**205CDE Developing the Modern Web**

**Individual Assignment**

**Reflective Report**

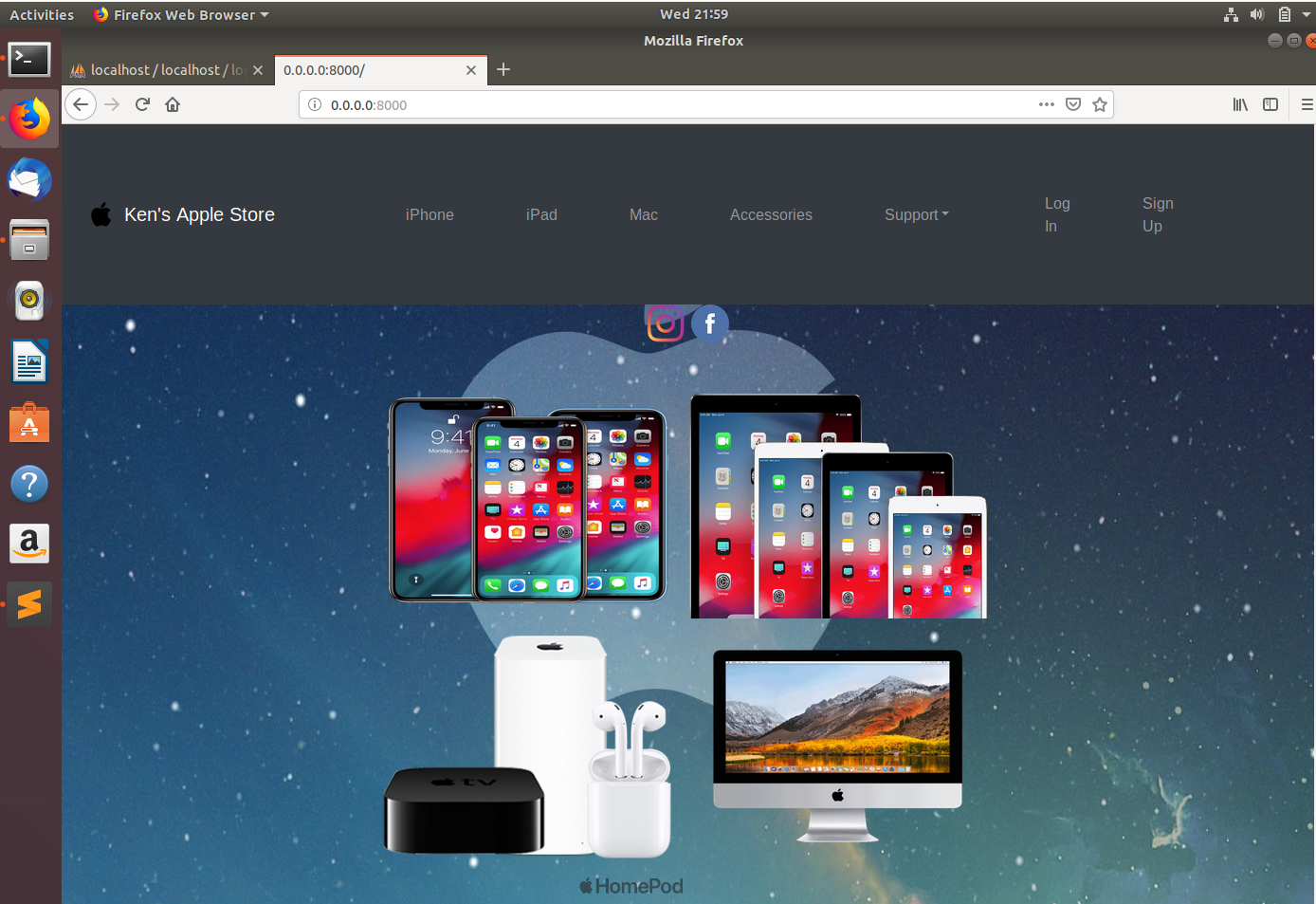
**Student Name : Chan Sing Ping**

**Student ID: 55201359**

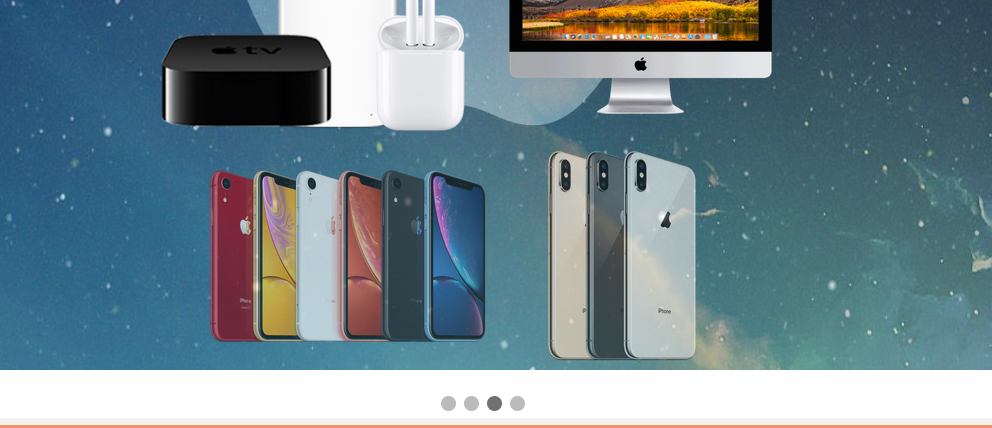
**Github:** <https://github.com/siusing1997/205CDE-Project/>

In this project, I have created a B2C website which the customers can join our member to buy our product with discount. Programming language like HTML, CSS, Javascript, jQuery, Python flask, Xampp, mySQL database and Bootstrap are being used to build up this whole website. Virtual machine Box in Ubuntu is installed with Linux OS. The purpose of creating this website is because my shop is extending the marketing size. By constructing online shop, more new customers can be attracted. Reflectively speaking, I think I need to reallocate the time spending on coding and the design part in the future, and improve my website to contain more function using different programming language, such as PHP C++. In this project, I have spending most of the time in constructing the login and shopping cart function, I hope my time management of will be better in the future

**Design and Function of Main Page**

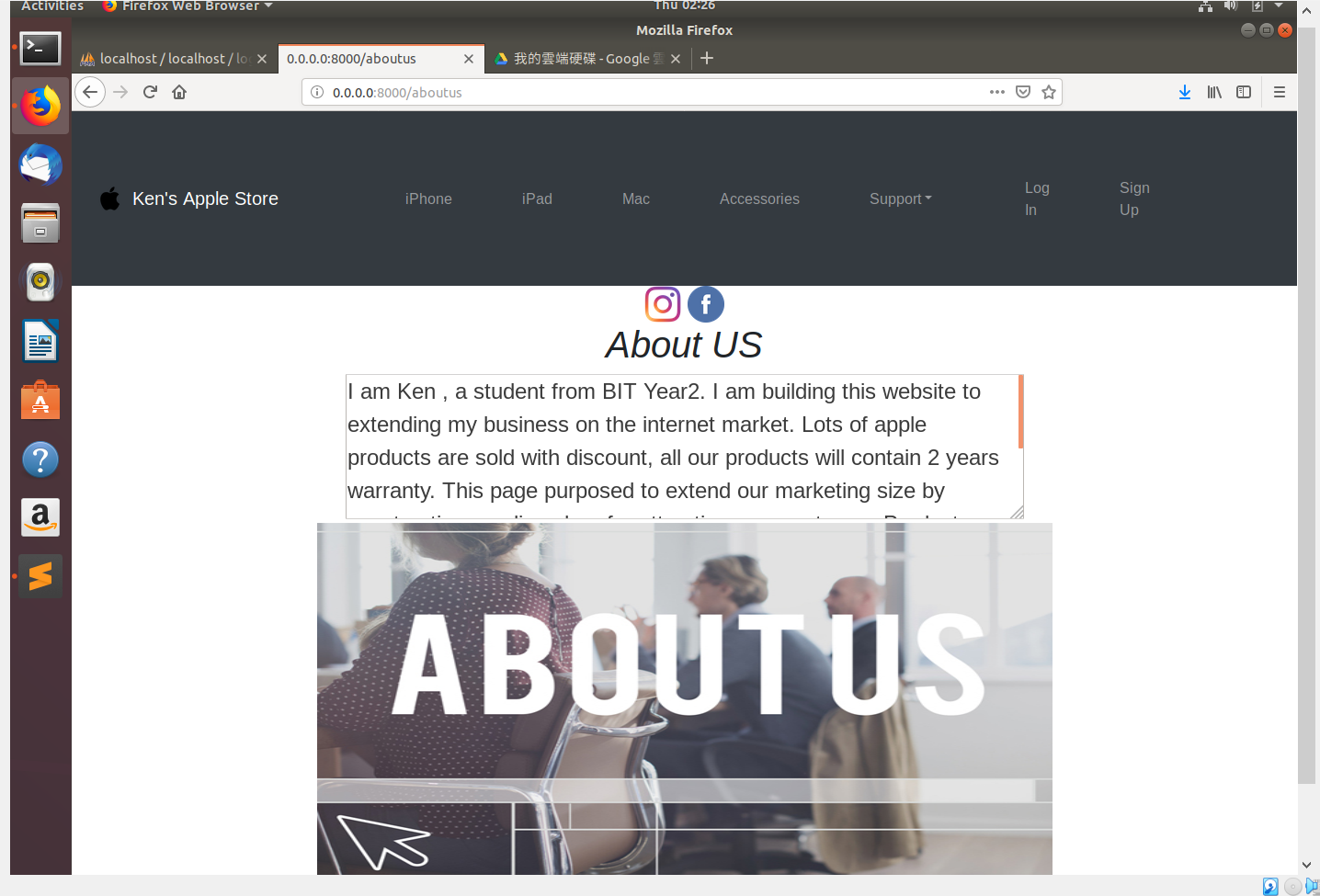


This is the main page of my website design. The main page included a navigation bar with CSS styling, jQuery which the menu bar icon will appear if the resolution is downsized. There are Login and Signup function for the member to buy our products in the product list. In this main page, image container is used for storing photos of four main kind of products that contain hyperlink to product pages. Transparent product images are being used, and the Instagram and Facebook logo can be linked to my personal account respectively.



