contrib.auth import authenticate, login

from django.contrib.auth.forms import UserCreationForm from django.contrib import messages

from django.contrib.auth.decorators import login\_required

# Create your views here. def index(request):

product=Product.objects.get(Id=1) context={

'image':product.image,

}

return render(request,'dashboard/index.html',context) def contact(request):

if request.method=="GET":

name=request.GET.get('name' ,'') email= request.GET.get('email' ,'') phone= request.GET.get('phone' ,'') desc= request.GET.get('desc' ,'')

info=Info(name=name,email=email,phone=phone,desc=desc) info.save()

return render(request,'dashboard/contact.html')

def about(request):

return render\_to\_response('dashboard/about.html') def returnpage(request):

return render\_to\_response('dashboard/return.html')

def register(request):

if request.method == 'POST':

f = UserCreationForm(request.POST) if f.is\_valid():

f.save()

messages.success(request, 'Account created successfully') return redirect('/')

else:

f = UserCreationForm()

return render(request, 'dashboard/register.html', {'form': f})

def auth\_view(request):

username = request.POST.get('username', '') password = request.POST.get('password', '')

user = auth.authenticate(username = username, password = password)

if user is not None: auth.login(request, user)

return HttpResponseRedirect(reverse('index')) else:

return HttpResponseRedirect('/dashboard/invalid') def cart(request):

return render\_to\_response('dashboard/cart.html')

#### URLS PAGE DJANGO

from django.urls import path from dashboard import views from django.conf import settings

from django.conf.urls.static import static

# from django.contrib.auth import views as auth\_views urlpatterns = [

path('', views.index,name='index'), path('contact/', views.contact,name='contact'), path('about/', views.about,name='about'), path('register/', views.register,name='register'), path('login/',views.login,name='login'), path('cart/',views.cart,name='cart'),

path('returnpage/',views.returnpage,name='returnpage'),

]

#### MODELS

from django.db import models # Create your models here. class Info(models.Model):

msg\_id= models.AutoField(primary\_key=True) name = models.CharField(max\_length=50) email = models.CharField(max\_length=50) phone = models.CharField(max\_length=50) desc =models.TextField(max\_length=500)

def Str (self):

return self.name

class Product(models.Model):

Id= models.AutoField(primary\_key=True) name = models.CharField(max\_length=50)

price=models.DecimalField(max\_digits=10,decimal\_places=2) category=models.TextField(max\_length=30)

image = models.ImageField(upload\_to='images/') desc =models.TextField(max\_length=500)

def Str (self):

return self.name

#### CSS CODING

/\*FOOTER\*/S

footer {

background: #16222A;

background: -webkit-linear-gradient(59deg, #3A6073, #16222A); background: linear-gradient(59deg, #3A6073, #16222A);

color: white; margin-top:100px;

}

footer a { color: #fff;

font-size: 14px; transition-duration: 0.2s;

}

footer a:hover {

color: rgb(125, 245, 70);

text-decoration: none;

}

.copy {

font-size: 12px; padding: 10px;

border-top: 1px solid #FFFFFF;

}

.footer-middle { padding-top: 2em; color: white;

}

/\*SOCİAL İCONS\*/

/\* footer social icons \*/

ul.social-network { list-style: none; display: inline;

margin-left: 0 !important;

padding: 0;

}

ul.social-network li { display: inline; margin: 0 5px;

}

/\* footer social icons \*/

.social-network a.icoFacebook:hover { background-color: #3B5998;

}

.social-network a.icoLinkedin:hover { background-color: #007bb7;

}

.social-network a.icoFacebook:hover i,

.social-network a.icoLinkedin:hover i { color: #fff;

}

.social-network a.socialIcon:hover,

.socialHoverClass { color: #44BCDD;

}

.social-circle li a { display: inline-block; position: relative; margin: 0 auto 0 auto;

-moz-border-radius: 50%;

-webkit-border-radius: 50%;

border-radius: 50%; text-align: center; width: 30px; height: 30px;

font-size: 15px;

}

.social-circle li i { margin: 0;

line-height: 30px; text-align: center;

}

.social-circle li a:hover i,

.triggeredHover {

-moz-transform: rotate(360deg);

-webkit-transform: rotate(360deg);

-ms--transform: rotate(360deg); transform: rotate(360deg);

-webkit-transition: all 0.2s;

-moz-transition: all 0.2s;

1. ransition: all 0.2s;

-ms-transition: all 0.2s; transition: all 0.2s;

}

.social-circle i { color: #595959;

-webkit-transition: all 0.8s;

