IMAGE RECOGNITION BASED ADVANCED E-COMMERCE WEBSITE

A PROJECT

Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF COMPUTER APPLICATIONS

BY

Prakhar Srivastava(BCA/40532/20)

Himanshu Swaraj(BCA/40521/20)

Yash Shrivastava(BCA/40581/20)

Nilay Shekhar(BCA/40555/20)

UNDER THE GUIDANCE OF

(Dr. Partha Sarathi Bishnu)

Department Of Computer Science

DEPARTMENT OF COMPUTER SCIENCE AND ENINEERING
BIRLA INSTITUTE OF TECHNOLGY
EXTENSION CENTER LALPUR, RANCHI

(2023)

BIRLA INSTITUTE OF TECHNOLOGY
MESRA, RANCHI-835215
(Deemed University)
EXTENSION CENTER LALPUR, RANCHI

Certificate

This is to certify that the contents of this project entitled "Image Recognition Based Advanced E-Commerce Website" is a bonafide work carried out by Prakhar Srivastava, Himanshu Swaraj, Yash Shrivastava and Nilay Shekhar in partial fulfillment of the requirement for the award of Bachelor of Computer Applications under my guidance.

This is to further certify that the matter of the project has not been submitted by anyone else for the award of any degree.

(Dr.Partha Sarathi Bishnu)
Department of Computer science.
Birla Institute of Technology.
Lalpur Off campus, Ranchi.

BIRLA INSTITUTE OF TECHNOLOGY
MESRA, RANCHI-835215

(Deemed University)

Extension Center Lalpur, Ranchi.

Certificate of Approval

The foregoing project is hereby approved as a ***creditable work*** *on* ***"Image Recognition Based Advanced E-Commerce Website", carried out*** *and* ***presented in the manner satisfactory to warrant its acceptance*** *as prerequisite to the degree for which it has been entitled.*

It is understood that by this approval the undersigned do not endorse any statement made or opinion expressed but approve work for the purpose for which it is submitted.

Committee for evaluation of the project

External Examiner
Date:

Internal Examiner
Date:

Director (In-Charge)
BILRLA INSTITUTE OF TECHNOLOGY
Extension Center Lalpur, Ranchi

ACKNOWLEDGEMENT

We would like to express our deepest gratitude to ***Dr. Partha Sarathi Bishnu***, **our esteemed project guide**, for his invaluable guidance, expertise, and unwavering support throughout the development of our project. His profound knowledge, unwavering dedication, and invaluable mentorship have played a pivotal role in shaping our project into its final form.

We are grateful to ***Dr. Vandana Bhattacharya***, **In-charge of Birla Institute of Technology, Lalpur Campus**, for extending all facilities that were required for carrying out this work.

We would like to further extend a heartfelt thanks to our friends and family for their unwavering support and encouragement which have been a driving force behind our determination to complete this project successfully.

NAME	**ROLL NO.**	**SIGNATURE**
1. Prakhar Srivastava	BCA/40532/20	
2. Himanshu Swaraj	BCA/40521/20	
3. Yash Shrivastava	BCA/40581/20	
4. Nilay Shekhar	BCA/40555/20	

INDEX

Abstract:

"Image Recognition Based Advanced E-Commerce Website" is an innovative platform that aims to provide a unique and user-friendly shopping experience to customers. The website is an e-commerce based platform that caters to the sale of a wide range of products such as televisions, phones, refrigerators, and more. The platform is designed to meet the needs of customers who are looking for a more efficient and seamless way to shop online. In addition to traditional text-based searches, the platform offers an advanced image recognition technology that allows customers to search for products by simply uploading an image of the desired product. The system will then generate a list of similar products available on the website, providing customers with a more accurate and convenient way of finding what they're looking for. The website is fully responsive, ensuring that customers can access it from any device with an internet connection. Customers can add products to their cart and purchase them with ease. Furthermore, the platform includes a "Contact Us" section where customers can communicate with the customer support team and ask any queries related to their purchases. To ensure efficient management of the platform, it has a robust backend system managed by an admin panel. The admin panel provides an overview of all the transactions and messages received by the website. It also allows its administrator to add new products and manage the website's inventory, ensuring that the customers have access to the latest and most relevant products.

Introduction:

The advent of e-commerce has brought about a fundamental transformation in the way we shop for goods and services. It has become an integral part of our daily lives, enabling us to purchase a wide range of products and services from the comfort of our homes. Nevertheless, the task of identifying the appropriate product from a plethora of alternatives can be quite challenging.

In light of this challenge, "The Image Recognition Based Advance E-commerce Website", an innovative platform, seeks to redefine the online shopping experience. This e-commerce-based website is specialized in retailing electronic items such as televisions, phones, refrigerators, and so forth. The website is tailored to cater to the needs of customers who require a more efficient and seamless way to shop online.

The website's key differentiator is its state-of-the-art image recognition technology. This technology allows customers to search for products by merely uploading an image. For example, if a customer desires a new phone, they can upload an image of a phone they prefer, and the website's sophisticated algorithm will generate a list of comparable products available on the website. This advanced feature streamlines the search process for customers, eliminating the tedious manual search process.

Furthermore, the website is designed to be user-friendly and responsive, making it accessible to a wide range of customers across various devices. Customers can effortlessly browse the website, add products to their cart, which would further allow them to make purchases with ease. The website's back-end is managed via an admin panel that provides a comprehensive overview of all the transactions and messages received by the website. The

admin panel also enables the website's administrator to add new products and manage the inventory.
The website features a “Contact Us” section, where customers can submit any queries related to their purchases. This feature ensures that customers have a convenient and reliable way to communicate with the website's customer support team. The dashboard for administrators helps them manage the website's inventory and transactions effectively. The admin panel provides a comprehensive view of all the products bought by customers, allowing administrators to keep track of the website's sales and manage the inventory accordingly.

Working of Image Search

1. The website is an advance ecommerce website that uses image recognition technology for searching products.
2. Instead of comparing the uploaded image directly with images in the database, the website generates a name (string class) for the uploaded image using the ResNet152V2 model.
3. The generated string name is then compared with the name of similar category of inventory products in the database.
4. This comparison produces results for the end user, showing them products that are similar to the searched image.

ResNet152V2
1. ResNet152V2 is a deep neural network architecture.
2. It has 152 layers and uses skip connections to enable training of very deep networks.
3. It uses residual blocks and bottleneck blocks as its basic building blocks.
4. ResNet152V2 has achieved state-of-the-art performance on a variety of image classification tasks, including the ImageNet dataset.
5. It can be computationally expensive to train and requires a lot of memory.

Scope:

1. **Online shopping:** Customers can easily browse and purchase products online from the comfort of their own homes.

2. **Convenience:** The website offers 24/7 accessibility and can be accessed from anywhere with an internet connection.

3. **Wide range of products:** The website can offer a vast range of products that may not be available in local physical stores.

4. **Increased customer reach:** The website can reach a wider audience and potentially attract customers from all over the world.

5. **Better customer service:** The website can provide efficient and responsive customer service via "Contact Us" section to ensure a smooth shopping experience for customers.

6. **Enhanced Shopping Experience through Image Recognition Technology:** The website offers an innovative feature that allows customers to upload images of products they like. The image recognition technology then provides recommendations for similar or matching items, leading to increased customer satisfaction and sales.

Advantages:

1. **Improved user experience:** Image recognition technology allows customers to search for products by uploading an image, which is much faster and more convenient than typing out a search query. This can make the shopping experience more enjoyable and increase customer satisfaction.

2. **Increased sales:** By making it easier for customers to find the products they're looking for, an image recognition-based e-commerce website can help increase sales. This is particularly true for products that are difficult to describe in words, such as fashion items or home decor.

3. **Reduced returns:** If customers are able to find exactly what they're looking for using image recognition technology, they are less likely to purchase the wrong item and subsequently return it. This can help reduce return rates and save the company money.

4. **Improved inventory management:** An image recognition-based e-commerce website can help companies keep track of their inventory more efficiently. By analyzing the images of products that customers are searching for, companies can better understand which products are most popular and adjust their inventory accordingly.

5. **Password encryption:** The website's security measures include encrypted storage of user passwords in the database. By using encryption, it prioritizes the privacy and security of the users. This ensures that even the admin cannot access or know the end user's password.

6. **Improvised Product Search:** The website has a smart search algorithm that can predict and display the exact product the user is looking for, even if he only types in a few letters of the product name in the search bar. This saves time and effort of the end user.

7. **User-Friendly interface:** The website has a simple and intuitive interface that is easy to navigate, even for first-time users. It has all the essential features right where the user needs them, with no clutter or unnecessary distractions which would provide him with a seamless and enjoyable shopping experience

8. **Extensive Product Filters:** The website offers a wide range of filters which refines the search results by category, brand and other criteria. It helps the user find the product he is looking for quickly and easily and saves time and effort.

Limitations:

a) **Image Search limitation:** In the website the searched image from the end user is not compared with the similar images in the database. Instead, a name, which is a string class, is generated for the searched image using an algorithm. And that generated string name is compared with the name of the similar category of inventory products in the database to produce results.

b) **Unavailability of "Wishlist" feature:** The "Wishlist" feature allows customers to save products for future purchase and without this feature, customers may forget about products they were interested in or have to spend time searching for them again. The unavailability of this feature might also lead to lesser tracking of customer preferences which may act as a hindrance in providing better personalized recommendations, flattening the curve of customer engagement and sales.

c) **Risk of data loss:** The complete data of the website is stored in a single and centralized database and there is no duplicate database for backup and recovery. Therefore, the website is prone to the permanent loss of critical information such as customer data, transactions, and inventory records.

d) **Entering of address multiple times:** The users have to manually input their address each time they make a purchase, which can be time-consuming and tedious and may cause inconvenience to the end user, which might deter customers from making repeat purchases.

e) **Alignment issue in Order Placed Section:** The display of purchased items in the order placed section is not properly aligned. It might lead to inconvenience for the customer to view their order details properly after placing multiple orders.

f) **Limitation in Order Management System:** The current order management system does not provide a separate classification of pending and completed orders for the website admin. This means that for the admin, all orders, whether pending or completed, are displayed together in the order management section of the database. And without proper classification of pending and completed orders, it can be difficult for the website admin to keep track of orders and ensure that they are processed and delivered in a timely manner.

g) **Limitation in multiple order management:** The current order management system does not support simultaneous updation or deletion of multiple orders from the website admin database. Without the ability to update or delete multiple orders at once, the admin may become overwhelmed and may struggle to keep up with the demands of the customers.

Potential Future Modifications:

1. **Expansion of Inventory:** The website could be enhanced via expansion of inventory by adding more products. This would allow customers to have a wider range of products to choose from and increase the chances of making a sale. The addition of new products would

also attract new customers who are interested in those products, which can lead to increased website traffic and revenue. Additionally, regularly updating the inventory can keep the website fresh and relevant to customers, which can help retain existing customers and attract new ones.

2. **Enhanced Product Filters:** Another potential future modification for the website is to add extra filters to help customers find their desired product more efficiently. By providing more filter options, customers can narrow down their search results to show only the products that meet their specific criteria. This can improve the user experience by making it easier and faster for customers to find what they are looking for. Additionally, this feature can help customers discover new products that they may not have found otherwise, which can lead to increased sales and customer satisfaction.

3. **Wishlist Feature:** Another enhancement for website is to add a "Wishlist" section. This feature would allow customers to save products that they are interested in buying but are not yet ready to purchase. By saving these products, customers can easily find them later when they are ready to make a purchase. This feature can also help the website keep track of popular products and provide insights for future inventory management. Additionally, this feature can improve customer engagement and loyalty by providing a personalized experience for each customer.

4. **Improved Image Recognition System:** The current image recognition system on the e-commerce website generates a string class for the searched image and then compares it with the category name in the database to display similar images. However, a potential future modification is to enhance this system by developing an algorithm that actually compares images based on several attributes such as color, texture, and shape to provide more accurate search results. By using image analysis techniques, the website can provide more relevant and personalized search results to customers, which can lead to increased customer satisfaction and sales. Additionally, this feature can help the website stand out from its competitors and improve its overall user experience.

5. **Database Redundancy:** The website could be further modified by enhancing its database system via implementation of database redundancy. Currently, the website only has a single database to store its data, which can make it prone to data loss and make data recovery difficult. By implementing database redundancy, multiple copies of the database can be created, which can help ensure data is not lost in the event of a failure. This can help improve the reliability and availability of the website, which can lead to increased customer trust and satisfaction. Additionally, this feature can help the website comply with data protection regulations and improve its overall data security.

6. **Enhanced Order Management System:** The e-commerce website's current order management system does not provide separate classification of pending and completed orders to the website admin. Additionally, it does not support simultaneous updation and deletion of orders from the database by the website admin. A potential future modification is to enhance this system by adding features such as separate classification of orders and the ability for the admin to update or delete multiple orders at the same time. This can

improve the efficiency and organization of the order management process, making it easier for the admin to keep track of orders and manage them effectively. Additionally, this feature can help the website provide better customer service by providing timely updates on order status and processing.

Hardware and Software Requirements:

Hardware Specifications (User's Side)

RAM	**1 GB and above**
ROM	**4 GB and above**
Processor	**i3 and above**
Input Devices	**Mouse, Keyboard**

Hardware Specifications(Developer's Side)

RAM	**2 GB and above**
ROM	**16 GB and above**
Processor	**i3 and above**
Input Devices	**Mouse, Keyboard**

Software Specifications(User's Side)

Operating System	**Windows 7 and higher**
Browser	**Chrome, Firefox etc**
Database	**MySQL – Version 10.4.28**

Software Specifications(Developer's Side)

Operating System	**Windows 7 and higher**
Browser	**Chrome, Firefox etc**
Database	**MySQL**

Packages and Modules Used:

1. **PDO (PHP Data Objects):** A database abstraction layer in PHP that provides a consistent interface for accessing different database systems.
2. **Font Awesome:** A popular icon set and toolkit that provides scalable vector icons that can be customized with CSS.
3. **Swiper:** A free and open-source library to create sliders, carousels, and other types of interactive components. It's loaded from a CDN (Content Delivery Network) with the link:"https://unpkg.com/swiper@8/swiper-bundle.min.css".

4. **os:** This package provides a way of using operating system-dependent functionality in Python. In the code, it is used to join file paths and save the uploaded image to the specified upload folder.
5. **Flask:** This is a web application framework for Python. It provides tools, libraries, and technologies for building web applications. In the code, it is used to create the web application, define routes, and handle requests and responses.
6. **numpy:** This is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. In the code, it is used to preprocess and manipulate the input image.
7. **tensorflow.keras.preprocessing.image:** This is a module of the TensorFlow library that provides utilities for preprocessing image data. In the code, it is used to load and preprocess the input image.
8. **keras.applications.resnet v2:** This module provides pre-trained models for the ResNet152V2 architecture. In the code, it is used to load the pre-trained ResNet152V2 model.
9. **keras.applications.imagenet_utils:** This module provides utilities for decoding predictions made by pre-trained models trained on the ImageNet dataset. In the code, it is used to decode the predictions made by the ResNet152V2 model.
10. **Flask_cors:** This is an extension for Flask that allows cross-origin resource sharing (CORS) between the web application and other domains. In the code, it is used to enable CORS for the Flask application.

