A PROJECT ON

# “ E-commerce “

SUBMITTED IN

PARTIAL FULFILLMENT OF THE REQUIREMENT

FOR THE COURSE OF DIPLOMA IN ADVANCED COMPUTING FROM CDAC

## ACKNOWLEDGEMENT

**CERTIFICATE**

## Table of Contents:

1. INTRODUCTION… 5
   1. Objective of project 5
2. REQUIREMENTS 6
   1. Functional Requirements… 6
      1. User Account… 7
      2. Registration and creation of user profile 7
      3. Add to cart 8
      4. View Order History 9
   2. Non-Functional Requirements… 11
      1. Hardware and Software Requirements… 12
3. DESIGN… 13
   1. Database design 13
4. CODING STANDARD IMPLIMENTED… 18
5. TEST REPORT 20
6. APENDIX A

Entity Relationship Diagram 21

1. APENDIX B

UI Screenshots… 24

1. REFERENCES 30
2. **INTRODUCTION TO PROJECT**

The web-based "Online Clothing-Website" project aims to simulate the fundamental concepts of an online platform for purchasing products and services. This system allows users to register on the website, enabling them to shop and file complaints regarding any incidents that may have occurred. The system provides registration functionality for different types of users, allowing each to create an account.

Once logged in, users can browse and purchase products, view and manage items in their shopping cart, and leave feedback and ratings for purchased products. The system includes three types of logins: User, Admin, and Delivery Boy, each with its own distinct features and responsibilities.

**Admin Module:** This module is essential for managing and overseeing the e-commerce platform. It provides administrative control and ensures that the platform runs smoothly and efficiently.

**User Module:** The heart of the e-commerce experience, this module is designed to provide customers with a seamless and engaging shopping journey.

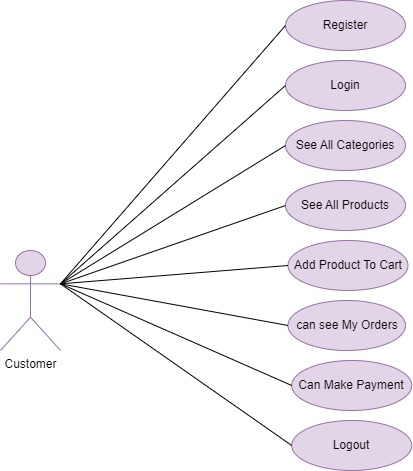
**Delivery Boy Module:** This module is designed to manage and optimize the logistics of order fulfilment , ensuring the timely and accurate delivery of products to customers.

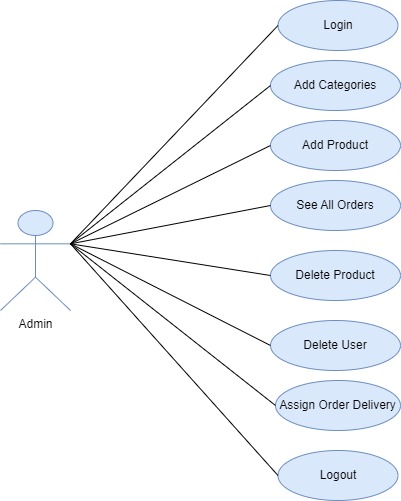
At "Online Clothing-Website," we are committed to offering a superior shopping experience by blending fashion-forward choices with excellent service. Our goal is to help customers look and feel their best, with clothing that not only enhances their style but also fits seamlessly into their lifestyle.

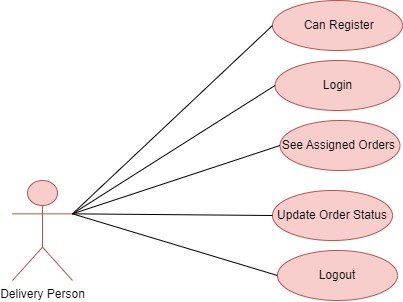
The E-Clothing Website Project is poised to set a new standard in online clothing retail by combining fashion with technology in an accessible and user-friendly format. We are excited to embark on this journey and look forward to creating a platform that delights users and meets the evolving needs of the modern shopper.

## REQUIREMENTS

#### FUNCTIONAL REQUIREMENTS:







#### User Account

The user, referred to as the "Customer," will be presented with a set of options by the Online Clothing Website System as the first step in their interaction. A customer can drag and drop products into the shopping cart and view the list of available products. They can choose any product to purchase or log out, depending on their needs. A customer may also check the availability of a product even if they are not logged in.

A customer who has registered with the system will be given a user ID and password. This "personal information" will be referred to as their "profile." A customer with a profile stored in the system's database will be called a "registered user." A registered user will be able to log into the system to buy new products.

A new customer, on the other hand, will need to register by providing personal information such as their name, contact details, and other relevant information. Upon registration, the new customer will become a registered user.

