DBMS Mini Project Report
“Online Shopping Cart System”

Submitted by

Poojan Panchal	PA13 (S1032180249)
Sahil Mondal	PA24 (S1032180389)
Vignesh Charan Raman	PA33 (S1032180459)
Pranav Parnerkar	PA36 (S1032180563)

Under the Guidance of

Shakti Kinger

At

School of Computer Engineering and Technology

Abstract

Over the last decade, the ecommerce industry has boomed, taking over physical shopping due to its simplicity and convenience for the consumer. They indulge even more in these websites over the year because of strong discounts on the items. In view of these positive features, we find it an avenue worth putting in our technological expertise. We aim to create a user-friendly e-commerce website in this project. To place orders, the customer must sign up for the website and log in using their credentials. The user can also place several orders at a time, and monitor the website's order status. The system is robust enough for updating and clearing the customer's cart after placing an order. It also incorporates the transaction gateway for payment of the customer's orders. An admin page is provided for the seller to add new products and update the stocks of current products. The application is powered by MySQL as the backend, is helmed by HTML and CSS in the GUI department and Flask Web Interface (python library) for connectivity. The stakeholders of the website are the customer, the admin and the development team. The website will offer electronics in two categories i.e. mobiles and laptops.

Contents

List of Figures

Introduction

Essentially, ecommerce (or electronic commerce) is buying and selling of goods (or services) on the internet. Because of numerous advantages and benefits, more and more people say they prefer online shopping over conventional shopping these days. The buyer's decision-making process has changed dramatically in recent years. Buyers are conducting extensive research online before ever speaking to a sales person. Buyers are also making more direct purchases online and via their smartphone, never stepping foot into traditional brick-and-mortar locations.

Since then, ecommerce has evolved to make products easier to discover and purchase through online retailers and marketplaces. Independent freelancers, small businesses, and large corporations have all benefited from ecommerce, which enables them to sell their goods and services at a scale that was not possible with traditional offline retail.

Motivation & Objective:

Online buying and selling of goods has been imperative since a decade now and. From mobile shopping to online payment encryption and beyond, ecommerce encompasses a wide variety of data, systems, and tools for both online buyers and sellers. Most businesses with an ecommerce presence use an ecommerce store and/or an ecommerce platform to conduct both online marketing and sales activities and to oversee logistics and fulfilment

Problem Statement

With increase in public demand for miscellaneous electronics products specifically mobiles and phones, develop a User-interactive E-commerce application for online shopping integrated with a comely database architecture based on MySQL.

