西北工業大學
NORTHWESTERN POLYTECHNICAL UNIVERSITY

Lab report

Name : ABID ALI

Student_No. : 2019380141

Experiment 8

Online Shopping System

Contents

西北工業大學
NORTHWESTERN POLYTECHNICAL UNIVERSITY

Online Shopping System

Lab report 1

Section 1: Goal 3

Section 2: Documentation 3

Section 3: Requirement: 5

Section :4 Pictures 16

Section : 5 Configurations 26

Section 6: Conclusion 40

Section 1: Goal

The goal of this project is to provide a realistic experience in the conceptual design, logical design, implementation, operation, and maintenance of a relational database and associated a application.

Section 2: Documentation

Online Shopping Cart Application

Abstract:

Electronic commerce, also known as ecommerce is a type of industry where buying and selling of a product is conducted over electronic systems such as the internet.
The purpose of this application is to bring knowledge to students about ecommerce and how an interactive ecommerce application can be designed from scratch using client-side languages, such as JavaScript and HTML, combined with the server-side PHP language through Apache Server Faces. It is responsible for taking information from the database and making it available to the UI by mapping the category or item ID to the respective IDs stored in the MySql/PHP database. The client side is responsible for showing the entire user interface, containing the CSS, HTML, JavaScript and PHP.

Introduction:

It is known globally that, in today's market, it is extremely difficult to start a new small-scale business and its sustenance with competition from the well-established and settled/brand owners. Most often, even if the quality of the product is really good, due to a lack of advertisement or business at the small scale, it just becomes another face in the sea, and the product does not reach a larger group of customers. In fast paced life of today when everyone is squeezed for time, the majority of people are finicky when it comes to doing physical shopping. Logistically, a consumer finds a product more interesting and attractive when they find it on the website of a retailer directly and are able to see item's details online.The customers of today are not only attracted because online shopping is very convenient, but also because they have broader selections, highly competitive prices, better information about the product (including people's reviews) and extremely simplified navigation for searching regarding the product. Moreover, business owners often offer online shopping options at low rates because the overhead expenses in opening and running a physical store are higher. Further, with online shopping, their products have access to a worldwide market, which increases the number of customers from different ethnic groups, adds customer value, and overall sustainable in the marketing.
Online web stores, such as Amazon and eBay, have gained huge popularity over the years because one can buy almost everything at these stores. These web stores also give an opportunity to a lot of small-scale companies and manufactures to reach the global market and to directly sell their products to people without involving different other companies or middlemen before their product can reach the shelves of a physical store. Further, instead of using the available platforms, manufacturers can bring a concept of designing their own web store to sell their products directly to the masses.

Aim of this project:

This project has developed to help me to learn about application designing using JavaScript, HTML and PHP from their basic capabilities. This application allows to understand the basics about the appearance of a first web page and how a complete working application can be built from scratch. It also allows students to understand the concept of user-integrated graphics and how JavaScript can be embedded into HTML. Further, it gives insight about how the client-side language interacts with the server-side language, Java, C# or PHP and finally with the database(MySql). This shopping-cart application is designed, primarily, for me to learn and understand the concept of application development, and can also be used to learn ecommerce and web-application topics. This shopping-cart application is very versatile and can be enhanced by adding more functions and modified graphics for use with commercial purposes.

Interface:

The two interface types found in the online shopping-cart application are as follows:

1. **User Interface**: Users are able to view the home page of the shopping-cart application, browse the different categories, browse and add any number of items from any categories in the shopping cart, look for information about each product, delete the items in the shopping cart, save the cart for later viewing, check out or continue shopping after adding the item to the cart, and check out the items by completing the required information in the order form.

2. **Admin Interface**: The administrator is able to view the users' information that was entered during checkout in the database, can update the information, price, add or remove items from the main display.

Section 3: Requirement:

Entity Relationship Diagram

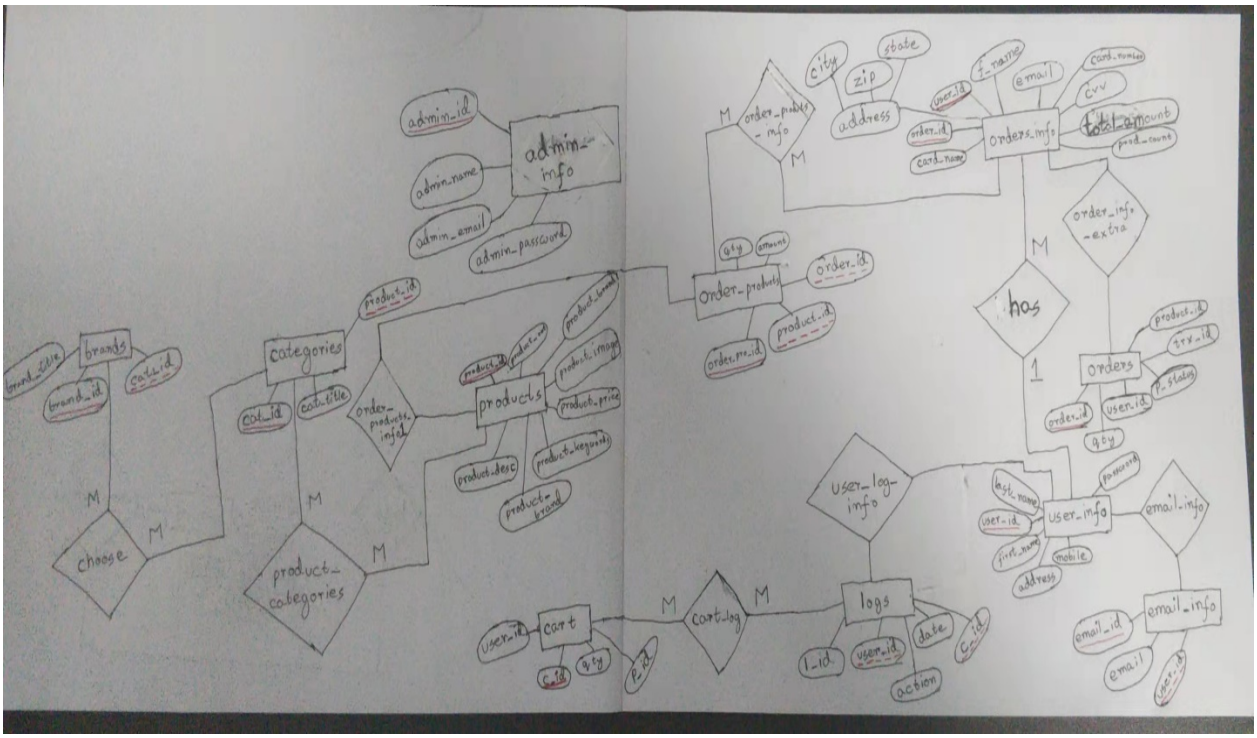


Table Dictionary

admin_info

Field Name	Data type	Size	Constants	Description	Extra
admin_id	int	11	PRI	Admin ID	auto_increment
admin_name	varchar	100	Not Null	Admin Name	
admin_email	varchar	100	Not Null	Admin's Email	
admin_password	Varchar	100	Not Null	Admin's Password	

brands

Field Name	Data type	Size	Constants	Description	Extra
brand_id	int	11	PRI	Brand's ID	auto_increment
brand_title	text		Not Null	Brand Title	

cart

Field Name	Data type	Size	Constants	Description	Extra
id	int	11	PRI	Cart ID	auto_increment
p_id	int	11	Not Null	Package ID	
ip_add	varchar	250	Not Null	IP Address	
user_id	int	11	Null	User ID	
qty	int	11	Not Null	Quatity	

categories

Field Name	Data type	Size	Constants	Description	Extra
cat_id	int	11	PRI	Categories ID	auto_increment
cat_title	text		Not Null	Categories Title	

email_info

Field Name	Data type	Size	Constants	Description	Extra
emai_id	int	11	PRI	Email's ID	auto_increment
email	text		Not Null	Email	

logs

Field Name	Data type	Size	Constants	Description	Extra
id	int	11	PRI	Log ID	auto_increment
user_id	varchar	50	Not Null	User ID	
action	varchar	50	Not Null	Action by user	
date	datetime		Not Null	Time	

orders

Field Name	Data type	Size	Constants	Description	Extra
order_id	int	11	PRI	Order ID	auto_increment
user_id	varchar	11	Not Null	User ID	
product_id	int	11	Not Null	Product ID	
qty	int	11	Not Null	Quantity	
trx_id	varchar	255	Not Null	Tracking ID	
p_status	varchar	20	Not Null	Status of Package	

order_products

Field Name	Data type	Size	Constants	Description	Extra
order_pro_id	int	11	PRI	Order Product ID	auto_increment
order_id	int	11	MUL	User ID	
product_id	int	11	MUL	Product ID	
qty	int	11	Not Null	Quantity	
amt	int	11	Not Null	Amount	

user_info

Field Name	Data type	Size	Constants	Description	Extra
user_id	**int**	**11**	**PRI**	**Order Product ID**	**auto_increment**
first_name	**varchar**	**100**	**MUL**	**First Name**	
last_name	**varchar**	**100**	**MUL**	**Last Name**	
email	**varchar**	**100**	**Not Null**	**Email**	
password	**varchar**	**100**	**Not Null**	**Password**	
mobile	**varchar**	**100**	**Not Null**	**Mobile Number**	
address1	**varchar**	**100**	**Not Null**	**Address**	
address2	**varchar**	**100**	**Not Null**	**Extra Address**	

The relational schema, along with primary-key and foreign-key constraints is shown below:

admin_info(admin_id , admin_email , admin_name , admin_password)

cart (c_id , user_id , p_id , ip_add , qty)

products (product_id , product_cart , product_brand , product_title ,
product_price , product_desc , product_image ,
product_keywords)

categories (cat_id , cat_title , product_id, foreign key product_id
references products)

brands (brand_id , brand_title , cat_id , product_id ,
foreign key cat_id references catagories)

user_info (user_id , first_name , last_name , password , address ,
mobile)

logs (l_id , date , action, user_id , c_id , foreign key user_id
references user_info, foreign key c_id references cart)

orders (order_id , product_id , qty , user_id , to_id , p_status)

order-products (order_pro_id , qty , amount , order_id , product_id
foreign key order_id references orders , foreign key
product_id references products)

email-info (email_id , email , user_id , user_id , foreign key user_id
references user_info)

order-info (user_id , order_id , f _name , email , card_number , exp
date, prod _ count , cvv , total_amount, card_name,
foreign key user_id references user_info , foreign key
order_id references order_id)

address (state , city , zip , user_id, order _id , foreign key
(user_id , order_id) references order_info)

choose (brand_id , cat_id , foreign key brand_id references brands ,
foreign key cat_id references categories)

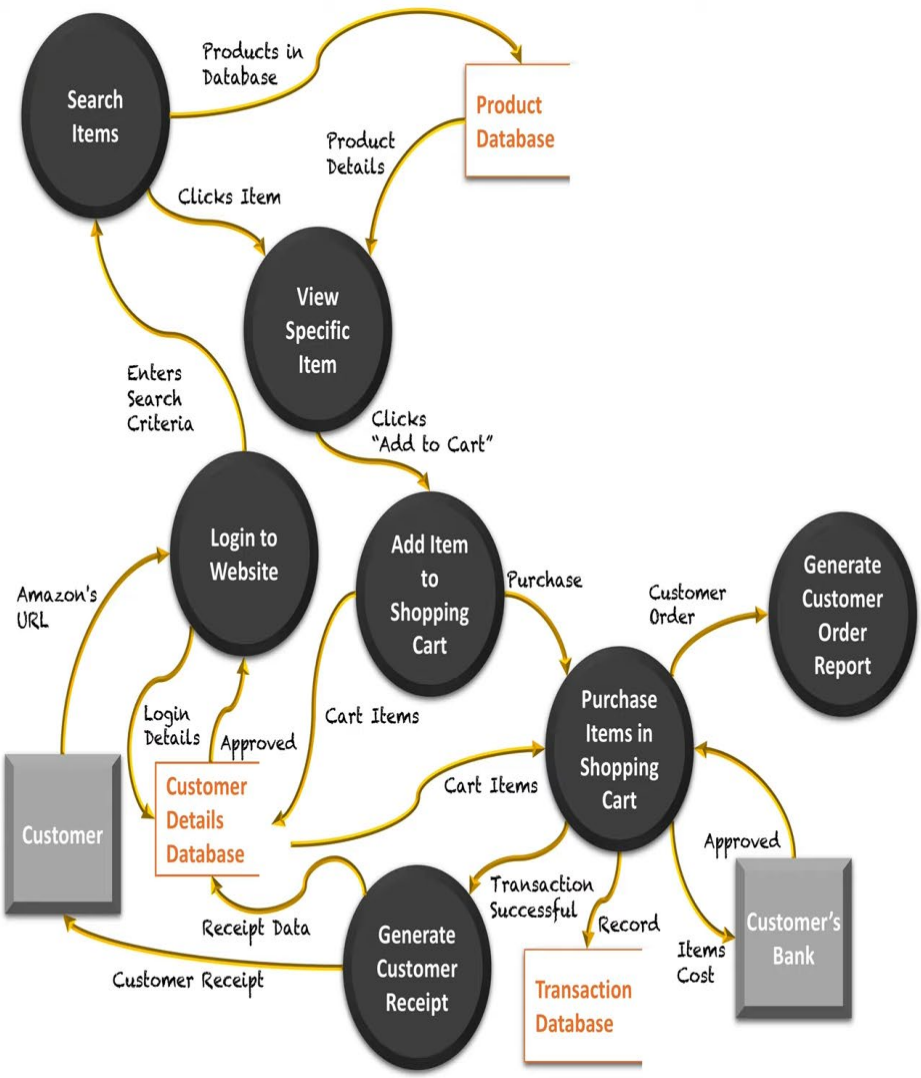
cart _log (l_id , c_id , foreign key l_id references logs
foreign key c_id references cart)

product_categories (cat_id , product_id , foreign key cat_id references
categories , foreign key product_id references products)

order_product_info1 (order_pro_id , product_id foreign key
order_pro_id references order_products,
foreign key product_id references products)

order _product_info (order_pro_id , user_id , order_id , foreign key
order_pro_id references order_products
foreign key(user_id , order_id references
orders_info)

Flowchart:


Search Items
Products in Database
Product Database
Product Details
Clicks Item
View Specific Item
Enters Search Criteria
Clicks "Add to Cart"
Login to Website
Add Item to Shopping Cart
Purchase
Customer Order
Generate Customer Order Report
Amazon's URL
Login Details
Approved
Cart Items
Purchase Items in Shopping Cart
Customer
Customer Details Database
Cart Items
Approved
Transaction Successful
Record
Customer's Bank
Receipt Data
Generate Customer Receipt
Items Cost
Customer Receipt
Transaction Database

NORMALIZATION: -

1st normalization:

Atomic Values

We cannot have two datas in the same column.

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

	user_id	first_name	last_name	email	password	mobile	address1	address2
Edit Copy Delete	12	puneeth	Reddy	puneethreddy951@gmail.com	puneeth	9448121558	Bangalore	Kumbalagodu Karnataka
Edit Copy Delete	15	hemu	ajhgdg	puneethreddy951@gmail.com	346778	536487276	mdnbca	asdmhmhvbv
Edit Copy Delete	16	venky	vs	venkey@gmail.com	1234534	9877654334	snhdgvajfehyfygv	asdjbhfkeur
Edit Copy Delete	19	abhishek	bs	abhishekbs@gmail.com	asdcsdcc	9871236534	bangalore	hassan
Edit Copy Delete	21	prajval	mcta	prajvalmcta@gmail.com	1234545662	202-555-01	bangalore	kumbalagodu
Edit Copy Delete	22	puneeth	v	hemu@gmail.com	1234534	9877654334	snhdgvajfehyfygv	asdjbhfkeur
Edit Copy Delete	23	hemanth	reddy	hemanth@gmail.com	Puneeth@123	9876543234	Bangalore	Kumbalagodu
Edit Copy Delete	24	newuser	user	newuser@gmail.com	puneeth@123	9535688928	Bangalore	Kumbalagodu
Edit Copy Delete	25	otheruser	user	otheruser@gmail.com	puneeth@123	9535688928	Bangalore	Kumbalagodu

We can see here ,we have separated the 2 address in 2 separate column.

Therefore,we can say that we have fulfilled the 1st Normalization.

