**CSE1005 SOFTWARE ENGINEERING LABORATARY**

**ONLINE BILLING SOFTWARE**

**PREPARED BY**

**GONUGUNTLA ROHINI-20BCD7174(Team Lead)**

**PRATHAPANI SATWIKA-20BCD7160**

**MEKALA POORNIMAI-20BCD7167**

**GODUGU SHARMINA-20BCD7084**

**DATE:09/12/2022**

# ONLINE BILLING SOFTWARE

**Problem Analysis**

**Overview of the project:**

In the past, most places of business had larger crowds, and customers frequently had to wait up to an hour just to get their money. Even the smallest businesses are adapting to new technologies, and the traditional methods of creating invoices from scratch have been replaced by the "Web based Billing Program," a web-based software that automates the billing and management of a network of department stores.

This project consists of a number of modules, including the register/login, admin, bill management, report generating, and giveaway modules. Customers may read bills, business owners can store or download invoices for new businesses, and firm owners can build up online bill management systems. It is built on a nationwide network of department stores. Additionally, it helps clients create, update, and calculate bills and invoices without having to manually list them.

**Why computerized?**

The current procedures involve a lot of paperwork and manual labor.

1. Large amounts of data are challenging to manually manage and preserve.
2. It's difficult to create reports from the data that's already available.
3. Time-consuming and involves a lot of paperwork.
4. lack of data modification capability.

By computerizing

It results in a safe, dependable, quick, and efficient management system for keeping bills/invoices or calculating entire bills. Additionally, it keeps digital records of data that can be stored for a specific amount of time and gives owners simple access to monthly, weekly, or yearly summary reports.

**Identification of project scope:**

1. The application is made to keep track of data from a wide range of businesses.
2. Order confirmation invoices to customer’s devices.
3. Provides an easy billing pattern solution.
4. Bill/Invoice creation is instantaneous.
5. On a regular basis, generates report summaries for the owners.
6. At times, a user-giveaway feature will be planned with a bill /invoice generation.

**Task involved:**

* + Feasibility study:
    1. Technical Feasibility
    2. Behavioral Feasibility
    3. Economic Feasibility
  + Risk identification
  + Implementation of security system
  + Database management system
  + Password and login management system

**OBJECTIVE:**

The major goal of this project is to generate and handle invoices quickly in order to reduce the amount of paperwork and time required to manually manage vast amounts of data. Customers and business owners gain from this effort. By giving owners summary reports, this programee enables the maintenance of all items with their shorthand notations, offering a practical solution to the billing pattern with Secured User Authentication and Validation. Additionally, use the giveaway generator when deals or discounts are offered.

1. Generating and Managing Invoices in a matter of few seconds.
2. Maintain all the products with their shorthand notation.
3. Provides summary reports to the owner monthly or weekly and yearly.
4. Provides a convenient solution to the billing pattern.
5. Make an easy to use environment for owners and customers.
6. Secured User Authentication and Validation.
7. User Giveaway generator.

**1.4 Infrastructure:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.**  **No.** | **Tools /**  **Technology needed** | **Description and uses** | **Type** |
| **1.** | HTML5 | Appearance of website | Front-end Software |
| **2.** | CSS3 | Adding style and presentation to the web-pages | Front-end Software |
| **3.** | JavaScript | Adding interactive user behavior to web pages | Front-end Software |
| **4.** | Bootstrap | front-end framework used for UI | Front-end Software |

|  |  |  |  |
| --- | --- | --- | --- |
| **5.** | PHP | Server side programming and interactive with phpMyAdmin | Back-end  Software |
| **6.** | MySQL | For the purpose of a web database and logging application | Database Software |
| **7.** | phpMyAdmin | Management of databases, tables, indexes, permissions | Database Software |
| **8.** | Lucid chart | a web-based proprietary platform to collaborate on drawing charts and  diagrams | Design Tool Software |
| **9.** | Heroku | a container-based cloud Platform as  a Service to deploy, manage, scale  apps | Cloud  Platform  Software |
| **10.** | Visual  Studio Code | a code editor redefined, optimized  for building, debugging web apps | Desktop  Software |
| **11.** | Chrome | a cross-platform web browser | Software |
| **12.** | Windows 10 | an operating system for building apps | OS |
| **13.** | PC / Laptop | Intel based processor-run computer system, which have keyboard and mouse as input devices. | Hardware |

**SOFTWARE REQUIREMENT ANALYSIS AND PLANNING**

**Description of individual phase/module:**

**User characteristics:**

**Administrator:**

The registered department store owners who have registered using this application can be monitored and contacted by the administrator.

**Registered user:**

The owner can then choose from the available items, add quantities, and print a bill for the buyer. The cost of a bill submitted on a particular date will be calculated by the online tool. If the owner needs to change the customer's information, they can do so on the Manage Bills page (involving edit, updating, delete details). For every bill paid, the Giveaway generator generates a giveaway for a random number of customers.

**General Constraints:**

Constant stable internet connection is required for WBS.

**Assumptions and Dependencies:**

WBS needs a steady, reliable online connection, and irregular updates might cause difficulties and issues when it comes to purchases and credits.

**Functional Requirements:**

**1)Signup/Registration:**

**Description:**

It displays the signup page for creating account.

**Input:** Store name, e-mail, mobile number., password, gstin/uin

**Output:**

It creates account.

**2)Login:**

**Description:**

It displays the login page.

**Input:** E-mail and password

**Output:** The homepage displays for the user.

**3) Bill generation:**

**Description:**

It creates bill for the specified products.

**Input:** Bill number, mode, name, mobile number., address, e-mail, item, code, quantity, price

**Output:** It generates bill.

**4)Product shorthand notations:**

**Description:**

It displays the page for creating shorthand for the product.

**Input:** Product code, product name, price

**Output:** Gives the shorthand representation for the product.

**5)Logout:**

**Description:**

User gets logged out.

**Input:** Nil

**Output:** Gets logged out

**Identify individual module deliverables**

**1 Signup / Registration Module:**

If a client wishes to create an invoice or manage, download, or print a bill, he or she must first register. Unregistered users are unable to do the above tasks.

1. **Login Module:**

The user logs in to the online application using a valid user id and password that has been successfully validated in order to manage invoices.

1. **Bill Generation Module:**

Every logged-in user will be able to create and generate invoices, as well as store current bills, print, and download those generated bills.

1. **Product shorthand notations Module:**

Every product has a shorthand representation of the entire product label. These shorthand notations can be found on user invoices, and users can search for them.

1. **Logout Module:**

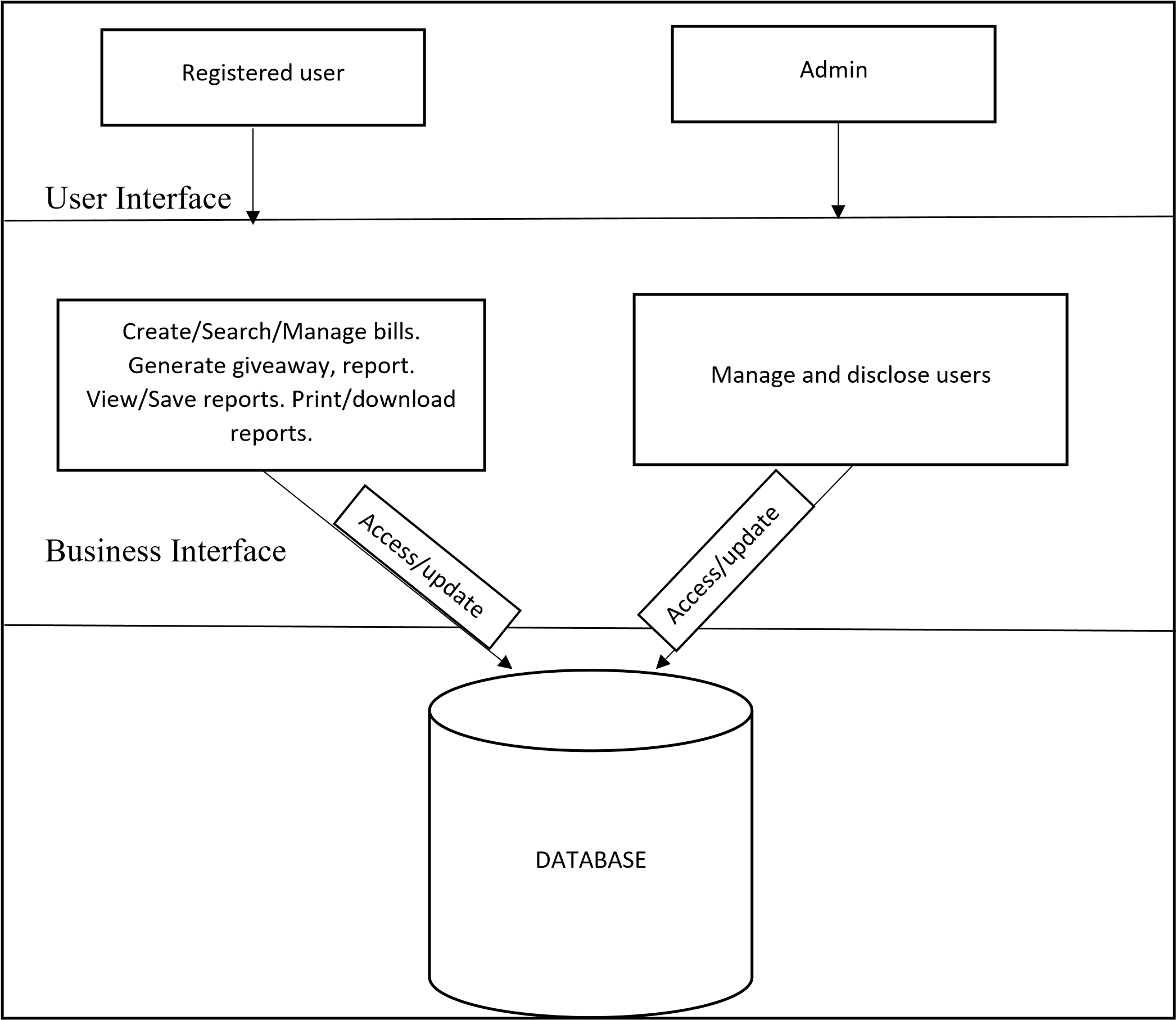
Users can logout after doing the following required processes and all session activities will be preserved in the database.

**DATA MODELING**

**System Architecture Design**

A system architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

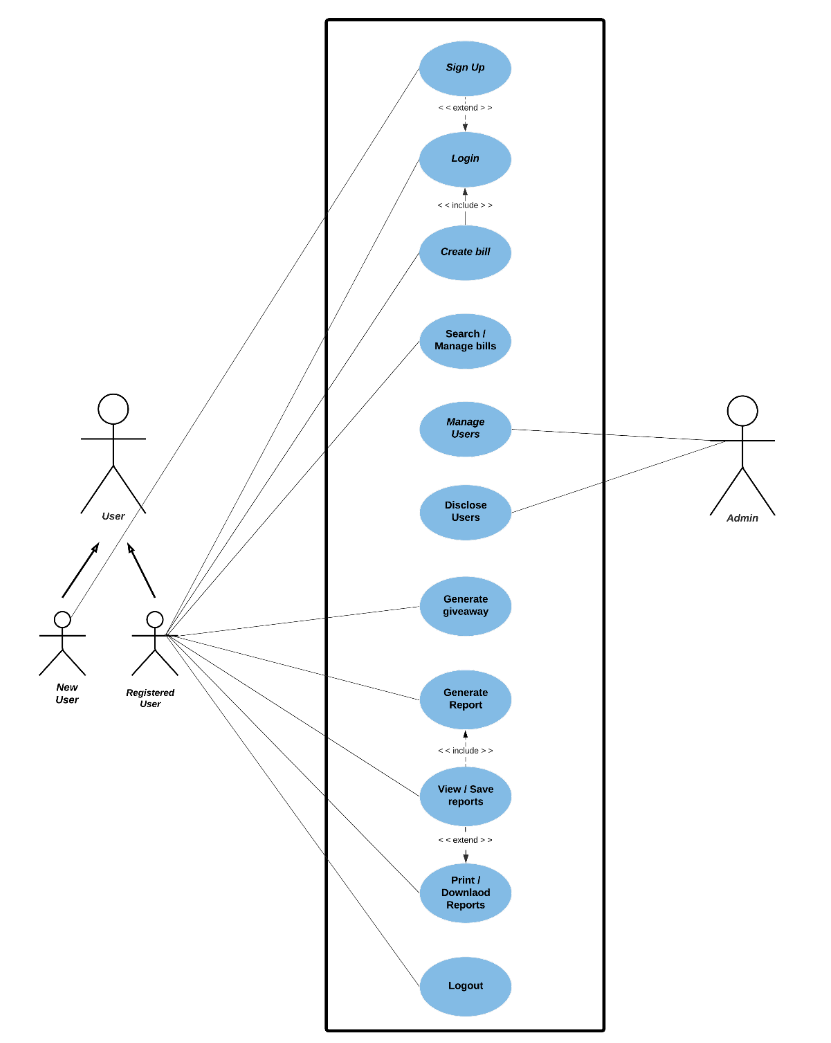
The system Architecture design for the project student mark analysis system.



**Use Case Diagram**

Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally.

The Use case diagram for the project Online Billing Software.



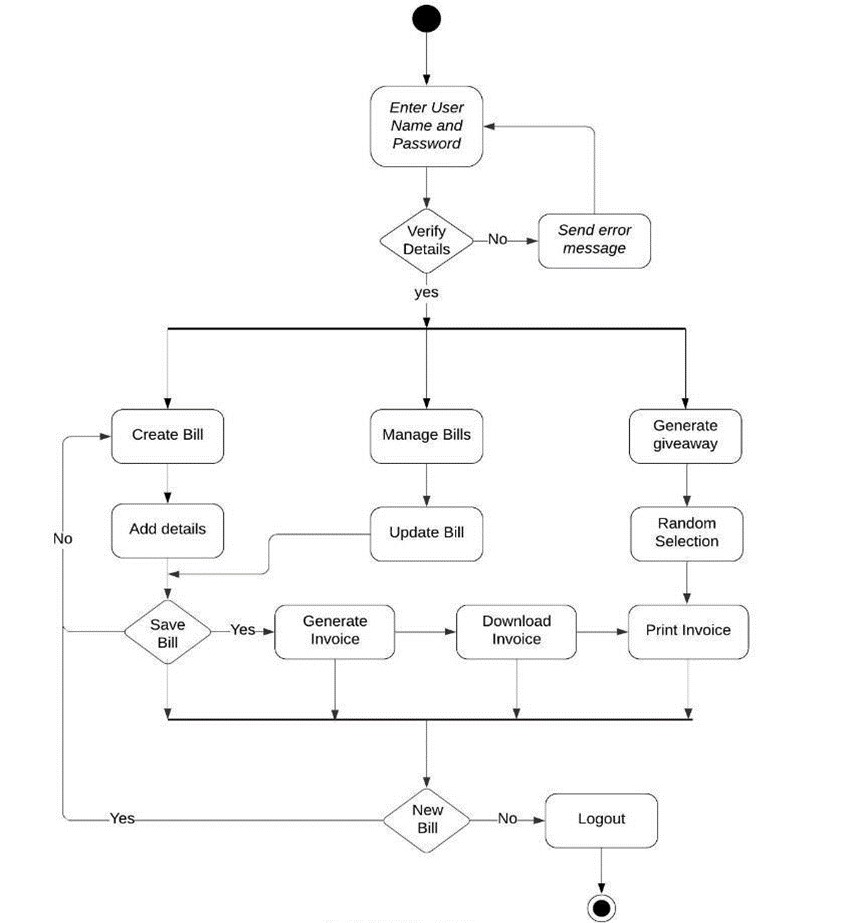
USE CASE diagram

**DATA MODELING**

**Activity Diagram**

An activity diagram is a behavioral diagram i.e. it depicts the behavior of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.