Once logged in, a registered user can buy and review products. The system provides the customer with the option to log out at any time during the shopping process.

#### Registration and creation of user profile

In the E-Clothing project, the registration process allows new customers to create their user profiles by providing essential personal information, such as their name, contact details, and preferred shipping address. Upon successful registration, a user ID and password are generated, enabling the customer to log into the system. The user profile, which contains the customer's personal information and shopping preferences, is securely stored in the system's database. Registered users can then access their profile to manage personal details, view order history, and make purchases seamlessly.

#### Add to cart

Here we provided Add to cart facility for any user to add product while shopping into account.

After logging in a user (either a registered user or a guest), the system shall

**Product Selection and Addition:**

* Users should be able to browse and select clothing items.
* The system must provide clear "Add to Cart" buttons on product detail pages, enabling users to add products with a single click.

**Cart Overview and Management:**

* Users should be able to view their cart at any time, displaying product details such as name, size, color, quantity, price, and thumbnail images.
* Users should be able to save items in the cart for later, ensuring that their cart contents persist across sessions.
* Registered users should have their cart synchronized across different devices when logged in.

**Product Availability and Stock Validation:**

* The system should check product availability before adding items to the cart. If an item is out of stock, the system should notify the user with an appropriate message.
* For low-stock items, the system should inform the user of the remaining quantity and prevent adding more than available stock.
  + 1. **View Order History**

The system shall allow a user to view all information about their previous orders. After logging in, the user will be prompted to select an order. The system will access the User Order table and retrieve the details of the selected order, presenting them to the user in a tabular format.

#### NON-FUNCTIONAL REQUIREMENTS:

Following Non-Functional Requirements will be there in the insurance to the internet:

1. Secure access to user’s confidential data.
2. 24X7 availability.
3. Better component design to get better performance at peak time.
4. Flexible service-based architecture will be highly desirable for future extension. Non-Functional requirements define system properties and constraints.

Various other Non-Functional Requirements are:

* Security
* Reliability
* Maintainability
* Portability
* Extensibility
* Reusability
* Compatibility

#### Hardware Other Requirements

* **Hardware Interfaces**

The SPMS is expected to function on Dual core Processor equivalent or above, 4 GB RAM, 512 GB HDD.

#### Software Interfaces

The SPMS shall work on MS Windows operating systems family (MS Windows 2007, MS Windows 2008, MS Windows 2011). It configures to work with MySQL database. This System works on Apache Tomcat server. It uses browser IE 5.0 & above. It uses IIS 5.0 server. Compatible with any web browser.

## DESIGN

#### Database Design

The following table structures depict the database design.

#### Table 1 : User

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/ Constraint** | **Column Name** | **Data Type** | **Length** | **Allow Null (1=Yes;0=No)** |
| 2 | id | INT | - | 0 |
| 1 | email\_id | varchar | 255 | 1 |
| 0 | |  | | --- | | first\_name | | varchar | 20 | 0 |
| 0 | |  | | --- | |  |   last\_name | varchar | 20 | 0 |
| 0 | password | varchar | 255 | 1 |
| 0 | phone\_no | varchar | 255 | 0 |
| 0 | role | varchar | 255 | 1 |
| 1 | address\_id | INT | - | 1 |

#### Table 2 : address

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/ Constraint** | **Column Name** | **Data Type** | **Length** | **Allow Null (1=Yes;0=No)** |
| 2 | id | INT | - | 0 |
| 0 | city | varchar | 255 | 0 |
| 0 | pincode | INT | - | 0 |
| 0 | street | varchar | 255 | 0 |

#### Table 3 : Category

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/ Constraint** | **Column Name** | **Data Type** | **Length** | **Allow Null (1=Yes;0=No)** |
| 2 | id | INT | - | 0 |
| 0 | description | varchar | 255 | 0 |
| 0 | title | varchar | 255 | 0 |

#### Table 4 : Product

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/ Constraint** | **Column Name** | **Data Type** | **Length** | **Allow Null (1=Yes;0=No)** |
| 2 | id | INT | - | 0 |
| 0 | description | varchar | 255 | 0 |
| 0 | image\_name | varchar | 255 | 0 |
| 0 | price | decimal | (19,2) | 0 |
| 0 | quantity | INT | - | 0 |
| 0 | title | varchar | 255 | 0 |
| 1 | category\_id | INT | - | 1 |

#### Table 5 : Review

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/ Constraint** | **Column Name** | **Data Type** | **Length** | **Allow Null (1=Yes;0=No)** |
| 2 | id | bigint | - | 0 |
| 0 | comment | varchar | 255 | 0 |
| 0 | rating | double | 255 | 0 |
| 1 | product\_id | INT | 255 | 1 |
| 1 | user\_id | INT | 255 | 1 |