2nd normalization:

As we fulfilled the 1st normalization then we can move to 2nd normalization.

```
SELECT * FROM `order_products`
```

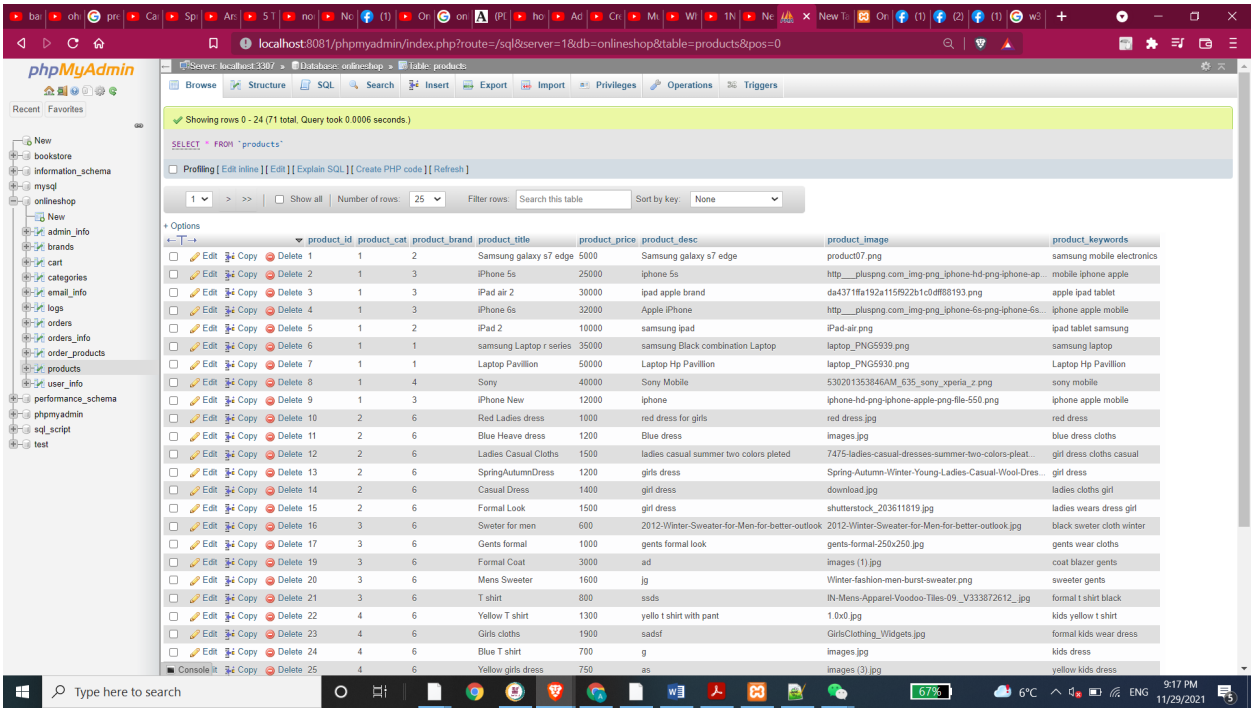
Profiling [Edit inline] [Edit] [Explain SQL] [Create

Show all | Number of rows: 25 Filter rows: Search this table

- Options

	order_pro_id	order_id	product_id	qty	amt
Edit Copy Delete	73	1	1	1	5000
Edit Copy Delete	74	1	4	2	64000
Edit Copy Delete	75	1	8	1	40000

For example, in the above picture we can see there is column named product_id which will connect with another table name product. Therefore,we have to create a separate table named products for that purpose.If we don't do that we can see redundancy and partial dependency.

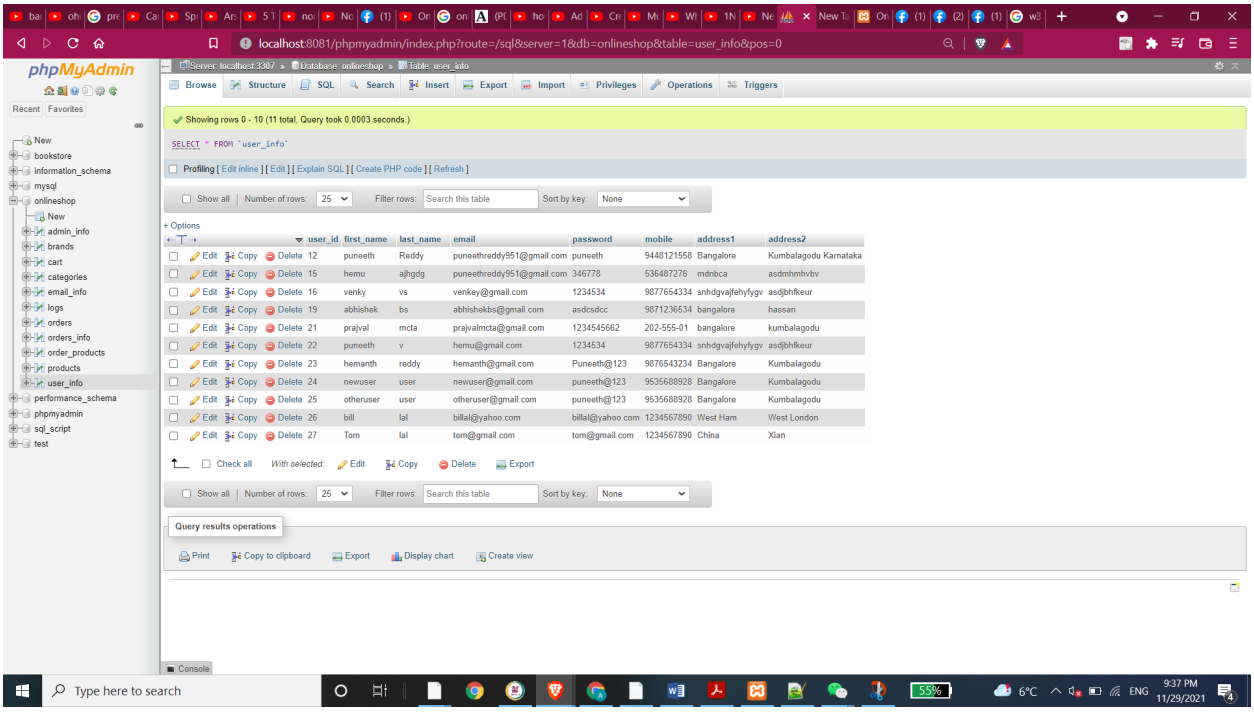


By creating a separate of table containing all the items of product available. We create 2nd normalization.By using 2nd normalization, we can solve the problem of update abnomaly.We can see at the above there is no partial dependency and the columns depends on whole key.

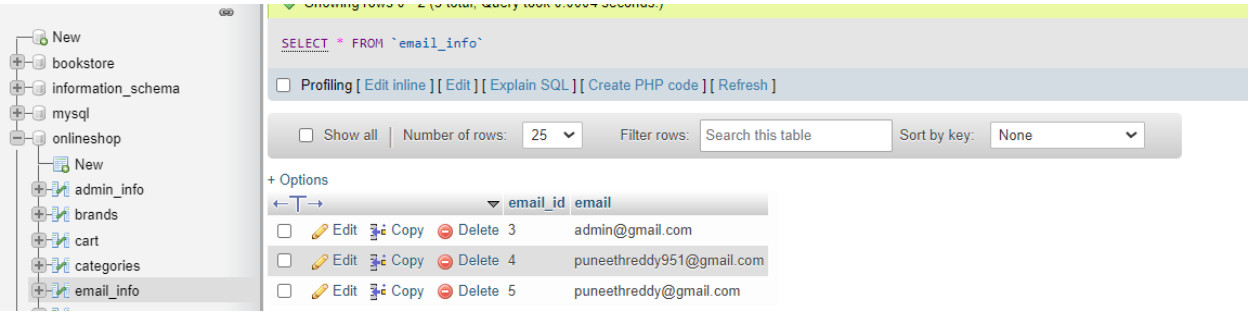
Therefore,we can say that we have fulfilled the 2nd Normalization.

3rd normalization:

As we fulfilled the 1st normalization and 2nd normalization then we can move to 3rd normalization.



As we can see ,a user or customer can more than one email.If we add their email address each time in the user table then it will redundancy and each time they buy anything we have to the copy their email address in the table .



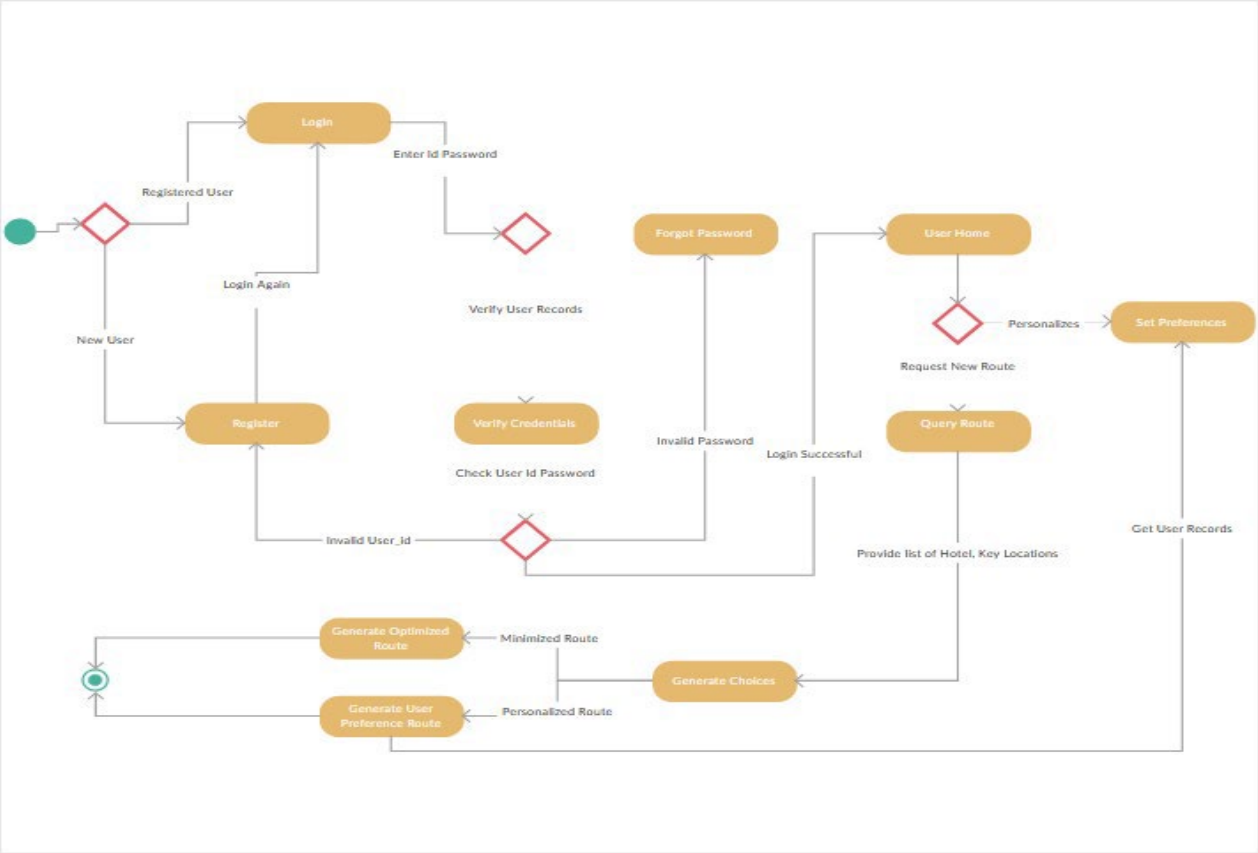
By making another table where information about Emails are stored.By doing that,we solve the update anamoly problem.Each time value of email will be stored in this table and won't store in the main table which would look messy.

Hence,we can say that, all of it's columns are not transitively dependent on the primary key

Therefore,we can say that we have fulfilled the 3rd Normalization.

