

CHAPTER 1

PREAMBLE

1.1 Introduction

Computer Peripheral Management System allows the salesman to buy the computer peripherals from the manufacturers in real-time, via online service. It is a form of electronic commerce. This system evokes a physical analogy of buying products or services at a bricks-and-mortar retailer or in a shopping centre. The process is called Business-to-Business (B2B) online shopping.

This system allows the salesman to purchase the items after he/she is logged in. An unregistered salesman needs to register themselves first before shopping at this website, by providing certain information such as the name of the customer, age, email ID, and other credentials such as username and password. This process is done by the admin. Once the salesman provides all this information, the salesman gets registered and can start buying and selling goods.

The salesman can view and check for the availability of the item or goods which is of his/her requirement. Our website provides the facility of giving the salesman an access to view the item availability without he/she logging in, but should login into the website if the user needs to purchase the same item. Another special feature of this website would be that it provides the details about the item or the product which is yet to be launched. It provides information to the salesman about the products which would be available in the market shortly.

This project covers the online marketing of all the computer peripherals of different companies. The project shows the product details and specifications and thus the product can be purchased by the salesman.

1.1.1 Database Management System (DBMS)

A Database Management System (DBMS) is system software for creating and managing databases. The DBMS provides users and programmers with a systematic way to create, retrieve, update and manage data.

Earlier, punched cards technology was used to store the data – later, files. But files have a lot of limitations.

[1968-1980] Era of hierarchical Database: Prominent hierarchical database model was IBM'S first DBMS called Information Management System.

Network Data Model: In Early 1960s, Charles Bachmann developed first DBMS at Honeywell, Integrated Data store.

It standardized in 1971 by the CODASYL group.

Instance- a table with rows or columns.

Schema-specifies the structure (name of relation, name and type of each column).

1970: Ted Codd at IBM's San Jose Lab proposed relational models. Two major projects start and both were operational in late 1970s.

INGRES at University of California, Berkeley became commercial and followed up POSTGRES which was incorporated into Informix.

System R at IBM San Jose Lab, later evolved into DB2, which became one of the first DBMS product based on the relational model. (Oracle produced a similar product just prior to DB2.)

1976: Peter Chen defined the Entity – Relationship (ER) model.

1980: Maturation of the relational database technology, more relational based DBMS were developed and SQL standard adopted by ISO and ANSI.

1985: Object-Oriented DBMS (OODBMS) develops.

1990: Incorporation of object – orientation in relational DBMSs, new application areas, such as data warehousing and OLAP, web and internet, interest in text and multimedia.

1991: Microsoft shift access, a personal DBMS created as element of Windows gradually supplanted all other personal DBMS products.

1995: First Internet database applications.

1997: XML applied to database processing, which solves long – standing database problems. Major vendors begin to integrate XML into DBMS products.

Relational DBMS (RDBMS): is the basis for SQL, and for all modern database systems like management system (DBMS) that is based on the relational model as introduced by E.F. Codd.

1.1.2 MySQL

MYSQL is an open source RDBMS. Its name is a combination of “My”, the name of co-founder Micheal widenius’s daughter, and “SQL”, the abbreviation for Structured Query Language. MySQL is written in C and C++.

Many programming languages with language – specific APIs include libraries for accessing MySQL databases. These include MySQL Connector/Net for integration with Microsoft’s visual Studio (languages such as C# and VB are most commonly used) and the JDBC driver for Java. In addition, an ODBC interface called MySQL Connector/ODBC allows additional programming languages that support the ODBC interface to communicate with a MySQL database, such as ASP or ColdFusion. The HTSQL – URL-based query method also ships with a MySQL database and web client via structured URLs.

1.1.3 XAMPP

- X-Cross Platform
- A-Apache
- M-MariaDB
- P-PHP
- P-Perl

XAMPP- is a free and open-source cross-platform web server solution stack package developed by Apache Friends. It consists of HTTP server, MariaDB, MySQL, PHP, PhpMyAdmin, etc., XAMPP stands for “Linux, XAMPP Apache + MariaDB (OR

MySQL)+PHP+Perl” as it is popularly installed by the Linux OS users. XAMPP is very popular due to its easy installation which requires .exe or zip files. It provides a wonderful platform for MariaDB, MySQL, and SQLite. Another added advantage of the XAMPP is that once it is installed, we can treat a local host as a remote host by connecting using FTP client. The Windows version of XAMPP contains a .exe file of size 123MB. The services provide by the XAMPP can be easily controlled through the XAMPP Control Panel.

1.1.4 PHP

PHP (Hypertext pre-processor) which is widely used open-source general purpose scripting language used for web development which can be embedded in HTML code. It is a server-side scripting language. It was originally created by Rasmus Lerdorf in 1994. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which maybe any type of data including images, with the generated web page.

1.1.5 Normalization

Normalization is a database design technique which organizes tables in a manner that reduces redundancy and dependency of data. It divides larger tables to smaller tables and links them using relationships. The four basic types of normalization are:

- First Normal Form (1NF)
- Second Normal Form (2NF)
- Third Normal Form (3NF)
- Boyce Codd Normal Form (BCNF)

The rules for each normal form are as follows:

- First Normal Form (1NF): The relation schema cannot have composite attributes, multivalued attributes or nested tuples. It can have only simple, atomic values.
- Second Normal Form (2NF): A relation schema R is in second normal form if every non-prime attribute A in R is fully functionally dependent on the primary key of R.

- Third Normal Form (3NF): A relation schema R is in third normal form if whenever a functional dependency $X \rightarrow A$ holds in R, either:
 - a) X is a super key of R or
 - b) A is a primary attribute of R.
- Boyce Codd Normal Form (BCNF): A relation schema R is in BCNF is whenever a functional dependency $X \rightarrow A$ holds in R, then X is a super key of R.

1.1.6 Proposed System

This application is used for online wholesale purchase of goods. The admin maintains the record of the sales done by the salesman. This record can be accessed by the salesman also. The salesman can sit at individual terminals and login to purchase the product which they are in need of. The new salesman can register themselves in the website via the admin and purchase the products available. Any salesman irrespective if they are registered or not can view the products available in the website which are for sale. The main agenda of this system is to provide the salesman the product, purchase them from the wholesale market and sell them to the customer. This record is maintained by the admin in this project. The admin privileges such as addition of salesman, shop details, and its branches are also provided. This project also aims in broadcasting the different materials lists which are yet to be launched in the market.

1.1.7 Existing System

Existing system is a manual one in which customers would have to go to the shop and purchase the product which they in need of. It is more time consuming and energy consuming. Hence this system would be more effective, efficient and very useful for the customers.

1.2 Objectives

The objective of the Computer peripheral system is to provide better service for the customers which improves time efficiency and to reduce energy consumption.

1. Computer peripheral system will reduce hectic job of contacting the salesman and dealing with them.

2. Being an integrated Computer peripheral system it will reduce paper work.
3. To allow the product owner to showcase their products.
4. To allow the salesman to register themselves via admin and sell the products to the customers.