-moz-transition: all 0.8s; ransition: all 0.8s;

-ms-transition: all 0.8s; transition: all 0.8s;

}

.social-network a { background-color: #F9F9F9;

}

.im {

background-color: #007bb7;

}

/\* /////////////////////////// \*/ h2{

text-align:center; padding: 20px;

}

/\* Slider \*/

.slick-slide {

margin: 0px 20px;

}

.slick-slide img { width: 100%;

}

.slick-slider

{

position: relative; display: block;

box-sizing: border-box;

-webkit-user-select: none;

-moz-user-select: none;

-ms-user-select: none; user-select: none;

-webkit-touch-callout: none;

-khtml-user-select: none;

-ms-touch-action: pan-y; touch-action: pan-y;

-webkit-tap-highlight-color: transparent;

}

.slick-list

{

position: relative; display: block; overflow: hidden; margin: 0;

padding: 0;

}

.slick-list:focus

{

outline: none;

}

.slick-list.dragging

{

cursor: pointer; cursor: hand;

}

.slick-slider .slick-track,

.slick-slider .slick-list

{

-webkit-transform: translate3d(0, 0, 0);

-moz-transform: translate3d(0, 0, 0);

-ms-transform: translate3d(0, 0, 0);

-o-transform: translate3d(0, 0, 0);

transform: translate3d(0, 0, 0);

}

.slick-track

{

position: relative; top: 0;

left: 0; display: block;

}

.slick-track:before,

.slick-track:after

{

display: table; content: '';

}

.slick-track:after

{

clear: both;

}

.slick-loading .slick-track

{

visibility: hidden;

}

.slick-slide

{

display: none; float: left; height: 100%; min-height: 1px;

}

[dir='rtl'] .slick-slide

{

float: right;

}

.slick-slide img

{

display: block;

}

.slick-slide.slick-loading img

{

display: none;

}

.slick-slide.dragging img

{

pointer-events: none;

}

.slick-initialized .slick-slide

{

display: block;

}

.slick-loading .slick-slide

{

visibility: hidden;

}

.slick-vertical .slick-slide

{

display: block; height: auto;

border: 1px solid transparent;

}

.slick-arrow.slick-hidden { display: none;

}

SETTING USRLS

from django.contrib import admin from django.urls import path,include # from accounts import views

from django.conf import settings

from django.conf.urls.static import static

# from django.contrib.auth import views as auth\_views urlpatterns = [

path('admin/', admin.site.urls), path('', include('dashboard.urls')),

# path('login', 'django.contrib.auth.views.login', name='login')

]

urlpatterns= urlpatterns+static(settings.MEDIA\_URL,documents\_root=settings.MEDIA\_ROOT)

#### ADMIN PAGE

from django.contrib import admin

# Register your models here. from .models import Info,Product

admin.site.register(Info) admin.site.register(Product) **REGISTRATION PAGE**

from django.contrib.auth.forms import UserCreationForm from django.contrib.auth.models import User

from django import forms

from django.core.exceptions import ValidationError class CustomUserCreationForm(forms.Form):

username = forms.CharField(label='Enter Username', min\_length=4, max\_length=150) email = forms.EmailField(label='Enter email')

password1 = forms.CharField(label='Enter password', widget=forms.PasswordInput) password2 = forms.CharField(label='Confirm password', widget=forms.PasswordInput) def clean\_username(self):

username = self.cleaned\_data['username'].lower() r = User.objects.filter(username=username)

if r.count():

raise ValidationError("Username already exists") return username

def clean\_email(self):

email = self.cleaned\_data['email'].lower() r = User.objects.filter(email=email)

if r.count():

raise ValidationError("Email already exists") return email

def clean\_password2(self):

password1 = self.cleaned\_data.get('password1') password2 = self.cleaned\_data.get('password2')

if password1 and password2 and password1 != password2: raise ValidationError("Password don't match")

return password2

def save(self, commit=True):

user = User.objects.create\_user( self.cleaned\_data['username'], self.cleaned\_data['email'], self.cleaned\_data['password1']

)

return user **MANAGE.PY PAGE**

"""Django's command-line utility for administrative tasks.""" import os

import sys

def main():

os.environ.setdefault('DJANGO\_SETTINGS\_MODULE', 'website.settings') try:

from django.core.management import execute\_from\_command\_line except ImportError as exc:

raise ImportError(

"Couldn't import Django. Are you sure it's installed and " "available on your PYTHONPATH environment variable? Did you " "forget to activate a virtual environment?"