Navigation Diagram:-

Login :-



Section :4 Pictures

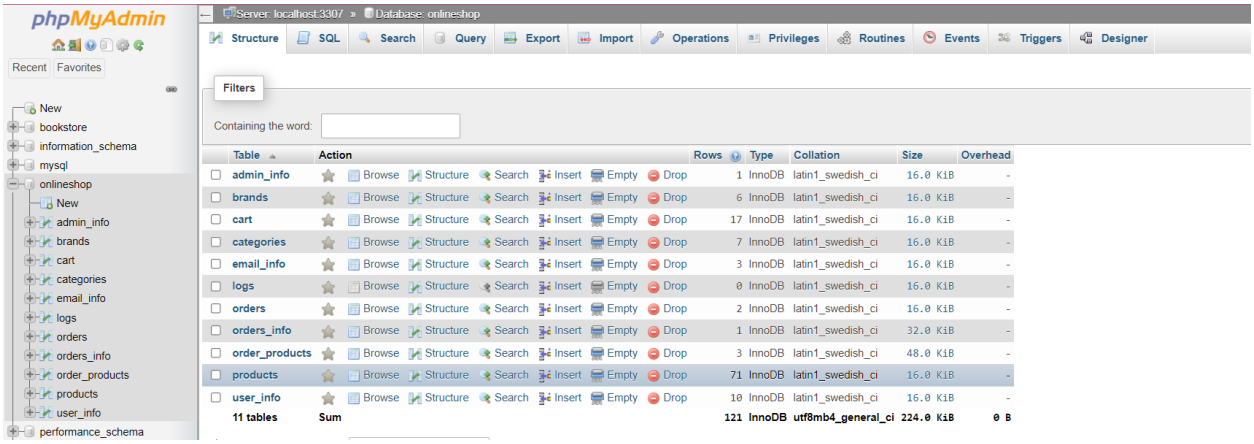


Fig:Online shopping system php database table

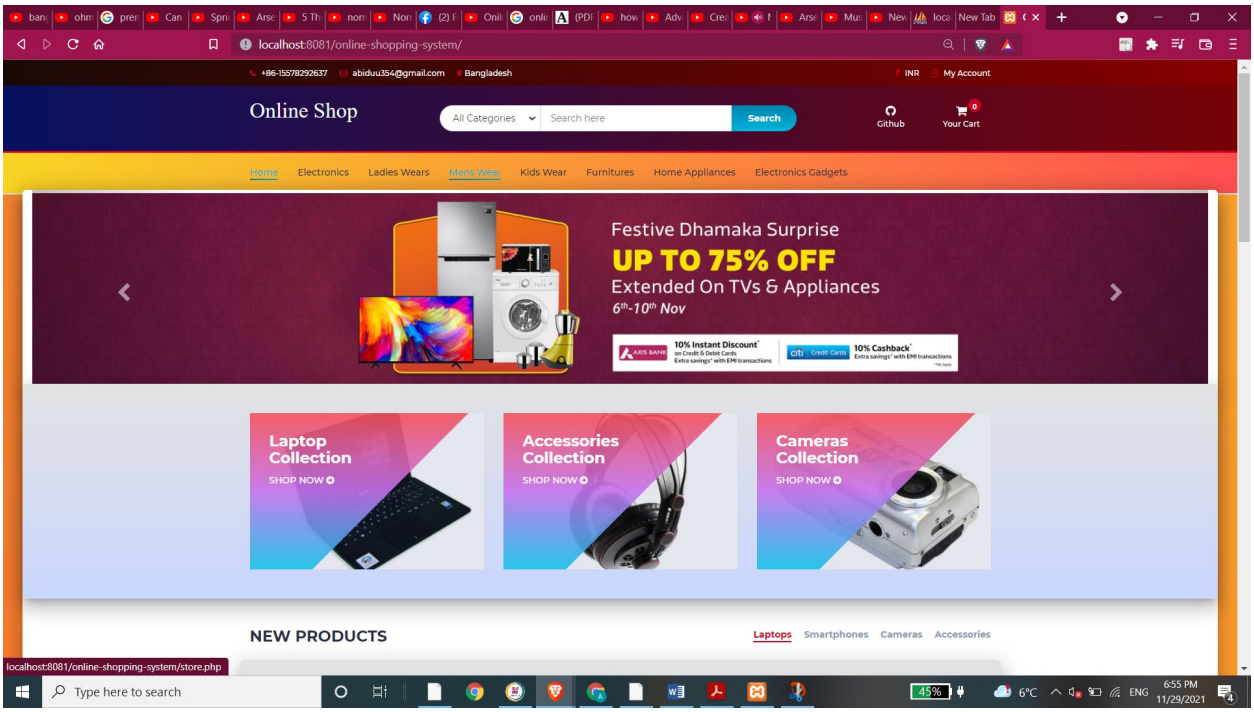


Fig:Homepage of the website

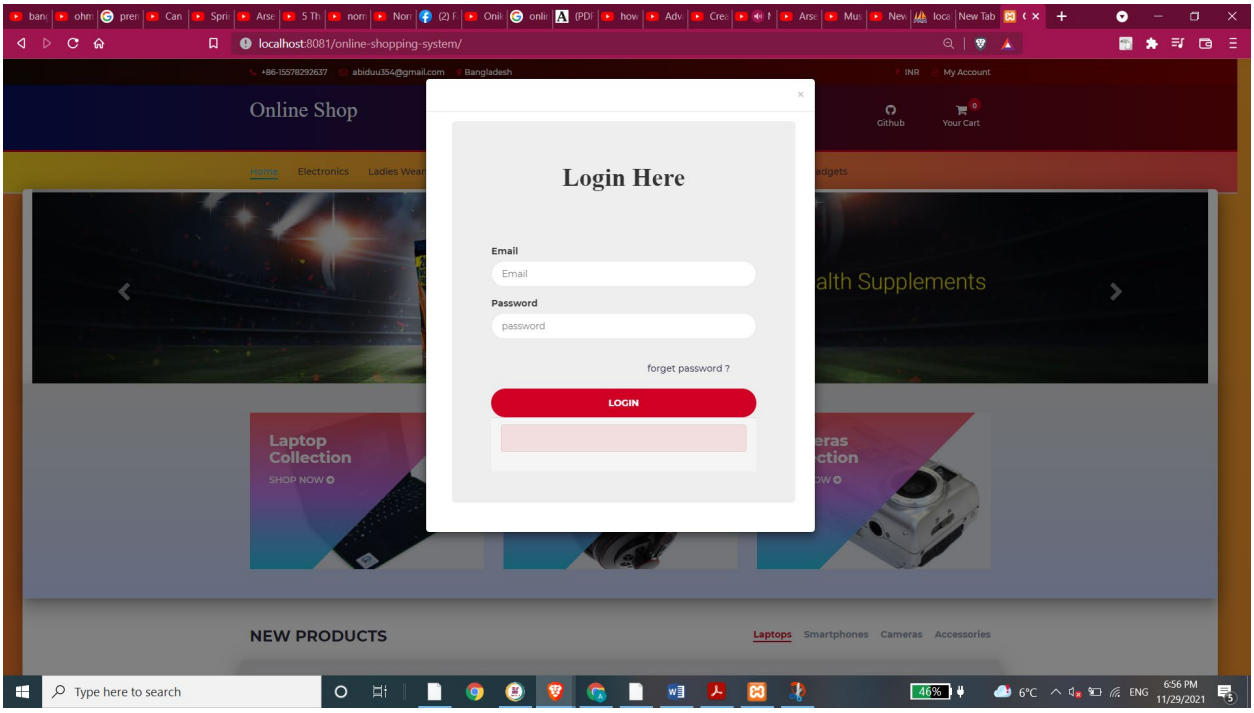


Fig:Login page of the website

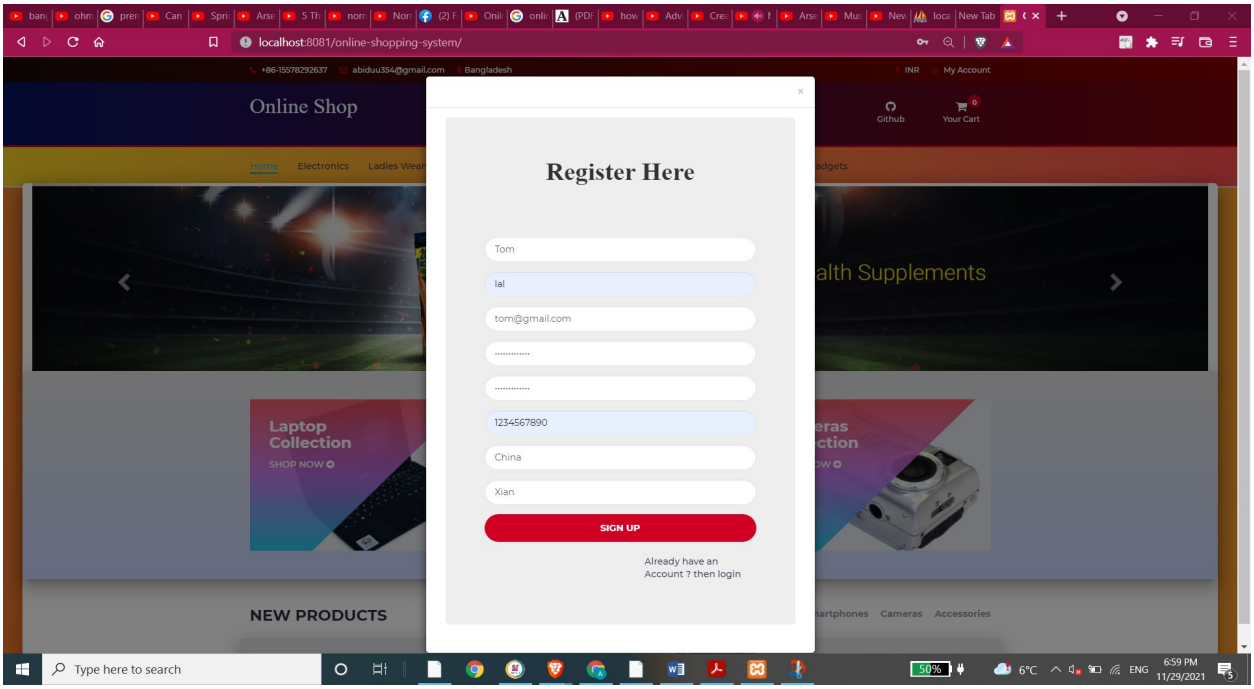


Fig: Registering page of the website

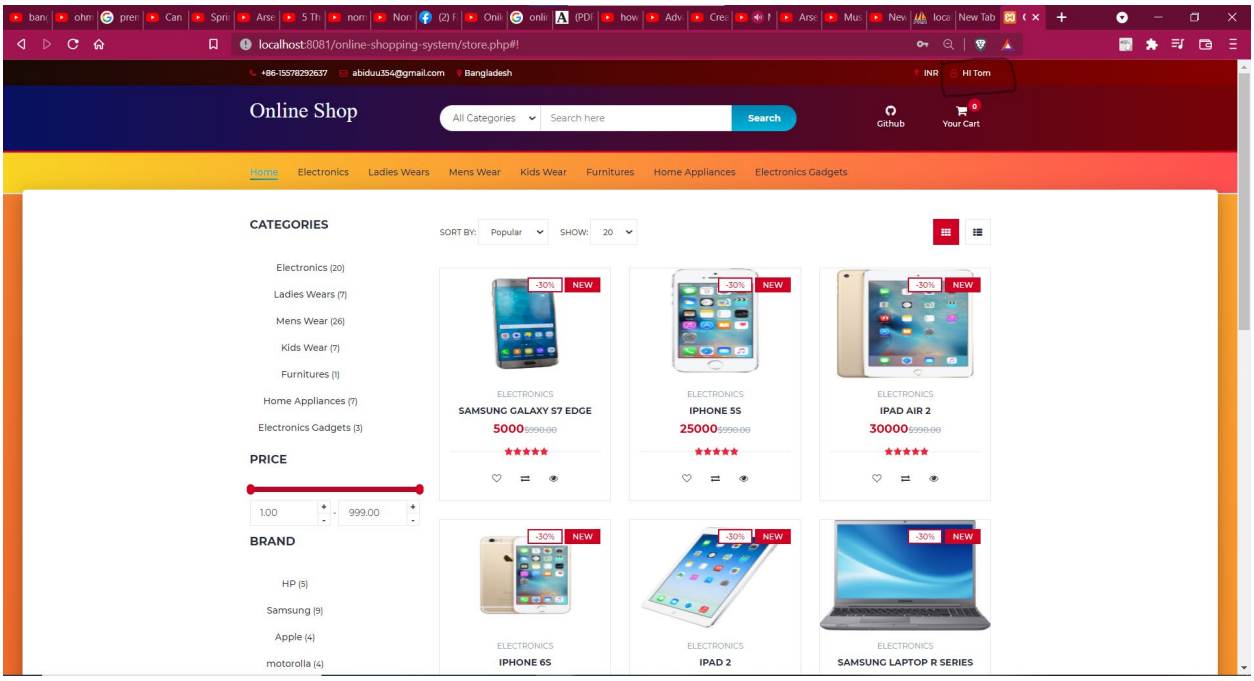
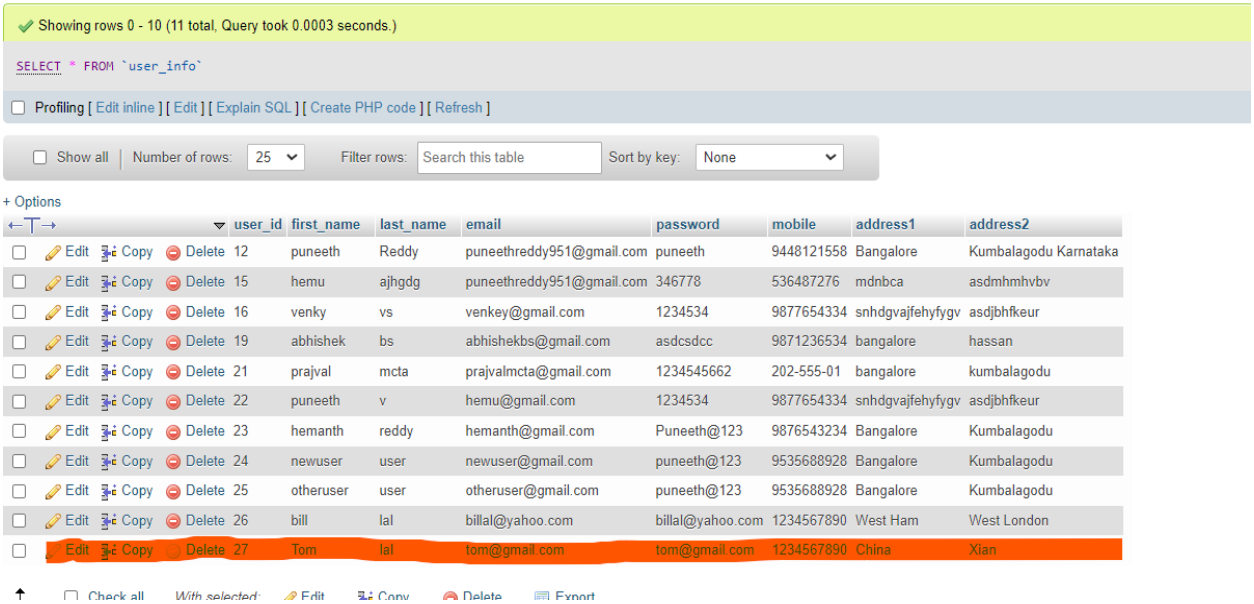


