**PROJECT REPORT ON**

**GAMESTORE**

**Submitted to**



**Developed By**

**Parth Pandya**

**Dwarkesh Gohel**

**Himanshu Dabhi**

**MCA SEM – II 2021-22**

**Guided By**

Prof. Dhruvi Sidhpura

**Affiliated By:**



**NARANDAS JETHALAL SONECHA MANAGEMENT & TECHNICAL INSTITUTE**

**Approved by AICTE Affiliated to GTU**

At. Chanduvav, Tal. Veraval, Dist. Gir Somnath. (Gujarat)

**Certificate**

This is to certify that  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

A student of MCA Semester \_\_\_\_\_\_\_\_\_  Enrollment No.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Has successfully carried out a project entitled  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ towards fulfillment of the subject

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ .

Date :

Signature of Guide Seal of the Institute Signature of HOD

**🡺PREFACE**

This report has been prepared for the “E-Commerce website”Project in which we developed the system for providing products to the customer on purchase and giving the reminder to the customer using preferable technology Python framework Django.

Design of the software and controls is currently at the concept design level. This project during study is a really precious one. It provides an opportunity to meet and work with different Tools with very good guidance of our professor. It helps to know their views and gain something from their attitudes and performances.

The project has developed, and is implementing, formal systems engineering processes for documenting requirements, configuration control, interface control, product assurance, integration test , verification and validation.

It was an advantage for us to work in a reputed organization like GUJARATTECHNOLOGICAL UNIVERSITY under such good guidance. This has given me an opportunity to work in a truly professional environment where teamwork scores over individual efforts.

**🡺ACKNOWLEDGEMENT**

I would like to thank my Internal Faculty Guide “Prof. Dhruvi Sidhpura (M.C.A.) NJSMTI, Chanduvav”for his valuable guidance and continual encouragement throughout the major project. I am heartily thankful to him for his time to time suggestions and the clarity of the concepts of the topic that helped me a lot during this project.

I am also thankful to my “Campus Director Sir K. Dwivedi (NJSMTI, Chanduvav)”for his continual kind words of encouragement and motivation throughout the Major Project.

I am thankful to all my faculty members of Department ``M.C.A. NJSMTI, Chanduvav”for their special attention and suggestions towards the project work.

The blessings of God and my Family members make the way for completion of the Major Project. I am very much grateful to them.

The friends, who always bear, motivate and share their project experience with me throughout this course, I am thankful to them.

**INDEX**

| **Sr.No** | **Title** | **Page No** |
| --- | --- | --- |
| 1 | Existing system | 05 |
| 2 | Need for the new system | 06 |
| 3 | Objective of the new system | 07 |
| 4 | Problem definition | 08 |
| 5 | Core components | 09 |
| 6 | Project profile | 15 |
| 7 | Assumption and Constraints | 15 |
| 8 | Advantages and Limitations of a Proposed System | 17 |
| 9 | Requirement and Determinations | 18 |
| 10 | Targeted Users | 20 |
| 11 | Use case diagram | 21 |
| 12 | Class diagram | 22 |
| 13 | Interaction diagram | 23 |
| 14 | Activity Diagram | 24 |
| 15 | Data dictionary | 27 |
| 16 | Development | 23 |
| 16 | Coding | 28 |
| 17 | Agile Documentations | 51 |
| 18 | Proposed Enhancements | 56 |
| 19 | Conclusion | 57 |
| 20 | Bibliography | 58 |

*INTRODUCTION*

🡺1.1 EXISTING SYSTEM

You can get several benefits with the **learning objectives of e-commerce**, like cost reduction, growth, gaining a popular brand image, and much more. Some of the most popular and primary objectives of E-Commerce are here:

### Cost-effective Management

The primary aim of every business is to focus on cost reduction, and E-Commerce helps them do the same effectively. The automatic process of e-commerce helps to reduce the cost of management. The proper use of technology, Digital Marketing, can help you gain more customers without investing much.

### Developing Business Relations

When businesses use e-commerce as a primary technique, they can easily grab achievement. When there is direct communication between the company and customers, the relationship boosts up. Eventually, companies can connect with more customers and result in increased growth.

### Sustaining Existing Customers

Customers are the core element of any business whether it is  b2b or b2c, and making a stable relationship with your existing customers is significant. E-commerce helps you sustain your customers for a longer period. When you continuously interact with your customers on a network on which they usually spend time, you will build a strong connection with customers. Social media networks are the most used platforms, and with these, you can grab your customers’ attention. With E-commerce, you can establish a brand name using social media networks.

### E-commerce increases Loyal Customers

According to [Google](https://www.google.co.in/), almost 60% of people prefer searching for things that they want to buy or opt for services on the internet. They check out the top 3 search results for their query. And, if you do not appear in that top 3 list, you might lose customers. So, E-commerce will help you to establish a strong brand presence so that you can capture new customers.

### Enhances the Efficiency of Services

By opting for the online E-commerce platform, you can boost up your efficiency. Opting for E-commerce not only increases your sales but also helps as a cost-effective method. With E-commerce, you can reduce your managing and warehousing costs. It eventually helps you save more funds at your disposal. You can also reduce delivery time with E-commerce and make your customers happy.

🡺1.2 NEED FOR THE NEW SYSTEM

#### (i) Computer System:

The presence of the computer system is the first requirement of e-commerce; because the basis of e-commerce is the Internet and the computer is the medium of transaction. The computer can be linked with the Internet, by pressing its keys. The business transaction under e-commerce can be seen on the computer screen.

#### (ii) Internet Connection:

For conducting e-commerce transactions, Internet connection is essential. We can get this facility of Internet connection, sitting at home, through Private Service Provider companies.

When we open our Internet connection, the service provider tells us to install the Web Browser. It controls the activities of e-commerce. It is the Web Browser which takes us into the world of e-commerce. (Browser means a programmer that lets us look at or read documents on the Internet).

#### (iii) Technically Qualified Workforce:

The business enterprise must have technically qualified people who are capable of working with computer networks and the Internet without difficulty.

#### (iv) System of Receiving Payments:

The business enterprise must have a full-proof system of receiving payment for the goods sold. The business enterprise must make arrangements with banks, credit card agencies etc. to facilitate receipts and payments electronically.

#### (v) Well-Designed Website:

To communicate effectively with customers and others, the business enterprise must develop a comprehensive website. The information must be detailed and hyperlinked with suitable supporting pictures etc. (Hyperlink is a place in an electronic document on a computer that is linked to another electronic document).

#### (vi) Effective Telecommunication System:

For a successful implementation of e-commerce, an effective telecommunication system is necessary. If telephone lines are getting frequently disconnected; e-commerce is not successful.

🡺1.3 OBJECTIVE OF THE NEW SYSTEM

1. INTRODUCTION

This section includes the overall view of the project i.e. the basic problem definition and the general overview of the problem which describes the problem in layman terms. It also specifies the software used and the proposed solution strategy.

2. SOFTWARE REQUIREMENT SPECIFICATION

This section includes the Software and hardware requirements for the smooth running of the application.

3. DESIGN & PLANNING

This section consists of the Software Development Life Cycle model. It also contains technical diagrams like the Data Flow Diagram and the Entity Relationship diagram.

4. IMPLEMENTATION DETAILS

This section describes the different technologies used for the entire development process of the Front-end as well as the Back-end development of the application.

5. RESULT AND DISCUSSION

This section has screenshots of all the implementations i.e. user interface and their description.

6. SUMMARY AND CONCLUSION

This section has screenshots of all the implementations i.e. user interface and their description.

🡺1.4 PROBLEM DEFINITION

E-commerce provides an easy way to sell products to a large customer base. However, there is a lot of competition among multiple e-commerce sites. When users land on an e-commerce site, they expect to find what they are looking for quickly and easily. Also, users are not sure about the brands or the actual products they want to purchase. They have a very broad idea about what they want to buy. Many customers nowadays search for their products on Google rather than visiting specific e-commerce sites. They believe that Google will take them to the e-commerce sites that have their product.

The purpose of any e-commerce website is to help customers narrow down their broad ideas and enable them to finalize the products they want to purchase. For example, suppose a customer is interested in purchasing a mobile. His or her search for a mobile should list mobile brands, operating systems on mobiles, screen size of mobiles, and all other features as facets. As the customer selects more and more features or options from the facets provided, the search narrows down to a small list of mobiles that suit his or her choice. If the list is small enough and the customer likes one of the mobiles listed, he or she will make the purchase.

The challenge is also that each category will have a different set of facets to be displayed. For example, searching for books should display their format, as in paperback or hardcover, author name, book series, language, and other facets related to books. These facets were different for mobiles that we discussed earlier. Similarly, each category will have different facets and it needs to be designed properly so that customers can narrow down to their preferred products, irrespective of the category they are looking into.

The takeaway from this is that categorization and feature listing of products should be taken care of. Misrepresentation of features can lead to incorrect search results. Another takeaway is that we need to provide multiple facets in the search results. For example, while displaying the list of all mobiles, we need to provide facets for a brand. Once a brand is selected, another set of facets for operating systems, network, and mobile phone features has to be provided. As more and more facets are selected, we still need to show facets within the remaining products.

🡺1.5 CORE COMPONENTS

#### User interface app components

This is a reference to the web pages that have a role that is related to the display, settings and configurations. It is related to the interface/experience, rather than the development, and consequently it deals with display dashboards, configuration settings, notifications, and logs etc.

#### Structural components

The structural components of a web application basically refer to the functionality of the web application with which a user interacts, the control and the database storage.

In other words, it has got more to do with the structural aspects of the architecture, as the name suggests. This basically comprises (1) The web browser or client, (2) The web application server and

(3) The database server.

The web browser or client permits the users to interact with the functions of the web apps and is generally developed using HTML, CSS, and JavaScript.

#### HTML:-

* + - * HTML stands for Hyper Text Markup Language that is used to develop web pages.
      * **Hypertext** refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a webpage is called Hypertext.
      * As its name suggests, HTML is a **Markup Language** which means you use HTML to simply "markup" a text document with tags that tell a Web browser how to structure it to display.
      * HTML is not a programming Language like C, C++ and JAVA etc…
      * It is a cross platform markup language that is designed to be flexible enough to display text and other elements like graphical on a variety of views.
      * The HTML document consists of special tags that are embedded in an ASCH document.
      * Web browsers like Internet Explorer, Netscape Navigator etc... Interprets these tags.