) from exc execute\_from\_command\_line(sys.argv)

if name == ' main ':

main() setting page """

Django settings for website project.

Generated by 'django-admin startproject' using Django 2.2.

For more information on this file, see https://docs.djangoproject.com/en/2.2/topics/settings/

For the full list of settings and their values, see https://docs.djangoproject.com/en/2.2/ref/settings/ """

import os

# Build paths inside the project like this: os.path.join(BASE\_DIR, ...) BASE\_DIR = os.path.dirname(os.path.dirname(os.path.abspath( file )))

# Quick-start development settings - unsuitable for production

# See https://docs.djangoproject.com/en/2.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret! SECRET\_KEY = 'mb69hy0s4wx(0=fh#\_pk0sxg\_t\_^+@m)931iw1&@7uf1p5w77^'

# SECURITY WARNING: don't run with debug turned on in production! DEBUG = True

ALLOWED\_HOSTS = []

# Application definition

INSTALLED\_APPS = [

'dashboard', 'django.contrib.admin', 'django.contrib.auth', 'django.contrib.contenttypes',

'django.contrib.sessions', 'django.contrib.messages', 'django.contrib.staticfiles',

]

MIDDLEWARE = [

'django.middleware.security.SecurityMiddleware', 'django.contrib.sessions.middleware.SessionMiddleware', 'django.middleware.common.CommonMiddleware', 'django.middleware.csrf.CsrfViewMiddleware', 'django.contrib.auth.middleware.AuthenticationMiddleware', 'django.contrib.messages.middleware.MessageMiddleware', 'django.middleware.clickjacking.XFrameOptionsMiddleware',

]

ROOT\_URLCONF = 'website.urls'

TEMPLATES = [

{

'BACKEND': 'django.template.backends.django.DjangoTemplates', 'DIRS': [],

'APP\_DIRS': True, 'OPTIONS': {

'context\_processors': [

'django.template.context\_processors.debug', 'django.template.context\_processors.request', 'django.contrib.auth.context\_processors.auth', 'django.contrib.messages.context\_processors.messages',

],

},

},

]

WSGI\_APPLICATION = 'website.wsgi.application' LOGIN\_REDIRECT\_URL = '/'

# Database

# https://docs.djangoproject.com/en/2.2/ref/settings/#databases

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.mysql', 'NAME': 'mysite',

'USER': 'root',

'PASSWORD': 'root',

'HOST': 'localhost',

'PORT': '3306',

}

}

# Password validation

# https://docs.djangoproject.com/en/2.2/ref/settings/#auth-password-validators

AUTH\_PASSWORD\_VALIDATORS = [

{

'NAME': 'django.contrib.auth.password\_validation.UserAttributeSimilarityValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.MinimumLengthValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.CommonPasswordValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.NumericPasswordValidator',

},

]

# Internationalization

# https://docs.djangoproject.com/en/2.2/topics/i18n/

LANGUAGE\_CODE = 'en-us'

TIME\_ZONE = 'UTC'

USE\_I18N = True

USE\_L10N = True

USE\_TZ = True

# Static files (CSS, JavaScript, Images)

# https://docs.djangoproject.com/en/2.2/howto/static-files/

STATIC\_URL = '/static/' # STATICFILES\_DIRS = [

# os.path.join(BASE\_DIR, "dashboard/static"),

# ]

# STATIC\_ROOT = os.path.join(BASE\_DIR,'assets') MEDIA\_URL = '/media/'

MEDIA\_ROOT = os.path.join(BASE\_DIR, 'media')

#### CODE EFFICIENCY

We can use to examine the run-time behavior of your programs. BY using profiler information , we can determation , we can determine which sections of your code are working efficiently . The profiler can produce information showing areas of code that are not being executed or that are taking a long time execute.

Because profiling is a tuning process , we should use the profiler to make your programs run better, not to find bugs . Once your program is fairly stable . you should start profiling to see where your code could perform better.

Use the profiler to determine whether,

* + An algorithm is effective (timing ),
  + A function is being called too many
  + A piece of code is being covered by software testing procedures
  + Function profiling is good detecting inefficient code . function profiling is faster than line profiling because there is less information to collect
  + Line profiling can be useful for checking the validity of an algorithm because it shoes how many times each line is executed in response to certain input data. Line profiling also lets you see

which aren’t executed at all.

**Code optimization**

Optimization programs

By writing your code carefully , you can write the fastest possible programs . There are several ways to improve program performance.