Fig: New user logged in the website



Showing rows 0 - 10 (11 total, Query took 0.0003 seconds.)

```
SELECT * FROM `user_info`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

	user_id	first_name	last_name	email	password	mobile	address1	address2
Edit Copy Delete	12	puneeth	Reddy	puneethreddy951@gmail.com	puneeth	9448121558	Bangalore	Kumbalagodu Karnataka
Edit Copy Delete	15	hemu	ajhgdg	puneethreddy951@gmail.com	346778	536487276	mdnbca	asdmhmhvbv
Edit Copy Delete	16	venky	vs	venkey@gmail.com	1234534	9877654334	snhdgvajfehyfygv	asdjbhfkeur
Edit Copy Delete	19	abhishek	bs	abhishekbs@gmail.com	asdcsdcc	9871236534	bangalore	hassan
Edit Copy Delete	21	prajval	mcta	prajvalmcta@gmail.com	1234545662	202-555-01	bangalore	kumbalagodu
Edit Copy Delete	22	puneeth	v	hemu@gmail.com	1234534	9877654334	snhdgvajfehyfygv	asdjbhfkeur
Edit Copy Delete	23	hemanth	reddy	hemanth@gmail.com	Puneeth@123	9876543234	Bangalore	Kumbalagodu
Edit Copy Delete	24	newuser	user	newuser@gmail.com	puneeth@123	9535688928	Bangalore	Kumbalagodu
Edit Copy Delete	25	otheruser	user	otheruser@gmail.com	puneeth@123	9535688928	Bangalore	Kumbalagodu
Edit Copy Delete	26	bill	lal	billal@yahoo.com	billal@yahoo.com	1234567890	West Ham	West London
Edit Copy Delete	27	Tom	lal	tom@gmail.com	tom@gmail.com	1234567890	China	Xian

Check all With selected: Edit Copy Delete Export

Fig: New user created in database

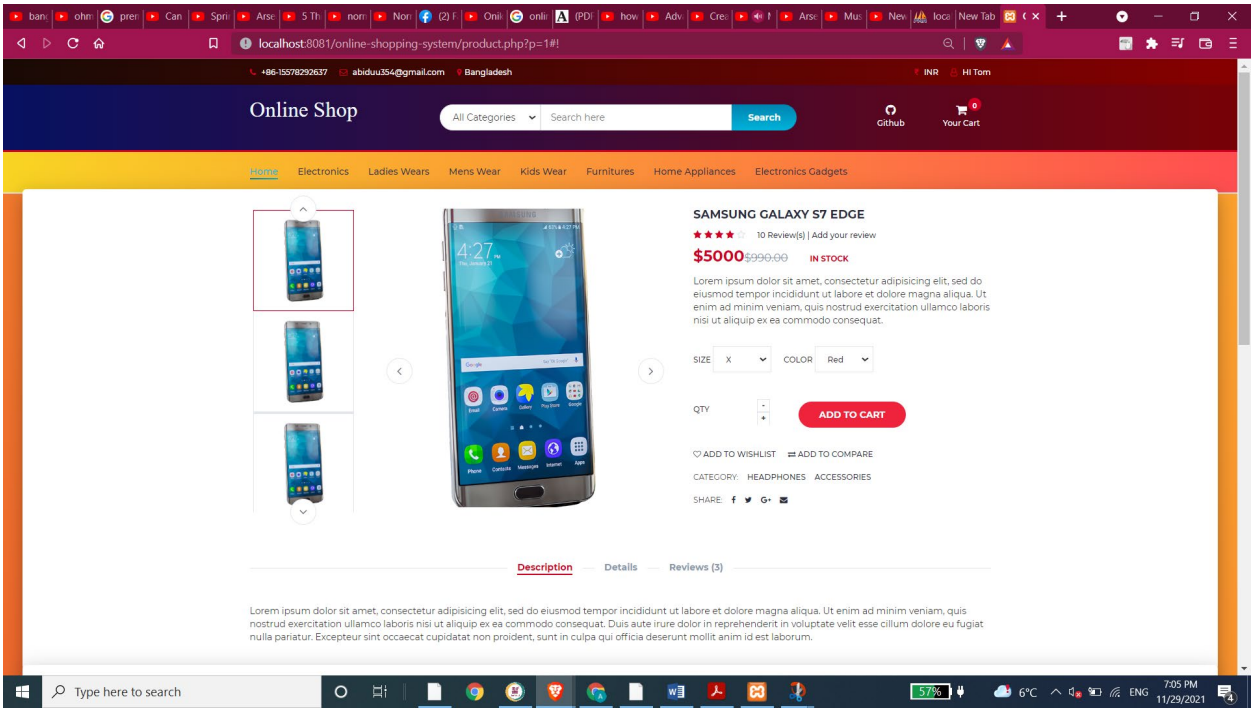
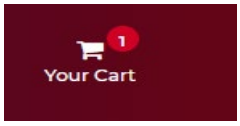


Fig: Detailed information of the products

If the customer press

that product is added to chart 

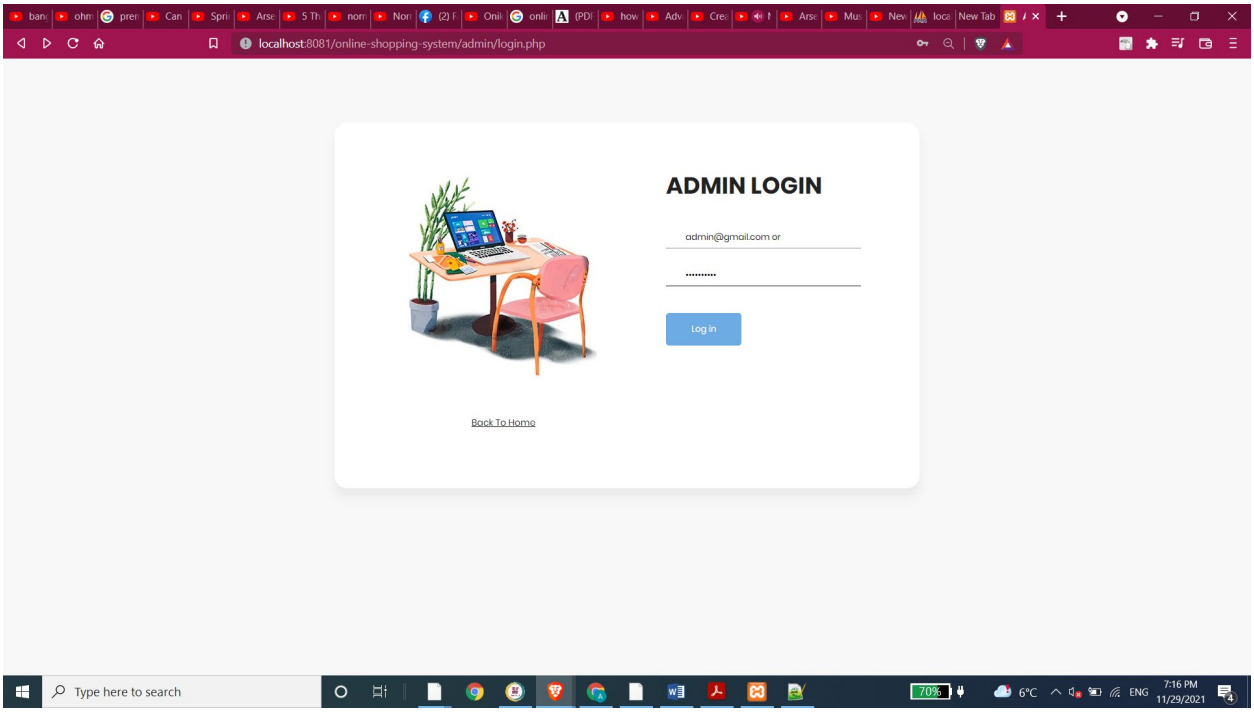


Fig:Admin login page

Admin login details :

Email=admin@gmail.com or username = admin

Password=123456789

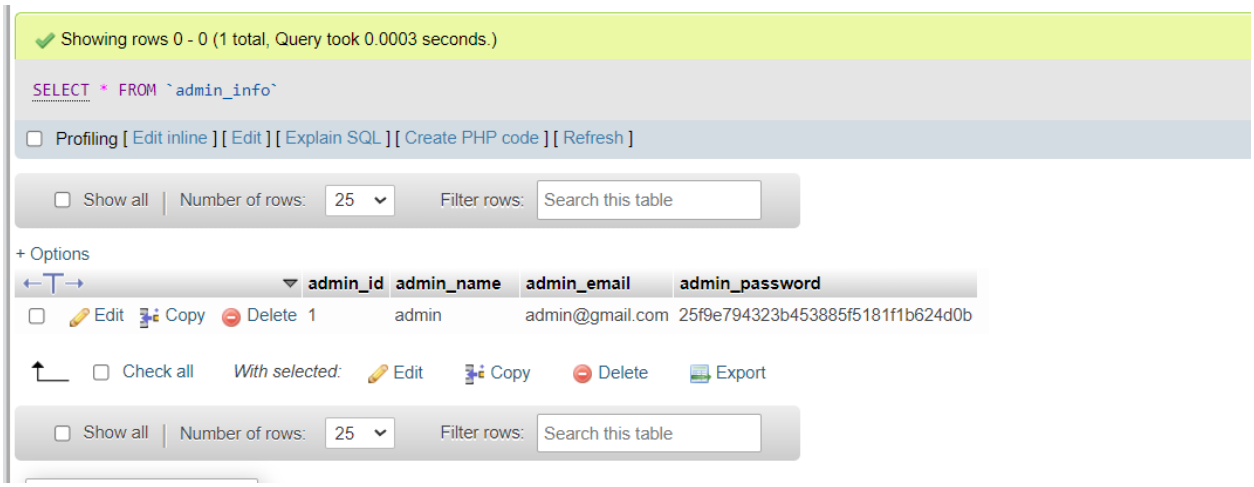


Fig:Admin php database table

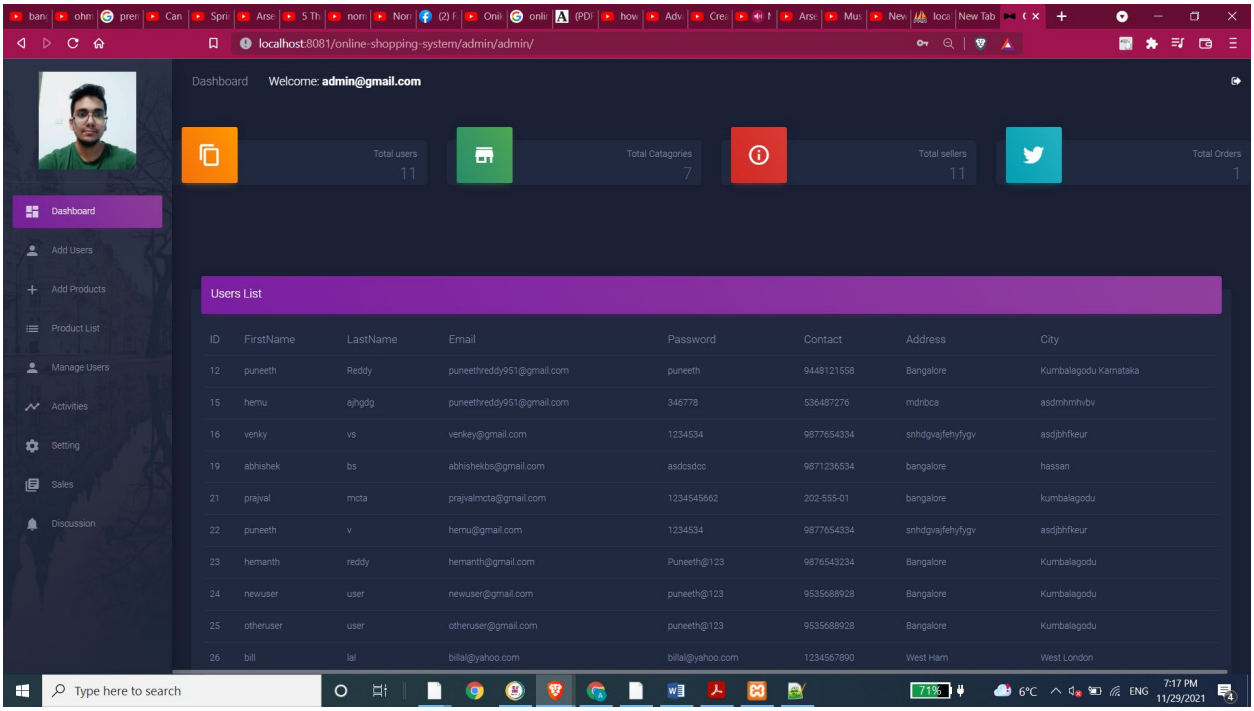


Fig:Homepage of the Admin page

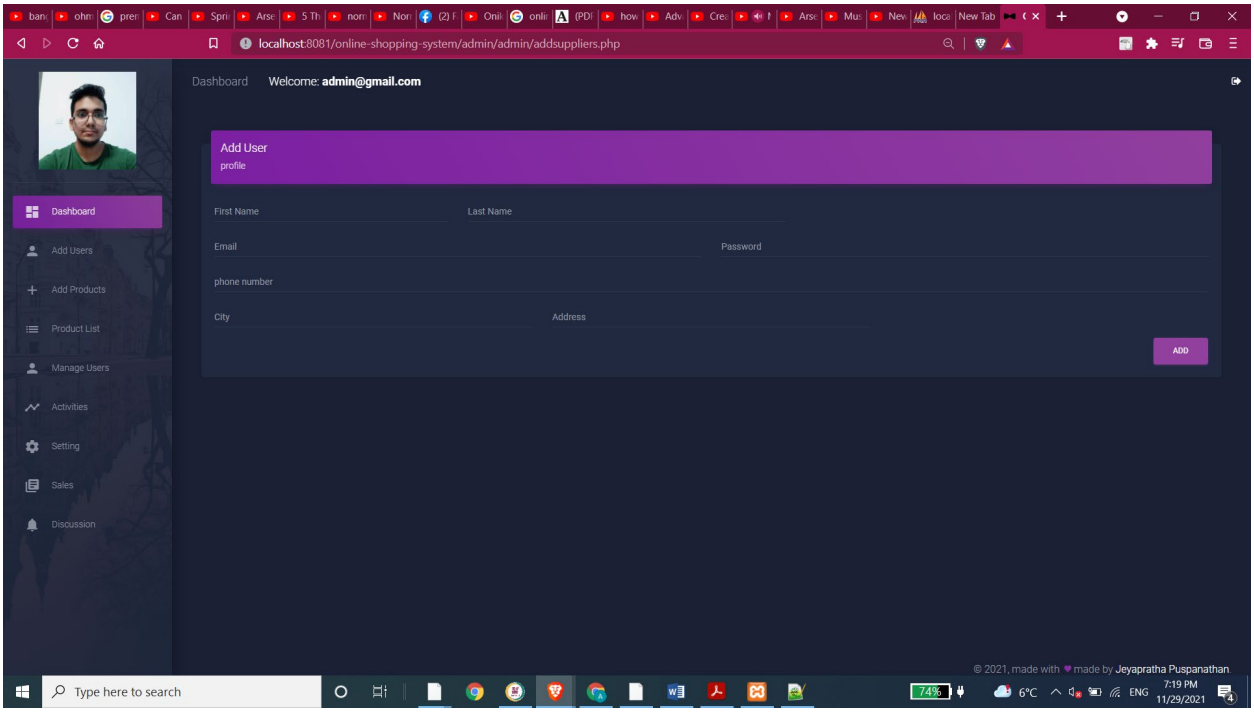


Fig:Add user from Admin page

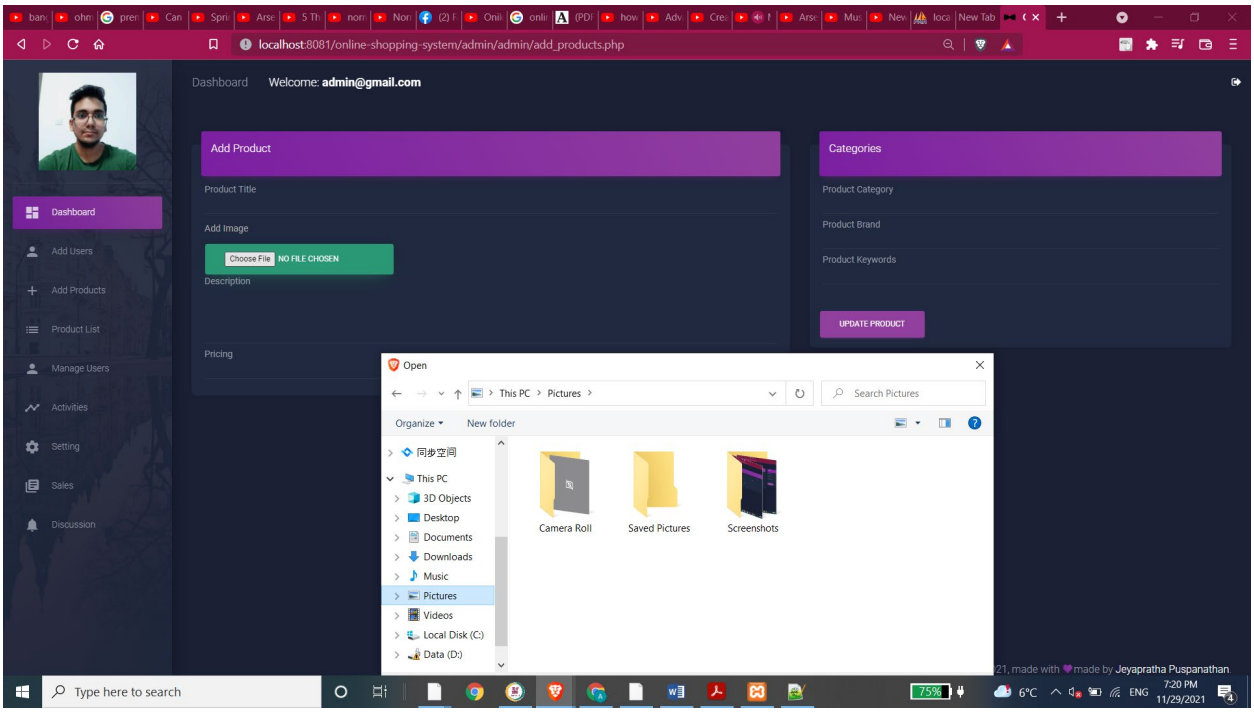


Fig:Add products from Admin page

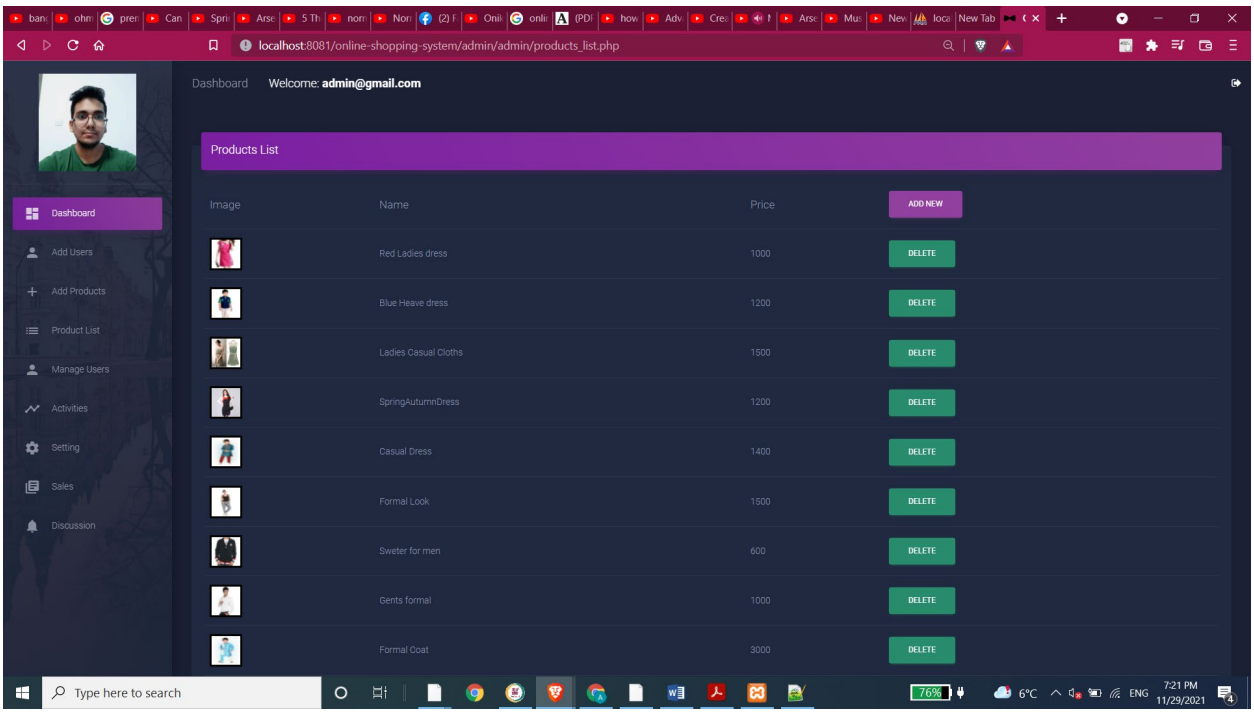


