**A Project Report**

On

**“QuickShop(Ecommerce Shopping Website)”**

Submitted in partial fulfilment of the requirement of

**University of Mumbai**

For the Degree of

**Bachelor of Computer Science**


Submitted By

**“Manas Harshal Aher”**

Under the Guidance of

**Prof. Mrs. Nilam Shelar**

**Ramniranjan Jhunjhunwala College of Arts, Science and Commerce (Empowered Autonomous), Ghatkopar(W)**

**Affiliated to University of Mumbai.**


**(2023-2024)**

# INDEX

# Acknowledgement

I have great pleasure for representing this project report entitled **"QuickShop(Ecommerce Shopping Website)"** and I grab this opportunity to convey my immense regards towards all the distinguished people who have their valuable contribution in the hour of need.

I take this opportunity to thank "**Dr.Himanshu Dawda**", our Principal of Ramniranjan Jhunjhunwala college, Ghatkopar(W) for giving mean opportunity to study in the institute and the most needed guidance throughout the duration of the course.

I also like to extend my gratitude to "**Prof.(Mrs.) Anita Gaikwad**", Head of the Department for their timely and prestigious guidance and necessary support during each phase of the project.

I also like extend my gratitude to "**Prof.(Mrs.)Nilam shelar**",for the timely and prestigious guidance and necessary support during each phase of the project.

I also owe to my fellow friends who have been constant source of help to solve the problems and also helped me during the project Analysis phase. I would like to thank my parents and friends who have helped in various phases of the completion of the project.

Thanking You ,

Manas Aher

# Declaration

I declare that this written submission represents my own ideas in my own words and where other's ideas or words have been included,I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/fact/data/source in my submission.

I have completed all the documentation work on my own under guidance of our professor. The project has not been copied, duplicated or plagiarized from any other paper, journal, document or book.

I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has notbeen taken when needed.

Manas Aher

# Preliminary Investigation

## 3.1 Software Requirements:

- **Frontend:**

**HTML**:- HTML (Hypertext Markup Language) is a standard markup language used for creating web pages. It uses tags to structure and format content, allowing for the display oftext, images, links, and other elements in web browsers.

**CSS:-** CSS (Cascading Style Sheets) is a stylesheet language used for describing thepresentation and layout of HTML documents, enabling the design and styling of web content.

**JavaScript:-** JavaScript is a versatile, high-level programming language primarily used for web development to add interactivity, manipulate web page content, and enhance userexperiences.

**Bootstrap:-** Bootstrap is a popular front-end framework that simplifies web development by providing pre-designed and responsive CSS and JavaScript components,making it easier to create attractive and consistent web interfaces.

- **Backend:**

**MySQL:-** A relational database stores data in separate tables rather than putting allthe data in one big storeroom. The database structure is organized into physical files optimized for speed.

**PHP:-** An extremely popular scripting language that is used to create dynamic Webpages. Combining syntax from the C, Java and Perl languages, PHP code is embedded within HTML pages for server side execution.

- **Operating System:** Windows/ Linux

## 3.2 Hardware Requirements:

- Processor : Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80 GHz
- RAM : 8.00 GB
- Solid State Drive : 145 GB
- Monitor :[resolution : 1920 x 1080 ]
- Keyboard
- Mouse

## 3.2 Description of System

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

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

Website allows customers to make purchases securely and efficiently. Distinct features such as 1-click purchase and secure, encrypted, credit card payment make this information management system highly reliable and easy.

User Login and Registration: The system allows customers to login and create accounts easily. Users can log in using their email or social media credentials to access the platform. After add to Cart and buy now product System asks for Customer Details like Shipping Address and Payment Method.

Product Details: Website provide different kind of product that are related to Computer components. Like Motherboard, Keyboard, Mouse, GPU, CPU, etc.

Efficient Customer Support: To provide exceptional customer service, the system should offer various support channels, such as a email support, or a helpline. I would prioritize responsive customer support to address any queries or concerns promptly.

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

## 3.3 Limitation of Current System

### 1. Security

The biggest drawback of e-commerce is the issue of security. People fear to provide personal and financial information, even though several improvements have been made in relation to data encryption. Certain websites do not have capabilities to conduct authentic transactions. Fear of providing credit card information and risk of identity limit the growth of e-commerce.

### 2. Lack of privacy

Many websites do not have high encryption for secure online transaction or to protect online identity. Some websites illegally collect statistics on consumers without their permission. Lack of privacy discourages people to use internet for conducting commercial transactions.

### 3. Fear

People fear to operate in a paperless and faceless electronic world. Some of the business organizations do not have physical existence, People do not know with whom they are conducting commercial transactions. This aspect makes people to opt physical stores for purchases.

### 4. Product suitability

People have to rely on electronic images to purchase products. Sometimes, when the products are delivered, the product may not match with electronic images. Finally, it may not suit the needs of the buyers. The lack of 'touch and feel' prevent people from online shopping.

### 5. High Labour cost

Highly talented and technically qualified workforce are required to develop and manage the websites of the organization. Since internet provides a lot of job opportunities, business organizations have to incur a lot of expenses to retain a talented pool of employees,

### 6. Huge technological cost

It is difficult to merge electronic business with traditional business. Technological infrastructure may be expensive and huge cost has to be incurred to keep pace with ever changing technology. It is necessary to allocate more funds for technological advancement to remain competitive in the electronic world.

# 3.4 Proposed System and its Advantages

## 1. Reduced overhead costs

Running an e-commerce store is a lot more cost-effective than running a physical store. You don't have to rent commercial real estate — instead, you can pay an affordable fee for web hosting. You don't have to invest in security for your commercial property, plus you don't have to worry about paying rent for a warehouse or hiring employees.

## 2. No need for a physical storefront

There are so many difficult aspects to running a physical storefront and using e-commerce means you don't have to face most of those obstacles. Renting a commercial property can be expensive, especially if you're in a big city. You also have to pay for electricity, water, and internet to ensure your space is up to code and can handle your business.

## 3. Ability to reach a broader audience

Perhaps the biggest advantage of e-commerce is the fact that it allows you to reach a massive audience. Your physical storefront can only get so many visitors in a day, especially if you live in a smaller town or a rural area. With an e-commerce store, you can reach potential customers all throughout the world and show them your products. Even if you're not selling your products overseas, you can still reach shoppers all the way across the United States to boost your sales.

## 4. Scalability

Of the advantages and disadvantages of a business using e-commerce, scalability is one of the most practical advantages for long-term growth. If you have a physical storefront, your business can only grow so much before you have to move to a larger storefront. You also have to move inventory and equipment from one location to another, which makes it even harder to scale your store with the growth of your business.

## 5. Track logistics

Keeping track of logistics is an essential part of e-commerce and retail marketing, and it's significantly easier with e-commerce than it is with a physical storefront. You can outsource fulfillment logistics so your customers can enjoy benefits like 2-day shipping and easy returns processing. You also have an electronic record of everything, which makes it easy to track sales and look for trends that help you grow your business over time.

# 4 Feasibility Study

## Introduction

A feasibility study is simply an assessment of the practicality of a proposed project plan or method. This is done by analyzing technical, economic, legal, operational and time feasibility factors.

An effective feasibility study points a project in the right direction by helping decision-makers have a holistic view of the potential benefits, disadvantages, barriers and constraints that could affect its outcome. The main purpose of a feasibility study is to determine whether the project can be not only viable but also beneficial from a technical, financial, legal and market standpoint.

- **What Is Included in a Feasibility Study Report?**

The findings of your project feasibility study are compiled in a feasibility report that usually includes the following elements.

1. Executive summary
2. Description of product/service
3. Technology considerations
4. Product/service marketplace
5. Marketing strategy
6. Organization/staffing
7. Schedule
8. Financial projections
9. Findings and recommendations

As you're researching the feasibility study, project management software can help you keep track that information. Project Manager does more than just collect items as with most to-do lists. Assign a team member to gather the feasibility analysis data you need and see their progress in real time. If needed, they can attach supporting documents.

## Types of Feasibility Study

a) Technical Feasibility Study

b) Economic Feasibility Study

c) Operational Feasibility Study

## 4.1 Technical Feasibility

Technical feasibility is a standard practice for companies to conduct feasibility studies before commencing work on a project. Businesses undertake a technical feasibility study to assess the practicality and viability of a product or service before launching it. Whether you are working as a product engineer, product designer or team manager, there may be plenty of situations in your career where you are required to prepare a technical feasibility study.

A technical feasibility study helps organizations determine whether they have the technical resources to convert the idea into a fully functional and profitable working system. It helps in troubleshooting the project before commencing work. The study identifies potential challenges and uncovers ways to overcome them. It also helps in long-term planning, as it can serve as a flowchart for how products and services evolve before they reach the market.

- What Is The Purpose Of A Technical Feasibility Study?

A technical feasibility study helps find the answers to the following questions:

1. Is it possible to develop the product with the available technology in the company?
2. Is the organization equipped with the necessary technology for project completion?
3. Are there technically strong employees who can deliver the product on time and within budget using the available technology?
4. Is there scope in the company's budget to add more technical resources?

Key considerations include:

- Data Security and Privacy: Ensuring data security and privacy is vital, especially when dealing with user information and payment details. Implementing encryption, secure authentication methods, and compliance with data protection regulations should be a priority.

- User Interface and Experience: Ensuring a user-friendly interface and seamless user experience is vital for the success of the system. Feasibility studies should include UX/UI design considerations.

- Mobile Responsiveness: Considering the prevalence of mobile usage, the software should be mobile-responsive and compatible with various devices and screen sizes.

- Testing and Quality Assurance: Feasibility studies should include plans for thorough testing and quality assurance to identify and rectify any bugs or issues.

## 4.2 Economical feasibility

The economic feasibility step of business development is that period during which a break-even financial model of the business venture is developed based on all costs associated with taking the product from idea to market and achieving sales sufficient to satisfy debt or investment requirements.

For an organization to achieve one of its main purposes — to grow and be profitable — it takes planning. In this context, one of the most important steps in the investment selection and prioritization process is the economic feasibility analysis.

In the economic feasibility project, all the factors that influence a given initiative must be analyzed. We can mention, for example, the company's market, products, and services, investment projection, competition, competitive advantages, cash flow, market trends, working capital, labor, among others.

This study will provide an overview of the current situation of the business, both in the present and in the future, in relation to the execution of a given project. This is a complex process that requires great effort, as there are several variables to be raised.

However, it is an extremely necessary action for those companies that intend to have consistency and secure evolution in relation to investments.

### 1. Development Cost

Whatever required for the project are easily available online.

- Equipment required for developing the software are easily available.
- Equipment maintenance is also minimum.
- Once the required hardware and software requirements get fulfilled there is no need for the users of our system to spend for any additionaloverhead.

### 2. Benefits which cannot be measured

- Increased customer Loyalty.
- Increased customer satisfaction.
- User friendly.
- Easy to use.

# 4.3 Operational feasibility

Operational feasibility is the ability to utilize, support and perform the necessary tasks of a system or program. It includes everyone who creates, operatesor uses the system. To be operationally feasible, the system must fulfill a need required by the business.

## The Need for Operational Feasibility Studies

Operational feasibility studies are generally utilized to answer the following questions:

1. Process – How do the end-users feel about a new process that may be implemented?
2. Evaluation – Whether or not the process within the organization will work but also if it can work.
3. Implementation – Stakeholder, manager, and end-user tasks.
4. Resistance – Evaluate management, team, and individual resistance and how that resistance will be handled.

The operational feasibility of the Ecommerce Website system assesses whether the proposed system can be effectively implemented and integrated into the existing Ecommerce Website workflow.

### Time feasibility

A time feasibility study will take into account the period in which the project is going to take up to its completion. A project will fail if it takes too long to be completed before it is useful. Typically this means estimating how long the system will take to develop, and if it can be completed in a given time period using some methods like payback period. Time feasibility is a measure of how reasonable the project timetable is.

### Resource feasibility

Describe how much time is available to build the new system, when it can be built, whether it interferes with normal business operations, type and amount of resources required, dependencies, and developmental procedures with company revenue prospectus.

### Financial feasibility

- Total estimated cost of the project
- Financing of the project in terms of its capital structure, debt to equity ratio and promoter's share of total cost
- Existing investment by the promoter in any other business

# 5 Gantt Chart

A Gantt chart is a popular project management tool used to visualize project schedules, tasks, and timelines. It provides a graphical representation of a project's tasks, their durations, and their interdependencies.

In a Gantt chart, tasks are listed on the vertical axis (Y-axis), and time is represented on the horizontal axis (X-axis). Each task is represented by a horizontal bar that spans across the timeframe it will take to complete. The bars may be color-coded to indicate different aspects, such as task type, priority, or responsible team member.

These charts are helpful for project managers and team members to understand the project's progress, identify potential bottlenecks, and manage resources effectively. Various project management software tools allow you to create dynamic Gantt charts with additional features such as task dependencies, milestones, and critical path analysis.

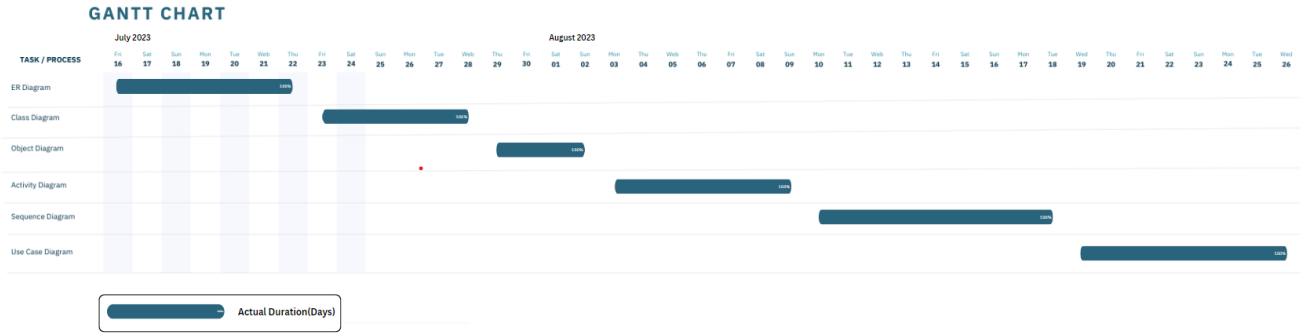
**A Gantt chart is made up of several different elements. Below given are 8 key components to know how to read a Gantt chart:**

- **Task list:** Runs vertically down the left of the Gantt chart to describe project work and may be organized into groups and subgroups**.**
- **Timeline:** Runs horizontally across the top of the Gantt chart and shows months, weeks, days, and years.
- **Dateline:** A vertical line that highlights the current date on the Gantt chart.
- **Bars:** Horizontal markers on the right side of the Gantt chart that represent tasks and show progress, duration, and start and end dates.
- **Milestones:** Yellow diamonds that call out major events, dates, decisions, and deliverables.
- **Dependencies:** Light gray lines that connect tasks that need to happen in a certain order.
- **Progress:** Shows how far along work is and may be indicated by *% Complete* and/or bar shading.
- **Resource assigned:** Indicates the person or team responsible for completing a task.

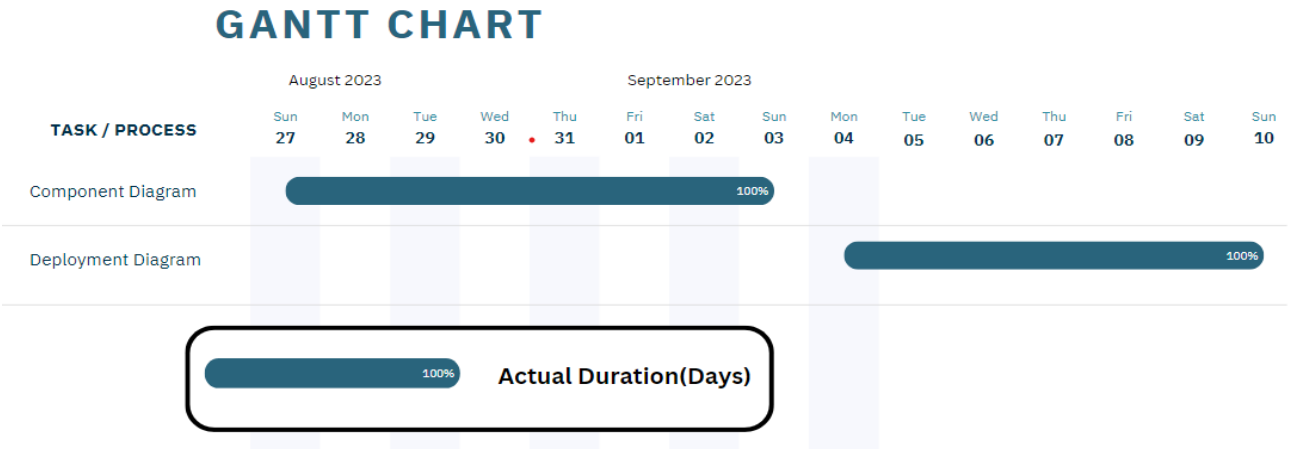
| Title | Start date | End date | Duration (days) |
|---|---|---|---|
| **Requirement Analysis** | | | |
| Preliminary Investigation | 16-06-2023 | 18-06-2023 | 3 |
| Project Topic Discussion | 19-06-2023 | 25-06-2023 | 6 |
| Current System Description | 26-06-2023 | 02-07-2023 | 7 |
| Proposed System Description | 03-07-2023 | 09-07-2023 | 7 |
| Feasibility Study | 10-07-2023 | 15-07-2023 | 6 |
| **System Analysis** | | | |
| ER Diagram | 16-07-2023 | 22-07-2023 | 7 |
| Class Diagram | 23-07-2023 | 28-07-2023 | 6 |
| Object Diagram | 29-07-2023 | 02-08-2023 | 5 |
| Activity Diagram | 03-08-2023 | 09-08-2023 | 7 |
| Sequence Diagram | 10-08-2023 | 18-08-2023 | 9 |
| Use Case Diagram | 19-08-2023 | 26-08-2023 | 8 |
| **System Design** | | | |
| Component Diagram | 27-08-2023 | 03-09-2023 | 8 |
| Deployment Diagram | 04-09-2023 | 10-09-2023 | 7 |
| **Design Phase** | | | |
| Table Design | 11/09/2023 | 21/09/2023 | 10 |
| Screen Layout | 22/09/2023 | 18/10/2023 | 27 |
| **System Coding** | | | |
| Coding | 19/10/2023 | 25/01/2024 | 97 |
| **System Testing** | | | |
| Testing | 26/01/2024 | 10/02/2024 | 15 |
| **Deployment** | | | |
| Deployment | 11/02/2024 | 15/02/2024 | 5 |

## 1. Requirement Analysis Gantt Chart:-

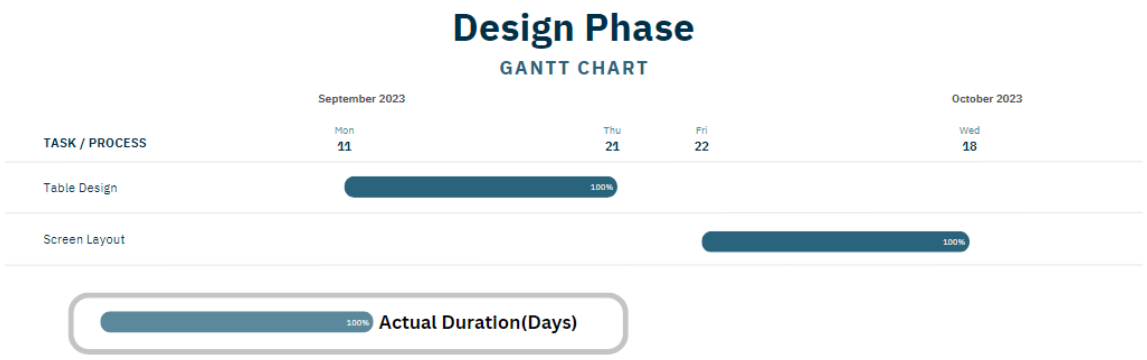

## 2. System Analysis Gantt Chart:-



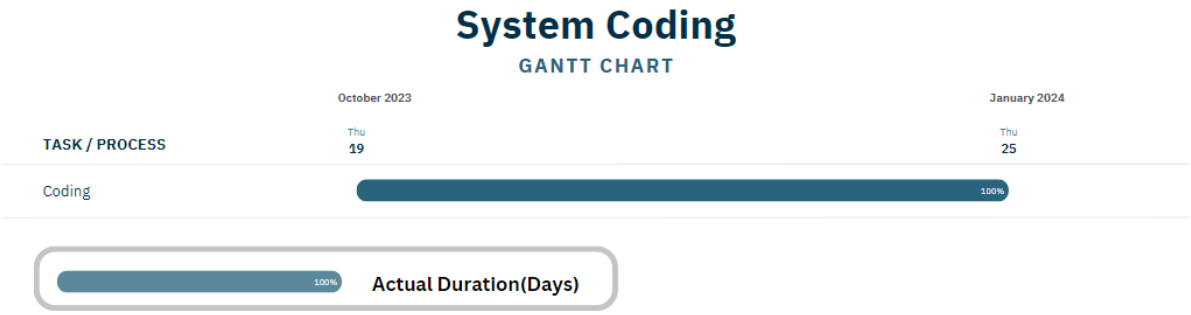

## 3. System Design Gantt Chart:-



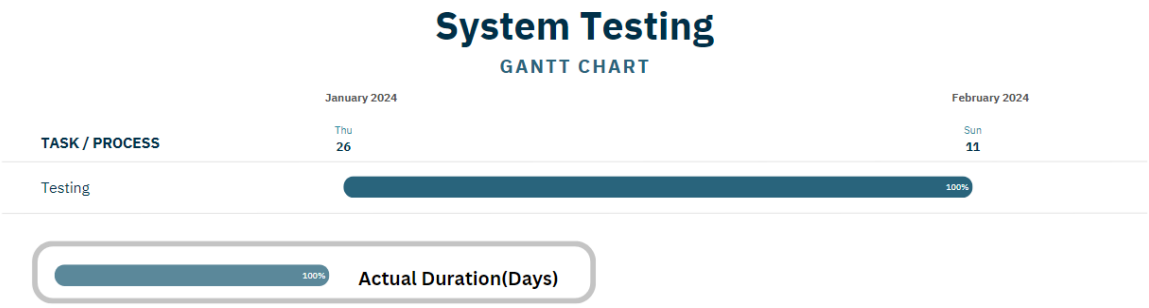
## 4. Design Phase Gantt Chart:-



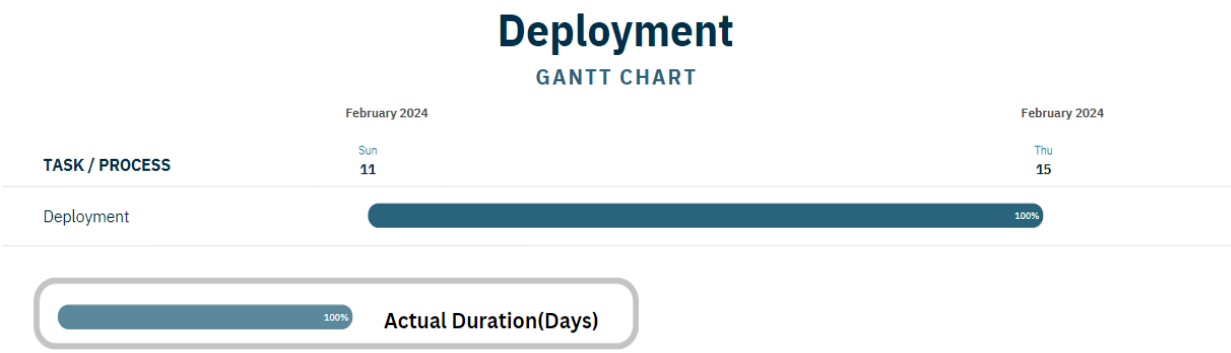

## 5. System Coding Gantt Chart:-




## 6. System Testing Gantt Chart:-




## 6. Deployment Gantt Chart:-



# 6 Software Development Model

## Agile Model :

The meaning of Agile is swift or versatile." Agile process model" refers to a software development approach based on iterative development. Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.

Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements. Each iteration involves a team working through a full software development life cycle including planning, requirements analysis, design, coding, and testing before a working product is demonstrated to the client.

### 1. Requirement Gathering

This is the initial stage of the development cycle in which the product needs are understood from the perspective of the client. This phase entails extensive discussion with the consumer to ascertain his expectations and precise requirements. This is a critical task that must be effectively handled because the majority of clients are unsure about what they require. At this point, acceptance test design is completed since business requirements may be utilised as input for acceptance testing.

### 2. Design The Requirements

During the design phase, the development team produces a software blueprint. They specify the architecture as well as the high-level technical specifics necessary to achieve the requirements' functionality. You may use a user flow diagram or a high-level UML diagram to demonstrate the functionality of new features and how they will interact with your existing system.

### 3. Construction\Iteration

Once the team has determined the criteria, the work may begin. Designers and developers start working on their projects with the objective of producing a viable product. Before being published, the product will go through multiple rounds of development, thus it will have minimal, primitive capabilities. Finally, a non-static product or service is deployed.

## 4. Testing

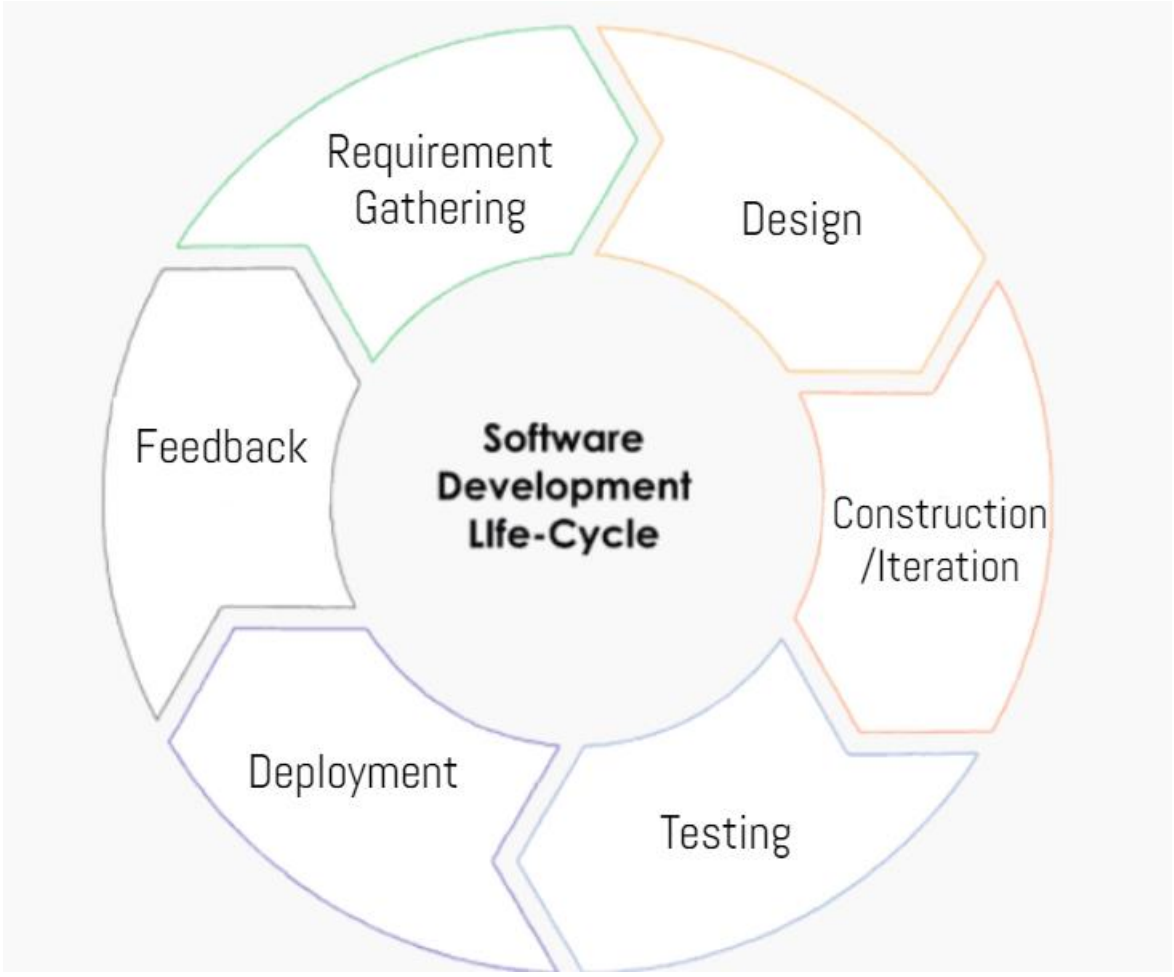
Testing is an integral part of each development iteration. The Quality Assurance team checks the product's performance and looks for bugs during this phase. The team performs various types of testing, such as unit testing, integration testing, and user acceptance testing, to ensure that the software works correctly and meets the specified requirements.

## 5. Development

After a sufficient number of iterations, the software is deemed ready for release and is deployed to the production environment for actual use.

## 6.Feedback

After the software has been launched, the Agile team continues to monitor its performance and collect user input. Based on the comments, they produce updates, problem repairs, and enhancements.



# 7 System Analysis

## 7.1 Entity Relationship Diagram

The ER or (Entity Relational Model) is a high-level conceptual data model diagram. Entity-Relation model is based on the notion of real-world entities andthe relationship between them.

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system.

ER diagrams are related to data structure diagrams (DSDs), which focus on the relationships of elements within entities instead of relationships between entitiesthemselves. ER modelling is something regarded as a complete approach to design a logical database schema. This is incorrect because the ER diagram is just an approximate description of data, constructed through a very subjective evaluation of the information collected during requirements analysis.

ER Diagrams are composed of entities, relationships and attributes. They alsodepict cardinality, which defines relationships in terms of numbers.

- **Entity:**

  An entity is an object or component of data. An entity isrepresented as rectangle in an ER diagram.
  For example: Student and College and these two entities have many to onerelationship as many student studies in a single college.
  An entity that cannot be uniquely identified by its own attributes and relieson the relationship with other entity is called **weak entity**. The weak entityis represented by a double rectangle.

- **Attribute:**

  An attribute describes the property of an entity. An attribute is represented as Oval in an ER diagram. There are four types of attributes:

  1. Key attribute
  2. Composite attribute
  3. Multivalued attribute
  4. Derived attribute

- **Relationship:**

A relationship is represented by diamond shape in ER diagram, it shows the relationship among entities. There are four types of relationships:

1. One to One
2. One to Many
3. Many to One
4. Many to Many

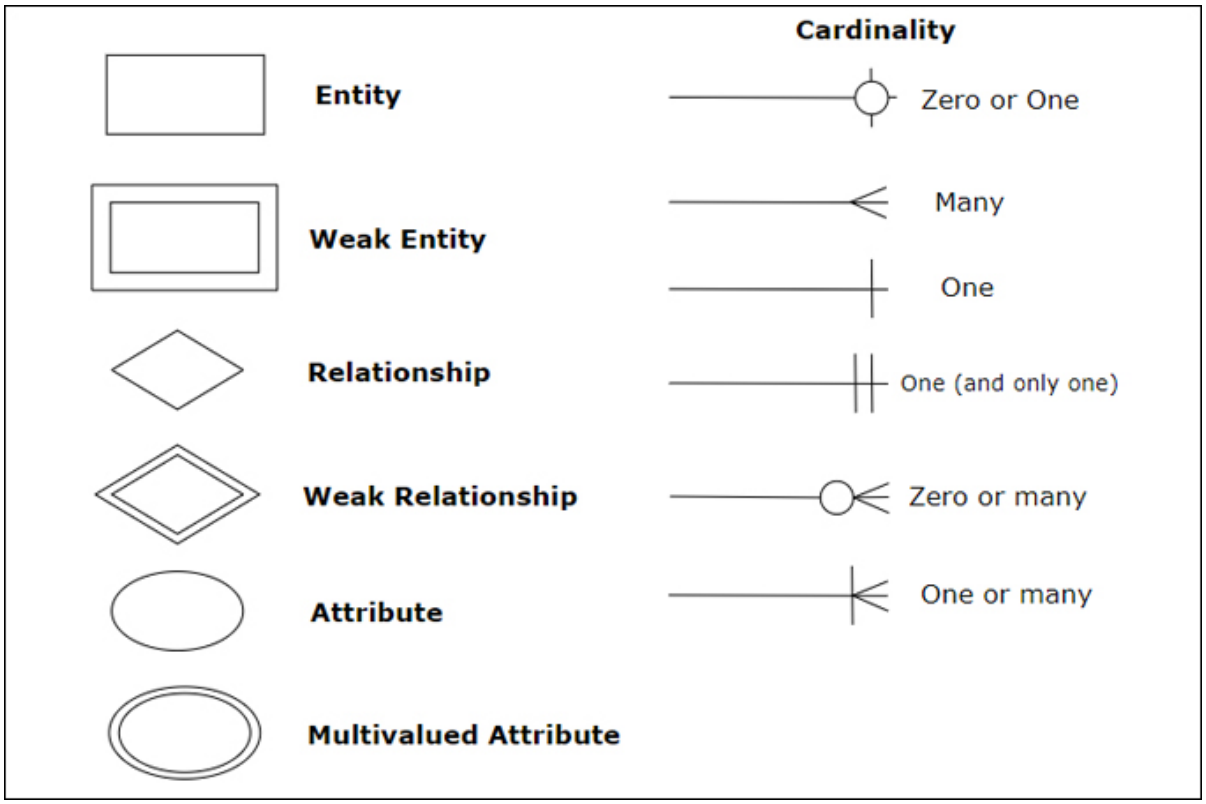
1. **One-to-One:** When each entity in each entity set can take part only once in the relationship, the cardinality is one-to-one. Let us assume that a male can marry one female and a female can marry one male. So the relationship will be one-to-one.
   The total number of tables that can be used in this is 2.

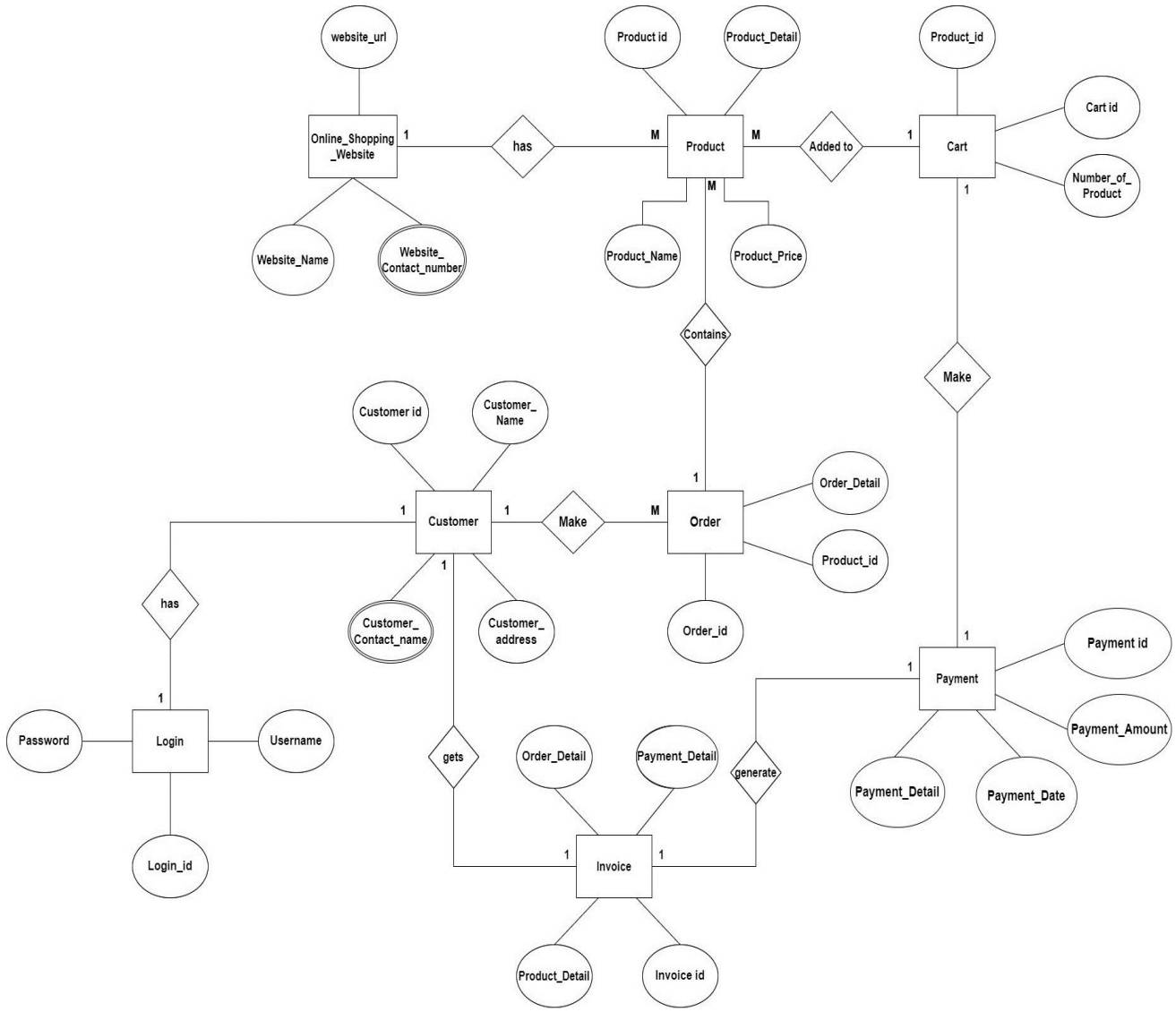
2. **One-to-Many:** In one-to-many mapping as well where each entity can be related to more than one relationship and the total number of tables that can be used in this is 2.

3. **Many-to-One:** When entities in one entity set can take part only once in the relationship set and entities in other entity sets can take part more than once in the relationship set, cardinality is many to one. Let us assume that a student can take only one course but one course can be taken by many students. So the cardinality will be n to 1. It means that for one course there can be n students but for one student, there will be only one course.
   The total number of tables that can be used in this is 3.

4. **Many-to-Many:** When entities in all entity sets can take part more than once in the relationship cardinality is many to many. Let us assume that a student can take more than one course and one course can be taken by many students. So the relationship will be many to many.

- **ER Diagram Symbols:**




website_url
Online_Shopping
_Website
1
has
M
Product
Product id
Product_Detail
M
Added to
1
Cart
Product_id
Cart id
Number_of_
Product
1
Website_Name
Website_
Contact_number
Product_Name
Product_Price
M
Contains
Make
Customer id
Customer_
Name
1
Customer
1
Make
M
Order
1
Order_Detail
Product_id
Order_id
1
has
Customer_
Contact_name
Customer_
address
1
Payment
1
Payment id
Payment_Amount
Payment_Detail
Payment_Date
Password
Login
Username
1
Login_id
gets
Order_Detail
Payment_Detail
generate
1
Invoice
1
Product_Detail
Invoice id

## 7.2 Class Diagram

A class diagram in the Unified Modeling Language (UML) is atype of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and therelationships among objects.

It is a model which is used to show the classes constituting a system and their interrelationship. It is based on UML. Only the important attributes and methods are shown in Class diagrams. In the initial period of analysis, the important attributes of the classes, which must be captured andthe functionalities provided by the class may not be very clear. As the analysis progresses, the attributes and methods may be added. If more focusis on interrelationships of classes, then the attributes and methods may not be shown in the class diagram.

The class diagram is used to identify and classify the objectswhich constitute a system. It also includes the important attributes of the objects which must be captured.

### Purpose of Class Diagrams

- Shows static structure of classifiers in a system
- Diagram provides a basic notation for other structure diagrams prescribedby UML
- Helpful for developers and other team members too
- Business Analysts can use class diagrams to model systems from a business perspective

### A UML class diagram is made up of:

- A set of classes and
- A set of relationships between class

### Class Notation

- A class notation consists of three parts:

### Class Name

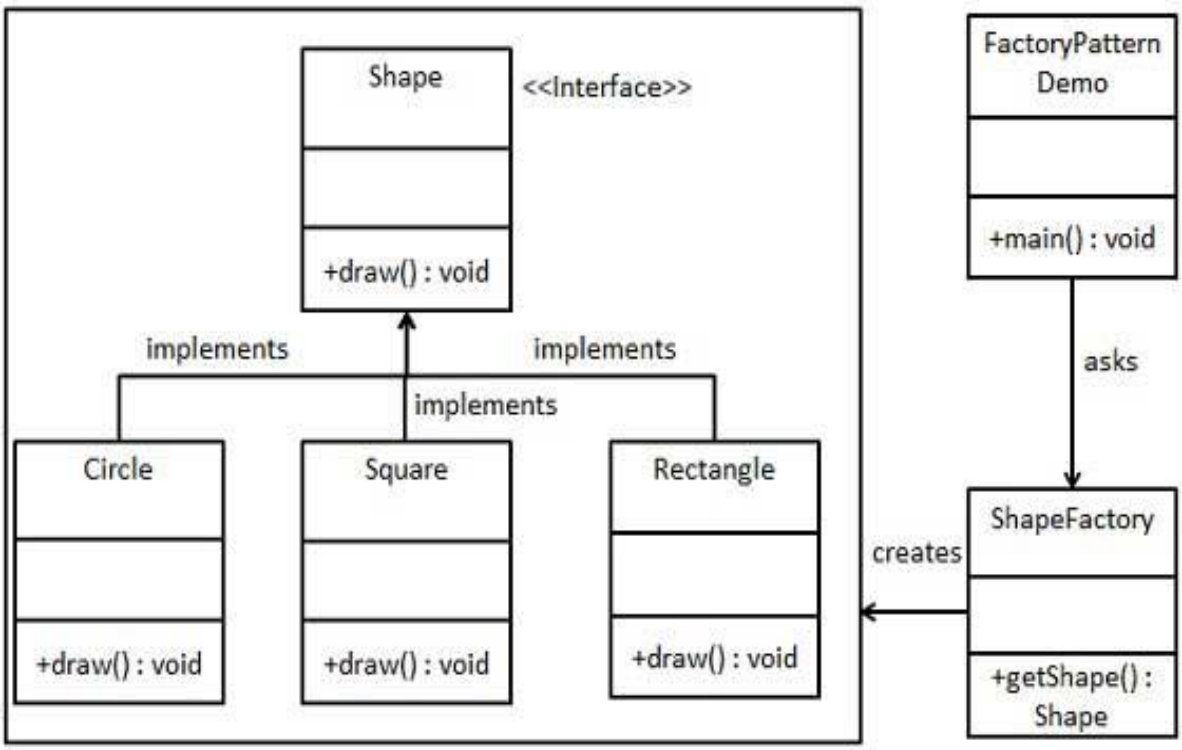
- The name of the class appears in the first partition.

## Class Attributes

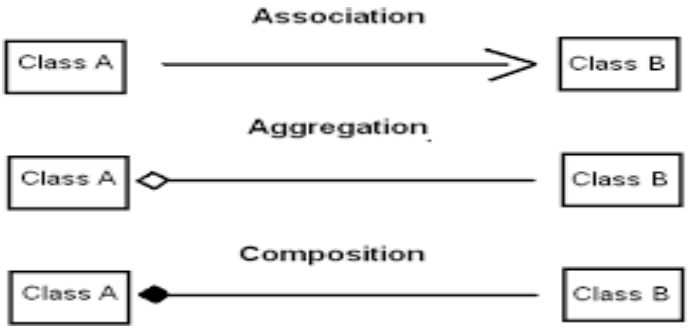
- Attributes are shown in the second partition.
- The attribute type is shown after the colon.
- Attributes map onto member variables (data members) in code.