#### Table 6 : Cart

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/ Constraint** | **Column Name** | **Data Type** | **Length** | **Allow Null (1=Yes;0=No)** |
| 2 | id | INT | - | 0 |
| 0 | quantity | varchar | 255 | 0 |
| 1 | product\_id | INT | 255 | 1 |
| 1 | user\_id | INT | 255 | 1 |

#### Table 7 : Orders

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/ Constraint** | **Column Name** | **Data Type** | **Length** | **Allow Null (1=Yes;0=No)** |
| 2 | id | INT | - | 0 |
| 0 | delivery\_assigned | varchar | 255 | 1 |
| 0 | delivery\_date | varchar | 255 | 1 |
| 0 | delivery\_person\_id | INT | - | 0 |
| 0 | delivery\_status | varchar | 255 | 1 |
| 0 | delivery\_time | varchar | 255 | 1 |
| 0 | order\_date | varchar | 255 | 1 |
| 0 | order\_id | varchar | 255 | 1 |
| 0 | quantity | INT | - | 0 |
| 1 | product\_id | INT | - | 1 |
| 1 | user\_id | INT | - | 1 |

## CODING STANDARDS IMPLEMENTED

**Naming and Capitalization**

Below summarizes the naming recommendations for identifiers in Pascal casing is used mainly (i.e. capitalize first letter of each word) with camel casing (capitalize each word except for the first one) being used in certain circumstances.

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | **Case** | **Examples** | **Additional Notes** |
| Class | Pascal | User, Product | Class names should be based on "objects" or "real things" and should generally be **nouns**. No ‘\_’ signs allowed. Do not use type prefixes like ‘C’  for class. |
| Method | Camel | addUser,  addCategory | Methods should use **verbs** or verb phrases. |
| Parameter | Camel | firstName, planeCode | Use descriptive parameter names. Parameter names should be descriptive enough that the name of the parameter and its type can be used to  determine its meaning in most scenarios. |
| Interface | Pascal with  "I" prefix | IUserService | Do not use the ‘\_’ sign |
| Property | Camel | ForeColor  BackColor  firstClassSeats | Use a noun or noun phrase to name properties. |

#### Comments :

* Comment each type, each non-public type member, and each region declaration.
* Use end-line comments only on variable declaration lines. End-line comments are comments that follow code on a single line.
* Separate comments from comment delimiters (apostrophe) or // with one space.
* Begin the comment text with an uppercase letter.
* End the comment with a period.
* Explain the code; do not repeat it.