The Activity diagram for the project Student mark analysis system.



# ACTIVITY diagram

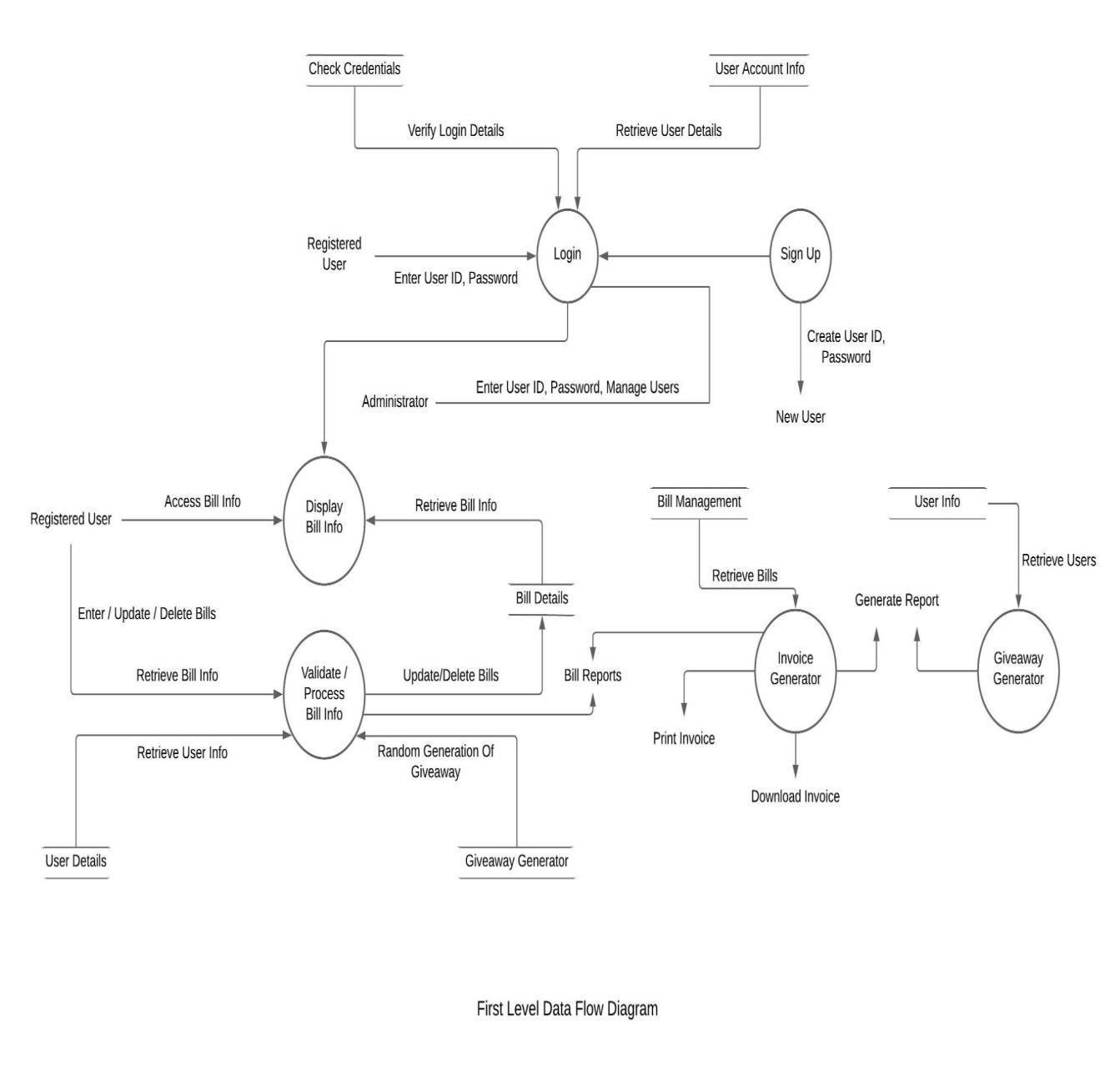
**DATA MODELLING:**

**DFD Diagram:**

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually “say” things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO.