Tools & Technologies

- Python
- MySQL Database
- HTML
- CSS and Bootstrap
- Java Script

ER Diagram

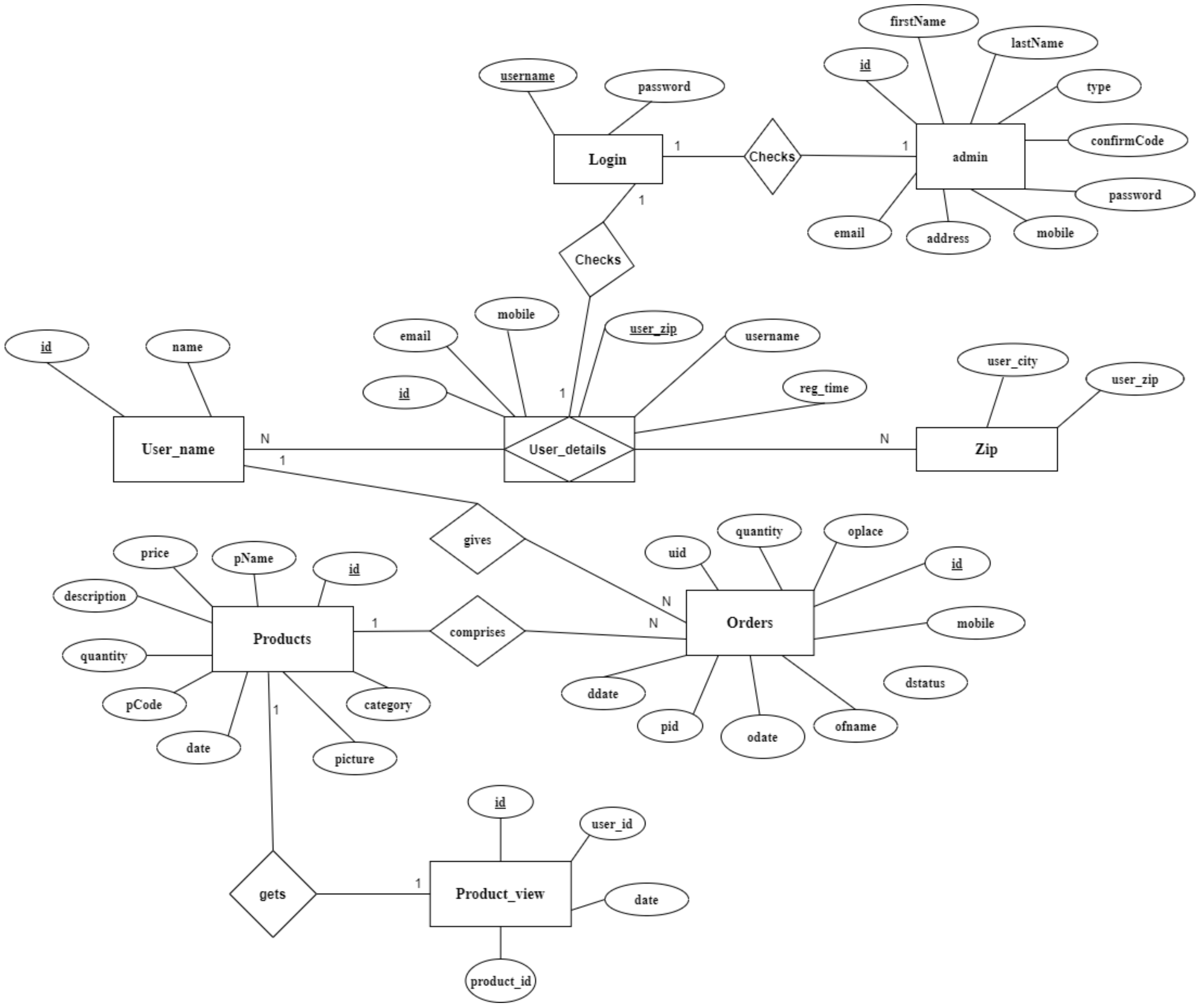


Figure 1: Entity Relationship Diagram

Schema Diagram

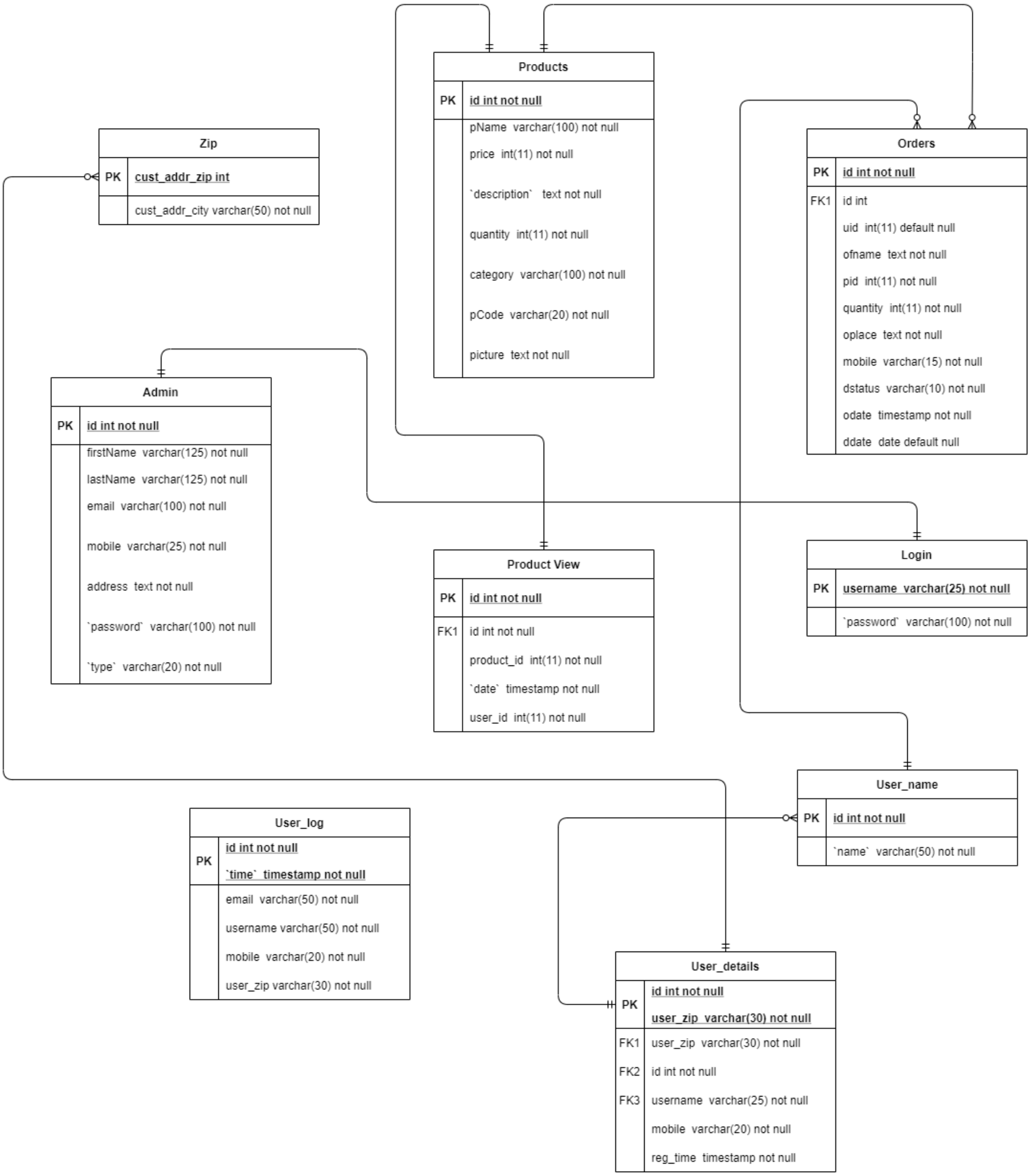

Products
PK
id int not null
pName varchar(100) not null
price int(11) not null
`description` text not null
quantity int(11) not null
category varchar(100) not null
pCode varchar(20) not null
picture text not null
Zip
PK
cust_addr_zip int
cust_addr_city varchar(50) not null
Orders
PK
id int not null
FK1
id int
uid int(11) default null
ofname text not null
pid int(11) not null
quantity int(11) not null
oplace text not null
mobile varchar(15) not null
dstatus varchar(10) not null
odate timestamp not null
ddate date default null
Admin
PK
id int not null
firstName varchar(125) not null
lastName varchar(125) not null
email varchar(100) not null
mobile varchar(25) not null
address text not null
`password` varchar(100) not null
`type` varchar(20) not null
Product View
PK
id int not null
FK1
id int not null
product_id int(11) not null
`date` timestamp not null
user_id int(11) not null
Login
PK
username varchar(25) not null
`password` varchar(100) not null
User_name
PK
id int not null
`name` varchar(50) not null
User_log
PK
id int not null
`time` timestamp not null
email varchar(50) not null
username varchar(50) not null
mobile varchar(20) not null
user_zip varchar(30) not null
User_details
PK
id int not null
user_zip varchar(30) not null
FK1
user_zip varchar(30) not null
FK2
id int not null
FK3
username varchar(25) not null
mobile varchar(20) not null
reg_time timestamp not null