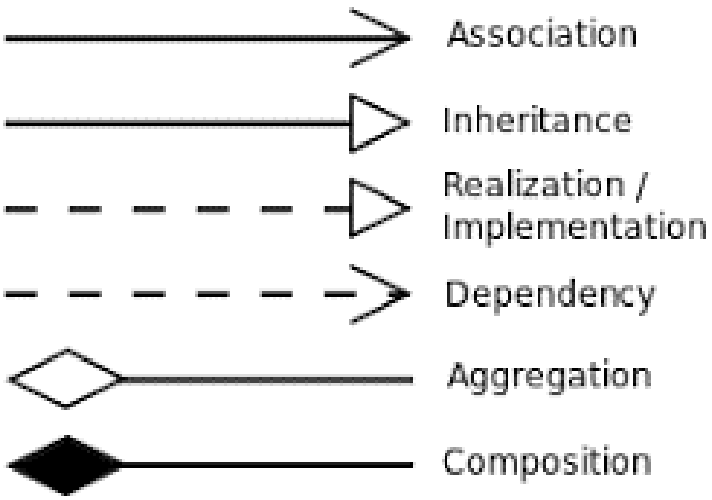
## Class Operations (Methods)

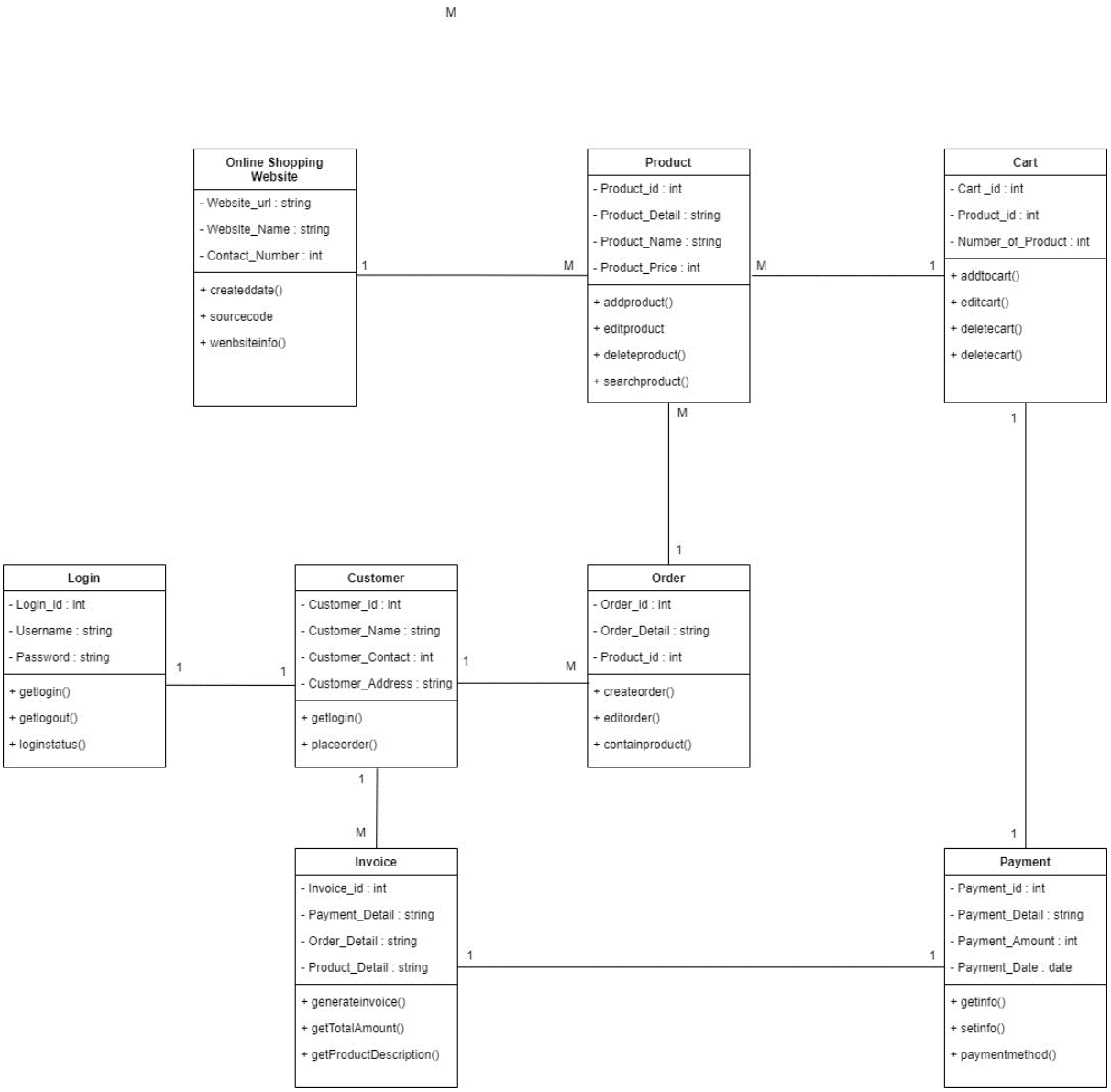
- Operations are shown in the third partition. They are services the class provides.
- The return type of a method is shown after the colon at the end of themethod signature.
- The return type of method parameters is shown after the colon followingthe parameter name.
- Operations map onto class methods in code.



**Class Relation Diagram**




M
Online Shopping Website
- Website_url : string
- Website_Name : string
- Contact_Number : int
+ createddate()
+ sourcecode
+ wenbsiteinfo()
1
M
Product
- Product_id : int
- Product_Detail : string
- Product_Name : string
- Product_Price : int
+ addproduct()
+ editproduct
+ deleteproduct()
+ searchproduct()
M
1
Cart
- Cart _id : int
- Product_id : int
- Number_of_Product : int
+ addtocart()
+ editcart()
+ deletecart()
+ deletecart()
M
1
1
1
Login
- Login_id : int
- Username : string
- Password : string
+ getlogin()
+ getlogout()
+ loginstatus()
1
1
Customer
- Customer_id : int
- Customer_Name : string
- Customer_Contact : int
- Customer_Address : string
+ getlogin()
+ placeorder()
1
M
Order
- Order_id : int
- Order_Detail : string
- Product_id : int
+ createorder()
+ editorder()
+ containproduct()
1
M
Invoice
- Invoice_id : int
- Payment_Detail : string
- Order_Detail : string
- Product_Detail : string
+ generateinvoice()
+ getTotalAmount()
+ getProductDescription()
1
1
Payment
- Payment_id : int
- Payment_Detail : string
- Payment_Amount : int
- Payment_Date : date
+ getinfo()
+ setinfo()
+ paymentmethod()

## 7.3 Object Diagram

An object diagram is a graph of instances, including objects and data values. A static object diagram is an instance of a class diagram; it shows a snapshot of thedetailed state of a system at a point in time.
Object diagrams and class diagrams are closely related and use almost identical notation. Both diagrams are meant to visualize static structure of a system. Whileclass diagrams show classes, object diagrams display instances of classes (objects).
Object diagrams are more concrete than class diagrams. They are often used to provide examples or act as test cases for class diagrams. Only aspects of currentinterest in a model are typically shown on an object diagram.

Object diagrams are derived from class diagrams so object diagrams aredependent upon class diagrams.

### Purpose of Object Diagrams

The use of object diagrams is fairly limited, mainly to show examples of data structures.

- During the analysis phase of a project, you might create a class diagram todescribe the structure of a system and then create a set of object diagrams as test cases to verify the accuracy and completeness of the class diagram.
- Before you create a class diagram, you might create an object diagram todiscover facts about specific model elements and their links, or to illustrate specific examples of the classifiers that are required.
- The purpose of a diagram should be understood clearly to implement itpractically. The purposes of object diagrams are similar to class diagrams.
- The difference is that a class diagram represents an abstract model consisting of classes and their relationships. However, an object diagram represents an instance at a particular moment, which is concretein nature.
- It means the object diagram is closer to the actual system behavior. Thepurpose is to capture the static view of a system at a particular moment.
- The purpose of the object diagram can be summarized as −
    - Forward and reverse engineering.
    - Object relationships of a system
    - Static view of an interaction.
    - Understand object behavior and their relationship from practicalperspective.

## Basic Object Diagram Symbols and Notations

## Object Names:

Every object is actually symbolized like a rectangle, that offers the name fromthe object and its class underlined as well as divided with a colon.

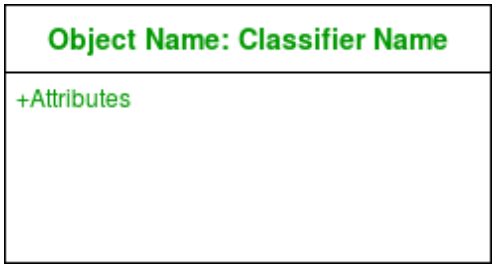
## Object Attributes:

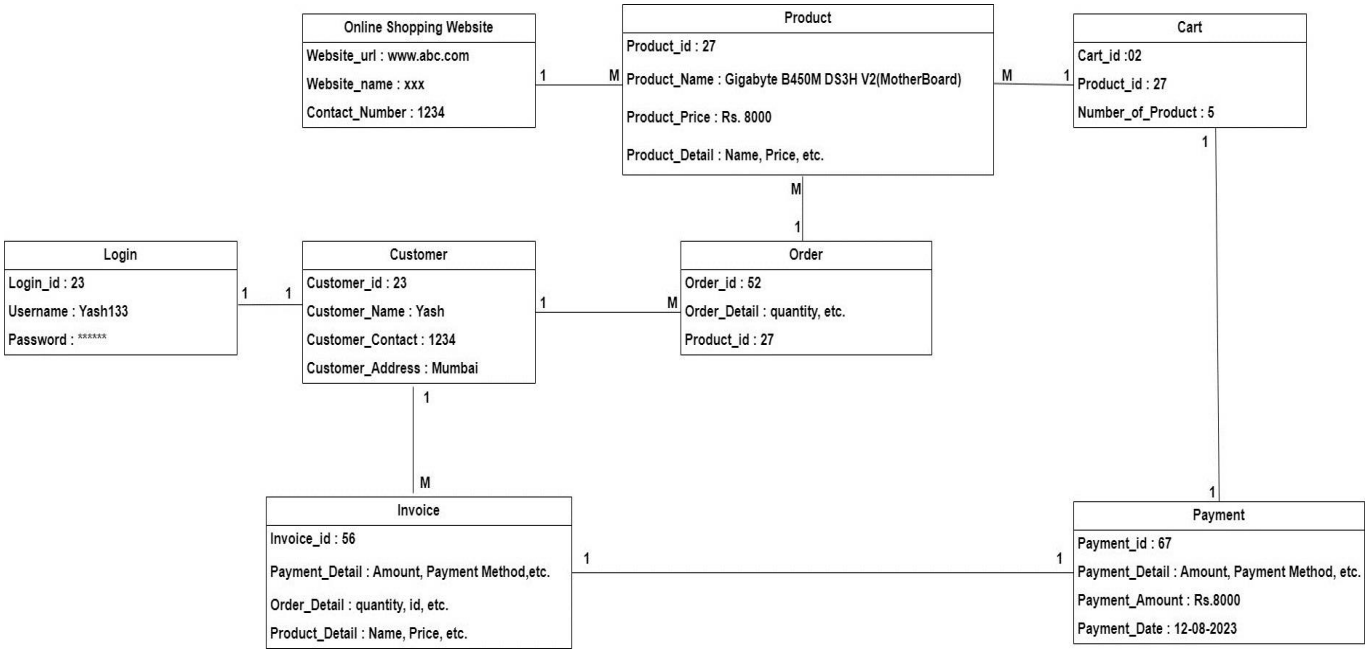
Similar to classes, you are able to list object attributes inside a separate compartment. However, unlike classes, object attributes should have valuesassigned for them.

## Links:

Links tend to be instances associated with associations. You can draw a linkwhile using the lines utilized in class diagrams.

## Notation:




Online Shopping Website
Website_url : www.abc.com
Website_name : xxx
Contact_Number : 1234
1
M
Product
Product_id : 27
Product_Name : Gigabyte B450M DS3H V2(MotherBoard)
Product_Price : Rs. 8000
Product_Detail : Name, Price, etc.
M
1
Cart
Cart_id :02
Product_id : 27
Number_of_Product : 5
1
M
1
Login
Login_id : 23
Username : Yash133
Password : ******
1
1
Customer
Customer_id : 23
Customer_Name : Yash
Customer_Contact : 1234
Customer_Address : Mumbai
1
M
Order
Order_id : 52
Order_Detail : quantity, etc.
Product_id : 27
1
M
Invoice
Invoice_id : 56
Payment_Detail : Amount, Payment Method,etc.
Order_Detail : quantity, id, etc.
Product_Detail : Name, Price, etc.
1
1
Payment
Payment_id : 67
Payment_Detail : Amount, Payment Method, etc.
Payment_Amount : Rs.8000
Payment_Date : 12-08-2023

## 7.4 Activity Diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system.

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc

### Purpose of Activity Diagrams

The basic purposes of activity diagrams is similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used toshow message flow from one activity to another.

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

It does not show any message flow from one activity to another. Activity diagramis sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

The purpose of an activity diagram can be described as −

- Draw the activity flow of a system.
- Describe the sequence from one activity to another.
- Describe the parallel, branched and concurrent flow of the system.

### When to Use Activity Diagram

Activity Diagrams describe how activities are coordinated to provide a service which can be at different levels of abstraction. Typically, an event needs to be achieved by some operations, particularly where the operation is intended to achieve a number of different things that require coordination, or how the events in a single use case relate to one another, in particular, use cases where activities may overlap and require coordination. It is also suitable for modelinghow a collection of use cases coordinate to represent business workflows

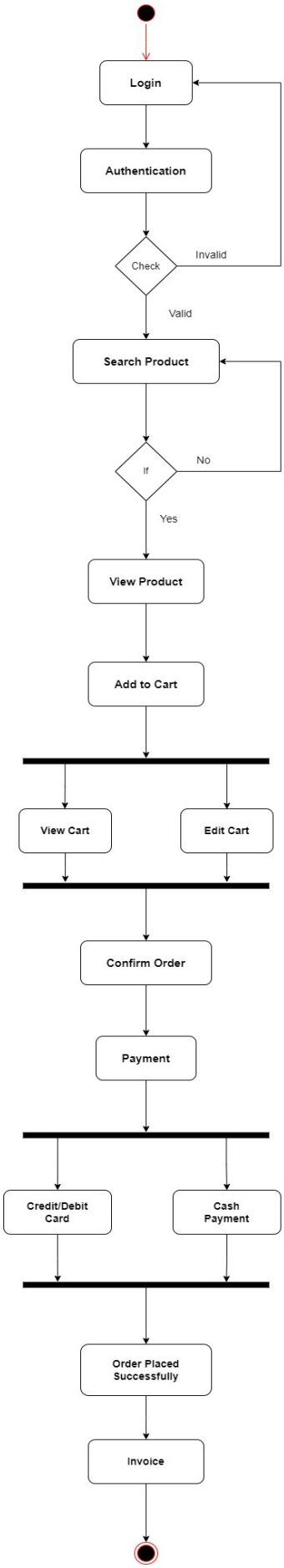
1. Identify candidate use cases, through the examination of business workflows
2. Identify pre- and post-conditions (the context) for use cases
3. Model workflows between/within use cases
4. Model complex workflows in operations on objects
5. Model in detail complex activities in a high level activity Diagram

## The most important shape types:

- Rounded rectangle represents activities.
- Diamonds represent decisions.
- Bars represent the start (split) or end (join) of concurrent activities.
- A black circle represents the start (initial state) of the workflow.
- An encircled black circle represents the end (final state).
- Arrows run from the start towards the end and represent the order in which activities happen.

- **Activity diagram can be used for:**

  - Modeling work flow by using activities.
  - Modeling business requirements.
  - High level understanding of the system's functionalities.
  - Investigating business requirements at a later stage.


Login
Authentication
Check
Invalid
Valid
Search Product
If
No
Yes
View Product
Add to Cart
View Cart
Edit Cart
Confirm Order
Payment
Credit/Debit Card
Cash Payment
Order Placed Successfully
Invoice

## 7.5 Sequence Diagram

A sequence diagram describes interactions among classes in terms of an "Exchange of message over time". Sequence diagrams are used to depict the timesequence of message exchanged between objects. Message can correspond to operation on class or an event trigger. Notations of a Sequence Diagram include:

- Lifeline: It is a vertical dashed line that represents the "lifetime" of an object.
- Arrows: They indicate flow of message between objects.

Activation: It is a thin rectangle showing period of time, during which an objectis performing an action.

### Purpose of Sequence Diagram

- Model high-level interaction between active objects in a system
- Model the interaction between object instances within a collaboration thatrealizes a use case
- Model the interaction between objects within a collaboration that realizesan operation
- Either model generic interactions (showing all possible paths through the interaction) or specific instances of a interaction (showing just one path through the interaction)

### Basic Sequence Diagram Notation Class

Roles and Participants

Class roles describe the way on object will behave in context. Use the UML object symbol to illustrate class roles, but don't list object attributes.

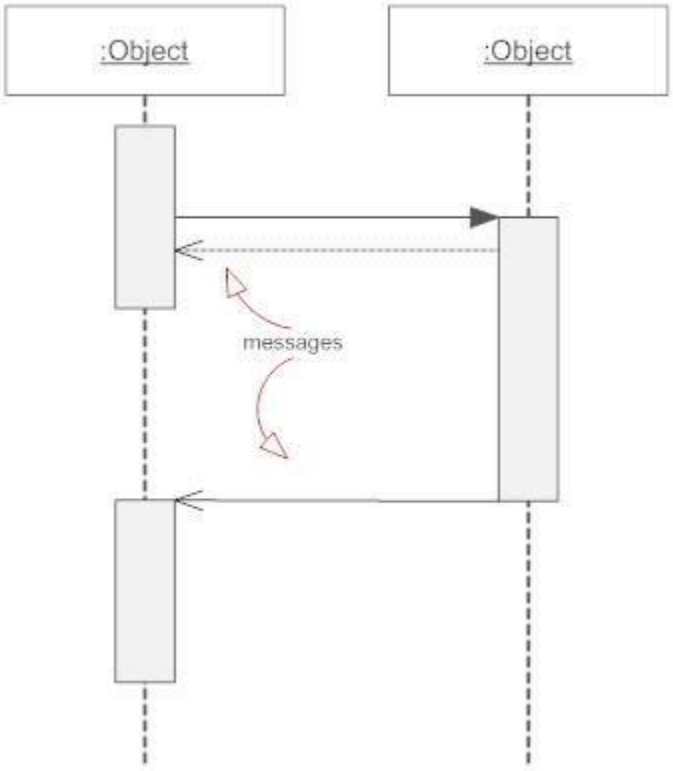

### Activation or Execution Occurrence

Activation boxes represent the time an object needs to complete a task. When anobject is busy executing a process or waiting for a reply message, use a thin grayrectangle placed vertically on its lifeline.

Activation or Execution Occurrence

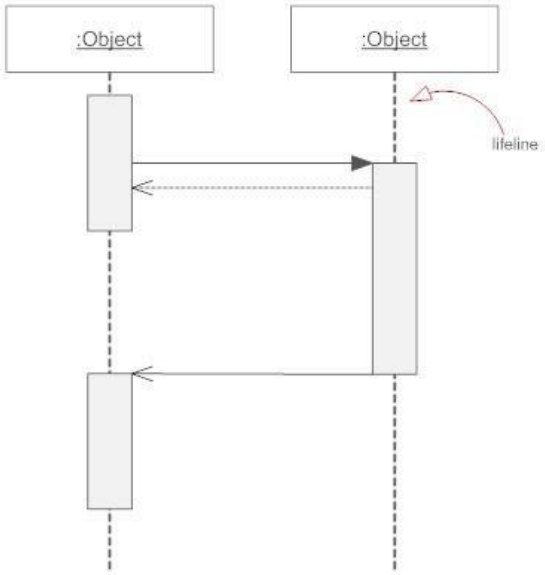
## Messages

Messages are arrows that represent communication between objects. Use half- arrowed lines to represent asynchronous messages. Asynchronous messages aresent from an object that will not wait for a response from the receiver before continuing its tasks. For message types, see below.




## Lifelines

Lifelines are vertical dashed lines that indicate the object's presence over time.




### Destroying Objects

Objects can be terminated early using an arrow labeled "<< destroy >>" that points to an X. This object is removed from memory. When that object's lifelineends, you can place an X at the end of its lifeline to denote a destruction occurrence.

### Loops

A repetition or loop within a sequence diagram is depicted as a rectangle. Placethe condition for exiting the loop at the bottom left corner in square brackets [ ].

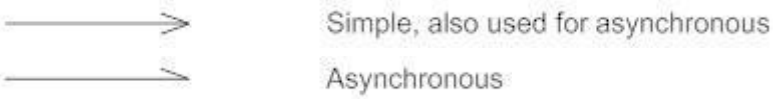
## Types Of Messages in Sequence Diagram

### Synchronous Message

A synchronous message requires a response before the interaction can continue. It's usually drawn using a line with a solid arrowhead pointing from one object toanother.


### Asynchronous Message

Asynchronous messages don't need a reply for interaction to continue. Like synchronous messages, they are drawn with an arrow connecting two lifelines; however, the arrowhead is usually open and there's no return message depicted

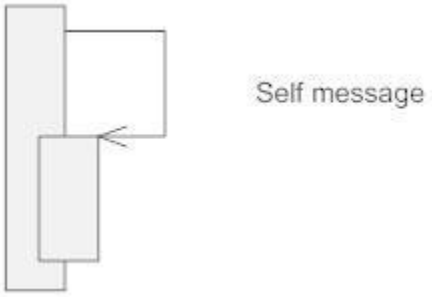



### Reply or Return Message

A reply message is drawn with a dotted line and an open arrowhead pointingback to the original lifeline.

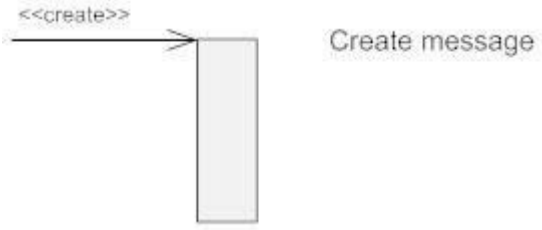
## Self Message

A message an object sends to itself, usually shown as a U shaped arrow pointingback to itself.




## Create Message

This is a message that creates a new object. Similar to a return message, it's depicted with a dashed line and an open arrowhead that points to the rectanglerepresenting the object created.




## Delete Message

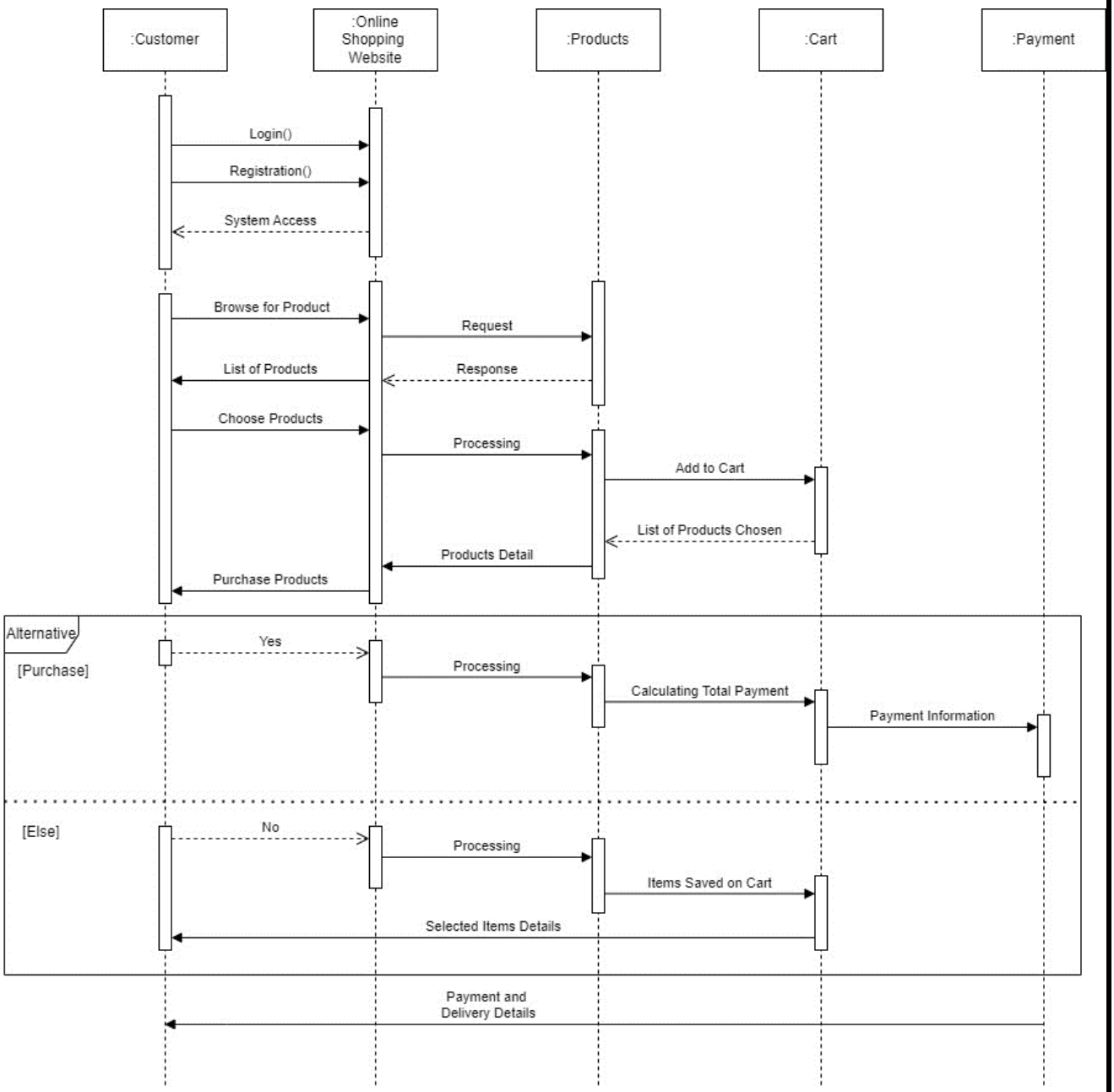
This is a message that destroys an object. It can be shown by an arrow with an xat the end.