Also, I have inserted a slideshow with CSS styling to show our new arrivals product at the bottom of the page. The whole page have insert hyperlinks to navigate our customers to buy their favourite products.

**About us**



This page is simply providing the purpose of why I am setting up this website and the background information of me. It used HTML textbox and insert with a banner under it.

**The Product list and Database Linkage**

In our products page, there are products list from the database which is being inserted to the product pages with the use of python flask function. In this case, Python flask is used for creating a function for adding product to the shopping cart. When the sql statement executed, the ‘rows’ will store all data get from the database table ‘iphone’. In HTMl file, form action used to execute the python function, and the {% for row in rows%} input value=” is the location where the product data will be shown.

**Python Coding of insert MYSQL database list (new app.py):**

cursor = db.cursor()

sql = (“SELECT \* FROM iphone “)

cursor.execute(sql)

db.commit()

rows = cursor.fetchall()

return render\_template(‘iphone,hml’, rows = rows)

**Table list in HTML file (iphone.html):**

{% for row in rows %}

<form action="{{url\_for('addProduct')}}" method = "POST">

<tr>

<td>

<input type="text" value="{{row[0]}}" name="model" readonly>

</td>

<td>

<input type="text" value="{{row[1]}}" name="color" readonly>

</td>

<td>

<input type="text" value="{{row[2]}}" name="capacity" readonly>

</td>

<td>

<input type="text" value="{{row[3]}}" name="price" readonly>

</td>

<td>

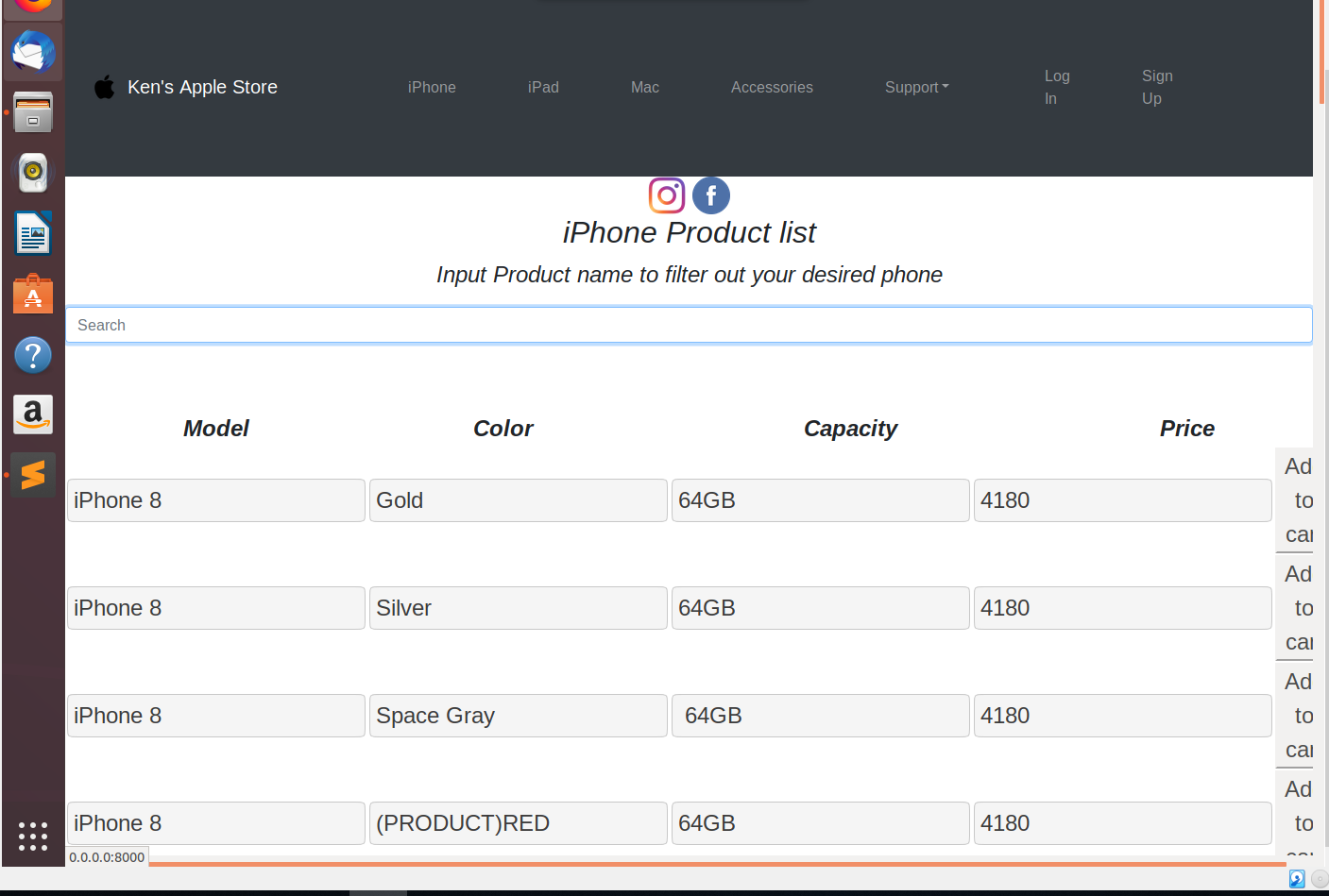
<button type="summit">Add to cart</button>