Figure 2: Schema Diagram

Database Normalization

Parent Table:

users (id, name, email, username, password, mobile, user_city, user_zip, reg_time)

1st NF:

user_name (id, name)
user_details (id, email, username, password, mobile, user_city, user_zip, reg_time)

2nd NF:

Fd1: id, username -> email, mobile, user_city, user_zip, reg_time (Primary Key)
Fd2: username -> password (Partial dependency)
Fd3: user_zip -> user_city (Transitive Dependency)

Tables:

user_name (**id**, name)
user_details (**id**, **user_zip**, username, email, mobile, user_city, reg_time)
login (**username**, password)

3rd NF:

id -> user_zip, user_zip -> user_city

Hence:
user_details (**id**, username, email, mobile, user_zip, reg_time)
zip (**user_zip**, user_city)

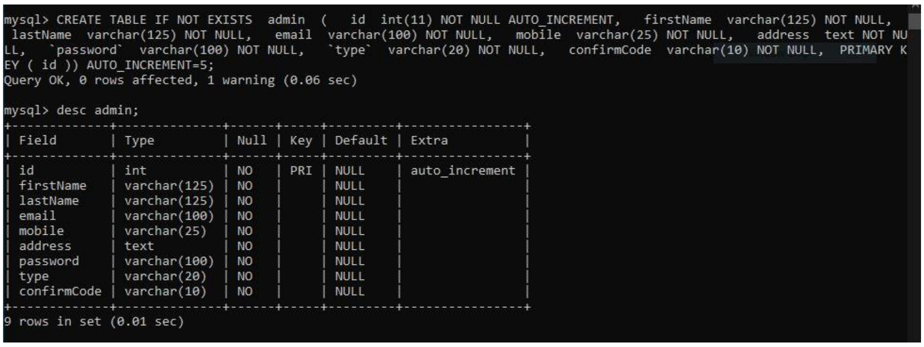
Final sub-tables of users after normalization :

user_name (**id**, name)
user_details (**id**, **user_zip** , username, email, mobile, reg_time)
zip (**user_zip**, user_city)
login (**username**, password)
Final Schema:

admin(**id**,firstName,lastName,email,mobile,address,’password’,’type’,confirmCode)
products(**id**, pName,price,’description’,quantity, category,pCode,picture,’date’)
orders(**id**,uid,ofname,pid,quantity,oplace,mobile,dstatus,odate,ddate)
product_view(**id,** user_id,product_id,’date’)
login(**username**,’password’)
user_name (**id**, name)
user_details (**id**, **user_zip**, username, email, mobile, reg_time)
zip (**user_zip**, user_city)

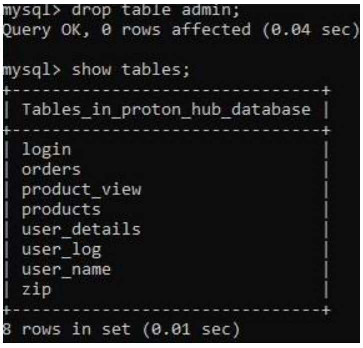
user_log(**<u>id</u>**, email, username, mobile, ‘time’, user_zip)

DDL Commands



```
mysql> CREATE TABLE IF NOT EXISTS  admin  (   id  int(11) NOT NULL AUTO_INCREMENT,   firstName  varchar(125) NOT NULL,
 lastName  varchar(125) NOT NULL,   email  varchar(100) NOT NULL,   mobile  varchar(25) NOT NULL,   address  text NOT NU
LL,   `password`  varchar(100) NOT NULL,   `type`  varchar(20) NOT NULL,   confirmCode  varchar(10) NOT NULL,  PRIMARY K
EY ( id )) AUTO_INCREMENT=5;
Query OK, 0 rows affected, 1 warning (0.06 sec)

mysql> desc admin;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| id          | int          | NO   | PRI | NULL    | auto_increment |
| firstName   | varchar(125) | NO   |     | NULL    |                |
| lastName    | varchar(125) | NO   |     | NULL    |                |
| email       | varchar(100) | NO   |     | NULL    |                |
| mobile      | varchar(25)  | NO   |     | NULL    |                |
| address     | text         | NO   |     | NULL    |                |
| password    | varchar(100) | NO   |     | NULL    |                |
| type        | varchar(20)  | NO   |     | NULL    |                |
| confirmCode | varchar(10)  | NO   |     | NULL    |                |
+-------------+--------------+------+-----+---------+----------------+
9 rows in set (0.01 sec)
```

I)