## Found Message

A message sent from an unknown recipient, shown by an arrow from an endpointto a lifeline.


## Lost Message

A message sent to an unknown recipient. It's shown by an arrow going from alifeline to an endpoint, a filled circle or an x.


:Customer
:Online Shopping Website
:Products
:Cart
:Payment
Login()
Registration()
System Access
Browse for Product
Request
List of Products
Response
Choose Products
Processing
Add to Cart
List of Products Chosen
Products Detail
Purchase Products
Alternative
[Purchase]
Yes
Processing
Calculating Total Payment
Payment Information
[Else]
No
Processing
Items Saved on Cart
Selected Items Details
Payment and Delivery Details

## 7.6 Use Case Diagram

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

The main purpose of a use case diagram is to portray the dynamic aspect of a system. It accumulates the system's requirement, which includes both internal as well as external influences. It invokes persons, use cases, and several things that invoke the actors and elements accountable for the implementation of use case diagrams. It represents how an entity from the external environment can interact with a part of the system.

Following are the purposes of a use case diagram given below:

1. It depicts the external view of the system.
2. It recognizes the internal as well as external factors that influence the system.
3. It gathers the system's needs.
4. It represents the interaction between the actors.

### How to draw a Use Case diagram?

It is essential to analyze the whole system before starting with drawing a use case diagram, and then the system's functionalities are found. And once every single functionality is identified, they are then transformed into the use cases to be used in the use case diagram.

After that, we will enlist the actors that will interact with the system. The actors are the person or a thing that invokes the functionality of a system. It may be a system or a private entity, such that it requires an entity to be pertinent to the functionalities of the system to which it is going to interact.

Once both the actors and use cases are enlisted, the relation between the actor and use case/ system is inspected. It identifies the no of times an actor communicates with the system. Basically, an actor can interact multiple times with a use case or system at a particular instance of time.

Following are some rules that must be followed while drawing a use case diagram:

1. A pertinent and meaningful name should be assigned to the actor or a use case of a system.
2. The communication of an actor with a use case must be defined in an understandable way.
3. Specified notations to be used as and when required.
4. The most significant interactions should be represented among the multiple no of interactions between the use case and actors.

Use case diagrams are drawn to capture the functional requirements of a system. After identifying the above items, we have to use the following guidelines to draw an efficient use case diagram

- The name of a use case is very important. The name should be chosen insuch a way so that it can identify the functionalities performed.
- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships, as the main purpose of thediagram is to identify the requirements.
- Use notes whenever required to clarify some important points.

### Use case diagrams can be used for −

- Requirement analysis and high level design.
- Model the context of a system.
- Reverse engineering.
- Forward engineering.

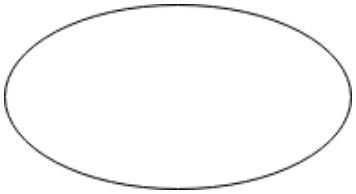
### Important tips for drawing a Use Case diagram

Following are some important tips that are to be kept in mind while drawing a use case diagram:

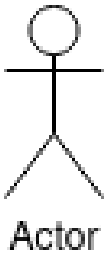
1. A simple and complete use case diagram should be articulated.
2. A use case diagram should represent the most significant interaction among the multiple interactions.
3. At least one module of a system should be represented by the use case diagram.
4. If the use case diagram is large and more complex, then it should be drawn more generalized.

**Use case Diagram depicts :-**

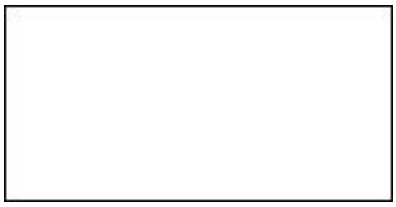
- **Use cases** : A use case represents a function or an action within the system. It's drawn as an oval and named with the function.



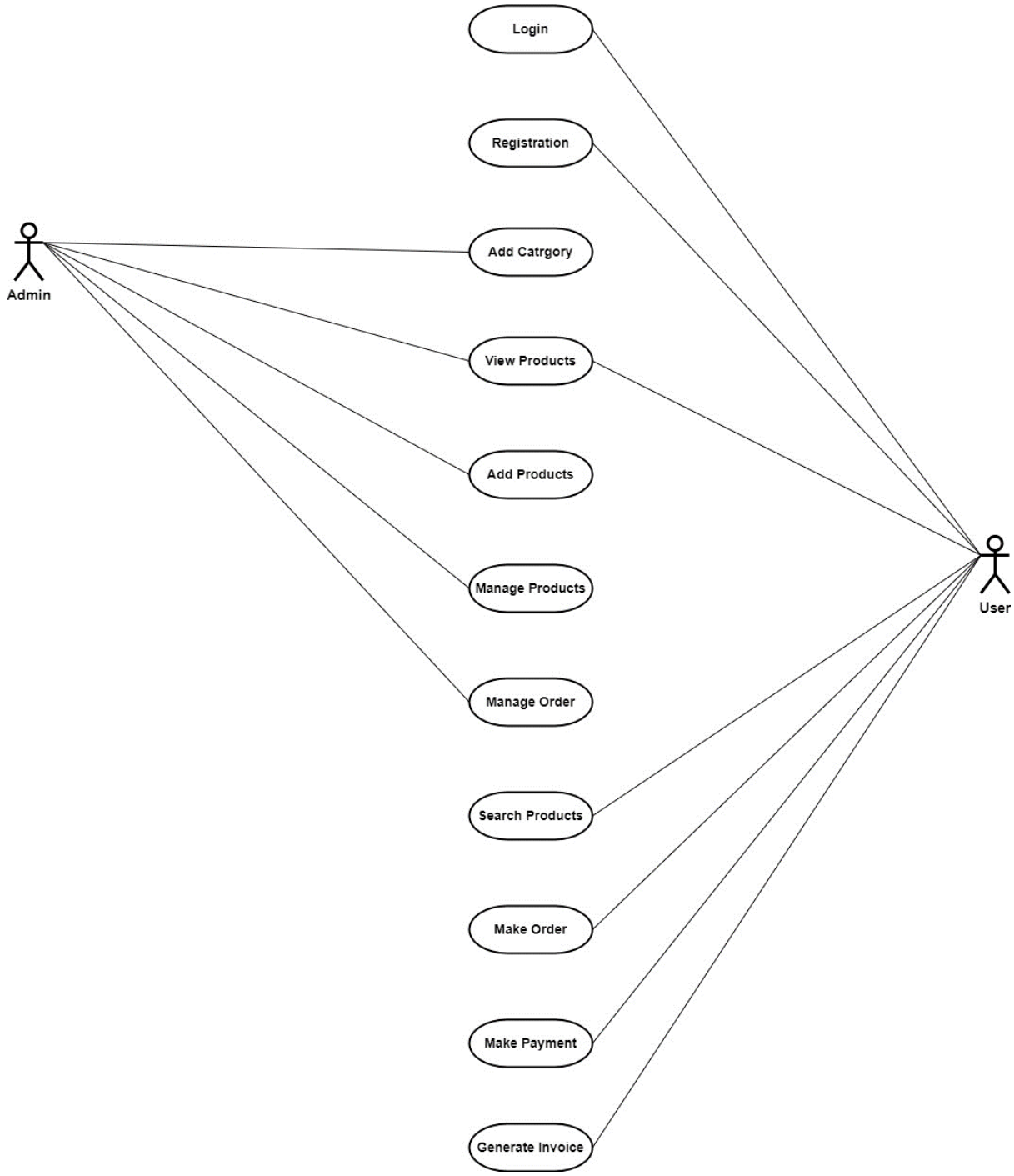
- **Actor** : Actor in a use case diagram is any entity that performs a role in one given system. This could be a person, organization or an external system and usually drawn like the skeleton shown below.




- **System** : The system is used to define the scope of the use case drawn as a rectangle. This is an optional element but useful when you're visualizing large systems. For example, you can create all the use cases and then use the system object to define the scope covered by your project. Or you can use it to show the different areas covered in different releases.



- **Association** : These are the connecting lines.


Login
Registration
Admin
Add Catrgory
View Products
Add Products
Manage Products
Manage Order
User
Search Products
Make Order
Make Payment
Generate Invoice

# 8 System Design

## 8.1 Component Diagram

Component diagrams are different in terms of nature and behavior. Component diagrams are used to model the physical aspects of a system. Now the question is,what are these physical aspects? Physical aspects are the elements such asexecutables, libraries, files, documents, etc. which reside in a node.

Component diagrams are used to visualize the organization and relationships among components in a system. These diagrams are also used to make executablesystems.

### Purpose of Component Diagrams

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe thefunctionality of the system but it describes the components used to make those functionalities.

Thus from that point of view, component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files,etc.

Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at aparticular moment.

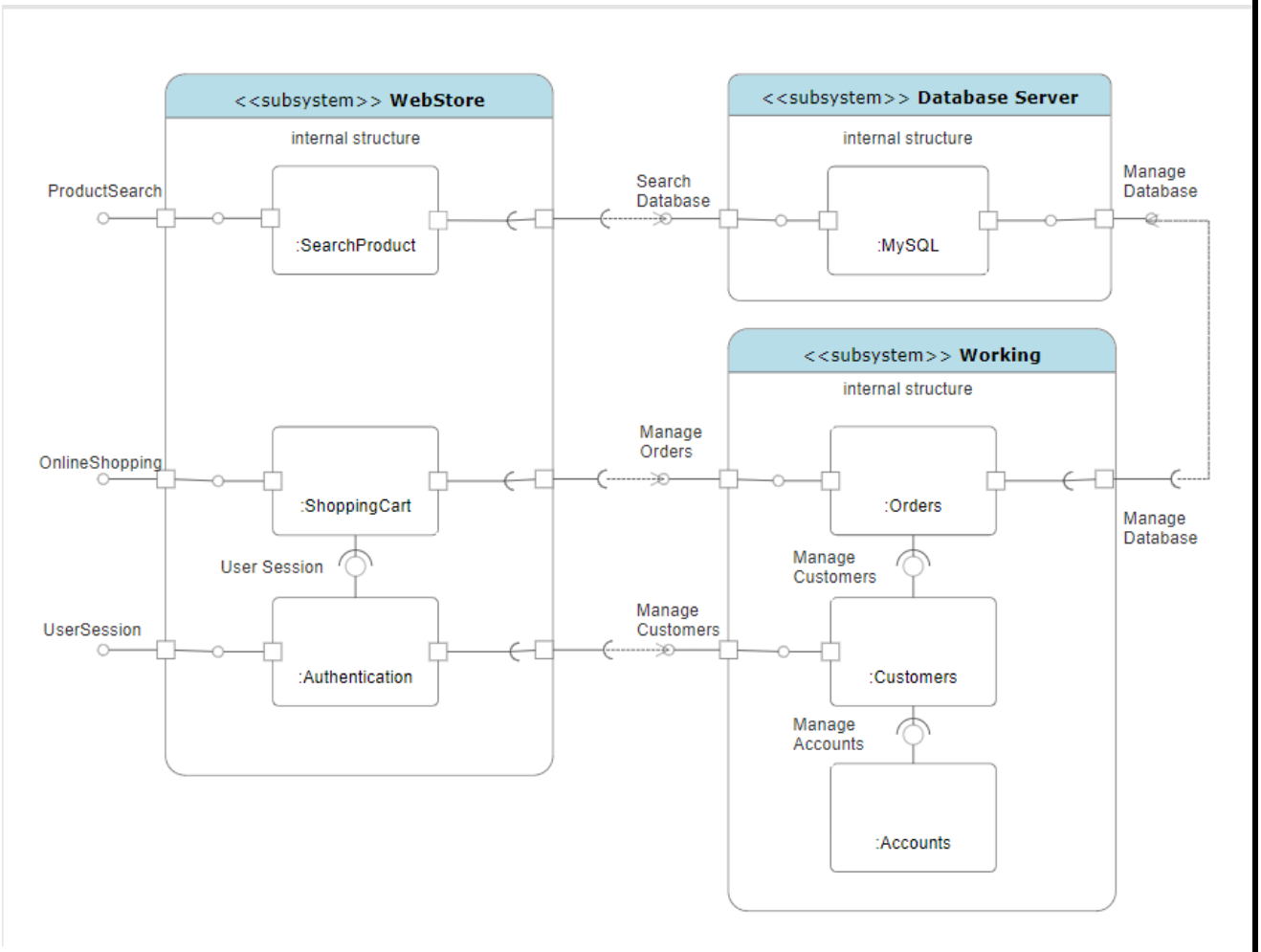
A single component diagram cannot represent the entire system but a collection of diagrams is used to represent the whole.

The purpose of the component diagram can be summarized as –

- Visualize the components of a system.
- Construct executables by using forward and reverse engineering.
- Describe the organization and relationships of the components.

### Component diagrams can be used to –

- Model the components of a system.
- Model the database schema.
- Model the executables of an application.
- Model the system's source code.


<<subsystem>> WebStore
internal structure
ProductSearch
:SearchProduct
Search Database
<<subsystem>> Database Server
internal structure
:MySQL
Manage Database
<<subsystem>> Working
internal structure
OnlineShopping
:ShoppingCart
Manage Orders
:Orders
Manage Database
User Session
Manage Customers
UserSession
:Authentication
Manage Customers
:Customers
Manage Accounts
:Accounts

## 8.2 Deployment Diagram

Deployment diagram is a structure diagram which shows architecture of the system as deployment (distribution) of software artifacts to deployment targets.

Artifacts represent concrete elements in the physical world that are the result of adevelopment process. Deployment Diagram is usually represented by a node which is either hardware device or some software execution environment. Nodescould be connected through communication paths to create networked systems ofarbitrary complexity.

Note, that component was directly deployed to nodes in UML 1.x deployment diagrams. In UML 2.x artifacts are deployed to nodes, and artifacts could manifest(implement) components. Components are deployed to nodes indirectly through artifacts.

Deployment diagrams could describe architecture at specification level (alsocalled type level) or at instance level (similar to class diagrams and objectdiagrams).

Specification level deployment diagram shows some overview of deployment of artifacts to deployment targets, without referencing specific instances of artifacts or nodes.

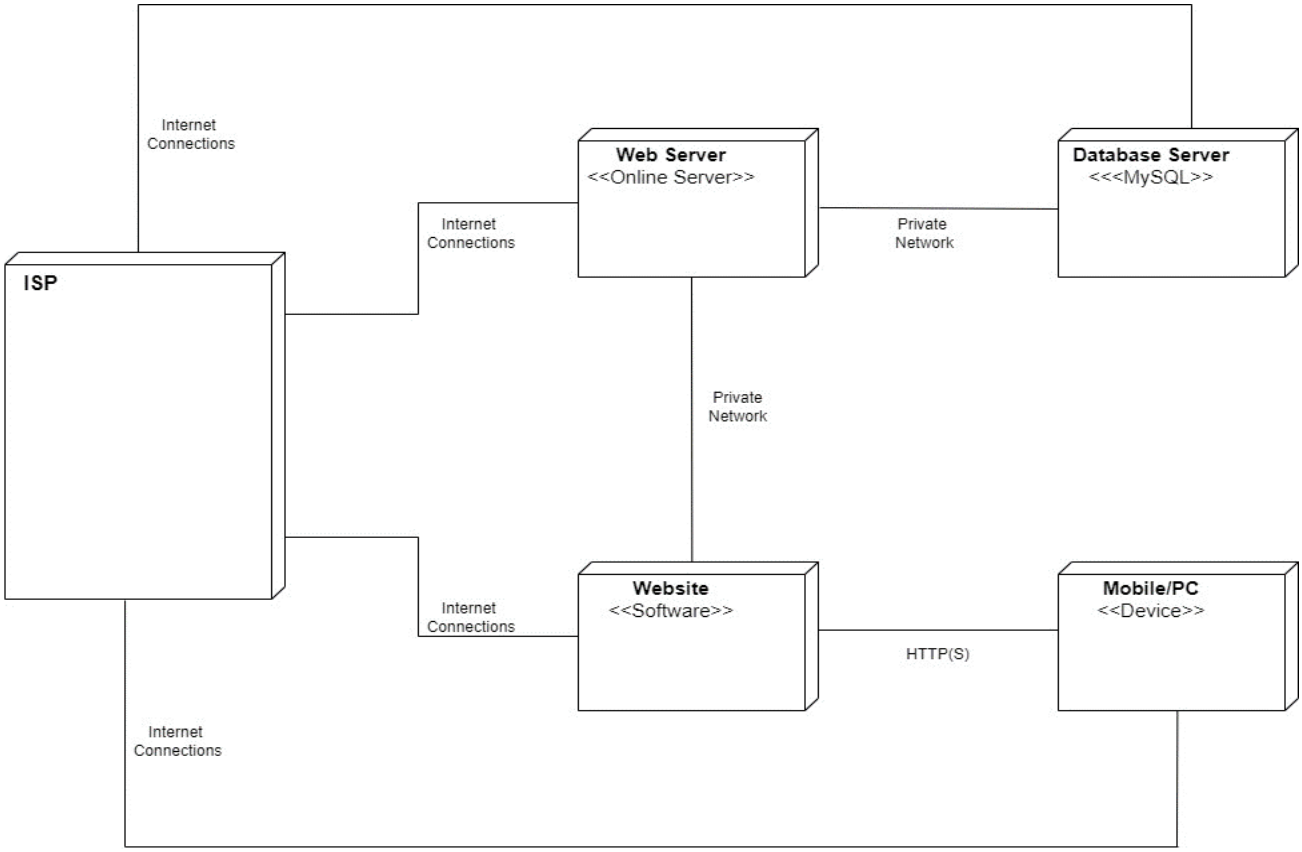
Instance level deployment diagram shows deployment of instances of artifacts to specific instances of deployment targets. It could be used for exampleto show differences in deployments to development, staging or production.

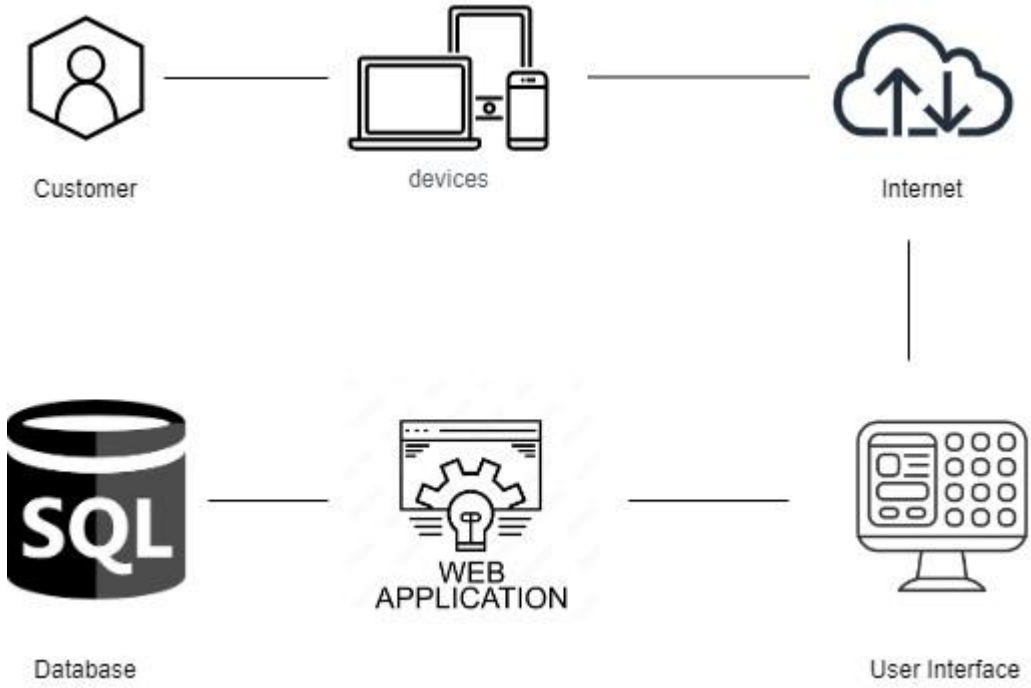
### Purpose of Deployment Diagrams

- They show the structure of the run-time system
- They capture the hardware that will be used to implement the systemand the links between different items of hardware.
- They model physical hardware elements and the communication pathsbetween them
- They can be used to plan the architecture of a system.
- They are also useful for Document the deployment of software components or nodes.

Deployment diagrams can be used –

- To model the hardware topology of a system.
- To model the embedded system.
- To model the hardware details for a client/server system.