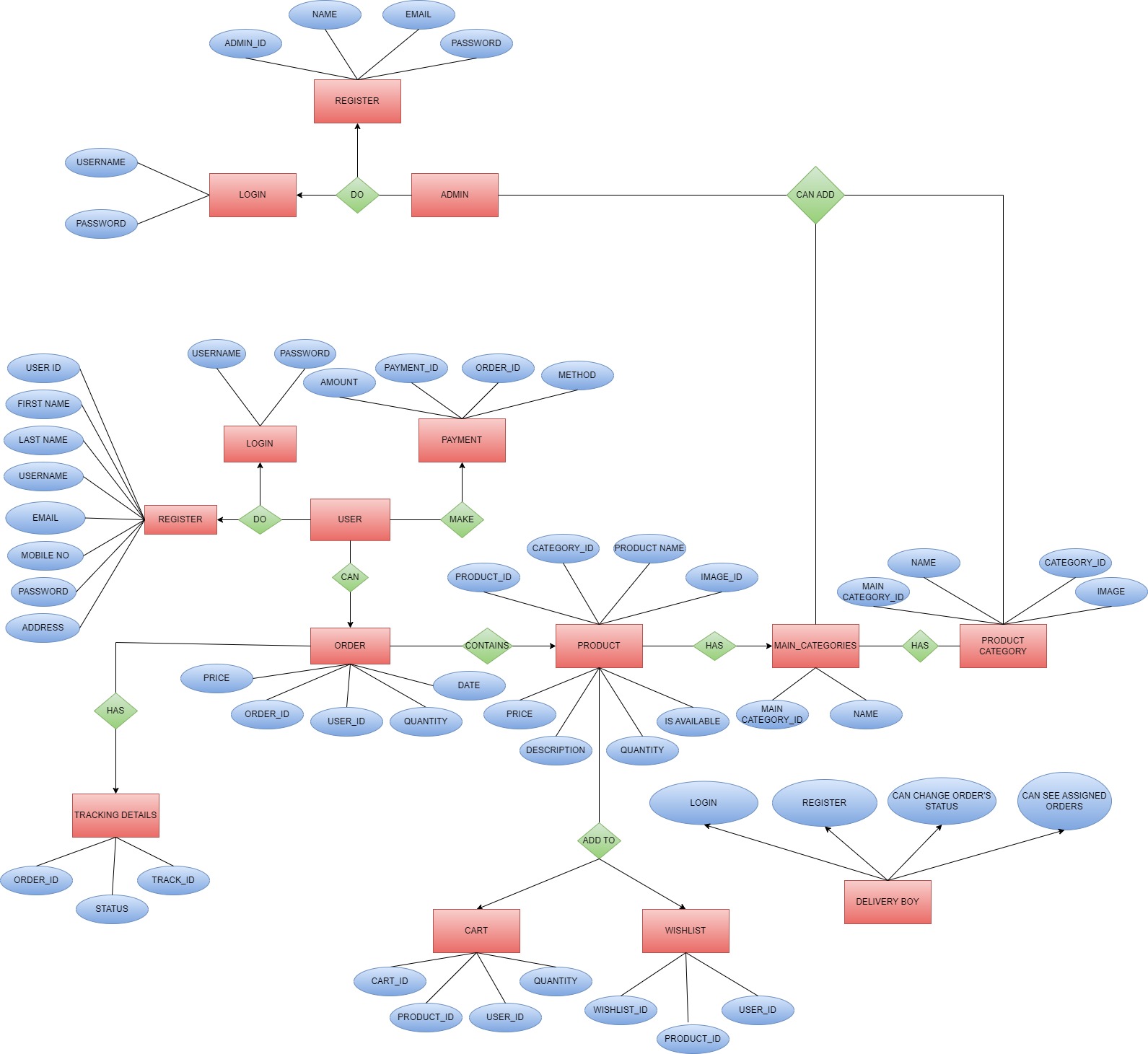
## TEST REPORT

**GENERAL TESTING :**

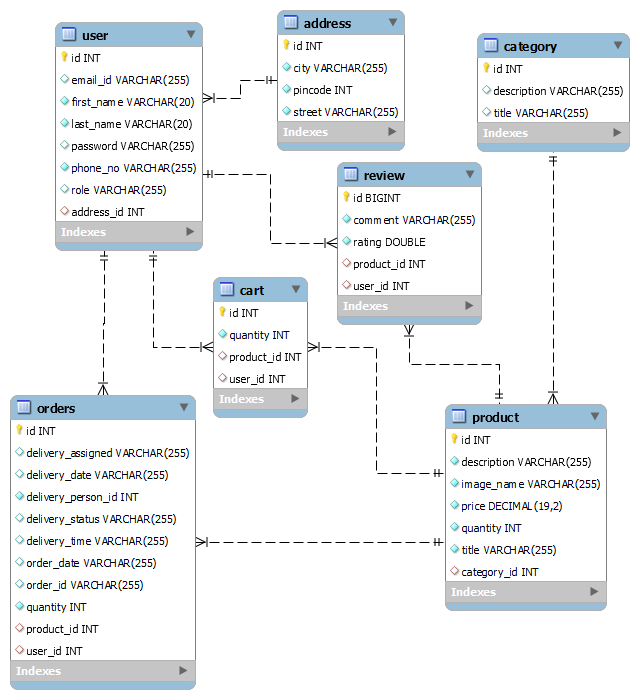
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SR- NO** | **TEST CASE** | **EXPECTED RESULT** | **ACTUAL RESULT** | **ERROR MESSAGE** |
| 1 | Register Page | Redirected to Next page | OK | Nothing |
| 2 | Login Page | Pop-up will come | Ok | Please enter username and password again . |
| 3 | User Register | Register a User | Ok | Nothing |
| 4 | Checking login or not | User is logged in or not | Ok | Nothing |
| 5 | Add Product | Add Information about Product | Ok | Nothing |
| 6 | Order Details | Get all orders | Ok | Nothing |
| 7 | Add to cart | Test adding products to cart | Ok | Nothing |
| 8 | Security (SQL Injection) | Test SQL injection vulnerability | Ok | Nothing |
| 9 | Mobile Responsiveness | Test website on mobile devices | Ok | Nothing |
| 10 | Logout | It will logout from user profile. | Ok | Nothing |

# Appendix A

### Entity Relationship Diagram

****

**Data Flow Diagram:**

****

# Appendix B

## Homepage :

**Login Popup :**

## User Registration :

**Admin login:**

## Add Category:

**Add Product:**

**Delete Product :**

**Delivery Status:**

**Add To Cart:**

**Customer Payment:**

**Delivery Person HomePage:**

**Delivery Update Order:**

#### 8.REFERENCES:

[http://www.google.com](http://www.google.com/) <http://www.Amazon.com/> [http://www.w3.org](http://www.w3.org/) [http://www.wikipedia.org](http://www.wikipedia.org/)