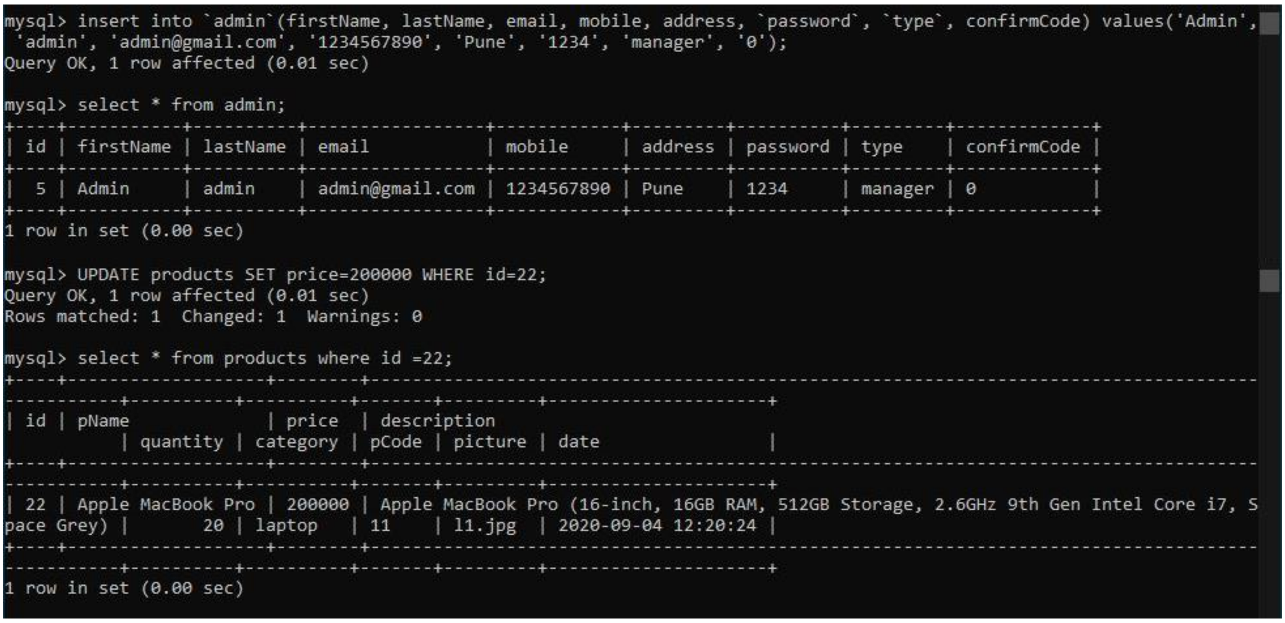


```
mysql> drop table admin;
Query OK, 0 rows affected (0.04 sec)

mysql> show tables;
+-----------------------------+
| Tables_in_proton_hub_database |
+-----------------------------+
| login                       |
| orders                      |
| product_view                |
| products                    |
| user_details                |
| user_log                    |
| user_name                   |
| zip                         |
+-----------------------------+
8 rows in set (0.01 sec)
```

II)

Figure 3: DDL Commands

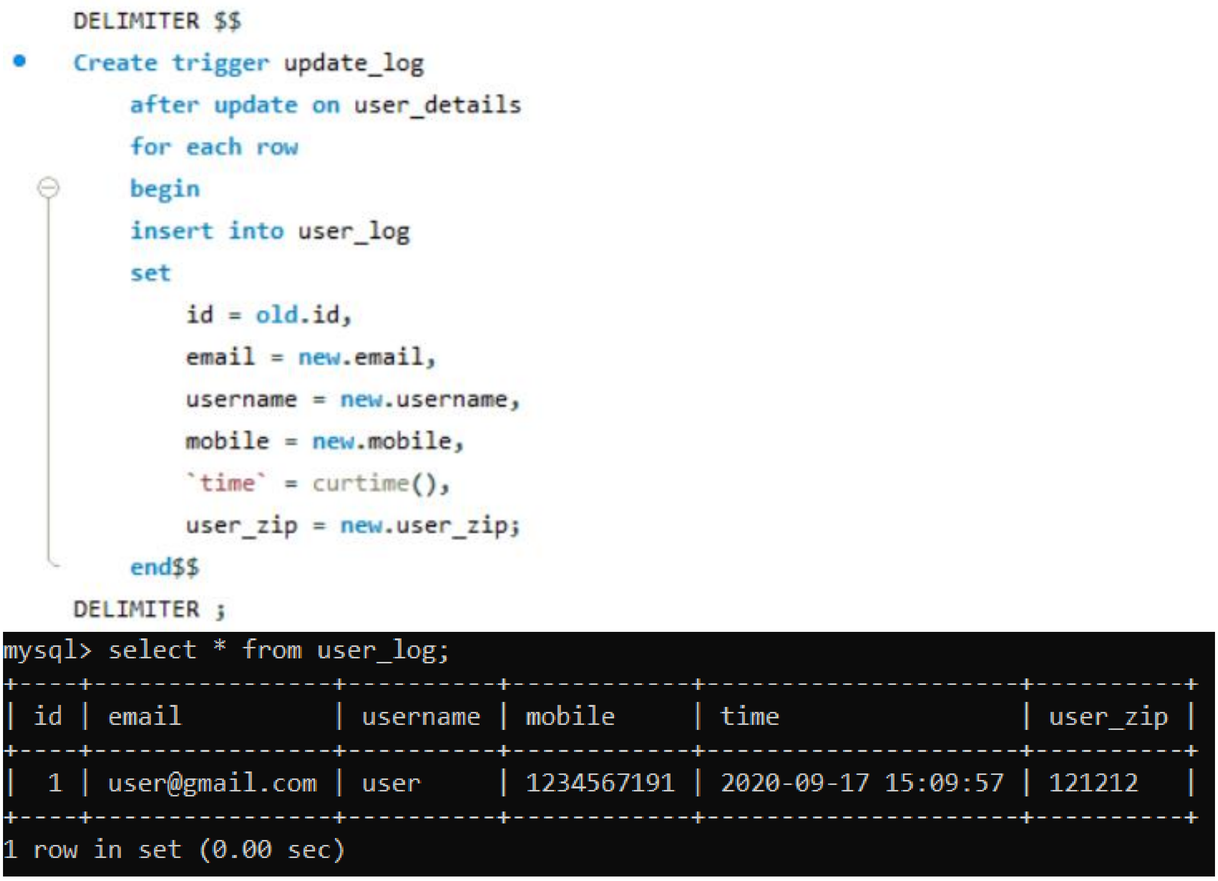
DML Commands



```
mysql> insert into `admin`(firstName, lastName, email, mobile, address, `password`, `type`, confirmCode) values('Admin',
 'admin', 'admin@gmail.com', '1234567890', 'Pune', '1234', 'manager', '0');
Query OK, 1 row affected (0.01 sec)

mysql> select * from admin;
+----+-----------+----------+-----------------+------------+---------+----------+---------+-------------+
| id | firstName | lastName | email           | mobile     | address | password | type    | confirmCode |
+----+-----------+----------+-----------------+------------+---------+----------+---------+-------------+
|  5 | Admin     | admin    | admin@gmail.com | 1234567890 | Pune    | 1234     | manager | 0           |
+----+-----------+----------+-----------------+------------+---------+----------+---------+-------------+
1 row in set (0.00 sec)

mysql> UPDATE products SET price=200000 WHERE id=22;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from products where id =22;
+----+-------------------+--------+------------------------------------------------------------------------------------------
----------+----------+----------+-------+---------+---------------------+
| id | pName             | price  | description
          | quantity | category | pCode | picture | date                |
+----+-------------------+--------+------------------------------------------------------------------------------------------
----------+----------+----------+-------+---------+---------------------+
| 22 | Apple MacBook Pro | 200000 | Apple MacBook Pro (16-inch, 16GB RAM, 512GB Storage, 2.6GHz 9th Gen Intel Core i7, S
pace Grey) |       20 | laptop   | 11    | l1.jpg  | 2020-09-04 12:20:24 |
+----+-------------------+--------+------------------------------------------------------------------------------------------
----------+----------+----------+-------+---------+---------------------+
1 row in set (0.00 sec)
```