The DFD diagram for online billing software

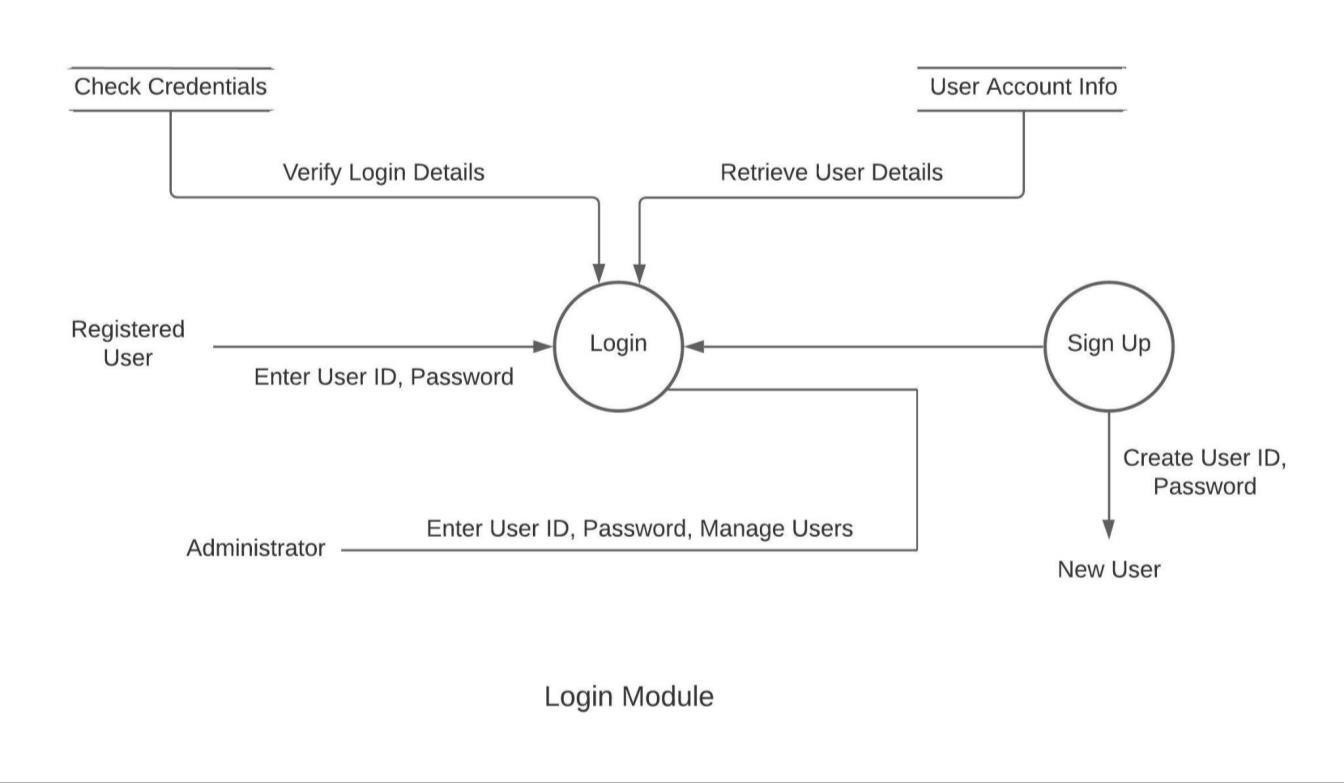
# Level DFD



# Level DFD

**.Login Module**

**1**



**2**



**.Bill Management Module**



**3**



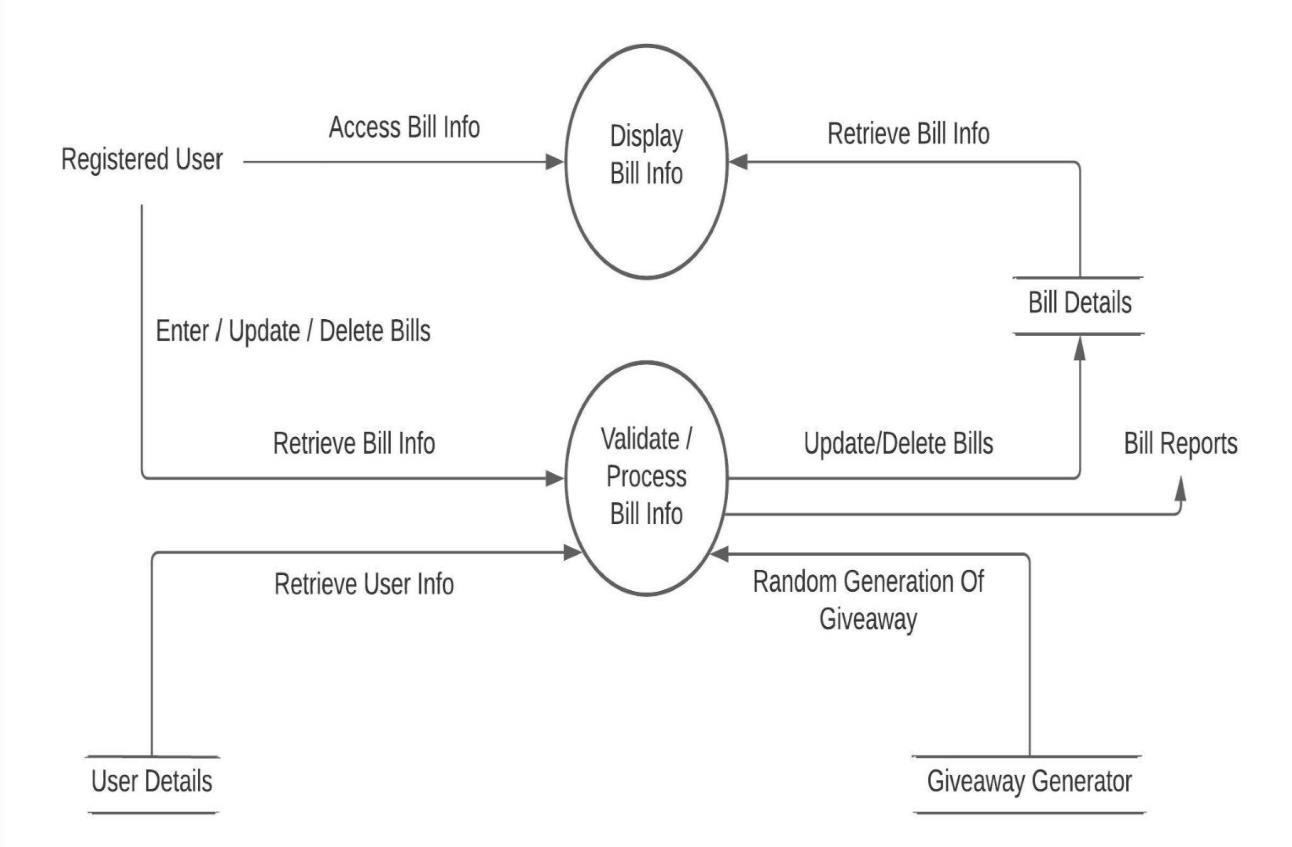
**.Report**



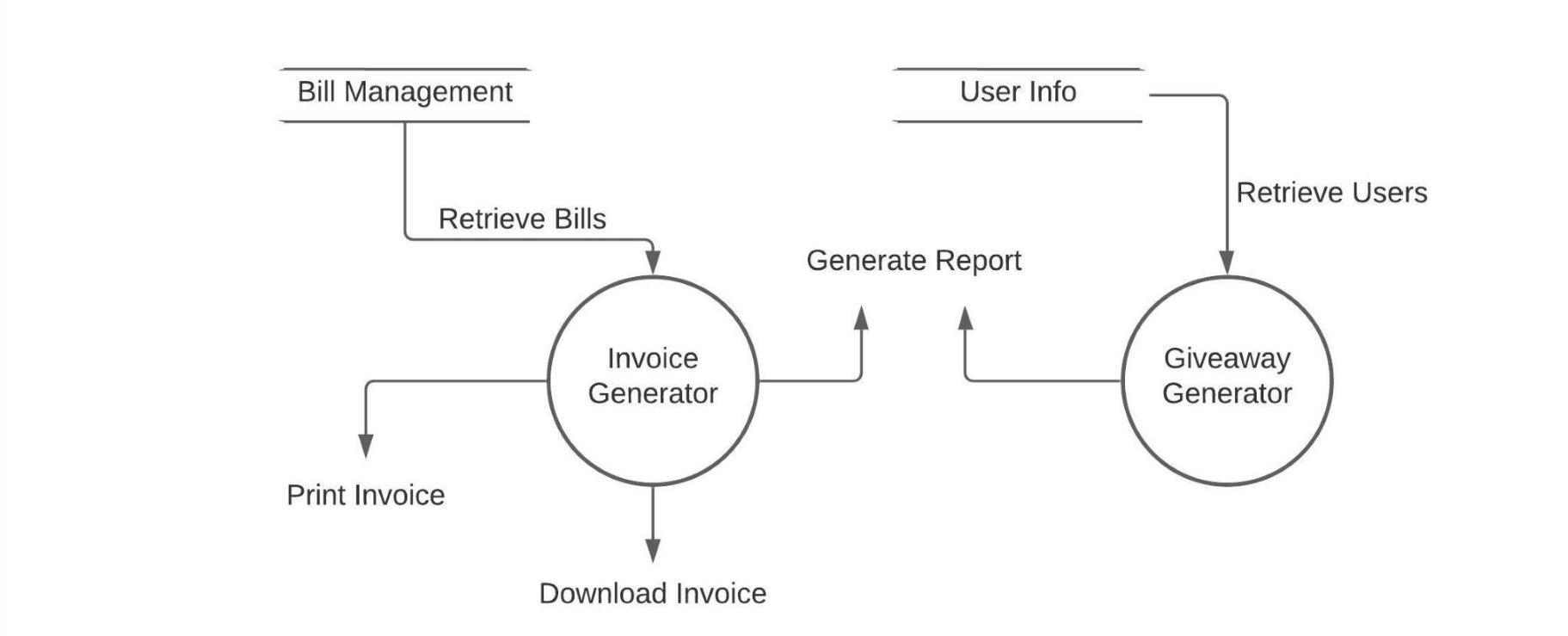
**G**



**eneration &Giveaway Module**



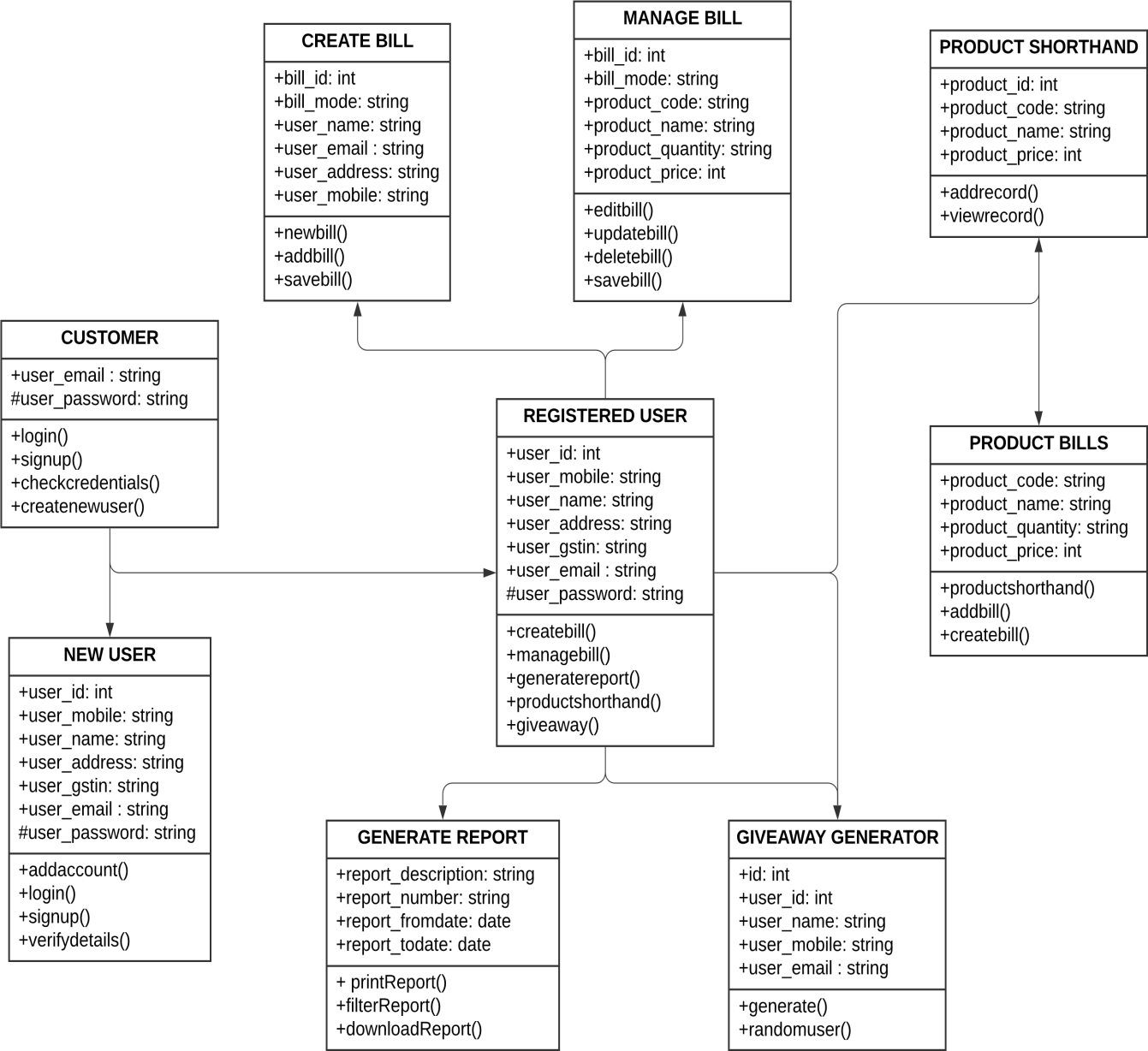
**3.Report Generation &Giveaway Module**



**DATA MODELLING:**

**Class diagram:**

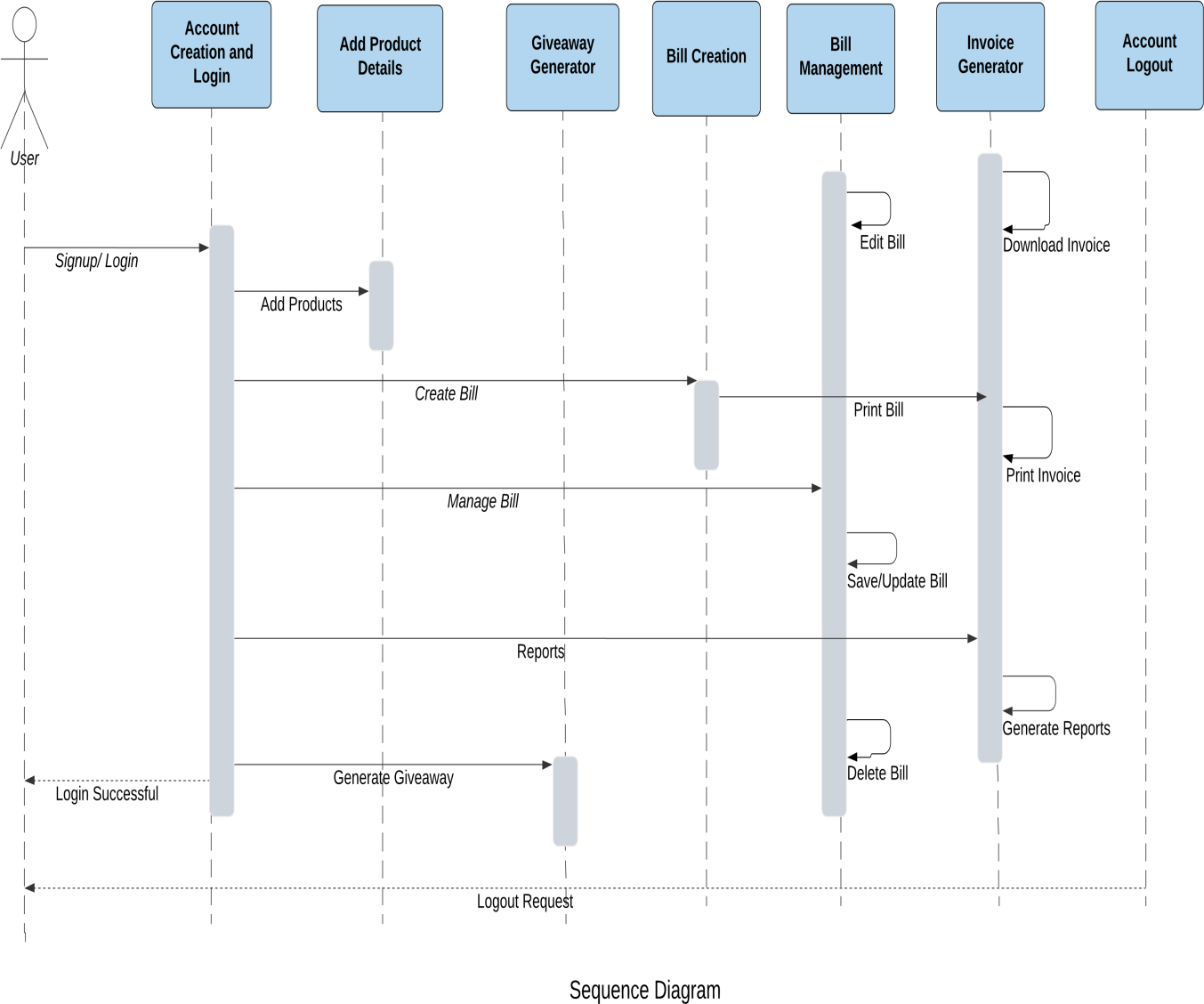
Class diagrams are the blueprints of your system or subsystem. You can use class diagrams to model the objects that make up the system, to display the relationships between the objects, and to describe what those objects do and the services that they provide. Class diagrams are useful in many stages of system design. The class diagram for online billing software



**Sequence Diagram:**

A sequence diagram is a Unified Modeling Language (UML) diagram that illustrates the sequence of messages between objects in an interaction. A sequence diagram consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction.

The sequence diagram for online billing software



**DEVELOPMENT**

**Database Structure**

**Sign Up :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD** | **DATA**  **TYPE** | **SIZE** | **CONSTRAINTS** | **DESCRIPTION** |
| Email | Varchar2 | 30 | Primary Key | A unique email id given by the user. |
| Store Name | Varchar2 | 30 | Not Null | Name of the store |
| Mobile  Number | Number | 10 | Not Null | Shows the mobile no. of the user. |
| Password | Varchar2 | 15 | Not Null | Password given by the user. |
| gst | Varchar2 | 15 | Not Null | A unique number is stored. |

**Login :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD** | **DATA**  **TYPE** | **SIZE** | **CONSTRAINTS** | **DESCRIPTION** |
| Email\_id | Varchar2 | 30 | Primary key | A unique email id that identifies the user. |
| Password | Varchar2 | 15 | Not Null | Password that allows the user to enter. |

**User :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD** | **DATA**  **TYPE** | **SIZE** | **CONSTRAINTS** | **DESCRIPTION** |
| User\_id | Number | 20 | Primary Key | A unique id that identifies the user. |
| Name | Varchar2 | 30 | Not Null | Shows the name of the user. |
| Email | Varchar2 | 30 | Not Null | A unique email id given by the user. |
| Mobile No. | Number | 10 | Not Null | Shows the mobile no. of the user. |
| Password | Varchar2 | 15 | Not Null | Password given by the user. |
| Gst | Varchar2 | 15 | Not Null | A unique number is stored. |

**Product :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD** | **DATA**  **TYPE** | **SIZE** | **CONSTRAINTS** | **DESCRIPTION** |
| Product Code | Number | 15 | Primary Key | Unique code given to a product |
| Product  Name | Varchar2 | 30 | Not Null | Describes the product |
| Price | Number | 20 | Not Null | Price of the product |

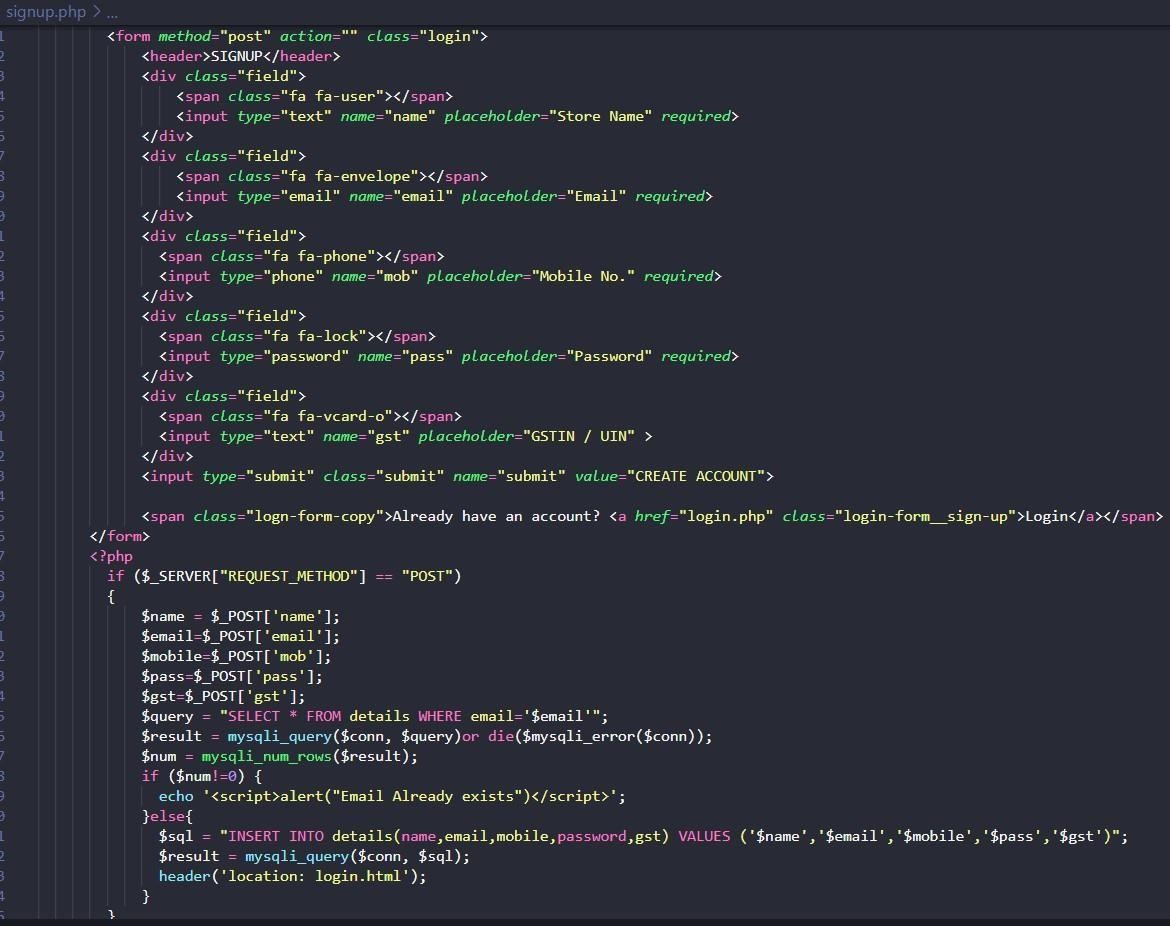
**Bills :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD** | **DATA**  **TYPE** | **SIZE** | **CONSTRAINTS** | **DESCRIPTION** |
| Bill\_id | Number | 20 | Primary key | A unique  number given to  the bill |
| Payment  Mode | Varchar2 | 20 | Not Null | Mode of the  Payment |
| Customer name | Varchar | 20 | Not Null | Name of the customer |
| Customer  address | Varchar2 | 50 | Not Null | Address of the customer |
| Customer  Mobile | Number | 10 | Not Null | Mobile number of the customer |
| Customer  Email | Varchar2 | 20 | Not Null | Email id of the customer |
| Product code | Number | 15 | Foreign Key | Unique code given to a product |
| Product quantity | Number | 30 | Not Null | Total quantity of the product purchased |
| Product price | Number | 20 | Not Null | Calculates the total price |
| Date | Varchar2 | 10 | Not Null | Date on which  bill is issued |

**DEVELOPMENT**

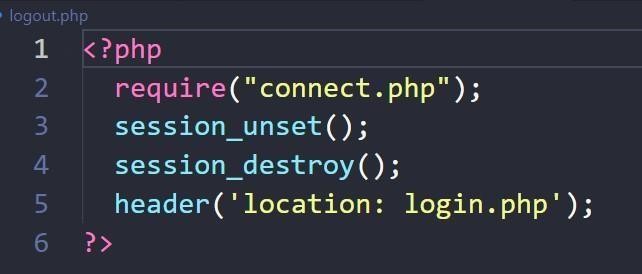
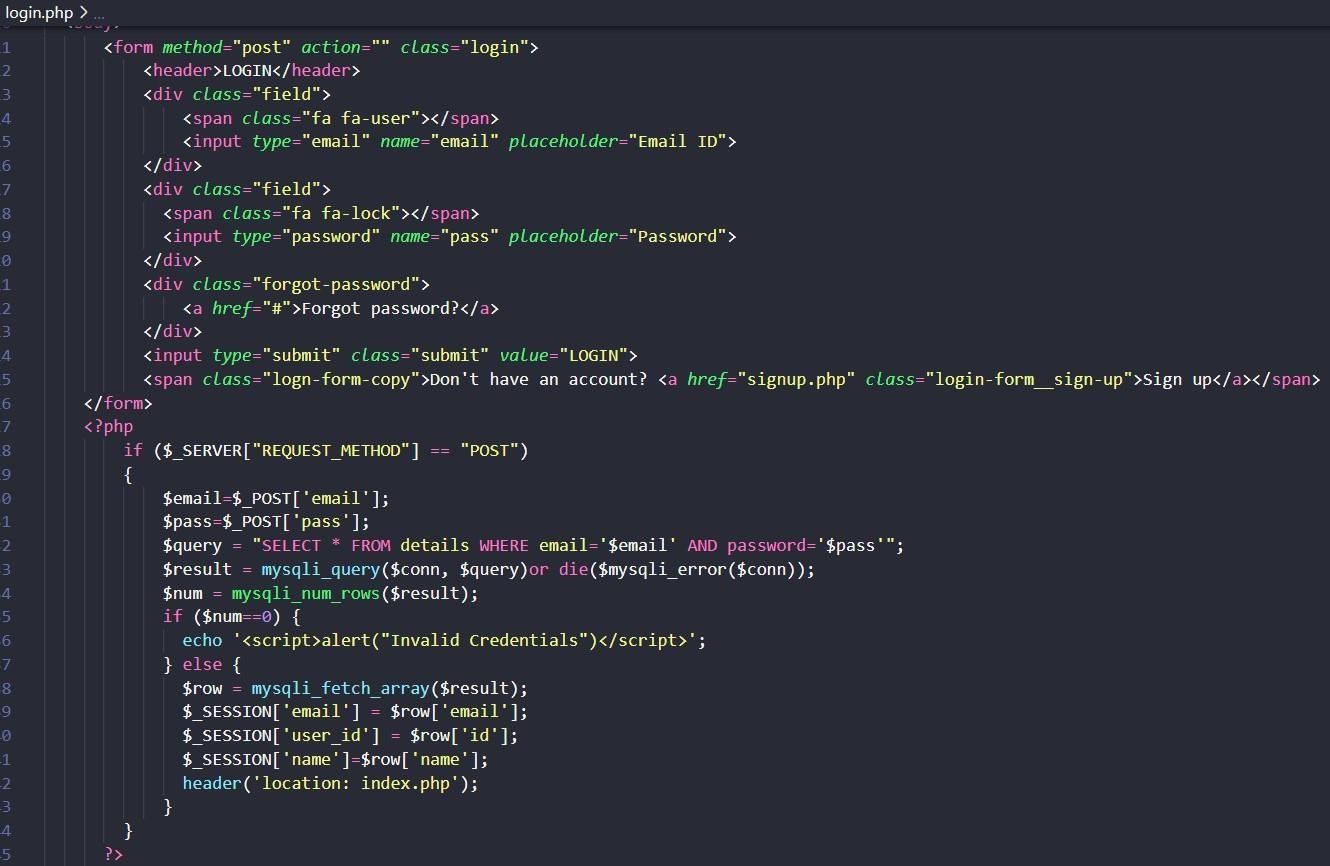
**Coding and Implementation**