1.3 Organisation of the report

- Chapter 2 discusses the hardware and software requirements to run the stated applications.
- Chapter 3 gives an idea about the project and its actual implementation.
- Chapter 4 discusses about testing and limitations of program.
- Chapter 5 concludes by giving a direction for future enhancement.

1.4 Summary

The chapter discussed above gives a clear overview about DBMS, its history and evolution; MYSQL RDBMS; Normalization concepts. The scope and goals of project have been clearly stated under Objectives. Organization of report section enables easy access to and gives a gist about all chapters in this report.

The upcoming chapters will depict how, using queries, we can populate the database, manipulate it and delete it, etc., according to the requirements.

CHAPTER 2

REQUIREMENT SPECIFICATIONS

2.1 Software Specifications

- Operating System: Windows 7/8/10
- Front End: HTML, CSS, JavaScript
 1. HTML: HTML is used to create and save web document. E.g. Notepad/Notepad++
 2. CSS: (Cascading Style Sheets): Create attractive Layout
 3. UIKIT: A lightweight and modular front-end framework for developing fast and powerful web interfaces.
- Back End: MYSQL, PHP
 1. PHP: Hypertext Pre-processor (PHP) is a technology that allows software developers to create dynamically generated web pages, in HTML, XML, or other document types, as per client request. PHP is open source software.
 2. MySQL: MySQL is a database, widely used for accessing querying, updating, and managing data in databases.
- Server: XAMPP

2.2 Hardware Specifications

- Processor: x86 compatible processor with 1.7 GHz clock speed
- RAM: 512 MB or greater
- Hard Disk: 20 GB or greater
- Monitor: VGA/SVGA
- Keyboard: 104 keys standard
- Mouse: 2/3 button. Optical/Mechanical

2.3 User Characteristics

- User must be comfortable with the basic working of a computer and must have basic English Knowledge.

CHAPTER 3

SYSTEM DESIGN AND IMPLEMENTATION

3.1 Introduction

Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy requirements. System design could be seen as the application of systems theory to product development.

3.2 E-R Diagram

An entity relationship diagram (ERD) shows the relationship of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties.

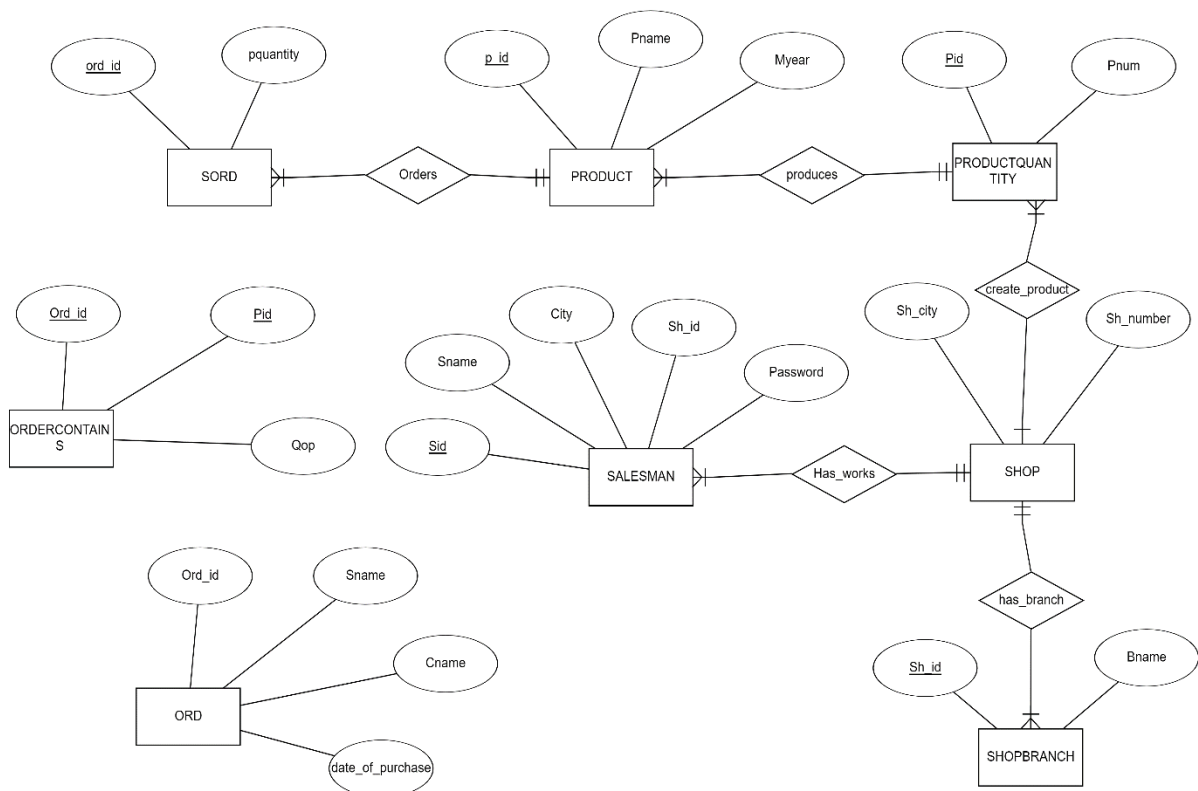


Figure 3.1: ER diagram

3.3 Schema Diagram

In database items, a schema is the organization and structure of a database.

A schema contains schema objects, which could be tables, columns, data types, views, stored procedures, relationships, primary keys, foreign keys, etc.

A database schema can be represented in a visual diagram, which shows the database objects and their relationship with each other.