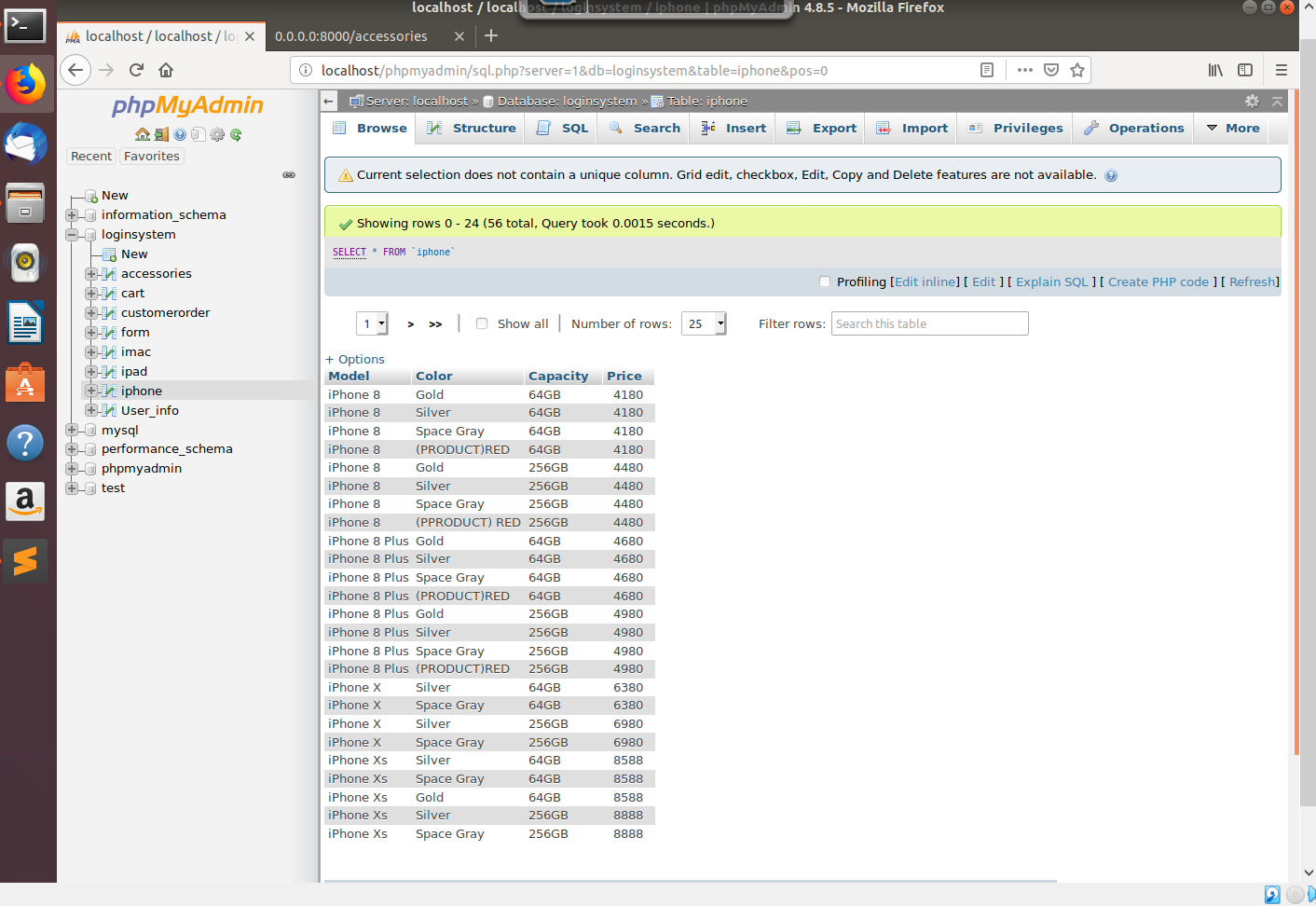
</td>

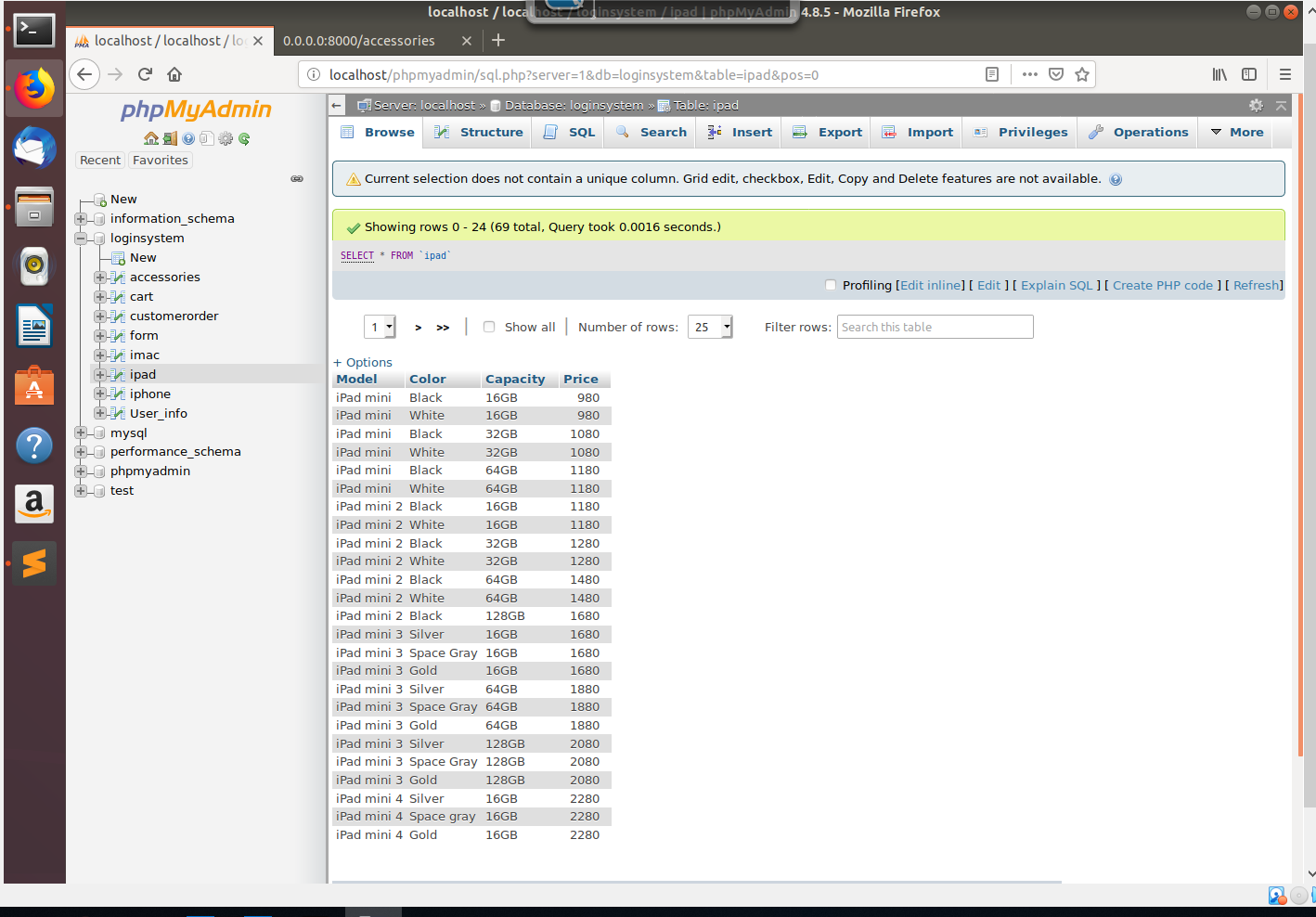
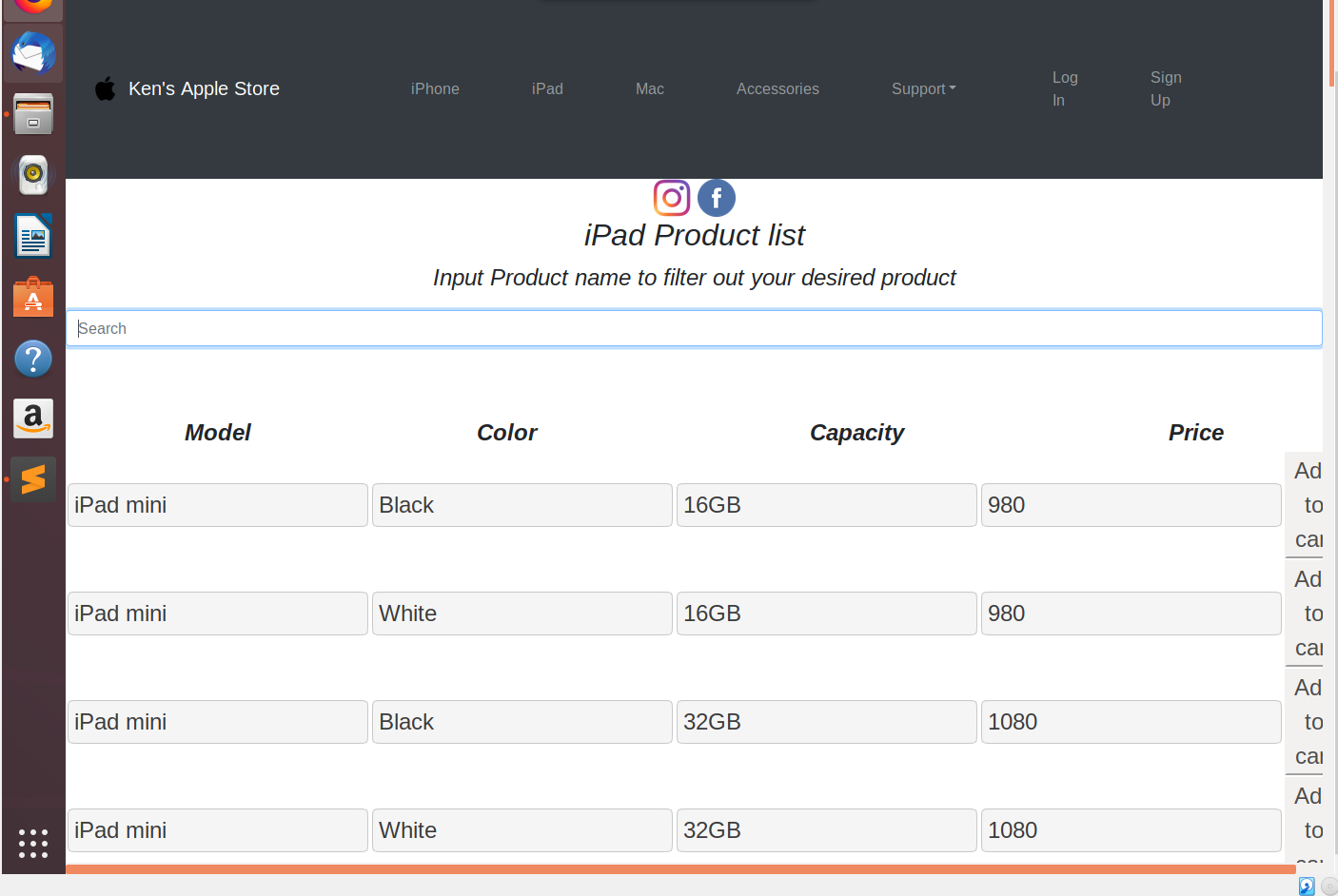
</tr>