Internet
Connections
Web Server
<<Online Server>>
Database Server
<<<MySQL>>
Internet
Connections
Private
Network
ISP
Private
Network
Website
<<Software>>
Mobile/PC
<<Device>>
Internet
Connections
HTTP(S)
Internet
Connections


Customer
devices
Internet
SQL
WEB
APPLICATION
Database
User Interface

# 9 Design Phase

## 9.1 Table Design

| Login and Signup | | | | |
|---|---|---|---|---|
| **Sr. No** | **Field Name** | **Datatype** | **Size** | **Constraints** |
| 1 | user_id | Int | 20 | Primary Key, Not Null, AUTO_INCREMENT |
| 2 | username | Varchar | 50 | Not null |
| 3 | phone | Varchar | 50 | Not null |
| 4 | email | Varchar | 50 | Not null |
| 5 | password | Varchar | 50 | Not null |
| 6 | confirm_password | Varchar | 50 | Not null |
| 7 | verification_code | Varchar | 50 | Not null |
| 8 | email_verified_at | Varchar | 50 | Not null |
| 9 | reset_token | Varchar | 50 | Not null |

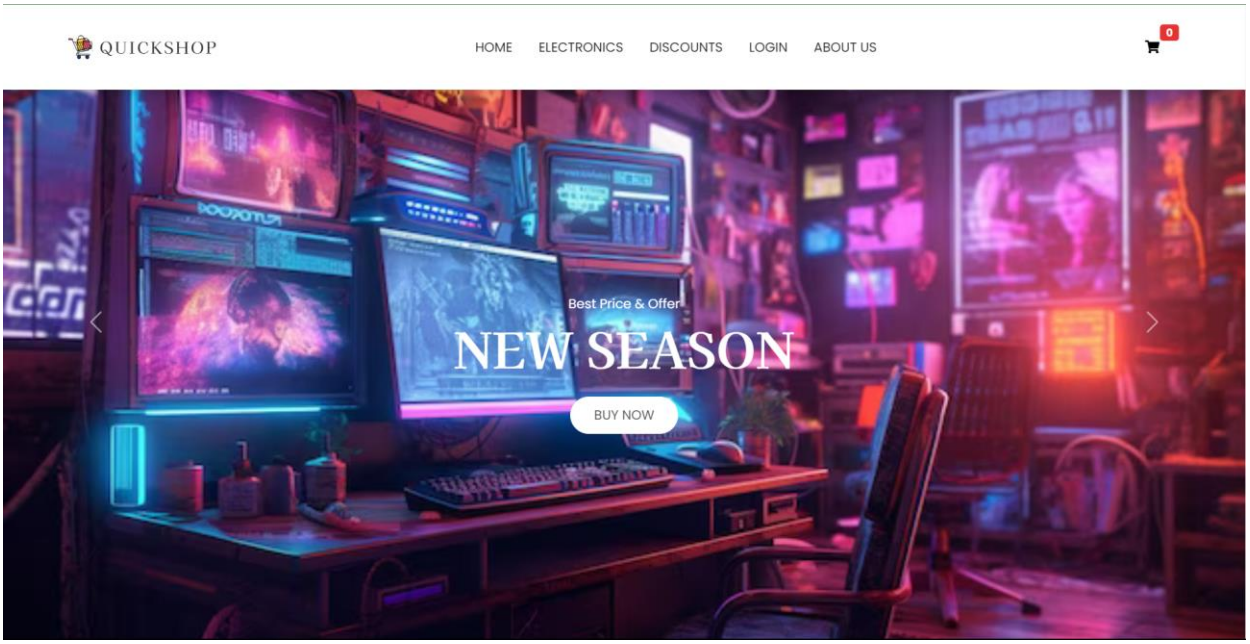
| Products | | | | |
|---|---|---|---|---|
| **Sr. No** | **Field Name** | **Datatype** | **Size** | **Constraints** |
| 1 | product_id | Int | 20 | Primary Key, Not Null, AUTO_INCREMENT |
| 2 | name | Varchar | 50 | Not null |
| 3 | price | Int | 50 | Not null |
| 4 | image | Varchar | 50 | Not null |
| 5 | description | Varchar | 50 | Not null |

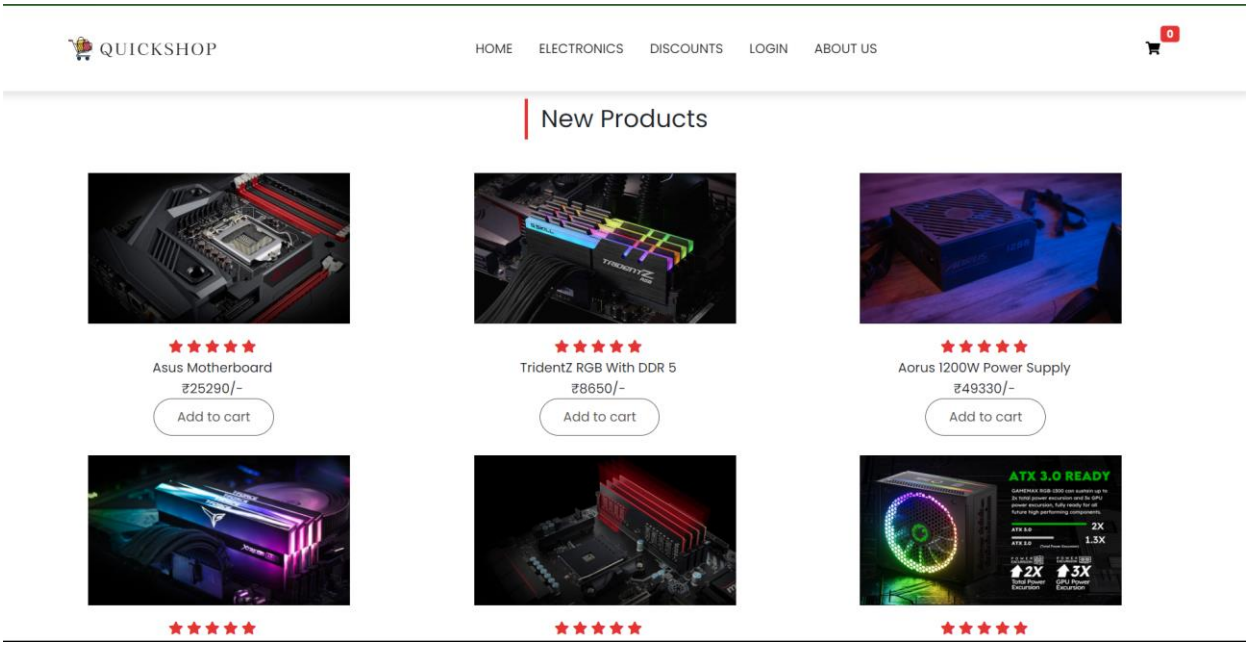
| Cart | | | | |
|---|---|---|---|---|
| **Sr. No** | **Field Name** | **Datatype** | **Size** | **Constraints** |
| 1 | cart_id | Int | 20 | Primary Key, Not Null, AUTO_INCREMENT |
| 2 | name | Varchar | 50 | Not null |
| 3 | price | Int | 20 | Not null |
| 4 | image | Varchar | 50 | Not null |
| 5 | quantity | Int | 20 | Not null |

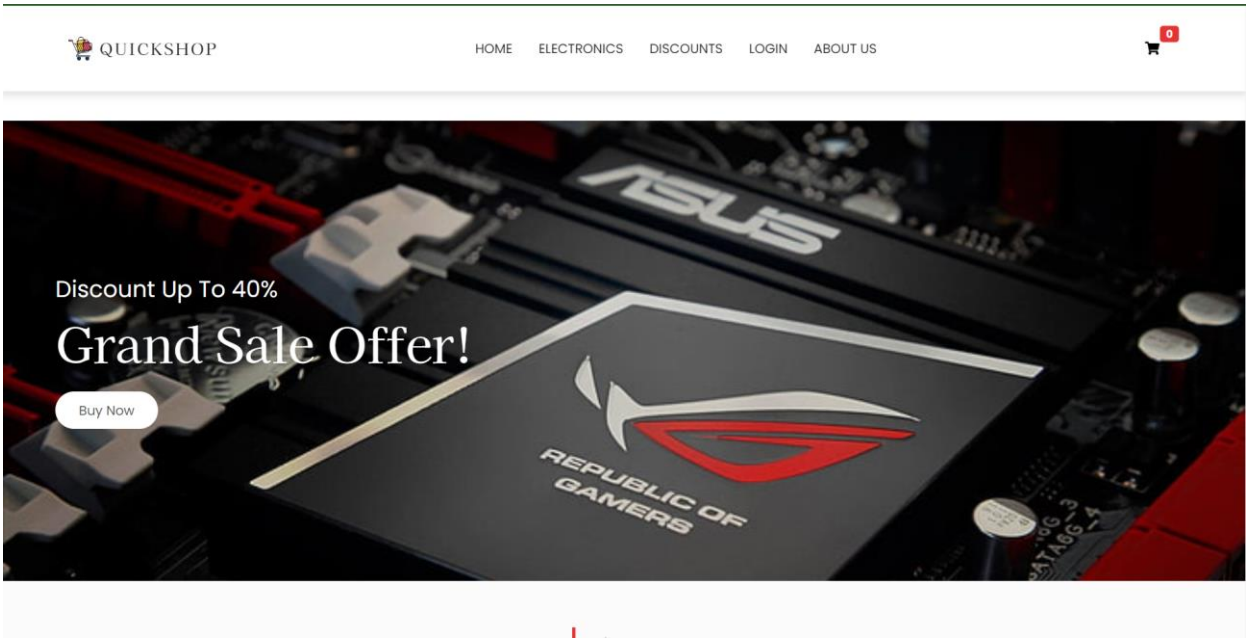
| Order | | | | |
|---|---|---|---|---|
| **Sr. No** | **Field Name** | **Datatype** | **Size** | **Constraints** |
| 1 | order_id | Int | 50 | Primary Key,<br>Not Null,<br>AUTO_INCREMENT |
| 2 | username | Varchar | 50 | Not null |
| 3 | number | Varchar | 50 | Not null |
| 4 | email | Varchar | 50 | Not null |
| 5 | method | Varchar | 50 | Not null |
| 6 | flat | Varchar | 50 | Not null |
| 7 | street | Varchar | 50 | Not null |
| 8 | city | Varchar | 50 | Not null |
| 9 | state | Varchar | 50 | Not null |
| 10 | country | Varchar | 50 | Not null |
| 11 | pin_code | Int | 10 | Not null |
| 12 | total_products | Varchar | 50 | Not null |
| 13 | total_price | Varchar | 50 | Not null |

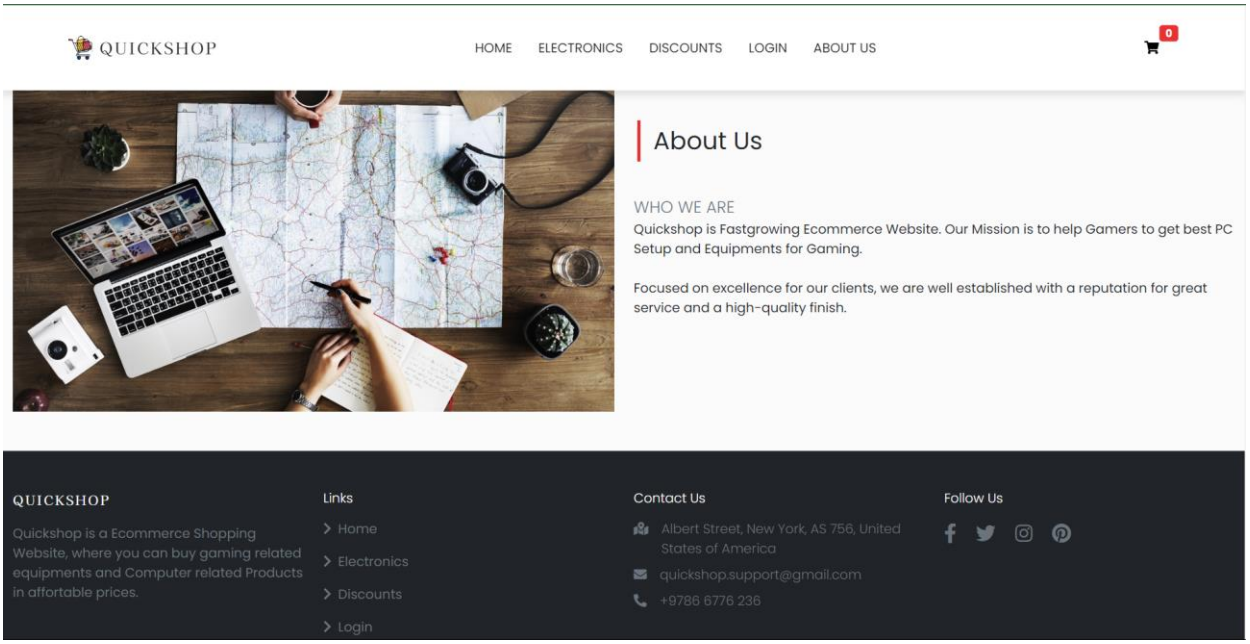
## 9.2 Screen Layout

### 1. Home Page

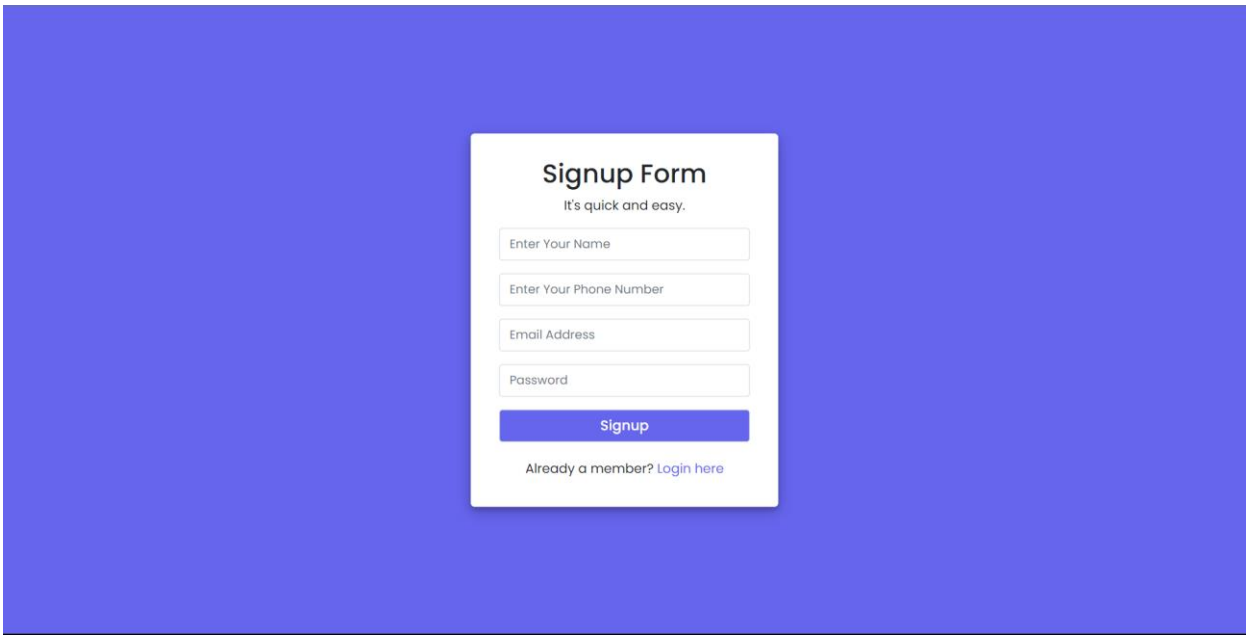


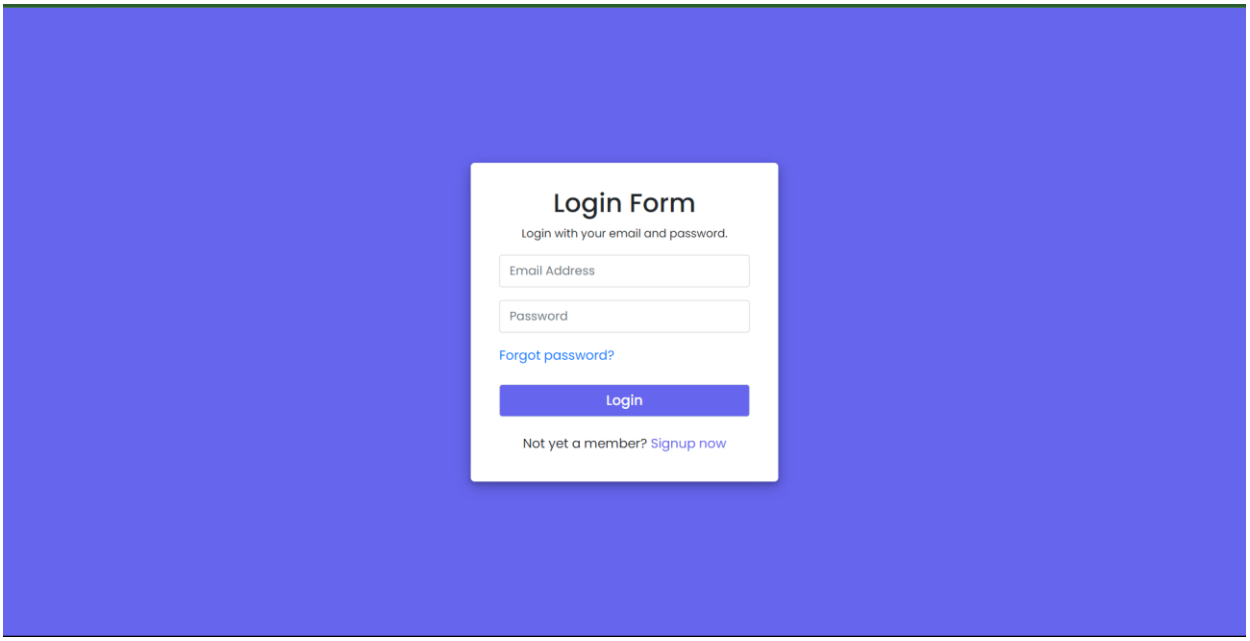

QUICKSHOP
HOME
ELECTRONICS
DISCOUNTS
LOGIN
ABOUT US
0
Discount Up To 40%
Grand Sale Offer!
Buy Now
REPUBLIC OF GAMERS


QUICKSHOP
HOME
ELECTRONICS
DISCOUNTS
LOGIN
ABOUT US
0
About Us
WHO WE ARE
Quickshop is Fastgrowing Ecommerce Website. Our Mission is to help Gamers to get best PC Setup and Equipments for Gaming.
Focused on excellence for our clients, we are well established with a reputation for great service and a high-quality finish.
QUICKSHOP
Quickshop is a Ecommerce Shopping Website, where you can buy gaming related equipments and Computer related Products in affortable prices.
Links
Home
Electronics
Discounts
Login
Contact Us
Albert Street, New York, AS 756, United States of America
quickshop.support@gmail.com
+9786 6776 236
Follow Us