#### 

#### CSS:-

If Cascading Style Sheets can be a bit intimidating at first glance, especially if you are new to webpage coding It might be best to avoid getting into Style until you have done a few pages and are comfortable handling basic html. Style sheet coding offers:

* + - * An impressive array of features not available in standard HTML4. The use of style permits you to really fine tune the appearance of your pages. And, it is not difficult to learn.
      * In addition to offering many options not available in standard html, style can save you time, increase accuracy, and improve uniformity of appearance throughout your web pages. Style Sheets, whether External, Internal or Inline, give you expanded design capabilities with flexibility. You can combine the best features of HTML4 with those of Style, and produce a better webpage.
      * For me, the biggest plus factor for style sheets is the time saved in not having to constantly type formatting statements for headlines and paragraphs. The time savings is not only in the keystrokes that are eliminated, but in the extreme accuracy obtained I can't make types in all that stuff I don't type and then have to go back and make corrections.
      * This is what my basic style sheet for this website looks like.
      * In effect, that one-line link statement tells the browser to go get the style.css file and make use of it during the execution of this .html file you are now viewing.

#### CSS3 :-

* + - * If Cascading Style Sheets can be a bit intimidating at first glance, especially if you are new to webpage coding. It might be best to avoid getting into Style until you have done a few pages and are comfortable handling basic html. Style sheet coding offers an impressive array of features not available in standard HTML4. The use of style permits you to really fine tune the 7appearance of your pages. And, it is not difficult to learn.
      * In addition to offering many options not available in standard html, style can save you time, increase accuracy, and improve uniformity of appearance throughout your web pages. Style Sheets, whether External, Internal or Inline, give you expanded design capabilities with

flexibility. You can combine the best features of HTML4 with those of Style, and produce a better webpage.

* + - * For me, the biggest plus factor for style sheets is the time saved in not having to constantly type formatting statements for headlines and paragraphs. The time savings is not only in the keystrokes that are eliminated, but in the extreme accuracy obtained I can't make types in all that stuff I don't type and then have to go back and make corrections.

flexibility. You can combine the best features of HTML4 with those of Style, and produce a better webpage.

* + - * This is what my basic style sheet for this website looks like.
      * In effect, that one-line link statement tells the browser to go get the style.css file and make use of it during the execution of this .html file you are now viewing.

#### Python :-

* Python is a simple, general purpose, high level, and object-oriented programming language.
* Python is an interpreted scripting language also. *Guido Van Rossum* is known as the founder of Python programming.
* Our Python tutorial includes all topics of Python Programming such as installation, control statements, [Strings](https://www.javatpoint.com/python-strings), [Lists](https://www.javatpoint.com/python-lists), [Tuples](https://www.javatpoint.com/python-tuples), [Dictionary](https://www.javatpoint.com/python-dictionary), [Modules](https://www.javatpoint.com/python-modules), [Exceptions](https://www.javatpoint.com/python-exception-handling), Date and Time, File I/O, Programs, etc. There are also given Python interview questions to help you better understand Python Programming.
* **Python** is a general purpose, dynamic, [high-level](https://www.javatpoint.com/classification-of-programming-languages), and interpreted programming language. It supports an Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures.
* Python is *easy to learn* yet powerful and versatile scripting language, which makes it attractive for Application Development.
* Python's syntax and *dynamic typing* with its interpreted nature make it an ideal language for scripting and rapid application development.
* Python supports *multiple programming patterns*, including object-oriented, imperative, and functional or procedural programming styles.
* Python is not intended to work in a particular area, such as web programming. That is why it is known as *multipurpose* programming language because it can be used with web, enterprise, 3D CAD, etc.
* We don't need to use data types to declare variable because it is *dynamically typed* so we can write a=10 to assign an integer value in an integer variable.
* Python makes the development and debugging *fast* because there is no compilation step included in Python development, and edit-test-debug cycle is very fast.

#### SQLite :-

* SQLite is an embedded relational database management system. It is self-contained, serverless, zero configuration and transactional SQL database engine.
* Our SQLite Tutorial includes all topics of SQLite such as SQLite with history, features, advantages, installation, commands, syntax, data types, operators, expressions, databases, table, crud operations, clauses, like, glob, limit, and clause, advance SQLite etc.
* SQLite is an embedded relational database management system. It is self-contained, serverless, zero configuration and transactional SQL database engine.
* SQLite is an embedded relational database management system. It is self-contained, serverless, zero configuration and transactional SQL database engine.
* SQLite is free to use for any purpose commercial or private. In other words, "SQLite is an open source, zero-configuration, self-contained, stand alone, transaction relational database engine designed to be embedded into an application".
* SQLite is different from other SQL databases because unlike most other SQL databases, SQLite does not have a separate server process. It reads and writes directly to ordinary disk files. A complete SQL database with multiple tables, indices, triggers, and views, is contained in a single disk file.

**Django (Framework) :-**

Django is a Web Application Framework which is used to develop web applications.

Our Django Tutorial includes all topics of Django such as introduction, features, installation, environment setup, admin interface, cookie, form validation, Model, Template Engine, Migration, MVT etc. All the topics are explained in detail so that the reader can get enough knowledge of Django.

Django is a web application framework written in Python programming language. It is based on the MVT (Model View Template) design pattern. Django is very demanding due to its rapid development feature. It takes less time to build an application after collecting client requirements.

This framework uses a famous tagline:**The web framework for perfectionists with deadlines.**

By using Django, we can build web applications in very less time. Django is designed in such a manner that it handles much of the configuration automatically, so we can focus on application development only.

🡺1.6 PROJECT PROFILE

| Project Profile | * Gamestore Website |
| --- | --- |
| Project Category | * Web Development |
| Front End | * Python (Django) |
| Back End | * SQLite |
| Browser | * Google Chrome, Mozilla Firefox, Microsoft Edge, Opera |
| Operating System | * Windows |
| Documentation | * Google Docs |
| Developed By | * Dwarkesh Gohel * Himanshu Dabhi * Parth Pandya |
| Project Guide | * Prof. Chirag Rachchh and Prof. Dhruvi Sidhpura |
| Submitted To | * Narandas Jethalal Sonecha Management and Technical Institute |
| University | * Gujarat Technological University (GTU) |

*REQUIREMENT DETERMINATION AND ANALYSIS*

🡺2.1 REQUIREMENT DETERMINATION

***WHAT IS REQUIREMENT DETERMINATION?***

A Requirement is a vital feature of a new system which may include processing or capturing of data, controlling the activities of business, producing information and supporting the management.

Requirements determination involves studying the existing system and gathering details to find out what are the requirements, how it works, and where improvements should be made

***Major Activities in Requirement Determination***

***Requirement Anticipation***

* It predicts the characteristics of a system based on previous experience which include certain problems or features and requirements for a new system.
* It can lead to analysis of areas that would otherwise go unnoticed by inexperienced analysts. But if shortcuts are taken and bias is introduced in conducting the investigation, then requirement anticipation can be half baked.

***Requirement Investigation***

* It is studying the current system and documenting its features for further analysis
* It is at the heart of system analysis where analysts document and describe system features using fact-finding techniques, prototyping, and computer assisted tools.

***Requirement Specifications***

* It includes the analysis of data which determine the requirement specification, description of features for the new system, and specifying what information requirements will be provided.
* It includes analysis of factual data, identification of essential requirements, and selection of requirement- fulfillment strategies

***TECHNICAL FEASIBILITY***

* It investigates the technical feasibility of each implementation alternative.
* It analyzes and determines whether the solution can be supported by existing technology or not.
* The analyst determines whether current technical resources be upgraded or added it that fulfill the new requirements.
* It ensures that the candidate system provides appropriate responses to what extent it can support the technical enhancement

***OPERATIONAL FEASIBILITY***

* It determines whether the system is operating effectively once it is developed and implemented.
* It ensures that the management should support the proposed system and its working feasible in the current organizational environment
* It analyzes whether the users will be affected and they accept the modified or new business methods that affect the possible system benefits
* It also ensures that the computer resources and network architecture of candidate system are workable.

***ECONOMIC FEASIBILITY***

* It is evaluating the effectiveness of candidate system by using cost/benefit analysis method.
* It demonstrates the net benefits from the candidate system in term of benefits and costs to the organization.
* The main aim of economic feasibility analysis is to estimate the economic requirements of candidate system before investments funds are committed to proposal
* It prefers the alternative which will maximize the net worth of organization by earliest and highest return of funds along with lowest level of risk involved in developing the candidate system.

***BEHAVIORAL FEASIBILITY***

* It evaluates and estimates the user attitude or behavior towards the development of new system.
* It helps in determining if the system requires special effort to educate, retrain, transfer, and changes in employees job status on new ways of conducting business

***SCHEDULE FEASIBILITY***

* It ensures that the project should be completed within given time constraint or schedule.
* It also verifies and validates whether the deadlines of project are reasonable or not.

2.2 TARGETED USERS

An ecommerce target market is a group of people or institutions that are most likely to share the same shopping interests and might be willing to buy a company’s products and services.

As these folks are the most potential prospects for an ecommerce store that’s dealing in its respective niche, it makes more sense for that ecommerce store to focus on these groups in terms of marketing and keeping them hooked with their online store.