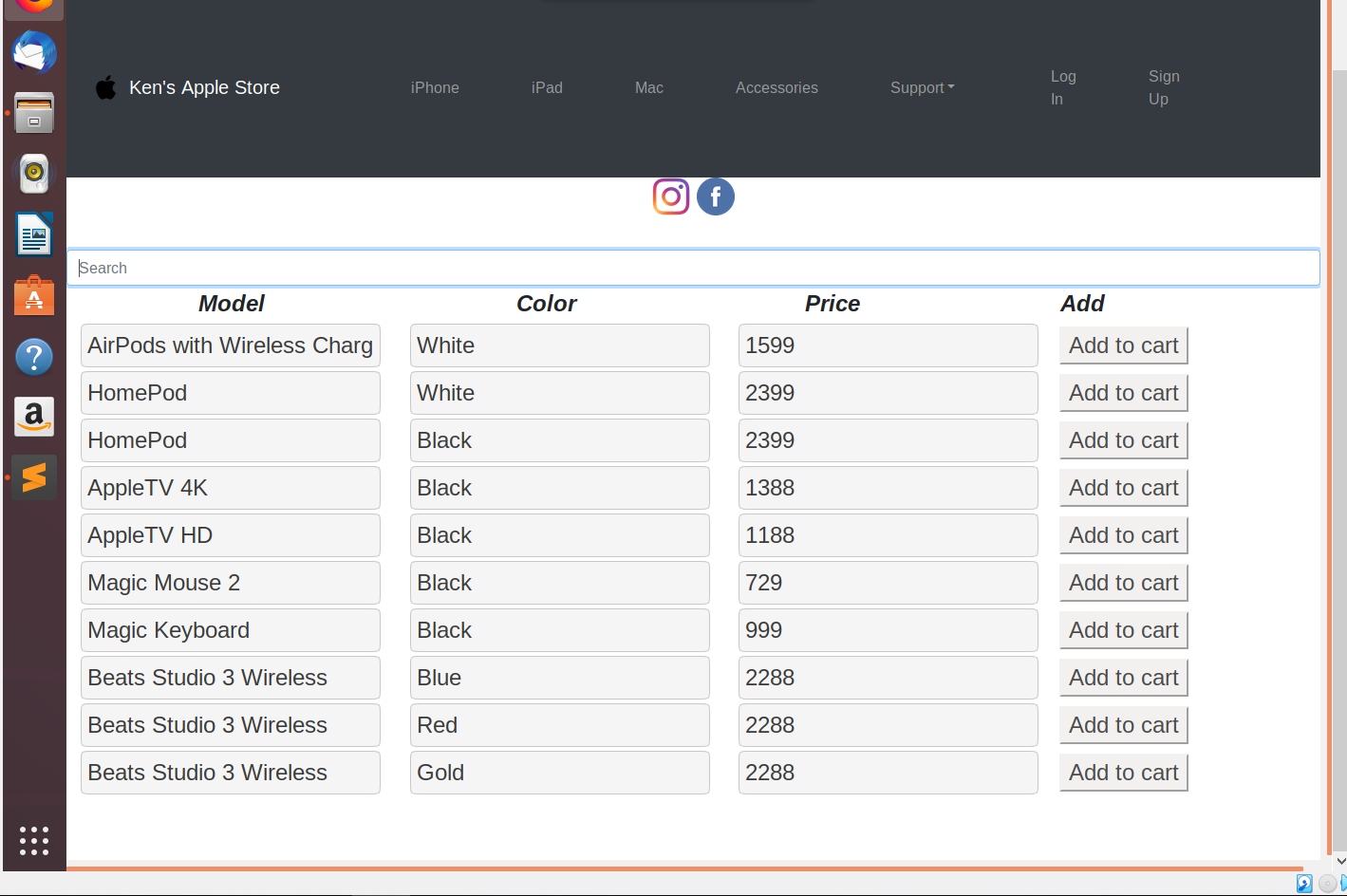
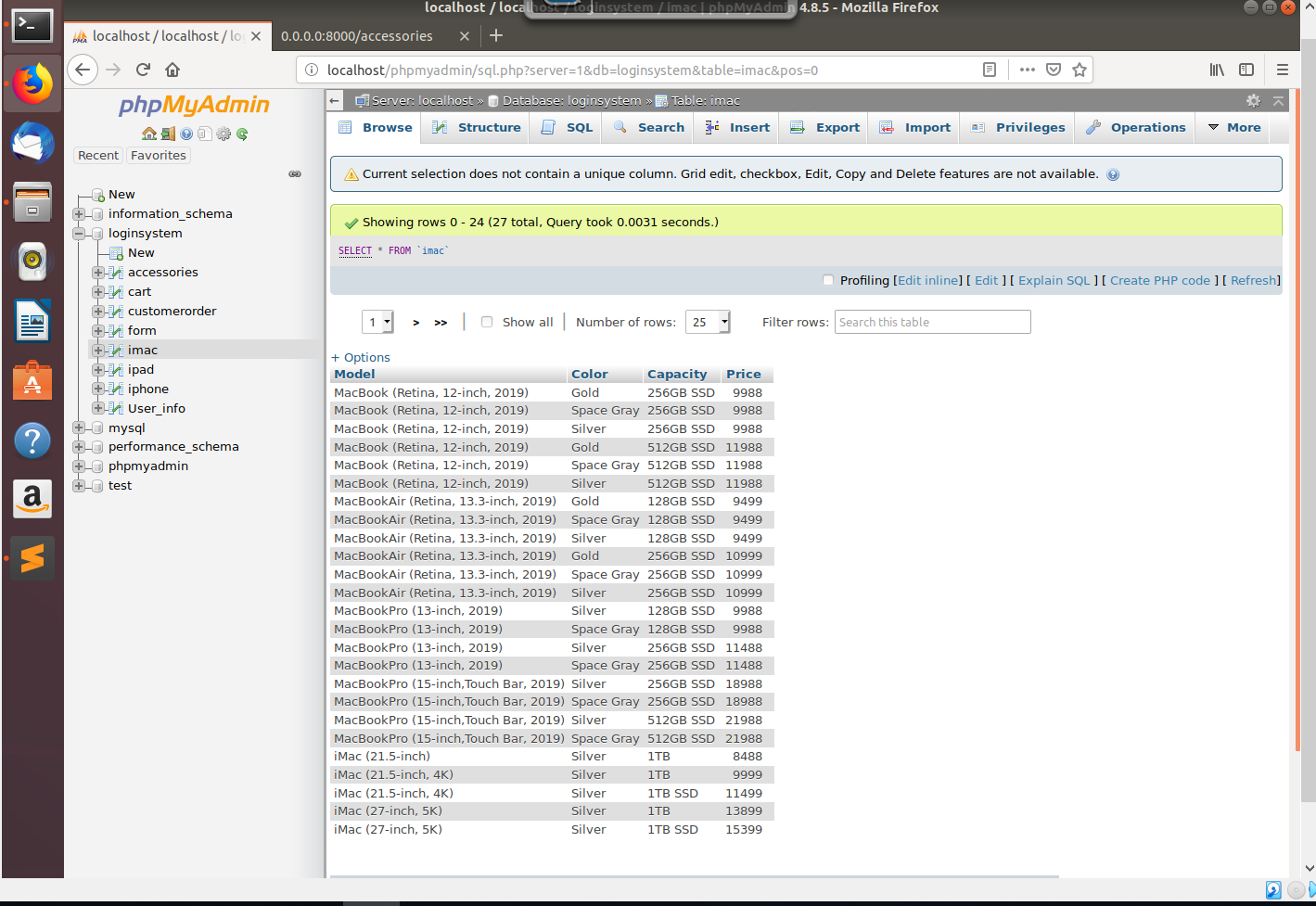
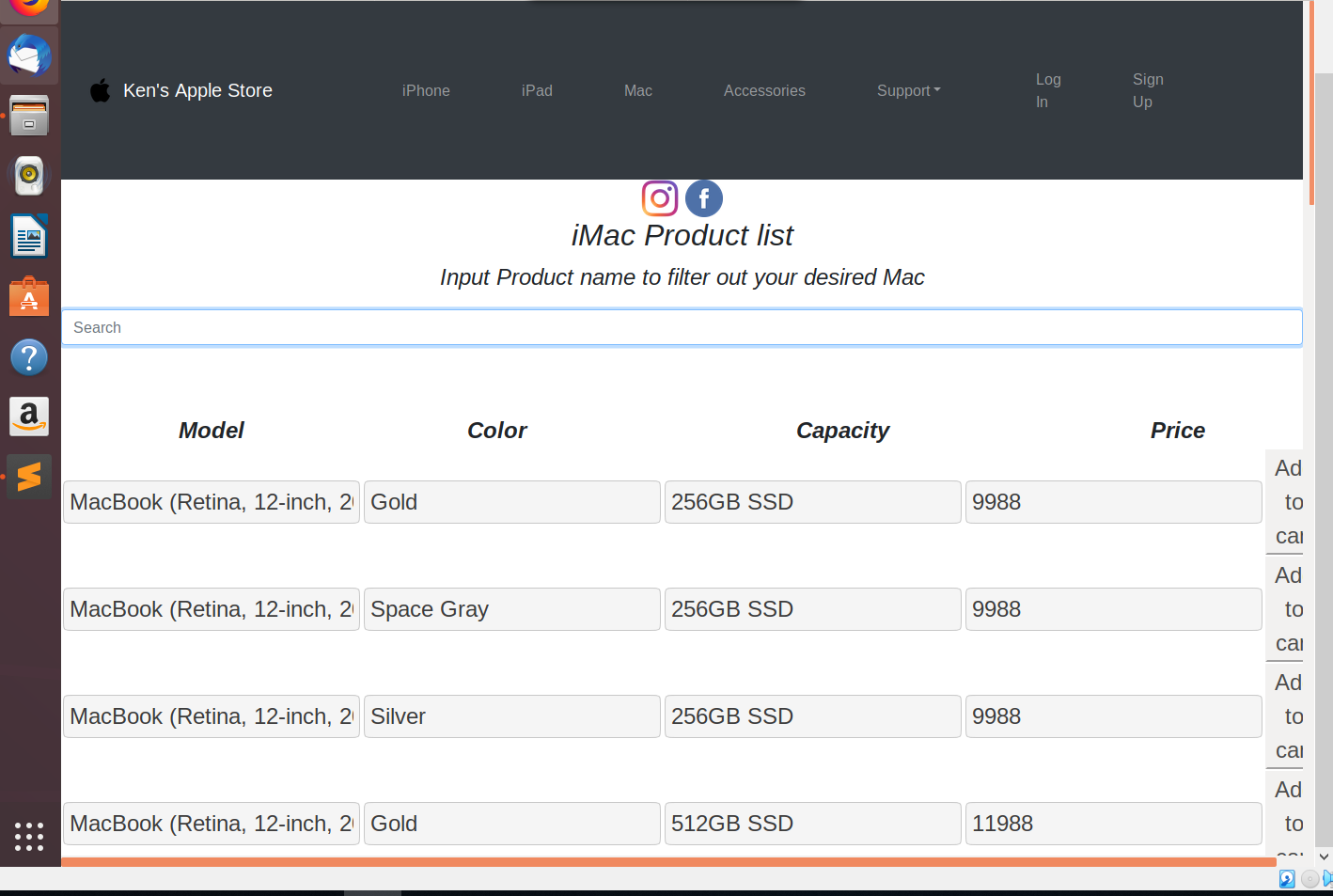
</form>