Figure 4: DML Commands

Triggers

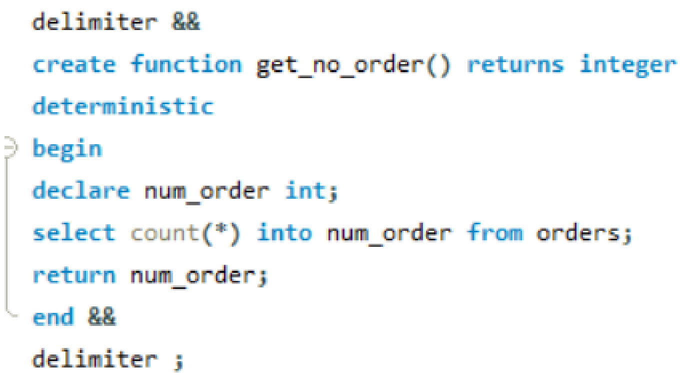


```
DELIMITER $$
Create trigger update_log
    after update on user_details
    for each row
    begin
    insert into user_log
    set
        id = old.id,
        email = new.email,
        username = new.username,
        mobile = new.mobile,
        `time` = curtime(),
        user_zip = new.user_zip;
    end$$
DELIMITER ;
```

```
mysql> select * from user_log;
+----+----------------+----------+------------+---------------------+----------+
| id | email          | username | mobile     | time                | user_zip |
+----+----------------+----------+------------+---------------------+----------+
|  1 | user@gmail.com | user     | 1234567191 | 2020-09-17 15:09:57 | 121212   |
+----+----------------+----------+------------+---------------------+----------+
1 row in set (0.00 sec)
```

Figure 5: Triggers Implementation

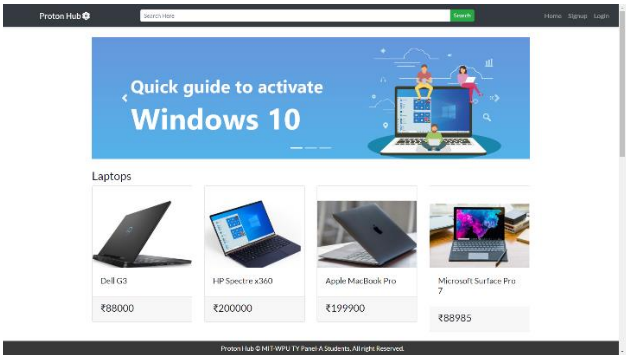
Functions



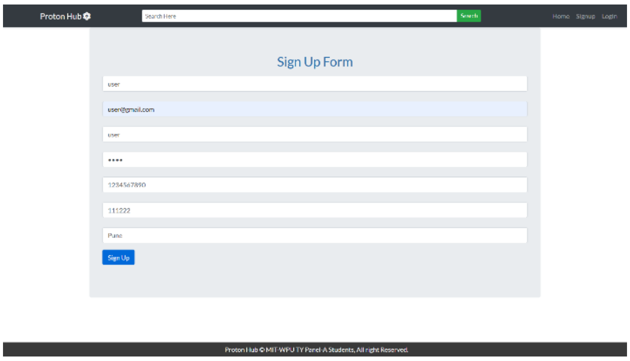
```
delimiter &&
create function get_no_order() returns integer
deterministic
begin
declare num_order int;
select count(*) into num_order from orders;
return num_order;
end &&
delimiter ;
```

Figure 6: Functions

Frontend GUI screenshots (User-End)

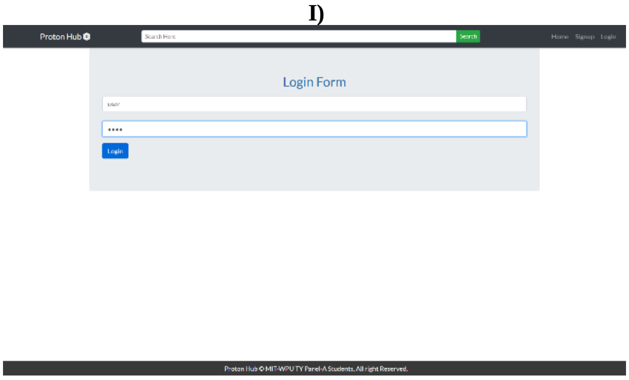


I)