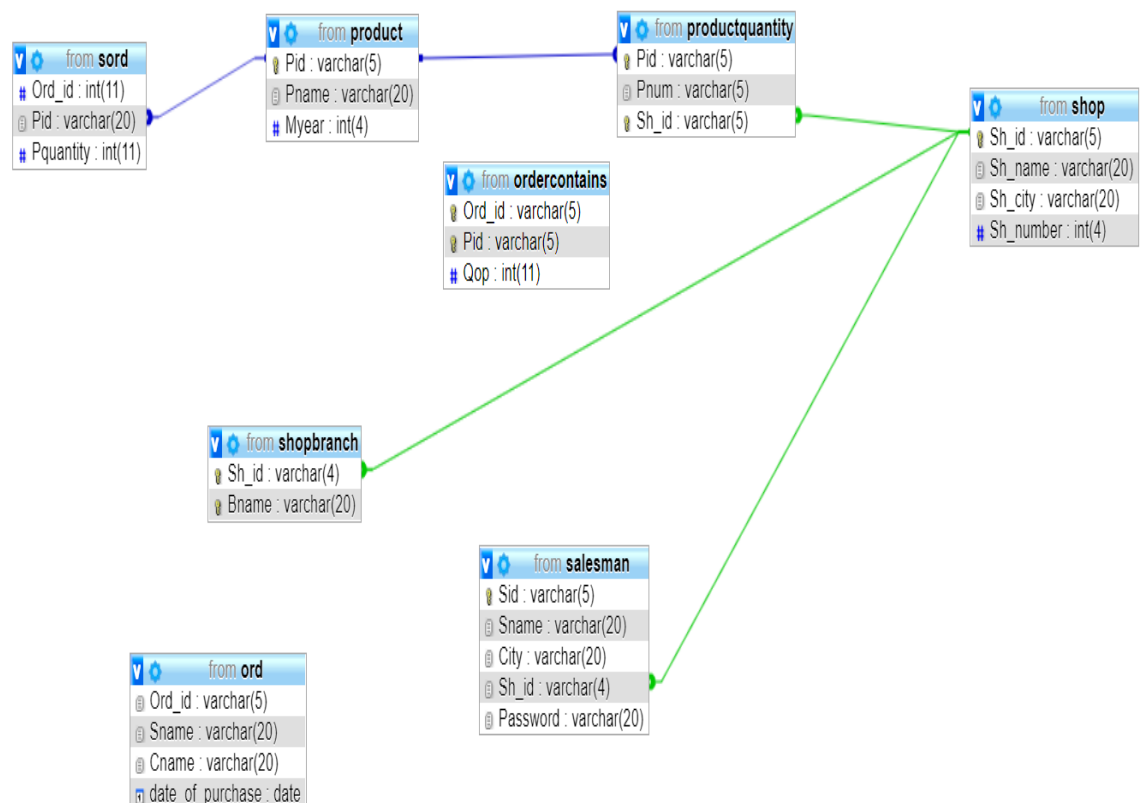


Figure 3.2: Schema diagram

The above schema represents the computer peripheral system database and also includes the primary keys and foreign keys.

3.4 Use case Diagram

A use case diagram is a dynamic or behaviour diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform.

Use case diagrams are valuable for visualising the functional requirements of a system that will translate into design choices and development priorities.

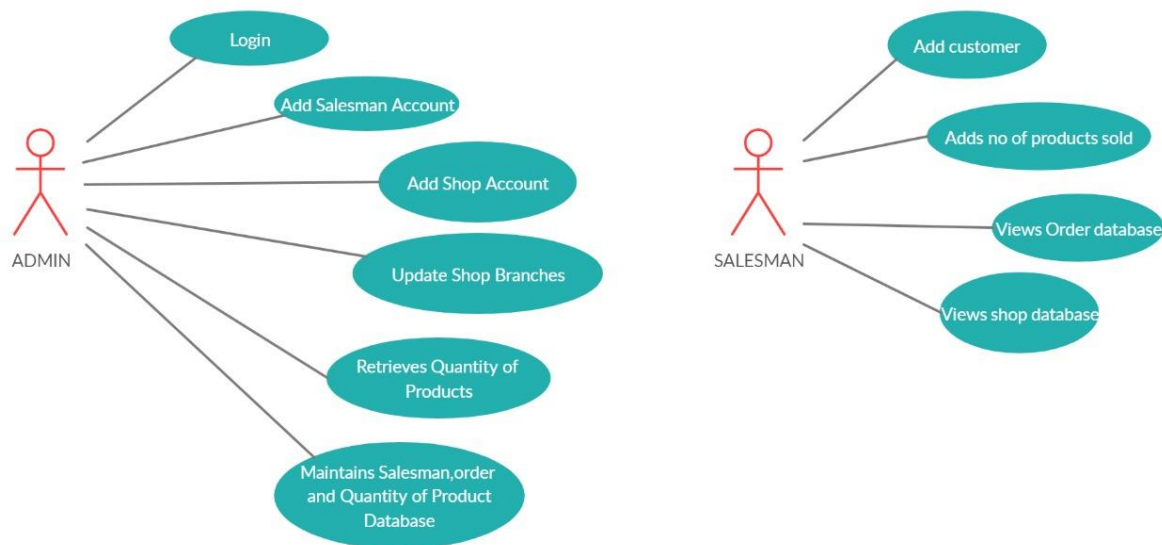


Fig 3.3 Use case Diagram

3.5 Queries

This section specifies all the SQL queries used to perform various tasks in MySQL such as insert, delete and update.

- a. **CREATE COMMANDS:** This command is used to create the tables in the database.

```
CREATE TABLE `ordercontains` (  
  `Ord_id` varchar(5) NOT NULL,  
  `Pid` varchar(5) NOT NULL,  
  `Qop` int(11) NOT NULL  
  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `product` (  
  `Pid` varchar(5) NOT NULL,  
  `Pname` varchar(20) NOT NULL,  
  `Myear` int(4) NOT NULL  
  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `productquantity` (  
  `Pid` varchar(5) NOT NULL,  
  `Pnum` varchar(5) NOT NULL,  
  `Sh_id` varchar(5) NOT NULL  
  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `salesman` (  
  `Sid` varchar(5) NOT NULL,  
  `Sname` varchar(20) NOT NULL,  
  `City` varchar(20) NOT NULL,  
  `Sh_id` varchar(4) NOT NULL,  
  `Password` varchar(20) DEFAULT NULL  
  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `shop` (  
  `Sh_id` varchar(5) NOT NULL,  
  `Sh_name` varchar(20) NOT NULL,  
  `Sh_city` varchar(20) NOT NULL,  
  `Sh_number` int(4) NOT NULL  
  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `shopbranch` (  
  `Sh_id` varchar(4) NOT NULL,  
  `Bname` varchar(20) NOT NULL  
  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `sord` (  
  `Ord_id` int(11) NOT NULL,  
  `Pid` varchar(20) NOT NULL,  
  `Pquantity` int(11) NOT NULL  
  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `userpass` (  
  `user` varchar(20) NOT NULL,  
  `pass` varchar(20) NOT NULL  
  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

b.ALTER COMMANDS: This command is used to alter the already created table. It alters the existing table such as addition of foreign key, addition and deletion of columns etc.,

```
Indexes for table `ordercontains`  
ALTER TABLE `ordercontains`  
ADD PRIMARY KEY (`Ord_id`,`Pid`);
```

```
Indexes for table `product`  
ALTER TABLE `product`  
ADD PRIMARY KEY (`Pid`);
```

```
Indexes for table `productquantity`  
  
ALTER TABLE `productquantity`  
ADD PRIMARY KEY (`Pid`,`Sh_id`),  
ADD KEY `e5` (`Sh_id`);
```

```
Indexes for table `salesman`  
ALTER TABLE `salesman`  
ADD PRIMARY KEY (`Sid`),  
ADD KEY `e2` (`Sh_id`);
```

Indexes for table `shop`

```
ALTER TABLE `shop`  
ADD PRIMARY KEY (`Sh_id`);
```

Indexes for table `shopbranch`

```
ALTER TABLE `shopbranch`  
ADD PRIMARY KEY (`Sh_id`,`Bname`);
```

Indexes for table `sord`

```
ALTER TABLE `sord`  
ADD KEY `d1` (`Pid`);
```

Constraints for table `productquantity`