Following the

* + Gereral programming performance hints.
  + Using name expressions instead of macro substitution
  + Referencing object properties efficiently Gerneral Programming Performance hints

To write the fastest program possible, follow the recommendation listed Below

Choose the correct data type for your data . in particular. Use the interger data type for numeric information whenever possible . as it is processes most efficiently . Whenever possible

. Use integer data types for primary and foreign key values , which will result in smaller data files, smaller (and therefore faster) indexes, and faster joins.

Avoid reopening files , which slows performance. Instead , assign files to work areas as you open the . and them user then the SELECT command to choode a specofoc work area a needed.

Use FOR ….END OR loops rather than Do WHILE…ENDDO loops when possible. Because they are faster.

To use memory most efficiently . avoid creationg object before you need them and clear object you finish with them to free memory

Using name expression instead of macro substitution

If you are use name expressions instead of macro substitution , program

Performance will greatly improve. For example, if you assign a value to the variable file , a name expression created with files is faster than macro substitution .

Referencing object properties efficiently

By understanding how python Django properties and objects , you can make your application Run more efficiently.

#### Validation Techniques

Validators detect problems in your web page and style sheet. It could be a tag that was opened and never closed. It could be a misspelled piece of code or forgotten element the tag or style requires to work properly. You become a detective, hunting and solving the little problems occurring in your web page. The resources and articles on validation below will help you learn more about validating your web page.

Validating your WordPress site means more than just checking the front page for errors. With the modular Themes and template files in WordPress, while you may fix all the errors associated with the index.php and sidebar.php when viewing your front page, errors may still lie within any other template files such as single.php, page.php, archives.php, or category.php. Validate these page views as well to make sure you cover all the template files.

Validation errors aren't limited to your template files. They can also happen inside of a post. When you are writing a post and using HTML, WordPress might not recognize the HTML and convert it to a character entity, or you may have entered it wrong. If you have a page with a lot of HTML coding, validate it to make sure you have it all correct. And occasionally check random posts to make sure everything is still okay from time to time as part of your general housekeeping.

Validation doesn't just mean putting your pages through some web driven testers. It also means test-driving it with friends, relatives, co-workers, and strangers. Everyone has a different system and way of working, so ask for others to test-drive your styles or themes before you make them public.

Validation Checklist

To help you validate your WordPress site, here is a quick checklist:

Validate HTML/XHTML Validate CSS

Validate for Section 508 Standards (accessibility) Validate for WAI standards (accessibility) Validate Links (check for dead links)

Validate Feeds

Check across different browsers (include handheld computers, Mac, PC, and cellphones, too) Re-validate HTML and CSS

Have friends, relatives, co-workers check your site

When ready, you can post your site on the WordPress Forum's Your WordPress for review

These two terminologies are also very important in software testing. So, what they denote, let's dig deep into it. Verification focuses on the concern that whether: "Developers are building the system right?". It ensures whether the system is meeting all the functionality as per requirement. The verification process takes place in the 1st position which includes documentation check, coding, etc.

On the other hand, validation deals with whether: "Developers are building the right software?". This ensures whether all functionalities are properly behaving or not. The validation process gets performed after the verification and largely engages in the inspection of your overall software.

#### Website test

A good E-commerce/Retail site is key to its success. It must be a worthy counterpart to the storefront. Because, when you go shopping at a physical store, the customer has already made a commitment to visit and might give the brand a chance.

Online, choices are many. So, unless there is engagement from the beginning, the user might just leave.

The better the site, the better the business.

Since so much lays on the application, it is critical that it undergoes thorough testing.

E-commerce application/sites are web applications or mobile application too. So, they undergo all the typical test types.

Functional Testing Usability Testing Security Testing Performance Testing Database Testing

Mobile Application Testing A/B testing.

For a quick look at most often performed tests on a typical web application, check out:

=> 180+ Sample Test Cases for Testing Web and Desktop Applications

However, Retail sites are highly dynamic in nature. There are new offers, new products, new bestsellers, Sales, etc. This means the site doesn’t stay the same for too long. Therefore, it could get overwhelming for many.

E-Commerce Testing Checklist

Below, we have listed important segments and test cases for eCommerce website testing.

#1) Homepage – Hero Image:

**The following are a few things to test:**

* Is it going to auto scroll?
* If yes, at what interval will the image be refreshed?
* When the user hovers over it, is it still going to scroll to the next one?
* Can it be hovered on?
* Can it be clicked on?
* If yes, is it taking you to the right page and right deal?
* Is it loading along with the rest of the page or loads last in comparison to the other elements on the page?
* Can the rest of the content be viewed?
* Does it render the same way in different browsers and different screen resolutions?