Fig: Previewing Product list from Admin page

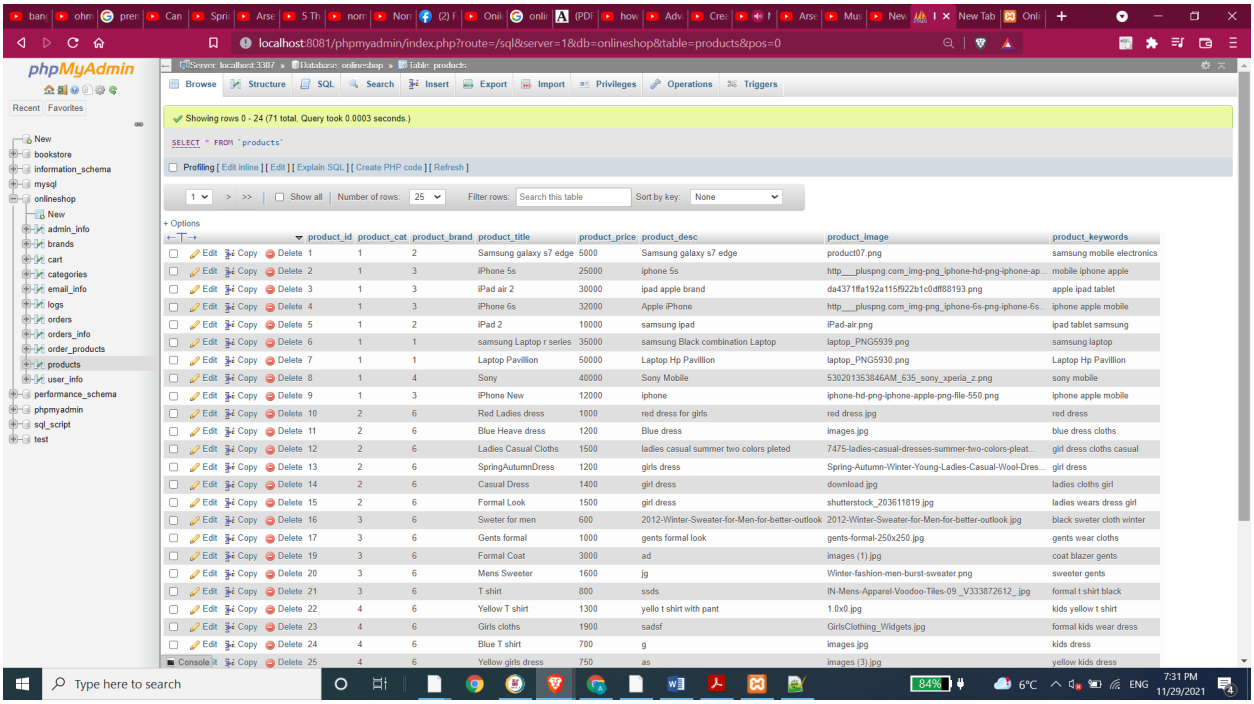


Fig: Product list in database

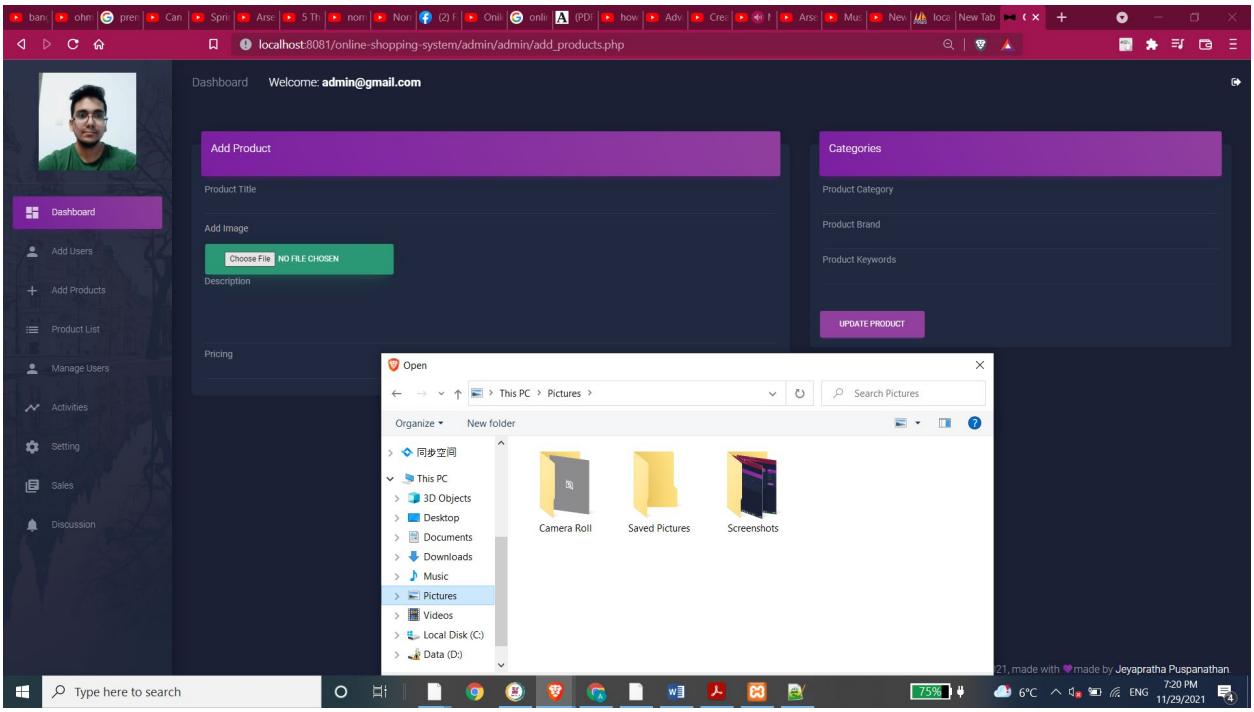


Fig: Adding or removing product list from Admin page

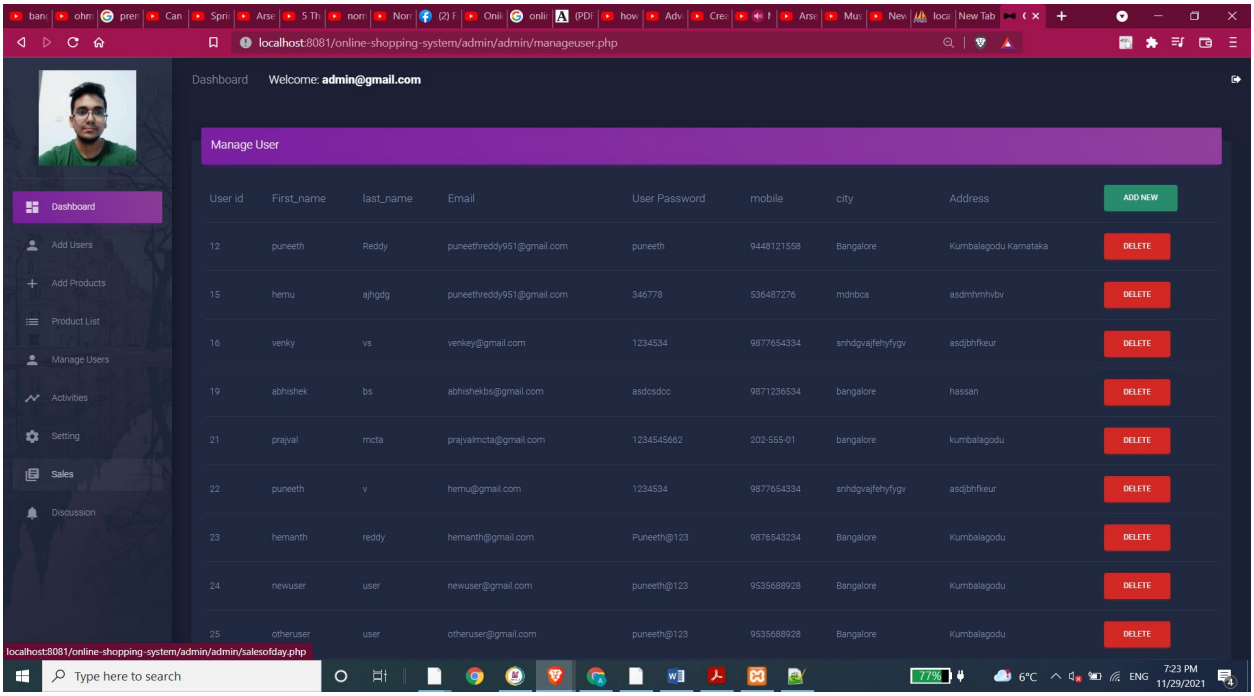


Fig: Managing user list from Admin page

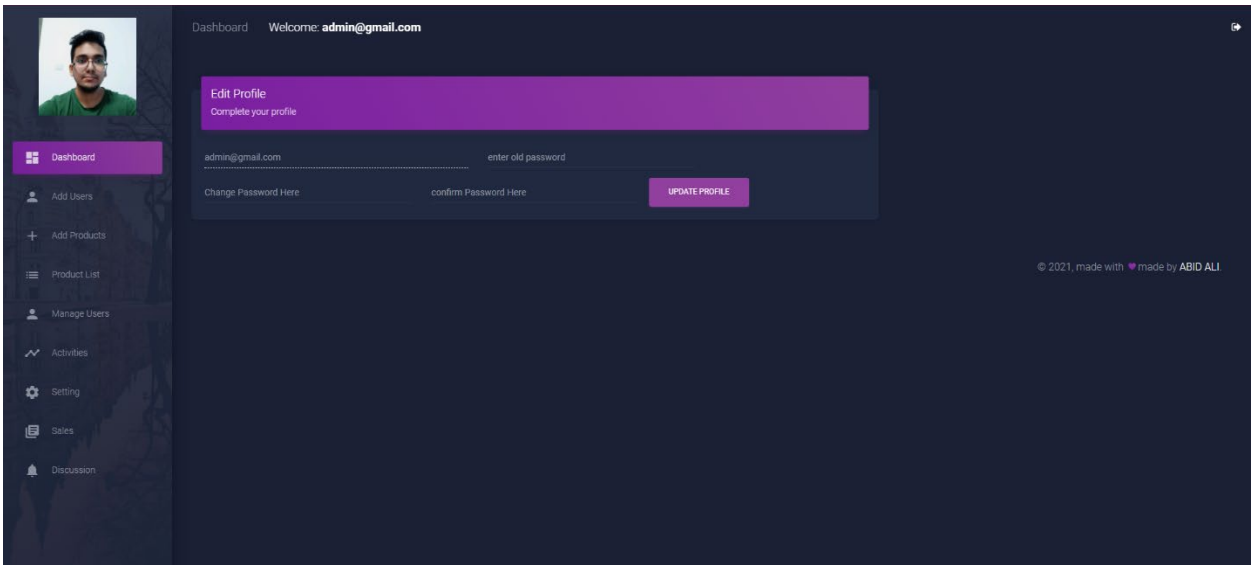


Fig: Change admin from Admin page

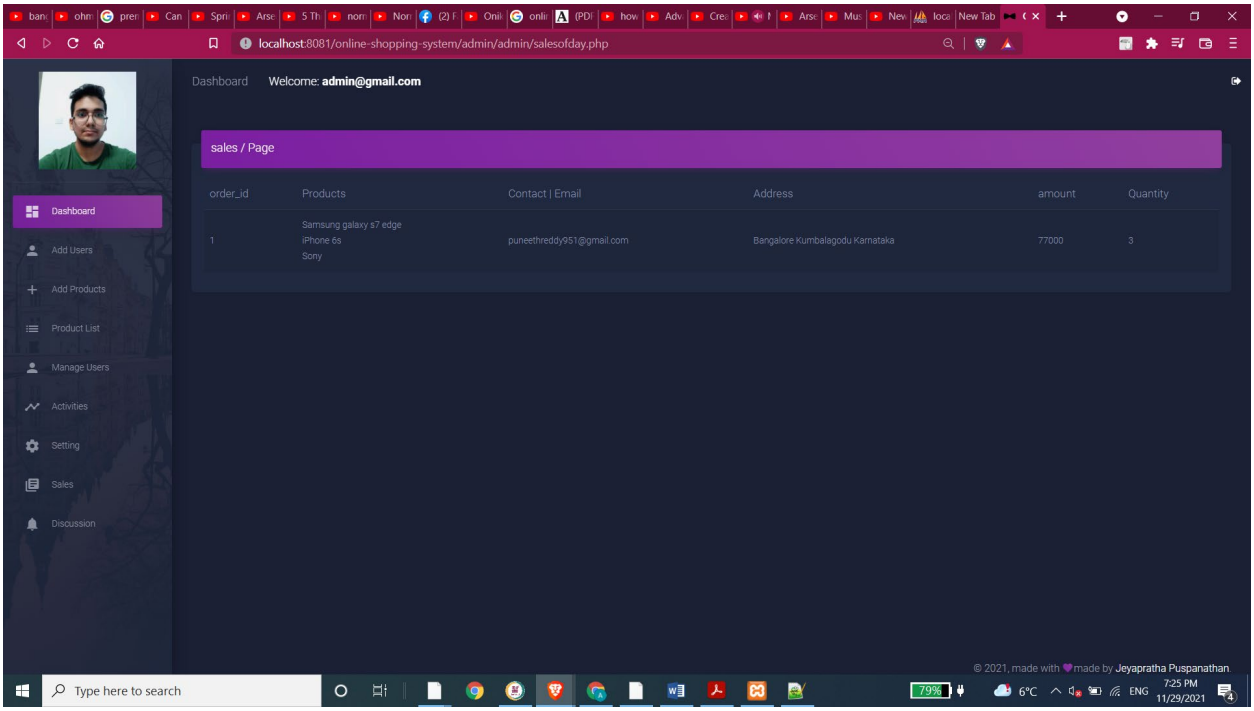


Fig: Admin view of sales record

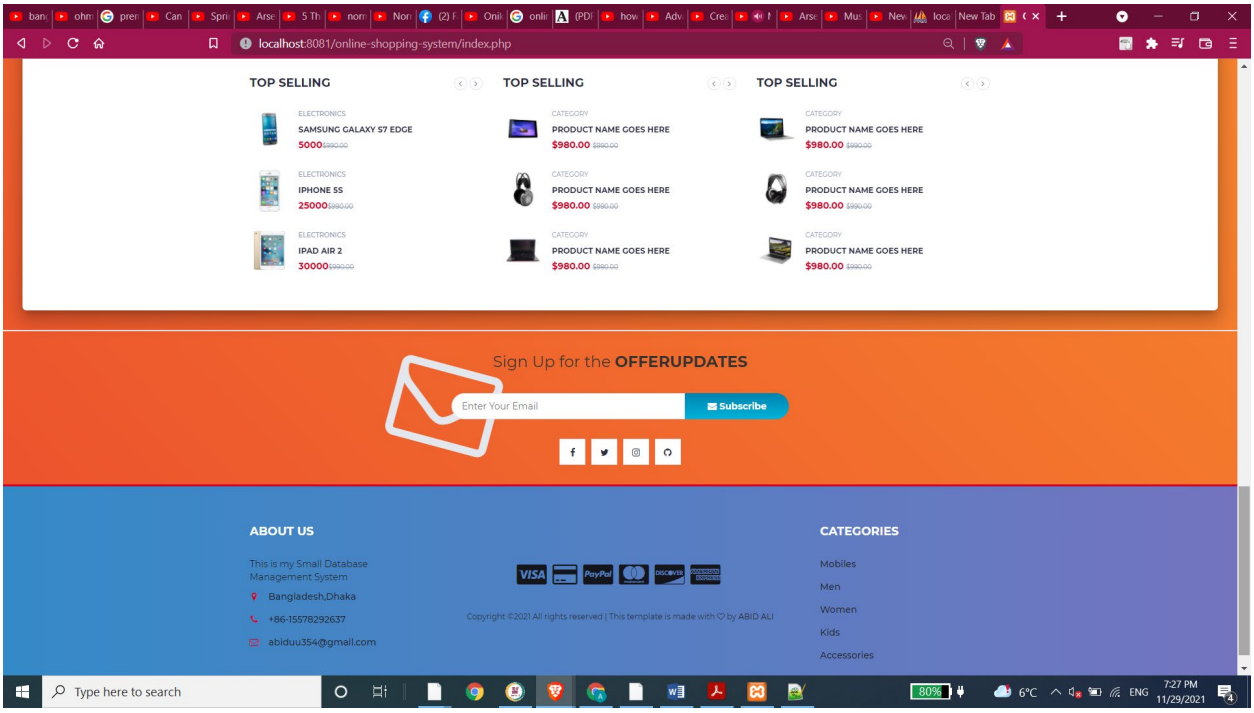


Fig: Homepage of the website(Bottom part)

Section : 5 Configurations

The process of configuring XAMPP and

1. Install XAMPP .

2. Open XAMPP Control panal and start [apache] and [mysql] .

3.Change the port value according to the video or report file description

4. Copy the online-shopping-system files in C:\\xampp\htdocs\.

5. open link http://localhost:8081/phpmyadmin

6. click on new at side navbar.

7. give a database name as (onlineshop) hit on create button.

8. after creating database name click on import.

9. browse the file in directory[online-shopping-system/database/onlineshop.sql]

10. after importing successfully.

11. open any browser and type http://localhost:8081/online-shopping-system/

12.I would recommend to use the link http://localhost:8081/online-shopping-system/ rather than installing the executable file and running.

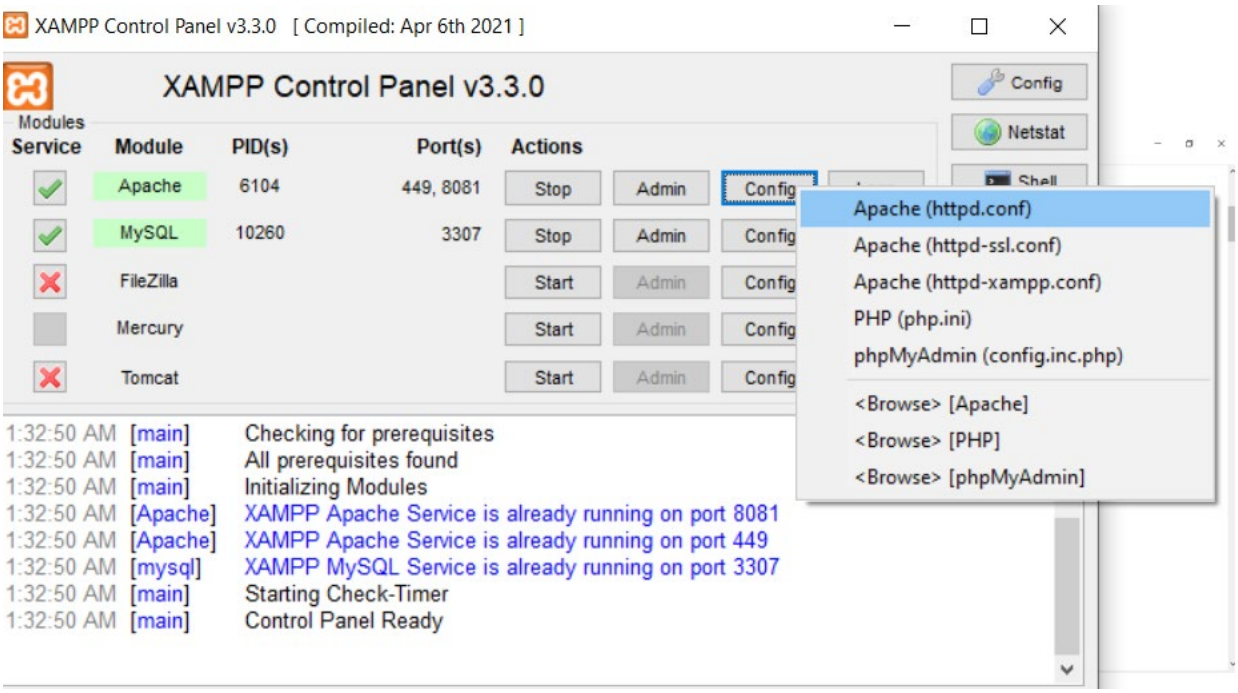
13.After successfully configuring

14. first register and then login

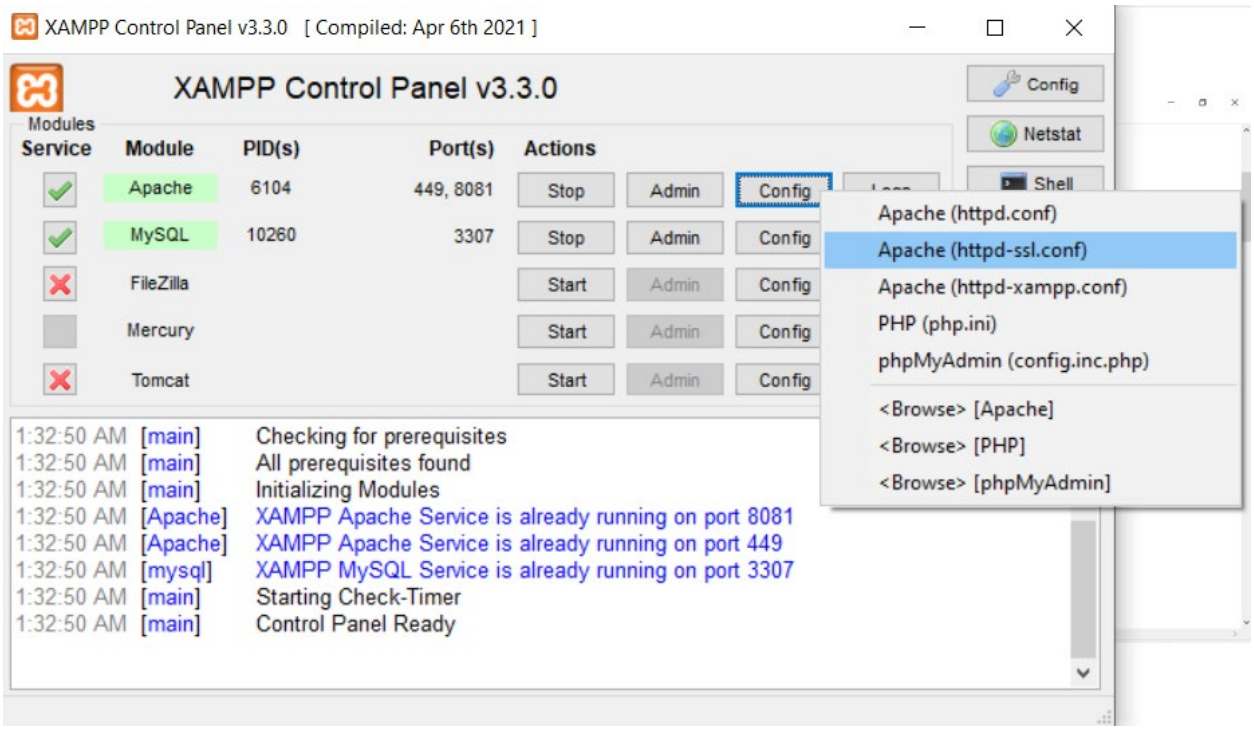
15. admin login details Email=admin@gmail.com or username = admin and Password=123456789.

Note:Watch the video for clear understanding of configuration

Apache setting configuration



We change listen the port to 8081 (Apache)



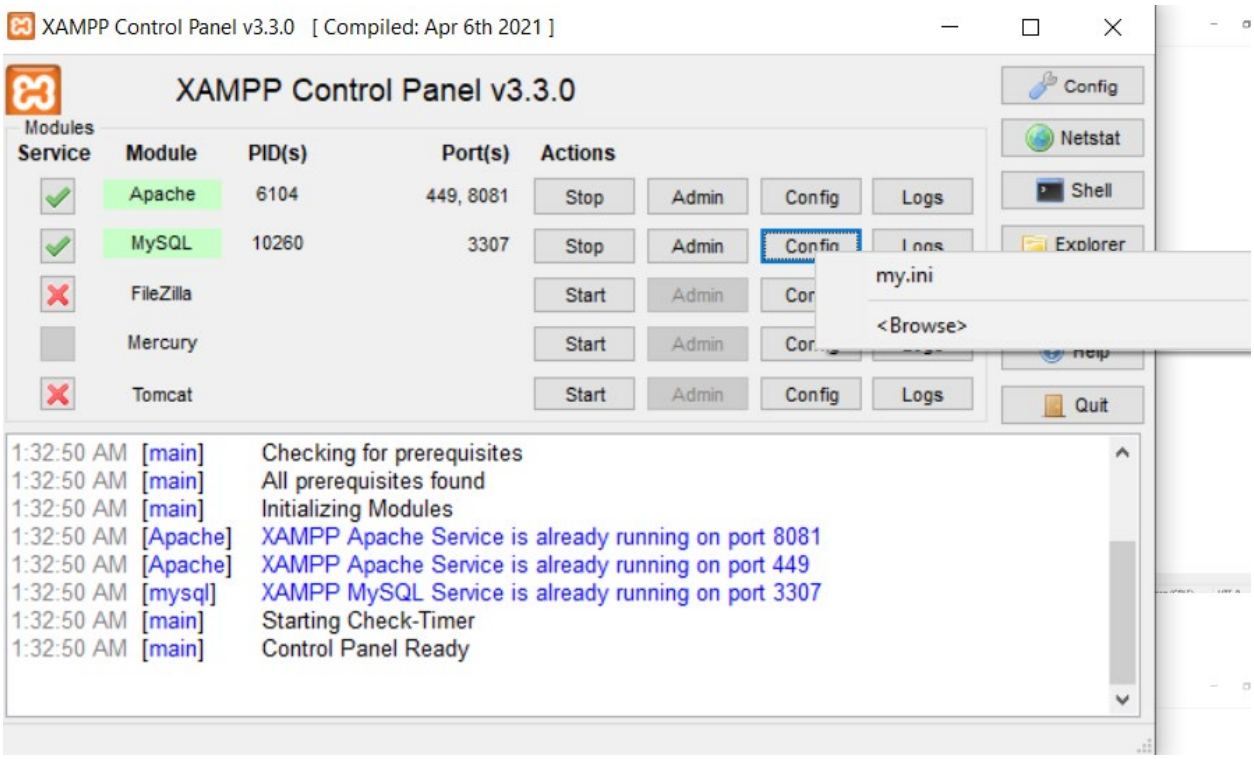
```
# block. So, if available, use this one instead. Read the mod_ssl User
# Manual for more details.
#
#SSLRandomSeed startup file:/dev/random  512
#SSLRandomSeed startup file:/dev/urandom 512
#SSLRandomSeed connect file:/dev/random  512
#SSLRandomSeed connect file:/dev/urandom 512