```
ALTER TABLE `productquantity`  
ADD CONSTRAINT `e5` FOREIGN KEY (`Sh_id`) REFERENCES `shop` (`  
Sh_id`),  
ADD CONSTRAINT `e6` FOREIGN KEY (`Pid`) REFERENCES `product` (`  
Pid`);
```

Constraints for table `salesman`

```
ALTER TABLE `salesman`  
ADD CONSTRAINT `e2` FOREIGN KEY (`Sh_id`) REFERENCES `shop` (`  
`Sh_id`);
```

Constraints for table `shopbranch`

```
ALTER TABLE `shopbranch`  
ADD CONSTRAINT `e1` FOREIGN KEY (`Sh_id`) REFERENCES `shop` (`  
Sh_id`),  
ADD CONSTRAINT `e4` FOREIGN KEY (`Sh_id`) REFERENCES `shop` (`  
Sh_id`);
```

Constraints for table `sord`

```
ALTER TABLE `sord`  
ADD CONSTRAINT `d1` FOREIGN KEY (`Pid`) REFERENCES `product` (`  
Pid`);
```

- c. **INSERT COMMANDS:** This command is used to insert the already created table. It alters the existing table such as addition of foreign key, addition and deletion of columns etc.,

```
INSERT INTO `userpass` (`user`, `pass`) VALUES  
('keerthana', 'keer@123');
```

3.6 Stored Procedure

A stored procedure is a set of Structured Query Language (SQL) statements with an assigned name, which are stored in a relational database management system as a group, so it can be reused and shared by multiple programs.

PROCEDURE:

The above stored procedure is used to fetch the details from signup database.

1. DELIMITER \$\$

Procedures

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `order` () NO SQL  
SELECT * FROM ord$$
```

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `products` () NO SQL  
SELECT* FROM productquantity$$
```

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `s1` () NO SQL  
SELECT * FROM salesman$$
```

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `shop` () NO SQL  
select * FROM shop$$  
DELIMITER ;
```

2. DELIMITER \$\$

Procedures

```
CREATE DEFINER='root'@'localhost' PROCEDURE `s1`() NO SQL
SELECT * FROM register$$
DELIMITER ;
```

3.7 Triggers

Triggers are stored programs, which are automatically executed or fired when some events occur. Triggers are stored into database and invoked repeatedly, when specific condition match.

TRIGGER 1:

Triggers `ordercontains`

1. DELIMITER \$\$

