**ONLINE Cosmetic SHOP MANAGEMENT SYSTEM**

**ABSTRACT**

The system aims to provide users with a seamless and secure platform to explore, select, and purchase cosmetic products, enhancing the online shopping experience within the beauty and skincare industry. The Online Cosmetic Shop System offers an intuitive interface for customers to browse a diverse catalog of cosmetic products, view detailed product information, and securely complete their purchases. The system incorporates features such as personalized user accounts, secure payment gateways, and order tracking for a convenient and transparent shopping experience. It also includes robust user authentication and authorization mechanisms, ensuring the privacy of user data and transactional information.

Key features include a responsive design for optimal accessibility across various devices, including desktops, laptops, tablets, and smartphones. The project prioritizes data integrity and security, utilizing MySQL as the relational database management system to efficiently store and retrieve information related to product listings, customer profiles, and order histories.

1. **INTRODUCTION**

Shopping has long been considered a recreational activity by many. Shopping online is no exception. The goal of this application is to develop a web-based interface for online retailers. The system would be easy to use and hence make the shopping experience pleasant for the users. The goal of this application is to develop an easy-to-use web-based interface where users can search for products, view a complete description of the products and order the products. A search engine that provides an easy and convenient way to search for products specific to their needs. The search engine would list a set of products based on the search term and the user can further filter the list based on various parameters. Drag and Drop feature which would allow the users to add a product to or remove a product from the shopping cart by dragging the product in to the shopping cart or out of the shopping cart. A user can view the complete specification of the product along with various images and also view the customer reviews of the product. They can also write their own reviews.

1. **IMPLEMENTATION**

Internet technology is the newest and exciting technology in the world. With highly speed development, it brings many conveniences to the people, also becomes the focus of network application in present. This technology involves not only in the traditional industry, but also in e-commerce, Online Shopping is a remarkable case. The system has realized the function of shopping online, and the condition of shopping will not be restricted by time and region, so it is convenient to understand the information of all commodities. This development model has promoted the economic development greatly.

**EXISTING SYSTEM**

There are large numbers of commercial Online Shopping websites offering large number of products tailored to meet the shopping interests of large number of customers. These online marketplaces have thousands of products listed under various categories.

• The basic problems with the existing systems are the non-interactive environment they provide to the users.

• The use of traditional user interfaces which make continuous post backs to the server; each post back makes a call to the server, gets the response and then refreshes the entire web form to display the result. This scenario adds an extra trade off causing a delay in displaying the results

• A search engine that would display the results without allowing the users to further filter the results based on various parameters.

• Use of traditional and non-user-friendly interfaces that are hard to use

**PROPOSED SYSTEM:**

The project Online Shopping Portal System is GUI based system so that it is easy to handle. It also increases the efficiency of the end user, because it will reduce the redundant job, which is tedious to complete. The Online System also has automated capability to complete job, so it reduces the work manually.

• The motive of this Online Shopping Web Application is to allow the user to play with the search tool and create different combinatorial search criterion to perform exhaustive search.

• Provide Interactive interface through which a user can interact with different areas of application easily.

• A search engine that provides an easy and convenient way to search for products specific to their needs. The search engine would list a set of products based on the search term and the user can further filter the list based on various parameters.

• Provide Drag and Drop feature thereby allowing the user to add products to or remove products from the shopping cart by dragging the products in to or out of the shopping cart.

1. **DESCRIPTION OFMODULES:**

There are two modules that are used,

* User
* Admin

**User Module:**

* Register
* Login
* search for product
* Add to cart

**User Details**

**Register**:

User has to register their basic details to get access with this application service. In this module user can add the personal details. Like name, email, Mob. User and the authorized person to access this module. Other user doesn’t get rights to access this module for security purpose.

**Login**:

Once they have registered they need to login to avail the service at the needy time. Here user has to login by using their unique user’s name and password. User and the authorized person to access user module for security  
purpose when the password is strictly confidential. So other  
unauthorized person can’t get rights to access this module

**Search for product:**

In this module user can view the product details by entering the name of the product. Once the search option is provided the user could able to view for the various items regarding the product. It may vary according to quality, rate , production of the company etc. User and the authorized person to access this module. Other user doesn’t get rights to access this module for security purpose.

**Add to cart:**

In this module user can store or save the product which has been searched by using the previous module. All the user could not have similar idea hence the cart section helps the user to store and could retrieve the product. Other user doesn’t get rights to access this module for security purpose.

**Admin**

* Login.
* Product Management(Add,Update, Delete)
* Order Management

**Login**:  
Here administrator has to login by using their unique user’s name and  
password. administrator and the authorized person to access user  
module for security purpose when the password is strictly  
confidential. So other unauthorized person can’t get rights to access  
this module.  
**Product Management**:

In this module, admin can add, update, alter and delete the product that are previously provided or stored in the data. The admin also has the authority to change the amount payable by means of providing the offer. The administrator and the authorized person to access this module. Other user doesn’t get rights to access this module for security purpose.  
**Order Management:**

In this module, admin has the authority to manage all the orders which has been placed by the user. That could be edited or deleted only by the authorized person. Other user doesn’t get rights to access this module for security purpose.

1. **SYSTEM SPECIFICATION**

**HARDWARE SPECIFICATION**

* System : Inteli3
* Hard Disk : 500 GB.
* Monitor : 15 VGA Colour.
* Mouse : Logitech.
* Ram : 4 GB.
* Keyboard : 101 Keyboard

**SOFTWARE SPECIFICATION**:

* Operating System : Windows 8.1
* Front End : PHP version 7
* Back End : MySQL version 7
* Server : XAMPP server

**SOFTWARE SPECIFICATION**

**SOFTWARE DESCRIPTION**

**XAMPP:**

XAMPP is a [free and open source](https://en.wikipedia.org/wiki/Free_software) [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) [web server](https://en.wikipedia.org/wiki/Web_server) [solution stack](https://en.wikipedia.org/wiki/Solution_stack) package developed by Apache Friends, consisting mainly of the [Apache HTTP Server](https://en.wikipedia.org/wiki/Apache_HTTP_Server), [Maria DB](https://en.wikipedia.org/wiki/MariaDB) [database](https://en.wikipedia.org/wiki/Database), and [interpreters](https://en.wikipedia.org/wiki/Interpreter_%28computing%29) for scripts written in the [PHP](https://en.wikipedia.org/wiki/PHP) and [Perl](https://en.wikipedia.org/wiki/Perl) [programming languages](https://en.wikipedia.org/wiki/Programming_language).

XAMPP stands for Cross-Platform (X), Apache (A), Maria DB (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes.

Everything needed to set up a web server – server application (Apache), database (Maria DB), and scripting language (PHP) – is included in an extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows.

XAMPP's designers intended it for use only as a development tool, to allow website designers and programmers to test their work on their own computers without any access to the Internet.

**CROSS PLATFORM**

Cross-platform software is a type of software application that which works on multiple operating systems or devices, which are often referred to as platforms. A platform means an operating system such as Windows, Mac OS, Android or iOS. When a software application works on more than one platform, the user can utilize the software on a wider choice of devices and computers.

**BENEFITS OF CROSS PLATFORM**

The benefit of a cross-platform software app or program is that you can use the same program whether you’re on a Windows PC or whether you’re logging in from your laptop or smartphone. The Microsoft Office suite of applications, which includes Word, Excel, and PowerPoint, are available on Windows, Mac OS, iOS (iPhone/iPad) and Android. While there are differences based on how the platforms work, you’ll have a similar experience within the application between all of your devices.

Having a similar experience across any platform means there’s a much smaller learning curve, if one even exists at all, so you’ll be more productive and be able to use a software product you’re familiar with regardless of the operating system or device you choose. In addition, your files can be moved much more easily between your devices so you can use the software with whatever device you have with you at the time. And there’s a way to keep all of your work in sync across all of your devices, by using the cloud.