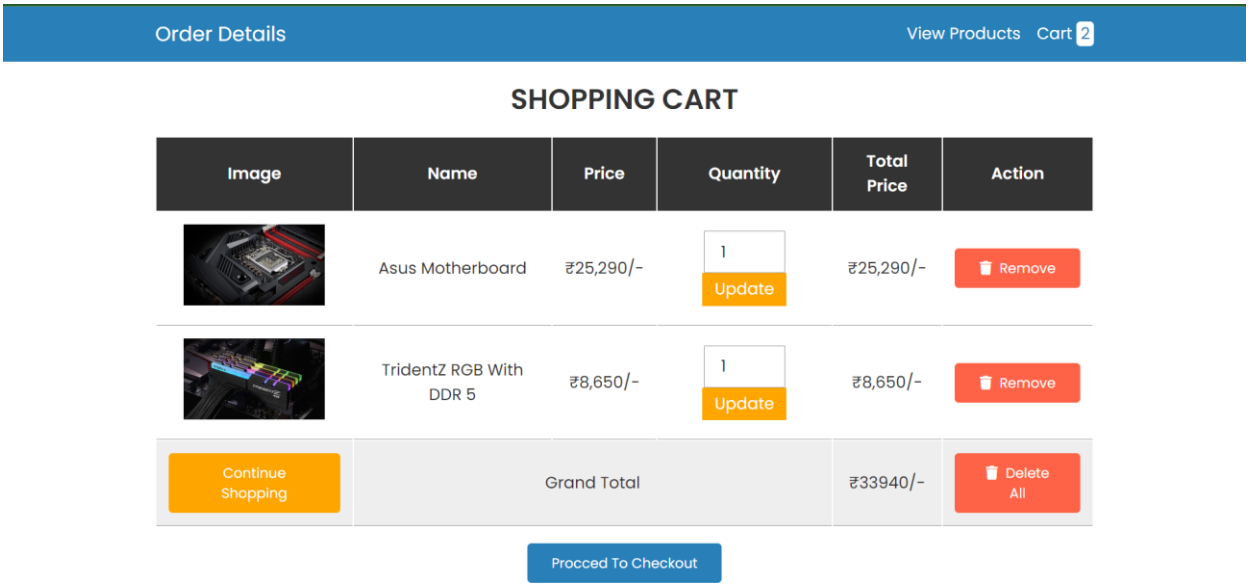
## 2. Signup Page



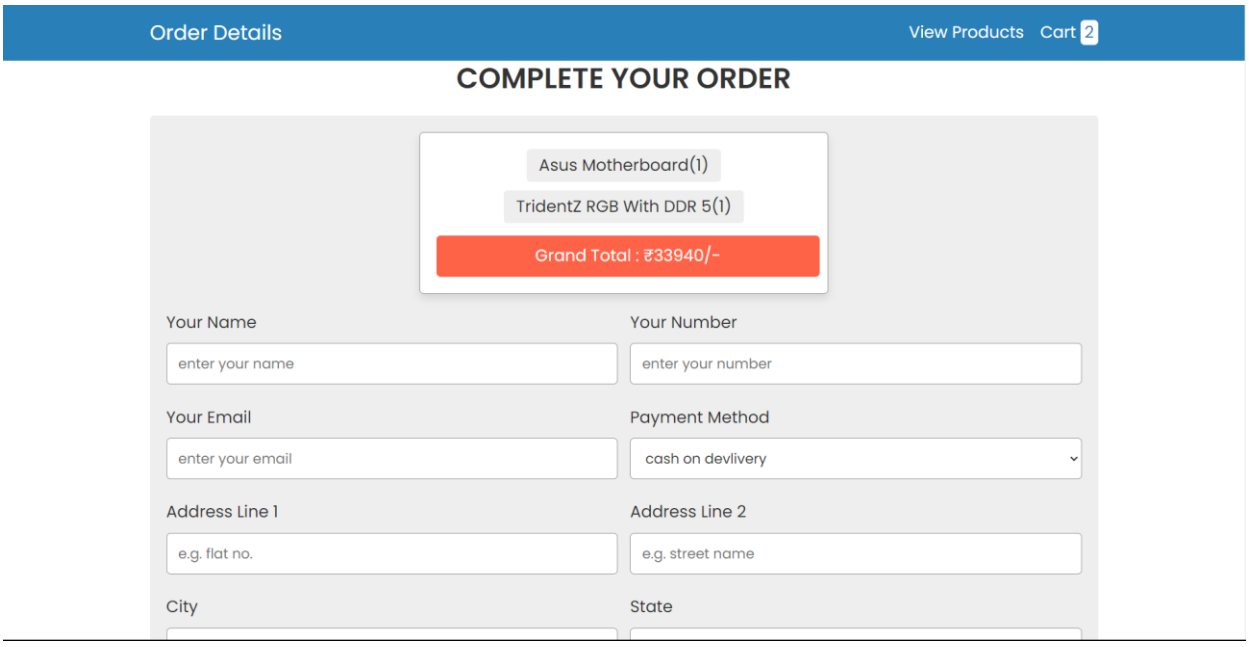

## 3. Login Page



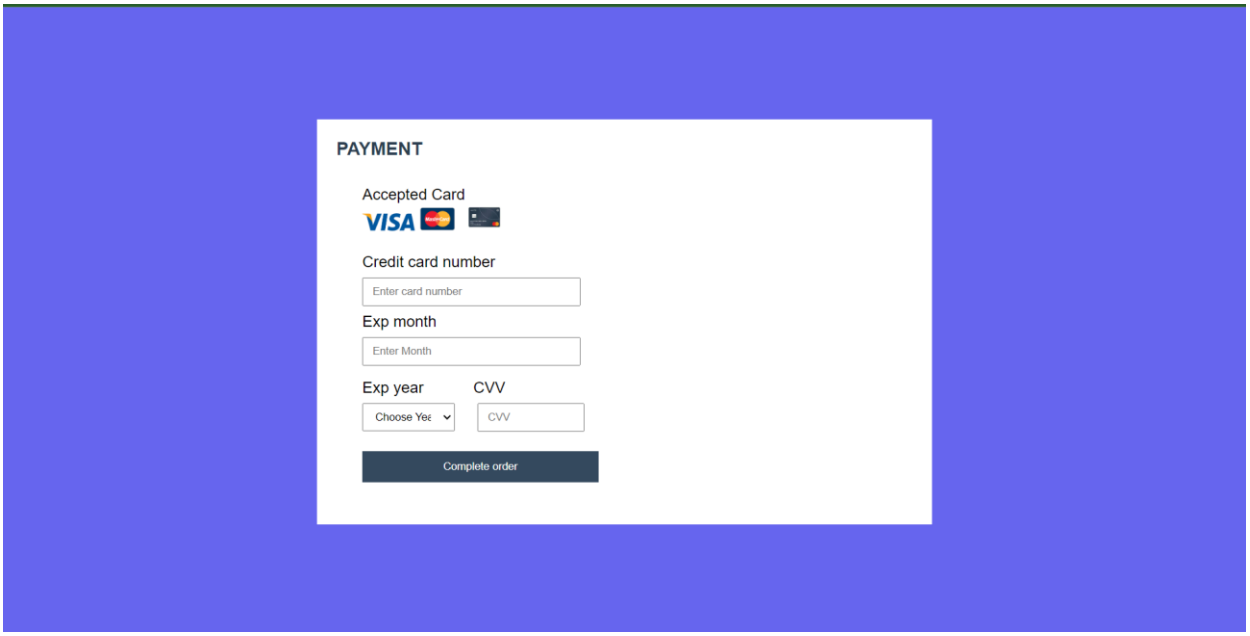
## 3. Add to Cart Page



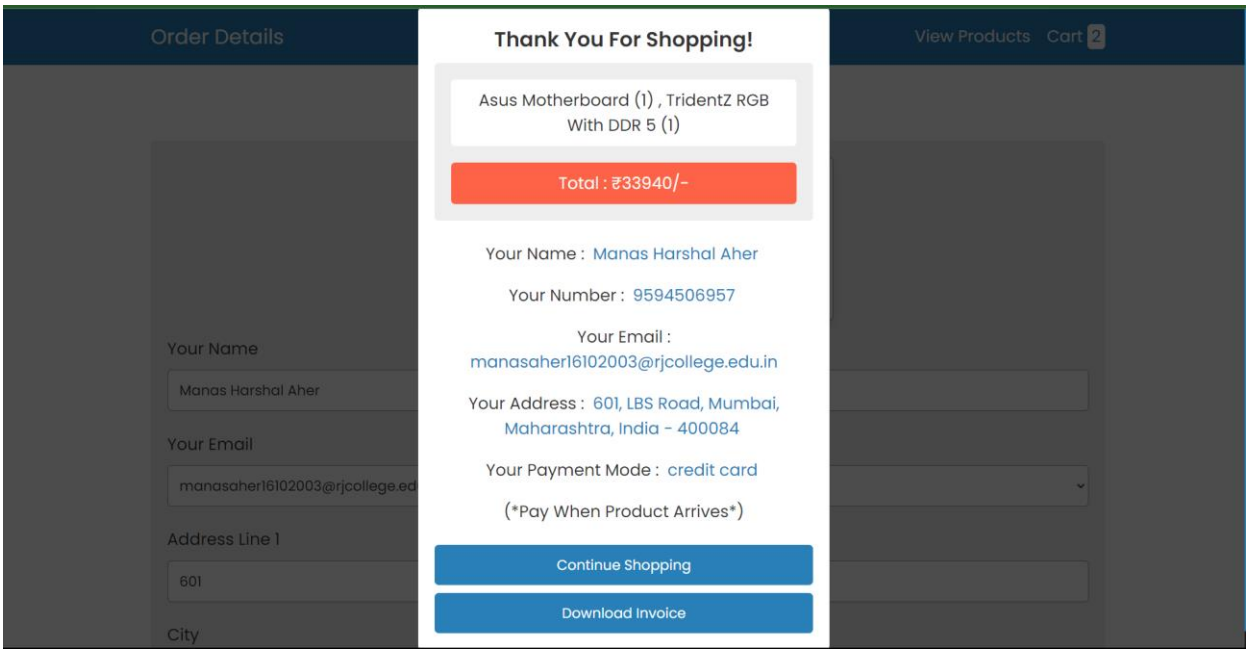

## 4. Checkout Page



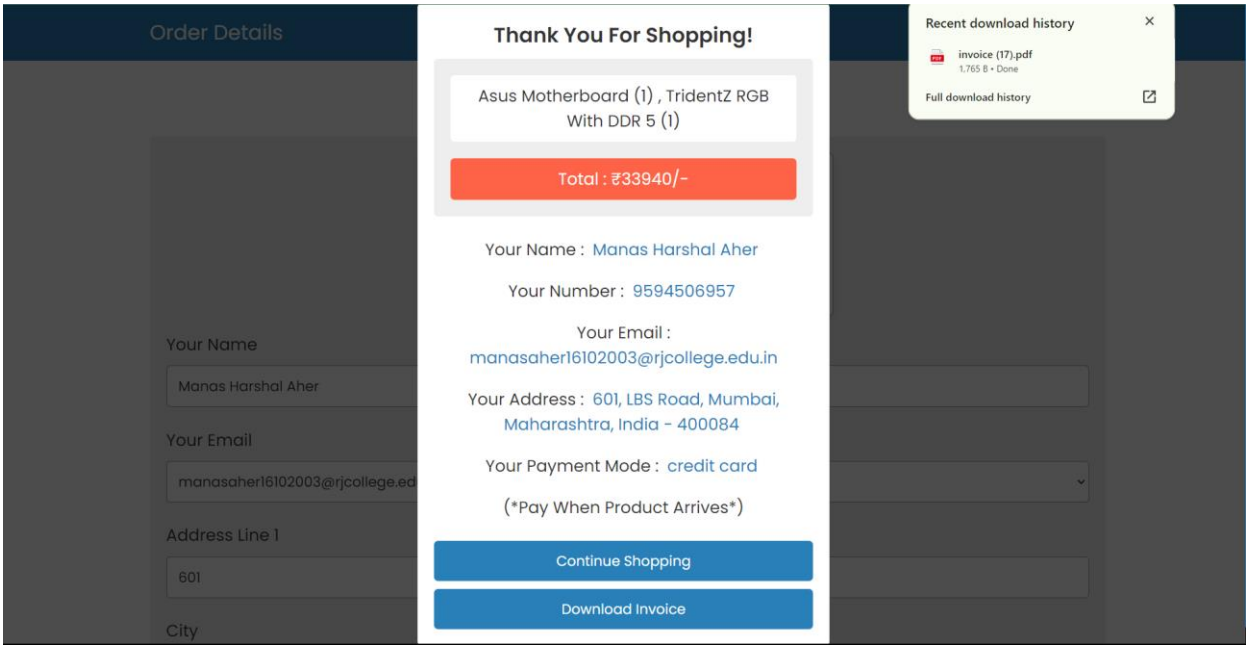
## 5. Credit Card Payment Page



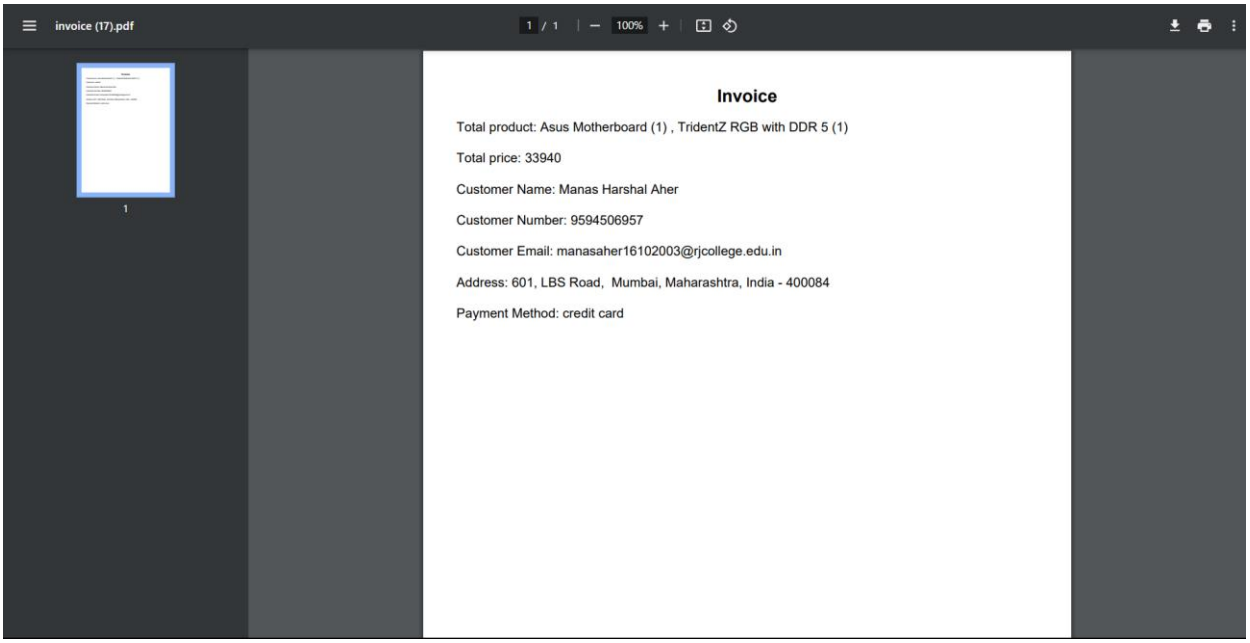

## 6. Order Detail Page



## 7. Invoice Downloaded in PDF Format




## 7. Downloaded Invoice



# 10 System Coding

**1) index.php**

```
<?php


@include 'config.php';


if(isset($_GET['add_to_cart'])){


   $product_name = $_GET['product_name'];

   $product_price = $_GET['product_price'];

   $product_image = $_GET['product_image'];

   $product_quantity = 1;


   $select_cart = mysqli_query($conn, "SELECT * FROM `cart` WHERE name =
'$product_name'");


   if(mysqli_num_rows($select_cart) > 0){

      $message[] = 'product already added to cart';

    }else{

      $insert_product = mysqli_query($conn, "INSERT INTO `cart`(name, price, image,
quantity) VALUES('$product_name', '$product_price', '$product_image', '$product_quantity')");

      $message[] = 'product added to cart succesfully';

    }

}


?>
```

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

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

    <title>Quickshop</title>

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.2.1/css/all.min.css"

    integrity="sha512-
MV7K8+y+gLIBoVD59lQIYicR65iaqukzvf/nwasF0nqhPay5w/9lJmVM2hMDcnK1OnMGCd
VK+iQrJ7lzPJQd1w=="

    crossorigin="anonymous" referrerpolicy="no-referrer" />

    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">

    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js"
integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.min.js"
integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>

<script src="https://cdn.tailwindcss.com"></script>

    <!-- fontawesome cdn -->
```

```
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css" integrity="sha512-1ycn6IcaQQ40/MKBW2W4Rhis/DbILU74C1vSrLJxCq57o941Ym01SwNsOMqvEBFlcgUa6xLiPY/NS5R+E6ztJQ==" crossorigin="anonymous" referrerpolicy="no-referrer" />

    <!-- bootstrap css -->

    <link rel = "stylesheet" href = "bootstrap-5.0.2-dist/css/bootstrap.min.css">

    <!-- custom css -->

    <link rel = "stylesheet" href = "css/main.css">


    <link rel="stylesheet" href="style1.css">

    <style>

        #dropdownMenu2{

            color:black;

        }

    </style>


</head>

<body>

<div id="productDetailsPrompt" class="product-details-prompt"></div>


    <!-- navbar -->

    <nav class = "navbar navbar-expand-lg navbar-light bg-white py-4 fixed-top">

        <div class = "container pl-16">

            <a class = "navbar-brand d-flex justify-content-between align-items-center order-lg-0" href = "index.php">

                <img src = "images/icon.png" alt = "site icon">

                <span class = "text-uppercase fw-lighter ms-2">Quickshop</span>
```

```
        </a>
        <?php

            $select_rows = mysqli_query($conn, "SELECT * FROM `cart`") or die('query failed');
            $row_count = mysqli_num_rows($select_rows);

        ?>
        <div class = "order-lg-2 nav-btns pr-16">
            <button type = "button" class = "btn position-relative">
                <a href="cart.php" style="color: black;"><i class = "fa fa-shopping-cart"></i></a>
              <span class = "position-absolute top-0 start-100 translate-middle badge bg-primary"><?php echo $row_count; ?></span>
            </button>
        </div>

        <button class = "navbar-toggler border-0" type = "button" data-bs-toggle = "collapse" data-bs-target = "#navMenu">
            <span class = "navbar-toggler-icon"></span>
        </button>

            <ul class = "navbar-nav mx-auto text-center">
                <li class = "nav-item px-2 py-2">
                    <a class = "nav-link text-uppercase text-dark" href = "#header">home</a>
                </li>
                <li class = "nav-item px-2 py-2">
```

```
                    <a class = "nav-link text-uppercase text-dark" href =
"#collection">Electronics</a>

                </li>

                <li class = "nav-item px-2 py-2">

                    <a class = "nav-link text-uppercase text-dark" href = "#offers">Discounts</a>

                </li>

                <li class = "nav-item px-2 py-2">

                    <a class = "nav-link text-uppercase text-dark" href =
"./Login/login.php">Login</a>

                </li>

                <li class = "nav-item px-2 py-2">

                    <a class = "nav-link text-uppercase text-dark" href = "#about">about us</a>

                </li>

                <li class = "nav-item px-2 py-2">



                <?php if(isset($_GET['name'])) { ?>



                    <div style="display:flex; flex-direction: row; justify-content: center; align-items:
center; width: 150px;height: 10px;margin-top: 18px;">



                <p style="width: 5px !important;"></p>

                 <h5 style="color: white;font-size: 16px;"></h5>



            <div class="dropdown">

                <button class="btn btn-secondary dropdown-toggle" type="button"
id="dropdownMenu2" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false"

                style="background-color:white;border: none ;">
```

```
<?php echo $_GET['name']; ?>
</button>
<div class="dropdown-menu" aria-labelledby="dropdownMenu2">
  <button class="dropdown-item" onclick="logout()" type="button">Logout</button>
  <!-- Inside the <head> section of your HTML -->
  <script>
      function logout() {
          // AJAX request to clear the cart
          var xhr = new XMLHttpRequest();
          xhr.open("GET", "clear_cart.php", true);
          xhr.onreadystatechange = function() {
              if (xhr.readyState == 4 && xhr.status == 200) {
                  // Redirect to the index page after clearing the cart
                  window.location.href = 'index.php';
              }
          };
          xhr.send();
      }
    </script>



</div>
</div>
     </div>
    <?php } ?>
```

```
                </li>
            </ul>

        </div>
    </nav>
    <!-- end of navbar -->

    <!-- header -->
    <header id = "header" class = "vh-100 carousel slide" data-bs-ride = "carousel" style =
"padding-top: 104px;">
        <div class = "container h-100 d-flex align-items-center carousel-inner">
            <div class = "text-center carousel-item active">
                <h2 class = "text-capitalize text-white">best Product</h2>
                <h1 class = "text-uppercase py-2 fw-bold text-white">new arrivals</h1>
                <a href = "#collection" class = "btn mt-3 text-uppercase">shop now</a>
            </div>
            <div class = "text-center carousel-item">
                <h2 class = "text-capitalize text-white">best price & offer</h2>
                <h1 class = "text-uppercase py-2 fw-bold text-white">new season</h1>
                <a href = "#collection" class = "btn mt-3 text-uppercase">buy now</a>
            </div>
        </div>
```

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

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

        </button>

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

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

        </button>

    </header>

    <!-- end of header -->


</br>

</br>

    <!-- collection -->

    <section id = "collection" class = "py-5">

        <div class = "container">

            <div class = "title text-center">

                <h2 class = "position-relative d-inline-block text-3xl">New Products</h2>

            </div>


            <?php include 'header1.php'; ?>


            <div class="container">


            <section class="products">

</br>
```

```
</br>


        <div class="box-container grid grid-cols-3 gap-4 pl-20">

            <?php

            $select_products = mysqli_query($conn, "SELECT * FROM `products`");

            if(mysqli_num_rows($select_products) > 0){

               while($fetch_product = mysqli_fetch_assoc($select_products)){

                  $id=$fetch_product["id"];
            ?>


               <div class="box pr-16+-" id="myProducts" >


                        <a href="product_detail.php?id=<?=$id?>"><img src="images/<?php echo $fetch_product['image']; ?>" class="w-80" alt="">

               </a>


                     <div class = "text-center pr-36">

                        <div class = "rating mt-3">

                           <span class = "text-primary"><i class = "fas fa-star"></i></span>
```

```
                        <span class = "text-primary"><i class = "fas fa-star"></i></span>
                        <span class = "text-primary"><i class = "fas fa-star"></i></span>
                        <span class = "text-primary"><i class = "fas fa-star"></i></span>
                        <span class = "text-primary"><i class = "fas fa-star"></i></span>
                    </div>
                    <h4 class="text-capitalize my-1"><?php echo $fetch_product['name'];
?></h4>
                    <div class="price" class = "fw-bold">₹<?php echo $fetch_product['price'];
?>/-</div>
                    </div>
                    <form action="./index.php" method="get">
                        <input type="hidden" name="product_name"  value="<?php echo
$fetch_product['name']; ?>">
                        <input type="hidden" name="product_price" value="<?php echo
$fetch_product['price']; ?>">
                        <input type="hidden" name="product_image"  value="<?php echo
$fetch_product['image']; ?>">
                        <input type="submit" class="btn ml-20" value="Add to cart"
name="add_to_cart">
            </br>
                    </form>


            </div>


        <?php
            };
```

```
            };
            ?>


        </div>


        </section>
      </div>
   </section>
</div>
   <!-- end of collection -->


   <!-- special products -->
   <section id = "special" class = "py-5">
      <div class = "container">
         <div class = "title text-center py-5">
            <h2 class = "position-relative d-inline-block text-3xl">CPU Cases</h2>
         </div>
         <?php include 'header1.php'; ?>


         <div class="container">


         <section class="products">
</br>
</br>
```

```
<div class="box-container grid grid-cols-3 gap-4 pl-20">

   <?php

   $select_products = mysqli_query($conn, "SELECT * FROM `product`");

   if(mysqli_num_rows($select_products) > 0){

      while($fetch_product = mysqli_fetch_assoc($select_products)){
   ?>


      <div class="box">


                  <img src="images/<?php echo $fetch_product['image']; ?>" class="w-80"
alt="">


               <div class = "text-center pr-36">
                  <div class = "rating mt-3">
                     <span class = "text-primary"><i class = "fas fa-star"></i></span>
                     <span class = "text-primary"><i class = "fas fa-star"></i></span>
                     <span class = "text-primary"><i class = "fas fa-star"></i></span>
                     <span class = "text-primary"><i class = "fas fa-star"></i></span>
```

```
                        <span class = "text-primary"><i class = "fas fa-star"></i></span>

                    </div>

                    <h4 class="text-capitalize my-1"><?php echo $fetch_product['name']; ?></h4>

                    <div class="price" class = "fw-bold">₹<?php echo $fetch_product['price']; ?>/-</div>

                    </div>

                    <form action="./index.php" method="get">

                        <input type="hidden" name="product_name"  value="<?php echo $fetch_product['name']; ?>">

                        <input type="hidden" name="product_price" value="<?php echo $fetch_product['price']; ?>">

                        <input type="hidden" name="product_image"  value="<?php echo $fetch_product['image']; ?>">

                        <input type="submit" class="btn ml-20" value="Add to cart" name="add_to_cart">

            </br>

                    </form>




            </div>



        <?php

            };

        };

        ?>
```

```
            </div>


            </section>


        </div>
        <script>
        function productDetails(a){


        }
        </script>
    </section>
    <!-- end of special products -->


    <!-- blogs -->
    <section id = "offers" class = "py-5">
        <div class = "container">
            <div class = "row d-flex align-items-center justify-content-center text-center justify-content-lg-start text-lg-start">
                <div class = "offers-content pl-16">
                    <span class = "text-white">Discount Up To 40%</span>
                    <h2 class = "mt-2 mb-4 text-white">Grand Sale Offer!</h2>
                    <a href = "#" class = "btn">Buy Now</a>
                </div>
            </div>
        </div>
    </section>
```

```
<!-- end of blogs -->


<!--Sign up-->
<section id = "newsletter" class = "py-5">
    <div class = "container">
        <div class = "d-flex flex-column align-items-center justify-content-center">
            <div class = "title text-center pt-3 pb-5">
                <h2 class = "position-relative d-inline-block ms-4 text-3xl">Sign up</h2>
            </div>

            <p class = "text-center text-muted">Get E-mail updates about our latest shop and special offers.</p>
            <div class = "input-group mb-3 mt-3">
                <input type = "text" class = "form-control" placeholder="Enter Your Email ...">
                <button class = "btn btn-primary" type = "submit">Sign up</button>
            </div>
        </div>
    </div>
</section>
<!--end of sign up-->


<!-- about us -->
<section id = "about" class = "py-5">
    <div class = "container">
        <div class = "row gy-lg-5 align-items-center">
            <div class = "col-lg-6 order-lg-1 text-center text-lg-start">
```

```
                <div class = "title pt-3 pb-5">

                    <h2 class = "position-relative d-inline-block ms-4 text-3xl">About Us</h2>

                </div>

                <p class = "lead text-muted">WHO WE ARE</p>

                <p>Quickshop is Fastgrowing Ecommerce Website. Our Mission is to help Gamers to get best PC Setup and Equipments for Gaming.</p>

</br>

                <p>Focused on excellence for our clients, we are well established with a reputation for great service and a high-quality finish.</p>

            </div>

            <div class = "col-lg-6 order-lg-0">

                <img src = "images/about_us1.jpg" alt = "" class = "img-fluid">

            </div>

        </div>

    </div>

</section>

<!-- end of about us -->




<!-- footer -->

<footer class = "bg-dark py-5">

    <div class = "container">

        <div class = "row text-white g-4">

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

                <a class = "text-uppercase text-decoration-none brand text-white" href = "index.php">Quickshop</a>
```

```
            <p class = "text-white text-muted mt-3">Quickshop is a Ecommerce Shopping Website, where you can buy gaming related equipments and Computer related Products in affortable prices.</p>

        </div>


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

            <h5 class = "fw-light">Links</h5>

            <ul class = "list-unstyled">

                <li class = "my-3">

                    <a href = "#" class = "text-white text-decoration-none text-muted">

                        <i class = "fas fa-chevron-right me-1"></i> Home

                    </a>

                </li>

                <li class = "my-3">

                    <a href = "#collections" class = "text-white text-decoration-none text-muted">

                        <i class = "fas fa-chevron-right me-1"></i> Electronics

                    </a>

                </li>

                <li class = "my-3">

                    <a href = "#offers" class = "text-white text-decoration-none text-muted">

                        <i class = "fas fa-chevron-right me-1"></i> Discounts

                    </a>

                </li>

                <li class = "my-3">

                    <a href = "./Login/login.php" class = "text-white text-decoration-none text-muted">

                        <i class = "fas fa-chevron-right me-1"></i> Login
```

```
            </a>
        </li>
        <li class = "my-3">
            <a href = "#about" class = "text-white text-decoration-none text-muted">
                <i class = "fas fa-chevron-right me-1"></i> About Us
            </a>
        </li>
    </ul>
</div>

<div class = "col-md-6 col-lg-3">
    <h5 class = "fw-light mb-3">Contact Us</h5>
    <div class = "d-flex justify-content-start align-items-start my-2 text-muted">
        <span class = "me-3">
            <i class = "fas fa-map-marked-alt"></i>
        </span>
        <span class = "fw-light">
            Albert Street, New York, AS 756, United States of America
        </span>
    </div>
    <div class = "d-flex justify-content-start align-items-start my-2 text-muted">
        <span class = "me-3">
            <i class = "fas fa-envelope"></i>
        </span>
        <span class = "fw-light">
```

```
                quickshop.support@gmail.com
            </span>
        </div>
        <div class = "d-flex justify-content-start align-items-start my-2 text-muted">
            <span class = "me-3">
                <i class = "fas fa-phone-alt"></i>
            </span>
            <span class = "fw-light">
                +9786 6776 236
            </span>
        </div>
    </div>

    <div class = "col-md-6 col-lg-3">
        <h5 class = "fw-light mb-3">Follow Us</h5>
        <div>
            <ul class = "list-unstyled d-flex">
                <li>
                    <a href = "#" class = "text-white text-decoration-none text-muted fs-4 me-
4">
                        <i class = "fab fa-facebook-f"></i>
                    </a>
                </li>
                <li>
                    <a href = "#" class = "text-white text-decoration-none text-muted fs-4 me-
4">
```

```
                        <i class = "fab fa-twitter"></i>
                    </a>
                </li>
                <li>
                    <a href = "#" class = "text-white text-decoration-none text-muted fs-4 me-
4">
                        <i class = "fab fa-instagram"></i>
                    </a>
                </li>
                <li>
                    <a href = "#" class = "text-white text-decoration-none text-muted fs-4 me-
4">
                        <i class = "fab fa-pinterest"></i>
                    </a>
                </li>
            </ul>
        </div>
    </div>
</div>
</div>
</footer>
<!-- end of footer -->
</body>
</html>
```