Tools and Technologies Used:

1. Visual Studio Code
    - XAMPP
    - Apache Tomcat
    - MySQL – Version 10.4.28

2. Programming languages:
    - Python
    - PHP
    - JavaScript

3. Markup language:
    - Html

4. Scripting language:
    - CSS

Visual Studio Code

Visual Studio Code is a freeware source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging. syntax highlighting, intelligent code completion, snippets, code refactoring. and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

Microsoft has released Visual Studio Code's source code on the microsoft/vscode (Code - OSS) repository of GitHub, under the permissive MIT License, while the releases by Microsoft are freeware.

In the Stack Overflow 2019 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool, with 50.7% of 87,317 respondents reporting that they use it.

Visual Studio Code was first announced on April 29, 2015, by Microsoft at the 2015 Build conference. A Preview build was released shortly thereafter.

On November 18, 2015, Visual Studio Code was released under the MIT License, having its source code available on GitHub. Extension support was also announced. On April 14, 2016, Visual Studio Code graduated from the public preview stage and was released to the Web.

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js, Python and C++.
It is based on the Electron framework, which is used to develop Node.js Web applications that run on the Blink layout engine. Visual Studio Code employs the same editor component (codenamed "Monaco") used in Azure DevOps (formerly called Visual Studio Online and Visual Studio Team Services).

Instead of a project system, it allows users to open one or more directories, which can then be saved in workspaces for future reuse. This allows it to operate as a language-agnostic code editor for any language. It supports a number of programming languages and a set of features that differs per language. Unwanted files and folders can be excluded from the project tree via the settings. Many Visual Studio Code features are not exposed through menus or the user interface but can be accessed via the command palette.

Visual Studio Code can be extended via extensions, available through a central repository. This includes additions to the editor and language support. A notable feature is the ability to create extensions that add support for new languages, themes, and debuggers, perform static code analysis, and add code linters using the Language Server Protocol.

Visual Studio Code includes multiple extensions for FTP, allowing the software to be used as a free alternative for web development. Code can be synced between the editor and the server, without downloading any extra software.

XAMPP

What is XAMPP?

- XAMPP is a free and open-source cross-platform web server solution stack.
- It includes Apache HTTP Server, MariaDB database, PHP scripting language, and other useful tools.
- XAMPP allows developers to easily create a local web server environment on their computer.

What can XAMPP do?

- XAMPP can be used to develop and test web applications on a local machine before deploying them to a live server.
- It supports PHP, Perl, and other programming languages.
- It includes phpMyAdmin, a popular tool for managing MySQL databases.
- XAMPP can be used to install and configure popular Content Management Systems like WordPress and Joomla.
- It provides an easy-to-use interface for configuring the web server and other components.

Why use XAMPP?

- XAMPP simplifies the process of setting up a local development environment.
- It allows developers to work on their web applications offline without the need for an internet connection.
- XAMPP is easy to install and use, even for those who are new to web development.
- It provides a complete development environment with all the necessary tools and components in one package.
- XAMPP is free and open-source, making it accessible to anyone who wants to develop and test web applications.

Apache Tomcat

What is Apache Tomcat?

- Apache Tomcat is an open-source web server and servlet container developed by the Apache Software Foundation.
- It provides a pure Java HTTP web server environment for running Java code.
- Apache Tomcat supports Java Servlets, JavaServer Pages (JSPs), and other Java-based web technologies.

What can Apache Tomcat do?

- Apache Tomcat can be used to run Java-based web applications on a web server.
- It provides a lightweight and efficient environment for running Java code.
- Apache Tomcat supports multiple web protocols and can be configured to use SSL/TLS encryption.
- It provides an easy-to-use management interface for configuring and monitoring the web server.
- Apache Tomcat is extensible and can be customized with additional libraries and plugins.

Why use Apache Tomcat?

- Apache Tomcat is a popular choice for deploying Java-based web applications.
- It is lightweight and fast, making it suitable for running web applications on low-resource systems.
- Apache Tomcat is free and open-source, making it accessible to anyone who wants to run Java-based web applications.
- It supports a wide range of web technologies, making it versatile and flexible for different types of web applications.
- Apache Tomcat has a large and active community of developers and users who contribute to its development and support.

MySQL

What is MySQL?

- MySQL is a widely used relational database management system (RDBMS).
- MySQL is free and open-source.
- MySQL is ideal for both small and large applications.

Where MySQL is used?

- Huge websites like Facebook, Twitter, Airbnb, Booking.com, Uber, GitHub, YouTube, etc.
- Content Management Systems like WordPress, Drupal, Joomla!, Contao, etc.
- A very large number of web developers around the world

To Show Data On the Web Site

To build a web site that shows data from a database, you will need:

- An RDBMS database program (like MySQL)

- A server-side scripting language, like PHP
- To use SQL to get the data you want
- To use HTML / CSS to style the page

Why MySQL?

MySQL is unquestionably the most reliable and secure database management system many renowned companies like Netflix and Amazon use. MySQL prevents your application's sensitive data from cyberattacks with data protection features.

Python

What is Python?

- Python is a high-level, interpreted programming language.
- It was first released in 1991 and has since become one of the most popular programming languages in the world.
- Python is known for its readability, ease of use, and extensive standard library.

What is a Python file?

- A Python file is a text file containing Python code.
- Python files typically have a .py extension.
- Python files can be executed by the Python interpreter, which reads the code and runs it.

What can Python do?

- Python can be used for a wide range of applications, including web development, data analysis, artificial intelligence, and more.
- Python has a large and active community of developers who create libraries and tools for various applications.
- Python supports object-oriented, imperative, and functional programming paradigms.
- Python is cross-platform, meaning that it can run on various operating systems, including Windows, Linux, and macOS.

Why use Python?

- Python is easy to learn and use, making it accessible to beginners and experienced developers alike.
- Python has a large and active community of developers who create libraries and tools for various applications.
- Python has a clear and readable syntax, making it easy to write and maintain code.
- Python is versatile and can be used for a wide range of applications.

- Python has excellent support for data analysis and scientific computing.

- Basic Python Syntax:

i. Comments start with the pound sign (#) and are ignored by the interpreter, for example –

```
#This is a comment in Python
```

ii. Here is an example of printing text to the console

```
print("Hello, world!")
```

iii. Python uses variables to store data

iv. Here is an example of creating a variable and assigning it a value

```
x = 5
```

v. You can perform arithmetic operations on variables

```
y = x + 3
```

vi. You can also use conditional statements to control the flow of your program

```
if y > 5:
print("Y is greater than 5")
else:
print("Y is not greater than 5")
```

vii. Python also has built-in functions that you can use-

Here is an example of using the len() function to get the length of a string

```
my_string = "Hello, world!"
string_length = len(my_string)
print("The length of the string is:", string_length)
```

PHP

What is PHP?

- PHP is an acronym for "PHP: Hypertext Pre-processor"
- PHP is a widely-used, open-source scripting language
- PHP scripts are executed on the server
- PHP is free to download and use

What is a PHP File?

- PHP files can contain text, HTML, CSS, JavaScript, and PHP code
- PHP code is executed on the server, and the result is returned to the browser as plain HTML
- PHP files have an extension ".php"

What Can PHP Do?

- PHP can generate dynamic page content
- PHP can create, open, read, write, delete, and close files on the server
- PHP can collect form data
- PHP can send and receive cookies
- PHP can add, delete, modify data in your database
- PHP can be used to control user-access
- PHP can encrypt data
- With PHP you are not limited to output HTML. You can output images or PDF files. You can also output any text, such as XHTML and XML.

Basic PHP Syntax

A PHP scripting block always starts with <?php and ends with ?>

Why PHP?

- PHP runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
- PHP is compatible with almost all servers used today (Apache, IIS, etc.)
- PHP supports a wide range of databases
- PHP is free. Download it from the official PHP resource: www.php.net
- PHP is easy to learn and runs efficiently on the server side

JavaScript

What is JavaScript?

- JavaScript is the Programming Language for the Web.
- JavaScript can update and change both HTML and CSS.
- JavaScript can calculate, manipulate and validate data.

What is a JavaScript File?

JS (JavaScript) are files that contain JavaScript code for execution on web pages. JavaScript files are stored with the .js extension. Inside the HTML document, you can either embed the JavaScript code using the <script></script> tags or include a JS file. Similar to CSS files, JS files can be included in multiple HTML documents for code reusability. JavaScript can be used to manipulate the HTML Document.

What can JavaScript do?

- JavaScript Can Change HTML Content

- JavaScript Can Change HTML Attribute Values
- JavaScript Can Change HTML Styles (CSS)
- JavaScript Can Hide HTML Elements
- JavaScript Can Show HTML Elements

Basic JavaScript Syntax

```
<html>
<head>
<meta charset="utf-8">
<title> JavaScript Code Structure </title>
</head>
<body>
   <h1>This is a web page.</h1>
<script>
    alert('I am the JavaScript Example');
</script>
</body>
</html>
```

Why JavaScript?

- Speed. Since JavaScript is an 'interpreted' language, it reduces the time required by other programming languages like Java for compilation.
- Simplicity. JavaScript is easy to understand and learn
- Popularity
- Interoperability
- Server Load
- Rich Interfaces
- Extended Functionality
- Versatility

HTML

What is HTML?

- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content

- HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

What is a HTML File?

- HTML stands for Hyper Text Markup Language.
- An HTML file is a text file containing small markup tags.
- The markup tags tell the Web browser how to display the page.
- An HTML file must have an html or html file extension.
- An HTML file can be created using a simple text editor.

What can HTML Do?

HTML (**H**yper **T**ext **M**arkup **L**anguage) is the code that is used to structure a web page and its content. For example, content could be structured within a set of paragraphs, a list of bulleted points, or using images and data tables. As the title suggests, this article will give you a basic understanding of HTML and its functions. HTML is a *markup language* that defines the structure of your content. HTML consists of a series of **elements**, which you use to enclose, or wrap, different parts of the content to make it appear a certain way, or act a certain way. The enclosing tags can make a word or image hyperlink to somewhere else, can italicize words, can make the font bigger or smaller, and so on.

Basic HTML Syntax

```
<html>
<head>
   <!--  Information about the page  -->
<title> The Great Indian Tourism </title>
</head>
<body>
   <!-- Contents of the webpage -->
</body>
</html>
```

Why HTML?

- HTML helps to build structure of a website and is a widely used Markup language.
- It is easy to learn.
- Every browser supports HTML Language.
- HTML is light weighted and fast to load.
- Storage of big files are allowed because of the application cache feature.
- Do not get to purchase any extra software because it's by default in every window.
- Loose syntax (although, being too flexible won't suit standards).

- HTML is simple to edit as being a plain text.

CSS

What is CSS?

- CSS stands for Cascading Style Sheets
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media
- CSS saves a lot of work. It can control the layout of multiple web pages all at once
- External stylesheets are stored in CSS files

What is a CSS File?

CSS (Cascading Style Sheets) are files that describe how HTML elements are displayed on the screen, paper, etc. With HTML, you can have either embedded styles or styles can be defined in an external stylesheet. For embedding the styles, the <style></style> tags are used. The external stylesheets are stored in files with the .css extension. With the external CSS, you can include it on multiple HTML pages to update the style of those pages. Even a single CSS file can be used to style a complete website.

What Can CSS Do?

CSS (Cascading Style Sheets) is used to style and layout web pages — for example, to alter the font, color, size, and spacing of your content, split it into multiple columns, or add animations and other decorative features. This module provides a gentle beginning to your path towards CSS mastery with the basics of how it works, what the syntax looks like, and how you can start using it to add styling to HTML.

Basic CSS Syntax

```
<html>
<head>
<style>
p {
      color: red;
      text-align: center;
   }
</style>
</head>
<body>
<p> Hello World! </p>
</body>
</html>
```

Why CSS?

- Avoid duplication

- Make maintenance easier
- Use the same content with different styles for different purposes

Bootstrap

What is Bootstrap?

- Bootstrap is a free front-end framework for faster and easier web development
- Bootstrap includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins
- Bootstrap also gives you the ability to easily create responsive designs

What is a Bootstrap File?

The bootstrap file contains ASCII information that permits precise specification of what files should be restored, what volume they are on, and where they are on the volume. It is a relatively compact form of specifying the information, is human readable, and can be edited with any text editor.

What does Bootstrap do?

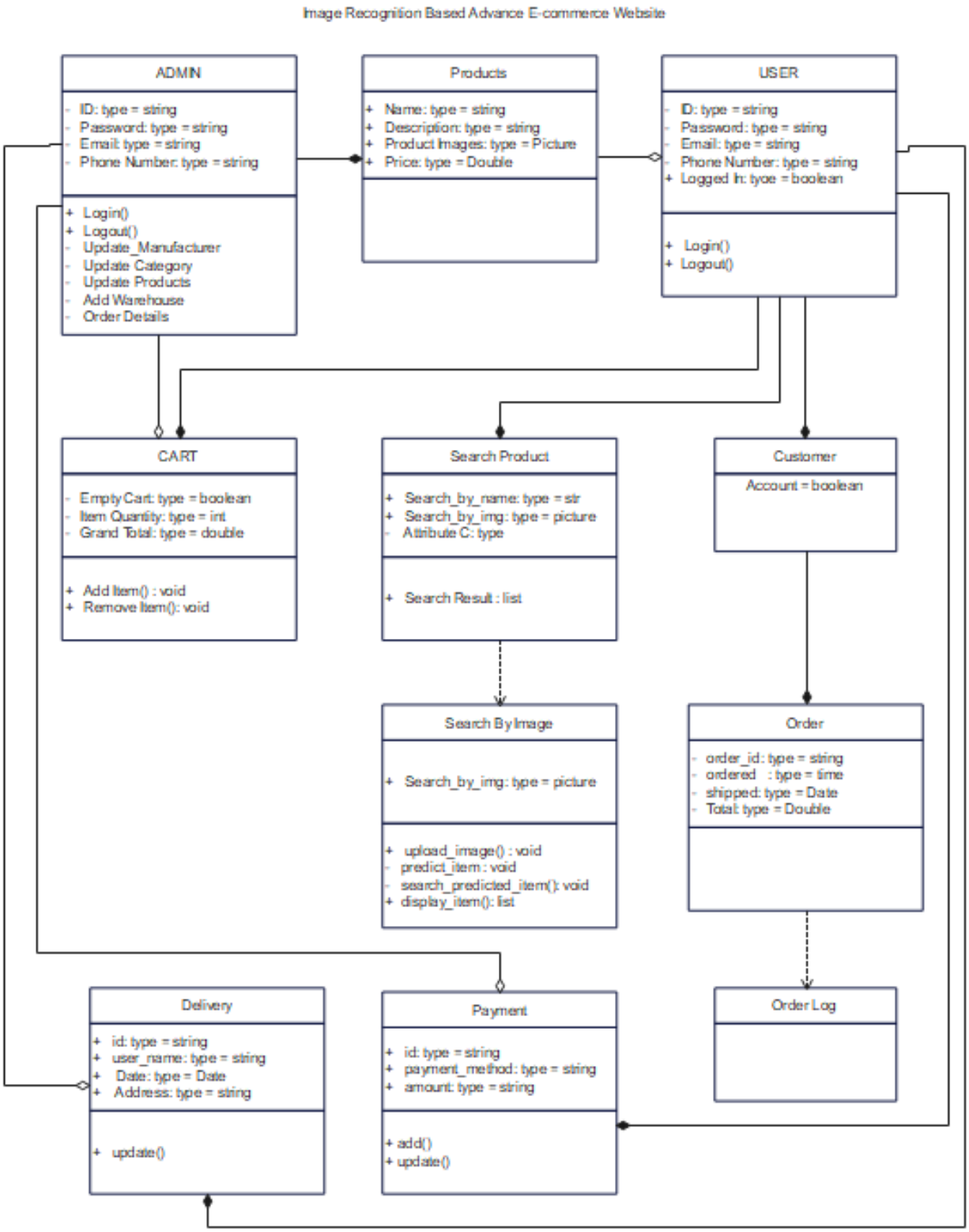
Bootstrap provides a collection of syntax for template designs for the creation of websites and web apps. As a framework, Bootstrap includes the basics for responsive web development, so developers only need to insert the code into a pre-defined grid system. The Bootstrap framework is built on Hypertext Markup Language (HTML), cascading style sheets (CSS) and JavaScript. Web developers using Bootstrap can build websites much faster without spending time worrying about basic commands and functions.