**EXAMPLES OF CROSS PLATFORM**

# Unity3D

First, let’s talk about Unity3D. I think the game engine should be preferred by people who want to write mobile games.  
You can develop games on 17 platforms using multiple languages, including Linux. Of course, ios, android and windows phone is also the most ideal game engine to develop games.

You can develop your application using C #, JS, C ++.

Link to: [https://unity3d.com](https://unity3d.com/)

# Xamarin

Xamarin Some time ago, it was purchased by Microsoft and is a perfect fit for developers using C #.

Because it is a C # language, it has a lot of documentation, and because of Microsoft support, Xamarin is the choice for C # developers.

In addition, you can do everything you can do in Objective-C, Swift and Javada with the Xamarian library.

Link to: [https://xamarin.com](https://xamarin.com/)

# React Native

React Native is an open-source JavaScript library developed by the new generation of React — Facebook, which was open to Github in 2013. Native application creation means writing applications only for a specific operating system. React Native helps developers reuse their code over the web and on the mobile. Developers will not have to create the same app from scratch for iOS and Android. They will be able to reuse the code in each operating system. The great thing about React Native is that there is little difference between a finished application in Objective-C or Java and an application built using React Native. Android and iOS code development environments are very different. So it takes time to remove the application to two different platforms. However, with React Native, only one developer can write on different mobile operating systems.

**APACHE:**

The Apache HTTP Server, colloquially called Apache is a [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source) [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) [web server](https://en.wikipedia.org/wiki/Web_server) software, released under the terms of [Apache License](https://en.wikipedia.org/wiki/Apache_License) 2.0. Apache is developed and maintained by an open community of developers under the auspices of the [Apache Software Foundation](https://en.wikipedia.org/wiki/Apache_Software_Foundation).

The vast majority of Apache HTTP Server instances run on a [Linux distribution](https://en.wikipedia.org/wiki/Linux_distribution), but current versions also run on [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows), [OpenVMS](https://en.wikipedia.org/wiki/OpenVMS),  and a wide variety of [Unix-like](https://en.wikipedia.org/wiki/Unix-like) systems. Past versions also ran on [NetWare](https://en.wikipedia.org/wiki/NetWare), [OS/2](https://en.wikipedia.org/wiki/OS/2) and other operating systems,  including ports to mainframes.

Originally based on the [NCSA HTTPd](https://en.wikipedia.org/wiki/NCSA_HTTPd) server, development of Apache began in early 1995 after work on the NCSA code stalled. Apache played a key role in the initial growth of the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web), quickly overtaking NCSA HTTPd as the dominant [HTTP](https://en.wikipedia.org/wiki/HTTP) server. In 2009, it became the first web server software to serve more than 100 million [websites](https://en.wikipedia.org/wiki/Website). As of January 2021, [Netcraft](https://en.wikipedia.org/wiki/Netcraft" \o "Netcraft) estimated that Apache served 24.63% of the million busiest websites, while [Nginx](https://en.wikipedia.org/wiki/Nginx) served 23.21% and Microsoft is in third place at 6.85% (for some of Netcraft's other stats Nginx is ahead of Apache), while according to W3Techs, Apache is ranked first at 35.0% and Nginx second at 33.0% and Cloudflare Server third at 17.3%.

**LANGUAGE SPECIFICATION**

**PHP**

**INTRODUCTION OF PHP**

PHP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994.

* PHP is a recursive acronym for "PHP: Hypertext Preprocessor".
* PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
* It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
* PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.
* PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.
* PHP is forgiving: PHP language tries to be as forgiving as possible.
* PHP Syntax is same as C language.

**What is a PHP File?**

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

**What Can PHP Do?**

* PHP can generate dynamic page content and it can create, open, read, write, delete, and close files on the server and it can collect form data.
* PHP can send and receive cookie and it can add, delete, modify data in your database and it can be used to control user-access and encrypt data.

**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.
* PHP is easy to learn and runs efficiently on the server side.

## **What is Database?**

* A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds.
* Other kinds of data stores can be used, such as files on the file system or large hash tables in memory but data fetching and writing would not be so fast and easy with those types of systems.
* So nowadays, we use relational database management systems (RDBMS) to store and manage huge volume of data. This is called relational database because all the data is stored into different tables and relations are established using primary keys or other keys known as foreign keys.

**MySQL Database**

* MySQL is released under an open-source license. So you have nothing to pay to use it. MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
* MySQL uses a standard form of the well-known SQL data language. MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc. MySQL works very quickly and works well even with large data sets.
* MySQL is very friendly to PHP, the most appreciated language for web development. MySQL supports large databases, up to 50 million rows or more in a table.
* The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).MySQL is customizable.
* The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments

**TABLE CREATION**

* Name of the table
* Names of fields
* Definitions for each field
* Field Attribute **NOT NULL** is being used because we do not want this field to be NULL. So if user will try to create a record with NULL value, then MySQL will raise an error.
* Field Attribute **AUTO\_INCREMENT** tells MySQL to go ahead and add the next available number to the id field.
* Keyword **PRIMARY KEY** is used to define a column as primary key. You can use multiple columns separated by comma to define a primary key.

## **ADMINISTRATIVE MYSQL COMMAND**

* **USE DATABASE NAME**: This will be used to select a particular database in MySQL work area.
* **SHOW DATABASES:** Lists the databases that are accessible by the MySQL DBMS.
* **SHOW TABLES:** Shows the tables in the database once a database has been selected with the use command.
* **SHOW COLUMNS FROM Table name:** Shows the attributes, types of attributes, key information, whether NULL is permitted, defaults, and other information for a table.
* **SHOW INDEX FROM Table name:** Presents the details of all indexes on the table, including the PRIMARY KEY

## **CREATING TABLES USING PHP SCRIPT:**

Create new table in any existing database you would need to use PHP function **mysqli\_query()**.

## **Dropping Tables Using PHP Script:**

Drop an existing table in any database, you would need to use PHP function **mysqli\_query()**.

## **INSERTING DATA USING PHP SCRIPT:**

**CREATE**

Create table statement is used to create a table in MySQL.

**SELECT**

The SELECT statement is used to select data from one or more tables.

**UPDATE**

The UPDATE statement is used to update existing records in a table:

## **DELETE**

The DELETE statement is used to delete records from a table:

**DATABASE DESIGN:**

The data in the system has to be stored and retrieved from database. Designing the database is part of system design. Data elements and data structures to be stored have been identified at analysis stage.

They are structured and put together to design the data storage and retrieval system. A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently.

The general objective is to make database access easy, quick, inexpensive and flexible for the user. Relationships are established between the data items and unnecessary data items are removed.

Normalization is done to get an internal consistency of data and to have minimum redundancy and maximum stability. This ensures minimizing data storage required, minimizing chances of data inconsistencies and optimizing for updates.

**INPUT DESIGN**

The Input design is the main feature of the system. Input design determines the format and validations criteria for data entering the system. Inputs originate with end-users; human factors play a significant role in input design. The input design is designed to control the input, to avoid delay, errors in data, to avoid extra steps, to keep the process simple. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy.

The following are the general principles, which are considered in designing inputs are,

* + - Enter only variable data
    - Do not input data that can be calculated
    - List of values
    - Sequence entry

**OUTPUT DESIGN**

Designing the output is more important than working up with few layout charts and reports. The outputs are designed based on the issue encountered. It will also take care of who will receive the output, what for it is produced how much details are needed, when it is needed and by what method.

The outputs designed in this system are easy to use and useful for their jobs. The outputs are simple to read interpret. The outputs obtained from this system are designed by using a few guidelines, which are given below. The information should be clear and accurate, yet concise and restricted to relevant data. Reports should have titles, the data and descriptive heading for columns of data, numbered pages and so on.