```
CREATE TRIGGER `t1` AFTER INSERT ON `ordercontains` FOR EACH
ROW UPDATE productquantity SET productquantity.Pnum=productquantity.P
num-NEW.Qop$$
DELIMITER ;
```

3.8 Pseudocode

Pseudocode is an informal way of programming description that does not require any strict programming language syntax or underlying technology considerations. It is used for creating an outline or a rough draft of a program. Pseudocode summarizes a program's flow, but excludes underlying details.

It is an informal high-level description of the operating principle of a computer program or other algorithm. It uses the structural conventions of a normal programming language, but is intended for human reading rather than machine reading.

3.8.1 Algorithm for Login

Step 1: BEGIN

Step 2: Enter username and password.

Step 3: Verify the credentials entered with that in the login table.

Step 4: If credentials match, then proceed to home page else.

Step 5: End If

Step 6: END

3.8.2 Algorithm for Table Display

Step 1: BEGIN

Step 2: Establish connection with the database through XAMPP server.

Step 3: Define the Select query to retrieve all the values from the DBMS.

Step 4: To display all the values stored in the table.

Step 5: END

3.8.3 Algorithm for Insert

Step 1: BEGIN

Step 2: Get all the necessary values required for insertion.

Step 3: Define the query for insert.

Step 4: Execute the query

Step 5: END.

3.8.4 Algorithm for Update

Step 1: BEGIN

Step 2: Get all the necessary values required for update

Step 3: Define the query for updating.

Step 4: Execute the query using execute.

Step 5: END