**USER SIGN UP**

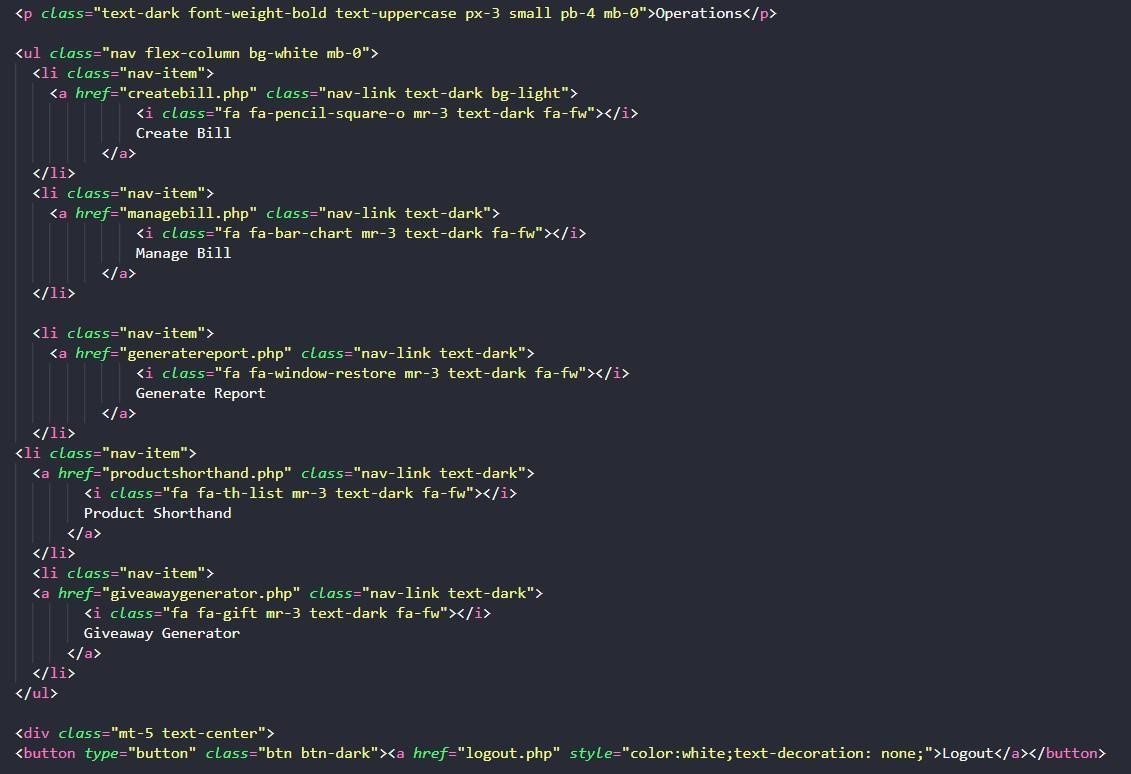


**USER LOGIN**

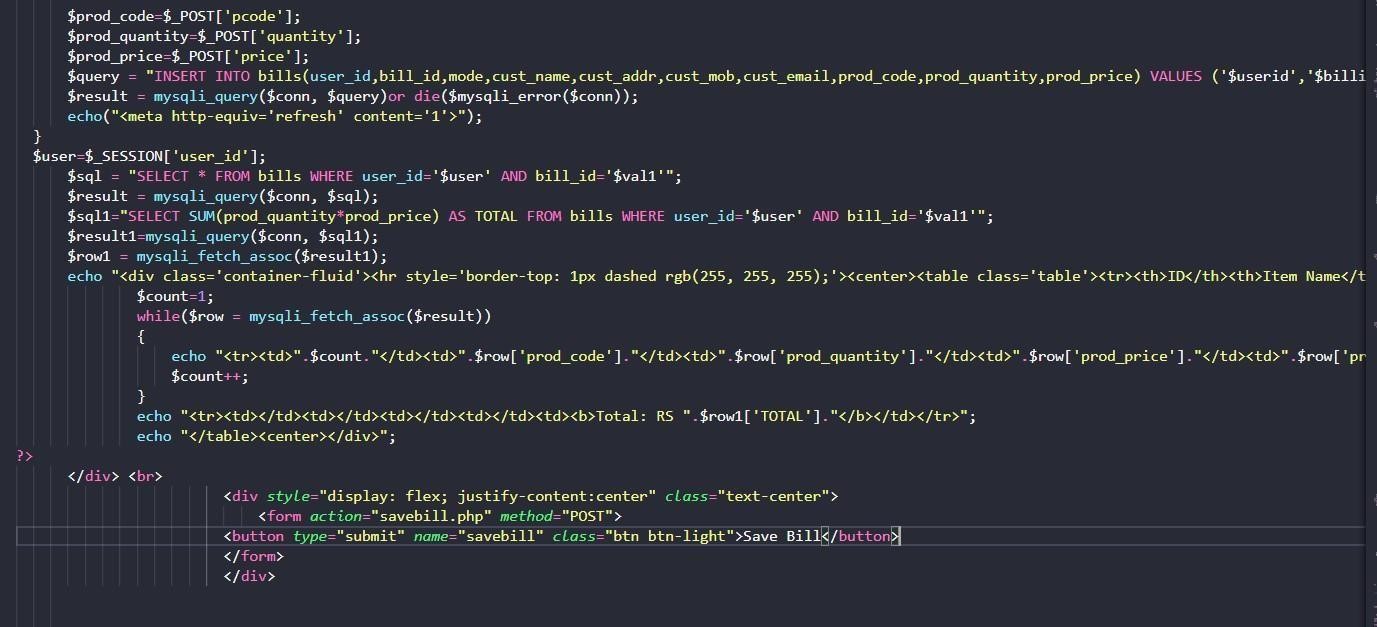
**USER LOGOUT**



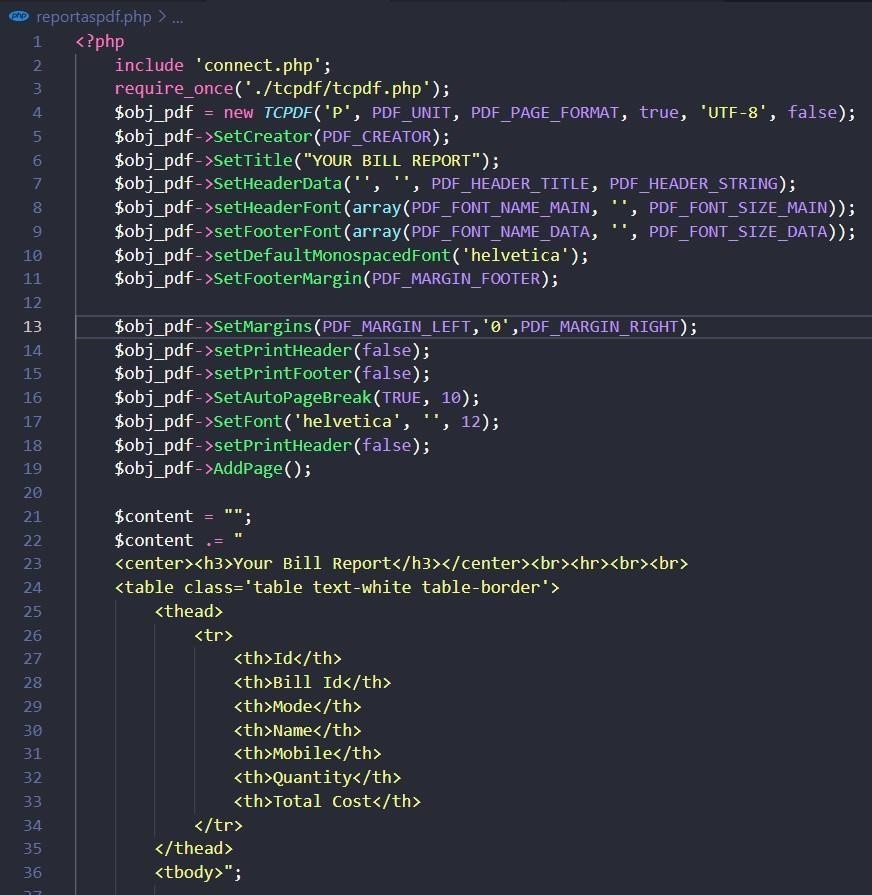
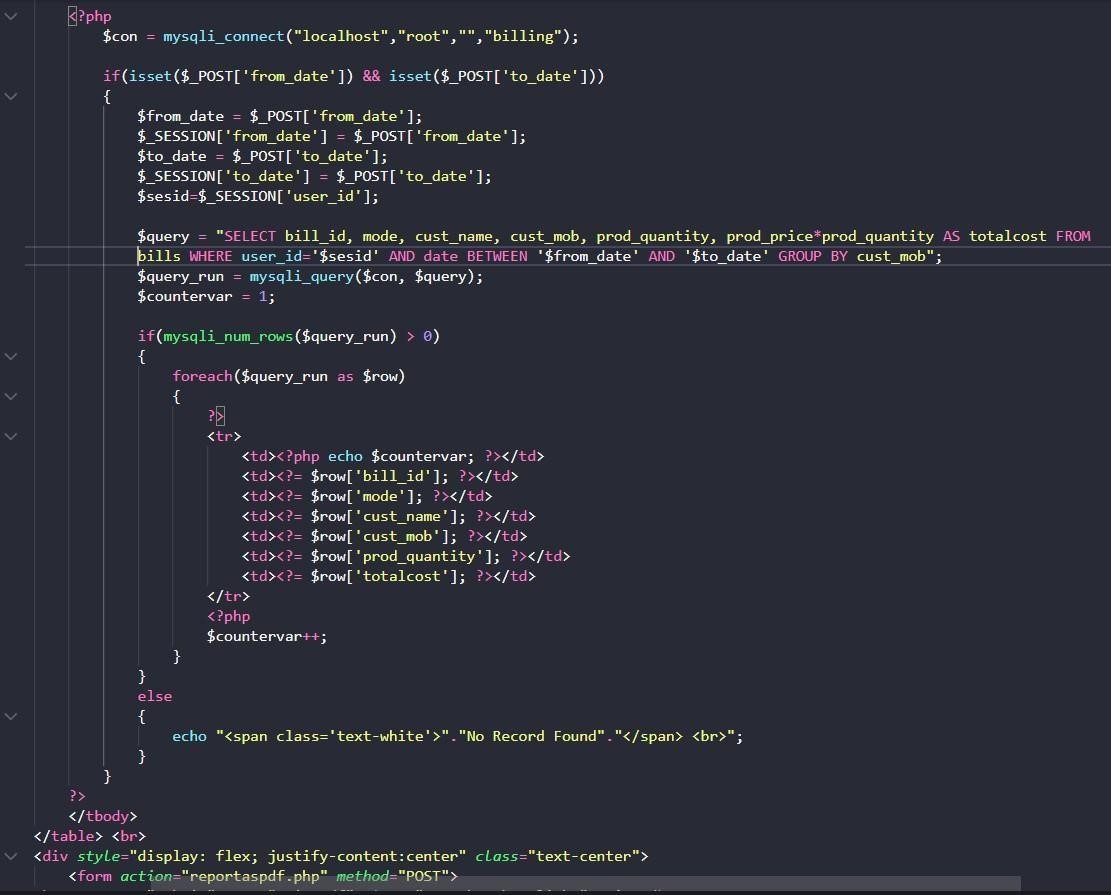
**USER OPERATIONS**