***Segments of Targeted User***

* College Students
* Bachelors
* Families
* Travelers
* Business Travelers
* Tourists

*SYSTEM DESIGN*

3.1 USE CASE DIAGRAM





3.2 INTERACTION DIAGRAM





3.4 ACTIVITY DIAGRAM (LOGIN)





🡺SDLC MODEL















3.5 DATA DICTIONARY

***Product Table:-***

| ***FieldType*** | ***DataType*** | ***Size*** |
| --- | --- | --- |
| name | CharField | 200 |
| image | imagefield | - |
| mini\_1 | imagefield | - |
| mini\_2 | imagefield | - |
| mini\_3 | imagefield | - |
| mini\_4 | imagefield | - |
| price | integerfield | 20 |
| description | textfield | 500 |

***Cart Table:-***

| ***FieldType*** | ***DataType*** | ***Size*** |
| --- | --- | --- |
| user | USER(FK) | - |
| product | Product(FK) | - |
| quantity | integrfield | - |
| total | integerfield | - |
| timestamp | datetimefield | - |
| active | boolean | - |

***Orders Table:-***

| ***FieldType*** | ***DataType*** | ***Size*** |
| --- | --- | --- |
| user | USER(FK) | - |
| cart | Cart(FK) | - |
| total | integerfield | - |
| firstname | charfield | 100 |
| lastname | charfield | 100 |
| email | charfield | 100 |
| phone | integerfield | - |
| city | charfield | 100 |
| state | charfield | 100 |
| address | charfield | 500 |
| payment | charfield | 50 |
| status | charfield | 50 |
| time | datetimefield | - |

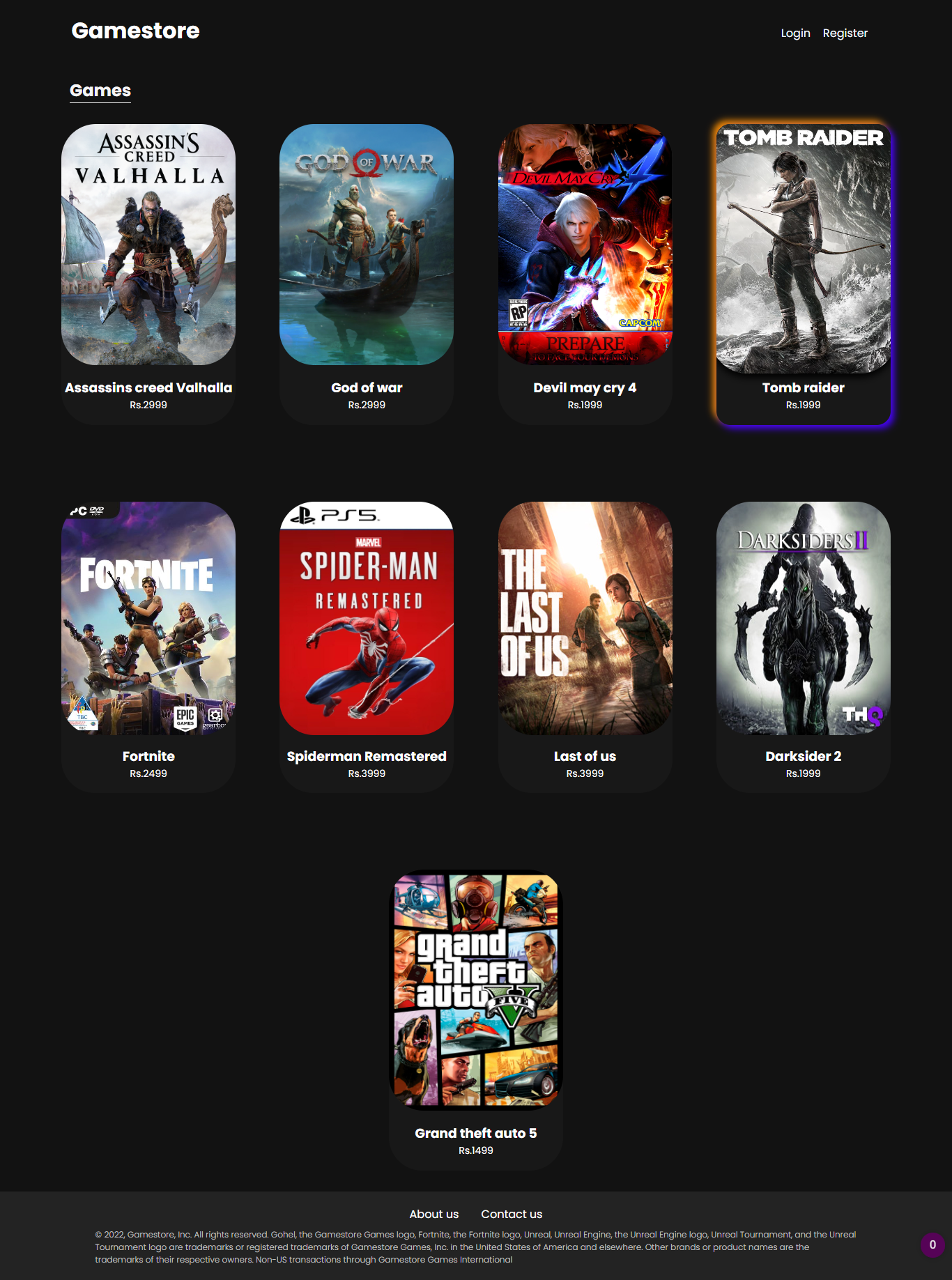
***Contactus Table:-***

| ***FieldType*** | ***DataType*** | ***Size*** |
| --- | --- | --- |
| name | charfield | 50 |
| email | charfield | 50 |
| phone | integerfield | - |
| message | charfield | 50 |
|  |  |  |
|  |  |  |