**2) login.php**

```
<?php

if(isset($_POST["login"]))
{
   $email = $_POST["email"];
   $password = $_POST["password"];

   $conn = mysqli_connect("localhost", "root", "", "register");

   $sql = "SELECT * FROM users WHERE email = '" . $email . "'";
   $result = mysqli_query($conn, $sql);

   if(mysqli_num_rows($result) == 0)
   {
      header("Location: login.php?error=Email not found");
      exit();
   }

   $user = mysqli_fetch_object($result);

   if(!password_verify($password, $user->password))
   {
      header("Location: login.php?error=Password is incorrect");
      exit();
   }

   if($user->email_verified_at == null)
   {
      header("Location: login.php?error=Please verify your email <a href='email-
verification.php?email=" . $email ."'>form here</a>");
      exit();
   }
   header("Location: http://localhost:8080/Quickshop/index.php?name=".$email."");
   exit();
}
?>
<?php if(isset($_GET['error'])) { ?>
                  <p class="error">
                     <?php echo $_GET['error']; ?>
                  </p>
               <?php } ?>
               <?php if(isset($_GET['success'])) { ?>
                  <p class="success">
                     <?php echo $_GET['success']; ?>
                  </p>
               <?php } ?>
```

**3) Signup.php**

```
<?php

use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;
use PHPMailer\PHPMailer\SMTP;

require './vendor/autoload.php';

if(isset($_POST["register"]))
{
   $name = $_POST["name"];
   $phone = $_POST["phone"];
   $email = $_POST["email"];
   $password = $_POST["password"];

   $mail = new PHPMailer(true);

   try{
      $mail->SMTPDebug = 0;

      $mail->isSMTP();

      $mail->Host = 'smtp.gmail.com';

      $mail->SMTPAuth = true;

      $mail->Username = 'manasaher123';

      $mail->Password = 'cslfmracbjrrqmxb';

      $mail->SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS;

      $mail->SMTPSecure = 'tls';

      $mail->Port = 587;
      $mail->setFrom('manasaher123@gmail.com','Manas Aher');

      $mail->addAddress($email,$name);
      $mail-> addReplyTo('manasaher123@gmail.com',"Manas");

      $mail->isHTML(true);

      $verification_code = substr(number_format(time() * rand(), 0, '', ''), 0, 6);

      $mail->Subject = 'Email verification';
      $mail->Body = '<p>Your verification code is: <b style="font: size 30px;">' .
$verification_code . '</b></p>';
```

```
        $mail->send();

        $encrypted_password = password_hash($password, PASSWORD_DEFAULT);

        $conn =mysqli_connect("localhost", "root", "", "register");
        $email_verified_at=date("l jS \of F Y h:i:s A");
        $sql = "INSERT INTO users(name, phone, email, password, verification_code,
email_verified_at) VALUES ('" . $name . "','" . $phone . "','" . $email . "','" .
$encrypted_password . "','" . $verification_code . "','".$email_verified_at."')";
        mysqli_query($conn, $sql);

        header("Location: email-verification.php?email=" . $email);
        exit();
    }
    catch(Exception $e)
    {
        echo "Message could not be sent. Mailer Error: {$mail->ErrorInfo}";
    }
}
?>
<?php if(isset($_GET['error'])) { ?>
                    <p class="error">
                        <?php echo $_GET['error']; ?>
                    </p>
                <?php } ?>
                <?php if(isset($_GET['success'])) { ?>
                    <p class="success">
                        <?php echo $_GET['success']; ?>
                    </p>
                <?php } ?>
<script>
        document.getElementById("numberInput").addEventListener("input", function() {
            var input = this.value.replace(/\D/g, ''); // Remove non-digit characters
            if (input.length > 10) {
                input = input.slice(0, 10); // Limit to 10 characters
            }
            this.value = input;
        });
    </script>
```

**4) forgotpassword.php**

```
<?php

use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\SMTP;
use PHPMailer\PHPMailer\Exception;

require './vendor/autoload.php';

if (isset($_POST['submit-btn'])) {

   $message = array();

   $conn = mysqli_connect("localhost", "root", "", "register");

   $email = mysqli_real_escape_string($conn,$_POST['email']);

   // Generate a unique token
   $token = substr(bin2hex(random_bytes(32)),0,32);

   // Update the user's reset_token in the database
   $update_query = mysqli_query($conn, "UPDATE `users` SET `reset_token`='$token'
WHERE `email`='$email'");
   if ($update_query) {
      // Send email with reset link
      $mail = new PHPMailer(true);

      try {
         //Server settings
         $mail->isSMTP();
         $mail->Host       = 'smtp.gmail.com';
         $mail->SMTPAuth   = true;
         $mail->Username   = 'manasaher123@gmail.com'; // Your Gmail email address
         $mail->Password   = 'cslfmracbjrrqmxb'; // Your app-specific password
         $mail->SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS;
         $mail->Port       = 587;

         //Recipients
         $mail->setFrom('manasaher123@gmail.com', 'Manas'); // Sender's email address and
name
         $mail->addAddress($email); // Recipient's email address

         //Content
         $mail->isHTML(true);
         $mail->Subject = 'Reset Your Password';
         $mail->Body    = "To reset your password, please click the following link: <a
href='http://localhost:8080/Quickshop/Login/resetpassword.php?token=$token'>Reset
Password</a>";

         $mail->send();
```

```
            $message[] = "Password reset link has been sent to your email.";
        } catch (Exception $e) {
            $message[] = "Message could not be sent. Mailer Error: {$mail->ErrorInfo}";
        }
    } else {
        $message[] = "Failed to generate reset link. Please try again later.";
    }
}
?>
<?php
                    if (isset($message)) {
                        foreach ($message as $msg) {
                            echo '<div>'.$msg.'</div>';
                        }
                    }
  ?>
```

**5) resetpassword.php**

```
<?php

if (isset($_GET['token'])) {

   $conn = mysqli_connect("localhost", "root", "", "register");
   $token_1=substr($_GET["token"],0,32);

   $token = mysqli_real_escape_string($conn, $token_1);

   // Check if the token exists in the database
   $token_check_query = mysqli_query($conn, "SELECT * FROM `users` WHERE
`reset_token`='$token'");
   if (mysqli_num_rows($token_check_query) == 1) {
      // Token is valid, allow the user to reset the password
      // Display a form to set a new password
      if (isset($_POST['submit-btn'])) {
         $password = mysqli_real_escape_string($conn, $_POST['password']);
         $confirm_password = mysqli_real_escape_string($conn, $_POST['confirm_password']);

         if ($password === $confirm_password) {
            // Hash the new password
            $hashed_password = password_hash($password, PASSWORD_BCRYPT);
            // Update the user's password in the database
            $update_query = mysqli_query($conn, "UPDATE `users` SET
`password`='$hashed_password', `reset_token`=NULL WHERE `reset_token`='$token'");
            if ($update_query) {
               $message = "Your password has been successfully reset.";
            } else {
               $error = "Failed to reset password. Please try again.";
            }
         } else {
            $error = "Passwords do not match.";
         }
      }
   } else {
      $error = "Invalid or expired token.";
   }
} else {
   $error = "Token not provided.";
}
?>
<?php
                  if (isset($error)) {
                     echo '<div>'.$error.'</div>';
                  }
                  if (isset($message)) {
                     echo '<div>'.$message.'</div>';
                  }
               ?>
```

**6) email-verification.php**

```
<?php

if(isset($_POST['verify_email'])){
   $email = $_POST["email"];
   $verification_code=$_POST["verification_code"];

   $conn = mysqli_connect("localhost", "root", "", "register");

   $sql = "UPDATE users SET email_verified_at = NOW() WHERE email = '" . $email . "'
AND verification_code = '" . $verification_code ."'";
   $result = mysqli_query($conn, $sql);

   if(mysqli_affected_rows($conn) == 0){
      header("Location: email-verification.php?error=verification code is invalid");

      exit();
   }

   header("Location: login.php?success=Your email has been verified successfully");
   exit();
}

?>
<?php
                if(isset($_GET["error"])){
                   echo $_GET["error"];
                }
                ?>
```

**7)checkout.php**

```
<?php

@include 'config.php';

if(isset($_POST['order_btn'])){

   $name = $_POST['name'];
   $number = $_POST['number'];
   $email = $_POST['email'];
   $method = $_POST['method'];
   $flat = $_POST['flat'];
   $street = $_POST['street'];
   $city = $_POST['city'];
   $state = $_POST['state'];
   $country = $_POST['country'];
   $pin_code = $_POST['pin_code'];

   $cart_query = mysqli_query($conn, "SELECT * FROM `cart`");
   $price_total = 0;
   $product_price = 0;
   if(mysqli_num_rows($cart_query) > 0){
      while($product_item = mysqli_fetch_assoc($cart_query)){
         $product_name[] = $product_item['name'] .' (. $product_item['quantity'] .') ';
         $product_price = intval($product_item['price']) * intval($product_item['quantity']);
         $product_price_formatted = number_format($product_price);
         $price_total += $product_price;
      };
   };

   $total_product = implode(', ',$product_name);
   $detail_query = mysqli_query($conn, "INSERT INTO `order`(name, number, email, method,
flat, street, city, state, country, pin_code, total_products, total_price)
VALUES('$name','$number','$email','$method','$flat','$street','$city','$state','$country','$pin_cod
e','$total_product','$price_total')") or die('query failed');

   if($cart_query && $detail_query){
      echo "
      <div class='order-message-container'>
      <div class='message-container'>
         <h3>thank you for shopping!</h3>
         <div class='order-detail'>
            <span>".$total_product."</span>
            <span class='total'> total :  ₹".$price_total."/-  </span>
         </div>
         <div class='customer-details'>
            <p> your name : <span>".$name."</span> </p>
            <p> your number : <span>".$number."</span> </p>
            <p> your email : <span>".$email."</span> </p>
```

```
            <p> your address : <span>".$flat.", ".$street.", ".$city.", ".$state.", ".$country." -
".$pin_code."</span> </p>
            <p> your payment mode : <span>".$method."</span> </p>
            <p>(*pay when product arrives*)</p>
          </div>
            <a href='index.php' class='btn'>continue shopping</a>
            <button id='download_invoice2' class='btn'>Download Invoice</button>
          </div>
        </div>
        ";
        }
      }

?>
<?php include 'header.php'; ?>

<div class="container">

<section class="checkout-form">

  <h1 class="heading">complete your order</h1>

  <form id="checkout_form" action="" method="post">

  <div class="display-order">
    <?php
      $select_cart = mysqli_query($conn, "SELECT * FROM `cart`");
      $total = 0;
      $grand_total = 0;
      if(mysqli_num_rows($select_cart) > 0){
        while($fetch_cart = mysqli_fetch_assoc($select_cart)){
        $total_price = $fetch_cart['price'] * $fetch_cart['quantity'];
        $grand_total = $total += $total_price;
    ?>
    <span><?= $fetch_cart['name']; ?>(<?= $fetch_cart['quantity']; ?>)</span>
    <?php
      }
    }else{
      echo "<div class='display-order'><span>your cart is empty!</span></div>";
    }
    ?>
    <span class="grand-total"> grand total : ₹<?= $grand_total; ?>/- </span>
  </div>

    <div class="flex">
      <div class="inputBox">
        <span>your name</span>
        <input type="text" placeholder="enter your name" name="name" id="name"
value="<?php echo isset($name) ? $name : ''; ?>">
      </div>
      <div class="inputBox">
```

```
          <span>your number</span>
          <input type="number" placeholder="enter your number" name="number" value="<?php
echo isset($number) ? $number : ''; ?>" required>
        </div>
        <div class="inputBox">
          <span>your email</span>
          <input type="email" placeholder="enter your email" name="email" value="<?php echo
isset($email) ? $email : ''; ?>" required>
        </div>
        <div class="inputBox">
          <span>payment method</span>
          <select name="method" onchange="redirectToCheckout(this)">
            <option value="cash on delivery" selected>cash on devlivery</option>
            <option value="checkout2.php" >credit card</option>
          </select>
        </div>
        <div class="inputBox">
          <span>address line 1</span>
          <input type="text" placeholder="e.g. flat no." name="flat" value="<?php echo
isset($flat) ? $flat : ''; ?>" required>
        </div>
        <div class="inputBox">
          <span>address line 2</span>
          <input type="text" placeholder="e.g. street name" name="street" value="<?php echo
isset($street) ? $street : ''; ?>" required>
        </div>
        <div class="inputBox">
          <span>city</span>
          <input type="text" placeholder="e.g. mumbai" name="city" value="<?php echo
isset($city) ? $city : ''; ?>" required>
        </div>
        <div class="inputBox">
          <span>state</span>
          <input type="text" placeholder="e.g. maharashtra" name="state" value="<?php echo
isset($state) ? $state : ''; ?>" required>
        </div>
        <div class="inputBox">
          <span>country</span>
          <input type="text" placeholder="e.g. india" name="country" value="<?php echo
isset($country) ? $country : ''; ?>" required>
        </div>
        <div class="inputBox">
          <span>pin code</span>
          <input type="text" placeholder="e.g. 123456" name="pin_code" value="<?php echo
isset($pin_code) ? $pin_code : ''; ?>" required>
        </div>
      </div>
      <input type="hidden" name="product" value="<?php echo $total_product; ?>">
      <input type="hidden" name="price" value="<?php echo $price_total; ?>">
      <input type="submit" value="order now" name="order_btn" class="btn">
```

```
  </form>

</section>

<script>
    function redirectToCheckout(element) {
        var selectedOption = element.value;
        if (selectedOption === "checkout2.php") {
            window.location.href = selectedOption;
        }
    }
</script>

<script>
document.getElementById("numberInput").addEventListener("input", function() {
    var input = this.value.replace(/\D/g, ''); // Remove non-digit characters
    if (input.length > 10) {
        input = input.slice(0, 10); // Limit to 10 characters
    }
    this.value = input;
});
</script>

<script>
document.getElementById('download_invoice2').addEventListener('click', function() {
    var form = document.getElementById('checkout_form');
    var formData = new FormData(form);

    // Iterate through all form elements
    var formElements = form.elements;
    console.log(formElements);
    for (var i = 0; i < formElements.length; i++) {
        var element = formElements[i];

            formData.append(element.name, element.value);
            console.log(element.value);

    }

    var xhr = new XMLHttpRequest();
    xhr.open('POST', './generate_invoice2.php', true);
    xhr.responseType = 'blob'; // Set response type to blob for downloading files
    xhr.onreadystatechange = function() {
        if (xhr.readyState === XMLHttpRequest.DONE) {
            if (xhr.status === 200) {
                // Create a blob URL from the response
                var blob = new Blob([xhr.response], { type: 'application/pdf' });
                var url = window.URL.createObjectURL(blob);

                // Create a link element and trigger the download
                var a = document.createElement('a');
```

```
            a.href = url;
            a.download = 'invoicedownload.pdf';
            document.body.appendChild(a);
            a.click();

            // Clean up
            window.URL.revokeObjectURL(url);
            document.body.removeChild(a);
          } else {
            // Handle error
            console.error('Failed to download invoice');
          }
        }
    };
    xhr.send(formData);
});
</script>
```

**7) cart.php**

```
<?php

@include 'config.php';

if(isset($_POST['update_update_btn'])){
   $update_value = $_POST['update_quantity'];
   $update_id = $_POST['update_quantity_id'];
   $update_quantity_query = mysqli_query($conn, "UPDATE `cart` SET quantity =
'$update_value' WHERE id = '$update_id'");
   if($update_quantity_query){
      header('location:cart.php');
   };
};

if(isset($_GET['remove'])){
   $remove_id = $_GET['remove'];
   mysqli_query($conn, "DELETE FROM `cart` WHERE id = '$remove_id'");
   header('location:cart.php');
};

if(isset($_GET['delete_all'])){
   mysqli_query($conn, "DELETE FROM `cart`");
   header('location:cart.php');
}

?>
<?php include 'header.php'; ?>

<div class="container">

<section class="shopping-cart">

   <h1 class="heading">shopping cart</h1>

   <table>

      <thead>
         <th>image</th>
         <th>name</th>
         <th>price</th>
         <th>quantity</th>
         <th>total price</th>
         <th>action</th>
      </thead>

      <tbody>

         <?php
```

```
        $select_cart = mysqli_query($conn, "SELECT * FROM `cart`");
        $total=0;
        if(mysqli_num_rows($select_cart) > 0){
          while($fetch_cart = mysqli_fetch_assoc($select_cart)){
            $total+= $fetch_cart['price']*$fetch_cart['quantity']
        ?>

        <tr>
          <td><img src="images/<?php echo $fetch_cart['image']; ?>" height="100" alt=""></td>
          <td><?php echo $fetch_cart['name']; ?></td>
          <td>₹<?php echo number_format($fetch_cart['price']); ?>/-</td>
          <td>
            <form action="" method="post">
              <input type="hidden" name="update_quantity_id"  value="<?php echo
$fetch_cart['id']; ?>" >
              <input type="number" name="update_quantity" min="1"  value="<?php echo
$fetch_cart['quantity']; ?>" >
              <input type="submit" value="update" name="update_update_btn">
            </form>
          </td>
          <td>₹<?php echo $sub_total = number_format($fetch_cart['price'] *
$fetch_cart['quantity']); ?>/-</td>
          <td><a href="cart.php?remove=<?php echo $fetch_cart['id']; ?>" onclick="return
confirm('remove item from cart?')" class="delete-btn"> <i class="fas fa-trash"></i>
remove</a></td>
        </tr>
        <?php
        };
      };
        ?>

        <tr class="table-bottom">
          <td><a href="index.php" class="option-btn" style="margin-top: 0;">continue
shopping</a></td>
          <td colspan="3">grand total</td>
          <td>₹<?php echo $total; ?>/-</td>
          <td><a href="cart.php?delete_all" onclick="return confirm('are you sure you want to
delete all?');" class="delete-btn"> <i class="fas fa-trash"></i> delete all </a></td>
        </tr>

      </tbody>

    </table>

    <div class="checkout-btn">
      <a href="checkout.php" class="btn <?= ($total > 1)?'':'disabled'; ?>">procced to
checkout</a>
    </div>

</section>
```