#2) Search:

Search algorithms are very important for the success of a retail site because we can’t always place what the users want to see right in front of their eyes.

Common tests are:

Search based on Product name, brand name or something more broadly, the category. For example Camera, Canon EOS 700D, electronics, etc.

Search Results have to be relevant

Different sort options have to be available- based on Brand, Price, and Reviews/ratings etc.

How many results to display per page?

For multi-page results, are there options to navigate to them

Also, search happens in many places. Please take the search drilling down into multiple levels into consideration when validating this functionality. For example: When I search on the home page, I might see something like this:

#3) Product Details Page:

Once a user finds a product either through search or by browsing or by clicking on it from the homepage, the user will be taken to the product information page.

Check:

Image or images of the product Price of the product

Product specifications Reviews

Check out options Delivery options Shipping information In stock/Out of stock

Multiple color or variations options

Breadcrumb navigation for the categories (highlighted in Red below). If navigation such as that is displayed, make sure every element of it is functional.

This is the penultimate stage before the user commits to the purchase.

Test the following:

Add items to the cart and continue shopping

If the user adds the same item to the cart while continuing to shop, the item count in the shopping cart should get incremented

All items and their totals should be displayed in the cart Taxes as per location should be applied

A user can add more items to the cart- total should reflect the same Update the contents added to the cart- total should reflect that too Remove items from the cart

Proceed to checkout

Calculate Shipping costs with different shipping options Apply coupons

Don’t check out, close the site and come back later. The site should retain the items in the cart

#5) Payments:

Check different payment options

If allowing check out as Guest, simply finish the purchase and provide an option to register at the end

Returning customers – Login to check out User sign up

If storing customer Credit card or any other financial information, perform security testing around this to make sure it is secure.(PCI compliance is a must)

If the user is signed up for a long time, make sure the session is timed out or not. Every site has a different threshold. For some, it is 10 minutes. For some, it might be different.

Emails/Text confirmation with the order number generated

#6) Categories/Featured Products/Related or Recommended products

The most popular FAQ I get from E-commerce testers is: Do I have to test every category/every product?

#7) After-Order tests Check:

Change the Order Cancel the Order Track the Order Returns

#8) Other tests:

Login FAQs

Contact Us page

Customer Service page etc.

Challenges Automating E-commerce Website

To remain on Safer Edge and deliver the desired results to the client you need to shift the focus on quality and performance of your E-commerce website while shrinking timeline as much as possible

In general Automation Testing starts by selecting right test automation framework which directly impacts on the result of the test automation project. The framework must include the test scripts and the scenarios of various automated processes.

Based on the framework, the testers can easily execute the tests and obtain relevant results by generating test reports. But selecting right tool to automate E-commerce Website depends on many key parameters. It is always important to compare the available tools based on key parameters like features, performance, extensibility, licensing cost, maintenance cost, and Training and support.

You must take advantage of many open source test automation tools to automate more testing efforts without investing additional funds.

#1) E-commerce websites are much entangled in nature, automating each action is not possible because we cannot assume the nature of the customer.

#2) Continuous changes for e-commerce demands Regression so run regression test suit every day to keep track the effects of change.

#3) Always go with Automating Integration type of scenarios that should cover from selecting a link on home page till checkout and payment gateway page. Hereby, you can at least cover maximum user experience with E-commerce Website, so that adequate testing can be achieved by automating regression cycle.

#4) Never waste time automating on the unstable application. A simple change will affect your whole test suits and you have to recreate it.

#5) Homepage of E-commerce Website is very important and contents many information and 1000 of links associated with each product and these links grow up every day as new offers or product is added to a page. So before proceeding to regression testing its best to verify every link in page by using HTTP status code.

#6) When you are executing test scripts on a different browser at the same time. If a product is added to shopping cart or removed that information should be reflected in other browsers too.

#7) When you running test parallel this will obviously fail your script in such scenario you have to periodically refresh your page to retain cart information. In real time you may come across this scenario such as a user may sometimes use mobile e-commerce app and also mobile e- commerce web application.

#8) Don’t neglect to verify each product details and pricing details whether it is 10 products or 1000 products it should be as per the seller requirement. This is the phase where you can make or break a customer slight mistake will lead to a big loss.

#9) Create yourself a lot of interrupted scenarios that usually user come across design your script very robust so that your script afford it and still run and pass the script.

For Example, you stored all the card information and clicked on submit due to low charge or network issue application stuck. In this case, a user is notified about his transaction status through email and message to phone you should validate this email or message in a test script.