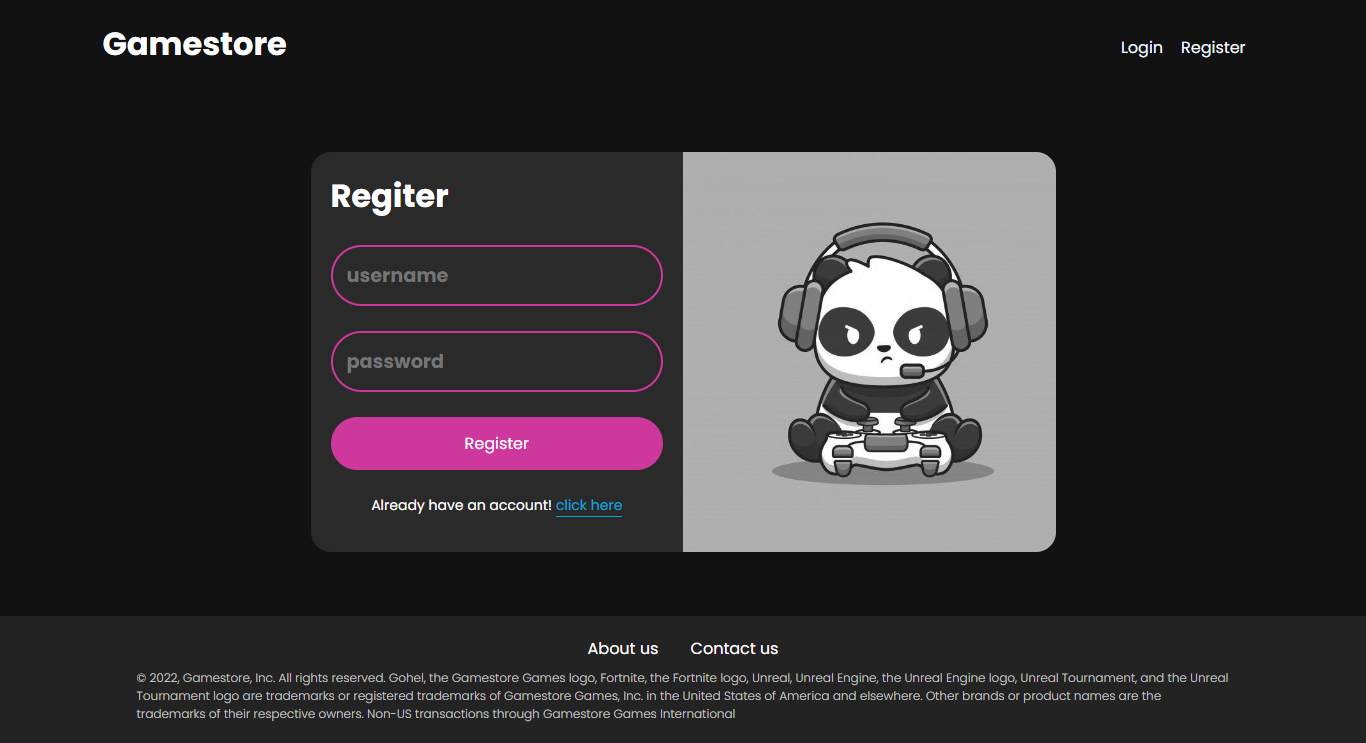
*DEVELOPMENT*

4.0 SCREENSHOTS

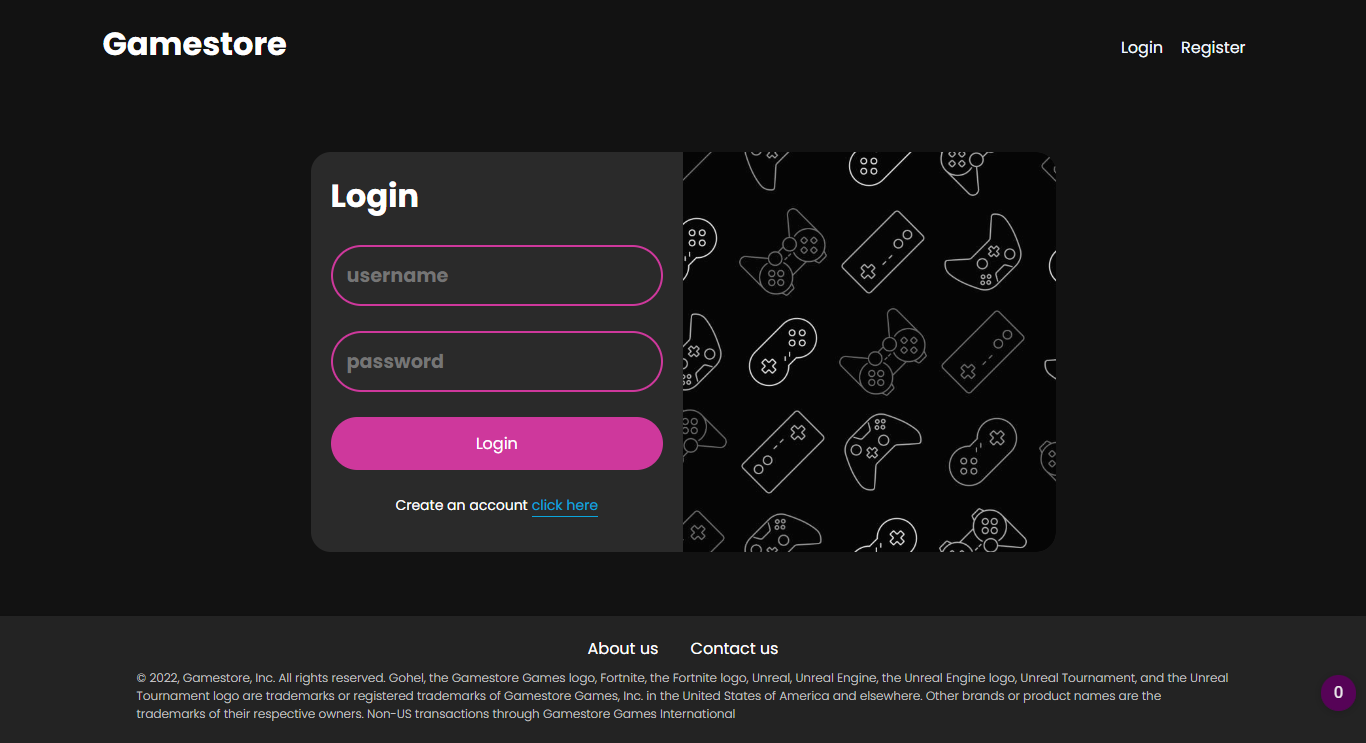
Hompage



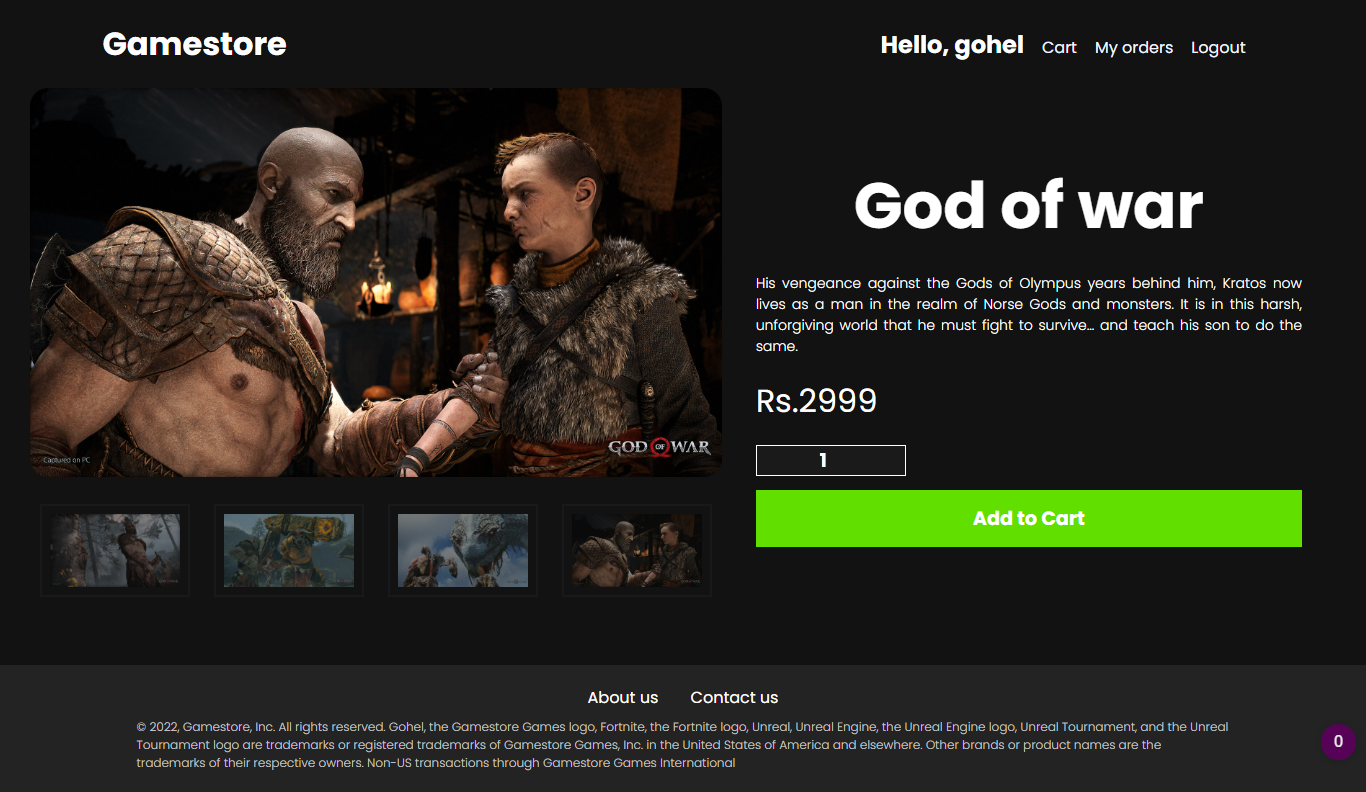
Register page

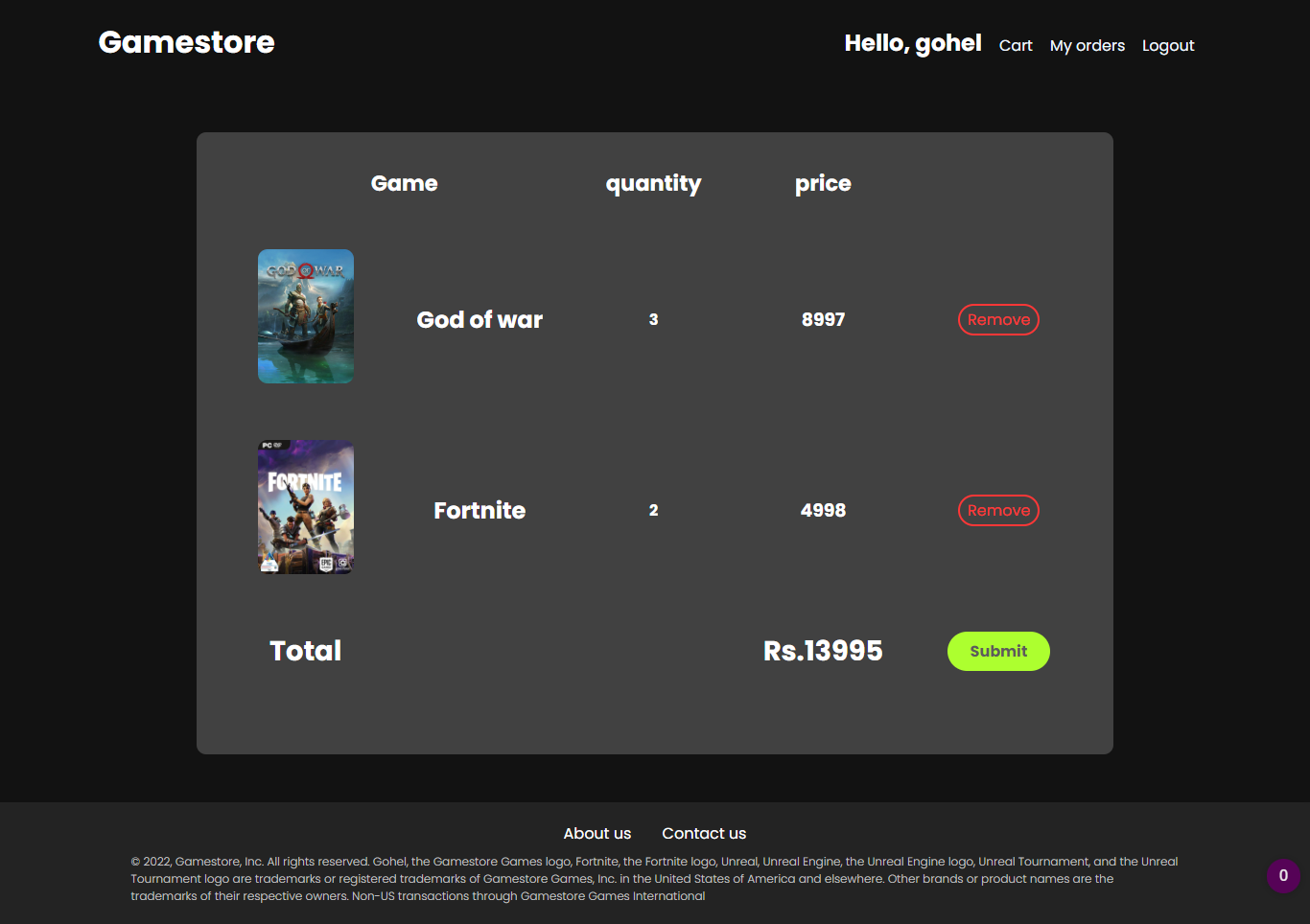


Login Page

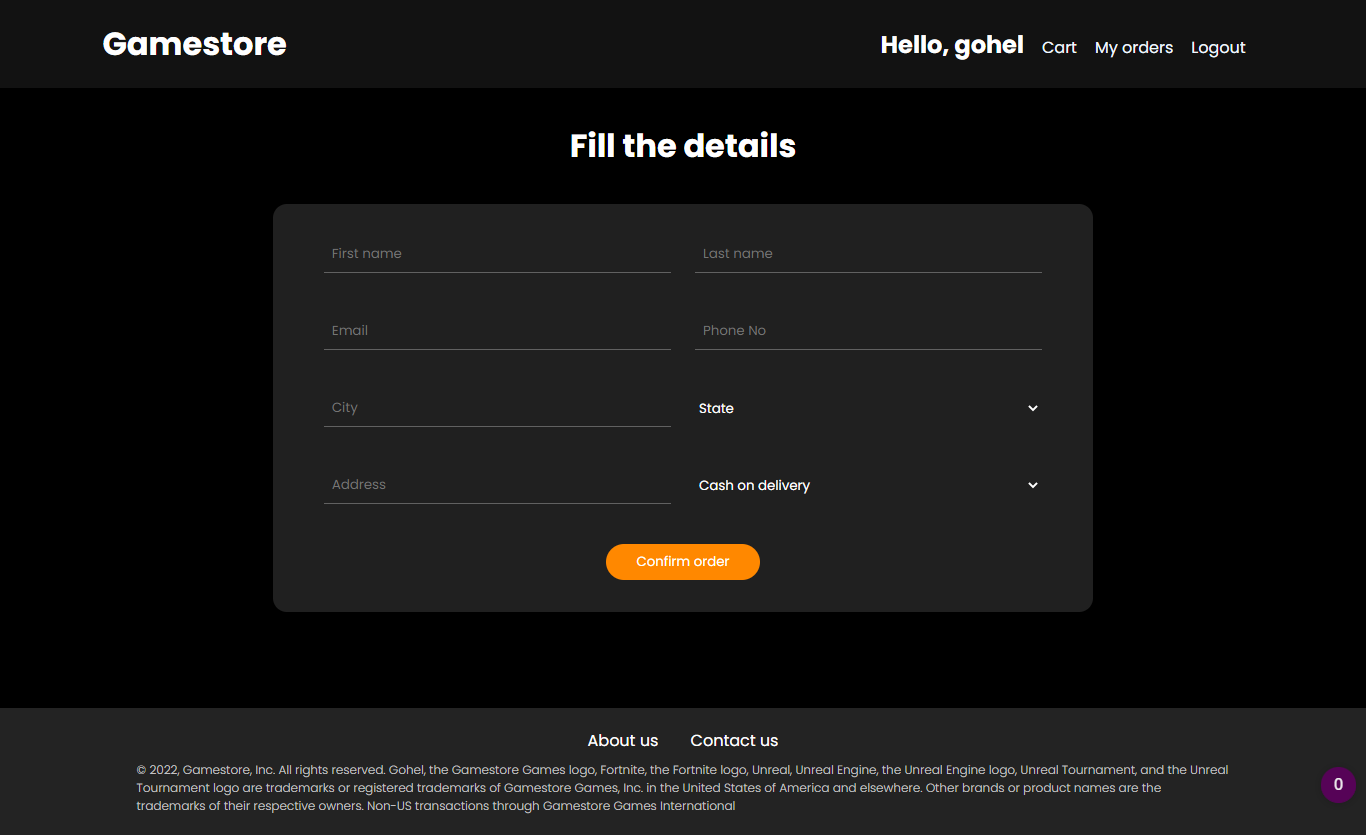


Gameview Page

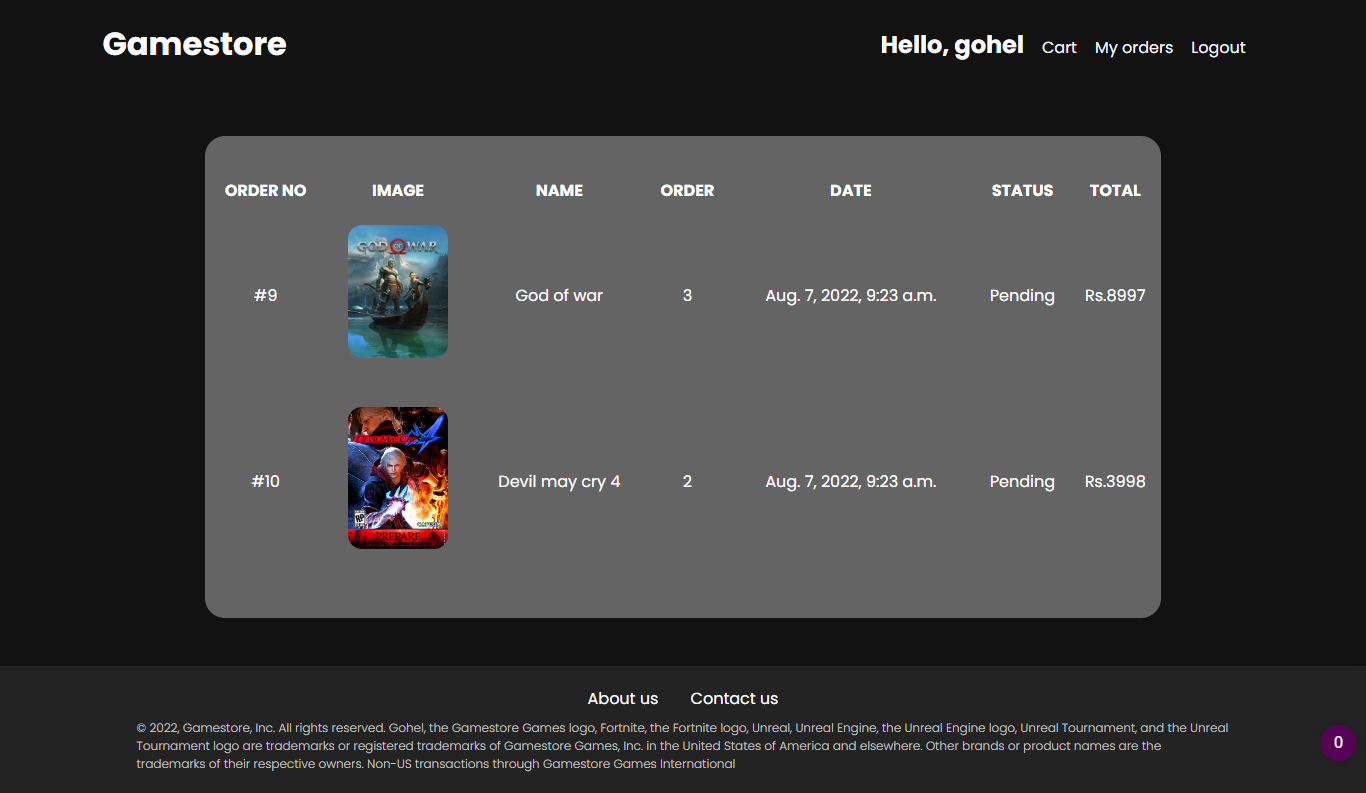


cart page

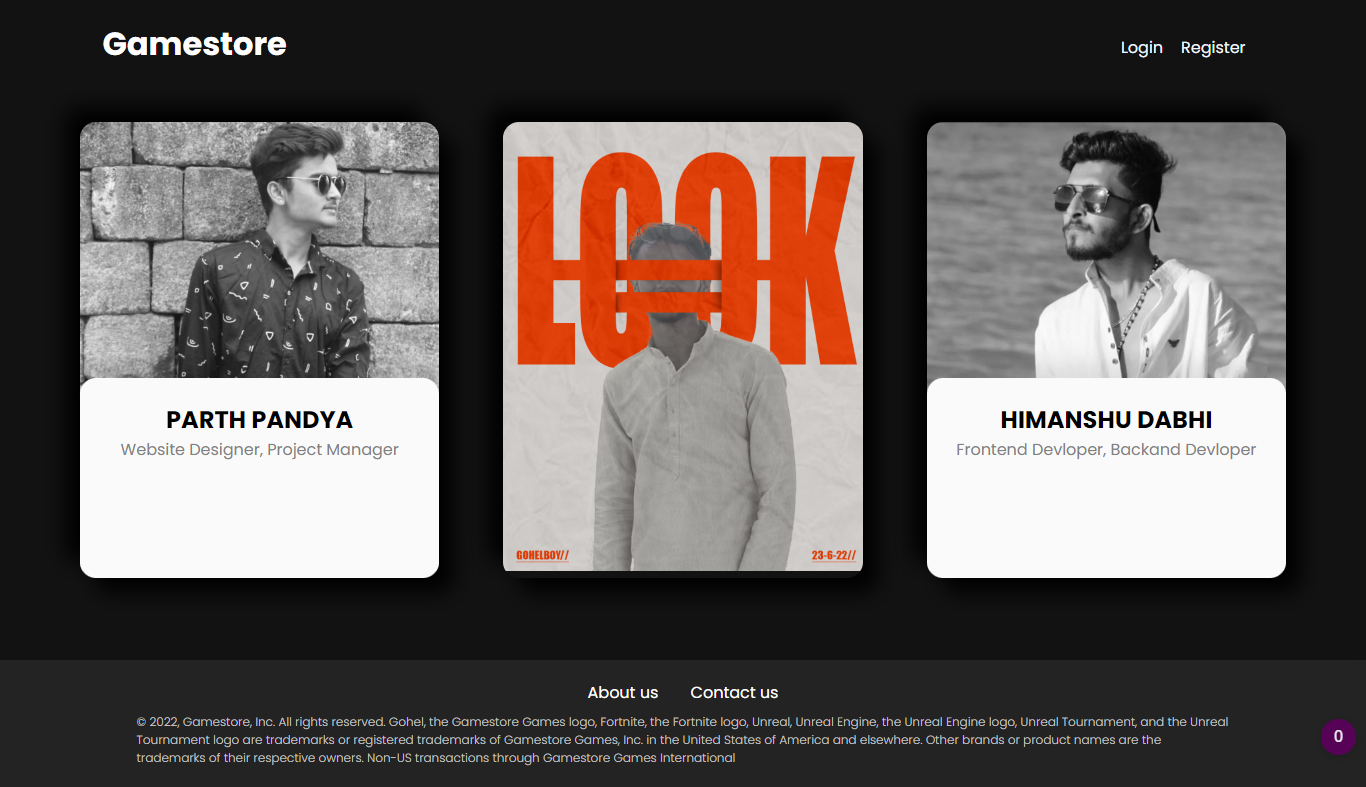
checkout page



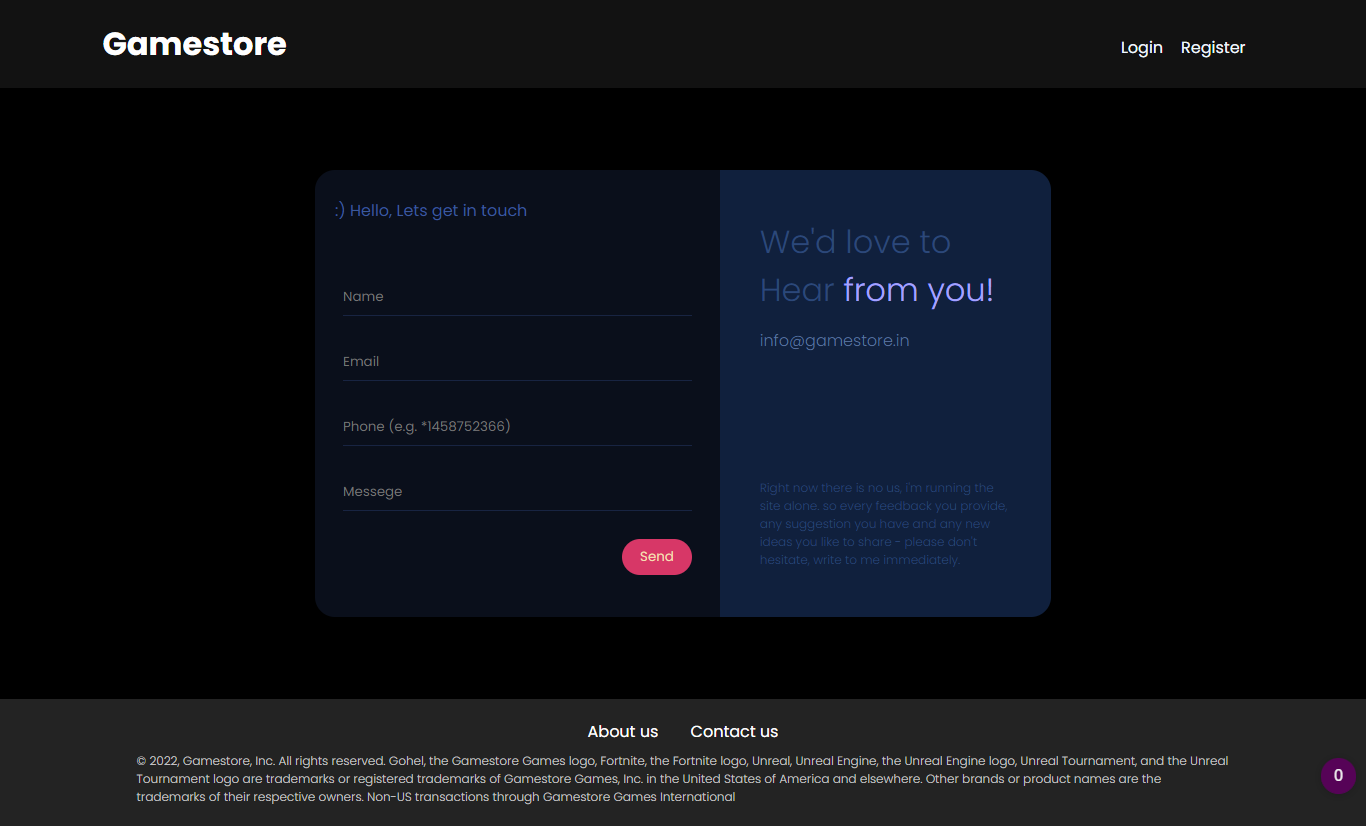
my orders page



about us page



contact us page



4.1 CODING STANDARD

Base.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

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

<link rel="icon" type="image/x-icon" href="/static/images/favicon.png" />

<link rel="stylesheet" href="/static/css/base.css" type="text/css" />

<link rel="stylesheet" href="/static/css/home.css" type="text/css" />

<link rel="stylesheet" href="/static/css/game-detail-view.css" type="text/css" />

<link rel="stylesheet" href="/static/css/cart.css" type="text/css" />

<link rel="stylesheet" type="text/css" href="/static/css/about-us.css" />

<link rel="stylesheet" type="text/css" href="/static/css/contact-us.css" />

<link rel="stylesheet" type="text/css" href="/static/css/checkout.css" />

<title>Gamestore</title>

</head>

<body>

<div class="navbar-container">

<div class="navbar">

<div class="logo">

<a href={% url 'home' %}>

<h1>Gamestore</h1>

</a>

</div>

<div class="nav-link">

<ul>

{% if user.is\_authenticated %}

<li>

<h4>Hello, {{user.username}}</h4>

</li>

<li>

<a href="{% url 'cart' %}">Cart</a>

</li>

<li>

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

</li>

<li>

<a href="{% url 'logout' %}">Logout</a>

</li>

{%else%}

<li>

<a href="{% url 'login' %}">Login</a>

</li>

<li>

<a href="{% url 'register' %}">Register</a>

</li>

{%endif%}

</ul>

</div>

</div>

</div>

{% block content%}

{% endblock content %}

<div class="footer">

<div class="footer-content">

<div class="footer-link">

<a href="{% url 'about-us' %}">About us</a>

<a href="{% url 'contact-us' %}">Contact us</a>

</div>

<p>

© 2022, Gamestore, Inc. All rights reserved. Gohel, the Gamestore Games logo, Fortnite, the Fortnite