Why Bootstrap?

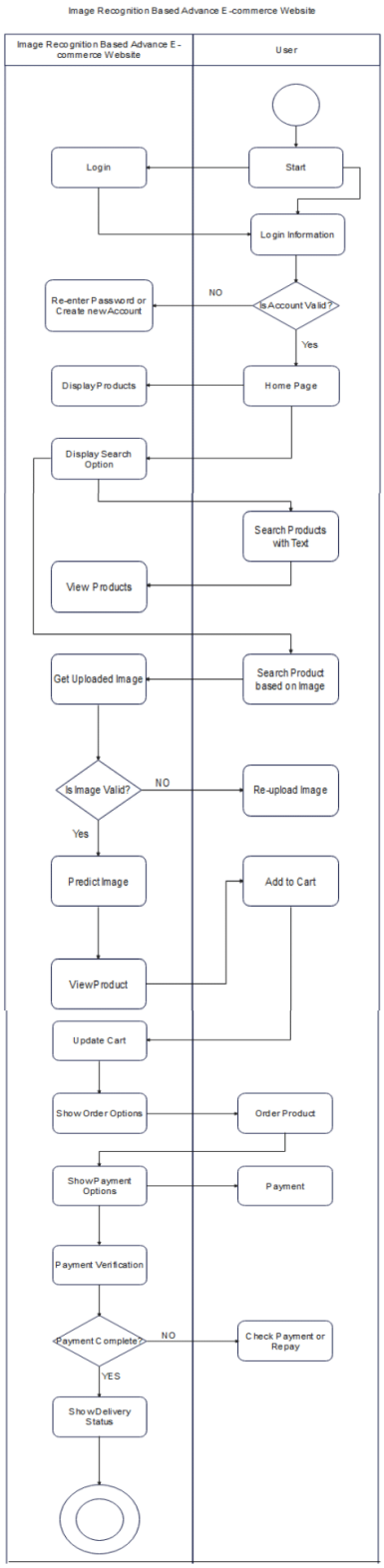
1. **Easy to use:** Anybody with just basic knowledge of HTML and CSS can start using Bootstrap
2. **Responsive features:** Bootstrap's responsive CSS adjusts to phones, tablets, and desktops
3. **Mobile-first approach:** In Bootstrap 3, mobile-first styles are part of the core framework
4. **Browser compatibility:** Bootstrap is compatible with all modern browsers (Chrome, Firefox, Internet Explorer, Edge, Safari, and Opera)

Use Case Diagram:

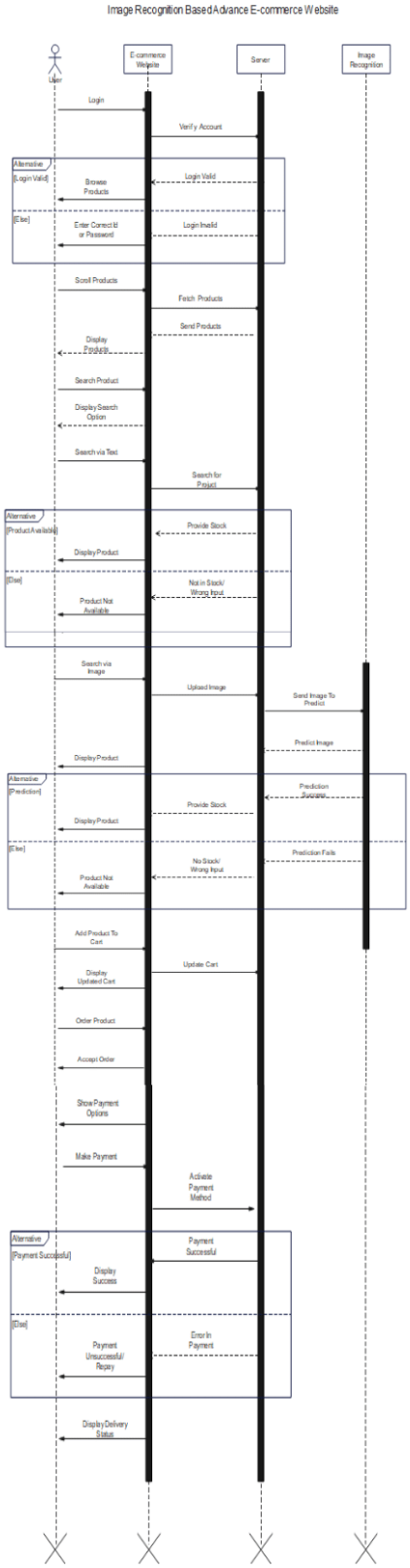
Class Diagram:



Activity Diagram:



Sequence Diagram



Testing:

Following were the tests conducted by us in the testing phase of our web application:

1. Alpha Testing:
Alpha testing is conducted by the development team before releasing the software to external users. The purpose of alpha testing is to identify any issues or bugs and to make sure that the software is ready for beta testing.

In our project, we conducted alpha testing by:

- Testing the website's features and functionality internally.
- Running the website through a variety of scenarios to identify any issues.
- Testing the website on multiple devices and browsers.
- Identifying and addressing any issues or bugs.
- Re-testing the website to ensure that the fixes were successful.
- After completing the alpha testing, we were confident that the website was ready for beta testing.

2. Beta Testing:
Beta testing is conducted by external users who are not part of the development team. The purpose of beta testing is to identify any issues or bugs that were not found during alpha testing and to gather feedback from users.

In our project, we conducted beta testing by:

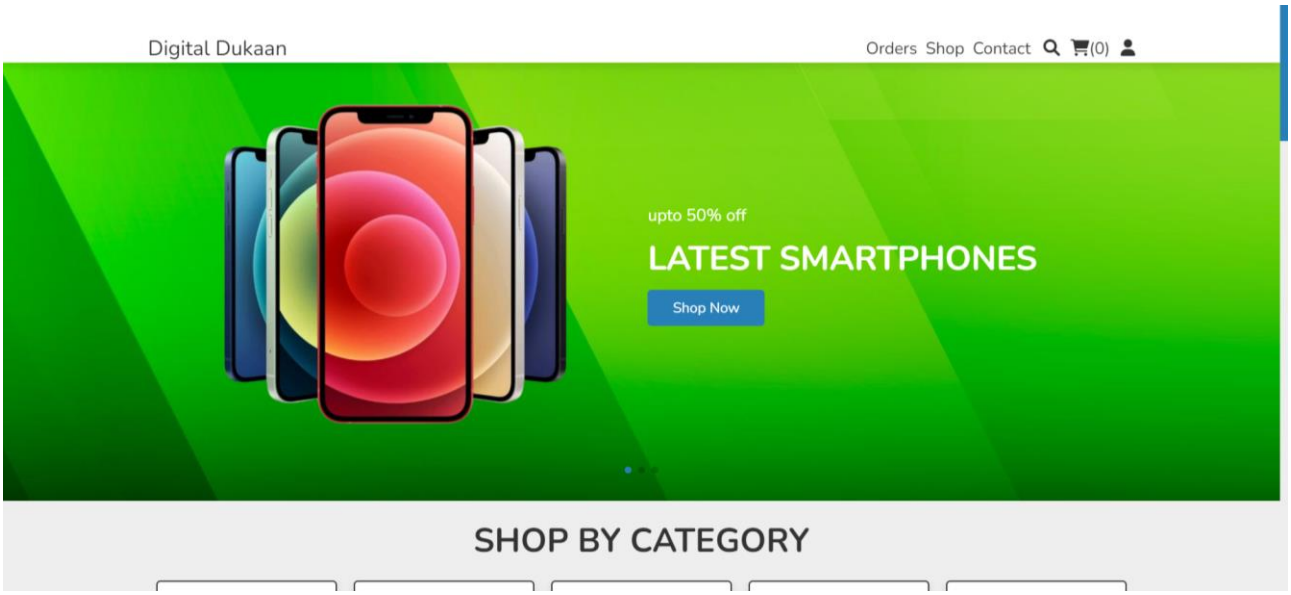
- Inviting external users to test the website.
- Providing instructions on how to use the website and what to look for during testing.
- Gathering feedback from users on their experience using the website.
- Addressing any issues or bugs that were identified during beta testing.
- Re-testing the website to ensure that the fixes were successful.
- By conducting beta testing, we were able to identify and address any issues or bugs that were not found during alpha testing.
- We also received valuable feedback from users, which helped us improve the user experience of the website.

Overall, alpha and beta testing were important steps in ensuring the quality and functionality of our website.

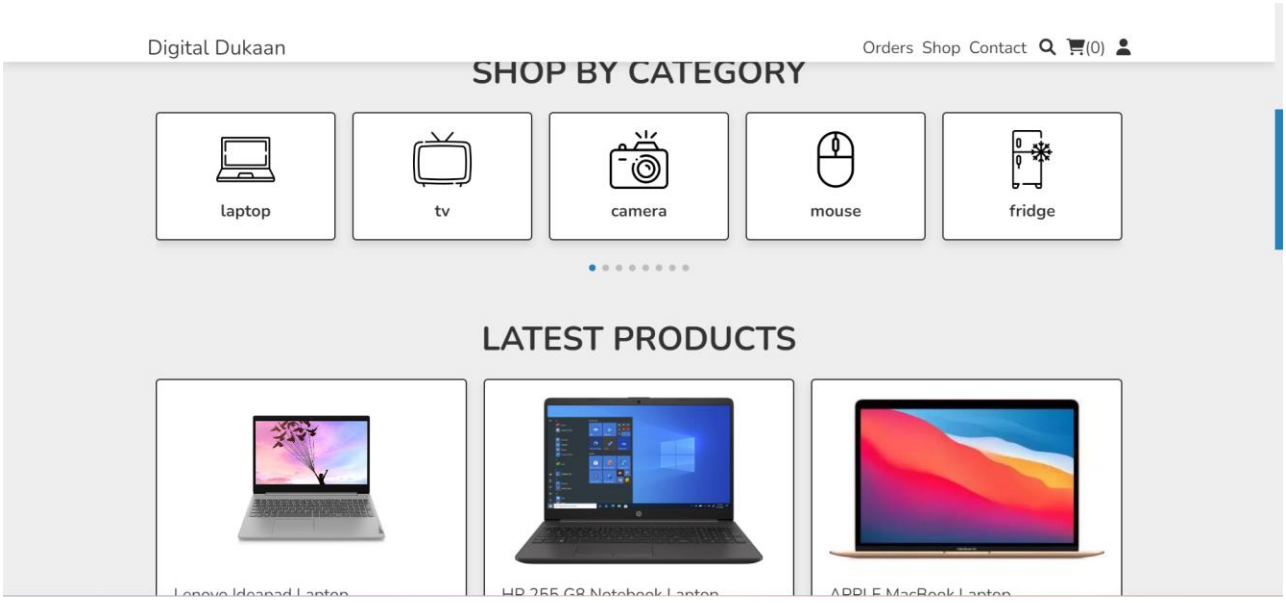
Glimpses of Final Website:

USER INTERFACE:

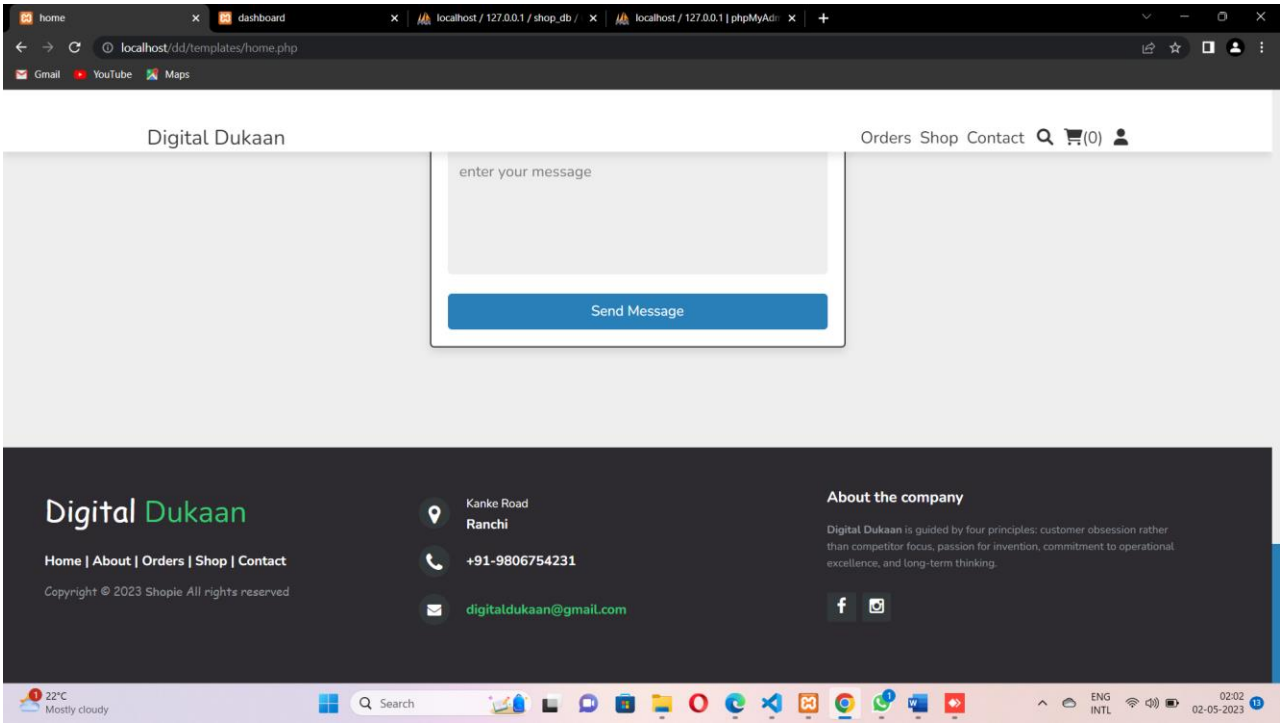
Home Page 1



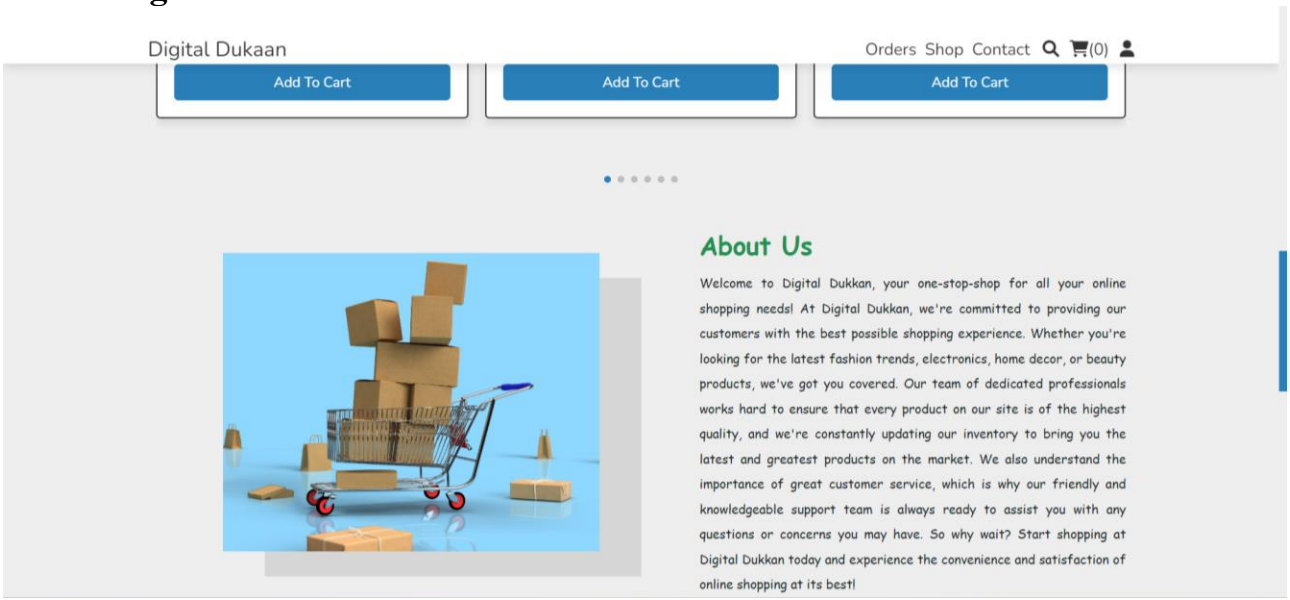
Home Page 2



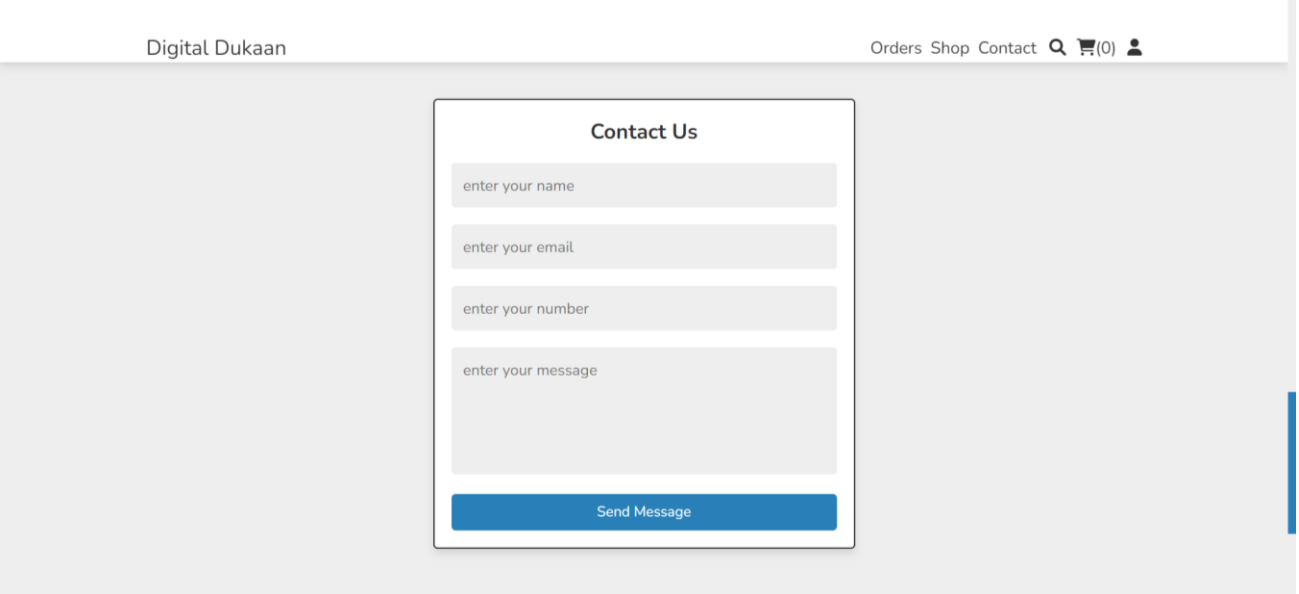
Home Page 3



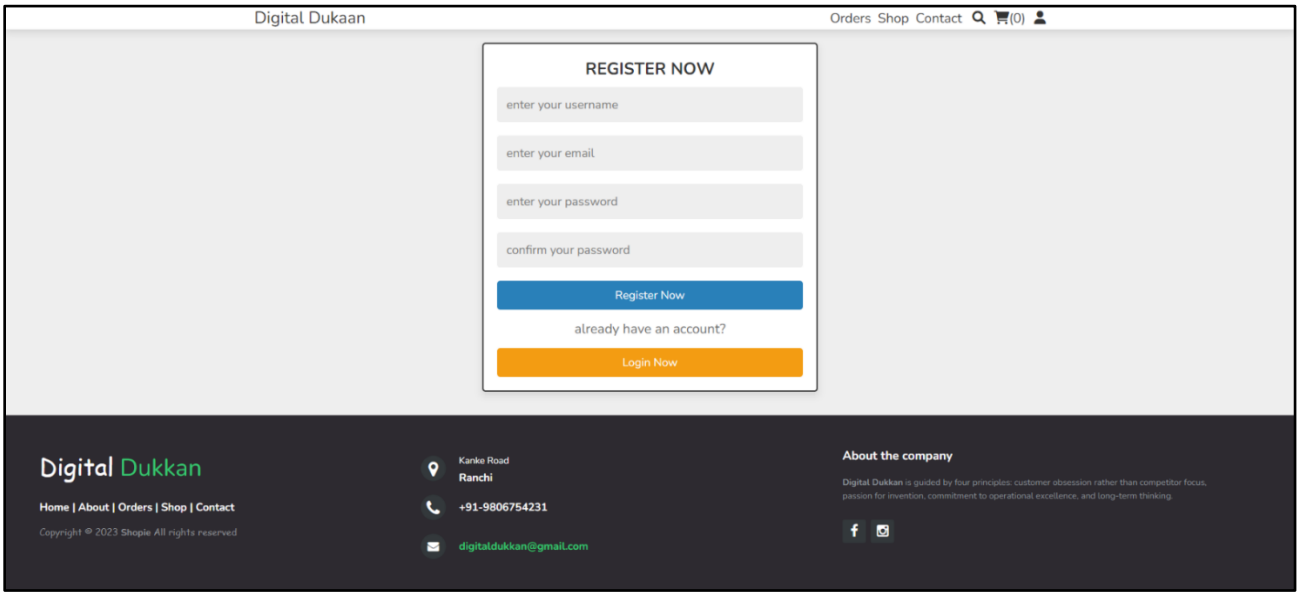
Home Page/About Us section



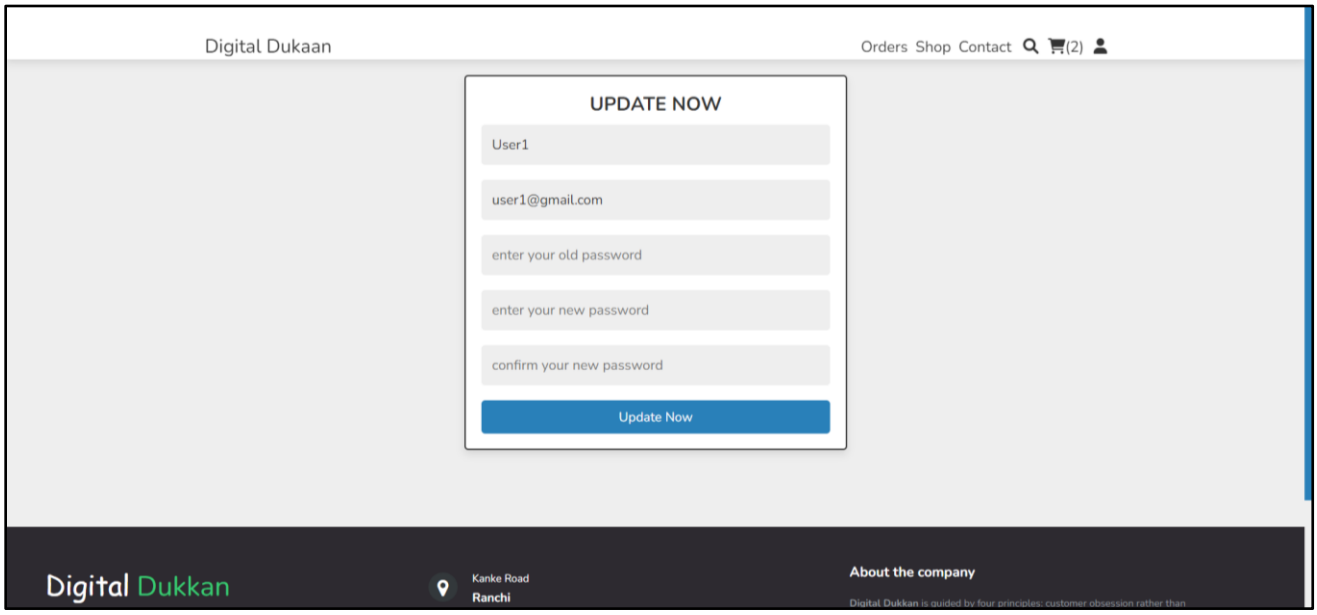
Home Page/Contact Us Section



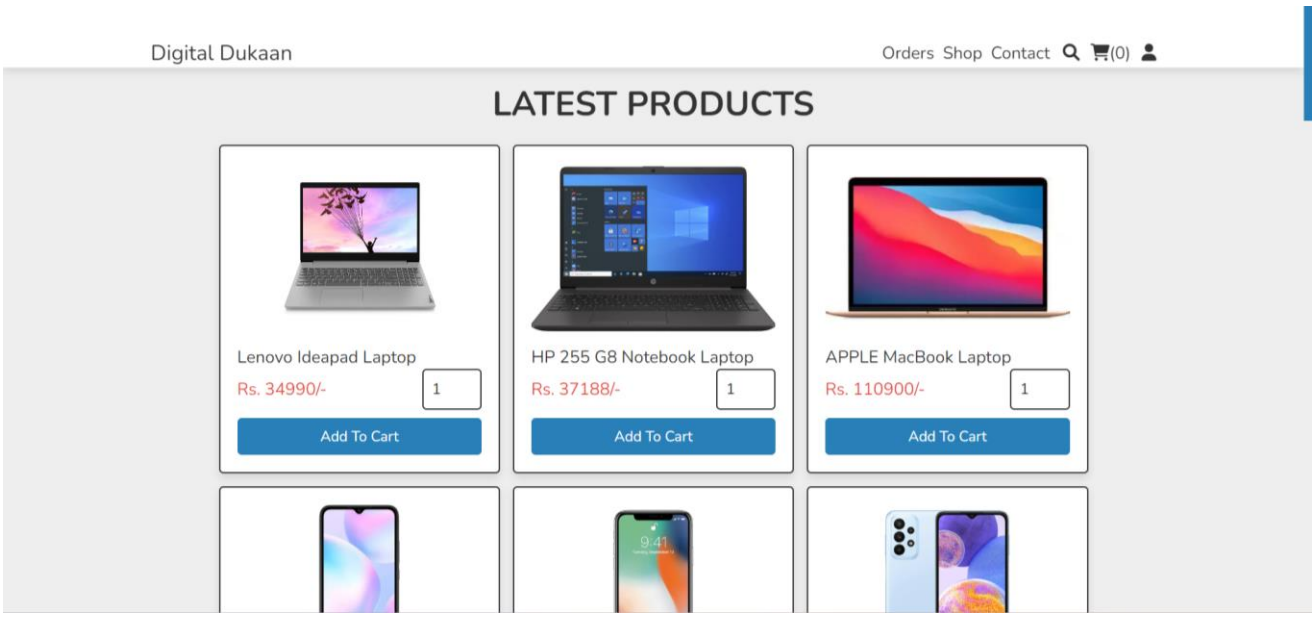
Login/Register page



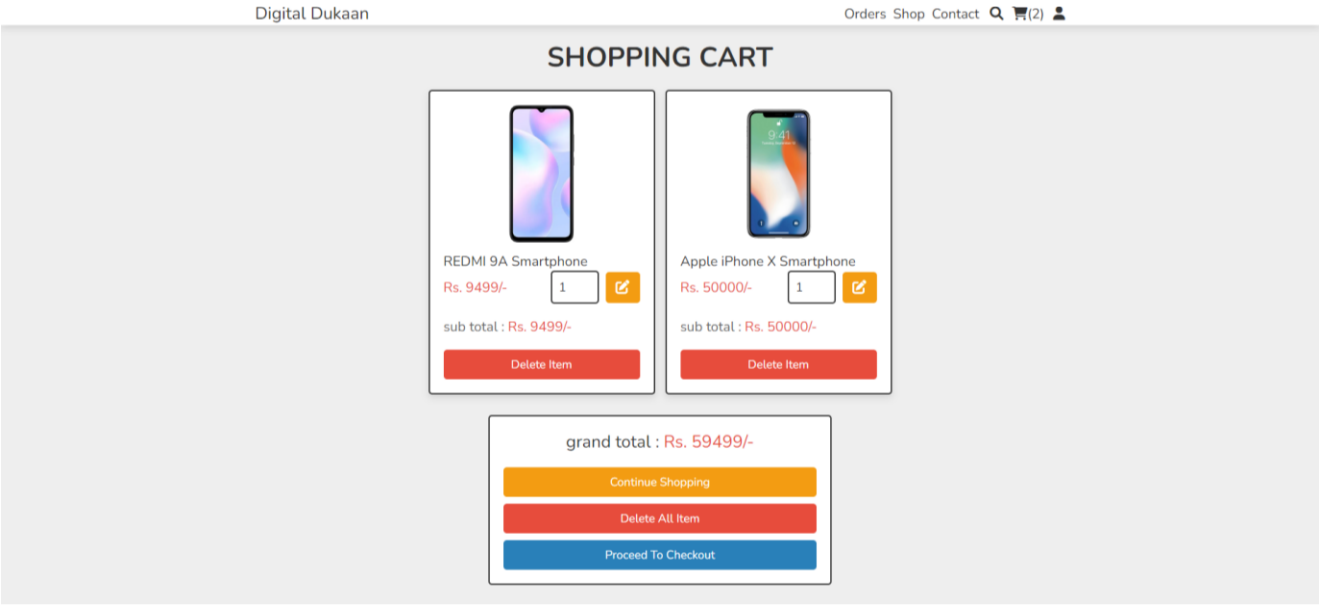