#10) Web element of E-commerce website keeps changing so always Create manual xpath. Some Web Elements attributes will be same so there will be no unique way of distinguishing in such scenario use contains() method of xpaths or scroll into view.

#11) Automate Accessibility Testing by keyboard actions without using mouse action you definitely will come across some of the problems and fix it. This plays a significant role in user interface testing.

#12) Tester should be carefully designed the scenario and add initiate checkpoint and insert login script whenever it is required.

#13) Maintain different scripts for a different mode of payment to avoid confusion. Check if what happens if an order is canceling after payment.

#14) Performance testing in other hand plays a very crucial role. The factors you need to test here request per second, Transaction Per minute, Execution per click, a Response time of page load, duration of the task, Length of time between click and page display and DNS lookup.

#15) Security Testing is where customer trust is gained on which e-commerce is built so here you have to spend a lot of time testing on DENIAL OF SERVICE ATTACK, User Account security, Data confidentiality, content security, credit card security, disable non-essential services.SSL Certificate Validation.

#16) Automating Localization testing is very challenging in e-commerce because of Compliance with accessibility standards to support multi-lingual markets and business regions.

**The Web Site Design and Development Process**

There are numerous steps in the web site design and development process. From gathering initial information, to the creation of your web site, and finally to maintenance to keep your web site up to date and current.

The exact process will vary slightly from designer to designer, but the basics are the same. Information Gathering Planning

Design Development

Testing and Delivery Maintenance

1. Information Gathering

The first step in designing a successful web site is to gather information. Many things need to be taken into consideration when the look and feel of your site is created.

This first step is actually the most important one, as it involves a solid understanding of the company it is created for. It involves a good understanding of you – what your business goals and dreams are, and how the web can be utilized to help you achieve those goals.

1. Planning

S

Using the information gathered from phase one, it is time to put together a plan for your web site. This is the point where a site map is developed.

The site map is a list of all main topic areas of the site, as well as sub-topics, if applicable. This serves as a guide as to what content will be on the site, and is essential to developing a consistent, easy to understand navigational system. The end-user of the web site – aka your customer – must be kept in mind when designing your site. These are, after all, the people who will be learning about your service or buying your product. A good user interface creates an easy to navigate web site, and is the basis for this.

During the planning phase, your web designer will also help you decide what technologies should be implemented. Elements such as what CMS (content management system) such as WordPress to incorporate, will any contact forms be needed, etc. are discussed when planning your web site.

1. Design

Drawing from the information gathered up to this point, it’s time to determine the look and feel of your site.

Target audience is one of the key factors taken into consideration. A site aimed at teenagers, for example, will look much different than one meant for a financial institution. As part of the design phase, it is also important to incorporate elements such as the company logo or colors to help strengthen the identity of your company on the web site.

1. Development

The developmental stage is the point where the web site itself is created. At this time, your web designer will take all of the individual graphic elements from the prototype and use them to create the actual, functional site

This is typically done by first developing the home page, followed by a “shell” for the interior pages. The shell serves as a template for the content pages of your site, as it contains the main navigational structure for the web site. Once the shell has been created, your designer will take your content and distribute it throughout the site, in the appropriate areas.

1. Testing and Delivery

At this point, your web designer will attend to the final details and test your web site. They will test things such as the complete functionality of forms or other scripts, as well last testing for last minute compatibility issues (viewing differences between different web browsers), ensuring that your web site is optimized to be viewed properly in the most recent browser versions.

A good web designer is one who is well versed in current standards for web site design and development. The basic technologies currently used are HTML andCSS (Cascading Style Sheets). As part of testing, your designer should check to be sure that all of the code written for your web site validates. Valid code means that your site meets the current web development standards – this is helpful when checking for issues such as cross-browser compatibility as mentioned above.

***Website Maintenance***

Website maintenance is the act of regularly checking your website for issues and mistakes and keeping it updated and relevant. This should be done on a consistent basis in order to keep your website healthy, encourage continued traffic growth, and strengthen your SEO and Google rankings.

Keeping a website well maintained and attractive is important to companies big and small in order to engage and retain customers. It’s easy for businesses, especially startups, to cut corners and let a few tasks slide. Website maintenance can easily become one of those things as it doesn’t always present immediate issues. However, just like your health can fall apart if you go too long without a regular check up, so can the health of your website.

**To be done weekly**

1. Check that all of your pages are loading without errors
2. Run a backup and make sure a previous version of your site is stored
3. Make updates to website software and plugins
4. Check that all of your forms are running properly
5. Remove any spam comments from pages and posts
6. Check your pages to see if there are any broken links
7. Search for 404 errors and fix or redirect
8. Write one or more blog posts to keep your community engaged and encourage SEO traffic.