**SYSTEM TESTING**

System testing is the process of exercising software with the intent of finding and ultimately correcting errors. This fundamental philosophy does not change for web applications, because Web-based systems and application reside on a network and interoperate with many different operating system, browsers, hardware platforms, and communication protocols; the search for errors represents a significant challenge for web application.

The distributed nature of client/server environments, the performance issues associated with transaction processing, the potential presence of a number of different hardware platforms, the complexities of network communication, the need to serve multiple clients from a centralized database and the requirements imposed on the server all combine to make testing of client\server architectures.

Testing issues

* Client GUI considerations
* Target environment and platform diversity considerations
* Distributed database considerations
* Distributed processing considerations

**TYPES OF TESTING**

1. Unit Testing

2. Integration Testing

3. Validation Testing

4. User acceptance Testing

5. System Testing

**Unit Testing**

All modules were tested and individually as soon as they were completed were checked for their correct functionality. Unit testing is carried out by verify and recover errors within the boundary of the smallest unit or a module. In this testing step, each module was found to be working satisfactory per the expected output of the module. In the package development, each module is tested separately after it has been completed and checked with valid data.

**Integration Testing**

The entire project was split into small programs; each of these single programs gives a frame as an output. These programs were tested individually; at last all these programs where combined together by creating another program where all these constructions were used. It gives a lot of problem by not functioning in an integrated manner.

The user interface testing is important since the user has to declare that the arrangements made in the frames are convenient and it is satisfied. When the frames are the test, the end user gave suggestion. Since they were much exposed to do the work manually.

**Validation Testing**

At the culmination of the black box testing software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of test i.e., validation succeeds when the software functions in a manner that can be reasonably accepted by the customer.

**User Acceptance Testing**

User acceptance testing of the system is the key factor the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system at the time of development and making change whenever required. This is done with regard to the input screen design and output screen design.

**System Testing**

This is to verify that all the system elements have been properly integrated and perform allocated functions. Testing is executing a program to test the logic changes made in it and with intention of finding errors. Tests are also conducted to find discrepancies between system and its original objective, current specification and documents.

**SYTEM IMPLEMENTATION**

Implementation is the stage in the project where the theoretical design is turned into a working system. The most crucial stage is achieving a successful new system & giving the user confidence in that the new system will work efficiently & effectively in the implementation state.

The stage consists of

* + - Testing the developed program with simple data.
    - Detection’s and correction of error.
    - Creating whether the system meets user requirements.
    - Testing whether the system.
    - Making necessary changes as desired by the user.
    - Training user personnel.

**Implementation Procedures**

The implementation phase is less creative than system design. A system project may be dropped at any time prior to implementation, although it becomes more difficult when it goes to the design phase.

The final report to the implementation phase includes procedural flowcharts, record layouts, report layouts, and a workable plan for implementing the candidate system design into an operational one. Conversion is one aspect of implementation.

**System Maintenance**

Maintenance is actually the implementation of the review plan. As important as it is, many programmers and analysts are to perform or identify themselves with the maintenance effort. There are psychological, personality and professional reasons for this. Analysts and programmers spend far more time maintaining programs than they do writing them. Maintenance accounts for 50-80 percent of total system development.

Maintenance is expensive. One way to reduce the maintenance costs are through maintenance management and software modification audits.

* Maintenance is not as rewarding as exciting as developing systems. It is perceived as requiring neither skill not experience.
* Users are not fully cognizant of the maintenance problem or its high cost.
* Few tools and techniques are available for maintenance.
* A good test plan is lacking.
* Standards, procedures, and guidelines are poorly defined and enforced.
* Programs are often maintained without care for structure and documentation.
* There are minimal standards for maintenance.
* Programmers expect that they will not be in their current commitment by time their programs go into the maintenance cycle.

**SYSTEM DESIGN**

System design is "the process of studying a procedure or business in order to identify its goals, purposes and create systems and procedures that will achieve them in an efficient way". Another view sees system analysis as a problem-solving technique that breaks down a system into its component pieces for the purpose of the studying how well those component parts work and interact to accomplish their purpose.

The field of system analysis relates closely to requirements analysis or to operations research. It is also "an explicit formal inquiry carried out to help a decision maker identify a better course of action and make a better decision than they might otherwise have made."

* **DESIGN NOTATION**

Design notations are used when planning and should be able to communicate the purpose of a program without the need for formal code. Commonly used design notations are:

* DFD
* ERD
* **DFD (DATA FLOW DIAGRAM):**

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an [information system](https://en.wikipedia.org/wiki/Information_system), modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the [visualization](https://en.wikipedia.org/wiki/Data_visualization) of [data processing](https://en.wikipedia.org/wiki/Data_processing) (structured design). A DFD shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel unlike a [flowchart](https://en.wikipedia.org/wiki/Flowchart) which also shows this information.

Data flow diagrams were popularized in the late 1970s, arising from the book Structured Design, by computing pioneers Ed Yourdon and Larry Constantine. They based it on the “data flow graph” computation models by David Martin and Gerald Estrin. The structured design concept took off in the software engineering field, and the DFD method took off with it. It became more popular in business circles, as it was applied to business analysis, than in academic circles

.

**DFD SYMBOLS:**

Process that transforms data flow

Source or Destination of data

Data flow

Data store

* **ENTITY RELATIONSHIP DIAGRAM**

The relation upon the system is structure through a conceptual ER-Diagram, which not only specifies the existential entities but also the standard relations through which the system exists and the cardinalities that are necessary for the system state to continue. The Entity Relationship Diagram (ERD) depicts the relationship between the data objects. The ERD is the notation that is used to conduct the data modeling activity the attributes of each data object noted is the ERD can be described resign a data object descriptions.

The set of primary components that are identified by the ERD are

* + Data object
  + Relationships
  + Attributes
  + Various types of indicators

The primary purpose of the ERD is to represent data objects and their relationships.

**ER-DIAGRAM SYMBOL**

Entity

Relationship

Flow

* **INPUT DESIGN**

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data into a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy.

Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system. It is achieved by creating user-friendly screens for the data entry to handle large volume of data.

The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities. When the data is entered it will check for its validity. Data can be entered with the help of screens.

* **DATABASE DESIGN**

Data base is designed to manage large bodies of information. The management of data involves both the definitions of structures for the storage of information. In addition the data base system must provide for the safety of the information solved, despite system crashes or due to attempts at unauthorized access. For developing an efficient database user have to fulfill certain conditions such as controlled redundancy.

* Defining the data
* Inputting the data
* Locating the data
* Accessing the data
* Communicating the data

Revising the data

**Objectives of Database design**

For designing data base design several objectives have to be met as follows:

* Ease of use
* Control of data integrity
* Control of redundancy
* Control of security
* Data independence (logical & physical)
* Data storage protection
* System performance
* **OUTPUT DESIGN**

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making.

Output design generally refers to the results and information that are generated by the system for many end-users; output is the main reason for developing the system and the basis on which they evaluate the usefulness of the application.

In this Online Repository System project output is to view customer details, employee list, order tracking details and attendance percentage result.