#
# When we also provide SSL we have to listen to the
# standard HTTP port (see above) and to the HTTPS port
#
Listen 449

##
##  SSL Global Context
##
##  All SSL configuration in this context applies both to
##  the main server and all SSL-enabled virtual hosts.
##

#   SSL Cipher Suite:
#   List the ciphers that the client is permitted to negotiate,
#   and that httpd will negotiate as the client of a proxied server.
#   See the OpenSSL documentation for a complete list of ciphers, and
#   ensure these follow appropriate best practices for this deployment.
#   httpd 2.2.30, 2.4.13 and later force-disable aNULL, eNULL and EXP ciphers,
#   while OpenSSL disabled these by default in 0.9.8zf/1.0.0r/1.0.1m/1.0.2a.
SSLCipherSuite HIGH:MEDIUM:!MD5:!RC4:!3DES
SSLProxyCipherSuite HIGH:MEDIUM:!MD5:!RC4:!3DES

#  By the end of 2016, only TLSv1.2 ciphers should remain in use.
#  Older ciphers should be disallowed as soon as possible, while the
#  kRSA ciphers do not offer forward secrecy.  These changes inhibit
#  older clients (such as IE6 SP2 or IE8 on Windows XP, or other legacy
#  non-browser tooling) from successfully connecting.
#
#  To restrict mod_ssl to use only TLSv1.2 ciphers, and disable
#  those protocols which do not support forward secrecy, replace
#  the SSLCipherSuite and SSLProxyCipherSuite directives above with
#  the following two directives, as soon as practical.
```

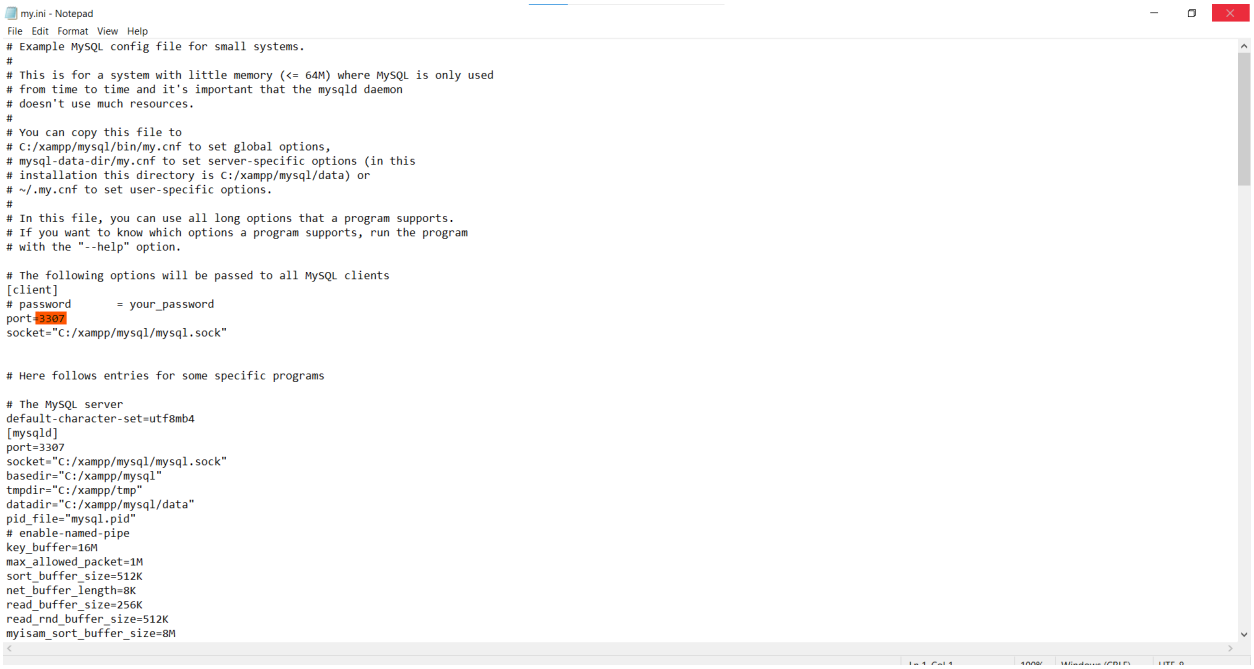
We change listen the port to 449(Apache)

```
#   Define a relatively small cache for OCSP Stapling using
#   the same mechanism that is used for the SSL session cache
#   above.  If stapling is used with more than a few certificates,
#   the size may need to be increased.  (AH01929 will be logged.)
#SSLStaplingCache "shmcb:${SRVROOT}/logs/ssl_stapling(32768)"

#   Seconds before valid OCSP responses are expired from the cache
#SSLStaplingStandardCacheTimeout 3600

#   Seconds before invalid OCSP responses are expired from the cache
#SSLStaplingErrorCacheTimeout 600

##
## SSL Virtual Host Context
##

<VirtualHost _default_:449>

#   General setup for the virtual host
DocumentRoot "C:/xampp/htdocs"
ServerName www.example.com:449
ServerAdmin admin@example.com
ErrorLog "C:/xampp/apache/logs/error.log"
TransferLog "C:/xampp/apache/logs/access.log"