**To be done monthly**

1. Check the load speed of your website and ensure that nothing is bogging it down
2. Review your security scans and make sure nothing is out of place
3. Analyze website statistics from the previous month
4. Check your blog to see if there are any articles that could be updated

**To be done quarterly**

1. Review your website design and structure – can be it improved?
2. Check graphics and images – should anything be updated?
3. Review SEO and meta titles and descriptions to ensure they are as effective as possible
4. Test and tweak popups, forms, and calls to action
5. Review your workload for efficiencies to see if anything can be automated
6. Test your website on all devices and browsers to see if it displays correctly
7. Review advertising and marketing campaigns to see if anything needs to be changed or updated.
8. Restore a previous version of the website to check your backup health

**To be done yearly**

1. Update any reference to the current year
2. Review each page for content accuracy, grammar, typos, and relevancy
3. Check any active email addresses and see if any are excessive and can be deleted
4. Ensure that your website domain name is renewed
5. Consider whether a website design update is due
6. Review all of your top performing blog articles and see if they can be updated with new content

**System security measures taken**

E-commerce websites are often a target for fraudsters, hackers, and relevant competitors. This means you have to enhance the security of your website to protect personal information, ID details, product prices, and catalog and financial information of your clients. These details are some of the factors that make e-commerce websites very attractive to criminals, as well as competitors who would wish to exploit your business.

While building the best website security may be a costly affair, downtime in the event of a security breach is more expensive. For this reason, you can always enhance the security of your e-commerce website through backup plans such as Ottomatik MySQL backup. Following are the other security essentials that you need to explore.

* 1. SSL and PCI Compliance

It is imperative to encrypt information in your web browser. This is crucial because clients and web visitors send their data online. The information could be passed through multiple systems before reaching the final server. With such a chain, the data could be compromised if it’s not encrypted with a secure sockets layer (SSL) certificate. SSL also promotes your business because it prompts clients that their financial and credit card information is well protected.

* 1. Real Team Bot Detection Technology

Generating quality and the right traffic to your website helps you to increase your return on investment. Even so, you ought to be careful to attract only the most legitimate visitors to your page. Research by Distil Networks Bad Bot Report 2018 https://resources.distilnetworks.com/press-releases/annual-2018-bad-bot-report shows bad bots for e-commerce website frauds and 50% bots for overall website traffic. This means that what may appear as good traffic flow to your website could be malicious bots from fraudsters and competitors.

* 1. Static Application Security Testing

It is also important that you carry out a static application security testing (SAST) on your website. It simply means assessing the security of your e-commerce site by checking up on all associated databases, applications, and servers. Through SAST, you will be able to analyze different applications inside out. The National Institute of Standards and Technology Security Configuration Checklist Repository will help you to learn how you can complete a secure configuration for your website.

* 1. Choose a Secured E-Commerce Platform

When choosing an e-commerce platform, ensure it is well secured. A good platform includes an object-oriented encoding semantic based on the inbuilt security protocol. For example, when you settle for Magento or WordPress as your secured platform, you need the same security plugin for extra protection. Remember, e-commerce platforms do not have a bot mitigation strategy which is crucial for e-commerce businesses.

6. Dynamic Application Security Testing

Dynamic application security testing (DAST) tools are also important for enhanced security of any e-commerce website. The tools are advanced and they automate security tests for possible security threats. They test HTML

and HTTP interfaces and identify risks in the applications. However, you need the best DAST solutions for the best results.

***cost estimate***

A cost estimate is the approximation of the cost of a program, project, or operation. The cost estimate is the product of the cost estimating process. The cost estimate has a single total value and may have identifiable component values. A problem with a cost overrun can be avoided with a credible, reliable, and accurate cost estimate. A cost estimator is the professional who prepares cost estimates. There are different types of cost estimators, whose title may be preceded by a modifier, such as building estimator, or electrical estimator, or chief estimator. Other professionals such as quantity surveyors and cost engineers may also prepare cost estimates or contribute to cost estimates.

Ecommerce website cost by type

To finalize all of the abovementioned and ive the idea of all added costs to set up and run an ecommerce site, we’ll divide them in 3 categories and make a brief summary chart below. All three categories are based on custom development and design

Small web store

A small beginner ecommerce site as determined by three aspects: small product catalogue (100-1000), small traffic, basic design. Estimated cost of custom development: $15,000 – $20,000.