**CREATE BILL**

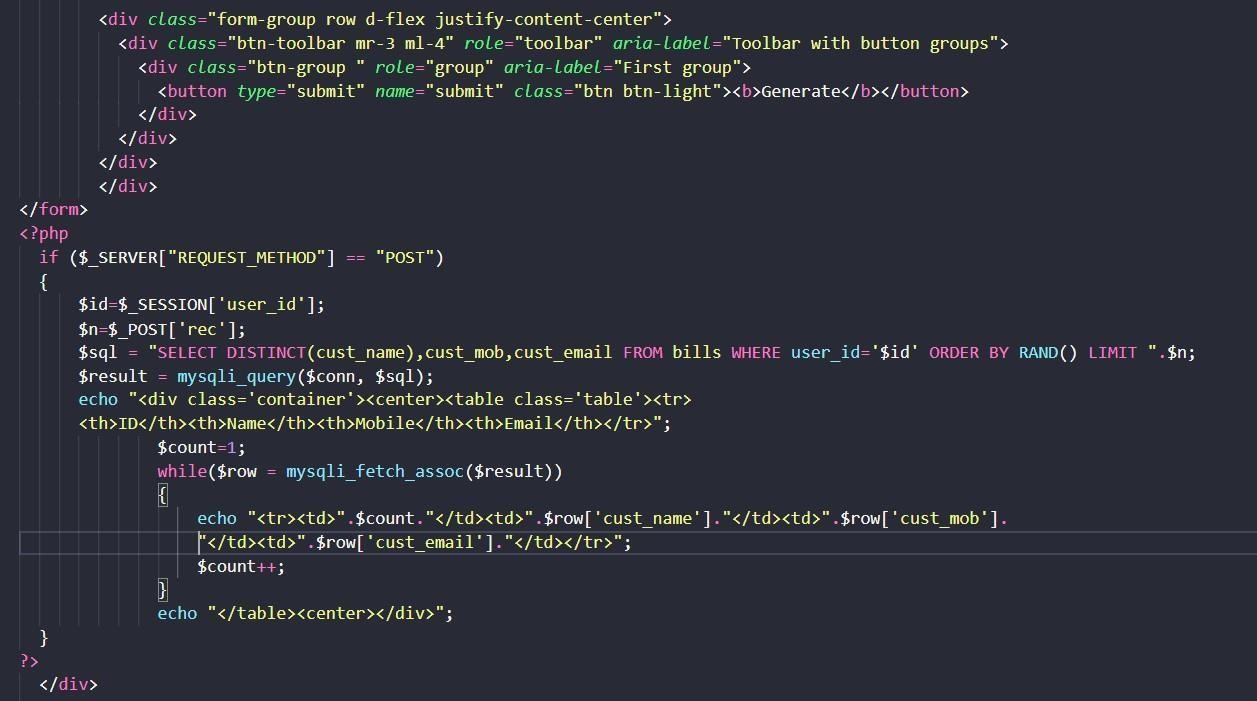


**GENERATE REPORT**



**GENERATE GIVEAWAY**

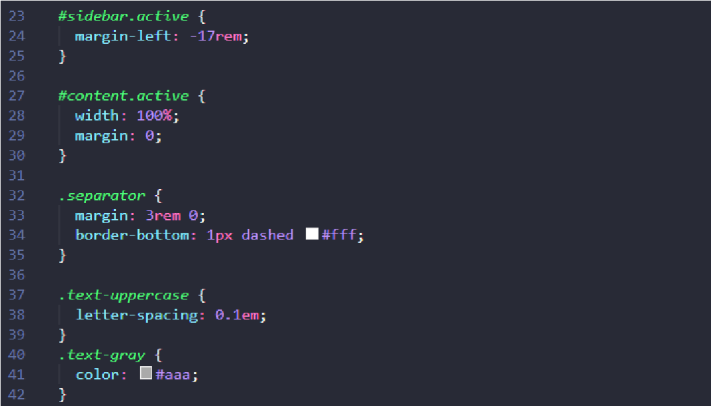
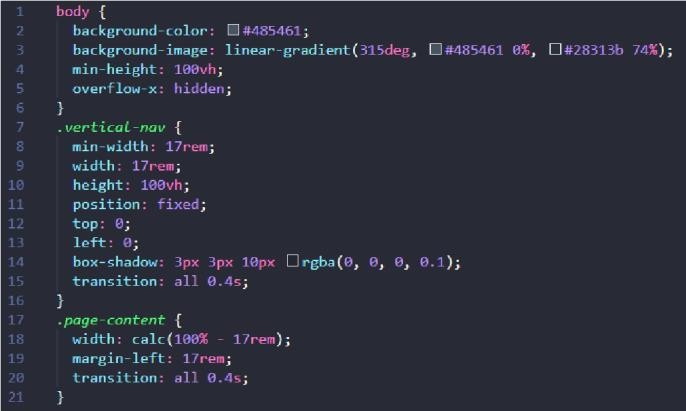
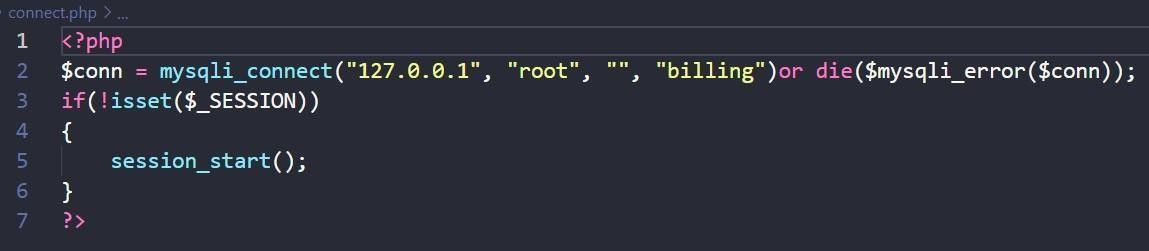
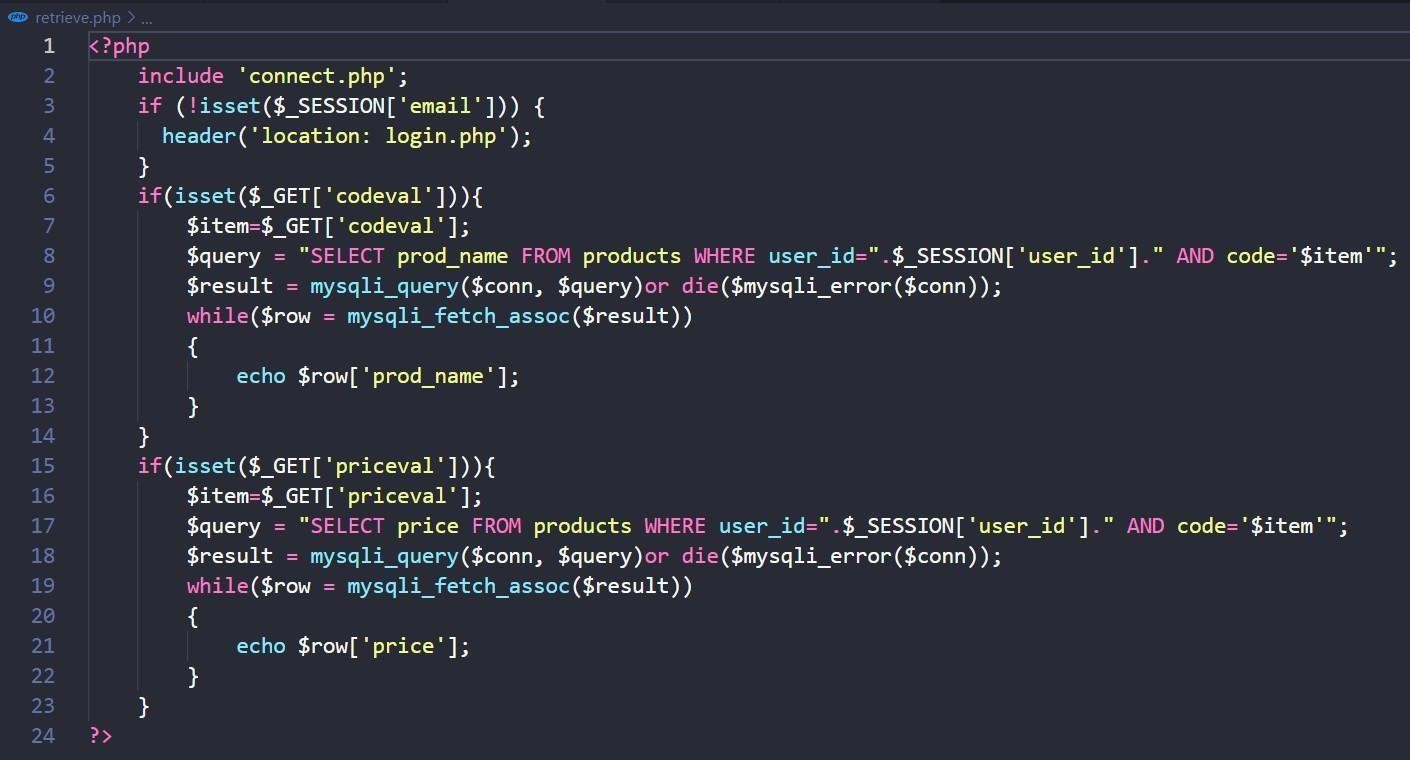
**PRODUCT SHORTHAND**



**RETRIEVE BILLS**

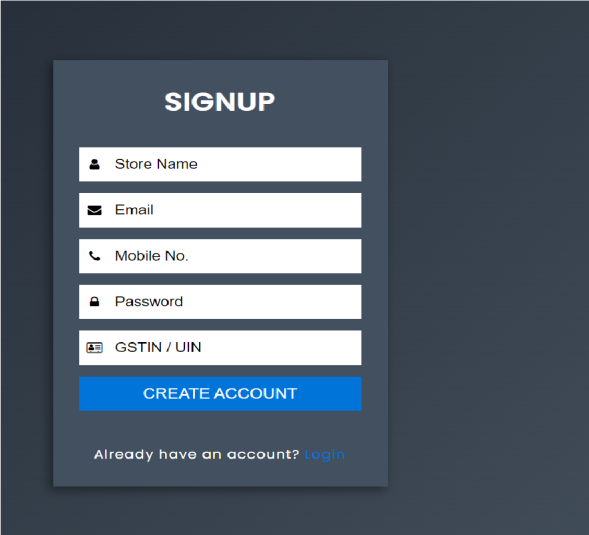
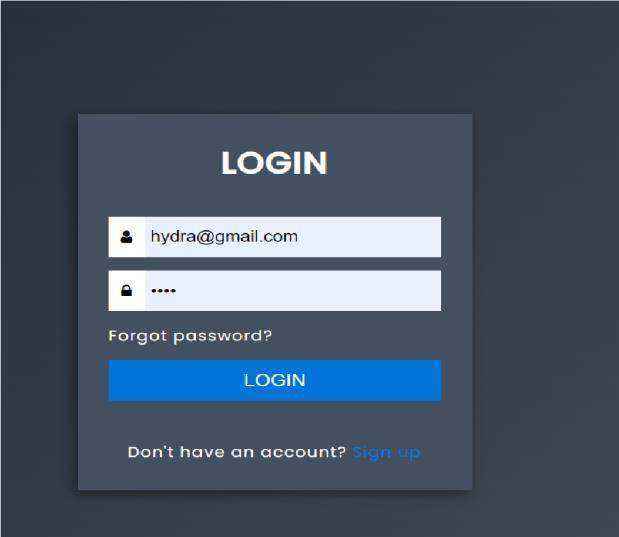
**CONNECTION PHP CODE**

**REQUIRED CSS**



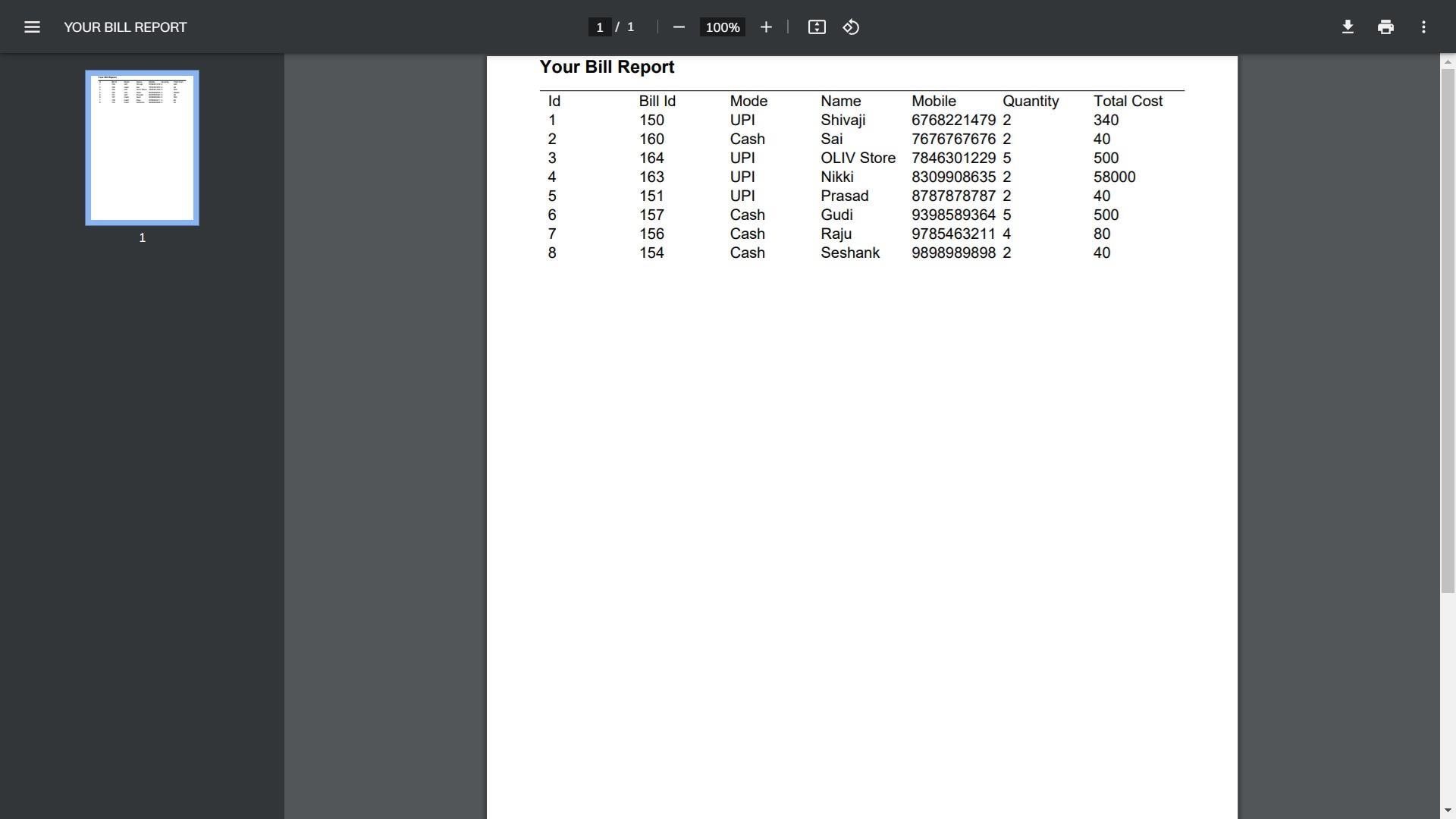
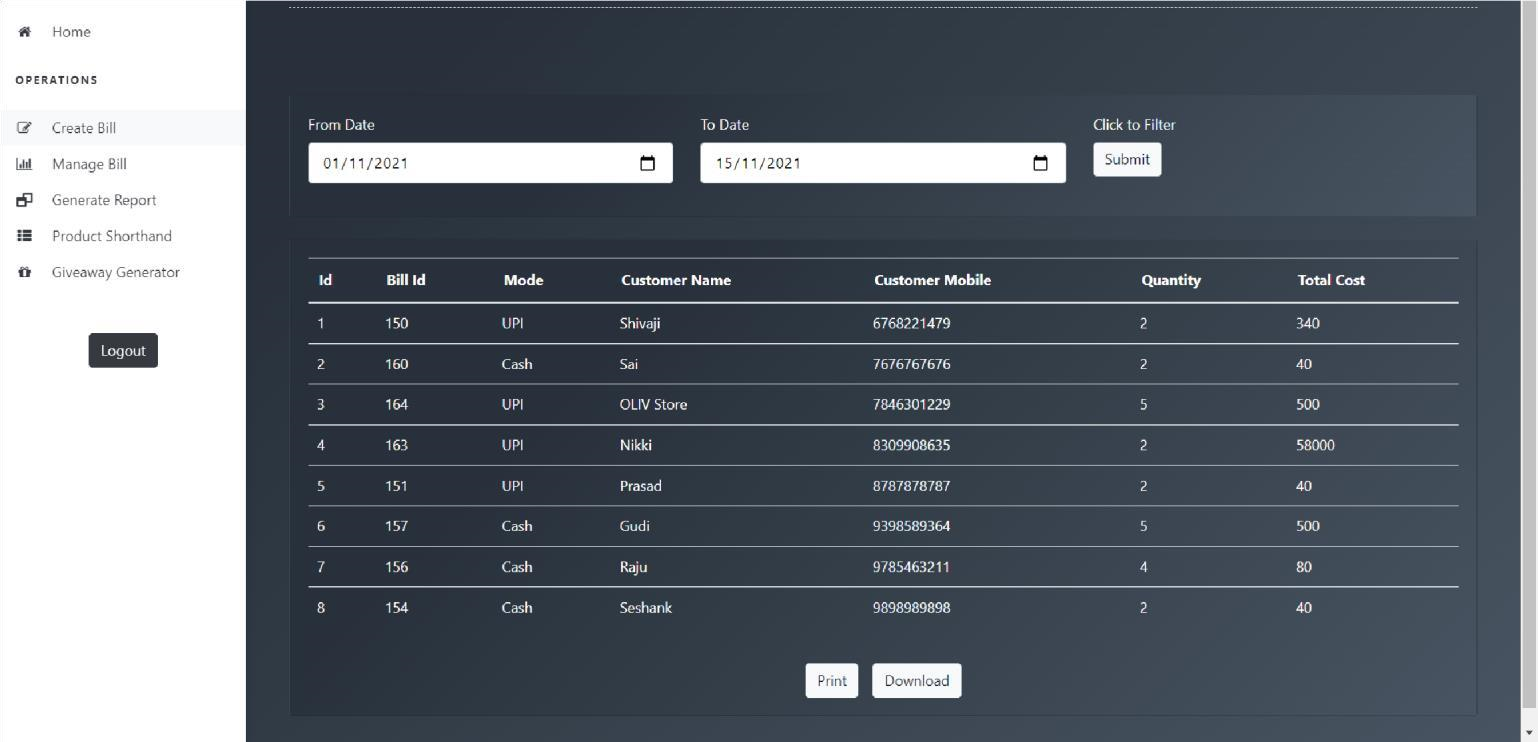
**OUTPUT SNAPSHOTS**