Update Page



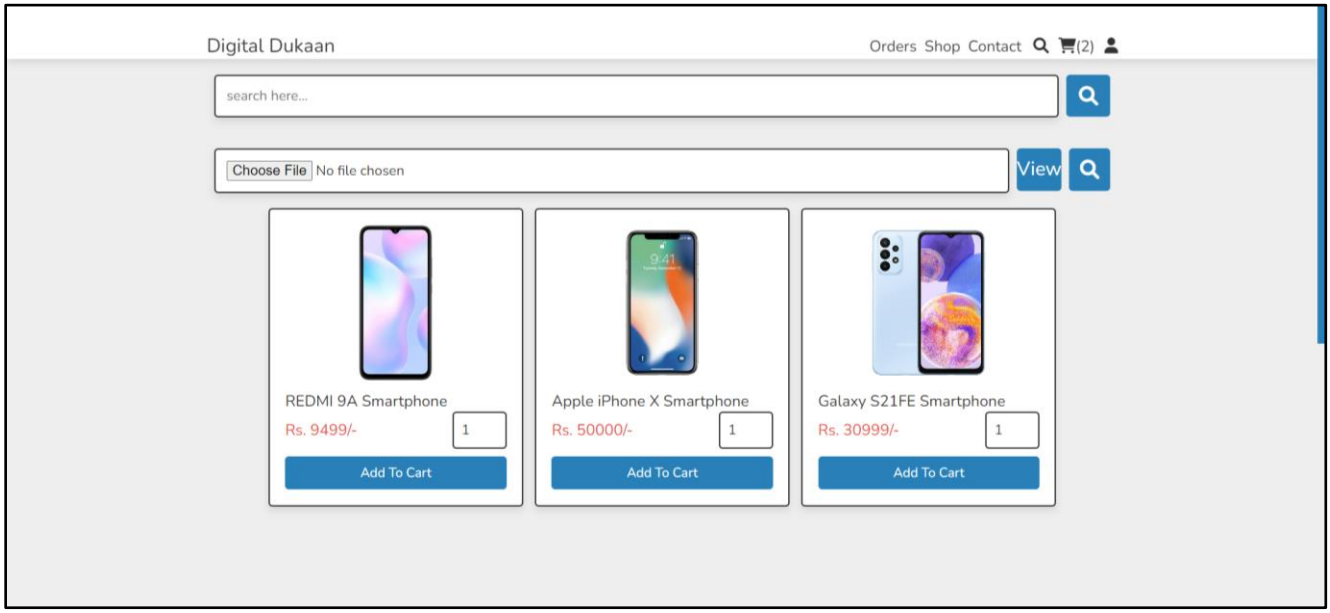
Shop page



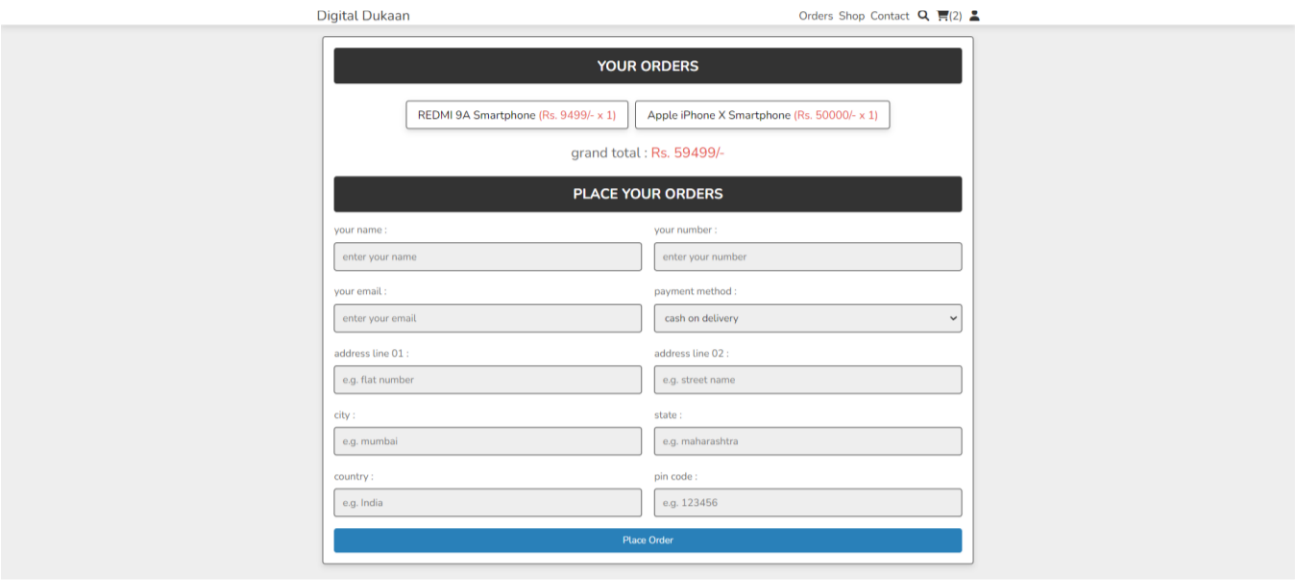
Add to Cart page



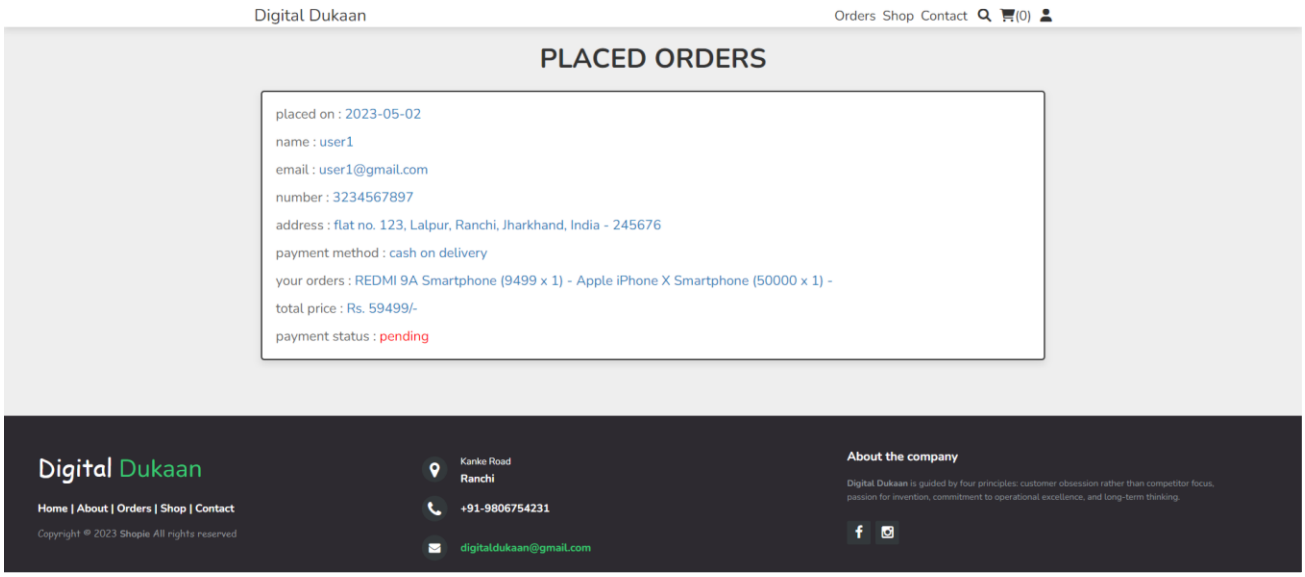
Product Search page



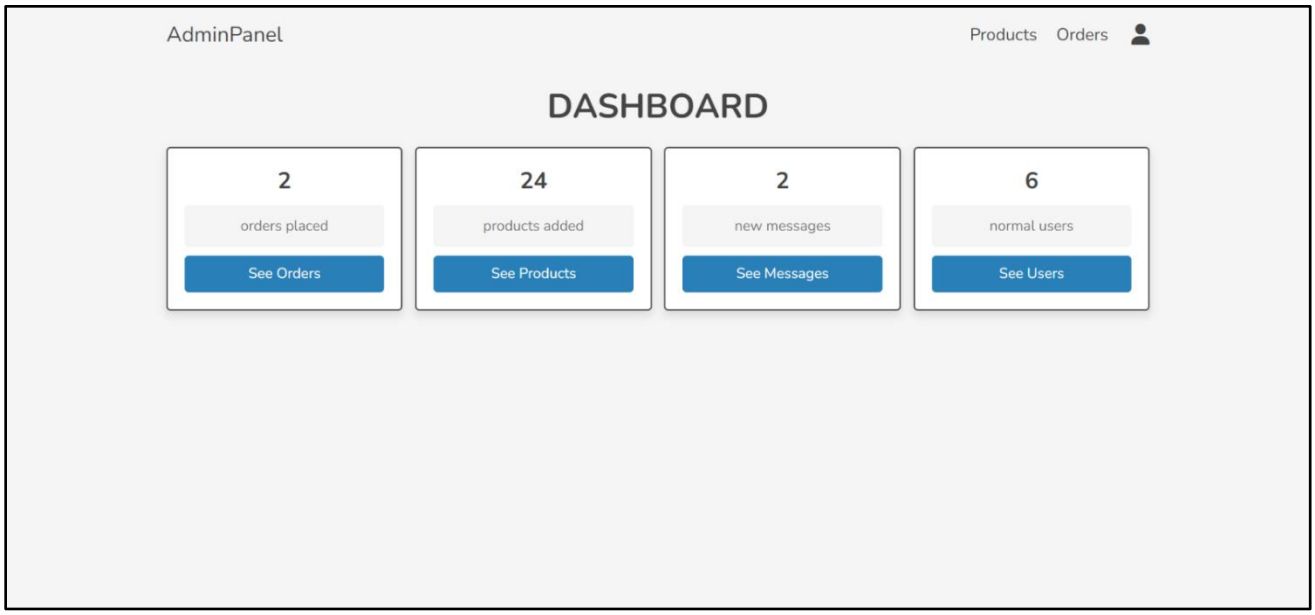
Proceed to Checkout Page



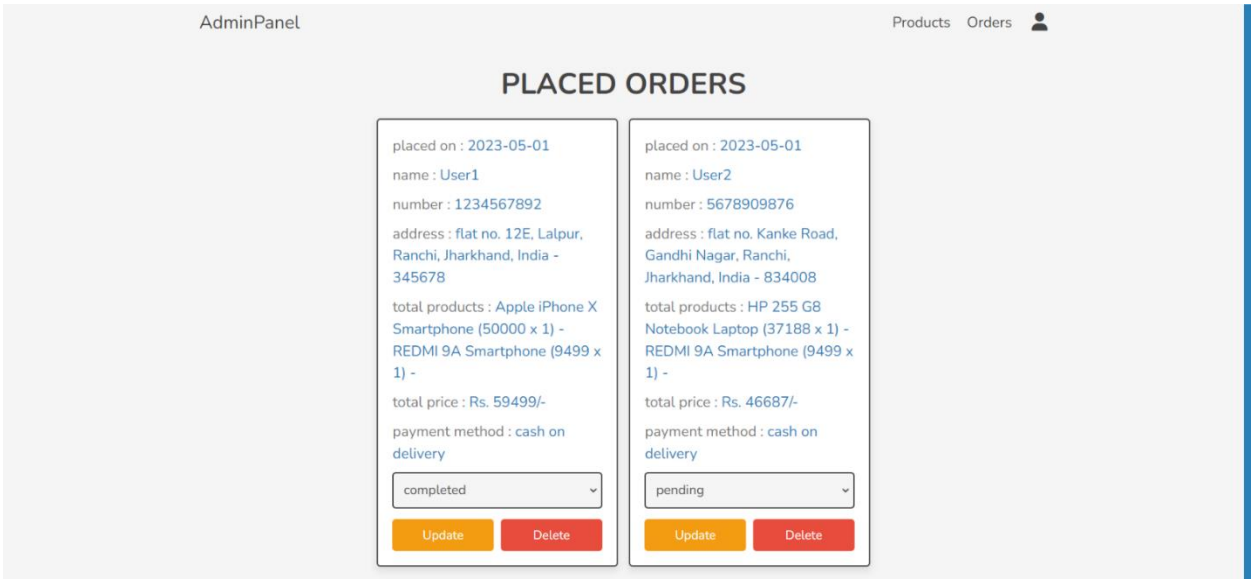
View Order page



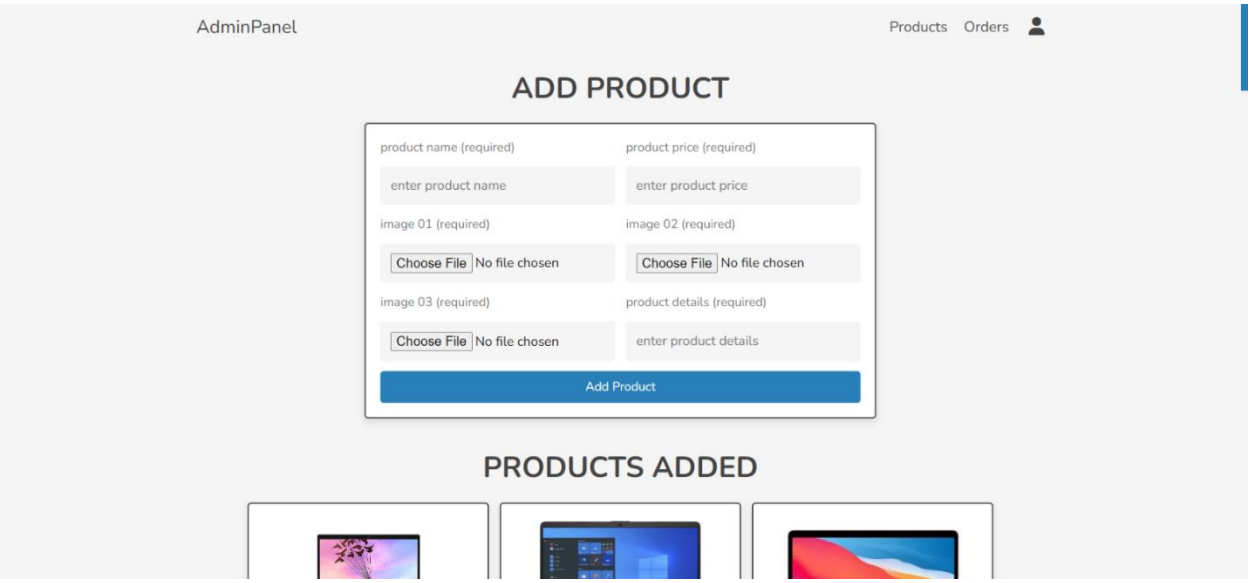
ADMIN INTERFACE:



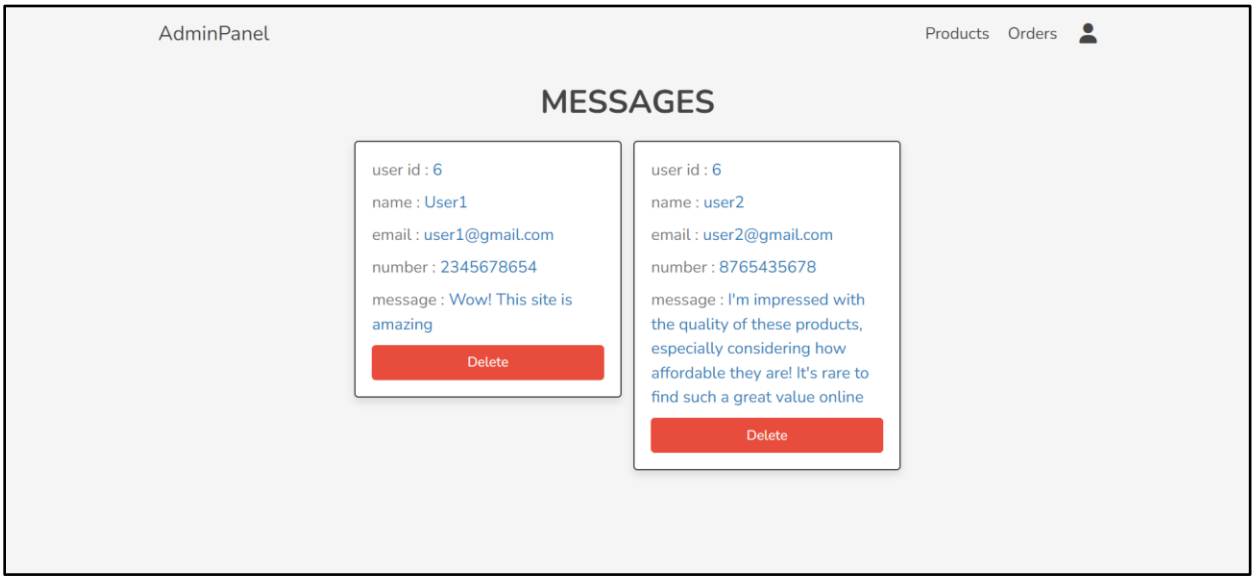
Placed Orders page



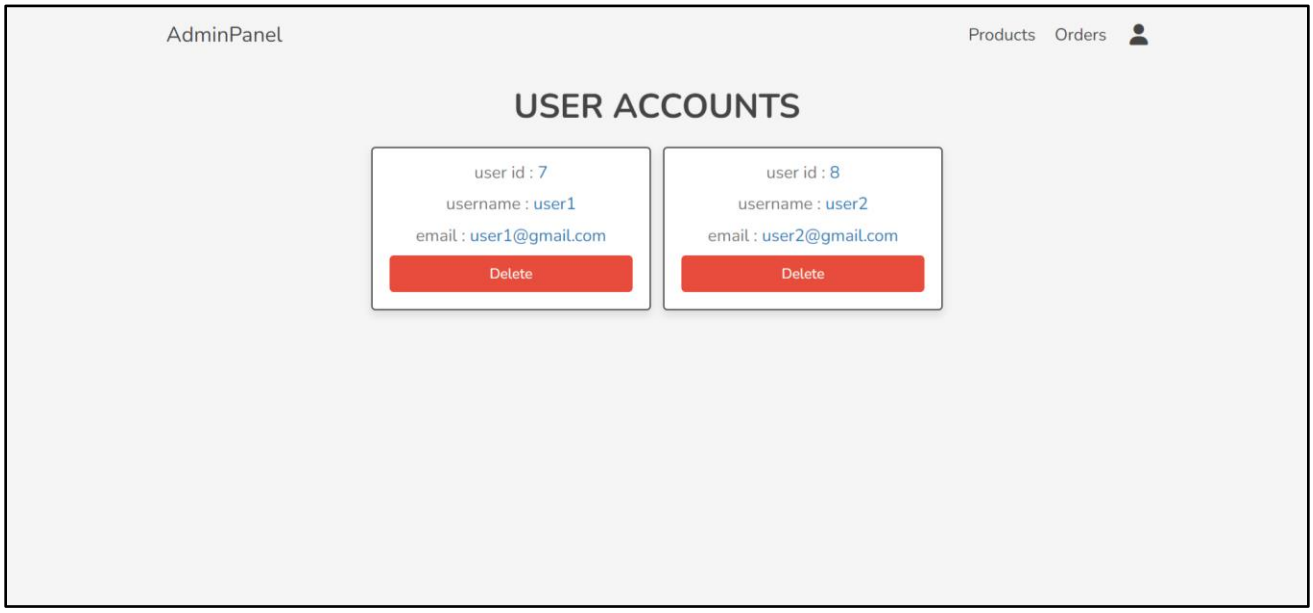
Product Page



Message Page



User Accounts Page



Codes and Algorithm

USERS SECTION

Cart.php

```
<?php
include 'components/connect.php';
session_start();
if(isset($_SESSION['user_id'])){
  $user_id = $_SESSION['user_id'];
}else{
  $user_id = '';
  header('location:user_login.php');
};
if(isset($_POST['delete'])){
  $cart_id = $_POST['cart_id'];
  $delete_cart_item = $conn->prepare("DELETE FROM `cart` WHERE id = ?");
  $delete_cart_item->execute([$cart_id]);
}
if(isset($_GET['delete_all'])){
  $delete_cart_item = $conn->prepare("DELETE FROM `cart` WHERE user_id = ?");
  $delete_cart_item->execute([$user_id]);
  header('location:cart.php');
}
if(isset($_POST['update_qty'])){
  $cart_id = $_POST['cart_id'];
  $qty = $_POST['qty'];
  $qty = filter_var($qty, FILTER_SANITIZE_STRING);
  $update_qty = $conn->prepare("UPDATE `cart` SET quantity = ? WHERE id = ?");
  $update_qty->execute([$qty, $cart_id]);
  $message[] = 'cart quantity updated';
}
?>
<!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>shopping cart</title>
  <!-- font awesome cdn link  -->
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.1/css/all.min.css">
  <!-- custom css file link  -->
  <link rel="stylesheet" href="css/style.css">
</head>
<body>
<?php include 'components/user_header.php'; ?>
<section class="products shopping-cart">
```

```
  <h3 class="heading">shopping cart</h3>
  <div class="box-container">
  <?php
    $grand_total = 0;
    $select_cart = $conn->prepare("SELECT * FROM `cart` WHERE user_id = ?");
    $select_cart->execute([$user_id]);
    if($select_cart->rowCount() > 0){
      while($fetch_cart = $select_cart->fetch(PDO::FETCH_ASSOC)){
  ?>
  <form action="" method="post" class="box">
    <input type="hidden" name="cart_id" value="<?= $fetch_cart['id']; ?>">
    <a href="quick_view.php?pid=<?= $fetch_cart['pid']; ?>" class="fas fa-eye"></a>
    <img src="uploaded_img/<?= $fetch_cart['image']; ?>" alt="">
    <div class="name"><?= $fetch_cart['name']; ?></div>
    <div class="flex">
      <div class="price">Rs. <?= $fetch_cart['price']; ?>/-</div>
      <input type="number" name="qty" class="qty" min="1" max="99" onkeypress="if(this.value.length == 2) return false;"
value="<?= $fetch_cart['quantity']; ?>">
      <button type="submit" class="fas fa-edit" name="update_qty"></button>
    </div>
    <div class="sub-total"> sub total : <span>Rs. <?= $sub_total = ($fetch_cart['price'] * $fetch_cart['quantity']); ?>/-</span>
</div>
    <input type="submit" value="delete item" onclick="return confirm('delete this from cart?');" class="delete-btn"
name="delete">
  </form>
  <?php
  $grand_total += $sub_total;
    }
  }else{
    echo '<p class="empty">your cart is empty</p>';
  }
  ?>
  </div>
  <div class="cart-total">
    <p>grand total : <span>Rs. <?= $grand_total; ?>/-</span></p>
    <a href="shop.php" class="option-btn">continue shopping</a>
    <a href="cart.php?delete_all" class="delete-btn <?= ($grand_total > 1)?'':'disabled'; ?>" onclick="return confirm('delete all
from cart?');">delete all item</a>
    <a href="checkout.php" class="btn <?= ($grand_total > 1)?'':'disabled'; ?>">proceed to checkout</a>
  </div>
</section>
<?php include 'components/footer.php'; ?>
<script src="js/script.js"></script>
</body>
</html>
```

Home.php

```
<?php
include 'components/connect.php';
session_start();
if(isset($_SESSION['user_id'])){
  $user_id = $_SESSION['user_id'];
}else{
  $user_id = '';
};
include 'components/wishlist_cart.php';
```

```
?>
<!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>home</title>
  <link rel="stylesheet" href="https://unpkg.com/swiper@8/swiper-bundle.min.css" />
  <!-- font awesome cdn link  -->
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.1/css/all.min.css">
  <!-- custom css file link  -->
  <link rel="stylesheet" href="css/style.css">
</head>
<body>
<?php include 'components/user_header.php'; ?>
<div class="home-bg">
<section class="home">
  <div class="swiper home-slider">
  <div class="swiper-wrapper">
  <div class="swiper-slide slide">
      <div class="image">
        <img src="images/home-img-2.png" alt="">
      </div>
      <div class="content">
        <span>upto 50% off</span>
        <h3>latest watches</h3>
        <a href="shop.php" class="btn">shop now</a>
      </div>
    </div>
    <div class="swiper-slide slide">
      <div class="image">
        <img src="images/home-img-1.png" alt="">
      </div>
      <div class="content">
        <span>upto 50% off</span>
        <h3>latest smartphones</h3>
        <a href="shop.php" class="btn">shop now</a>
      </div>
    </div>
    <div class="swiper-slide slide">
      <div class="image">
        <img src="images/home-img-3.png" alt="">
      </div>
      <div class="content">
        <span>upto 50% off</span>
        <h3>latest headsets</h3>
        <a href="shop.php" class="btn">shop now</a>
      </div>
    </div>
  </div>
    <div class="swiper-pagination"></div>
  </div>
</section>
</div>
<section class="category">
  <h1 class="heading">shop by category</h1>
  <div class="swiper category-slider">
```

```
  <div class="swiper-wrapper">
  <a href="category.php?category=laptop" class="swiper-slide slide">
    <img src="images/icon-1.png" alt="">
    <h3>laptop</h3>
  </a>
  <a href="category.php?category=tv" class="swiper-slide slide">
    <img src="images/icon-2.png" alt="">
    <h3>tv</h3>
  </a>
  <a href="category.php?category=camera" class="swiper-slide slide">
    <img src="images/icon-3.png" alt="">
    <h3>camera</h3>
  </a>
  <a href="category.php?category=mouse" class="swiper-slide slide">
    <img src="images/icon-4.png" alt="">
    <h3>mouse</h3>
  </a>
  <a href="category.php?category=fridge" class="swiper-slide slide">
    <img src="images/icon-5.png" alt="">
    <h3>fridge</h3>
  </a>
  <a href="category.php?category=washing" class="swiper-slide slide">
    <img src="images/icon-6.png" alt="">
    <h3>washing machine</h3>
  </a>
  <a href="category.php?category=smartphone" class="swiper-slide slide">
    <img src="images/icon-7.png" alt="">
    <h3>smartphone</h3>
  </a>
  <a href="category.php?category=watch" class="swiper-slide slide">
    <img src="images/icon-8.png" alt="">
    <h3>watch</h3>
  </a>
  </div>
  <div class="swiper-pagination"></div>
  </div>
</section>
<section class="home-products">
  <h1 class="heading">latest products</h1>
  <div class="swiper products-slider">
  <div class="swiper-wrapper">
  <?php
    $select_products = $conn->prepare("SELECT * FROM `products` LIMIT 6");
    $select_products->execute();
    if($select_products->rowCount() > 0){
     while($fetch_product = $select_products->fetch(PDO::FETCH_ASSOC)){
  ?>
  <form action="" method="post" class="swiper-slide slide">
    <input type="hidden" name="pid" value="<?= $fetch_product['id']; ?>">
    <input type="hidden" name="name" value="<?= $fetch_product['name']; ?>">
    <input type="hidden" name="price" value="<?= $fetch_product['price']; ?>">
    <input type="hidden" name="image" value="<?= $fetch_product['image_01']; ?>">
    <a href="quick_view.php?pid=<?= $fetch_product['id']; ?>" class="fas fa-eye"></a>
    <img src="uploaded_img/<?= $fetch_product['image_01']; ?>" alt="">
    <div class="name"><?= $fetch_product['name']; ?></div>
    <div class="flex">
      <div class="price"><span>Rs. </span><?= $fetch_product['price']; ?><span>/-</span></div>
```

```
      <input type="number" name="qty" class="qty" min="1" max="99" onkeypress="if(this.value.length == 2) return false;"
value="1">
    </div>
    <input type="submit" value="add to cart" class="btn" name="add_to_cart">
  </form>
  <?php
    }
  }else{
    echo '<p class="empty">no products added yet!</p>';
  }
  ?>
  </div>
  <section id="about">
</section>
  <div class="swiper-pagination"></div>
  </div>
</section>
<!-- About Section -->
<section class="about">
<div class="image">
<img src="images/5.jpg" alt="">
</div>
  <div class="content">
      <h3>About Us</h3>
      <p align="justify">Welcome to Walmart, your one-stop-shop for all your online shopping needs!
At Walmart, we're committed to providing our customers with the best possible shopping experience. Whether you're looking
for the latest fashion trends, electronics, home decor, or beauty products, we've got you covered.
Our team of dedicated professionals works hard to ensure that every product on our site is of the highest quality, and we're
constantly updating our inventory to bring you the latest and greatest products on the market.
We also understand the importance of great customer service, which is why our friendly and knowledgeable support team is
always ready to assist you with any questions or concerns you may have.
So why wait? Start shopping at Walmart today and experience the convenience and satisfaction of online shopping at its
best!</p>
      <a href="#contact" class="button"></a>
   </div>
</section>
<!-- About Section Ends -->
<section id="contact">
</section>
<!-- Contact -->
<section class="contact1">
<?php
include 'components/connect.php';
if(isset($_SESSION['user_id'])){
  $user_id = $_SESSION['user_id'];
}else{
  $user_id = '';
};
if(isset($_POST['send'])){
  $name = $_POST['name'];
  $name = filter_var($name, FILTER_SANITIZE_STRING);
  $email = $_POST['email'];
  $email = filter_var($email, FILTER_SANITIZE_STRING);
  $number = $_POST['number'];
  $number = filter_var($number, FILTER_SANITIZE_STRING);
  $msg = $_POST['msg'];
  $msg = filter_var($msg, FILTER_SANITIZE_STRING);
```

```
  $select_message = $conn->prepare("SELECT * FROM `messages` WHERE name = ? AND email = ? AND number = ? AND
message = ?");
  $select_message->execute([$name, $email, $number, $msg]);
  if($select_message->rowCount() > 0){
    $message[] = 'already sent message!';
  }else{
    $insert_message = $conn->prepare("INSERT INTO `messages`(user_id, name, email, number, message) VALUES(?,?,?,?,?)");
    $insert_message->execute([$user_id, $name, $email, $number, $msg]);
    $message[] = 'Message Sent Successfully!';
  }
}
if(isset($_GET['success']) && $_GET['success'] == 'true'){
  // display success message
  echo '<div class="success-message">' . $_SESSION['success_message'] . '</div>';
}
?>
<section class="contact" id="contact">
<form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>#contact" method="post" autocomplete="off"
id="contact-form">
    <h3>Contact Us</h3>
    <input type="text" name="name" placeholder="enter your name" required maxlength="20" class="box">
    <input type="email" name="email" placeholder="enter your email" required maxlength="50" class="box">
    <input type="number" name="number" min="0" max="9999999999" placeholder="enter your number" required
onkeypress="if(this.value.length == 10) return false;" class="box">
    <textarea name="msg" class="box" placeholder="enter your message" cols="30" rows="10"></textarea>
    <input type="submit" value="send message" name="send" class="btn">
    <?php include 'components/user_header1.php'; ?>
  </form>
</section>
</section>
</section>
<?php include 'components/footer.php'; ?>
<script src="js/script.js"></script>
<script src="https://unpkg.com/swiper@8/swiper-bundle.min.js"></script>
<script>
var swiper = new Swiper(".home-slider", {
  loop:true,
  spaceBetween: 20,
  pagination: {
    el: ".swiper-pagination",
    clickable:true,
   },
});
 var swiper = new Swiper(".category-slider", {
  loop:true,
  spaceBetween: 20,
  pagination: {
    el: ".swiper-pagination",
    clickable:true,
  },
  breakpoints: {
    0: {
      slidesPerView: 2,
     },
    650: {
     slidesPerView: 3,
    },
    768: {
```

```
      slidesPerView: 4,
    },
    1024: {
      slidesPerView: 5,
    },
  },
});
var swiper = new Swiper(".products-slider", {
  loop:true,
  spaceBetween: 20,
  pagination: {
    el: ".swiper-pagination",
    clickable:true,
  },
  breakpoints: {
    550: {
      slidesPerView: 2,
    },
    768: {
      slidesPerView: 2,
    },
    1024: {
      slidesPerView: 3,
    },
  },
});
</script>
<script>
success: function(response) {
  // handle success response (e.g. show success message)
  console.log(response);
  window.location.hash = '#contact';
},
</script>
</div>
</body>
</html>
```