Mid-size ecommerce site

Web stores with average product catalogue, medium traffic (thousands of monthly visitors), advanced design and development of “pro”

features. Estimated cost of custom development: starting with $30,000

– $50000, and more. Large web store

An enterprise-scale ecommerce website with huge product catalogue, high traffic, professional unique web design and integration of various systems. With top-level images, videos, multiple payment methods, shipping options, order tracking, refund processing and other advanced features the cost may soar up to millions, eventually. Estimated cost of custom development: $55,000 – $160,000.

Mind, that these estimations cover only the initial costs to set up an online store, and do not include either further annual fees to maintain a site or additional expenses to promote it, such as content, marketing, branding.

SmallMedium Enterprise

Domain / SSL / Setup $1,000 $2,000 – $4,000 $5,000 – $10,000 Hosting (annual)$3,000 $8,000 $12,000

Database $500 $1,000 $4,000

Development $5,000 – $10,000 $10,000 – $20,000 $20,000 –

$100,000

Design $3,000 $5,000 – $10,000$10,000 – $30,000

SEO (initial) $2,000 $4,000 $5,000

Total (setup) $14,500 – $19,500 $30,000 – $47,000 $56,000 –

$161,000

July1 july15 August1 August10 August18 August23 August28

deploye

present

Integrity

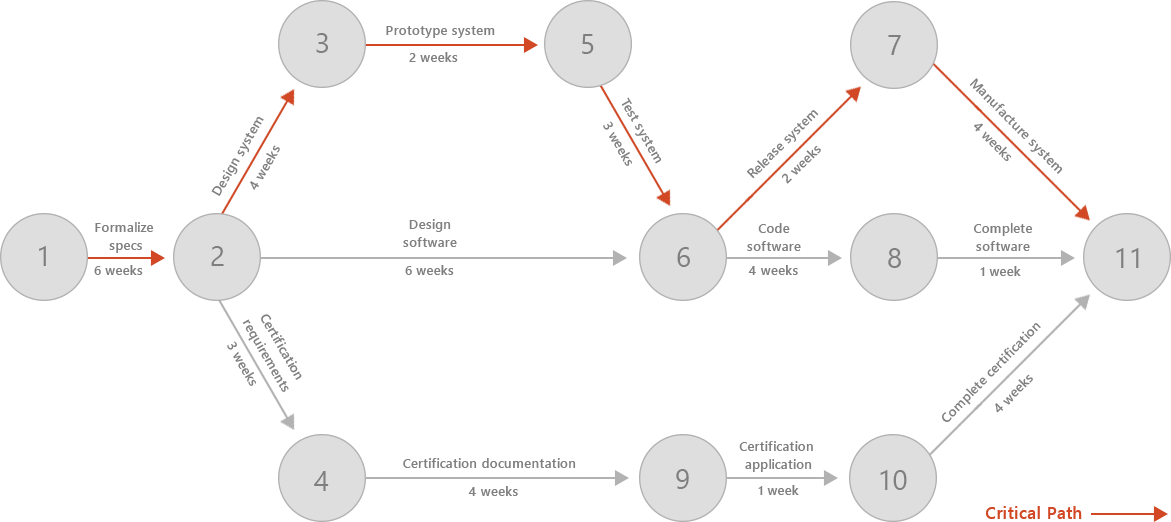
Unit test

code

Design

plan

### A Gantt chart is a type of bar chart that illustrates a project schedule, named after its inventor, Henry Gantt (1861–1919), who designed such a chart around the years 1910–1915.[1][2] Modern Gantt charts also show the dependency relationships between activities and current schedule status.



PERT Chart

PERT charts are used after a project has been planned and broken into smaller pieces, called tasks. They show how much time is needed to complete each task and they help project managers visualize which tasks must be completed before others can begin.

PERTs also illustrate what tasks are critical (critical path) and which ones are less important.

#### Future scope and further enhancement of the project:

Today, the market place is flooded with several e-commerce options for shoppers to choose from. A variety of innovative products and services are being offered spoiling customers for choice. Online shopping is no more a privilege enjoyed by your friends and family living in the US or UK. Today, it is a reality in India. In the last couple of years, the growth of e-commerce industry in India has been phenomenal as more shoppers have started discovering the benefits of using this platform. There is enough scope for online businesses in the future if they understand the Indian shoppers psyche and cater to their needs.

**BIBLIOGRAPHY**

To make this project I have taken source from the following books;- 1 python and django

I have taken source form internet too and taken help of our it teacher and my brother I have also taken the help form ignou’s teacher