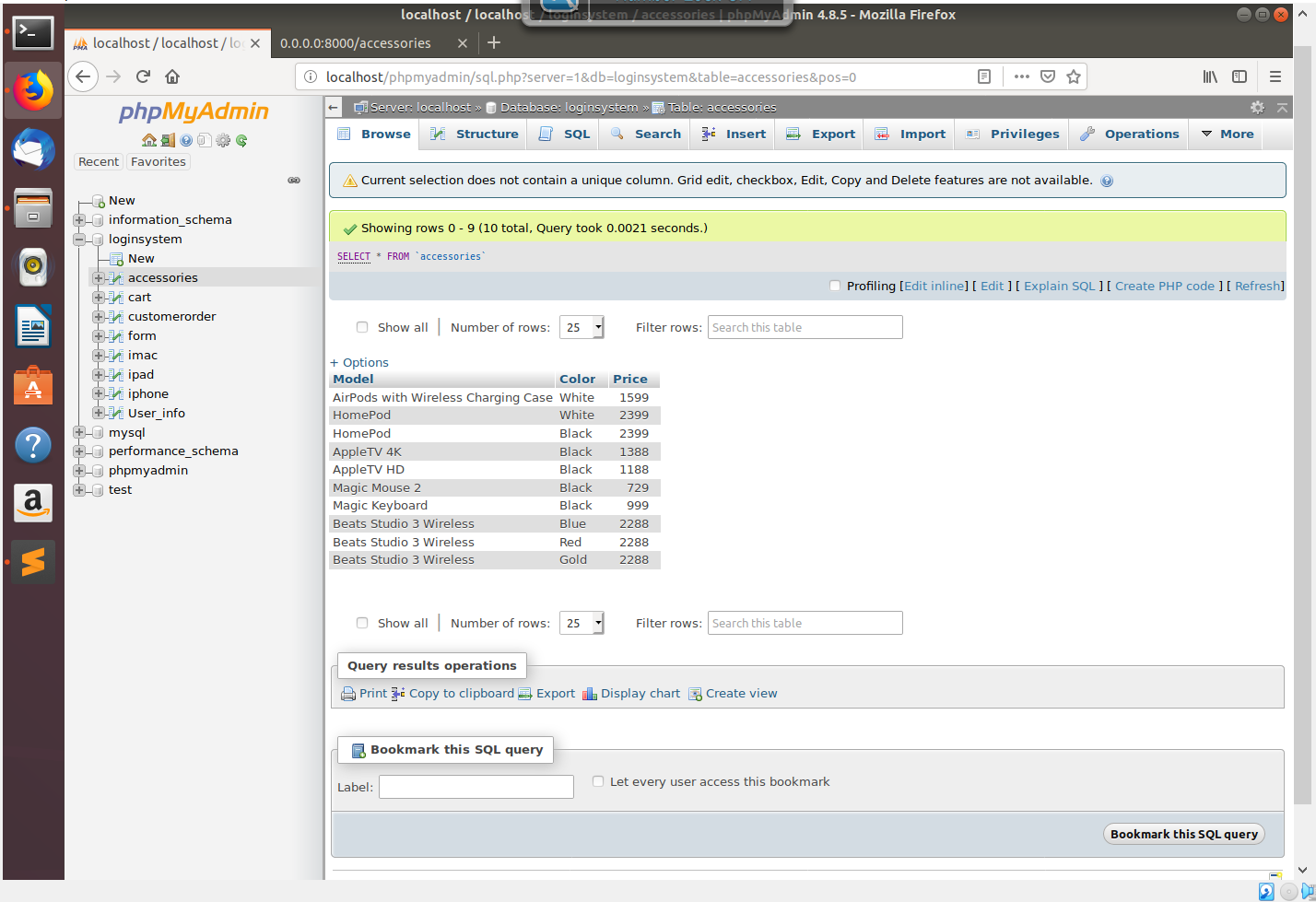
{% endfor %}











Also, the table have a javascript function that can search and filter out the desired product.

**Javascript coding (inside html file)**

<script>

function SearchAndFilterThingy() {

var input, filter, table, tr, td, x;

input = document.getElementById("UserInput");

filter = input.value.toUpperCase();

table = document.getElementById("product");

tr = table.getElementsByTagName("tr");

for (x = 0; x < tr.length; x++) {

td = tr[x].getElementsByTagName("td")[0];

if (td) {

if (td.innerHTML.toUpperCase().indexOf(filter) > -1) {

tr[x].style.display = "";

}

else {

tr[x].style.display = "none";

}

}

}

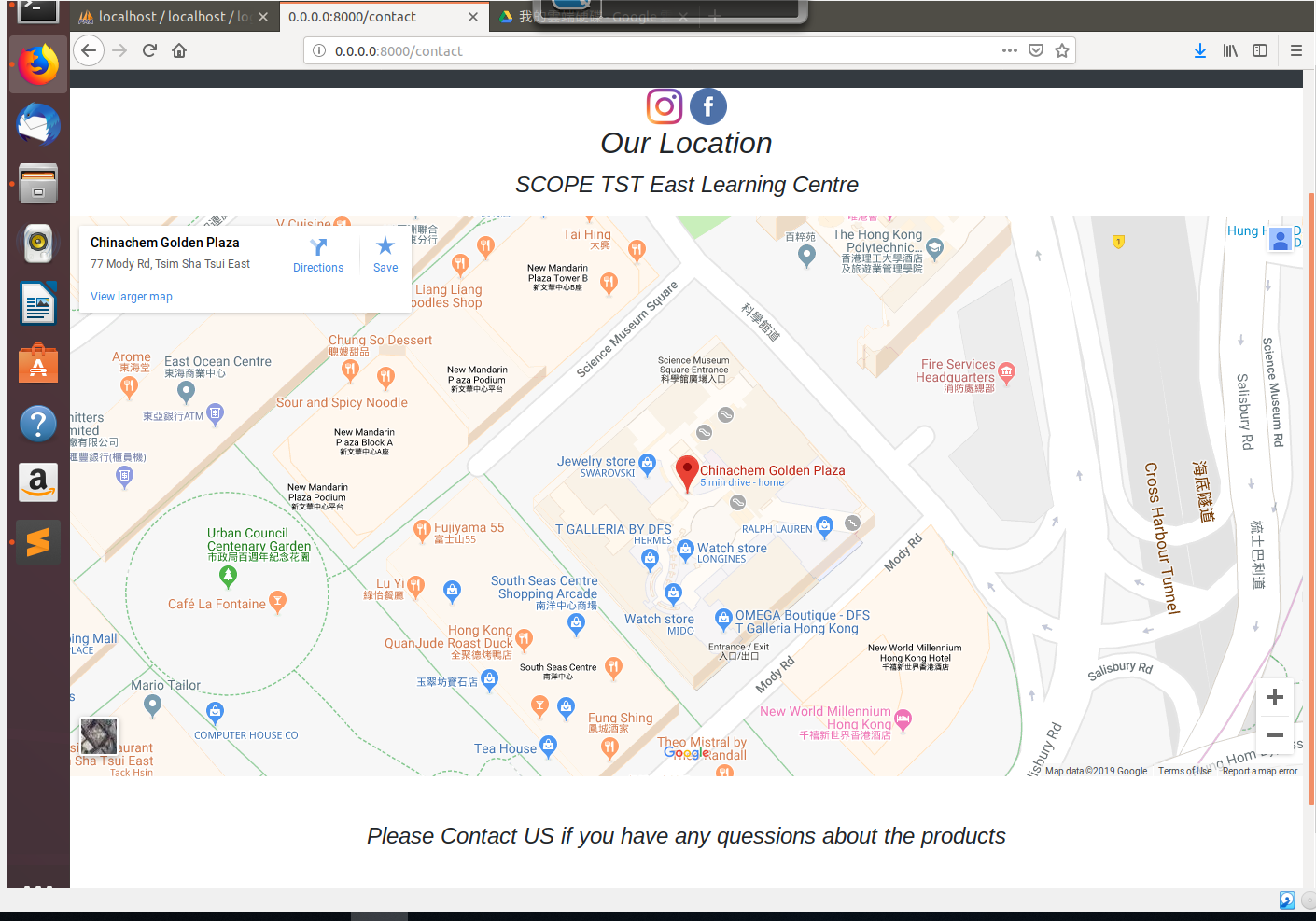
}

</script>

In the Contact page, the google map html function iframe with CSS styling is being used to display our location and the tag shows the exact location of SCOPE TST East Learning Centre. Also, there are contact methods including personal email and contact number.

**Iframe coding (Embedded GoogleMap)**

<iframe width="100%" height="600" src="https://maps.google.com/maps?width=100%&amp;height=600&amp;hl=en&amp;q=77%20Mody%20Rd%2C%20Tsim%20Sha%20Tsui%20East+(SCOPE)&amp;ie=UTF8&amp;t=&amp;z=19&amp;iwloc=B&amp;output=embed" frameborder="0" scrolling="no" marginheight="0" marginwidth="0">



Apart from providing contact method, there are opinion form for customers to fill in, they can comment anything they want us to improve. After submitting the form, the forms data will be insert into data base as well. This python function insert data into database table called ‘form’

with sql statement, and format it to become string in order to store in database. If the input box have nothing, the command window will print ("You have not input anything"). If the data is filled in, the data will be sent to the database table ‘form’.