logo,

Unreal, Unreal Engine, the Unreal Engine logo, Unreal Tournament, and the Unreal Tournament logo are

trademarks or

registered trademarks of Gamestore Games, Inc. in the United States of America and elsewhere. Other

brands

or

product names

are the trademarks of their respective owners. Non-US transactions through Gamestore Games International

</p>

</div>

</div>

</body>

</html>

Index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

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

{% extends 'base.html'%}

<link rel="stylesheet" href="/static/css/home.css" media="screen" type="text/css" />

<title>Document</title>

</head>

<body>

{%block content%}

<div class="game-list">

<h1>Games</h1>

<div class="container">

{% for i in products %}

<div class="card">

<div class="img">

<a href='/{{i.pk}}/'><img src="{{i.image.url}}"></a>

</div>

<div class="card-detail">

<h3>{{i.name}}</h3>

<p>Rs.{{i.price}}</p>

</div>

</div>

{% endfor %}

</div>

</div>

{%endblock content%}

</body>

</html>

login.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

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

<link rel="stylesheet" type="text/css" href="/static/css/forms.css" />

{% extends 'base.html' %}

<title>Document</title>

</head>

<body>

{% block content %}

<div class="form-container">

<div class="form-box">

<div class="form">

<h1>Login</h1><br>

<form method="POST">

{% csrf\_token %}

<input type="text" class="if" name="username" placeholder="username" required /><br><br>

<input type="password" class="if" name="password" placeholder="password" required /><br><br>

<button class="if-btn" type="submit">Login</button>

</form><br>

<div class="gotolink">

Create an account <a href="{% url 'register' %}"> click here</a>

</div>

</form>

</div>

<div class="img">

<img src="/static/images/loginbg.jpg" />

</div>

</div>

</div>

{% endblock content %}

</body>

</html>

register.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

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

<link rel="stylesheet" type="text/css" href="/static/css/forms.css" />

{% extends 'base.html' %}