3.8.5 Algorithm for Delete

Step 1: BEGIN

Step 3: Define the query for deleting.

Step 4: Execute the query.

Step 5: End

CHAPTER 4

RESULTS AND DISCUSSION

Snapshots

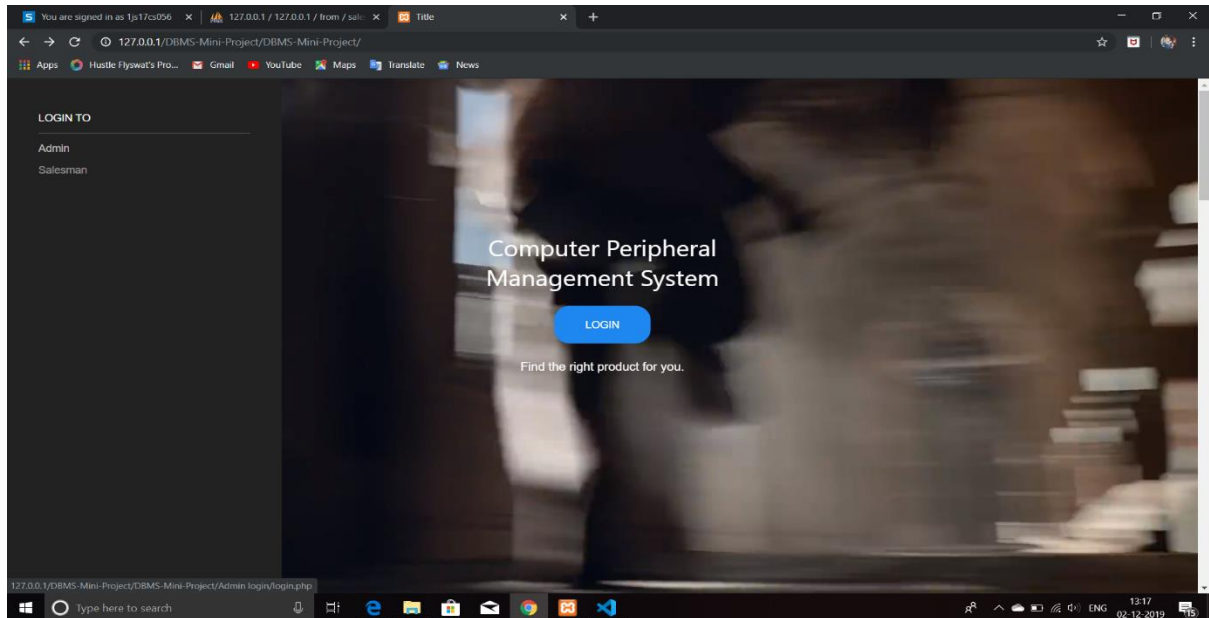


Fig 4.1 Welcome page.

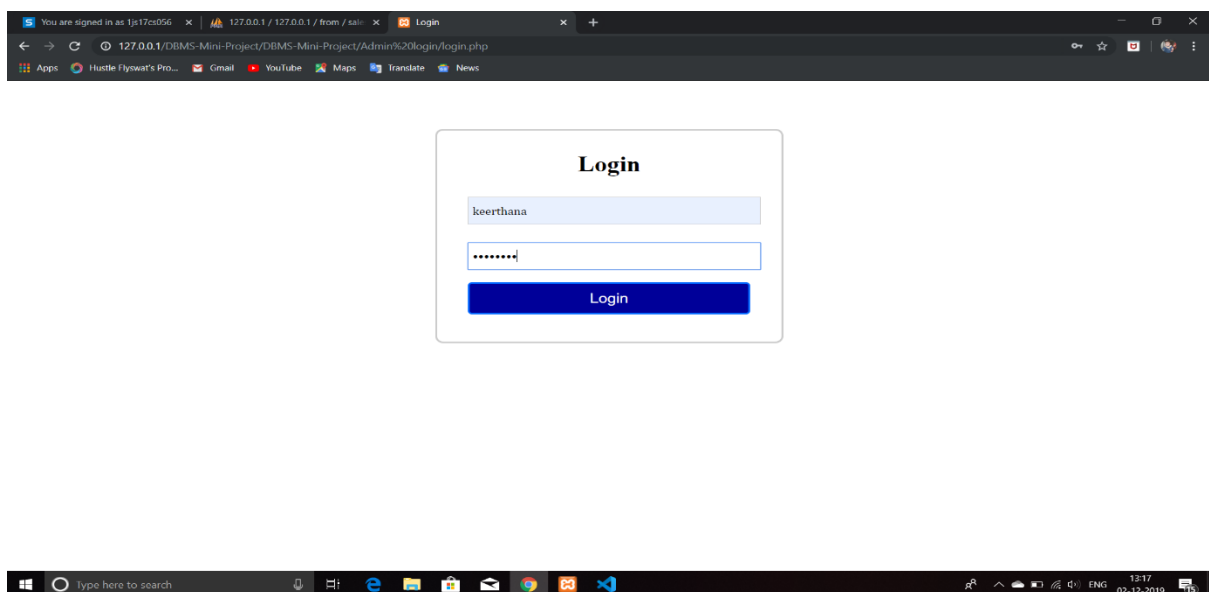


Fig 4.2 Login page for the admin

COMPUTER PHERIPERALS MANAGEMENT SYSTEM

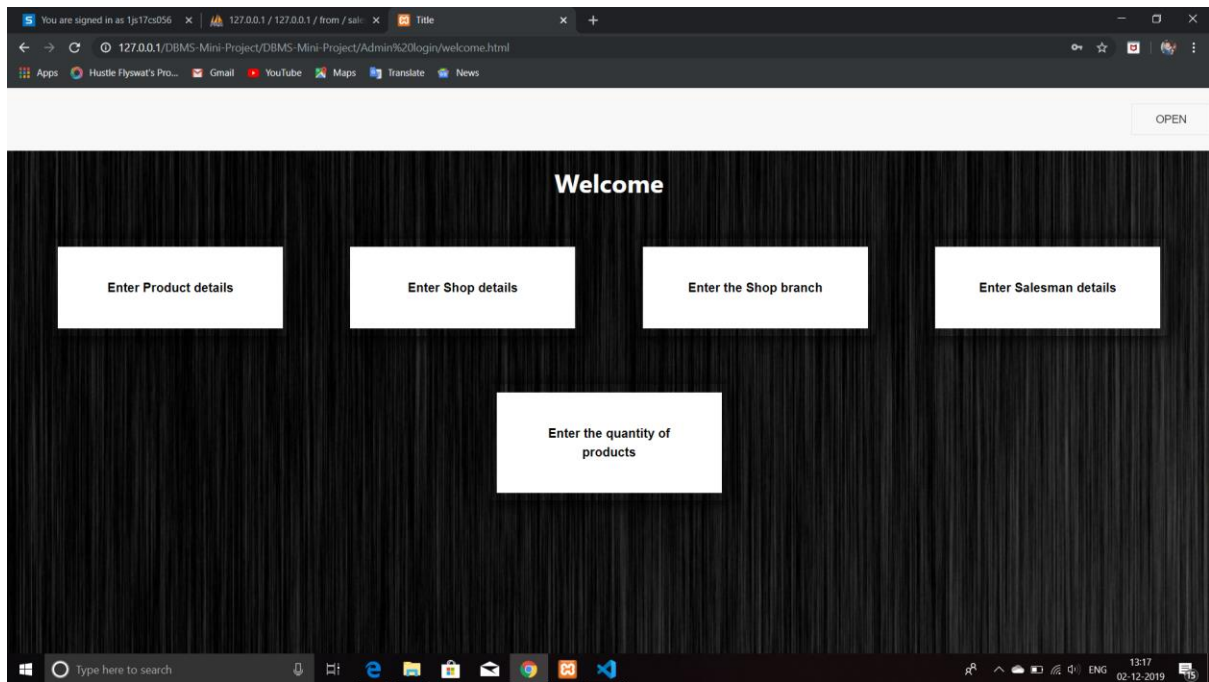


Fig 4.3 Welcome admin page

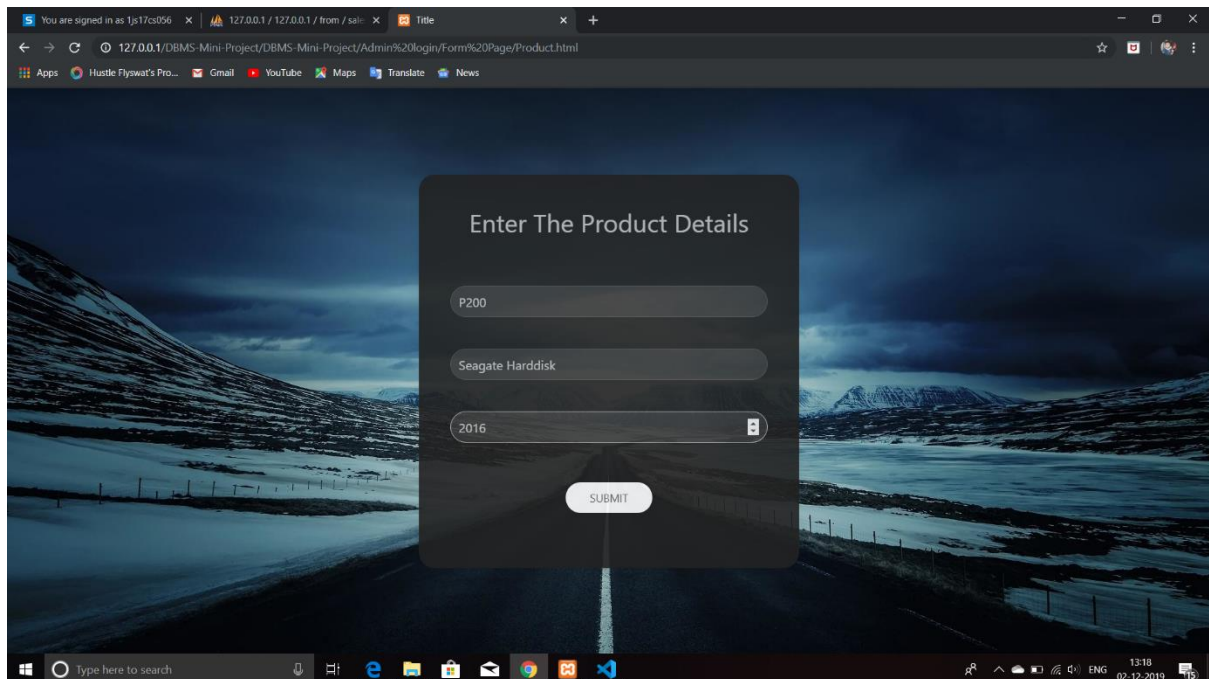


Fig 4.4 Entering the Product details

COMPUTER PHERIPERALS MANAGEMENT SYSTEM

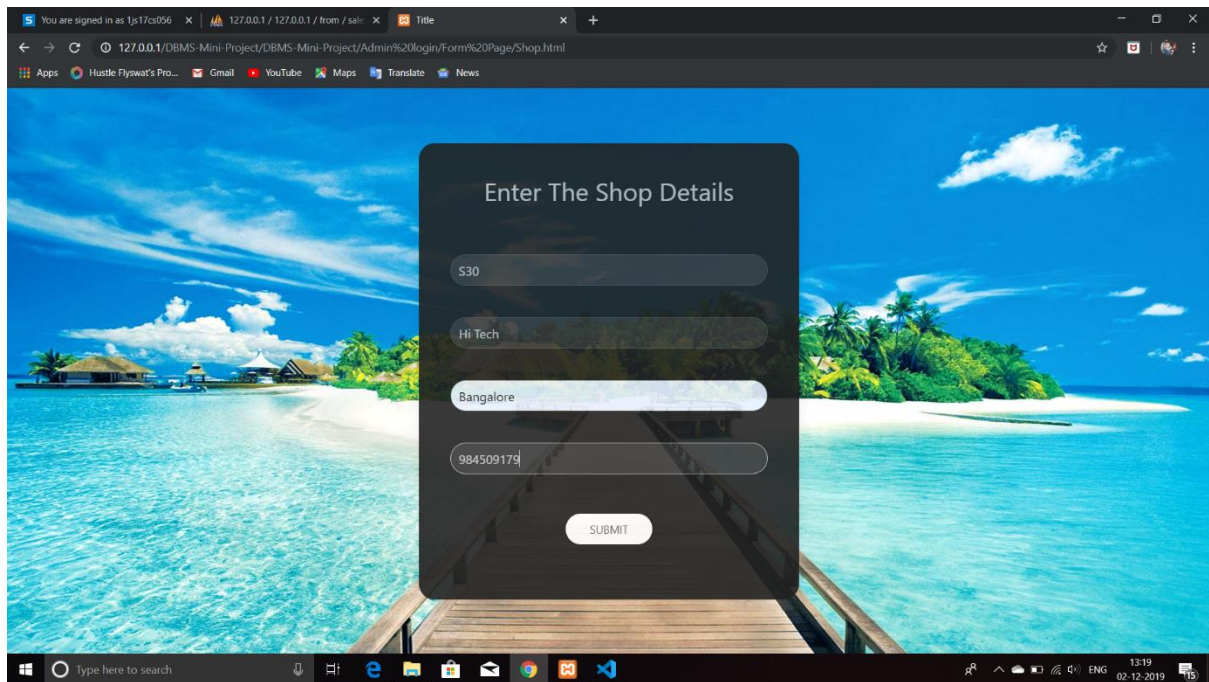


Fig 4.5 Entering of Shop Details

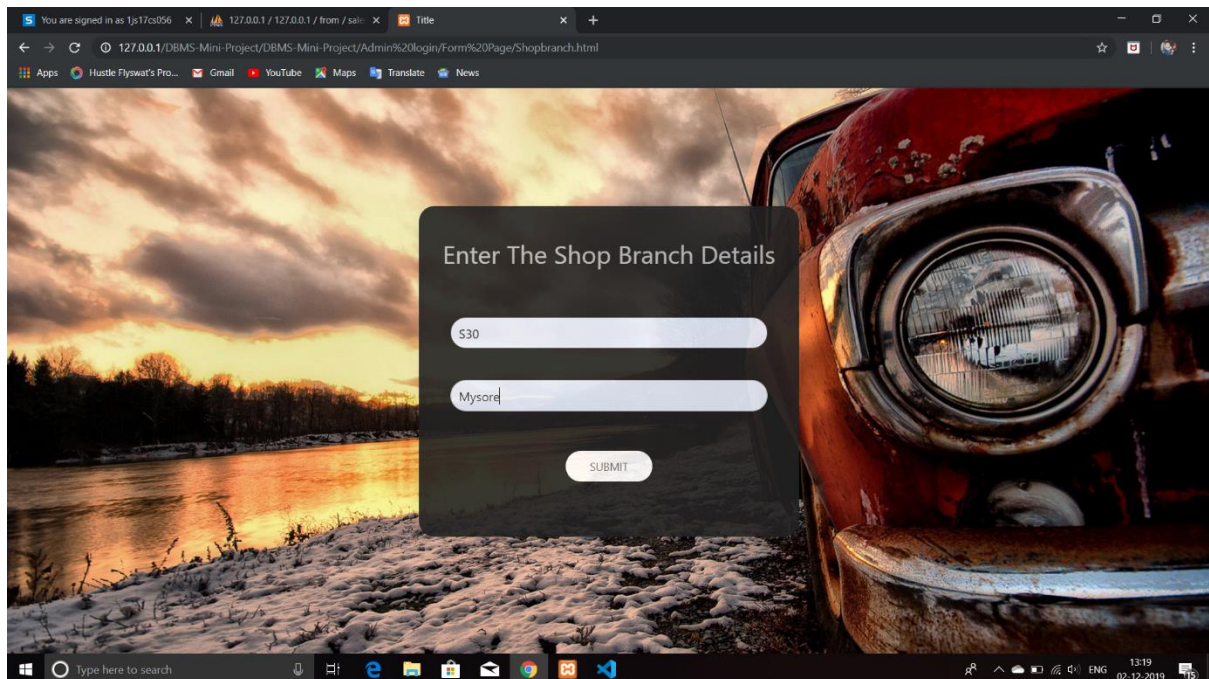


Fig 4.6 Entering the Shop Branch Details

COMPUTER PHERIPERALS MANAGEMENT SYSTEM

The screenshot shows a web browser window with the address bar displaying '127.0.0.1 / 127.0.0.1 / from / sal...'. The page title is 'Title'. The browser's address bar shows the URL '127.0.0.1/DBMS-Mini-Project/DBMS-Mini-Project/Admin%20login/Form%20Page/salesman.html'. The page content features a dark, semi-transparent modal box titled 'Enter The Salesman Details'. Inside the modal, there are five input fields: 'S15', 'Dhanush', 'Bangalore', 'S30', and 'Password'. A 'SUBMIT' button is located at the bottom of the modal. The background of the page is a blurred image of a city street at night with bokeh lights. The Windows taskbar is visible at the bottom, showing the search bar and various application icons.

Fig 4.7 Entering the Salesman Details