**Python flask Function (Form)**

**Coding**

@app.route('/form', methods=['GET','POST'])

def form():

request.method == 'POST'

firstname = request.form.get("firstname")

lastname = request.form.get("lastname")

contact = request.form.get("contact")

comment = request.form.get("comment")

cursor = db.cursor()

# Insert data to database

sql ='''

INSERT INTO form (first\_name, last\_name, contact\_no, comment)

VALUES ('%s','%s','%s','%s')

'''

cursor.execute(sql%(firstname, lastname, contact, comment))

db.commit()

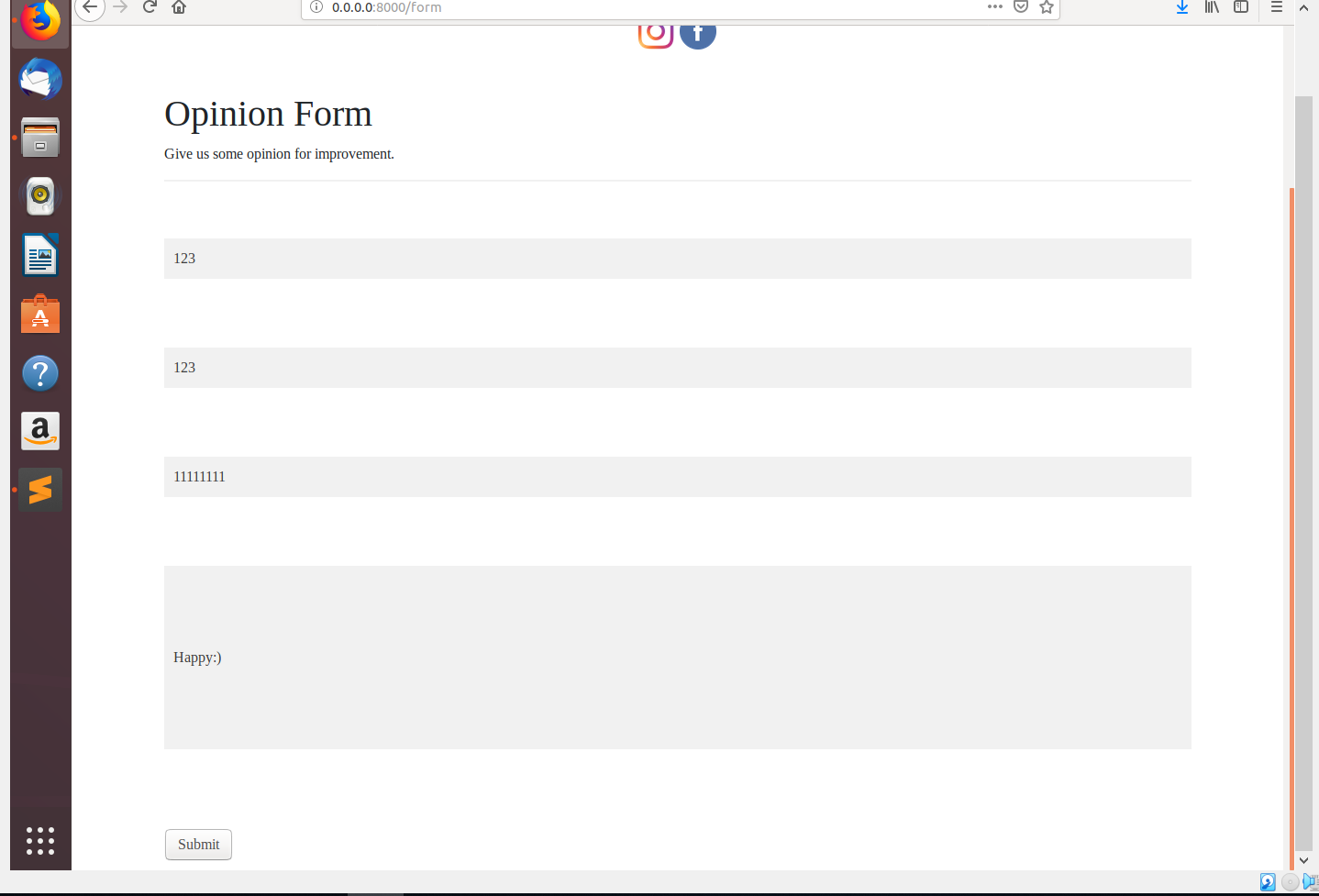
if comment is None:

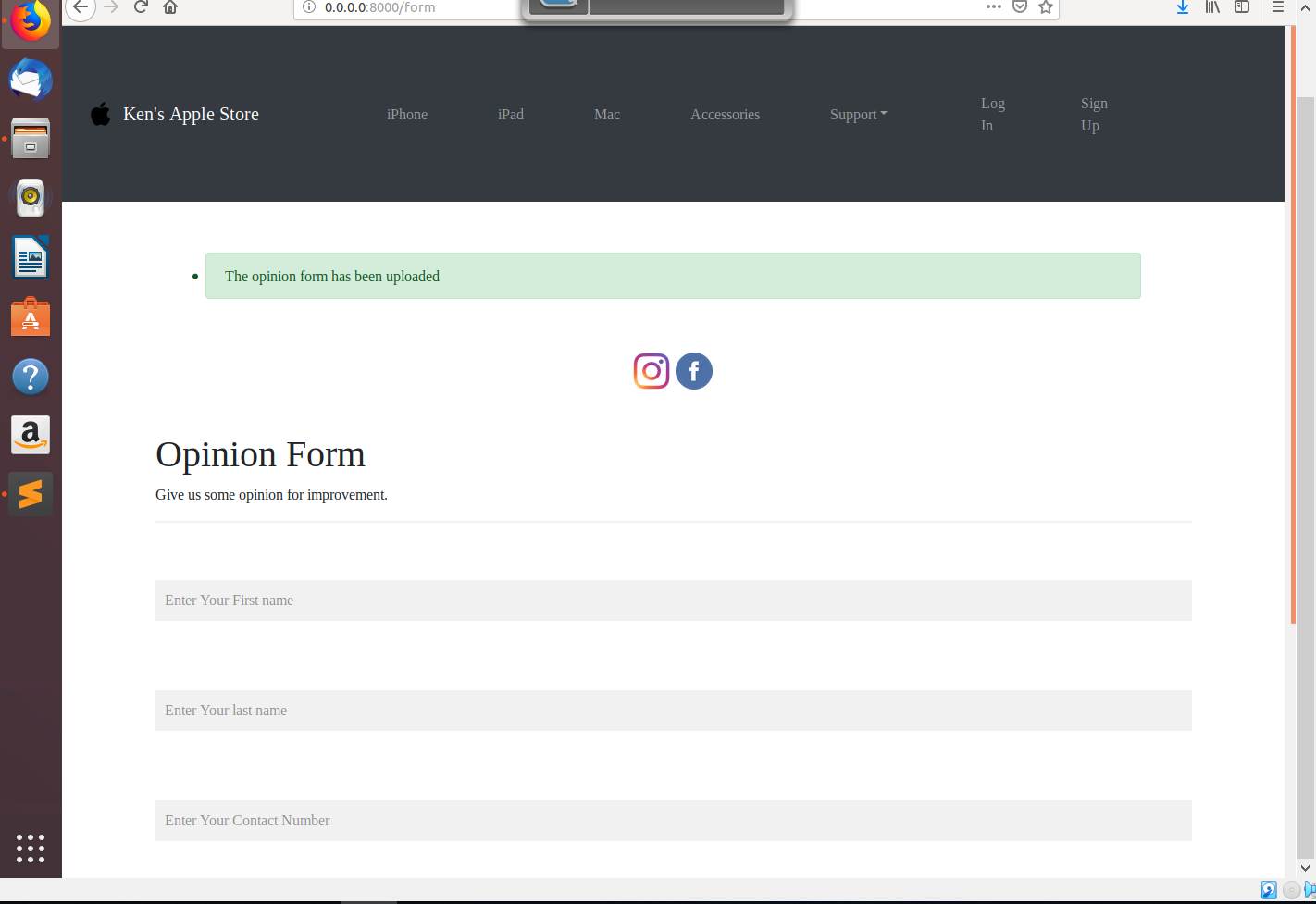
print("You have not input anything")

else:

flash("The opinion form has been uploaded","success")

return render\_template("form.html")