**CREATE BILL PAGE**



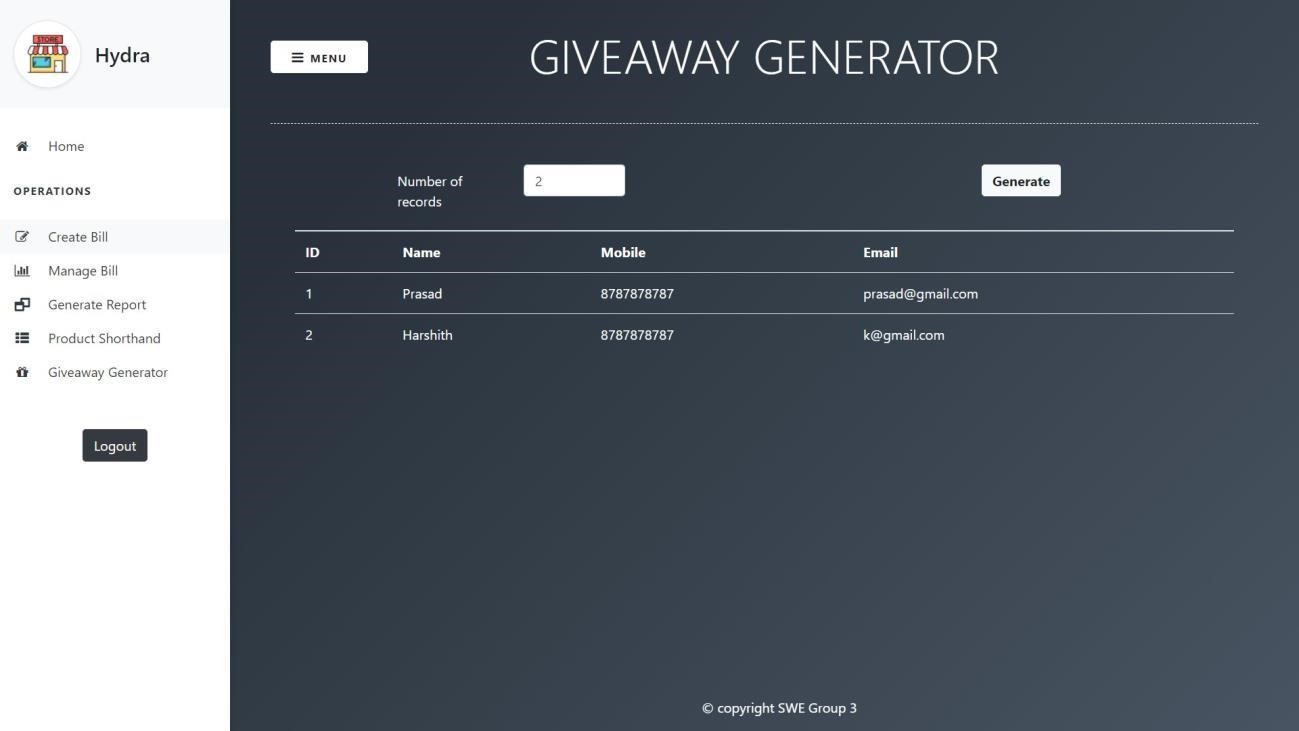
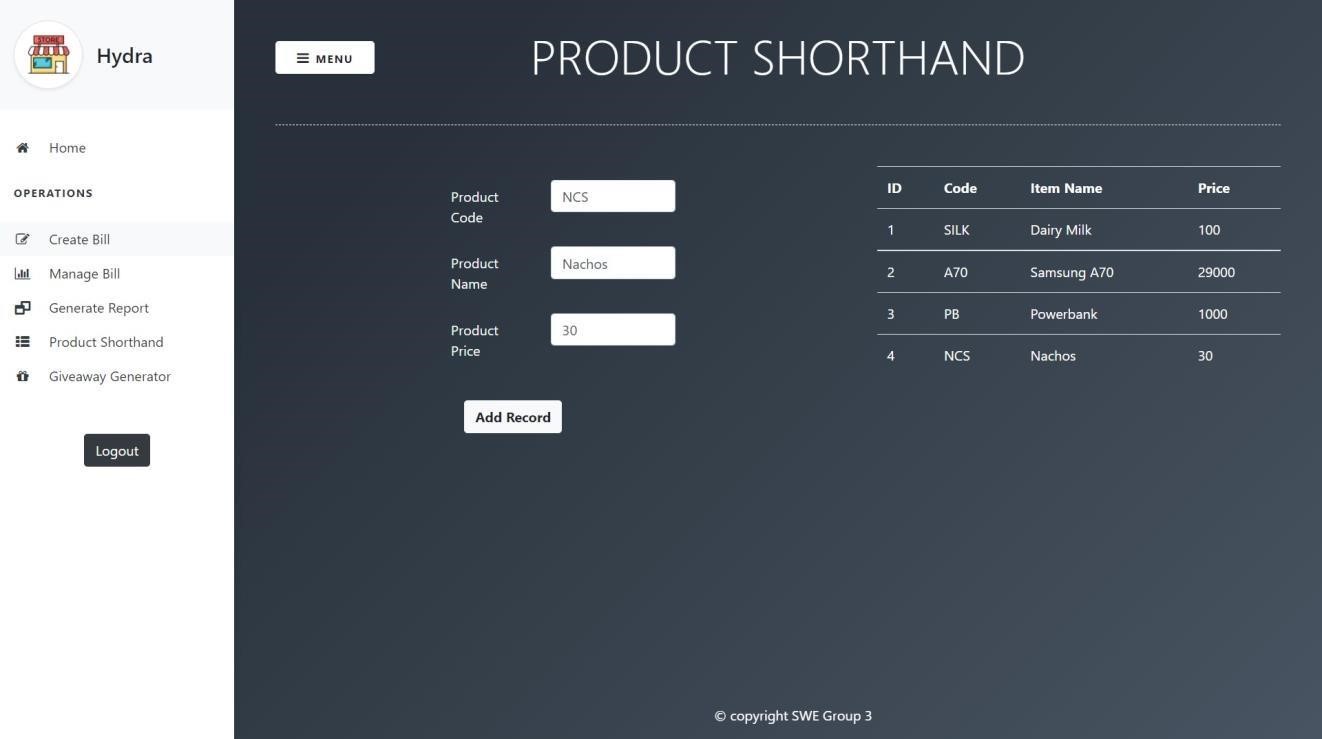
**REPORT GENERATION PAGE**

**PDF REPORT GENERATED**



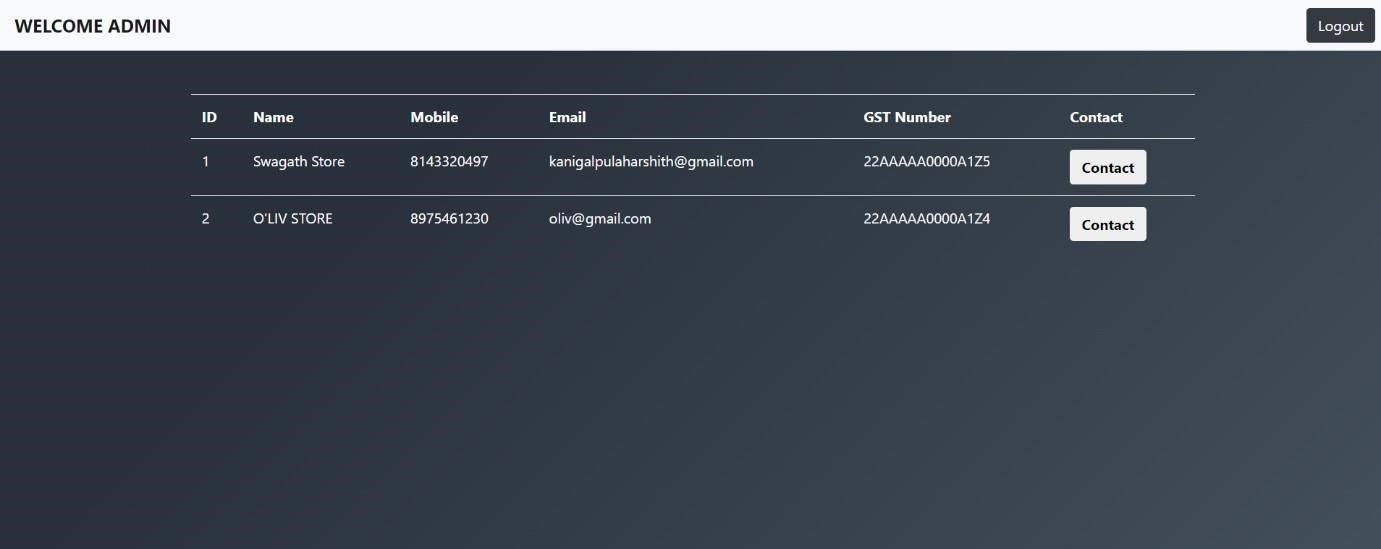
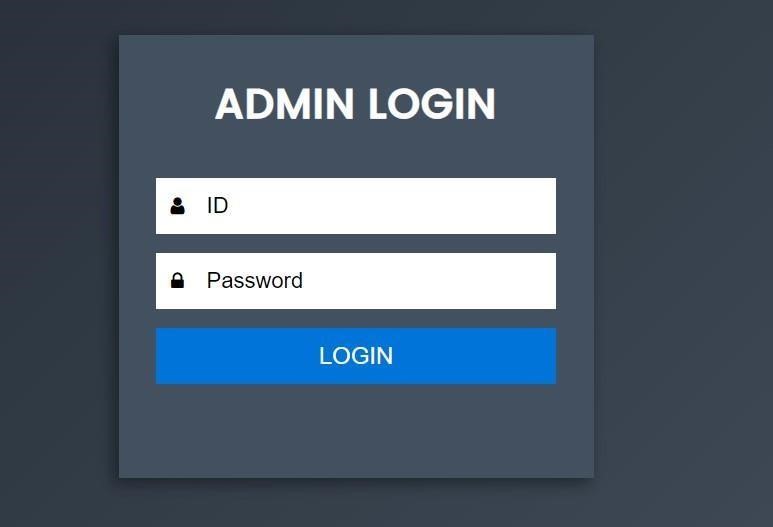
**PRODUCT SHORTHAND PAGE**

**GIVEAWAY PAGE**



**ADMIN LOGIN PAGE**

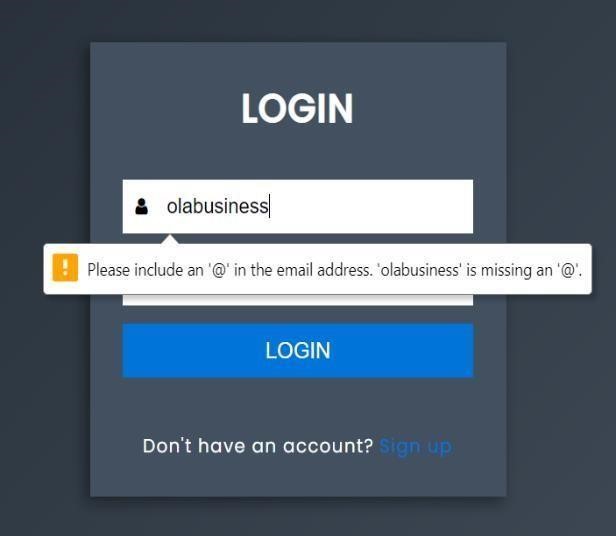
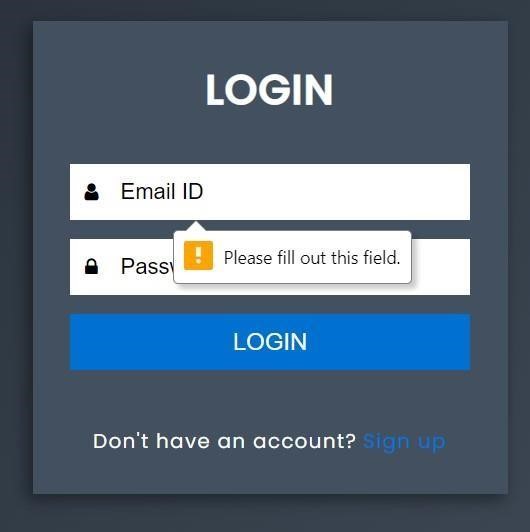
**ADMIN PAGE**



**TESTING**

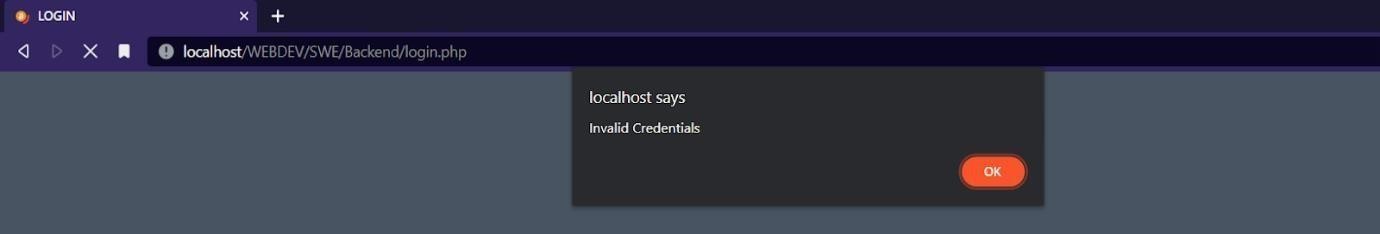
Software testing is the process of comparing software to user requirements and system specifications. Testing occurs at the phase level of the software development life cycle or at the module level of the programme code. Validation and verification are components of software testing.

1.Testing Login Page, The user credentials are properly validated. Expected output is matched with actual output.