The screenshot shows a web browser window with the address bar displaying '127.0.0.1 / 127.0.0.1 / from / sal...'. The page title is 'Title'. The browser's address bar shows the URL '127.0.0.1/DBMS-Mini-Project/DBMS-Mini-Project/Admin%20login/Form%20Page/productquantity.html'. The page content features a dark, semi-transparent modal box titled 'Enter The Amount Of Products Required Details'. Inside the modal, there are three input fields: 'S30', 'P200', and '10'. A 'SUBMIT' button is located at the bottom of the modal. The background of the page is a scenic image of a wooden pier extending into a turquoise lagoon under a blue sky with white clouds. The Windows taskbar is visible at the bottom, showing the search bar and various application icons.

Fig 4.8 Entering the amount of products for a specific product

COMPUTER PHERIPERALS MANAGEMENT SYSTEM

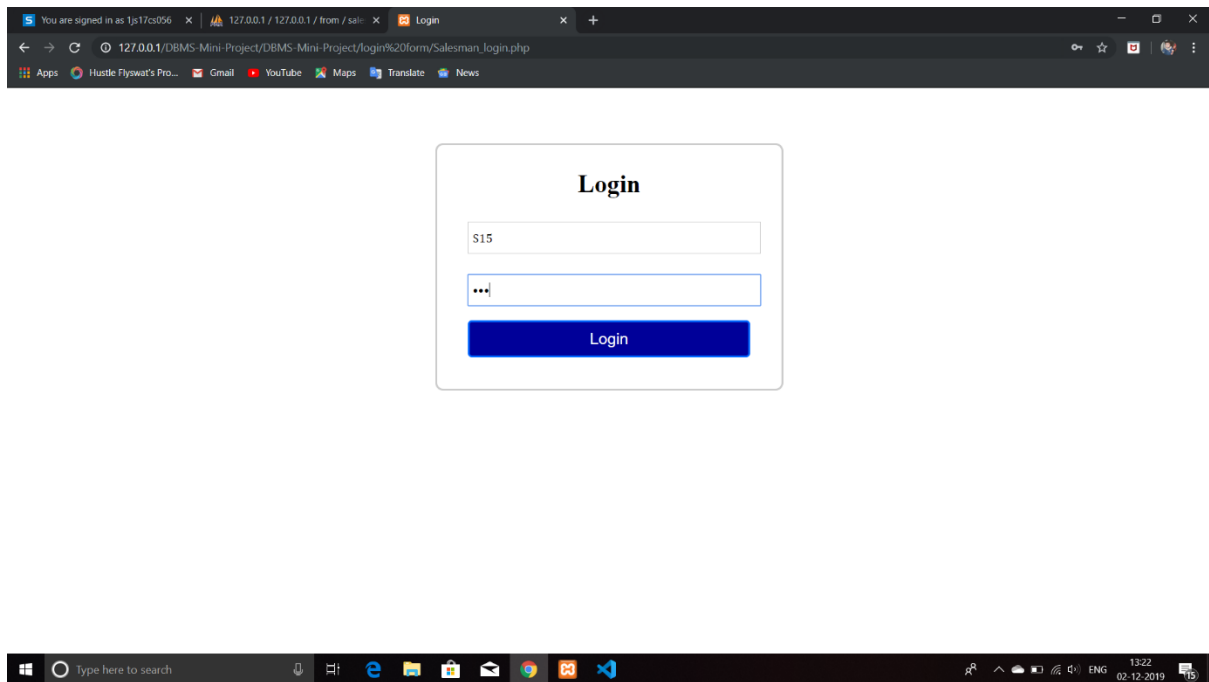


Fig 4.8 Login page for the salesman

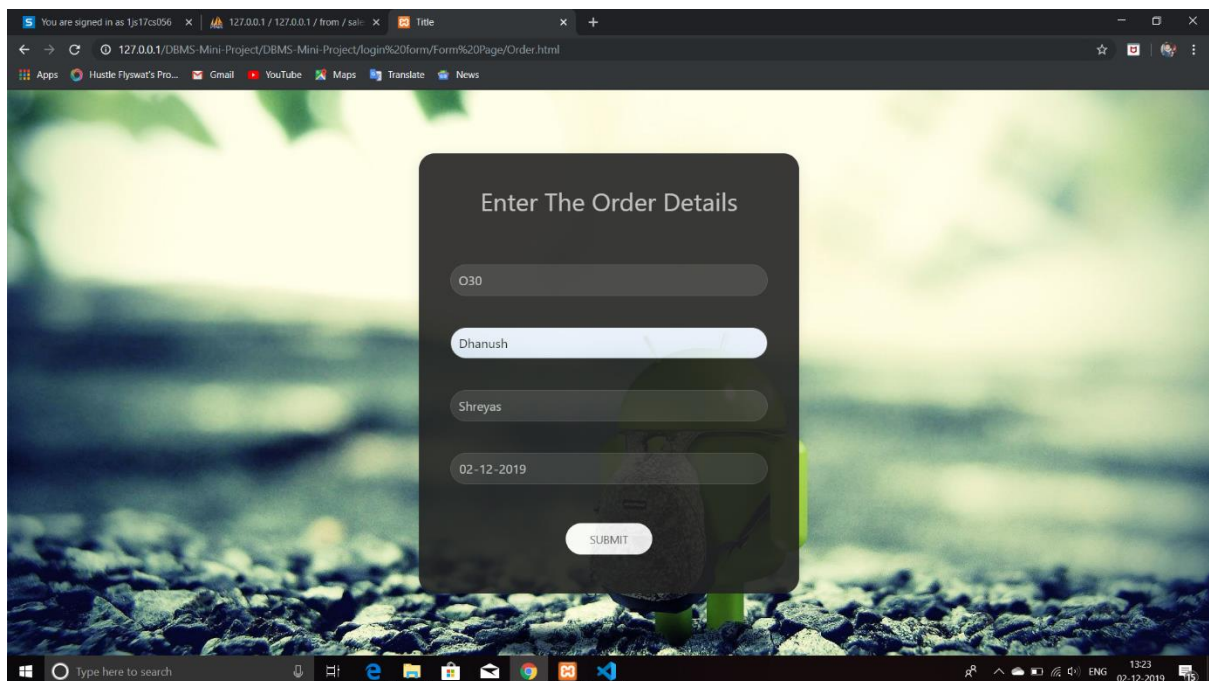
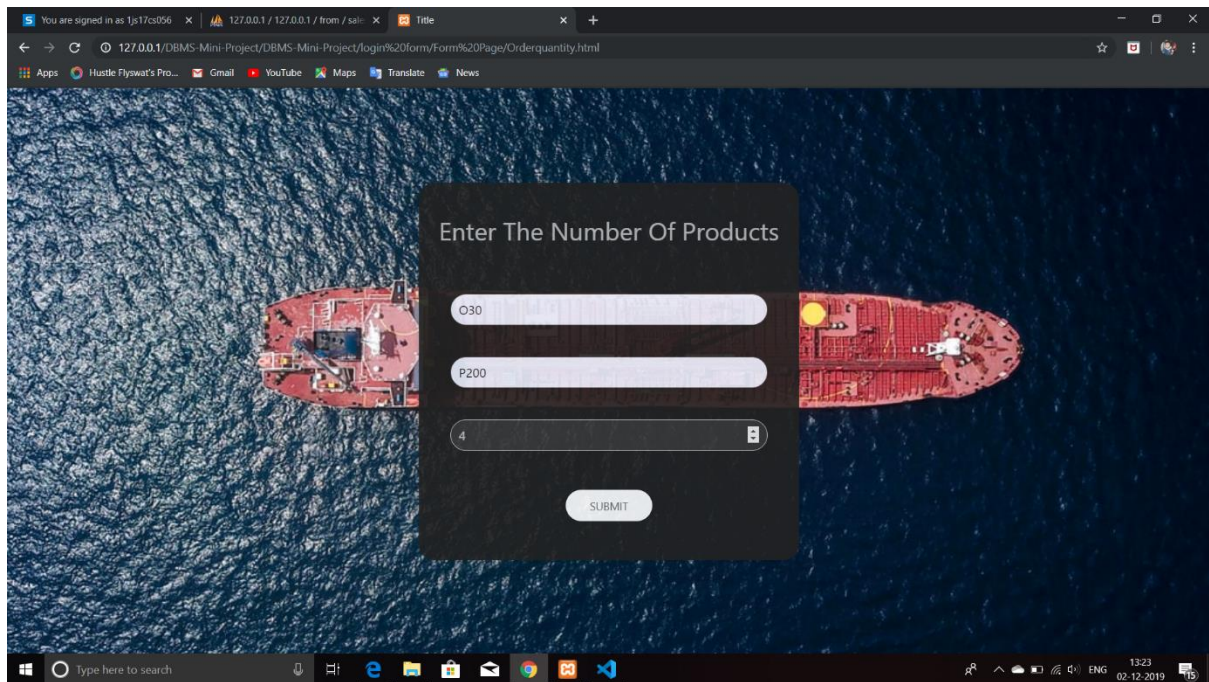


Fig 4.9 Entering the order details

COMPUTER PHERIPERALS MANAGEMENT SYSTEM



Enter The Number Of Products

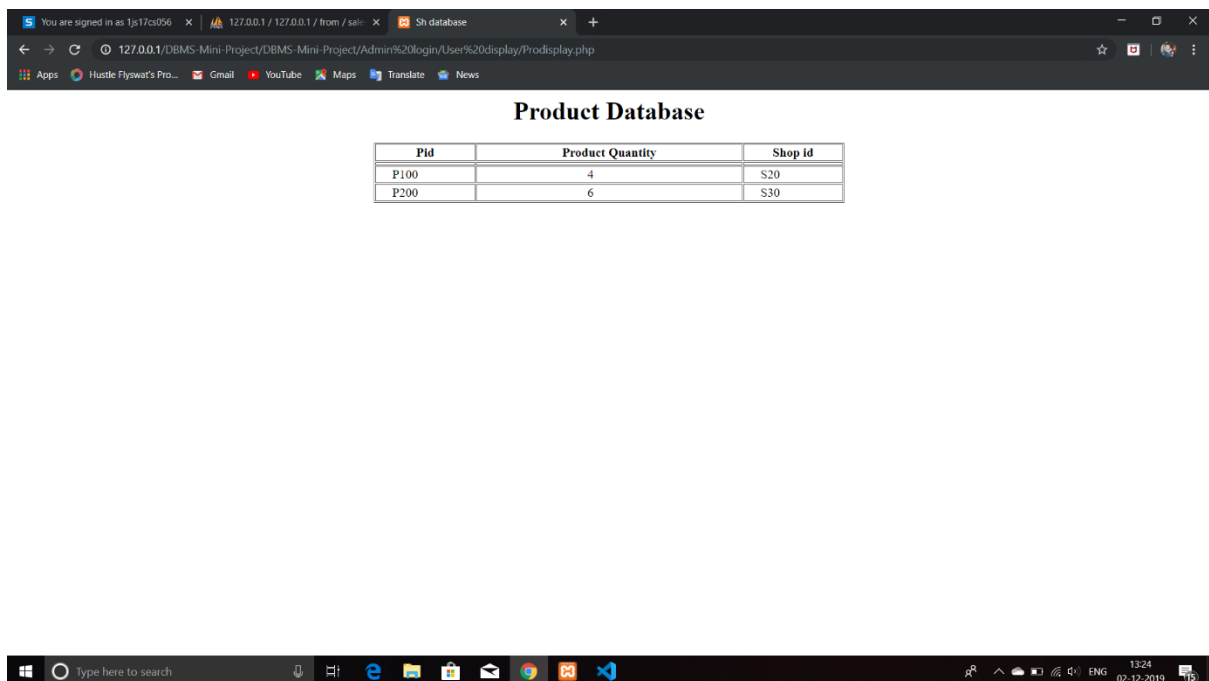
O30

P200

4

SUBMIT

Fig 4.10 Entering the number of products details required for the customer



Product Database

Pid	Product Quantity	Shop id
P100	4	S20
P200	6	S30

Fig 4.11 The total number of each products present in the database

CHAPTER 5

CONCLUSION AND FUTURE ENHANCEMENTS

5.1 Conclusion

The package was designed in such a way that future modifications can be done easily. The following conclusions can be deduced from the development of the project.

- Automation of the entire system improves the efficiency