**SYSTEM FLOW DIAGRAM**

**DATA FLOW DIAGRAM**

**LEVEL 0:**

User

Admin

Search for product

Request

Get results

Response

**LEVEL 1:**

User

Register

Login

Search for product

Add to cart

**Level 2**

Admin

Login

Product management

(Edit/ Delete)

Order management

(All Orders)

**Use case diagram**

USER

ADMIN

Register/ Login

Login

Order Management

Add to cart

Product details

Search For Product

Order

User

Admin

**ER Digram**

Product Management

**DATABASE DESIGN:**

**Table Name: register**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **LENGTH** | **DESCRIPTION** |
| User\_id | Int | 10 | unique id |
| F\_name | Varchar | 30 | User First name |
| L\_name | Varchar | 30 | User Last name |
| Gen | Varchar | 30 | User Gender |
| Id | Varchar | 50 | User id |
| Pass | Varchar | 30 | User Password |
| Phone | Varchar | 20 | User Phone number |
| Add | Varchar | 300 | User Address |
| City | Varchar | 30 | User city |
| Coun | Varchar | 30 | User country |
| Dob | Varchar | 30 | User date of birth |

**Table Name: category**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **LENGTH** | **DESCRIPTION** |
| Cat\_id | Varchar | 30 | Category unique id |
| Category | varchar | 50 | Product category |

**Table Name: subcategory**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **LENGTH** | **DESCRIPTION** |
| subcat\_id | Int | 10 | Sub-Category unique id |
| Cat\_id | varchar | 30 | Category unique id |
| Subcategory | Varchar | 50 | Subcategory details of product |

**CONCLUSION:**

The ‘Online Shopping’ is designed to provide a web-based application that would make searching, viewing and selection of a product easier. The search engine provides an easy and convenient way to search for products where a user can Search for a product interactively and the search engine would refine the products available based on the user’s input. The user can then view the complete specification of each product. They can also view the product reviews and also write their own reviews. The components that are used would make the application interactive and prevents annoying post backs. Its drag and drop feature would make it easy to use.

**BIBLIOGRAPHY**

**BOOKS:**

* Luke Welling, Laura Thomson, **“PHP and MYSQL Web Development”,** Fourth Edition, Publisher: Pearson Education Inc, Year: 2008.
* RasmusLerdof, Kevin Tatroe, **“PROGRAMMING PHP”** Second Edition, and Publisher: O’Reilly Media Inc, Year: 2006.
* W.Jason Gilmore, **“BEGINNING PHP and MYSQL: From Novice to Professional”** Third Edition, Publisher: Apress, year: 2008.
* Brad Bulger, Jay Greenspan, David Wall, **“MySQL and PHP Database Applications”,** Second Edition, publisher: Wiley, Year: 2003.
* Matt Zandstra, **“Sams Teach Yourself PHP in 24 Hours”,** Third Edition, Publisher: Sams Publishing, year: 2003.
* Michael Kofler, **“The Definitive Guide to MySQL 5”,** Third Edition, Publisher: Apress, Year: 2005.