Search_page.php

```
<?php
include 'components/connect.php';
session_start();
if(isset($_SESSION['user_id'])){
  $user_id = $_SESSION['user_id'];
}else{
  $user_id = '';
};
?>
<?php
if(isset($_SESSION['message'])){
  $message = $_SESSION['message'];
  unset($_SESSION['message']);
}else{
  $message = '';
};
?>
```

```
<!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>search page</title>
  <!-- font awesome cdn link  -->
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.1/css/all.min.css">
  <!-- custom css file link  -->
  <link rel="stylesheet" href="css/style.css">
</head>
<body>
<?php include 'components/user_header.php'; ?>
<section class="search-form">
  <form action="" method="post">
    <input id="toSearch" type="text" name="search_box" placeholder="search here..." maxlength="100" class="box" required>
    <button type="submit" id="searchSubmit" class="fas fa-search" name="search_btn"></button>
  </form>
</section>
<section class="search-form">
  <form id="upload-form">
    <input type="file" name="file" id="file">
    <button type="button" onclick="previewImage()">View</button>
    <button type="submit" class="fas fa-search" name="search_btn"></button>
  </form>
  <?php if(!empty($message)): ?>
    <div id="message"><?= $message ?></div>
  <?php endif; ?>
</section>
<script type="text/javascript">
      //"http://127.0.0.1:5000/predict_image"
      document.querySelector("#upload-form").addEventListener("submit",async (event)=>{
        event.preventDefault();
        const fileInput=document.querySelector("#file")
        console.log("the is log")
        const data = new FormData();
        data.append("file",fileInput.files[0]);
        let res=await fetch("http://127.0.0.1:5000/predict_image",{
          body:data,
          method:"post"
        });
        console.log("run till now")
        let search=await res.json();
        document.querySelector("#toSearch").value=search.name;
        document.querySelector("#searchSubmit").click();
      });
 function previewImage() {
 var file = document.getElementById("file").files[0];
 var reader = new FileReader();

 reader.onloadend = function () {
  var preview = window.open("", "Image Preview", "width=500,height=500");
  preview.document.write("<img src='" + reader.result + "' style='max-width: 100%; max-height: 100%; object-fit: contain;'>");
}
 if (file) {
  reader.readAsDataURL(file);
 }
```

```
}
 </script>
<section class="products" style="padding-top: 0; min-height:100vh;">
  <div class="box-container">
  <?php
    if(isset($_POST['search_box']) OR isset($_POST['search_btn'])){
    $search_box = $_POST['search_box'];
    $select_products = $conn->prepare("SELECT * FROM `products` WHERE name LIKE '%{$search_box}%'");
    $select_products->execute();
    if($select_products->rowCount() > 0){
     while($fetch_product = $select_products->fetch(PDO::FETCH_ASSOC)){
  ?>
  <form action="" method="post" class="box">
    <input type="hidden" name="pid" value="<?= $fetch_product['id']; ?>">
    <input type="hidden" name="name" value="<?= $fetch_product['name']; ?>">
    <input type="hidden" name="price" value="<?= $fetch_product['price']; ?>">
    <input type="hidden" name="image" value="<?= $fetch_product['image_01']; ?>">
    <a href="quick_view.php?pid=<?= $fetch_product['id']; ?>" class="fas fa-eye"></a>
    <img src="uploaded_img/<?= $fetch_product['image_01']; ?>" alt="">
    <div class="name"><?= $fetch_product['name']; ?></div>
    <div class="flex">
      <div class="price"><span>Rs. </span><?= $fetch_product['price']; ?><span>/-</span></div>
      <input type="number" name="qty" class="qty" min="1" max="99" onkeypress="if(this.value.length == 2) return false;"
value="1">
    </div>
    <input type="submit" value="add to cart" class="btn" name="add_to_cart">
  </form>
  <?php
       }
    }else{
      echo '<p class="empty">no products found!</p>';
    }
  }
  ?>
  </div>
</section>
<?php include 'components/footer.php'; ?>
<script src="js/script.js">
</script>
</body>
</html>
```

Shop.php

```
<?php
include 'components/connect.php';
session_start();
if(isset($_SESSION['user_id'])){
  $user_id = $_SESSION['user_id'];
}else{
  $user_id = '';
};
?>
<!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>shop</title>
  <!-- font awesome cdn link  -->
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.1/css/all.min.css">
  <!-- custom css file link  -->
  <link rel="stylesheet" href="css/style.css">
</head>
<body>
<?php include 'components/user_header.php'; ?>
<section class="products">
  <h1 class="heading">latest products</h1>
  <div class="box-container">
  <?php
   $select_products = $conn->prepare("SELECT * FROM `products`");
   $select_products->execute();
   if($select_products->rowCount() > 0){
    while($fetch_product = $select_products->fetch(PDO::FETCH_ASSOC)){
  ?>
  <form action="" method="post" class="box">
    <input type="hidden" name="pid" value="<?= $fetch_product['id']; ?>">
    <input type="hidden" name="name" value="<?= $fetch_product['name']; ?>">
    <input type="hidden" name="price" value="<?= $fetch_product['price']; ?>">
    <input type="hidden" name="image" value="<?= $fetch_product['image_01']; ?>">
    <a href="quick_view.php?pid=<?= $fetch_product['id']; ?>" class="fas fa-eye"></a>
    <img src="uploaded_img/<?= $fetch_product['image_01']; ?>" alt="">
    <div class="name"><?= $fetch_product['name']; ?></div>
    <div class="flex">
      <div class="price"><span>Rs. </span><?= $fetch_product['price']; ?><span>/-</span></div>
      <input type="number" name="qty" class="qty" min="1" max="99" onkeypress="if(this.value.length == 2) return false;"
value="1">
    </div>
    <input type="submit" value="add to cart" class="btn" name="add_to_cart">
    <?php include 'components/user_header1.php'; ?>
  </form>
  <?php
    }
  }else{
    echo '<p class="empty">no products found!</p>';
  }
  ?>
  </div>
</section>
<?php include 'components/footer.php'; ?>
<script src="js/script.js"></script>
</body>
</html>
```

ADMIN SECTION

Admin_login.php

```
<?php
include '../components/connect.php';
session_start();
if(isset($_POST['submit'])){
  $name = $_POST['name'];
  $name = filter_var($name, FILTER_SANITIZE_STRING);
  $pass = sha1($_POST['pass']);
  $pass = filter_var($pass, FILTER_SANITIZE_STRING);
  $select_admin = $conn->prepare("SELECT * FROM `admins` WHERE name = ? AND password = ?");
  $select_admin->execute([$name, $pass]);
  $row = $select_admin->fetch(PDO::FETCH_ASSOC);
  if($select_admin->rowCount() > 0){
    $_SESSION['admin_id'] = $row['id'];
    header('location:dashboard.php');
  }else{
    $message[] = 'incorrect username or password!';
  }
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>login</title>
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.1/css/all.min.css">
  <link rel="stylesheet" href="../css/admin_style.css">
</head>
<body>
<section class="form-container">
  <form action="" method="post">
    <h3>login now</h3>
    <input type="text" name="name" required placeholder="enter your username" maxlength="20"  class="box"
oninput="this.value = this.value.replace(/\s/g, '')">
    <input type="password" name="pass" required placeholder="enter your password" maxlength="20"  class="box"
oninput="this.value = this.value.replace(/\s/g, '')">
    <input type="submit" value="login now" class="btn" name="submit">
    <?php
  if(isset($message)){
    foreach($message as $message){
      echo '
      <div class="message">
        <span>'.$message.'</span>
      </div>
      ';
    }
    echo '
    <script>
      setTimeout(function(){
        var messages = document.getElementsByClassName("message");
        for(var i=0; i<messages.length; i++){
```

```
            messages[i].remove();
          }
        }, 4000);
    </script>
    ';
  }
?>
  </form>
</section>
</body>
</html>
```

Dashboard.php

```
<?php
include '../components/connect.php';
session_start();
$admin_id = $_SESSION['admin_id'];
if(!isset($admin_id)){
  header('location:admin_login.php');
}
?>
<!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>dashboard</title>
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.1/css/all.min.css">
  <link rel="stylesheet" href="../css/admin_style.css">
</head>
<body>
<?php include '../components/admin_header.php'; ?>
<section class="dashboard">
  <h1 class="heading">dashboard</h1>
  <div class="box-container">
    <div class="box">
      <?php
        $select_orders = $conn->prepare("SELECT * FROM `orders`");
        $select_orders->execute();
        $number_of_orders = $select_orders->rowCount()
      ?>
      <h3><?= $number_of_orders; ?></h3>
      <p>orders placed</p>
      <a href="placed_orders.php" class="btn">see orders</a>
    </div>
   <div class="box">
      <?php
        $select_products = $conn->prepare("SELECT * FROM `products`");
        $select_products->execute();
        $number_of_products = $select_products->rowCount()
      ?>
      <h3><?= $number_of_products; ?></h3>
      <p>products added</p>
      <a href="products.php" class="btn">see products</a>
    </div>
```

```
    <div class="box">
    <?php
        $select_messages = $conn->prepare("SELECT * FROM `messages`");
        $select_messages->execute();
        $number_of_messages = $select_messages->rowCount()
      ?>
      <h3><?= $number_of_messages; ?></h3>
      <p>new messages</p>
      <a href="messages.php" class="btn">see messages</a>
    </div>
    <div class="box">
      <?php
        $select_users = $conn->prepare("SELECT * FROM `users`");
        $select_users->execute();
        $number_of_users = $select_users->rowCount()
      ?>
      <h3><?= $number_of_users; ?></h3>
      <p>normal users</p>
      <a href="users_accounts.php" class="btn">see users</a>
    </div>
  </div>
</section>
<script src="../js/admin_script.js"></script>
</body>
</html>
```

Messages.php

```
<?php
include '../components/connect.php';
session_start();
$admin_id = $_SESSION['admin_id'];
if(!isset($admin_id)){
   header('location:admin_login.php');
};
if(isset($_GET['delete'])){
   $delete_id = $_GET['delete'];
   $delete_message = $conn->prepare("DELETE FROM `messages` WHERE id = ?");
   $delete_message->execute([$delete_id]);
   header('location:messages.php');
}
?>
<!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>messages</title>
   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.1/css/all.min.css">
   <link rel="stylesheet" href="../css/admin_style.css">
</head>
<body>
<?php include '../components/admin_header.php'; ?>
<section class="contacts">
<h1 class="heading">messages</h1>
<div class="box-container">
```

```
  <?php
    $select_messages = $conn->prepare("SELECT * FROM `messages`");
    $select_messages->execute();
    if($select_messages->rowCount() > 0){
      while($fetch_message = $select_messages->fetch(PDO::FETCH_ASSOC)){
  ?>
  <div class="box">
  <p> user id : <span><?= $fetch_message['user_id']; ?></span></p>
  <p> name : <span><?= $fetch_message['name']; ?></span></p>
  <p> email : <span><?= $fetch_message['email']; ?></span></p>
  <p> number : <span><?= $fetch_message['number']; ?></span></p>
  <p> message : <span><?= $fetch_message['message']; ?></span></p>
  <a href="messages.php?delete=<?= $fetch_message['id']; ?>" onclick="return confirm('delete this message?');" class="delete-
btn">delete</a>
  </div>
  <?php
      }
    }else{
      echo '<p class="empty">you have no messages</p>';
    }
  ?>
</div>
</section>
<script src="../js/admin_script.js"></script>
</body>
</html>
```