<title>Document</title>

</head>

<body>

{% block content %}

<div class="form-container">

<div class="form-box">

<div class="form">

<h1>Regiter</h1><br>

<form method="POST">

{% csrf\_token %}

<input type="text" class="if" name="username" placeholder="username" required /><br><br>

<input type="password" class="if" name="password" placeholder="password" required /><br><br>

<button class="if-btn" type="submit">Register</button>

</form><br>

<div class="gotolink">

Already have an account! <a href="{% url 'login' %}"> click here</a>

</div>

</form>

</div>

<div class="img">

<img src="/static/images/registerbg.jpg" />

</div>

</div>

</div>

{% endblock content %}

</body>

</html>

game-detail-view.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

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

{% extends 'base.html'%}

<link rel="stylesheet" href="/static/css/game-detail-view.css" type="text/css" />

<title>Document</title>

</head>

<body>

{% block content %}

<div class="game-detail-view-container">

<div class="images-container">

<div class="video-box">

<img src="{{game.mini\_1.url}}" id="main-img" onclick="change(*this*)" />

</div>

<div class="mini-images">

<img src="{{game.mini\_1.url}}" onclick="change(*this*)" />

<img src="{{game.mini\_2.url}}" onclick="change(*this*)" />

<img src="{{game.mini\_3.url}}" onclick="change(*this*)" />

<img src="{{game.mini\_4.url}}" onclick="change(*this*)" />

</div>

</div>

<div class="game-info-container">

<div class="mini-container">

<div class="game-title">

<h1>{{game.name}}</h1>

</div>

<div class="game-desc">

<p>{{game.description}}</p>

</div>

<div class="game-price">

<h3>Rs.{{game.price}}</h3>

</div>

<div class="btns">

<form action="/add-to-cart/" method="POST">

{% csrf\_token %}

<input type="text" name="game-id" value="{{game.id}}" hidden />

<input type="number" class="quantity-count" name="quantity" value="1" min="1" />

<input type="submit" class="addtocart" value="Add to Cart" />

</form>

</div>

</div>

</div>

</div>

<script>

function change(mini) {

var fullimg = document.getElementById("main-img");

fullimg.src = mini.src;