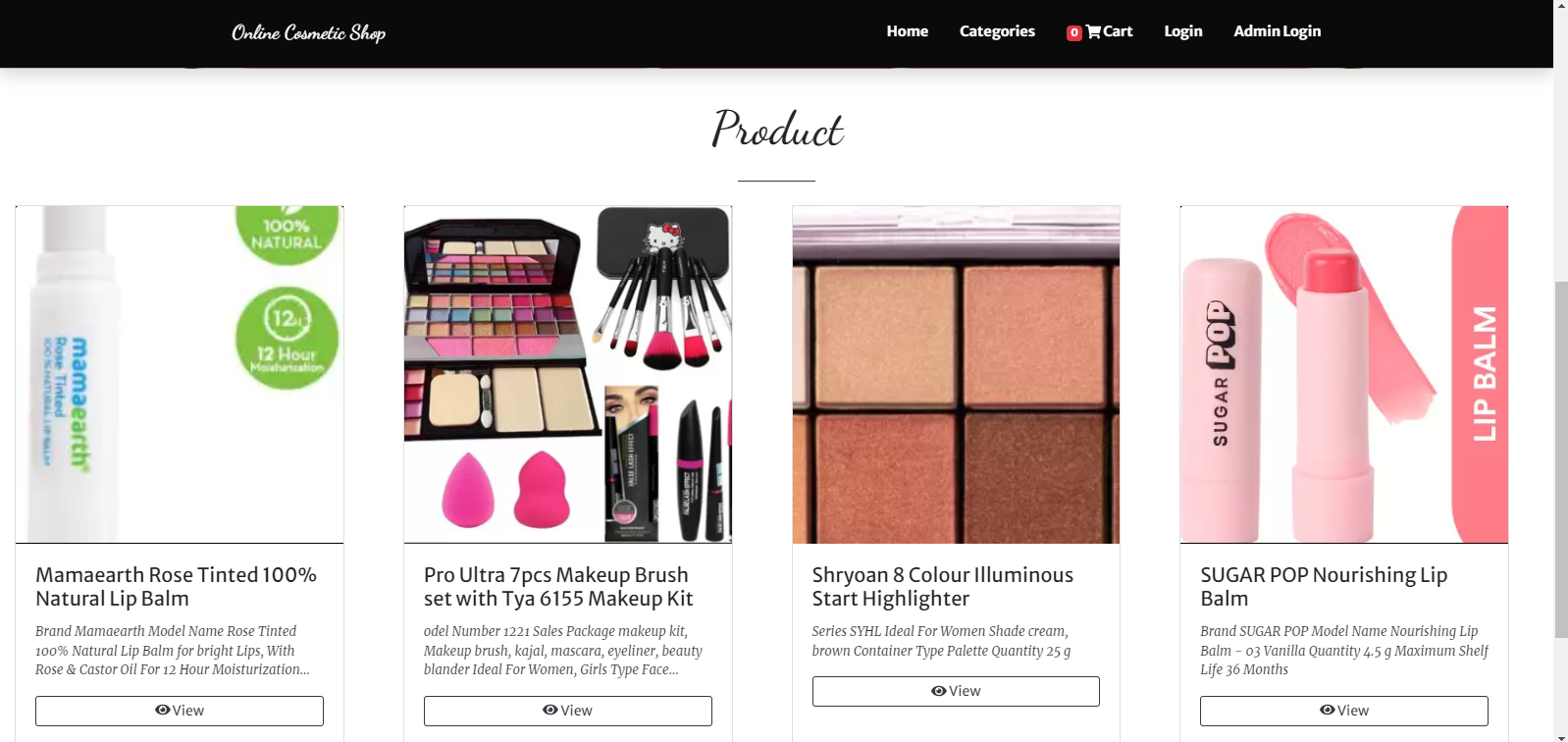
**WEB SITE REFRENCE:**

* www.php.net
* www.mysql.com
* www.planet-php.net
* www.w3schools.com

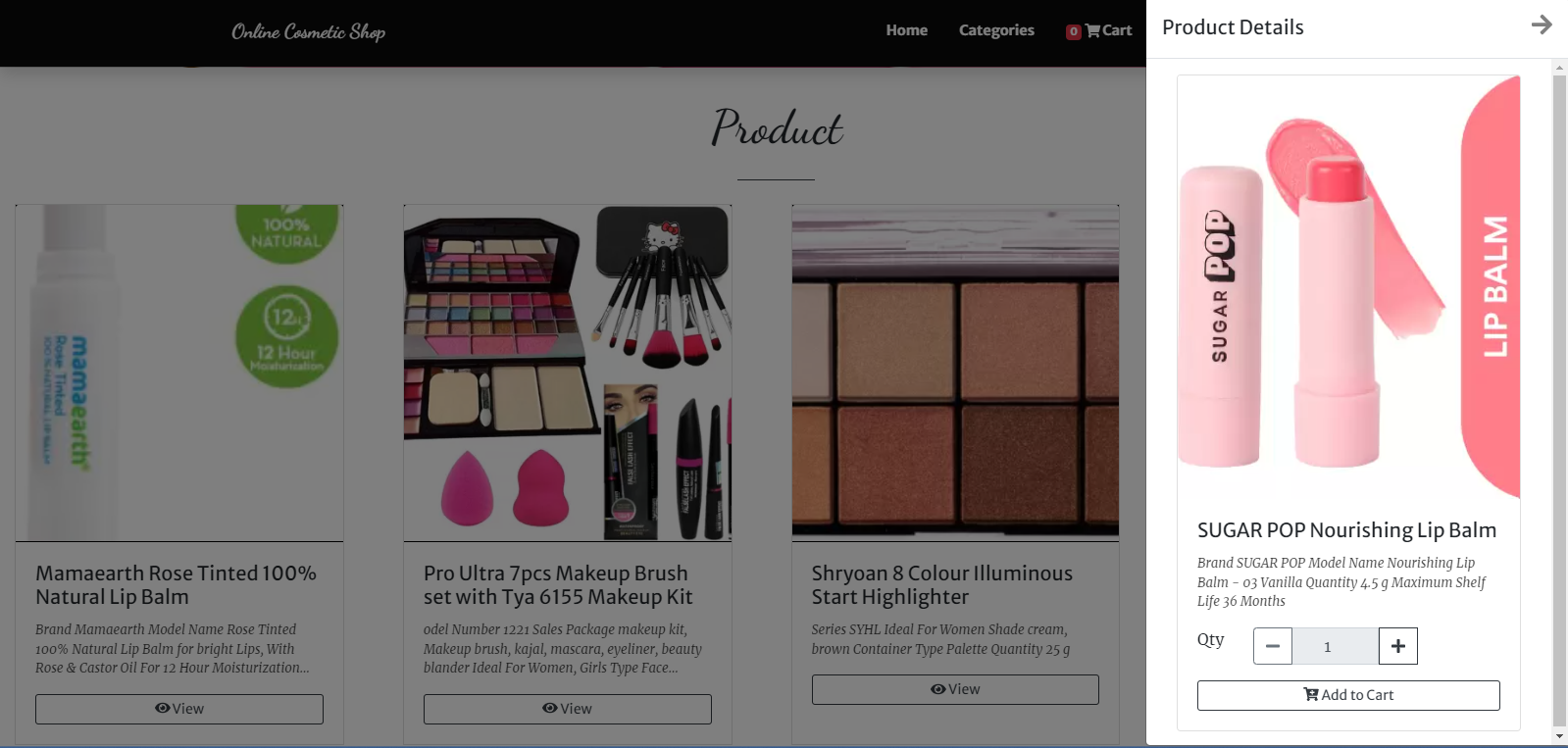
**SCREENSHOT:**

**Home page:**

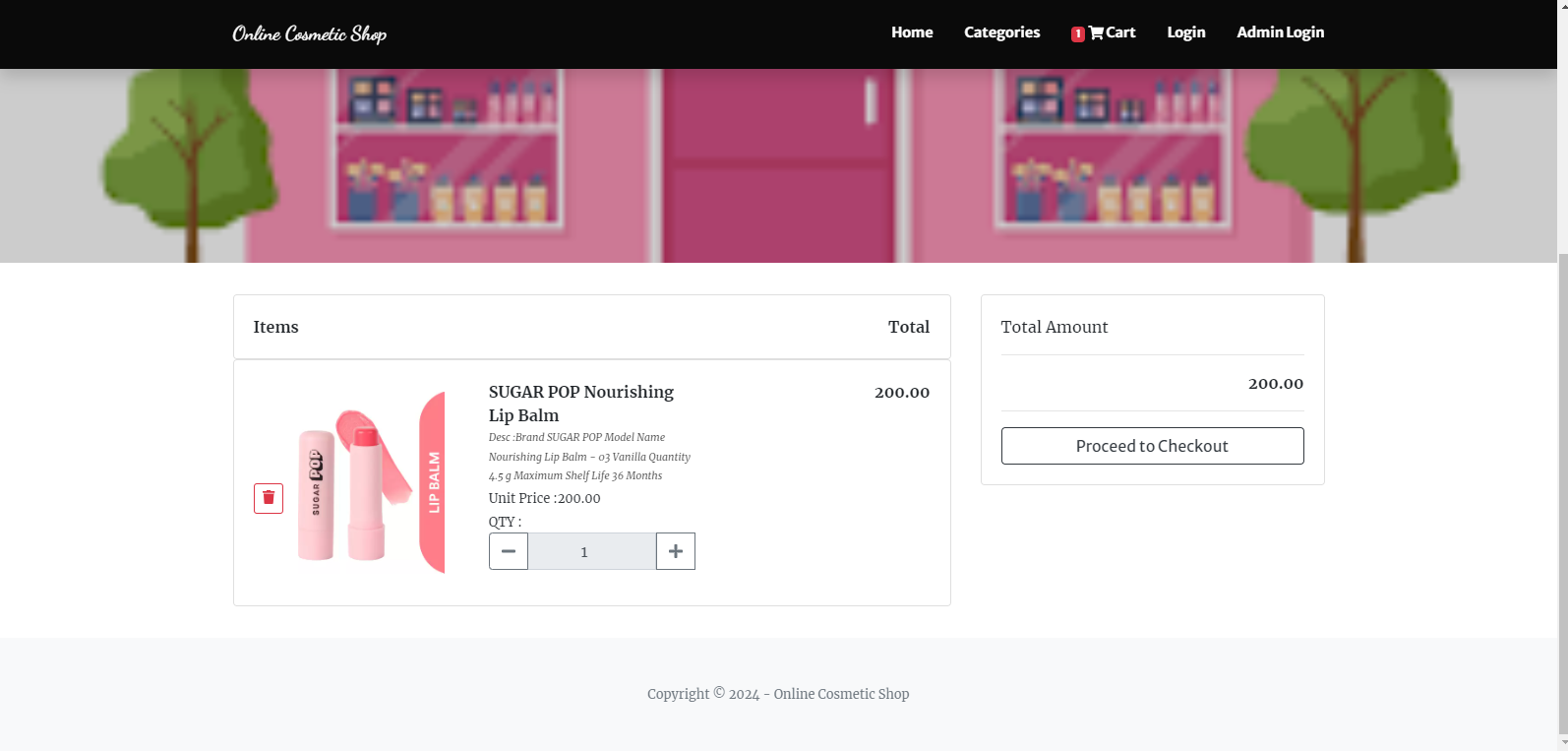
****

****

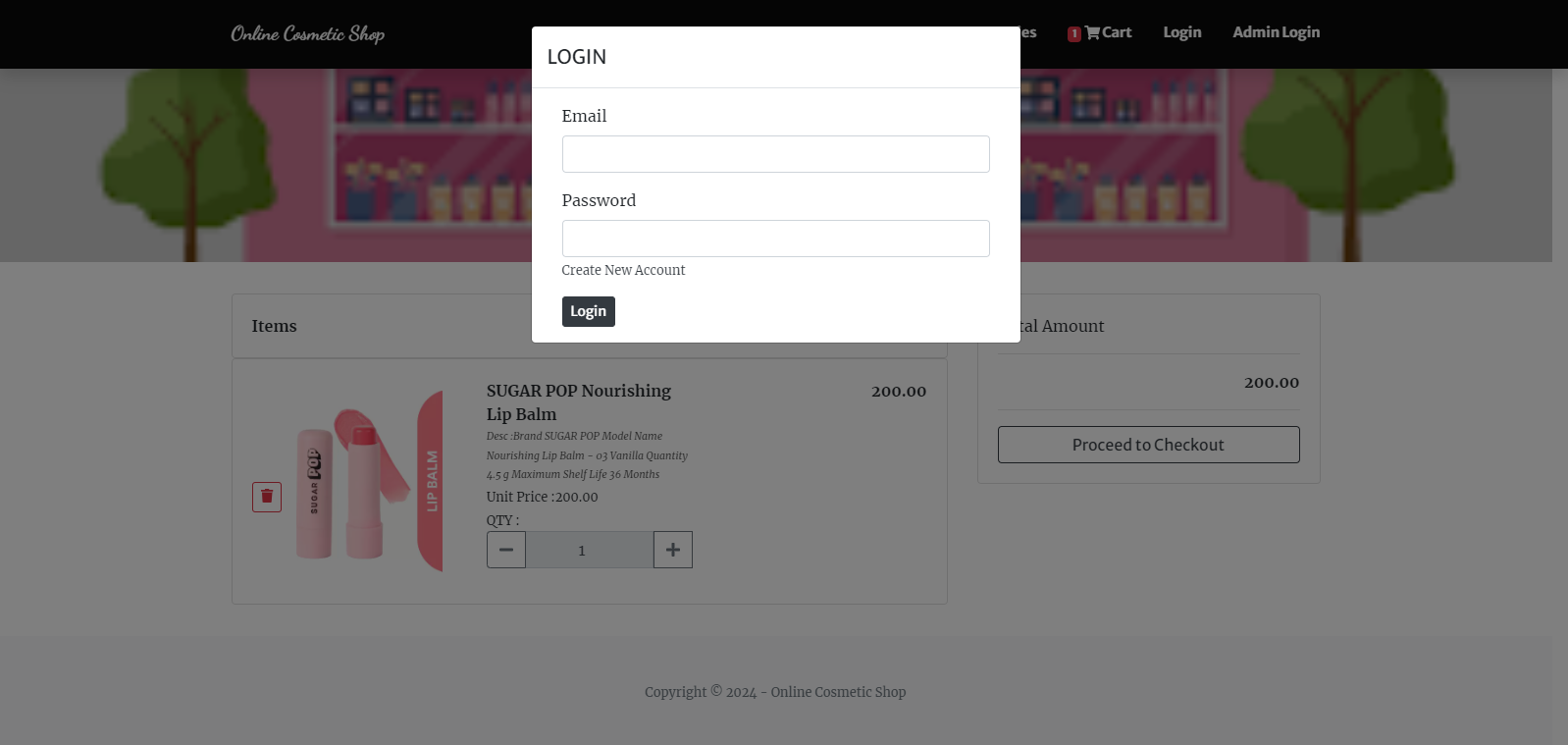
**Product Detail View:**

****

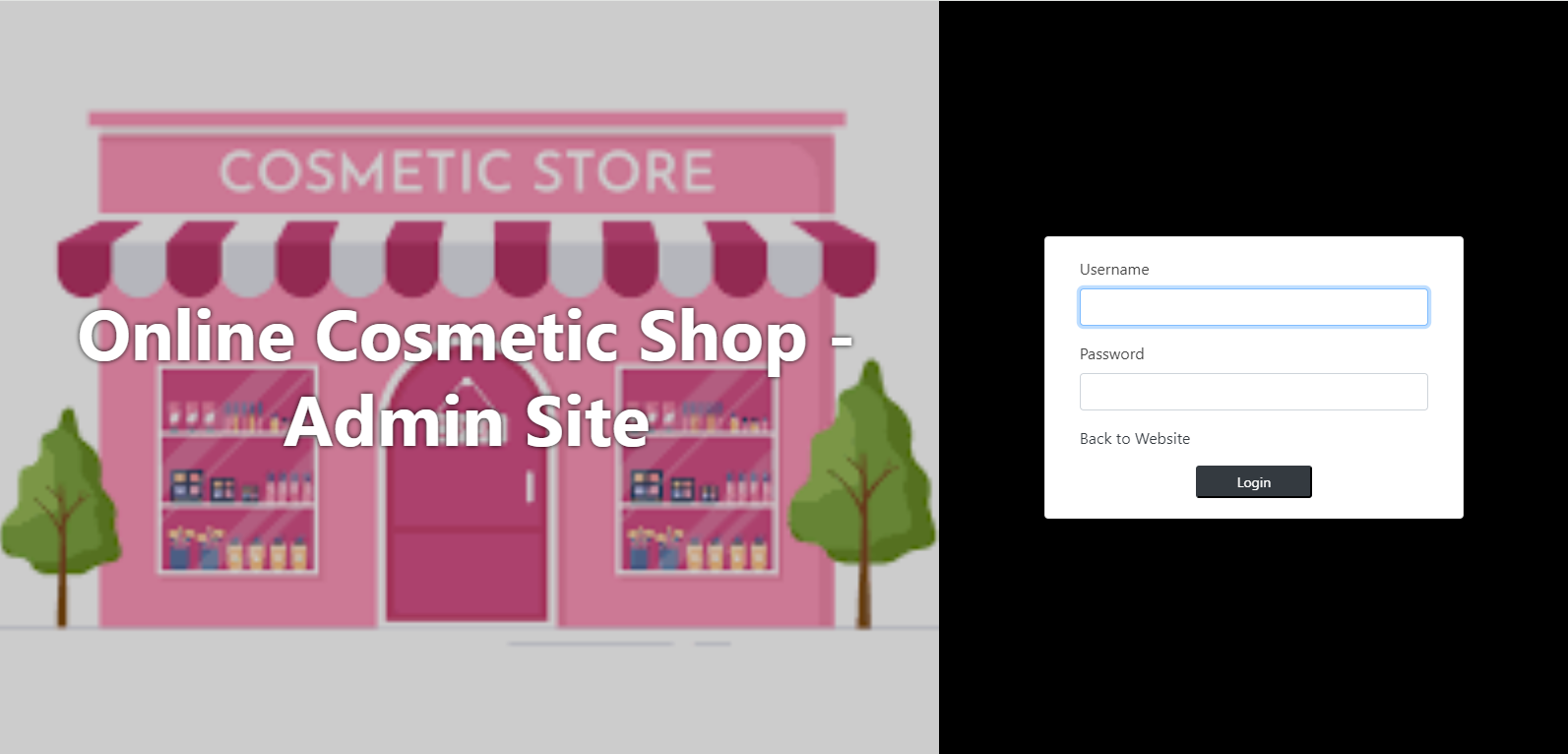
**Cart Page:**

****

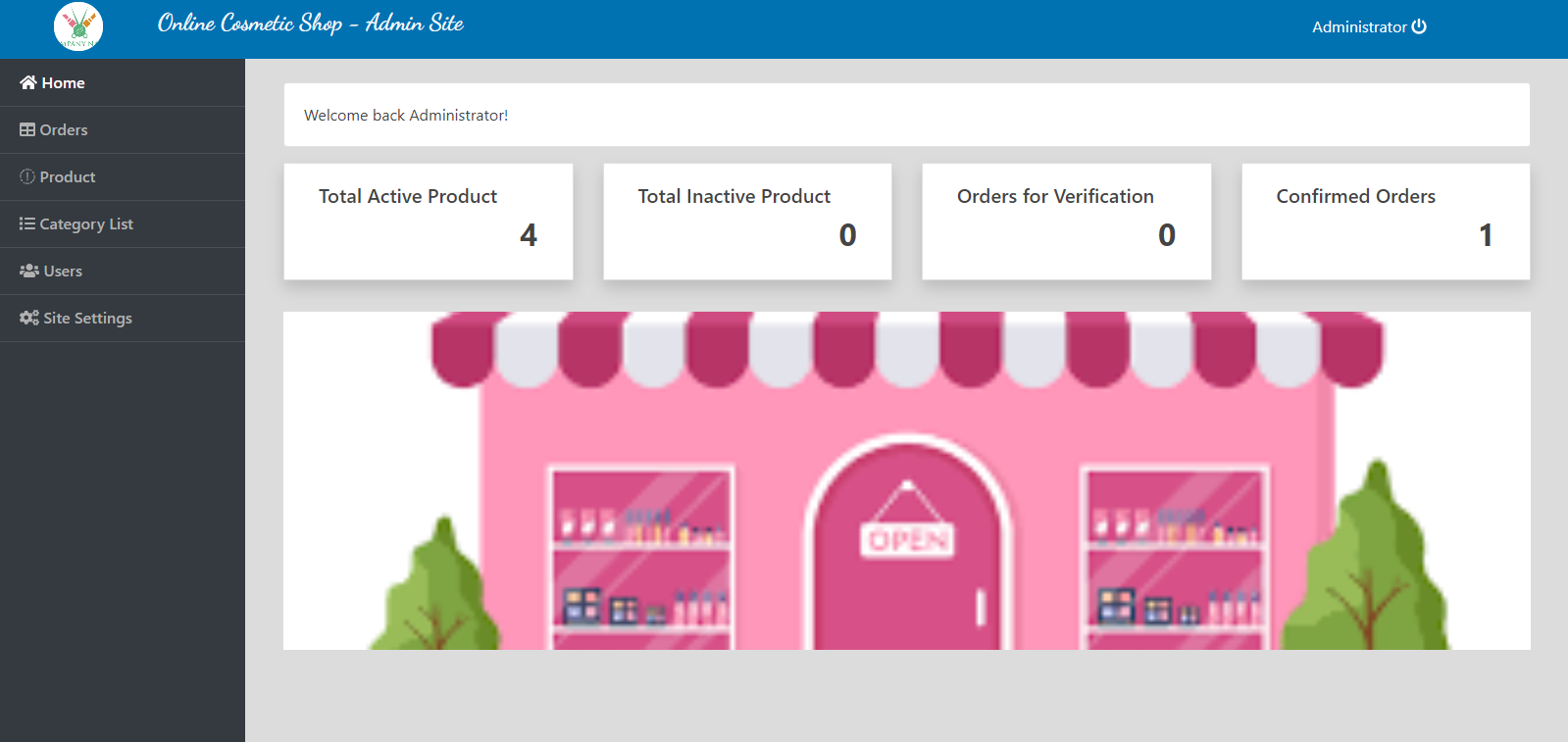
**User Login:**

****

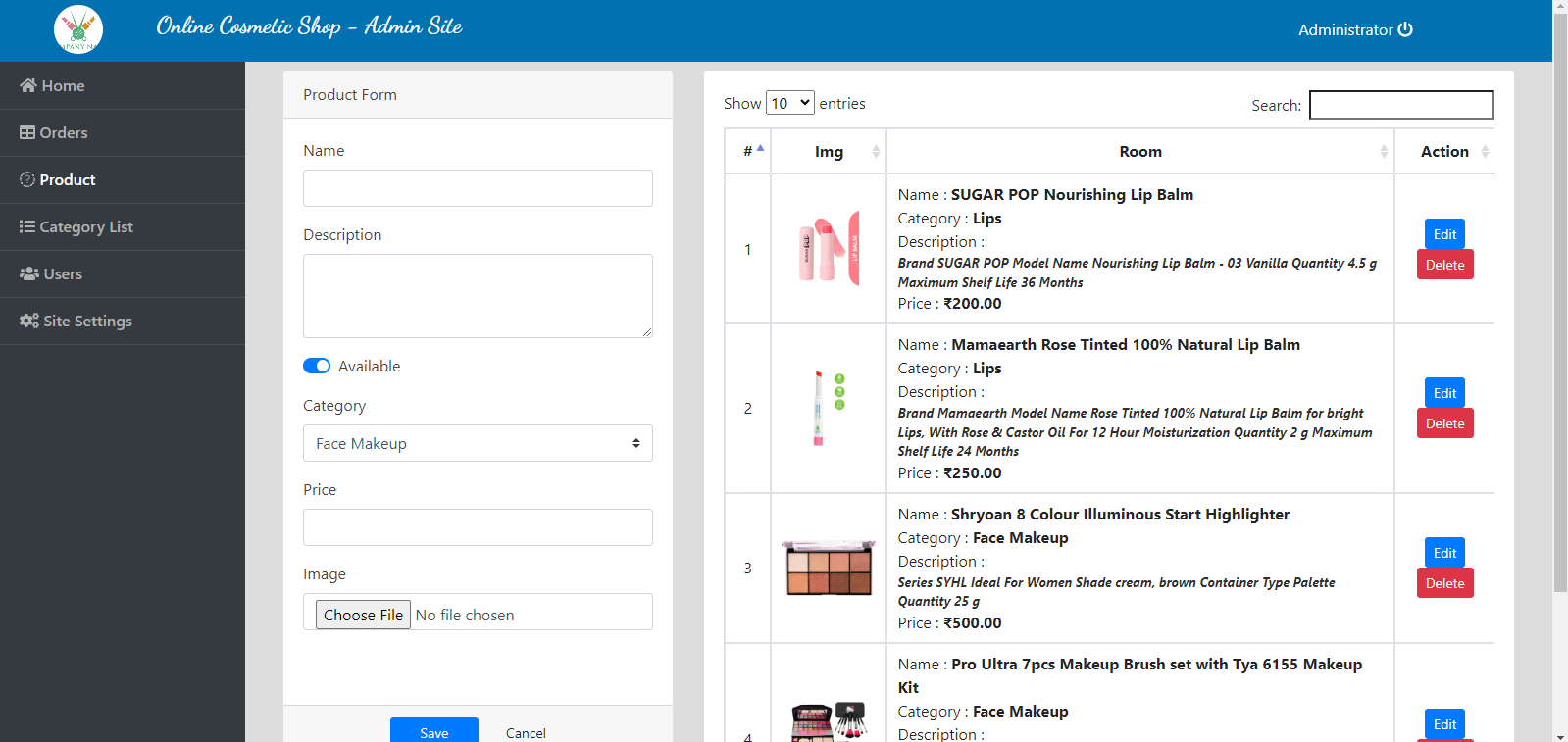
**Admin Login Page:**

****

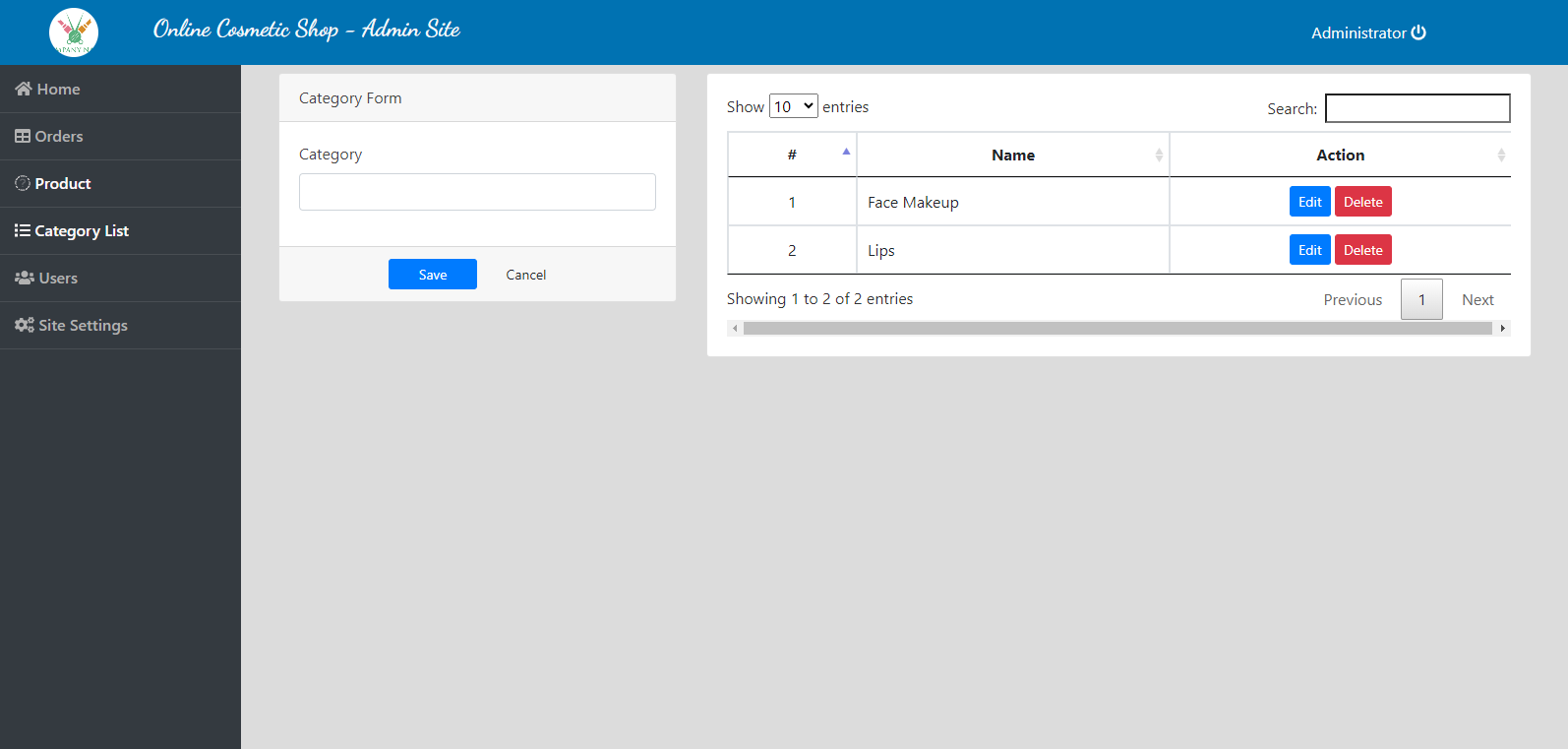
**Admin Dashboard:**

****

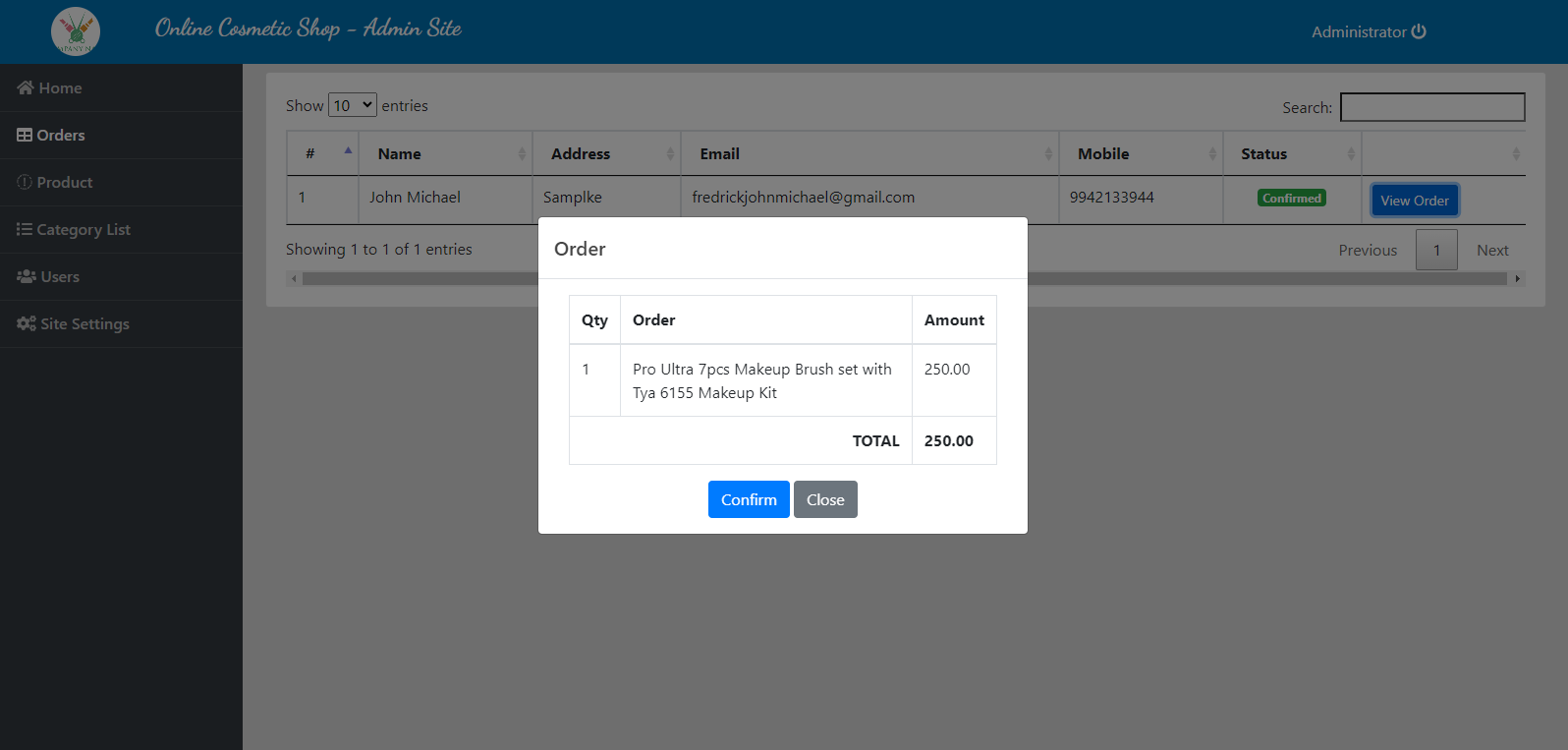