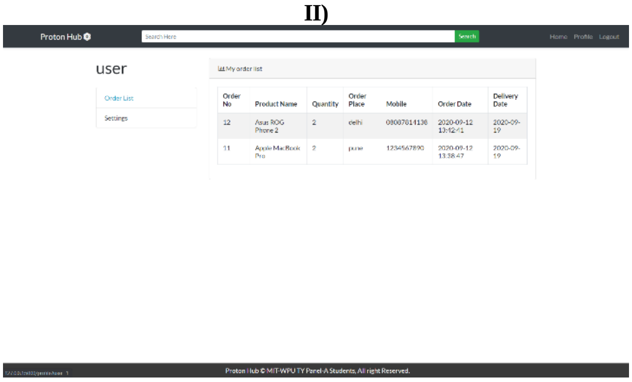
user
user@gmail.com
user
****
1234567890
111222
Pune

II)


Proton Hub
Search Here
Search
Home Signup Login
Login Form
Login
Proton Hub © MIT-WPU TY Panel-A Students, All right Reserved.

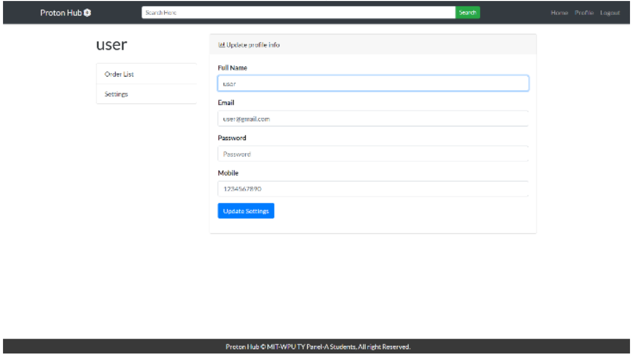
user
****

III)



Order No	Product Name	Quantity	Order Place	Mobile	Order Date	Delivery Date
12	Asus ROG Phone 2	2	delhi	08087814138	2020-09-12 13:12:41	2020-09-19
11	Apple MacBook Pro	2	pune	1234567890	2020-09-12 13:38:47	2020-09-19

IV)



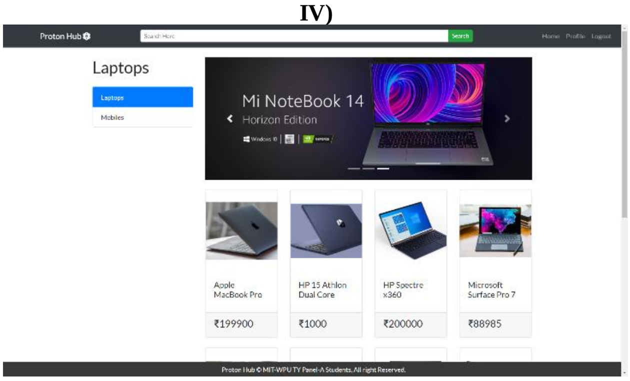
Full Name
user

Email
user@gmail.com

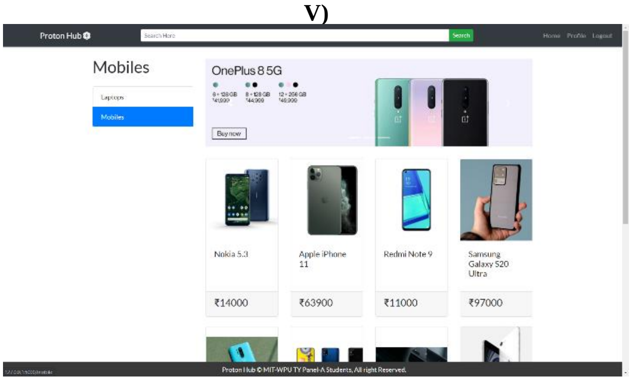
Password
Password

Mobile
1234567890

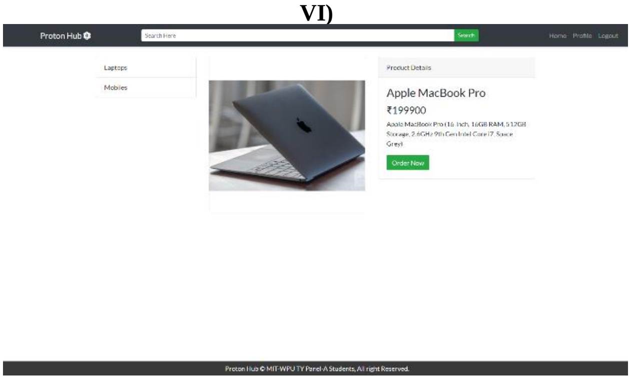
V)



VI)



VII)

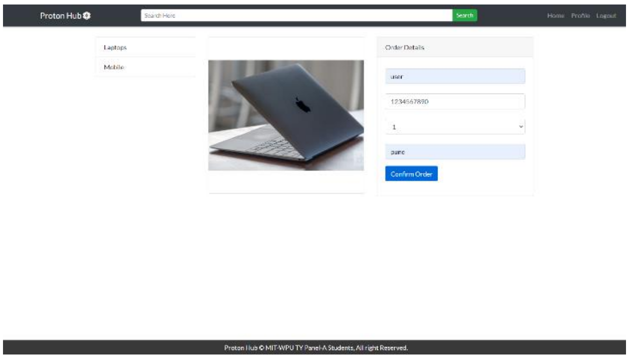


Apple MacBook Pro

₹199900

Apple MacBook Pro (16-inch, 16GB RAM, 512GB Storage, 2.6GHz 9th Gen Intel Core i7, Space Grey)