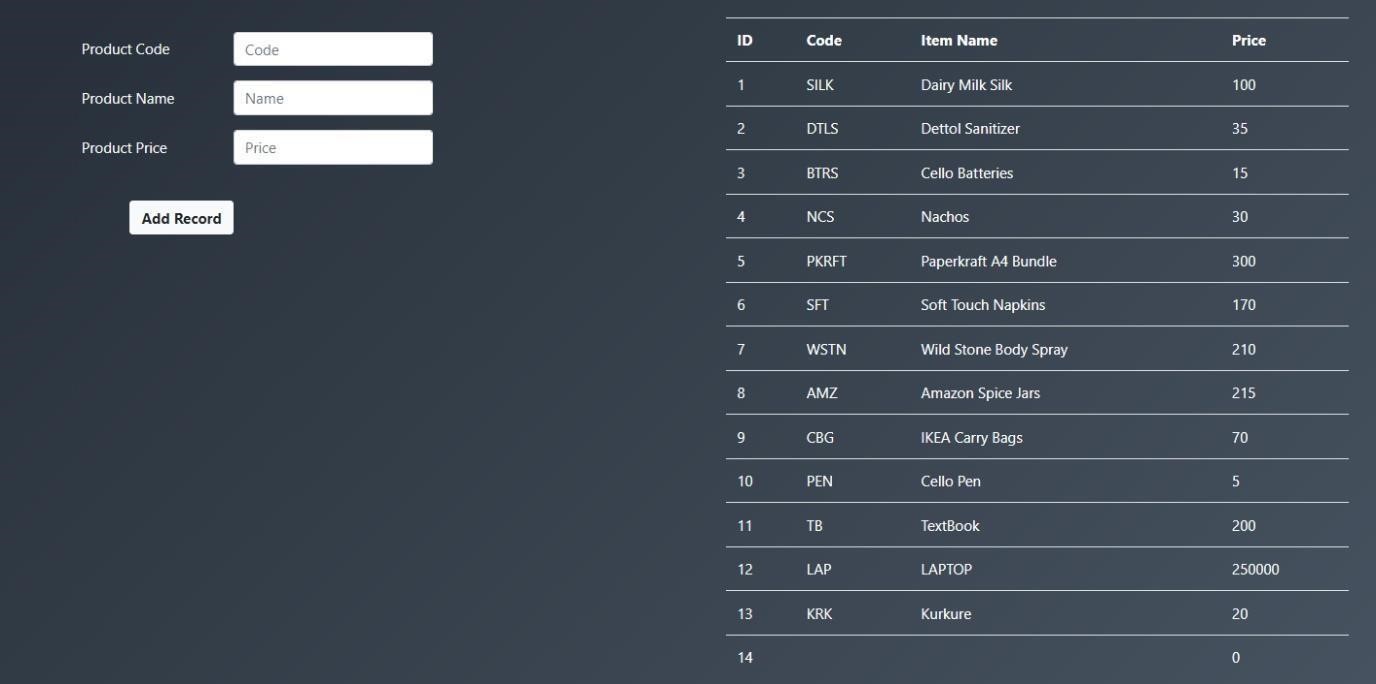
**Testing Login Page**

2.When a user credentials and actual credentials don't match, handling the error and displaying with proper error message.



**Invalid Credentials**

3. When empty product details are added to the product list, it is not being handled properly. Expected output is not matching with actual output.

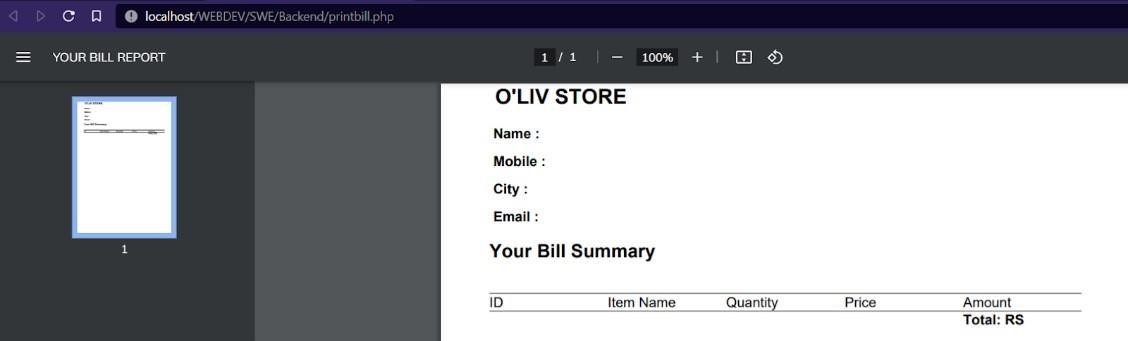


**Empty Product Details**

4.Billing for an empty product details. It is not being handled properly. Expected output is not matching with actual output.

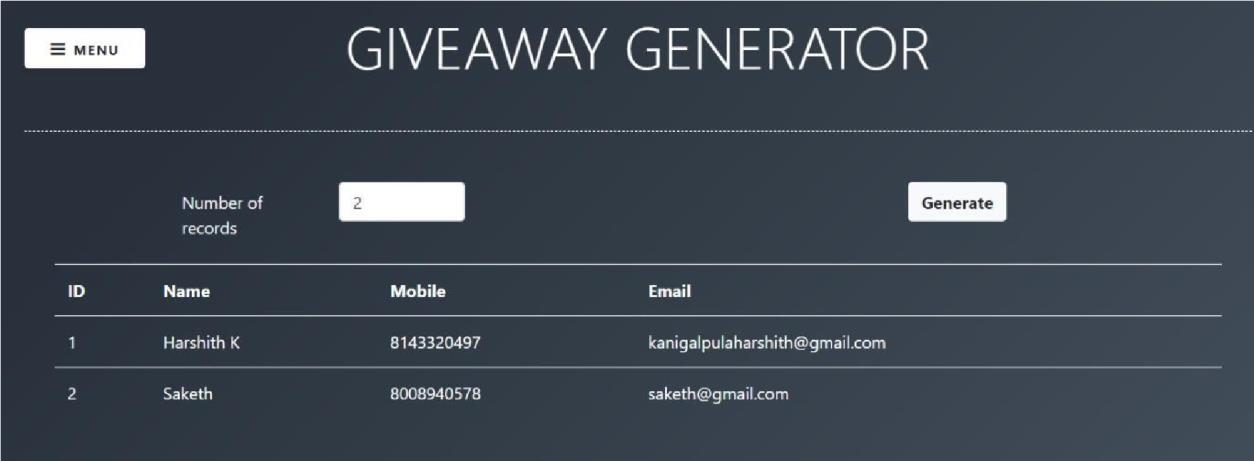


**Empty Product Details**



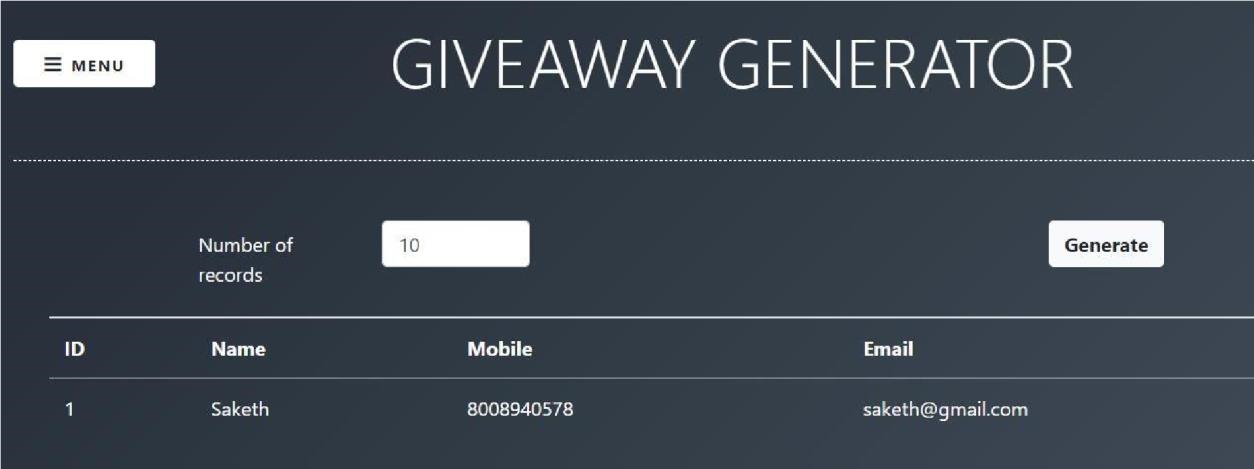
**Empty Bill**

5.The list of random user records that is generated for no. of records is accurate. Expected output matches with actual output.



**Giveaway generator**

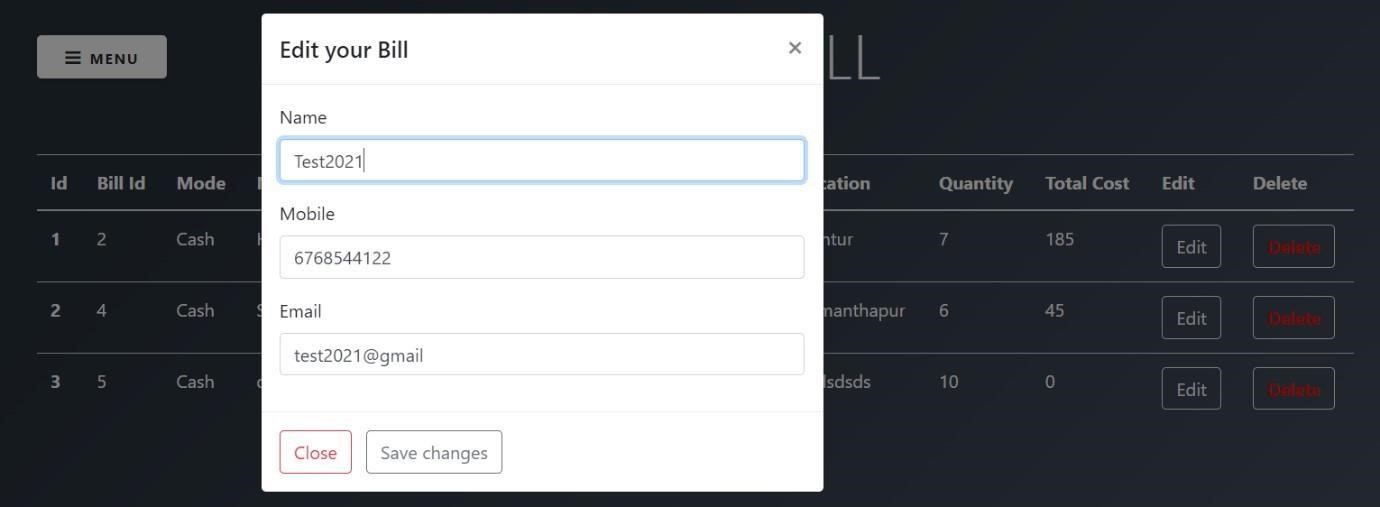
6.The list of random user records that is generated for no. of records entered is not validated. Expected output is not matching with actual output.



**Giveaway generator**

7.A working modal (popup) for Manage bills - Edit button.

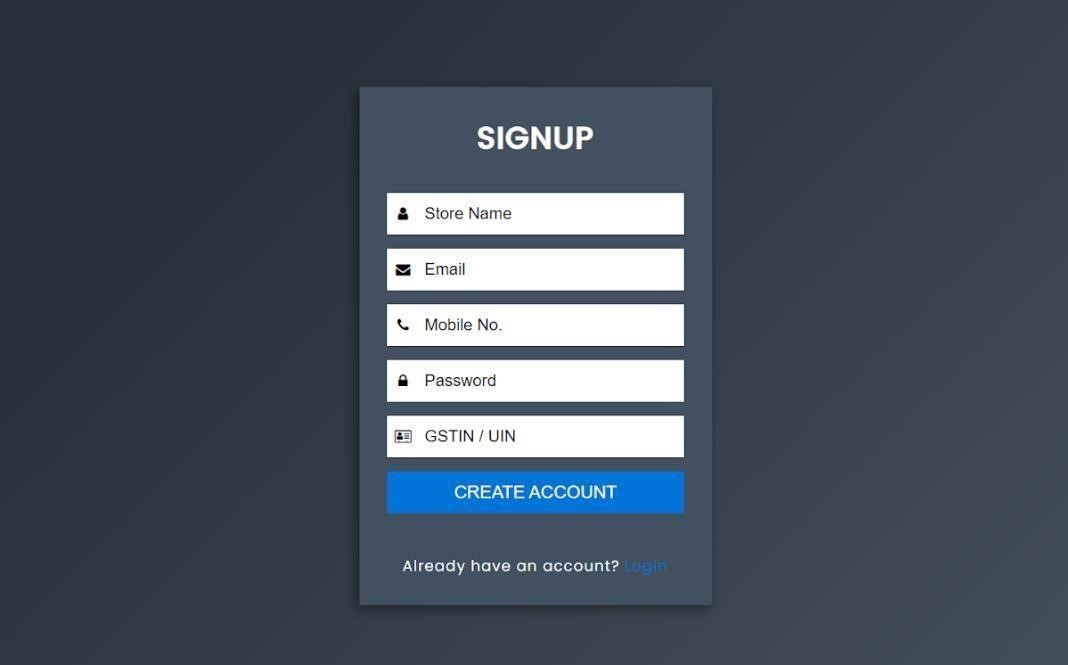
Expected Output matches with actual output.