**Admin add item:**

****

**Admin Category:**

****

**Admin Order Management:**

****

**Sample Code:**

<?php

include'admin/db\_connect.php';

$qry = $conn->query("SELECT \* FROM product\_list where id = ".$\_GET['id'])->fetch\_array();

?>

<div class="container-fluid">

<div class="card ">

<img src="assets/img/<?php echo $qry['img\_path'] ?>" class="card-img-top" alt="...">

<div class="card-body">

<h5 class="card-title"><?php echo $qry['name'] ?></h5>

<p class="card-text truncate"><?php echo $qry['description'] ?></p>

<div class="form-group">

</div>

<div class="row">

<div class="col-md-2"><label class="control-label">Qty</label></div>

<div class="input-group col-md-7 mb-3">

<div class="input-group-prepend">

<button class="btn btn-outline-secondary" type="button" id="qty-minus"><span class="fa fa-minus"></button>

</div>

<input type="number" readonly value="1" min = 1 class="form-control text-center" name="qty" >

<div class="input-group-prepend">

<button class="btn btn-outline-dark" type="button" id="qty-plus"><span class="fa fa-plus"></span></button>

</div>

</div>

</div>

<div class="text-center">

<button class="btn btn-outline-dark btn-sm btn-block" id="add\_to\_cart\_modal"><i class="fa fa-cart-plus"></i> Add to Cart</button>

</div>

</div>

</div>

</div>

<style>

#uni\_modal\_right .modal-footer{

display: none;

}

</style>

<script>

$('#qty-minus').click(function(){

var qty = $('input[name="qty"]').val();

if(qty == 1){

return false;

}else{

$('input[name="qty"]').val(parseInt(qty) -1);

}

})

$('#qty-plus').click(function(){

var qty = $('input[name="qty"]').val();

$('input[name="qty"]').val(parseInt(qty) +1);

})

$('#add\_to\_cart\_modal').click(function(){

start\_load()

$.ajax({

url:'admin/ajax.php?action=add\_to\_cart',

method:'POST',

data:{pid:'<?php echo $\_GET['id'] ?>',qty:$('[name="qty"]').val()},

success:function(resp){

if(resp == 1 )

alert\_toast("Order successfully added to cart");

$('.item\_count').html(parseInt($('.item\_count').html()) + parseInt($('[name="qty"]').val()))

$('.modal').modal('hide')

end\_load()

}

})

})

</script>