VIII)



user

1234567890

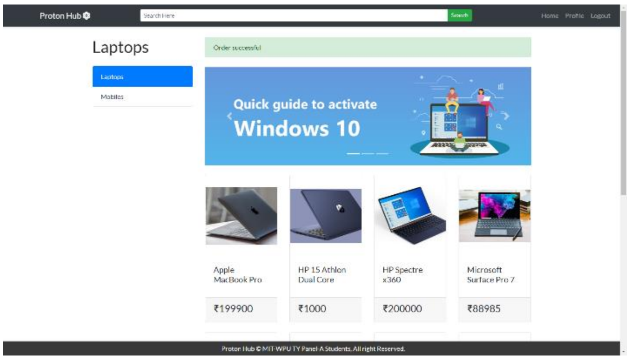
1

pune

Confirm Order

Proton Hub © MIT-WPU TY Panel-A Students, All right Reserved.

IX)

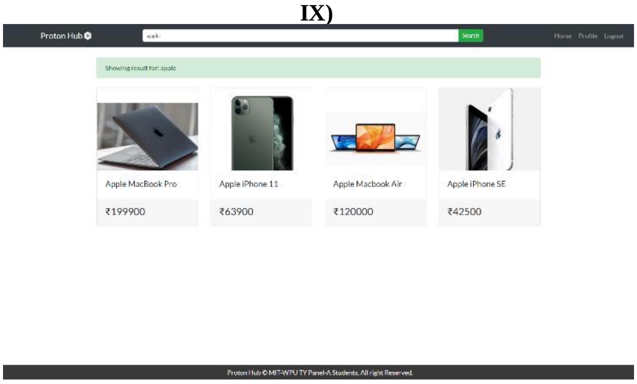


Order successful

Apple MacBook Pro	HP 15 Athlon Dual Core	HP Spectre x360	Microsoft Surface Pro 7
₹199900	₹1000	₹200000	₹88985

Proton Hub © MIT-WPU TY Panel-A Students, All right Reserved.

X)



Showing result for: apple

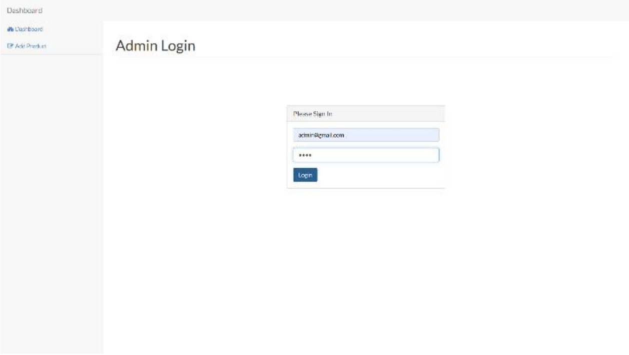
Apple MacBook Pro	Apple iPhone 11	Apple Macbook Air	Apple iPhone SE
₹199900	₹63900	₹120000	₹42500

Proton Hub © MIT-WPU TY Panel-A Students, All right Reserved.

XII)

Figure 7: User-side pages

Frontend GUI screenshots (Admin-End)

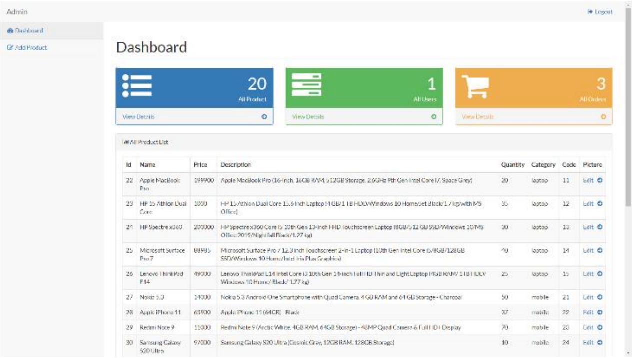


admin@gmail.com

••••

Login

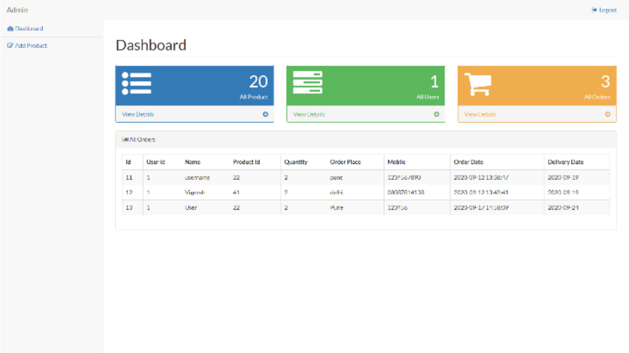
I)