- It provides a friendly graphical user interface which proves to be better when compared to the existing system.
- It gives appropriate access to the authorized users depending on their permissions.
- It effectively overcomes the delay in communications.
- Updating of information becomes so easier.
- System security, data security and reliability are the striking features.
- The System has adequate scope for modification in future if it is necessary.

5.2 Future Enhancements

- This application avoids the manual work and the problems concern with it. It is an easy way to obtain the information regarding the different scheduled examinations information that are currently issued.
- Well I and my team members have worked hard in order to present an improved website better than the existing one's regarding the information about the various activities.
- This helps the wholesale manufacturers to find a platform to showcase their products and sell them to the salesman. It also maintains a record of all the salesmen connected, the products sold, and the orders received. More enhancements such as billing process can also be added to the project.

REFERENCES

- [1] The Complete Reference- PHP by Steven Holzner (Indian Edition, McGraw Hill publications).
- [2] <http://www.mysqltutorial.org/calling-mysql-stored-procedure-from-xampp/>
- [3] XAMPP tutorial: <https://www.youtube.com/watch?v=ueWpNe0PG34>
- [4] Database systems models, languages, design and application programming- Ramez Elmasri and Shamkant B. Navathe (7th Edition, 2017, Pearson publications).
- [5] <https://mhaadi.wordpress.com/2010/10/18/the-evolution-of-database/>.
- [6] SoloLearn PHP tutorial