Products.php

```
<?php
include '../components/connect.php';
session_start();
$admin_id = $_SESSION['admin_id'];
if(!isset($admin_id)){
  header('location:admin_login.php');
};
if(isset($_POST['add_product'])){
  $name = $_POST['name'];
  $name = filter_var($name, FILTER_SANITIZE_STRING);
  $price = $_POST['price'];
  $price = filter_var($price, FILTER_SANITIZE_STRING);
  $details = $_POST['details'];
  $details = filter_var($details, FILTER_SANITIZE_STRING);
  $image_01 = $_FILES['image_01']['name'];
  $image_01 = filter_var($image_01, FILTER_SANITIZE_STRING);
  $image_size_01 = $_FILES['image_01']['size'];
  $image_tmp_name_01 = $_FILES['image_01']['tmp_name'];
  $image_folder_01 = '../uploaded_img/'.$image_01;
  $image_02 = $_FILES['image_02']['name'];
  $image_02 = filter_var($image_02, FILTER_SANITIZE_STRING);
  $image_size_02 = $_FILES['image_02']['size'];
  $image_tmp_name_02 = $_FILES['image_02']['tmp_name'];
  $image_folder_02 = '../uploaded_img/'.$image_02;
  $image_03 = $_FILES['image_03']['name'];
  $image_03 = filter_var($image_03, FILTER_SANITIZE_STRING);
  $image_size_03 = $_FILES['image_03']['size'];
  $image_tmp_name_03 = $_FILES['image_03']['tmp_name'];
```

```
   $image_folder_03 = '../uploaded_img/'.$image_03;
   $select_products = $conn->prepare("SELECT * FROM `products` WHERE name = ?");
   $select_products->execute([$name]);
   if($select_products->rowCount() > 0){
      $message[] = 'product name already exist!';
   }else{
      $insert_products = $conn->prepare("INSERT INTO `products`(name, details, price, image_01, image_02, image_03)
VALUES(?,?,?,?,?,?)");
      $insert_products->execute([$name, $details, $price, $image_01, $image_02, $image_03]);
      if($insert_products){
         if($image_size_01 > 2000000 OR $image_size_02 > 2000000 OR $image_size_03 > 2000000){
            $message[] = 'image size is too large!';
         }else{
            move_uploaded_file($image_tmp_name_01, $image_folder_01);
            move_uploaded_file($image_tmp_name_02, $image_folder_02);
            move_uploaded_file($image_tmp_name_03, $image_folder_03);
            $message[] = 'new product added!';
         }
      }
   }
};
if(isset($_GET['delete'])){
   $delete_id = $_GET['delete'];
   $delete_product_image = $conn->prepare("SELECT * FROM `products` WHERE id = ?");
   $delete_product_image->execute([$delete_id]);
   $fetch_delete_image = $delete_product_image->fetch(PDO::FETCH_ASSOC);
   unlink('../uploaded_img/'.$fetch_delete_image['image_01']);
   unlink('../uploaded_img/'.$fetch_delete_image['image_02']);
   unlink('../uploaded_img/'.$fetch_delete_image['image_03']);
   $delete_product = $conn->prepare("DELETE FROM `products` WHERE id = ?");
   $delete_product->execute([$delete_id]);
   $delete_cart = $conn->prepare("DELETE FROM `cart` WHERE pid = ?");
   $delete_cart->execute([$delete_id]);
   $delete_wishlist = $conn->prepare("DELETE FROM `wishlist` WHERE pid = ?");
   $delete_wishlist->execute([$delete_id]);
   header('location:products.php');
}
?>
<!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>products</title>
   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.1/css/all.min.css">
   <link rel="stylesheet" href="../css/admin_style.css">
</head>
<body>
<?php include '../components/admin_header.php'; ?>
<section class="add-products">
   <h1 class="heading">add product</h1>
   <form action="" method="post" enctype="multipart/form-data">
      <div class="flex">
         <div class="inputBox">
            <span>product name (required)</span>
            <input type="text" class="box" required maxlength="100" placeholder="enter product name" name="name">
         </div>
```

```
      <div class="inputBox">
        <span>product price (required)</span>
        <input type="number" min="0" class="box" required max="9999999999" placeholder="enter product price"
onkeypress="if(this.value.length == 10) return false;" name="price">
      </div>
      <div class="inputBox">
        <span>image 01 (required)</span>
        <input type="file" name="image_01" accept="image/jpg, image/jpeg, image/png, image/webp" class="box" required>
      </div>
      <div class="inputBox">
        <span>image 02 (required)</span>
        <input type="file" name="image_02" accept="image/jpg, image/jpeg, image/png, image/webp" class="box" required>
      </div>
      <div class="inputBox">
        <span>image 03 (required)</span>
        <input type="file" name="image_03" accept="image/jpg, image/jpeg, image/png, image/webp" class="box" required>
      </div>
      <div class="inputBox">
        <span>product details (required)</span>
        <textarea name="details" placeholder="enter product details" class="box" required maxlength="500" cols="30"
rows="10"></textarea>
      </div>
    </div>
   <input type="submit" value="add product" class="btn" name="add_product">
  </form>
</section>
<section class="show-products">
  <h1 class="heading">products added</h1>
  <div class="box-container">
  <?php
    $select_products = $conn->prepare("SELECT * FROM `products`");
    $select_products->execute();
    if($select_products->rowCount() > 0){
      while($fetch_products = $select_products->fetch(PDO::FETCH_ASSOC)){
  ?>
  <div class="box">
    <img src="../uploaded_img/<?= $fetch_products['image_01']; ?>" alt="">
    <div class="name"><?= $fetch_products['name']; ?></div>
    <div class="price">Rs. <span><?= $fetch_products['price']; ?></span>/-</div>
    <div class="details"><span><?= $fetch_products['details']; ?></span></div>
    <div class="flex-btn">
      <a href="update_product.php?update=<?= $fetch_products['id']; ?>" class="option-btn">update</a>
      <a href="products.php?delete=<?= $fetch_products['id']; ?>" class="delete-btn" onclick="return confirm('delete this
product?');">delete</a>
    </div>
  </div>
  <?php
      }
    }else{
      echo '<p class="empty">no products added yet!</p>';
    }
  ?>
    </div>
</section>
<script src="../js/admin_script.js"></script>
</body>
</html>
```

Users_account.php

```
<?php
include '../components/connect.php';
session_start();
$admin_id = $_SESSION['admin_id'];
if(!isset($admin_id)){
  header('location:admin_login.php');
}
if(isset($_GET['delete'])){
  $delete_id = $_GET['delete'];
  $delete_user = $conn->prepare("DELETE FROM `users` WHERE id = ?");
  $delete_user->execute([$delete_id]);
  $delete_orders = $conn->prepare("DELETE FROM `orders` WHERE user_id = ?");
  $delete_orders->execute([$delete_id]);
  $delete_messages = $conn->prepare("DELETE FROM `messages` WHERE user_id = ?");
  $delete_messages->execute([$delete_id]);
  $delete_cart = $conn->prepare("DELETE FROM `cart` WHERE user_id = ?");
  $delete_cart->execute([$delete_id]);
  $delete_wishlist = $conn->prepare("DELETE FROM `wishlist` WHERE user_id = ?");
  $delete_wishlist->execute([$delete_id]);
  header('location:users_accounts.php');
}
?>
<!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>users accounts</title>
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.1/css/all.min.css">
  <link rel="stylesheet" href="../css/admin_style.css">
</head>
<body>
<?php include '../components/admin_header.php'; ?>
<section class="accounts">
  <h1 class="heading">user accounts</h1>
  <div class="box-container">
  <?php
    $select_accounts = $conn->prepare("SELECT * FROM `users`");
    $select_accounts->execute();
    if($select_accounts->rowCount() > 0){
      while($fetch_accounts = $select_accounts->fetch(PDO::FETCH_ASSOC)){
  ?>
  <div class="box">
    <p> user id : <span><?= $fetch_accounts['id']; ?></span> </p>
    <p> username : <span><?= $fetch_accounts['name']; ?></span> </p>
    <p> email : <span><?= $fetch_accounts['email']; ?></span> </p>
    <a href="users_accounts.php?delete=<?= $fetch_accounts['id']; ?>" onclick="return confirm('delete this account? the user
related information will also be delete!')" class="delete-btn">delete</a>
  </div>
  <?php
      }
    }else{
      echo '<p class="empty">no accounts available!</p>';
```

```
    } ?>
  </div>
</section>
<script src="../js/admin_script.js"></script>
</body>
</html>
```

JAVASCRIPT CODES

Admin_script.js

```
let navbar = document.querySelector('.header .flex .navbar');
let profile = document.querySelector('.header .flex .profile');
document.querySelector('#menu-btn').onclick = () =>{
  navbar.classList.toggle('active');
  profile.classList.remove('active');
}
document.querySelector('#user-btn').onclick = () =>{
  profile.classList.toggle('active');
  navbar.classList.remove('active');
}
window.onscroll = () =>{
  navbar.classList.remove('active');
  profile.classList.remove('active');
}
let mainImage = document.querySelector('.update-product .image-container .main-image img');
let subImages = document.querySelectorAll('.update-product .image-container .sub-image img');
subImages.forEach(images =>{
  images.onclick = () =>{
    src = images.getAttribute('src');
    mainImage.src = src;
  }
});
```

script.js

```
let navbar = document.querySelector('.header .flex .navbar');
let profile = document.querySelector('.header .flex .profile');
document.querySelector('#menu-btn').onclick = () =>{
  navbar.classList.toggle('active');
  profile.classList.remove('active');
}
document.querySelector('#user-btn').onclick = () =>{
  profile.classList.toggle('active');
  navbar.classList.remove('active');
}
window.onscroll = () =>{
  navbar.classList.remove('active');
  profile.classList.remove('active');
}
let mainImage = document.querySelector('.quick-view .box .row .image-container .main-image img');
let subImages = document.querySelectorAll('.quick-view .box .row .image-container .sub-image img');
subImages.forEach(images =>{
```

```
  images.onclick = () =>{
    src = images.getAttribute('src');
    mainImage.src = src;
  }
});
```

PYTHON CODE

App.py

```
import os
from flask import Flask, jsonify, request, render_template
from flask_cors import CORS
import numpy as np
from tensorflow.keras.preprocessing.image import load_img, img_to_array
from keras.applications.resnet_v2 import preprocess_input as resnet152v2_preprocess_input
from keras.applications.resnet_v2 import ResNet152V2
from keras.applications.imagenet_utils import decode_predictions
# Load the pre-trained ResNet152V2 model
model = ResNet152V2(weights='imagenet')
# Define a dictionary to map the original class names to new class names
class_dict = {
    'notebook': 'laptop',
   'cellular_telephone': 'Smartphone',
   'hand-held_computer': 'Smartphone',
   'remote_control': 'Smartphone',
   'iPod': 'Smartphone',
   'microphone': 'Headphone',
   'espresso_maker': 'Headphone',
   'washer': 'Washing Machine',
   'analog_clock': 'Watch',
   'digital_watch': 'Watch',
   'reflex_camera': 'Camera',
   'vending_machine': 'Refrigerator',
   'television': 'TV',
   'ashcan': 'Washing Machine'
}
app = Flask(__name__)
CORS(app)
# Define the upload folder
UPLOAD_FOLDER = 'uploads'
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
# Define the allowed file extensions
ALLOWED_EXTENSIONS = {'png', 'jpg', 'jpeg', 'gif'}
app.config['ALLOWED_EXTENSIONS'] = ALLOWED_EXTENSIONS
def allowed_file(filename):
   """Function to check if the file extension is allowed"""
   return '.' in filename and \
         filename.rsplit('.', 1)[1].lower() in app.config['ALLOWED_EXTENSIONS']
@app.route('/predict_image', methods=['POST'])
def predict_image():
      # Check if a file was submitted with the request
   if request.method == "POST":
      if 'file' not in request.files:
         return render_template('search_page.php', message='No file selected')
```

```
        file = request.files['file']
        # Check if the file is an allowed file type
        if not allowed_file(file.filename):
            return render_template('search_page.php', message='File type not allowed')
        # Save the file to the upload folder
        filename = file.filename
        file.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
        # Load the input image
        img_path = os.path.join(app.config['UPLOAD_FOLDER'], filename)
        img = load_img(img_path, target_size=(224, 224))
        # Preprocess the input image
        img = img_to_array(img)
        img = np.expand_dims(img, axis=0)
        img = resnet152v2_preprocess_input(img)
        # Make a prediction
        preds = model.predict(img)
        # Decode the prediction and print the predicted class
        decoded_preds = decode_predictions(preds, top=1)[0]
        for pred in decoded_preds:
            # Replace the original class name with the new class name
            pred = (pred[0], class_dict.get(pred[1], pred[1]), pred[2])
            item,prob = pred[1], pred[2]
            return jsonify(
                name=item
            )
@app.route('/view/<filename>')
def view_image(filename):
    """Function to display an uploaded image"""
    return '<img src="../uploads/{}">'.format(filename)
if __name__ == '__main__':
    app.run(debug=True)
```

Conclusion

The “Image Recognition Based Advance E-commerce Website” will transform the way users shop. With the introduction of the Image Recognition feature, the online shopping experience has been further improved. It has streamlined the product search process, making it more efficient and seamless for customers. Additionally, the website's user-friendly design and responsive layout ensure that it is accessible to a wide range of customers across various devices. The inclusion of a "Contact Us" section further provides a reliable means for customers to communicate with the customer support team, enhancing the overall customer experience. With its comprehensive admin panel, the website is well-equipped to manage its inventory and transactions effectively. Overall, the website stands out as an innovative platform that redefines the online shopping experience, catering to the needs of modern-day customers.

BIBILOGRAPHY

Books

- Computer Science with Python – by Sumita Arora (Publisher: Dhanpat Rai & Co.; 2020th edition (1 January 2020); Dhanpat Rai & Co.)

Websites

- https://www.tutorialspoint.com/
- https://www.brainly.com/
- https://www.geeksforgeeks.org/
- https://www.w3schools.com/
- https://www.stackoverflow.com/
- https://www.codeproject.com/