All Product List

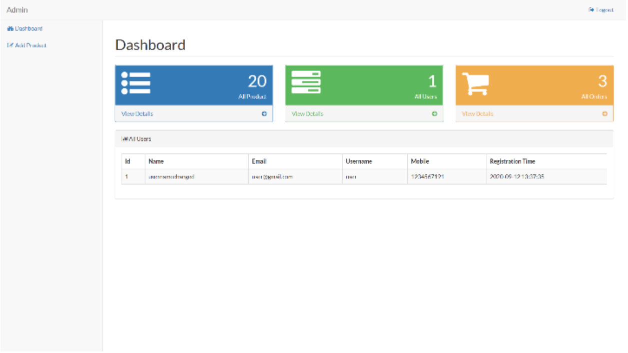
Id	Name	Price	Description	Quantity	Category	Code	Picture
22	Apple MacBook Pro	199900	Apple MacBook Pro (16-inch, 16GB RAM, 512GB Storage, 2.6GHz 9th Gen Intel Core i7, Space Grey)	20	laptop	11	Edit
23	HP 15 Athlon Dual Core	1000	HP 15 Athlon Dual Core 15.6 inch Laptop (4 GB/1 TB HDD/Windows 10 Home/Jet Black/1.7 kg/with MS Office)	35	laptop	12	Edit
24	HP Spectre x360	200000	HP Spectre x360 Core i5 10th Gen 13-inch FHD Touchscreen Laptop (8GB/512 GB SSD/Windows 10/MS Office 2019/Nightfall Black/1.27 kg)	30	laptop	13	Edit
25	Microsoft Surface Pro 7	88985	Microsoft Surface Pro 7 12.3 inch Touchscreen 2-in-1 Laptop (10th Gen Intel Core i5/8GB/128GB SSD/Windows 10 Home/Intel Iris Plus Graphics)	40	laptop	14	Edit
26	Lenovo ThinkPad E14	49000	Lenovo ThinkPad E14 Intel Core i3 10th Gen 14-inch Full HD Thin and Light Laptop (4GB RAM/ 1TB HDD/ Windows 10 Home/ Black/ 1.77 kg)	25	laptop	15	Edit
27	Nokia 5.3	14000	Nokia 5.3 Android One Smartphone with Quad Camera, 4 GB RAM and 64 GB Storage - Charcoal	50	mobile	21	Edit
28	Apple iPhone 11	63900	Apple iPhone 11 (64GB) - Black	37	mobile	22	Edit
29	Redmi Note 9	11000	Redmi Note 9 (Arctic White, 4GB RAM, 64GB Storage) - 48MP Quad Camera & Full HD+ Display	70	mobile	23	Edit
30	Samsung Galaxy S20 Ultra	97000	Samsung Galaxy S20 Ultra (Cosmic Gray, 12GB RAM, 128GB Storage)	10	mobile	24	Edit

II)



All Orders

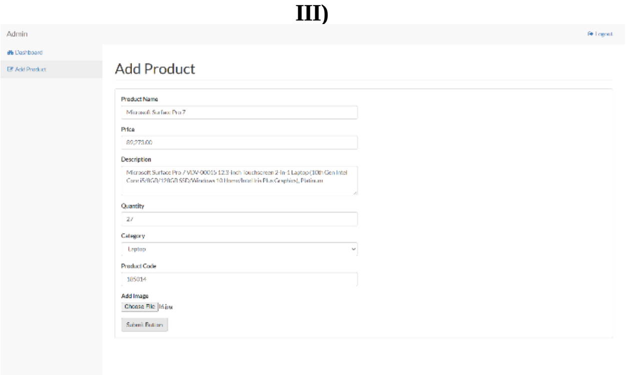
Id	User Id	Name	Product Id	Quantity	Order Place	Mobile	Order Date	Delivery Date
11	1	username	22	2	pune	1234567890	2020-09-12 13:38:17	2020-09-19
12	1	Vignesh	41	2	delhi	08087814138	2020-09-12 13:42:41	2020-09-19
13	1	User	22	2	Pune	123456	2020-09-17 14:58:09	2020-09-24



All Users

Id	Name	Email	Username	Mobile	Registration Time
1	usernamechanged	user@gmail.com	user	1234567121	2020-09-12 13:37:35

III) IV)



Add Product

Product Name
Microsoft Surface Pro 7

Price
89,273.00

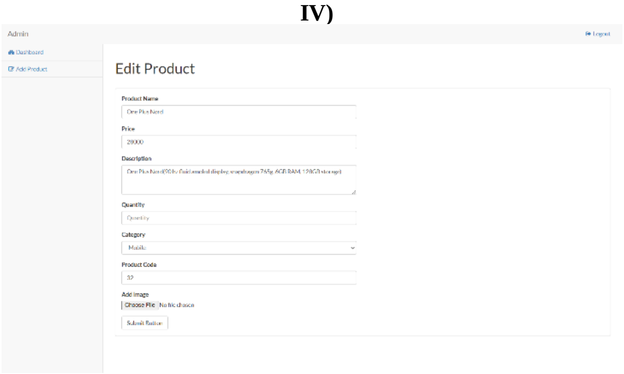
Description
Microsoft Surface Pro 7 VDV-00015 12.3-inch Touchscreen 2-in-1 Laptop(10th Gen Intel Core i5/8GB/128GB SSD/Windows 10 Home/Intel Iris Plus Graphics), Platinum

Quantity
27

Category
Laptop

Product Code
105014

Add Image
16.jpg



Edit Product

Product Name
One Plus Nord

Price
20000

Description
One Plus Nord(90 hz fluid amoled display, snapdragon 765g, 6GB RAM, 128GB storage)

Quantity
Quantity

Category
Mobile

Product Code
32

Add Image
No file chosen

V) VI)

Figure 8: Admin side pages

Conclusion

In conclusion, a database is a far more efficient mechanism to store and organize data than spreadsheets, it allows for a centralized facility that can easily be modified and quickly shared among multiple users. Having a web based front end removes the requirement of users having to understand and use a database directly, and allows users to connect from anywhere with an internet connection and a basic web browser. It also allows the possibility of queries to obtain information for various surveys.

References

[1] JavaScript Enlightenment, Cody Lindley-First Edition, based on JavaScript 1.5, ECMA-262, Edition

[2] Complete CSS Guide, Maxine Sherrin and John Allsopp-O'Reilly Media; September 2012

[3] Database Systems Models, Languages, Design and Application Programming, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson.

[4] http://www.w3schools.com/html/defualt.asp,http://www.w3schools.com/css/default.asp,http://www.w3schools.com/js/default.asp

[5] Cai Y. Cude B. J. (2008). Online shopping. In Xiao J. J. (Ed.), Handbook of Consumer Finance Research. New York: Springer Publishing. 10.1007/978-0-387-75734-6_9