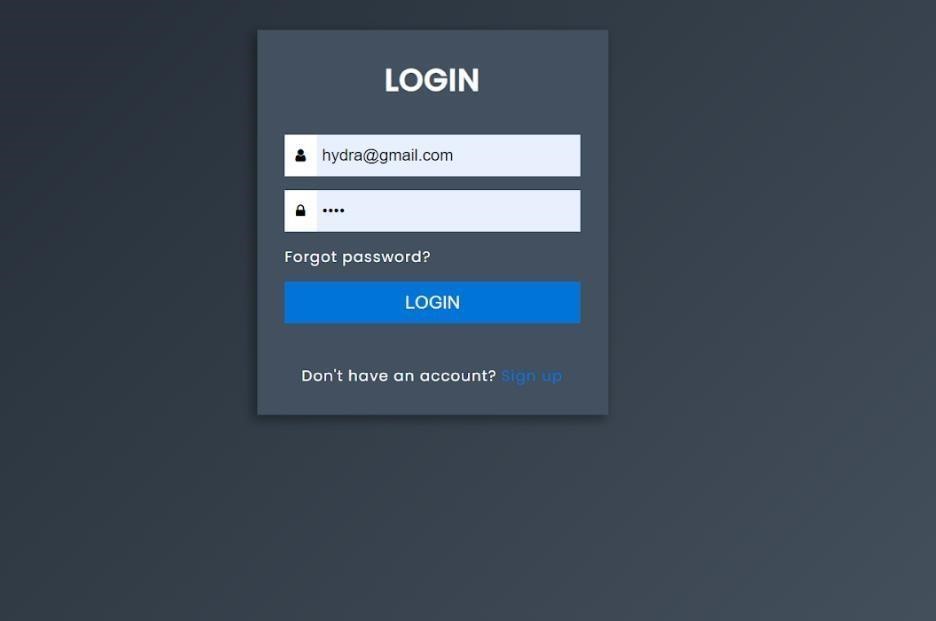
**Giveaway generator**

**PROJECT DEMO:**

**Screen shots**



# Signup Page



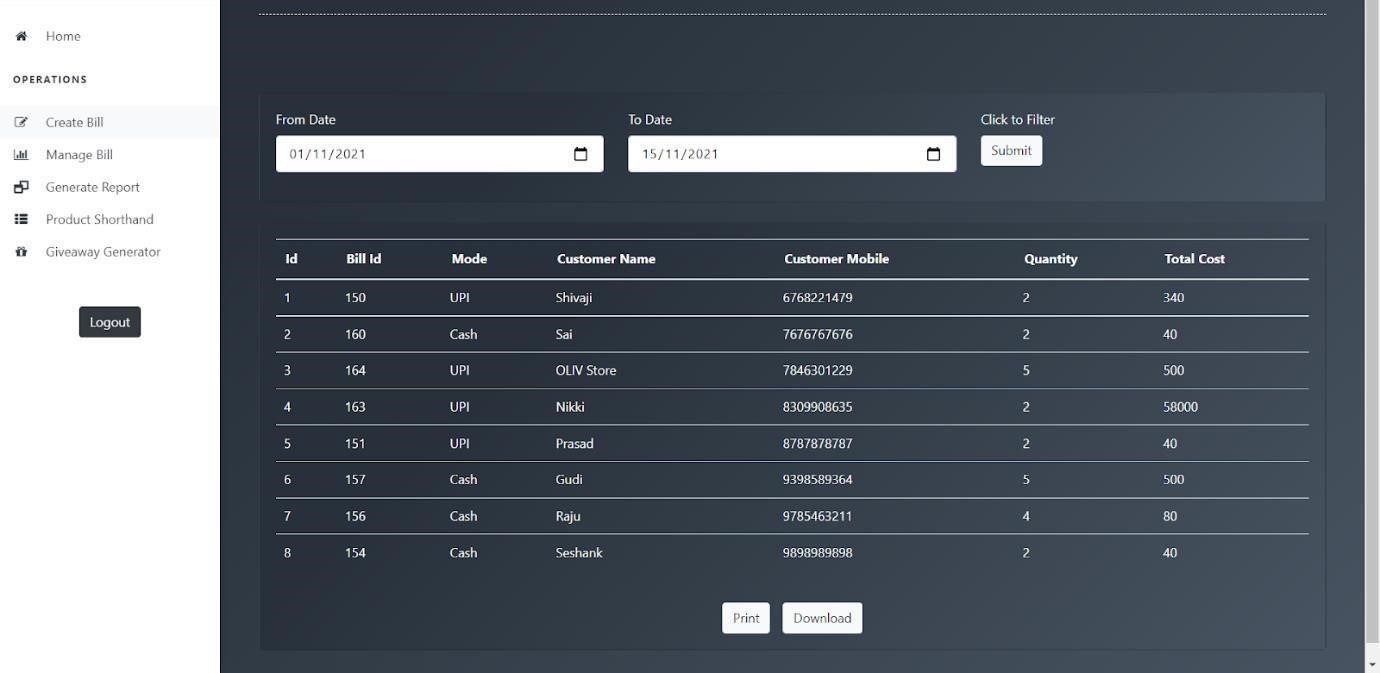
# Login Page



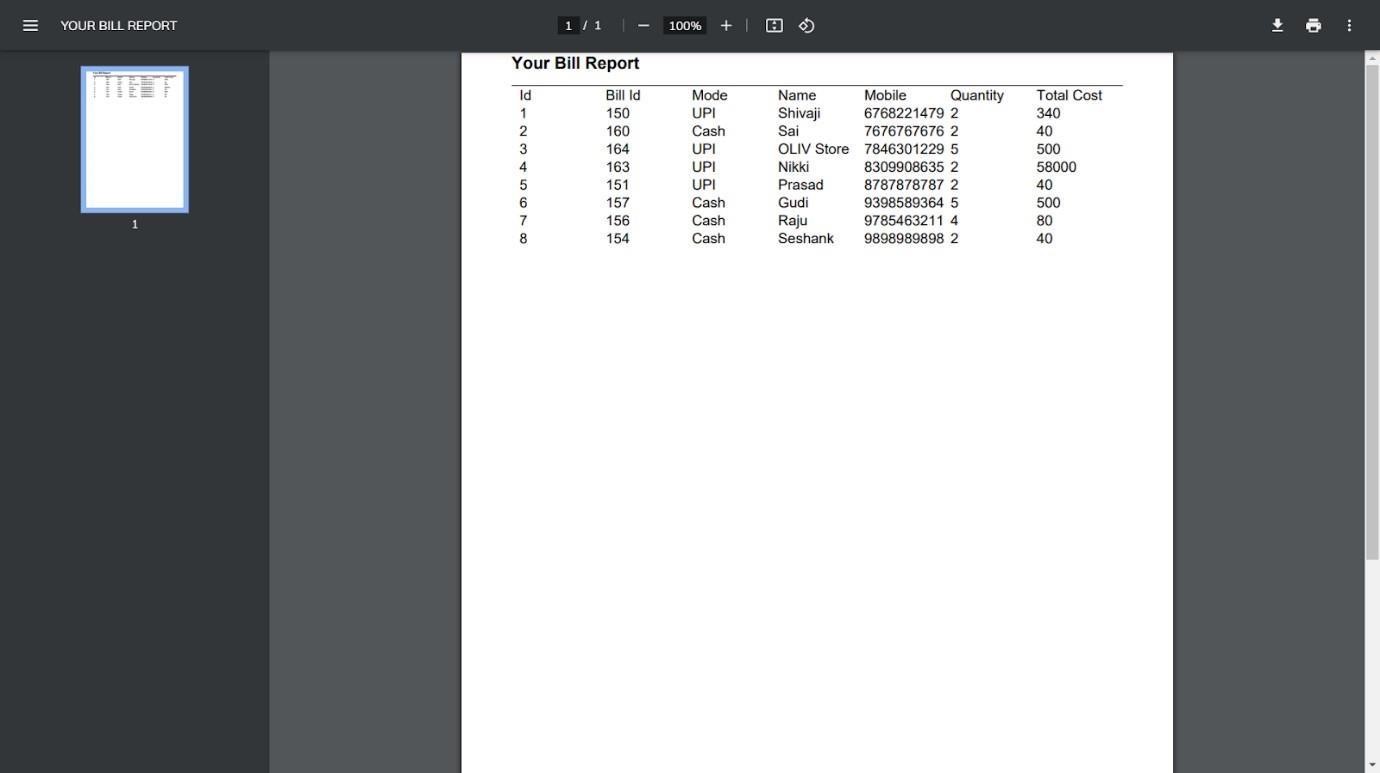
**Home Page**



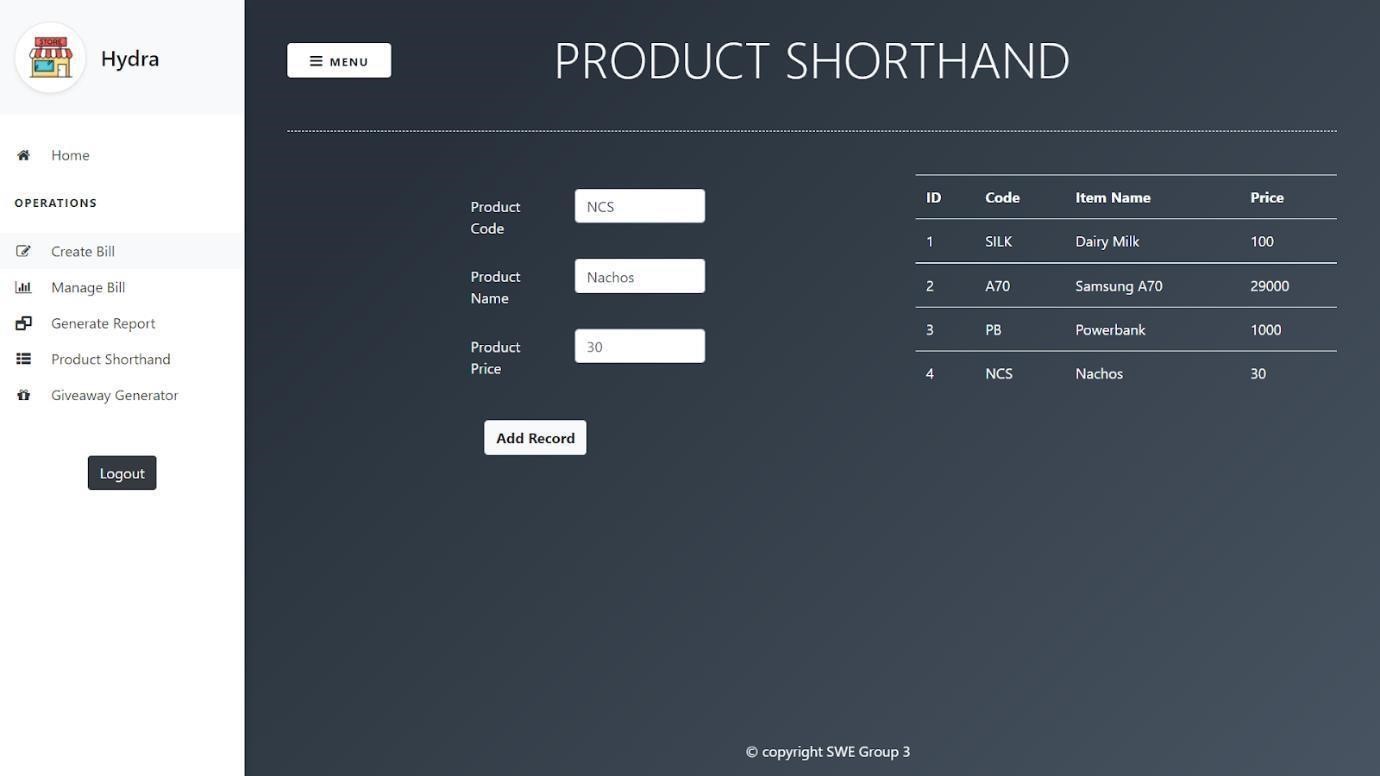
**Create Bill Page**



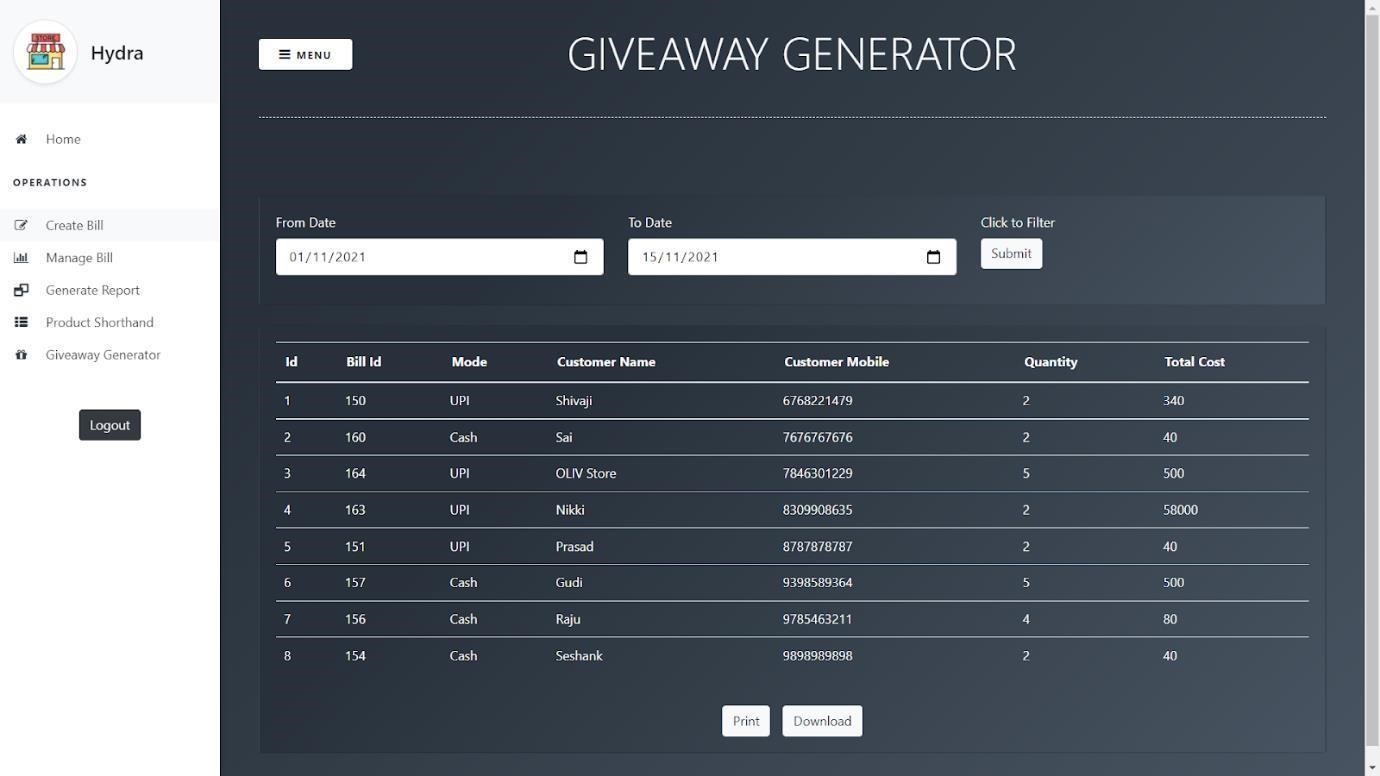
**Manage Bill Page**



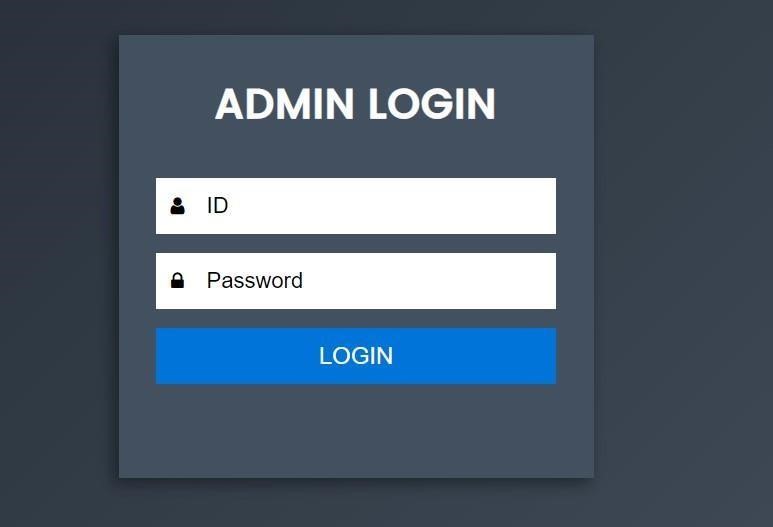
**Report Generation Page for Print**



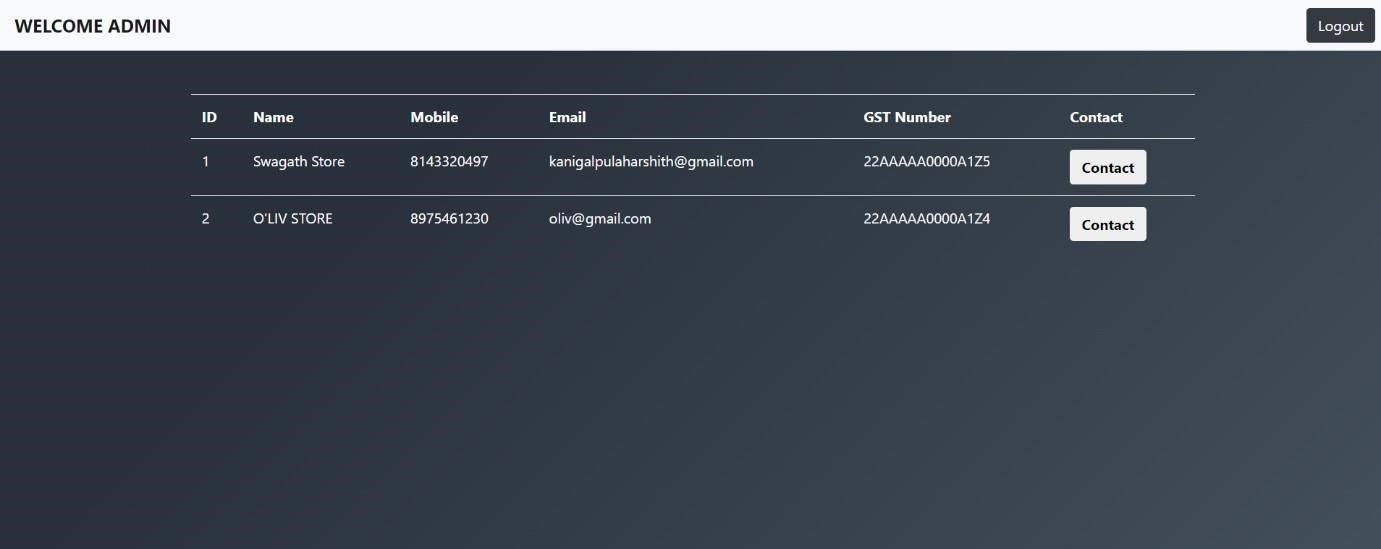
**Product Short Hand Page**



**Giveaway Generator Page**



**Admin Login Page**



**Admin Page**