}

</script>

{% endblock %}

</body>

</html>

cart.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

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

{%extends 'base.html'%}

<link rel="stylesheet" href="/static/css/cart.css" type="text/css" />

<title>Document</title>

</head>

<body>

{% block content %}

<div class="main-cart-container">

<table class="cart">

<tr class="th">

<th colspan="2">Game</th>

<th>quantity</th>

<th>price</th>

</tr>

{% for i in items %}

<tr class="items">

<th>

<img src="{{i.product.image.url}}" width="100px" height="140px" />

</th>

<th>

<h2>{{i.product.name}}</h2>

</th>

<th>

<h4>{{i.quantity}}</h4>

</th>

<th>

<h3>{{i.total}}</h3>

</th>

<th>

<form action="/remove-from-cart/" method="post">

{%csrf\_token%}

<input type="text" name="cart-id" value="{{i.id}}" hidden />

<input type="submit" value="Remove">

</form>

</th>

</tr>

{% endfor %}

<tr class="total-section">

<th>

<h2>Total</h2>

</th>

<td></td>

<td></td>

<th>

<h2>Rs.{{total}}</h2>

</th>

<th>

{%if total > 0%}

<form action="/checkout/" method="post">

{% csrf\_token %}

<input type="number" name="total" value="{{total}}" hidden />

<input type="submit" class="checkout" />

</form>

{%endif%}

</th>

</tr>

</table>

</div>

{% endblock content %}

</body>

</html>

checkout.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

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

{%extends 'base.html'%}

<link rel="stylesheet" type="text/css" href="/static/css/checkout.css" />

<title>checkout</title>

</head>

<body>

{% block content %}

<div class="checkout-container">

<h1>Fill the details</h1>

<div class="box">

<form action="/order/" method="post">

{%csrf\_token%}

<input type="text" name="first-name" placeholder="First name" required />

<input type="text" name="last-name" placeholder="Last name" required />

<input type="email" name="emaial" placeholder="Email" required />

<input type="number" name="phone" placeholder="Phone No" required />

<input type="text" name="city" placeholder="City" required />

<select name="state" id="state" class="form-control">

<option value="none">State</option>

<option value="Andhra Pradesh">Andhra Pradesh</option>

<option value="Andaman and Nicobar Islands">Andaman and Nicobar Islands</option>

<option value="Arunachal Pradesh">Arunachal Pradesh</option>

<option value="Assam">Assam</option>

<option value="Bihar">Bihar</option>

<option value="Chandigarh">Chandigarh</option>

<option value="Chhattisgarh">Chhattisgarh</option>

<option value="Dadar and Nagar Haveli">Dadar and Nagar Haveli</option>

<option value="Daman and Diu">Daman and Diu</option>

<option value="Delhi">Delhi</option>

<option value="Lakshadweep">Lakshadweep</option>

<option value="Puducherry">Puducherry</option>

<option value="Goa">Goa</option>

<option value="Gujarat">Gujarat</option>

<option value="Haryana">Haryana</option>

<option value="Himachal Pradesh">Himachal Pradesh</option>

<option value="Jammu and Kashmir">Jammu and Kashmir</option>

<option value="Jharkhand">Jharkhand</option>

<option value="Karnataka">Karnataka</option>

<option value="Kerala">Kerala</option>

<option value="Madhya Pradesh">Madhya Pradesh</option>

<option value="Maharashtra">Maharashtra</option>

<option value="Manipur">Manipur</option>

<option value="Meghalaya">Meghalaya</option>

<option value="Mizoram">Mizoram</option>

<option value="Nagaland">Nagaland</option>

<option value="Odisha">Odisha</option>

<option value="Punjab">Punjab</option>

<option value="Rajasthan">Rajasthan</option>

<option value="Sikkim">Sikkim</option>

<option value="Tamil Nadu">Tamil Nadu</option>

<option value="Telangana">Telangana</option>

<option value="Tripura">Tripura</option>

<option value="Uttar Pradesh">Uttar Pradesh</option>

<option value="Uttarakhand">Uttarakhand</option>

<option value="West Bengal">West Bengal</option>

</select>

<input type="text" name="address" placeholder="Address" required />

<select name="pm" class="form-control">

<option value="Cash on delivery">Cash on delivery</option>

<option value="Credit Card">Credit card</option>

<option value="Debit Card">Debit card</option>

</select>

<input type="number" value="{{total}}" name="total" hidden />

<input type="submit" class="btn" value="Confirm order" />

</form>

</div>

</div>

{%endblock%}

</body>

</html>

contact-us.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

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

{% extends 'base.html' %}

<link rel="stylesheet" type="text/css" href="/static/css/contact-us.css" />

<title>Contact us</title>

</head>

<body>

{% block content %}

<div class="contact-us-container">

<div class="contact-us-box">

<div class="left">

<p> :) Hello, Lets get in touch</p>

<form method="post" autocomplete="off">

{% csrf\_token %}

<input type="text" name="person-name" placeholder="Name" required />

<input type="email" name="email" placeholder="Email" required />

<input type="number" name="phone" placeholder="Phone (e.g. \*1458752366)" required />

<input name="msg" placeholder="Messege" required />

<input type="submit" class="sendbtn" value="Send" />

</form>

</div>

<div class="right">

<div class="top">

<h1>

We'd love to Hear <span>from you!</span>

</h1>

<p>info@gamestore.in</p>

</div>

<div class="bottom">

<p>Right now there is no us, i'm running the site alone. so every feedback you provide, any

suggestion

you have and any new ideas you like to share - please don't hesitate, write to me immediately.

</p>

</div>

</div>

</div>

</div>

{%endblock%}

</body>

</html>

about-us.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

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

{% extends 'base.html' %}

<link rel="icon" type="image/x-icon" href="/static/images/favicon.png" />

<link rel="stylesheet" type="text/css" href="/static/css/about-us.css" />

<title>Gamestore</title>

</head>

<body>

{%block content%}

<div class="about-box">

<div class="about-card">

<div class="image"><img src="/static/images/parth.png" /></div>

<div class="desc">

<h5>PARTH PANDYA</h5>

<p>Website Designer, Project Manager</p>

</div>

</div>

<div class="about-card">

<div class="image"><img src="/static/images/gohel.png" /></div>

<div class="desc">

<h5>DWARKESH GOHEL</h5>

<p>Website Designer, Frontend Devloper, Backand Devloper, Project Manager</p>

</div>

</div>

<div class="about-card">

<div class="image"><img src="/static/images/hems.png" /></div>

<div class="desc">

<h5>HEMANSHU DABHI</h5>

<p>Frontend Devloper, Backand Devloper</p>

</div>

<a href="#">

<div class="property-social-icons"></div>

</a>

</div>

</div>

{%endblock content%}

</body>

myorders.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

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

{% extends 'base.html' %}

<link rel="stylesheet" type="text/css" href="/static/css/myorders.css"/>

<title>My orders</title>

</head>

<body>

{% block content %}

<div class="myorders-container">

<div class="myorders-box">

<table>

<tr>

<th>ORDER NO</th>

<th>IMAGE</td>

<th>NAME</th>

<th>ORDER</th>

<th>DATE</th>

<th>STATUS</th>

<th>TOTAL</th>

</tr>

{% for i in items %}

<tr class="orders">

<td>#{{i.pk}}</td>

<td><img src="{{i.cart.product.image.url}}" width="100px" ></td>

<td>{{i.cart.product.name}}</td>

<td>{{i.cart.quantity}}</td>

<td>{{i.time}}</td>

<td>{{i.get\_status\_display}}</td>

<td>Rs.{{i.total}}</td>

</tr>

{% endfor %}

</table>

</div>

</div>

{% endblock %}

</body>

</html>

Apps.py

from django.apps import AppConfig

class CoreConfig(AppConfig):

default\_auto\_field = 'django.db.models.BigAutoField'

name = 'core'

Models.py

from django.contrib.auth.models import User

from django.db import models

class Product(models.Model):

name = models.CharField(max\_length=200, null=True)

image = models.ImageField(upload\_to='uploads/games/', null=True, blank=True)

mini\_1 = models.ImageField(upload\_to='uploads/games/', null=True,blank=True)

mini\_2 = models.ImageField(upload\_to='uploads/games/', null=True,blank=True)

mini\_3 = models.ImageField(upload\_to='uploads/games/', null=True,blank=True)

mini\_4 = models.ImageField(upload\_to='uploads/games/', null=True,blank=True)

price = models.CharField(max\_length=20,blank=True, null=True)

description = models.TextField(max\_length=500, null=True)

def \_\_str\_\_(self):

return (*self*.name)

class Cart(models.Model):

user = models.ForeignKey(User, on\_delete=models.CASCADE)

product = models.ForeignKey(Product, on\_delete=models.CASCADE)

quantity = models.IntegerField(default=1, blank=True, null=True)

total = models.IntegerField(blank=True, null=True)

timestamp = models.DateTimeField(auto\_now\_add=True)

updated = models.DateTimeField(auto\_now\_add=True)

active = models.BooleanField(default=True)

def \_\_str\_\_(self):

return (str(*self*.user) +" - "+str(*self*.product)+" - "+str(*self*.quantity)+ " - "+str(*self*.total))

class Contactus(models.Model):

name = models.CharField(max\_length=50, blank=True, null=True)

email = models.CharField(max\_length=50, blank=True, null=True)

phone = models.IntegerField(blank=True, null=True)

message = models.CharField(max\_length=50, blank=True, null=True)

def \_\_str\_\_(self):

return (*self*.email)

order\_status = (

('1','Pending'),

('2','Delivered')

)

class Orders(models.Model):

user = models.ForeignKey(User, on\_delete=models.CASCADE)

cart = models.ForeignKey(Cart, on\_delete=models.CASCADE)

total = models.IntegerField(blank=True, null=True)

firstname = models.CharField(max\_length=100, blank=True, null=True)

lastname = models.CharField(max\_length=100, blank=True, null=True)

email = models.CharField(max\_length=100, blank=True, null=True)

phone = models.IntegerField(blank=True, null=True)

city = models.CharField(max\_length=100, blank=True, null=True)

state = models.CharField(max\_length=100, blank=True, null=True)

address = models.CharField(max\_length=200, blank=True, null=True)

payment = models.CharField(max\_length=50, blank=True, null=True)

status = models.CharField(max\_length=50, choices=order\_status, default='1')

time = models.DateTimeField(auto\_now\_add=True)

def \_\_str\_\_(self):

return str(*self*.firstname)+" "+str(*self*.lastname)+" - "+str(*self*.total)

Urls.py

from django.urls import path

from . import views

urlpatterns = [

path('',views.home ,name='home'),

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

path('login/', views.login\_user, name='login'),

path('logout', views.logout\_user, name='logout'),

path('about-us/', views.about\_us, name='about-us'),

path('contact-us/', views.contact\_us, name='contact-us'),

path('add-to-cart/', views.add\_to\_cart),

path('remove-from-cart/', views.remove\_from\_cart),

path('cart/', views.go\_to\_cart, name='cart'),

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

path('order/', views.makeOrder, name='order'),

path('my-order/', views.myOrders, name='my-order'),

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

path('<pk>/',views.gameDetailView),

]

admin.py

from django.contrib import admin

from .models import Product, Cart, Contactus, Orders

*# Register your models here.*

admin.site.register(Product)

admin.site.register(Cart)

admin.site.register(Contactus)

admin.site.register(Orders)

Views.py

from django.shortcuts import redirect, render

from django.contrib.auth import login, logout

from .models import Orders, Product, Cart, Contactus

from django.contrib.auth.models import User

*# Create your views here.*

def home(request):

products = Product.objects.all()

context = { 'products' : products }

return render(request, "index.html", context)

def gameDetailView(request, pk):

*# view clicked game*

game = Product.objects.get(id=pk)

context = { 'game' : game }

return render(request, "game-detail-view.html", context)

def add\_to\_cart(request):

if request.method == "POST":

if request.user.is\_authenticated:

user = request.user

quantity = int(request.POST.get('quantity'))

game = Product.objects.get(id=request.POST.get('game-id'))

*# calculate total for specific game as per quantity*

price = int(game.price)

total = int(price)\*int(quantity)

*# get a game that is (active), users, product=game*

if Cart.objects.filter(user=user ,product=game, active=True).exists():

game\_item=Cart.objects.get(user=user, product=game)

*# increase total and quantity*

if game\_item != None:

game\_item.total += total

game\_item.quantity += quantity

game\_item.save()

else:

*# if not exists then create one*

cart = Cart.objects.create(user=user, product=game, quantity=quantity, total=total, active=True)

cart.save()

return redirect('cart')

return redirect('home')

def remove\_from\_cart(request):

if request.method == "POST":

*# get id cart-item id from hidden from and remove*

cart\_id\_item = request.POST.get('cart-id')

cart\_item = Cart.objects.filter(id=cart\_id\_item).exists()

if cart\_item:

Cart.objects.filter(id=cart\_id\_item).delete()

return redirect('cart')

return redirect('cart')

def go\_to\_cart(request):

if request.user.is\_authenticated:

user = request.user

*# if there is games in users cart*

if Cart.objects.filter(user=user).exists():

*# take all the (active) games that belongs to user*

cart\_items = Cart.objects.filter(user=user, active=True)

*# calculate total for all games in cart for every game that user put in cart*

cart\_total = 0

for i in cart\_items.values('total'):

cart\_total += i['total']

context = {'items':cart\_items, 'total':cart\_total}

return render(request, 'cart.html', context)

return redirect('home')

def register(request):