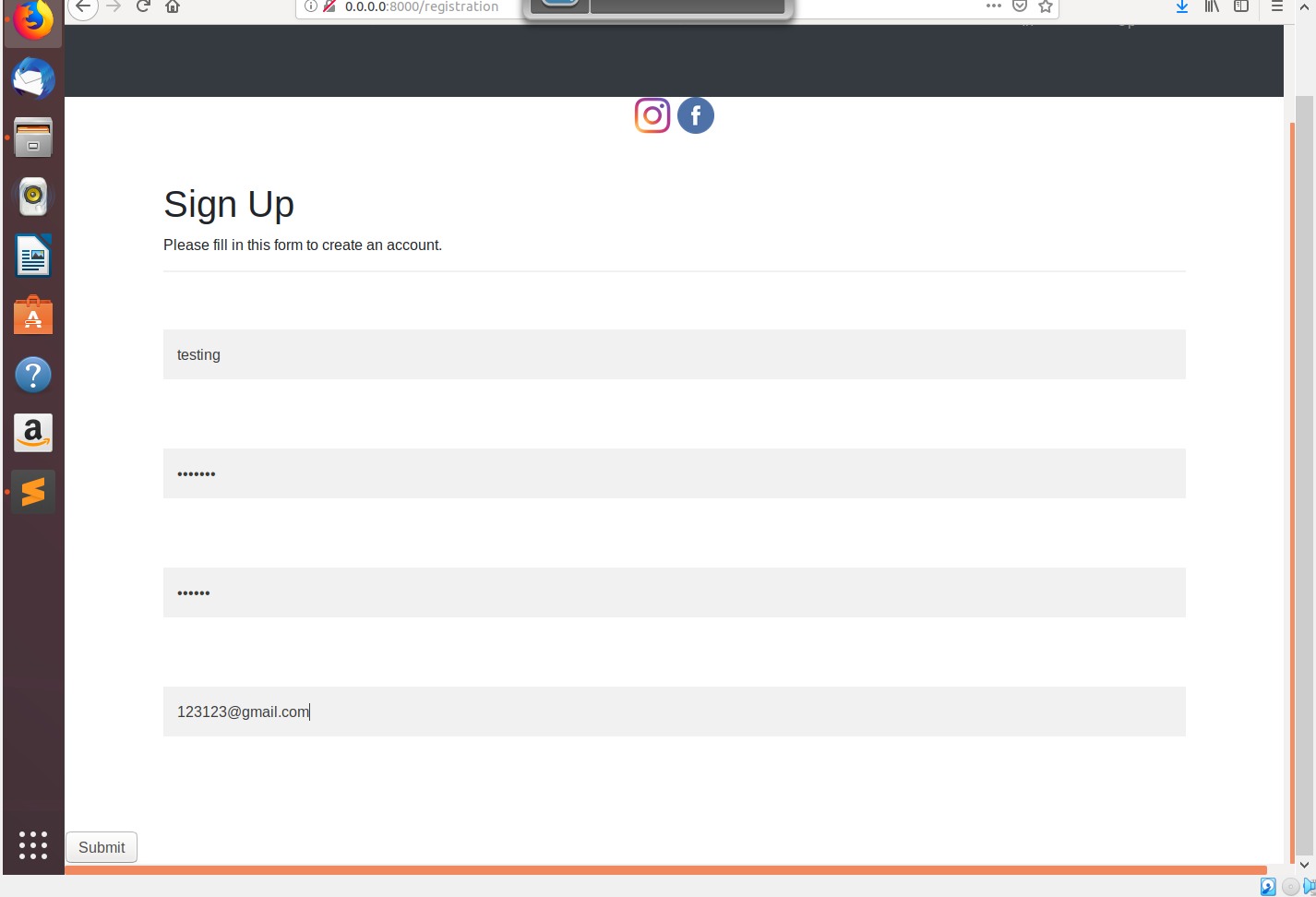






**Sign Up Page**

**This is the sign up page for our new customers to become our member.**



If the password input is not same as the confirm password , it will have a alert box “Pass word is not appropriate.”

Coding:

flash("Password is not appropiate","danger")



This registration method is to fill up all the required input box, and the registration data of the member will be uploaded and stored in database table ‘User\_Info’. If the password and repeat password data is correct, the page will redirect you to login page, and have an alert box saying that you have created an account successfully. And the database will store the user’s account data after the registration.

Coding:

@app.route('/registration', methods=['GET','POST'])

def registration():

request.method == 'POST'

username = request.form.get("username")

pwd = request.form.get("pwd")

pwdrepeat = request.form.get("pwd-repeat")

email = request.form.get("email")

if pwd is None:

print("Please register below")

return render\_template("registration.html")

if pwd == pwdrepeat:

# Prepare a cursor object

cursor = db.cursor()

# Insert data to database

sql ='''

INSERT INTO User\_info (user\_name, user\_pwd, user\_email)

VALUES ('%s','%s','%s')

'''

cursor.execute(sql%(username, pwd, email))

db.commit()

flash("You have created an account successfully","success")

return render\_template('login.html')

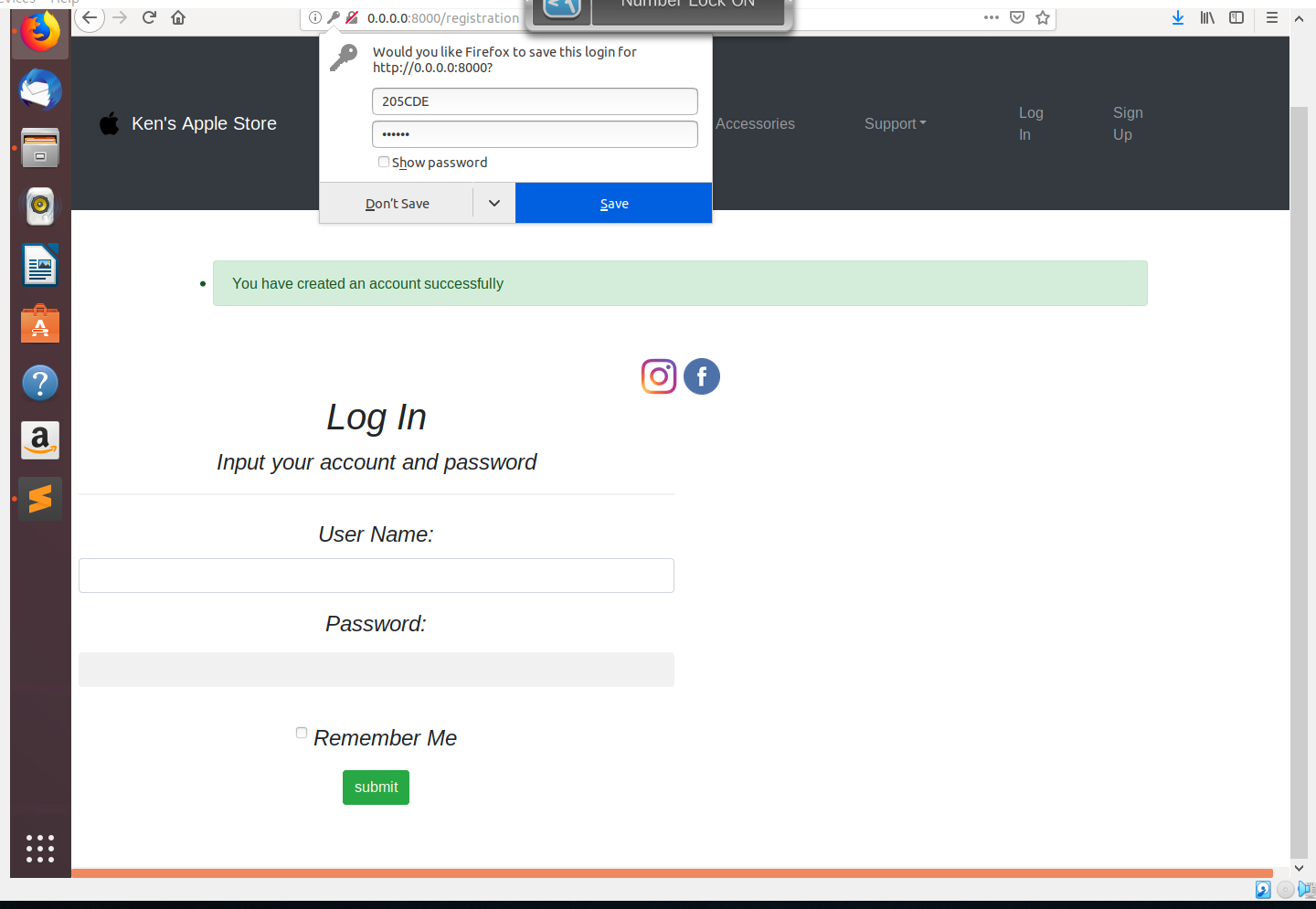
else:

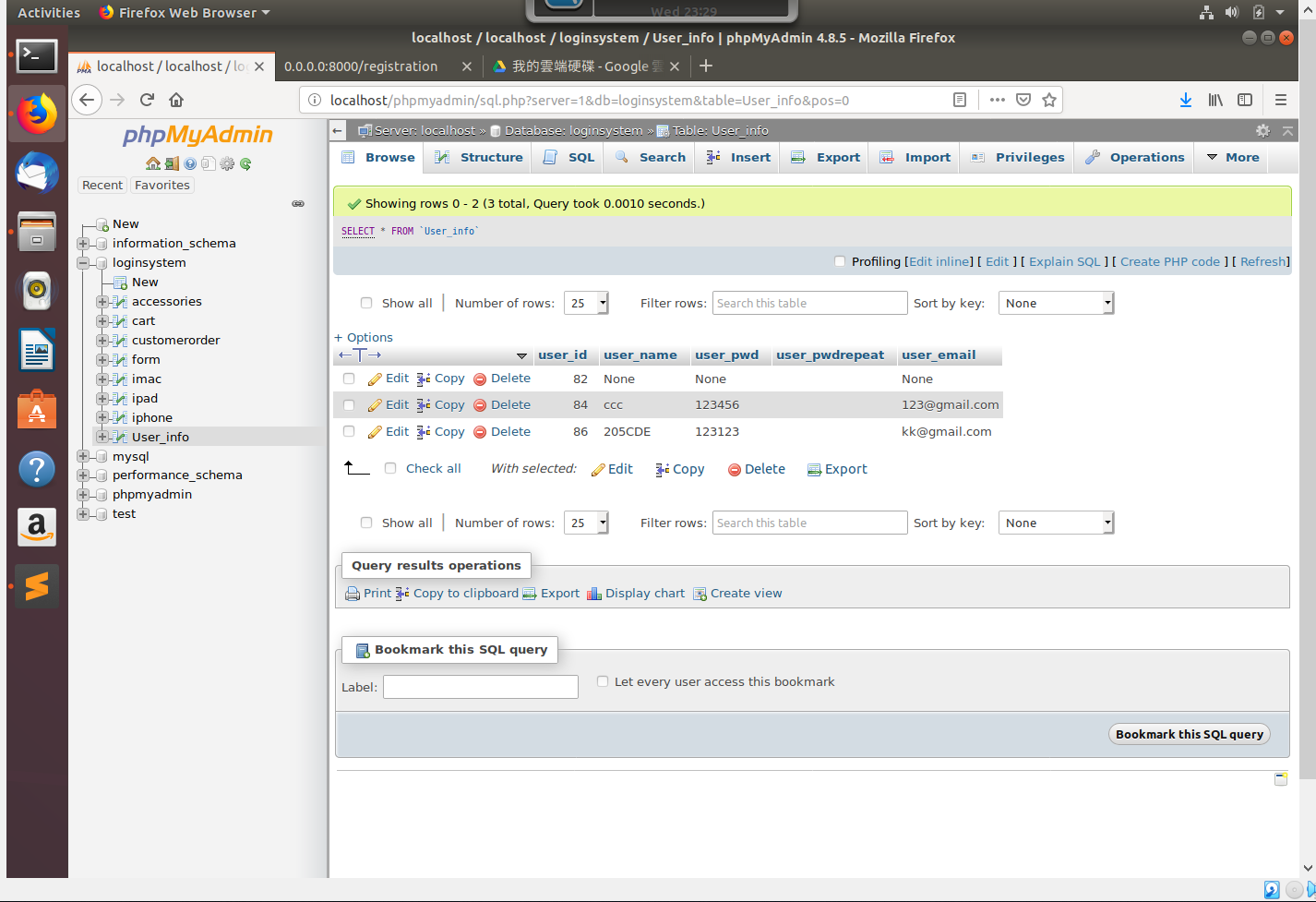
flash("Password is not appropiate","danger")