**8) checkout2.php**

```
<?php

@include 'config.php';

if(isset($_POST['update_update_btn'])){
   $update_value = $_POST['update_quantity'];
   $update_id = $_POST['update_quantity_id'];
   $update_quantity_query = mysqli_query($conn, "UPDATE `cart` SET quantity =
'$update_value' WHERE id = '$update_id'");
   if($update_quantity_query){
      header('location:cart.php');
   };
};

if(isset($_GET['remove'])){
   $remove_id = $_GET['remove'];
   mysqli_query($conn, "DELETE FROM `cart` WHERE id = '$remove_id'");
   header('location:cart.php');
};

if(isset($_GET['delete_all'])){
   mysqli_query($conn, "DELETE FROM `cart`");
   header('location:cart.php');
}

?>

<?php

@include 'config.php';

if(isset($_POST['order_btn'])){

   $name = $_POST['name'];
   $number = $_POST['number'];
   $email = $_POST['email'];
   $method = $_POST['method'];
   $flat = $_POST['flat'];
   $street = $_POST['street'];
   $city = $_POST['city'];
   $state = $_POST['state'];
   $country = $_POST['country'];
   $pin_code = $_POST['pin_code'];

   $cart_query = mysqli_query($conn, "SELECT * FROM `cart`");
   $price_total = 0;
   $product_name = [];
   if(mysqli_num_rows($cart_query) > 0){
      while($product_item = mysqli_fetch_assoc($cart_query)){
```

```
      $product_name[] = $product_item['name'] .' (. $product_item['quantity'] .') ';
      $product_price = intval($product_item['price']) * intval($product_item['quantity']);
      $price_total += $product_price;
    }
  }

  $total_product = implode(', ',$product_name);
  $detail_query = mysqli_query($conn, "INSERT INTO `order`(name, number, email, method,
flat, street, city, state, country, pin_code, total_products, total_price)
VALUES('$name','$number','$email','$method','$flat','$street','$city','$state','$country','$pin_cod
e','$total_product','$price_total')") or die('query failed');

  if($cart_query && $detail_query){
    session_start();
    $_SESSION['order_data'] = array(
      'name' => $name,
      'number' => $number,
      'email' => $email,
      'method' => $method,
      'flat' => $flat,
      'street' => $street,
      'city' => $city,
      'state' => $state,
      'country' => $country,
      'pin_code' => $pin_code,
      'total_product' => $total_product,
      'price_total' => $price_total
    );

    echo "
    <div class='order-message-container'>
    <div class='message-container'>
      <h3>thank you for shopping!</h3>
      <div class='order-detail'>
        <span>".$total_product."</span>
        <span class='total'> total :  ₹".$price_total."/-  </span>
      </div>
      <div class='customer-details'>
        <p> your name : <span>".$name."</span> </p>
        <p> your number : <span>".$number."</span> </p>
        <p> your email : <span>".$email."</span> </p>
        <p> your address : <span>".$flat.", ".$street.", ".$city.", ".$state.", ".$country." -
".$pin_code."</span> </p>
        <p> your payment mode : <span>".$method."</span> </p>
        <p>(*pay when product arrives*)</p>
      </div>
        <a href='index.php' class='btn'>continue shopping</a>
        <a href='payment.php' class='btn'>Pay</a>
      </div>
    </div>";
  }
```

```
}

?>

<?php include 'header.php'; ?>

<div class="container">

<section class="checkout-form">

  <h1 class="heading">complete your order</h1>

  <form id="checkout_form" action="" method="post">

  <div class="display-order">
    <?php
      $select_cart = mysqli_query($conn, "SELECT * FROM `cart`");
      $total = 0;
      $grand_total = 0;
      if(mysqli_num_rows($select_cart) > 0){
        while($fetch_cart = mysqli_fetch_assoc($select_cart)){
        $total_price = $fetch_cart['price'] * $fetch_cart['quantity'];
        $grand_total = $total += $total_price;
    ?>
    <span><?= $fetch_cart['name']; ?>(<?= $fetch_cart['quantity']; ?>)</span>
    <?php
      }
    }else{
      echo "<div class='display-order'><span>your cart is empty!</span></div>";
    }
    ?>
    <span class="grand-total"> grand total : ₹<?= $grand_total; ?>/- </span>
  </div>

    <div class="flex">
      <div class="inputBox">
        <span>your name</span>
        <input type="text" placeholder="enter your name" name="name" id="name"
value="<?php echo isset($name) ? $name : ''; ?>">
      </div>
      <div class="inputBox">
        <span>your number</span>
        <input type="number" placeholder="enter your number" name="number" value="<?php
echo isset($number) ? $number : ''; ?>" required>
      </div>
      <div class="inputBox">
        <span>your email</span>
        <input type="email" placeholder="enter your email" name="email" value="<?php echo
isset($email) ? $email : ''; ?>" required>
      </div>
```

```
        <div class="inputBox">
          <span>payment method</span>
          <select name="method">
            <option value="credit card" selected>credit card</option>
          </select>
        </div>
        <div class="inputBox">
          <span>address line 1</span>
          <input type="text" placeholder="e.g. flat no." name="flat" value="<?php echo
isset($flat) ? $flat : ''; ?>" required>
        </div>
        <div class="inputBox">
          <span>address line 2</span>
          <input type="text" placeholder="e.g. street name" name="street" value="<?php echo
isset($street) ? $street : ''; ?>" required>
        </div>
        <div class="inputBox">
          <span>city</span>
          <input type="text" placeholder="e.g. mumbai" name="city" value="<?php echo
isset($city) ? $city : ''; ?>" required>
        </div>
        <div class="inputBox">
          <span>state</span>
          <input type="text" placeholder="e.g. maharashtra" name="state" value="<?php echo
isset($state) ? $state : ''; ?>" required>
        </div>
        <div class="inputBox">
          <span>country</span>
          <input type="text" placeholder="e.g. india" name="country" value="<?php echo
isset($country) ? $country : ''; ?>" required>
        </div>
        <div class="inputBox">
          <span>pin code</span>
          <input type="text" placeholder="e.g. 123456" name="pin_code" value="<?php echo
isset($pin_code) ? $pin_code : ''; ?>" required>
        </div>
      </div>
      <input type="hidden" name="product" value="<?php echo $total_product; ?>">
      <input type="hidden" name="price" value="<?php echo $price_total; ?>">
      <input type="submit" value="order now" name="order_btn" class="btn">

    </form>

</section>
```

**9) generateinvoice.php**

```
<?php
session_start();

// Check if order data is available in the session
if(isset($_SESSION['order_data'])) {
    // Retrieve order data from the session
    $name = $_SESSION['order_data']['name'];
    $number = $_SESSION['order_data']['number'];
    $email = $_SESSION['order_data']['email'];
    $method = $_SESSION['order_data']['method'];
    $flat = $_SESSION['order_data']['flat'];
    $street = $_SESSION['order_data']['street'];
    $city = $_SESSION['order_data']['city'];
    $state = $_SESSION['order_data']['state'];
    $country = $_SESSION['order_data']['country'];
    $pin_code = $_SESSION['order_data']['pin_code'];
    $total_product = $_SESSION['order_data']['total_product'];
    $price_total = $_SESSION['order_data']['price_total'];

    // Include FPDF library
    require('fpdf186/fpdf.php');

    // Create a new instance of FPDF
    $pdf = new FPDF();
    $pdf->AddPage();

    // Set font for the document
    $pdf->SetFont('Arial', 'B', 16);

    // Add logo and company name to the PDF
    $pdf->Image('./images/icon.png',10,5,20); // Adjust the path and dimensions of the logo as
needed
    $pdf->Cell(30);
    $pdf->SetFont('Arial', 'B', 21); // Move to the right to leave space between the logo and the
company name
    $pdf->Cell(0, 10, 'QuickShop', 0, 1); // Adjust the company name as needed

    $pdf->Ln(10);

    // Add title to the PDF
    $pdf->SetFont('Arial', 'B', 16);
    $pdf->Cell(0, 10, 'Invoice', 0, 1, 'C');

    $pdf->Ln(10);

    // Set font for the content
    $pdf->SetFont('Arial', '', 14);

    // Add order details to the PDF
```

```
    $pdf->Cell(0, 10, 'Customer Name: ' . $name, 0, 1);
    $pdf->Cell(0, 10, 'Customer Number: ' . $number, 0, 1);
    $pdf->Cell(0, 10, 'Customer Email: ' . $email, 0, 1);
    $pdf->Cell(0, 10, 'Address: ' . $flat . ', ' . $street . ', ' . $city . ', ' . $state . ', ' . $country . ' - ' .
$pin_code, 0, 1);
    $pdf->Cell(0, 10, 'Total Products: ' . $total_product, 0, 1);
    $pdf->Cell(0, 10, 'Total Price: &#8377;' . $price_total, 0, 1);
    $pdf->Cell(0, 10, 'Payment Method: ' . $method, 0, 1);

    // Output the PDF
    $pdf->Output();
} else {
    echo "Order data not found!";
}
?>
```

**10) generate_invoice2.php**

```
<?php
require('fpdf186/fpdf.php');
// Check if all required parameters are set
if(isset($_POST['name'], $_POST['number'], $_POST['email'], $_POST['method'],
$_POST['flat'], $_POST['street'], $_POST['city'], $_POST['state'], $_POST['country'],
$_POST['pin_code'])) {
   // Assign values from POST data
   $name = $_POST['name'];
   $number = $_POST['number'];
   $email = $_POST['email'];
   $method = $_POST['method'];
   $flat = $_POST['flat'];
   $street = $_POST['street'];
   $city = $_POST['city'];
   $state = $_POST['state'];
   $country = $_POST['country'];
   $pin_code = $_POST['pin_code'];
   $product=$_POST["product"];
   $price=$_POST["price"];
   // Create PDF
   $pdf = new FPDF();
   $pdf->AddPage();
   $pdf->Image('./images/icon.png',10,5,20); // Adjust the path and dimensions of the logo as
needed
   $pdf->Cell(30);
   $pdf->SetFont('Arial', 'B', 21); // Move to the right to leave space between the logo and the
company name
   $pdf->Cell(0, 10, 'QuickShop', 0, 1); // Adjust the company name as needed
   $pdf->Ln(10);
   $pdf->SetFont('Arial', 'B', 16);
   $pdf->Cell(0, 10, 'Invoice', 0, 1, 'C');

   $pdf->Ln(10);

   $pdf->SetFont('Arial', '', 12);
   $pdf->Cell(0,10,"Total product: ".$product,0,1);
   $pdf->Cell(0,10,"Total price: ".$price,0,1);
   $pdf->Cell(0, 10, 'Customer Name: ' . $name, 0, 1);
   $pdf->Cell(0, 10, 'Customer Number: ' . $number, 0, 1);
   $pdf->Cell(0, 10, 'Customer Email: ' . $email, 0, 1);
   $pdf->Cell(0, 10, 'Address: ' . $flat . ', ' . $street . ', ' . $city . ', ' . $state . ', ' . $country . ' - ' .
$pin_code, 0, 1);
   $pdf->Cell(0, 10, 'Payment Method: ' . $method, 0, 1);
   // Output PDF
   $pdf->Output();
} else {
   echo "Required parameters are missing!";
}
?>
```

**11) payment.php**

```
<script>
        function formatCardNumber(input) {
            // Remove any non-numeric characters
            let cardNumber = input.value.replace(/\D/g, '');

            // Insert hyphens after every 4 digits
            cardNumber = cardNumber.replace(/(\d{4})(?=\d)/g, '$1-');

            // Update the input value
            input.value = cardNumber;
        }

        // Add event listener to format the card number as the user types
        document.getElementById('card_number').addEventListener('input', function() {
            formatCardNumber(this);
        });
 </script>

<script>
        document.getElementById('exp_month').addEventListener('input', function() {
            // Remove non-numeric characters
            this.value = this.value.replace(/\D/g, '');

            // Ensure the value is within the range 01-12
            const month = parseInt(this.value, 10);
            if (month < 1 || month > 12) {
                this.setCustomValidity('Invalid month. Please enter a number between 01 and
12.');
            } else {
                this.setCustomValidity('');
            }
        });
</script>

<script>
        document.getElementById('exp_month').addEventListener('input', function() {
            // Remove non-numeric characters
            this.value = this.value.replace(/\D/g, '');

            // Ensure the value is within the range 01-12
            const month = parseInt(this.value, 10);
            if (month < 1 || month > 12) {
                this.setCustomValidity('Invalid month. Please enter a number between 01 and
12.');
            } else {
                this.setCustomValidity('');
            }
        });
</script>
```

**12) paymentsuccess.php**

```
<?php
session_start();

// Check if order data is available in the session
if(isset($_SESSION['order_data'])) {
    // Retrieve order data from the session
    $name = $_SESSION['order_data']['name'];
    $number = $_SESSION['order_data']['number'];
    $email = $_SESSION['order_data']['email'];
    $method = $_SESSION['order_data']['method'];
    $flat = $_SESSION['order_data']['flat'];
    $street = $_SESSION['order_data']['street'];
    $city = $_SESSION['order_data']['city'];
    $state = $_SESSION['order_data']['state'];
    $country = $_SESSION['order_data']['country'];
    $pin_code = $_SESSION['order_data']['pin_code'];
    $total_product = $_SESSION['order_data']['total_product'];
    $price_total = $_SESSION['order_data']['price_total'];

    // Now you can use these variables to display the order details or generate the PDF
} else {
    echo "Order data not found!";
}
?>

<script>
document.getElementById('download_invoice3').addEventListener('click', function() {
    // Create a new FormData object
    var formData = new FormData();

    // Add order data to formData
    formData.append('name', "<?php echo $name; ?>");
    formData.append('number', "<?php echo $number; ?>");
    formData.append('email', "<?php echo $email; ?>");
    formData.append('method', "<?php echo $method; ?>");
    formData.append('flat', "<?php echo $flat; ?>");
    formData.append('street', "<?php echo $street; ?>");
    formData.append('city', "<?php echo $city; ?>");
    formData.append('state', "<?php echo $state; ?>");
    formData.append('country', "<?php echo $country; ?>");
    formData.append('pin_code', "<?php echo $pin_code; ?>");

    // Create an XMLHttpRequest object
    var xhr = new XMLHttpRequest();
    xhr.open('POST', './generateinvoice.php', true);
    xhr.responseType = 'blob'; // Set response type to blob for downloading files

    // Define onload event handler
    xhr.onload = function() {
```

```
        if (xhr.status === 200) {
            // Create a blob URL from the response
            var blob = new Blob([xhr.response], { type: 'application/pdf' });
            var url = window.URL.createObjectURL(blob);

            // Create a link element and trigger the download
            var a = document.createElement('a');
            a.href = url;
            a.download = 'invoicedownloads.pdf';
            document.body.appendChild(a);
            a.click();

            // Clean up
            window.URL.revokeObjectURL(url);
            document.body.removeChild(a);
        } else {
            // Handle error
            console.error('Failed to download invoice');
        }
    };

    // Send formData as the request body
    xhr.send(formData);
});
</script>
```

**13) product_detail.php**

```
<?php

@include 'config.php';

if(isset($_GET['add_to_cart'])){

   $product_name = $_GET['product_name'];
   $product_price = $_GET['product_price'];
   $product_image = $_GET['product_image'];
   $product_quantity = 1;

   $select_cart = mysqli_query($conn, "SELECT * FROM `cart` WHERE name =
'$product_name'");

  if(mysqli_num_rows($select_cart) > 0){
    $message[] = 'product already added to cart';
  }else{
    $insert_product = mysqli_query($conn, "INSERT INTO `cart`(name, price, image, quantity)
VALUES('$product_name', '$product_price', '$product_image', '$product_quantity')");
    $message[] = 'product added to cart succesfully';
  }
}

?>

<?php
@include 'config.php';

if(isset($_GET['id'])){

   $product_id = $_GET['id'];

   $product = mysqli_query($conn, "SELECT * FROM `products` WHERE id=$product_id");

 $select_product=mysqli_fetch_row($product);
}
?>
<div class="product-price">
        <span>₹<?php echo $select_product[2]; ?></span>
        <form action="./index.php" method="get">
          <input type="hidden" name="product_name"  value="<?php echo $select_product[1];
?>">
          <input type="hidden" name="product_price" value="<?php echo $select_product[2];
?>">
          <input type="hidden" name="product_image"  value="<?php echo $select_product[3];
?>">
          <input type="submit" class="cart-btn" value="Add to cart" name="add_to_cart">
        </form>
</div>
```

**14) script.js**

```
// init Isotope
var $grid = $('.collection-list').isotope({
  // options
});
// filter items on button click
$('.filter-button-group').on( 'click', 'button', function() {
  var filterValue = $(this).attr('data-filter');
  resetFilterBtns();
  $(this).addClass('active-filter-btn');
  $grid.isotope({ filter: filterValue });
});

var filterBtns = $('.filter-button-group').find('button');
function resetFilterBtns(){
  filterBtns.each(function(){
    $(this).removeClass('active-filter-btn');
  });
}
```

**15) script1.js**

```
let menu = document.querySelector('#menu-btn');
let navbar = document.querySelector('.header .navbar');

menu.onclick = () =>{
   menu.classList.toggle('fa-times');
   navbar.classList.toggle('active');
};

window.onscroll = () =>{
   menu.classList.remove('fa-times');
   navbar.classList.remove('active');
};


document.querySelector('#close-edit').onclick = () =>{
   document.querySelector('.edit-form-container').style.display = 'none';
   window.location.href = 'admin.php';
};
```

**16) clear_cart.js**

```
<?php
@include 'config.php';

// Check if the request is a GET request
if ($_SERVER['REQUEST_METHOD'] === 'GET') {
    // Clear the cart
    mysqli_query($conn, "DELETE FROM `cart`");
    // Respond with a success message
    echo "Cart cleared successfully";
} else {
    // Respond with an error message
    echo "Invalid request method";
}
?>
```

**17) header.js**

```
<header class="header">

   <div class="flex">

      <a href="#" class="logo">Order Details</a>

      <nav class="navbar">
         <a href="index.php">view products</a>
      </nav>

      <?php

      $select_rows = mysqli_query($conn, "SELECT * FROM `cart`") or die('query failed');
      $row_count = mysqli_num_rows($select_rows);

      ?>

      <a href="cart.php" class="cart">cart <span><?php echo $row_count; ?></span> </a>

      <div id="menu-btn" class="fas fa-bars"></div>

   </div>

</header>
```

# 11 System Testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called assemblages) or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter- assemblages" and also within the system as a whole.

## Testing the whole system :

System testing is performed on the entire system in the context of a Functional Requirement Specification(s) (FRS) and/or a System Requirement Specification(SRS). System testing tests not only the design, but also the behavior and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification(s).

## Testing Used:

At the beginning when they installed the women safety app ,at first instance the SOS message was lately sending and functioning of application were incorporated together as smooth as expected for that various set of functions

Test Cases, Test Results and Test Data

What is Test Case?

“A Test Case has a component that describe an input, action or event expected response, to determine if a feature of an application is working correctly.” Software testing can be stated as the process of validating and verifying that a computer program/application/product:

- Meets the requirements that guided its design and development.
- Works as expected
- Can be implemented with the same characters.
- And satisfies the needs of Stakeholders.

Why we Write Test Case?

A Test Case in Software Engineering is a set of conditions or variables under which atester will determine whether an application, software system or one of its features is working as it was originally established for it to do. Test Cases bring some sort of standardization and minimize the ad-hoc approach in testing.

## Case Study

**1. Test Case:** User Registration

**Objective:** User registers in successfully

**Steps:**

Open registration page

Enter username, phone number, email address, password

Click on register.

**Expected Result**: Email Verification, Validation successful and user is redirected to login page.

**Actual Result**: User registration successful.

**2. Test Case:** User Login

**Objective:** User logs in successfully

**Steps:**

Open login page

Enter username, password

Click on login.

**Expected Result**: Validation successful and user is redirected to Home page.

**Actual Result**: User login successful.

**3. Test Case: Add Products to Cart**

**Objective:** Add to Cart

**Steps:**

Products add to cart,

Edit products, increase quantity

Click on Proceed to checkout.

**Expected Result**: Product add to cart and proceed to checkout

**Actual Result**: Product add Successfully.

**4. Test Case: Checkout Details**

**Objective:** Checkout to add User details

**Steps:**

Enter valid Details

At checkout page shows buy product and total price

Click on Pay as Per Method.

**Expected Result**: With Order Details and Billing Information Order is place and Download invoice.

**Actual Result**: Billing Details is add Successfully.

**5. Test Case: Credit card Payment**

**Objective:** Add Credit Card Details

**Steps:**

Enter card details to make payment

Click on Pay.

**Expected Result**: Payment Successful message and Download Invoice button.

**Actual Result**: Payment Done Successfully.

# 12 Future Enhancement

As the developer and owner of Quickshop, I take pride in the success of this project. However, I believe there's always room for improvement, and I'm excited to share some future enhancements I plan to implement to elevate the Quickshop experience further.

1. **User Authentication and Profiles**: Implement user authentication to allow users to create accounts, log in, and manage their profiles. This would include features like order history, saved addresses, and preferences.

2. **Admin Panel**: Create an admin panel where administrators can manage products, view orders, and generate reports. This would involve adding roles and permissions to restrict access to certain functionalities.

3. **Product Reviews and Ratings**: Allow users to leave reviews and ratings for products they have purchased. This can help build trust and provide valuable feedback to other users.

4. **Search and Filtering**: Improve the product browsing experience by adding search functionality and filters such as price range, category, and brand.

5. **Responsive Design**: Ensure that your website is fully responsive and optimized for mobile devices. This will improve the user experience and accessibility across different screen sizes.

6. **Email Notifications**: Implement email notifications to send order confirmations, shipping updates, and promotional offers to customers.

7. **Payment Gateway Integration**: Integrate popular payment gateways to allow users to make secure online payments. Options like PayPal, Stripe, and others can be considered based on your target audience.

8. **Inventory Management**: Develop a system to track inventory levels and automatically update product availability based on purchases.

9. **Localization**: Add support for multiple languages and currencies to make your website accessible to a wider audience.

10. **Social Media Integration**: Allow users to share products on social media platforms and incorporate social login options to streamline the registration process.

11. **Live Chat Support**: Provide real-time customer support through a live chat feature to address any queries or concerns users may have while browsing or making a purchase.

12. **SEO Optimization**: Optimize your website for search engines to improve visibility and attract more organic traffic. This includes optimizing meta tags, creating unique product descriptions, and building quality backlinks.

13. **Analytics and Insights**: Implement tracking tools like Google Analytics to gather data on user behavior, traffic sources, and conversion rates. Use this data to make informed decisions and optimize your website for better performance.

14. **Continuous Testing and Improvement**: Regularly test your website for usability, performance, and security issues. Gather feedback from users and incorporate improvements based on their suggestions and preferences.

# 13 References

## 13.1 Textbook Reference

1) "PHP and MySQL Web Development" by Luke Welling and Laura Thomson.
2) "PHP and MySQL for Dynamic Web Sites" by Larry Ullman.
3) "Learning PHP, MySQL & JavaScript" by Robin Nixon.

## 13.2 Website Reference

1)https://www.youtube.com/results?search_query=code+with+harry+php
2) https://www.tutorialspoint.com/php/index.htm
3) https://www.youtube.com/watch?v=aSTLAjLjJq