*# create new user store in DB*

if request.method == "POST":

if User.objects.filter(username=request.POST.get("username"), password=request.POST.get("password")).exists():

return redirect('error')

else:

user = User.objects.create(username=request.POST.get("username"), password=request.POST.get("password"))

user.save()

return redirect('home')

else:

if request.user.is\_authenticated == False:

return render(request, 'register.html')

else:

return redirect('home')

def login\_user(request):

*# login user if data is valid*

if request.method == "POST":

username = request.POST.get("username")

password = request.POST.get("password")

user = User.objects.filter(username=username, password=password).exists()

if user:

user\_logger = User.objects.get(username=username, password=password)

login(request, user\_logger)

return redirect('home')

else:

return render(request, 'login.html')

else:

if request.user.is\_authenticated == False:

return render(request, 'login.html')

else:

return redirect('home')

def logout\_user(request):

logout(request)

return redirect('home')

def about\_us(request):

return render(request, 'about-us.html')

def contact\_us(request):

*# take from data and store in DB*

if request.method == "POST":

name = request.POST.get('person-name')

email = request.POST.get('email')

phone = request.POST.get('phone')

msg = request.POST.get('msg')

data = Contactus.objects.create(name=name, email=email, phone=phone, message=msg)

data.save()

return render(request, 'contact-us.html')

return render(request, 'contact-us.html')

def makeOrder(request):

if request.method == "POST":

if request.user.is\_authenticated:

user = request.user

fn = request.POST.get('first-name')

ln = request.POST.get('last-name')

email = request.POST.get('email')

phone = request.POST.get('phone')

city = request.POST.get('city')

state = request.POST.get('state')

address = request.POST.get('address')

pm = request.POST.get('pm')

*#get cart total from hidden field*

total = request.POST.get('total')

*#get games that is (active and user belongs)*

user\_cart\_items = Cart.objects.filter(user=user, active=True)

for i in user\_cart\_items:

order = Orders.objects.create(user=user, cart=i, total=i.total, firstname=fn, lastname=ln, email=email, phone=phone, city=city, state=state, address=address, payment=pm)

i.active = False

i.save()

*# redirect to my orders page*

return redirect('my-order')

else:

return redirect('home')

def checkout(request):

*#pass the total to store in order DB*

if request.method == "POST":

total = request.POST.get('total')

context = {'total':total}

return render(request, 'checkout.html', context)

else:

return redirect('home')

def myOrders(request):

if request.user.is\_authenticated:

user = request.user

*# get deavtive carts means purchesd games*

user\_cart = Cart.objects.filter(user=user, active=False)

user\_orders = Orders.objects.filter(user=user)

context = {'items':user\_orders}

return render(request, 'myorders.html', context)

def error(request):

context = {'user\_exist':'User already exist try using different username'}

return render(request, 'error.html', context)

*AGILE DOCUMENTATION*

5.1 AGILE PROJECT CHARACTER

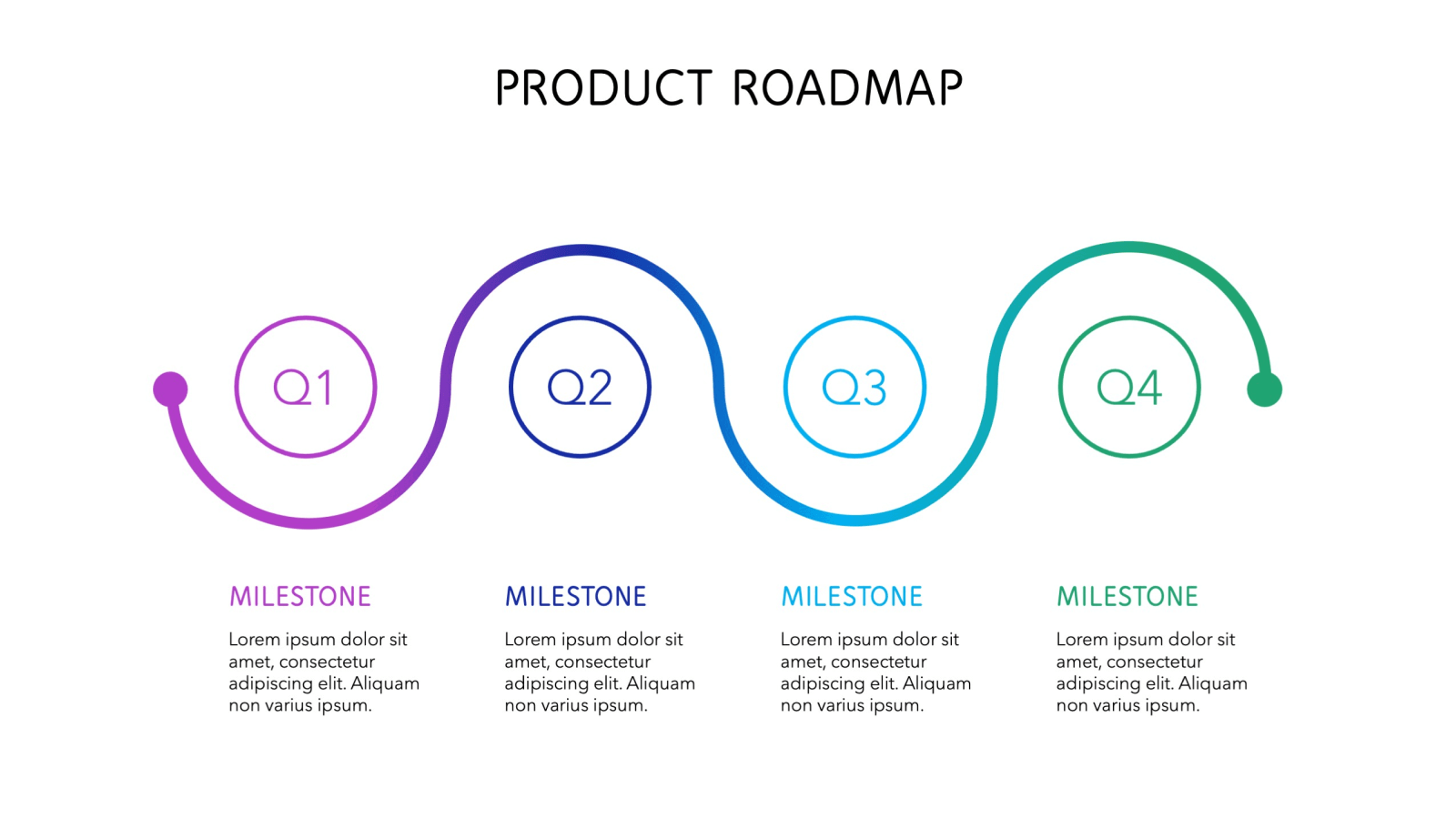
An Agile team charter is **a guiding document that outlines the overall mission, goals, values, and operational scope of an Agile team**. It is used to keep teams on track as they undertake various Agile projects, ensuring they do not stray too far from their original purpose.

An **Agile project charter** is a simple, one-page document created at the beginning of the development of this project. This chart serves as a road-map for achieving the intended results of the system. It consists of three items: the project's vision, a mission statement, and success criteria.

| Project name | E-Commerce System |
| --- | --- |
| Project Developer | Dwarkesh Gohel (215530694005)  Himanshu Dabhi (215530694004)  Parth Pandya (215530694002) |
| Project Guide | Prof. Chirag Rachcha  Miss.Dhuvi Sidhpura |
| Expected start date | 20-5-22 |
| Expected end date | 05-06-22 |

5.2 AGILE ROADMAP/ SCHEDULE

A product roadmap enables the product manager of an agile team to set a vision for the product and share that vision with stakeholders.This a time based plan that defines what system milestone is, what it wants to serve, how to accomplish this milestone.



5.3 AGILE PROJECT PLANE

Most project managers are used to a project plan that has a series of tasks laid out for the entire project, listing task durations, responsibility assignments, and dependencies. Plans are developed in this manner based on the assumption that the Project Manager, hopefully along with the team, can predict up front everything that will need to happen in the project, how long it will take, and who will be able to do it

5.4 AGILE USER STORY A good way to think about a user story is that it is a reminder to have a conversation with your customer, which is another way to say it's a reminder to do some just-in-time analysis. In short, user stories are very slim and high-level requirements artifacts.

5.5 AGILE RELEASE PLANNING **Agile release planning** is an approach to product management that takes into account the intangible and flexible nature of software development—as part of this approach, teams **plan** iteration sprints across incremental releases

5.6 AGILE TEST PLANE

An Agile Test Plan is a very important document because it gives your Quality Assurance (QA) team the ability to have all high-level scenarios, business requirements and estimates in one place. Your QA Analyst or Agile Tester should fill out an Agile Test Plan during each sprint planning event. And just like the actual live document, it is always changing and evolving, depending on sprint requirements and overall timelines.

An Agile Test Plan should have a proper and clear structure containing business inputs and QA tasks.

This part specifies how the delivery of the future application will be tested. It defines the following areas:

* User stories
* QA environment
* Testing scope
* Testing process
* Risks and dependencies
* Estimates and Exit criteria

*PROPOSED ENHANCEMENTS*

FEATURE OF SYSTEM

* User friendliness
* Flexibility

***USER FRIENDLINESS:-***

To make system more user friendly, following action have been taken

* System provide appropriate message
* User can purchase the product
* Attempts are made to reduce response time
* Attractive screen are generated so familiar person can also operate the system very comfortably

***FLEXIBILITY:-***

The system is made as possible as possible so that it can be used in any kind of situation and accommodate and kind of change in the real world.

For that at the time of designing all possibilities were considered and also possible changes in future were considered

*CONCLUSION*

* While developing this project this project we have learnt a lot about E-Commerce system
* The online ordering system was designed to replace manual process of bookings for a product ordering and other facility
* The old system does not serve the customer in a better way, rather than it makes customer data vulnerable
* The new system keeps proper records of customers for emergency and security purposes.
* We have learn how to make a system user friendly
* We also realized the importance of maintaining a minimal margin for error.
* During the development process we studied carefully and understood the criteria of implementation process.

*BIBLIOGRAPHY*

To develop this “Web application of online information of computer courses” we used Python(Django) as front end and SQL SERVER as back end. We take some knowledge towards automation system from some books and web references that are given below.

* [www.javatpoint.com](http://www.javatpoint.com)
* [www.tutorialspoint.com](http://www.tutorialspoint.com)
* [www.w3schools.com](http://www.w3schools.com)

***Developed By***

Dwarkesh Gohel (215530694005)

Himanshu Dabhi (215530694004)

Parth Pandya (215530694002)