#   SSL Engine Switch:
#   Enable/Disable SSL for this virtual host.
SSLEngine on

#   Server Certificate:
#   Point SSLCertificateFile "conf/ssl.crt/server.crt"
#   the certificate is encrypted, then you will be prompted for a
#   pass phrase.  Note that a kill -HUP will prompt again.  Keep
#   in mind that if you have both an RSA and a DSA certificate you
#   can configure both in parallel (to also allow the use of DSA
#   ciphers, etc.)
#   Some ECC cipher suites (http://www.ietf.org/rfc/rfc4492.txt)
#   require an ECC certificate which can also be configured in
#   parallel.
SSLCertificateFile "conf/ssl.crt/server.crt"
#SSLCertificateFile "conf/ssl.crt/server.crt"
#SSLCertificateFile "conf/ssl.crt/server.crt"
```

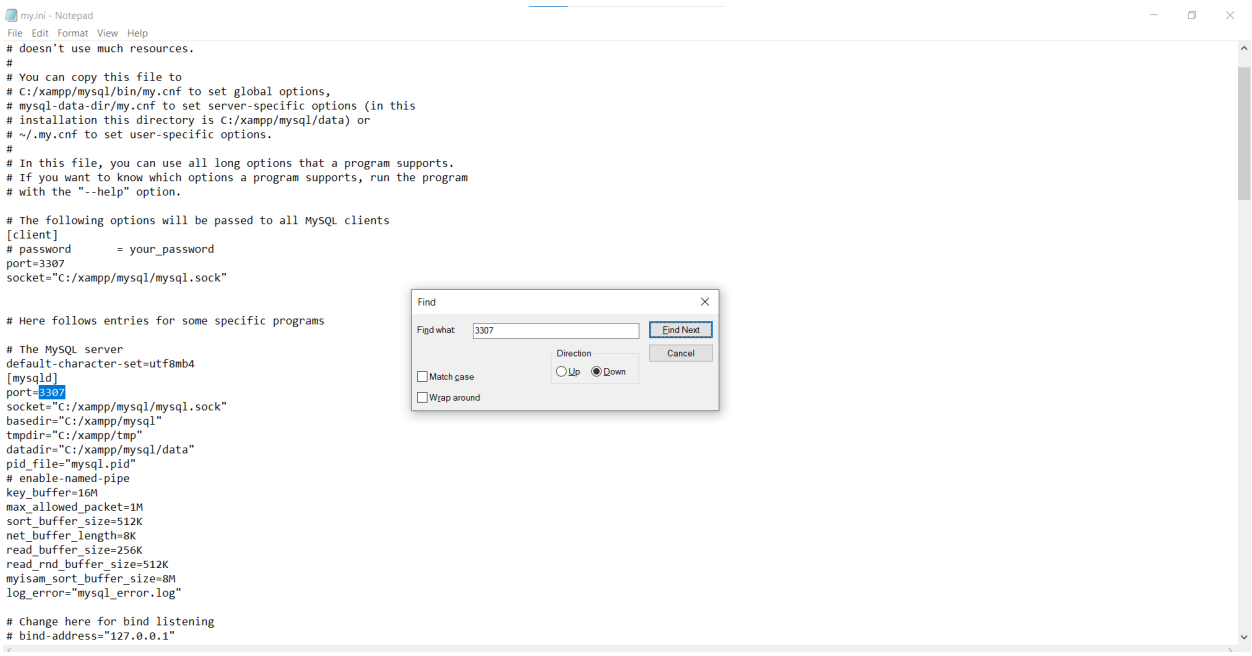
We change listen the port to 449(Apache)

Mysql setting configuration

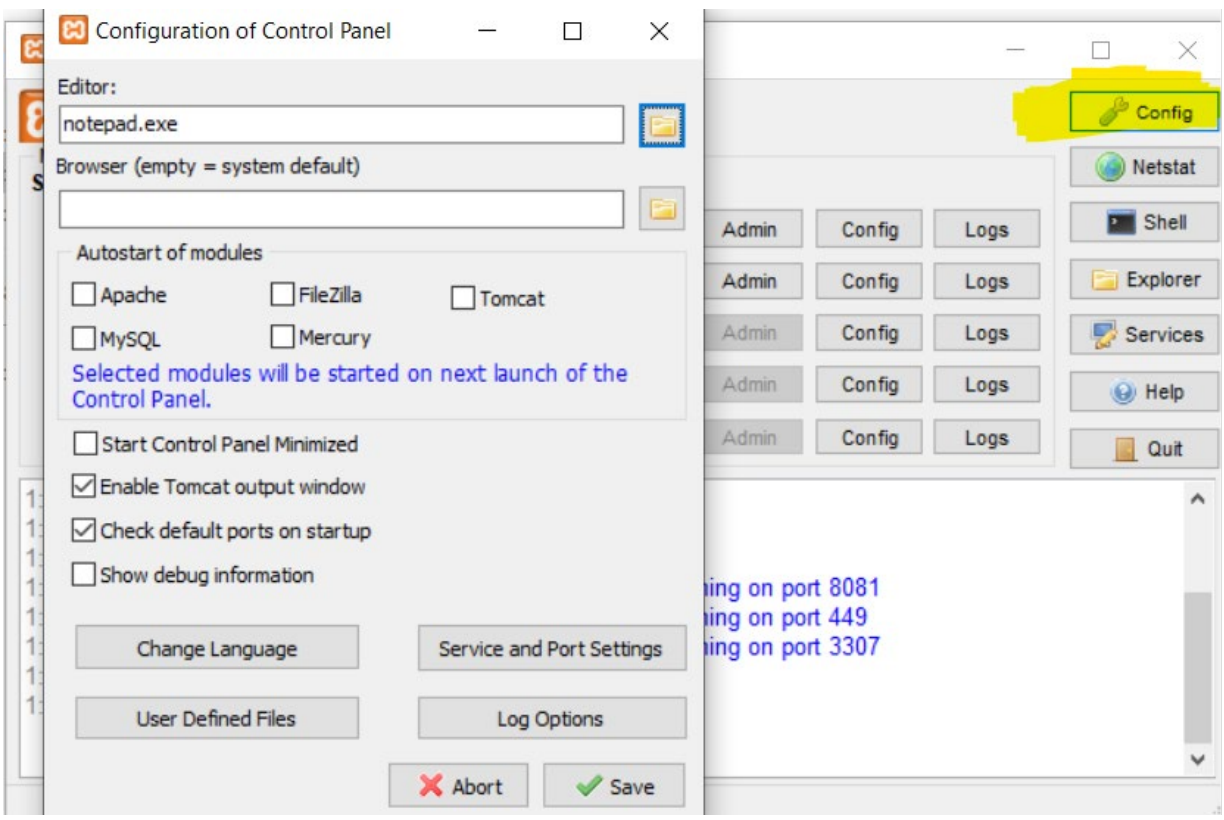


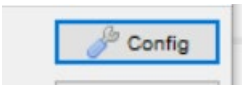


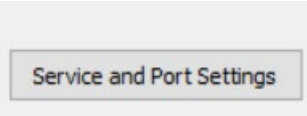
We change listen the port to 3307(Mysql)

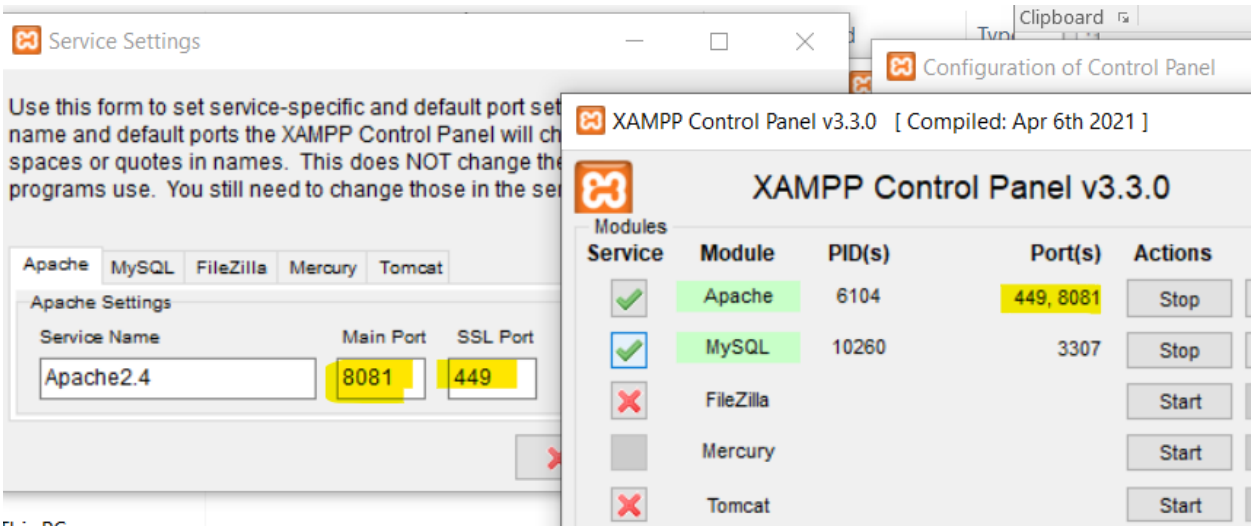


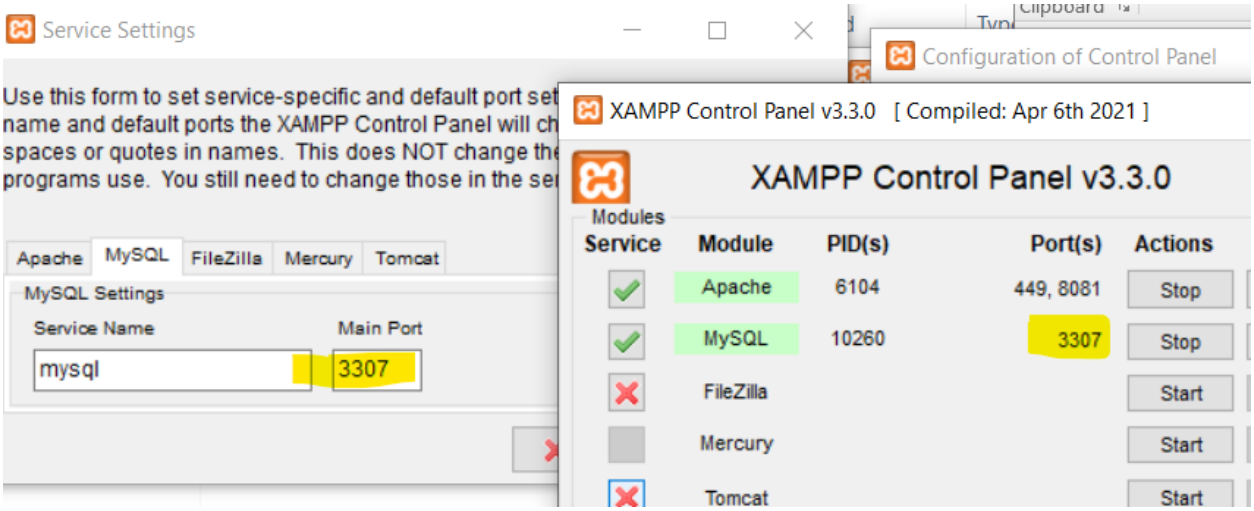
We change listen the port to 3307(Mysql)



Press this button 

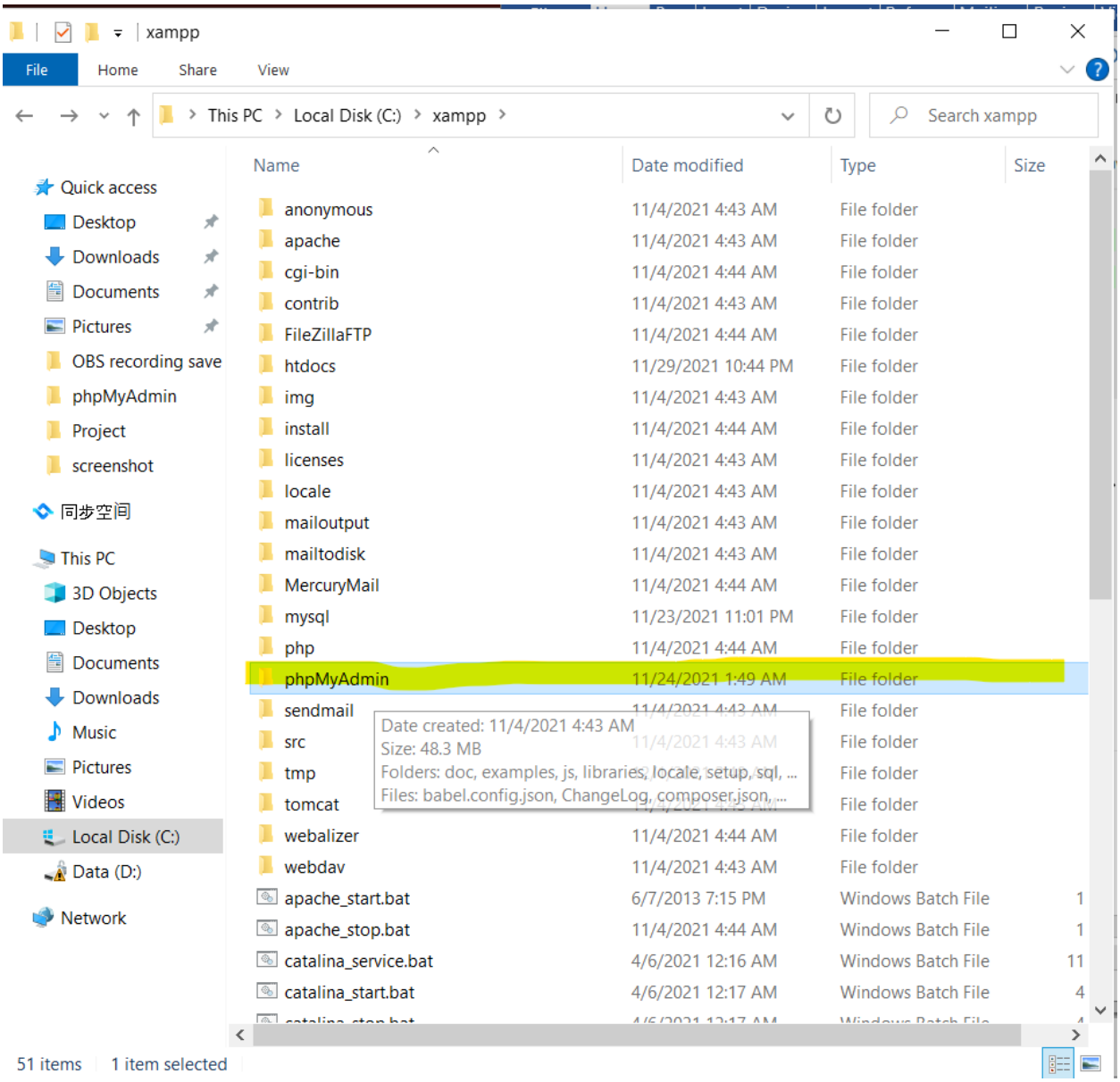
Then,press this button 



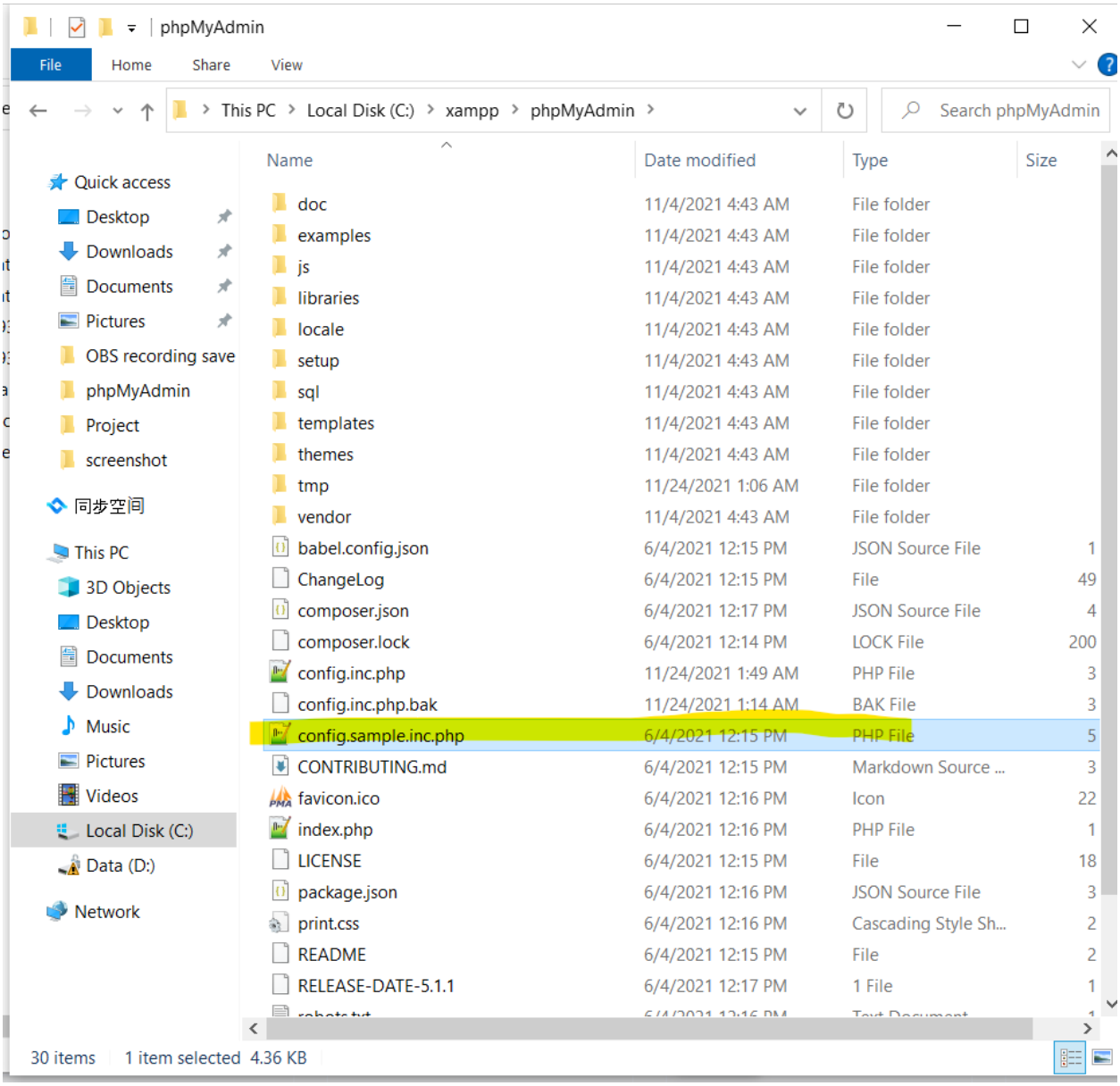


Then,we press save buttons.

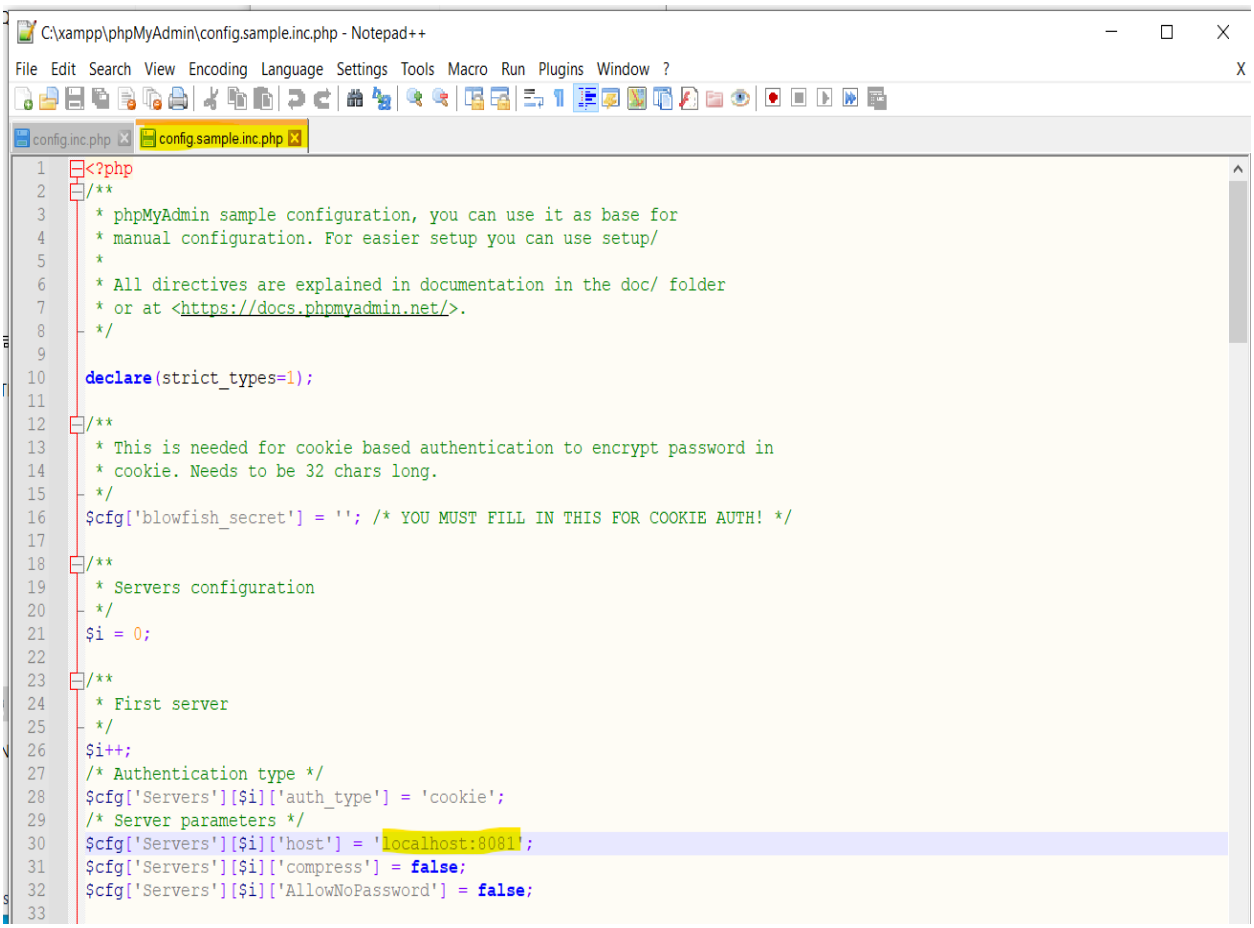
These are the configuration of the XAMPP for Apache and MySql.



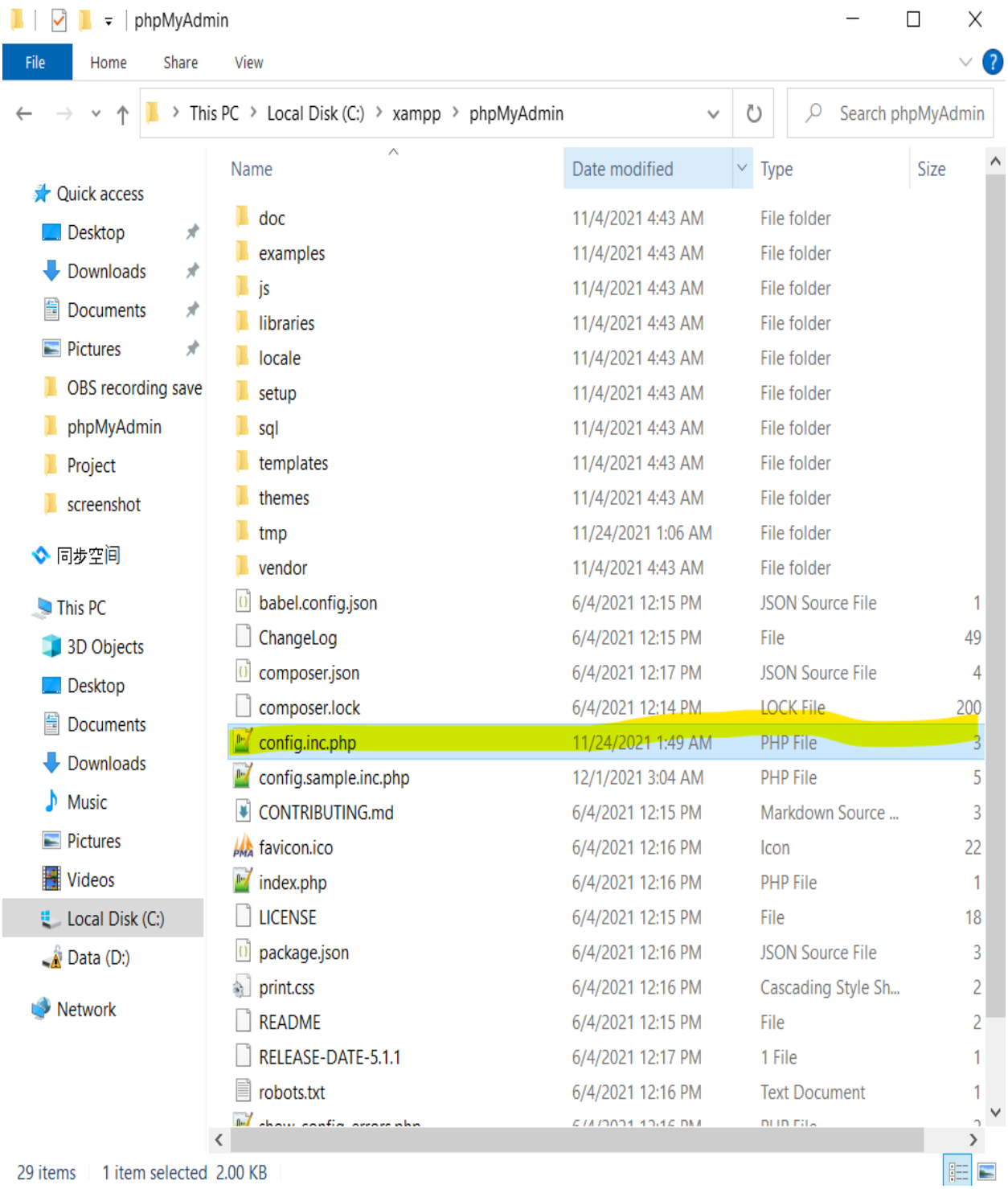
We go here C:\xampp\phpMyAdmin



Click this file showed above.



Then,replace whatever value is there with localhost:8081


phpMyAdmin
File
Home
Share
View
This PC › Local Disk (C:) › xampp › phpMyAdmin
Search phpMyAdmin
Quick access
Desktop
Downloads
Documents
Pictures
OBS recording save
phpMyAdmin
Project
screenshot
同步空间
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Data (D:)
Network
Name
Date modified
Type
Size
doc 11/4/2021 4:43 AM File folder
examples 11/4/2021 4:43 AM File folder
js 11/4/2021 4:43 AM File folder
libraries 11/4/2021 4:43 AM File folder
locale 11/4/2021 4:43 AM File folder
setup 11/4/2021 4:43 AM File folder
sql 11/4/2021 4:43 AM File folder
templates 11/4/2021 4:43 AM File folder
themes 11/4/2021 4:43 AM File folder
tmp 11/24/2021 1:06 AM File folder
vendor 11/4/2021 4:43 AM File folder
babel.config.json 6/4/2021 12:15 PM JSON Source File 1
ChangeLog 6/4/2021 12:15 PM File 49
composer.json 6/4/2021 12:17 PM JSON Source File 4
composer.lock 6/4/2021 12:14 PM LOCK File 200
config.inc.php 11/24/2021 1:49 AM PHP File 3
config.sample.inc.php 12/1/2021 3:04 AM PHP File 5
CONTRIBUTING.md 6/4/2021 12:15 PM Markdown Source ... 3
favicon.ico 6/4/2021 12:16 PM Icon 22
index.php 6/4/2021 12:16 PM PHP File 1
LICENSE 6/4/2021 12:15 PM File 18
package.json 6/4/2021 12:16 PM JSON Source File 3
print.css 6/4/2021 12:16 PM Cascading Style Sh... 2
README 6/4/2021 12:15 PM File 2
RELEASE-DATE-5.1.1 6/4/2021 12:17 PM 1 File 1
robots.txt 6/4/2021 12:16 PM Text Document 1
29 items
1 item selected 2.00 KB

C:\xampp\phpMyAdmin\config.inc.php - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

config.inc.php | config.sample.inc.php

```php
<?php
/*
 * This is needed for cookie based authentication to encrypt password in
 * cookie
 */
$cfg['blowfish_secret'] = 'xampp'; /* YOU SHOULD CHANGE THIS FOR A MORE SECURE COOKIE AUTH! */

/*
 * Servers configuration
 */
$i = 0;

/*
 * First server
 */
$i++;

/* Authentication type and info */
$cfg['Servers'][$i]['auth_type'] = 'config';
$cfg['Servers'][$i]['user'] = 'root';
$cfg['Servers'][$i]['password'] = '';
$cfg['Servers'][$i]['extension'] = 'mysqli';
$cfg['Servers'][$i]['AllowNoPassword'] = true;
$cfg['Lang'] = '';

/* Bind to the localhost ipv4 address and tcp */
$cfg['Servers'][$i]['host'] = 'localhost:3307';
$cfg['Servers'][$i]['connect_type'] = 'tcp';

/* User for advanced features */
$cfg['Servers'][$i]['controluser'] = 'pma';
$cfg['Servers'][$i]['controlpass'] = '';
```

Then,replace whatever value is there with localhost:3307

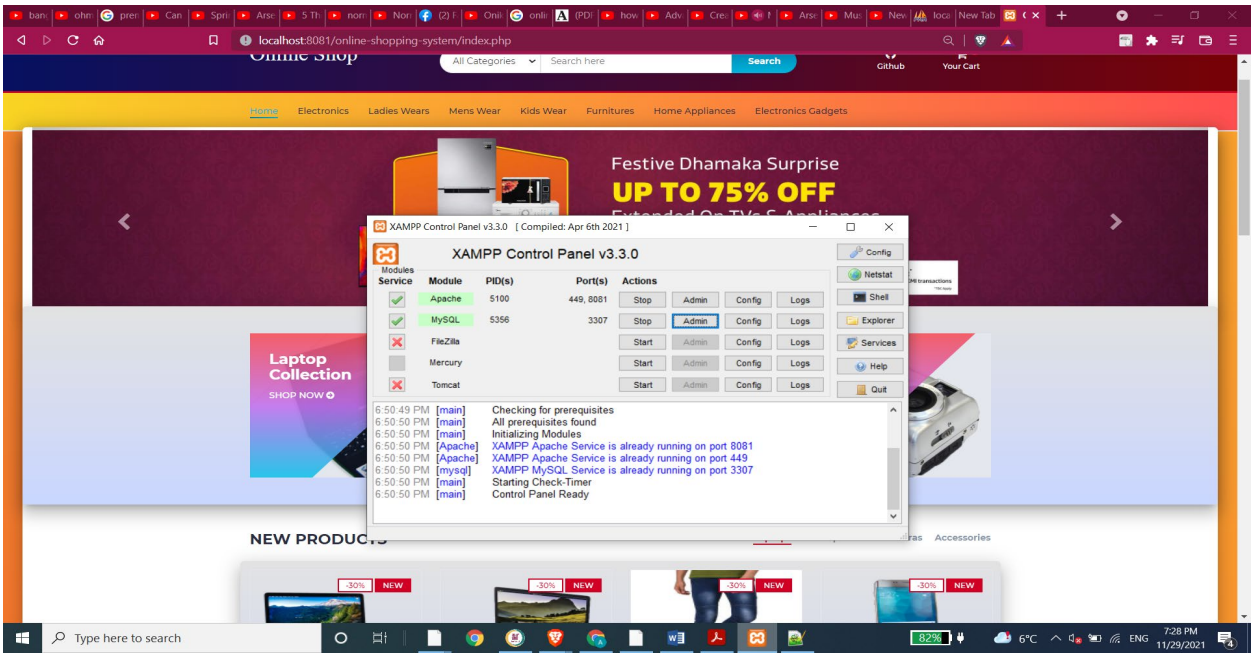
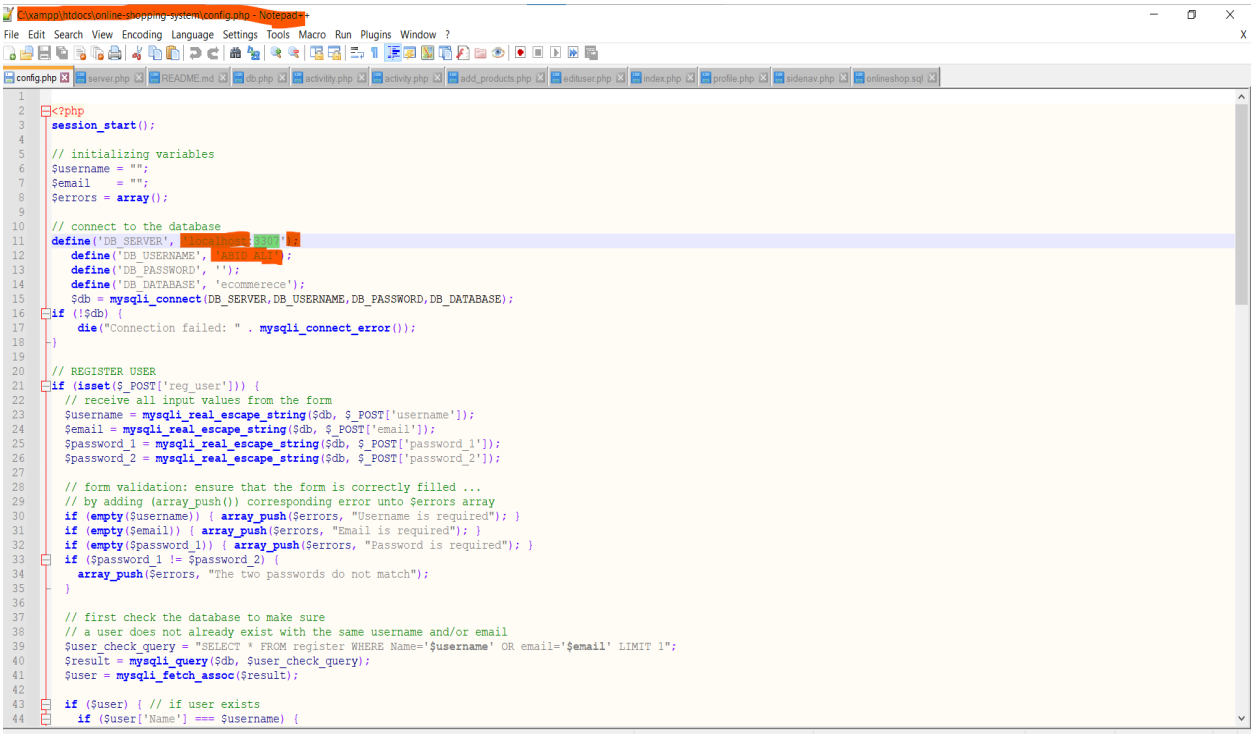


Fig: XAMPP configuration



```
<?php
 session_start();

 // initializing variables
 $username = "";
 $email    = "";
 $errors = array();

 // connect to the database
 define('DB_SERVER', '[redacted]');
    define('DB_USERNAME', '[redacted]');
    define('DB_PASSWORD', '');
    define('DB_DATABASE', 'ecommerece');
    $db = mysqli_connect(DB_SERVER,DB_USERNAME,DB_PASSWORD,DB_DATABASE);
if (!$db) {
     die("Connection failed: " . mysqli_connect_error());
}

 // REGISTER USER
if (isset($_POST['reg_user'])) {
   // receive all input values from the form
   $username = mysqli_real_escape_string($db, $_POST['username']);
   $email = mysqli_real_escape_string($db, $_POST['email']);
   $password_1 = mysqli_real_escape_string($db, $_POST['password_1']);
   $password_2 = mysqli_real_escape_string($db, $_POST['password_2']);

   // form validation: ensure that the form is correctly filled ...
   // by adding (array_push()) corresponding error unto $errors array
   if (empty($username)) { array_push($errors, "Username is required"); }
   if (empty($email)) { array_push($errors, "Email is required"); }
   if (empty($password_1)) { array_push($errors, "Password is required"); }
   if ($password_1 != $password_2) {
     array_push($errors, "The two passwords do not match");
   }

   // first check the database to make sure
   // a user does not already exist with the same username and/or email
   $user_check_query = "SELECT * FROM register WHERE Name='$username' OR email='$email' LIMIT 1";
   $result = mysqli_query($db, $user_check_query);
   $user = mysqli_fetch_assoc($result);

   if ($user) { // if user exists
     if ($user['Name'] === $username) {
```

Config.php we use the port number of the local host of XAMPP that is 3307

It's the file config.php in the folder online-shopping-system.

C:\xampp\htdocs\online-shopping-system\admin\admin\includes\db.php - Notepad++

```php
<?php

$servername = "localhost:3307";
$username = "root";
$password = "";
$db = "onlineshop";


// Create connection
$con = mysqli_connect($servername, $username, $password,$db);

// Check connection
if (!$con) {
    die("Connection failed: " . mysqli_connect_error());
}


?>
```

To activate the admin page ,we need to change the port number according to XAMPP that is 3307.

To find this find look at the location I marked at the top.

C:\xampp\htdocs\online-shopping-system\admin\server\server.php - Notepad++

```php
<?php
session_start();
?>

<?php

// initializing variables
$name = "";
$username = "";
$usn = "";
$email    = "";
$errors = array();
$reg_date = date("Y/m/d");

// connect to the database
$db = mysqli_connect('localhost:3307', 'root', '', 'onlineshop');


// REGISTER USER
if (isset($_POST['reg_user'])) {
  // receive all input values from the form
  $username = mysqli_real_escape_string($db, $_POST['admin_name']);
  $email = mysqli_real_escape_string($db, $_POST['admin_email']);
  $password_1 = mysqli_real_escape_string($db, $_POST['password_1']);
  $password_2 = mysqli_real_escape_string($db, $_POST['password_2']);

  // form validation: ensure that the form is correctly filled ...
  // by adding (array_push()) corresponding error unto $errors array
  if (empty($username)) { array_push($errors, "Username is required"); }
  if (empty($email)) { array_push($errors, "Email is required"); }
  if (empty($password_1)) { array_push($errors, "Password is required"); }
  if ($password_1 != $password_2) {
  array_push($errors, "The passwords do not match");
  }

  // first check the database to make sure
  // a user does not already exist with the same username and/or email
  $user_check_query = "SELECT * FROM admin_info WHERE admin_name='$username' OR admin_email='$email' LIMIT 1";
  $result = mysqli_query($db, $user_check_query);
  $user = mysqli_fetch_assoc($result);

  if ($user) { // if user exists
    if ($user['admin_name'] === $username) {
      array_push($errors, "Username already exists");
```

PHP Hypertext Preprocessor file | length : 3,029 lines : 101 | Ln : 30 Col : 67 Pos : 902 | Unix (LF) | UTF-8 | INS

To activate the server ,we need to change the port number according to XAMPP that is 3307.

To find this find look at the location I marked at the top.

After setting up we can start installing the onlie shopping system installer.

Online Shopping System Installer.exe	11/30/2021 7:51 PM	Application	85,824 KB

This is the installer file

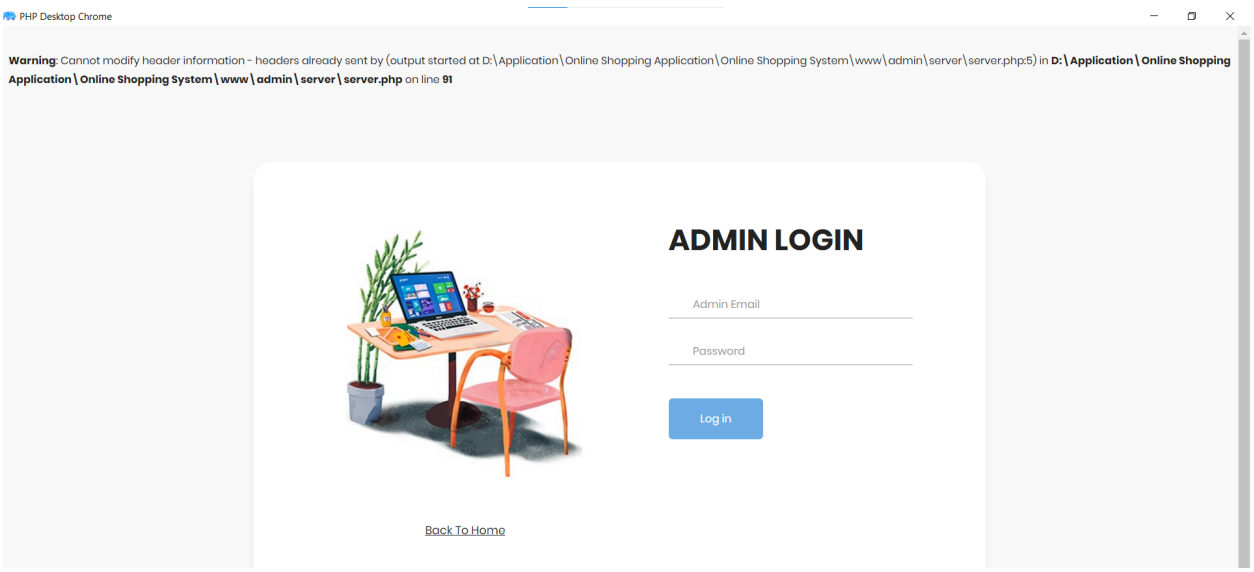
This is

the executable file

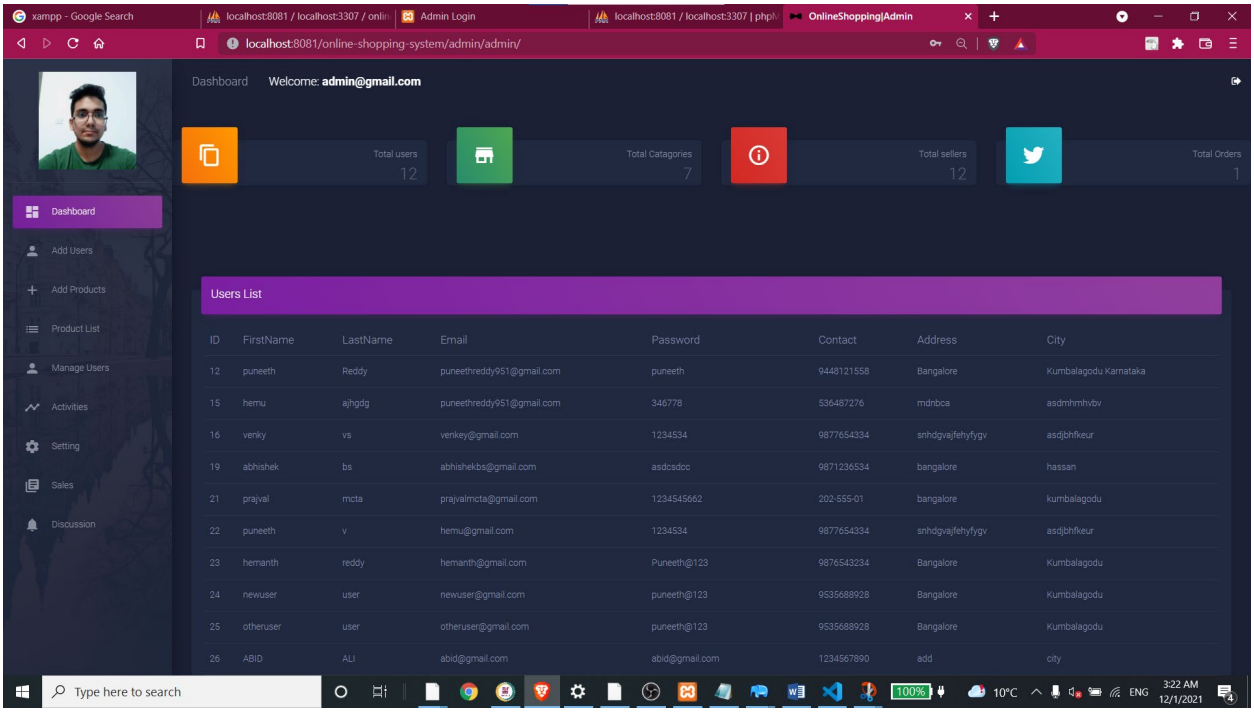
Section 6: Conclusion

Problems:-

The problem that we faced were to create the ER-diagram and bugs came to create the online shopping system. When we run the program by executable we can't access the admin function because the compiler of the executable is php5 version. I looked at online and can't find any solution to solve this problem



.But,this problem can be avoided if we search using the link because the XAMPP has php8 functions.



http://localhost:8081/online-shopping-system/

We can see that,when we use the link in browser it doesn't give error because XAMPP has upto date php8 version.

Solutions:-

Whenever,I faced problem then I searched for the solution in YouTube ,Stackoverflow and W3S school.

Summary:-

The goal of this project is to provide a realistic experience in the conceptual design, logical design, implementation, operation, and maintenance of a relational database and associated a application. admin_info(admin_id , admin_email , admin_name , admin_password)and orders (order_id , product_id , qty , user_id , to_id , p_status) and user_info (user_id , first_name , last_name , password , address , mobile). The function of the system is divided into two parts: customer oriented and administrator oriented. The customer oriented part contains login, search, choose and buy; the administrator oriented part contains complete the addition, deletion and modification of products and data related to customer, as well as monitoring the customer's details.

This program was completely done by me.I was doing alone so I had to give lot of effort and time. I tried to make the project like production level & professional as much possible.

Future Work

The following section discusses the work that will be implemented with future releases of the software.

1. **Detailed categories:** Future work could involve adding more categories which are more detailed and have additional items.

2. **Enhanced User Interface:** Work on enhancing the user interface by adding more user-interactive features.

3. **Recommended Items:** Add a bar that would display the most-recommended items which would depend on the number of times an item has been purchased by any users.

4. **Payment Options:** Add different payment options, such as Visa, MasterCard, PayPal, etc., where a user can also save the card information for later checkouts.

5. **Shipping Options:** Add different types of shipping options: regular shipping, expedited shipping, international shipping, etc.

6. **Recent History:** Display the user's recently browsed items in the recent-history tab.

Attachments:-

1) DB8_2019380141_ABID ALI.docx
2) DB8_2019380141_ABID ALI.pdf
3)ER-Diagram.jpg
4) Configuration(Short Description).md
5)Readme.txt
6) Online Shopping System Installer.exe
7) Configuration Video&Webpage preview .mp4
8) Tour of website.mp4
9)Readme.txt
10) online-shopping-system(Folder)
It contains:
(a) Source File
(b) database folder(onlineshop.sql)
(c) screenshot folder(Results of the projects)

Reference:-

1) https://www.youtube.com/

2) https://www.youtube.com/watch?v=-f8N4FEQWyY

3) https://www.youtube.com/watch?v=fUK94jOFwBc

4)https://www.ionos.com/digitalguide/server/tools/xampp-tutorial-create-your-own-local-test-server/

5) https://www.youtube.com/watch?v=OK_JCtrrv-c

6) https://www.w3schools.com/php/

Link to run the program

http://localhost:8081/online-shopping-system/