return render\_template('registration.html')

db.commit()

return render\_template("registration.html")



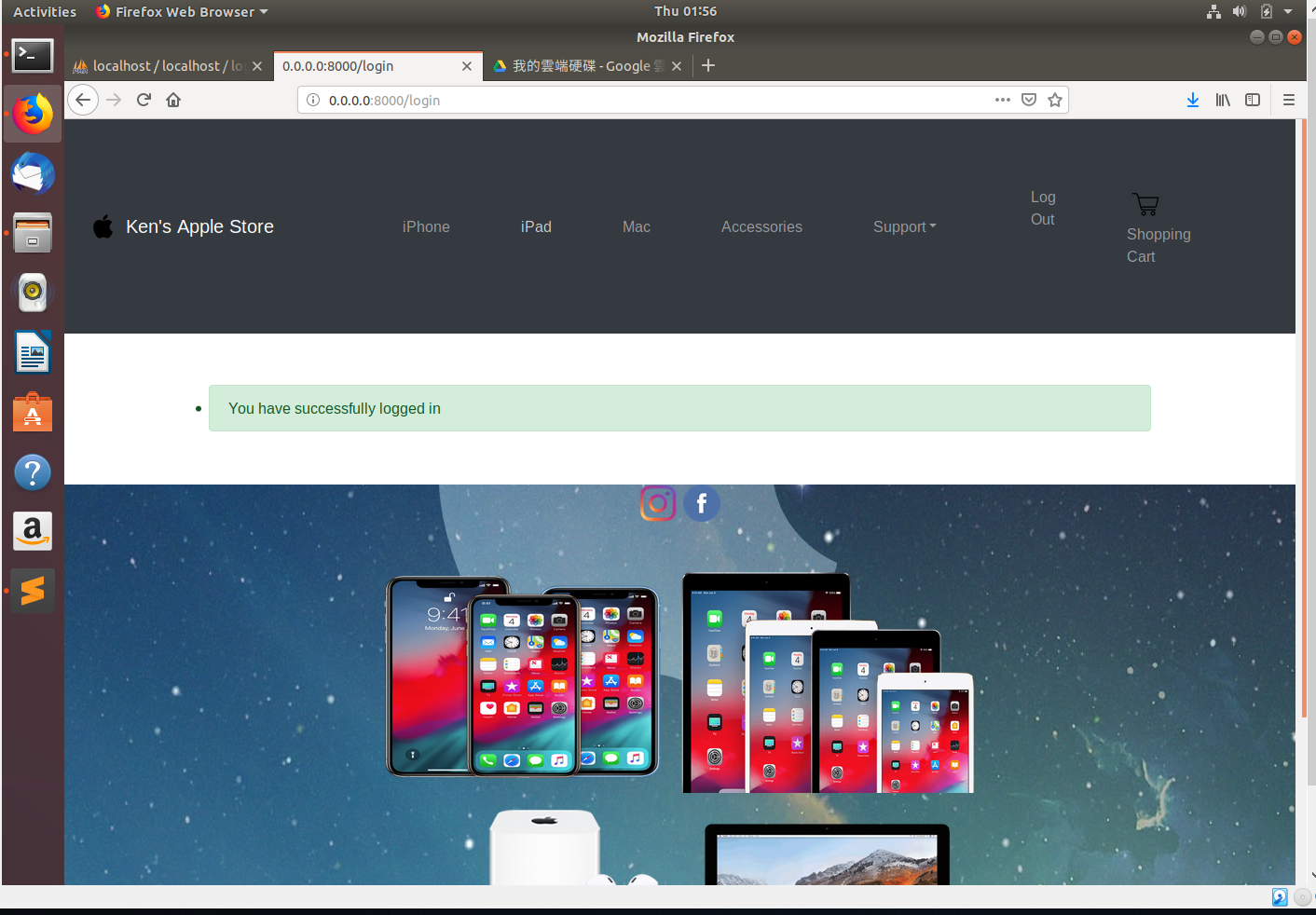


After successfully login, the session(‘user’) is used for storing the username data. The Cart page and Log out function will appear only if the customers was logged in.

**Coding:**

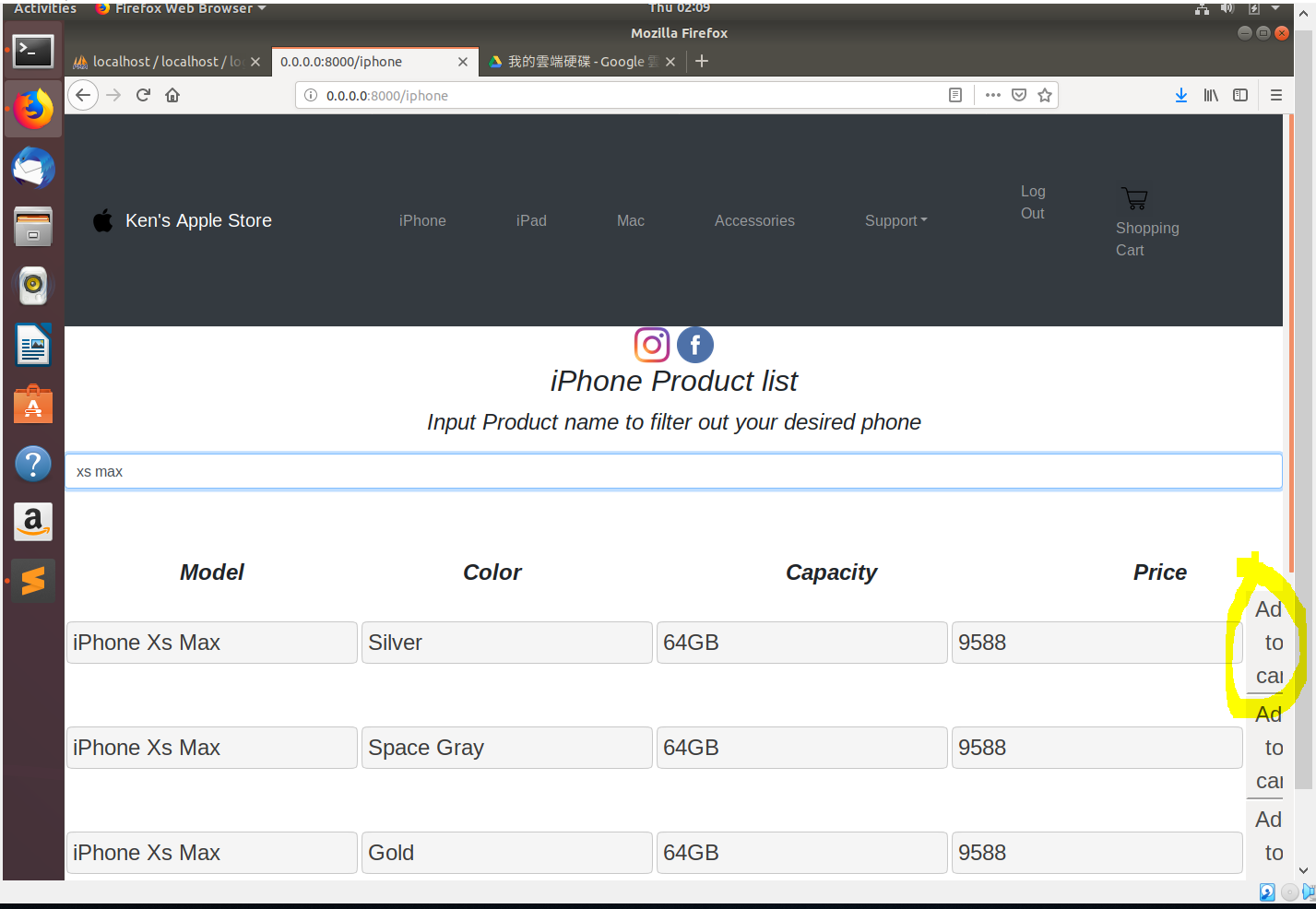
**flash("You have successfully logged in","success")**

**return render\_template('index.html')**



**Add to cart function**

In my products page, the product list is inserted from the database table. The product selected can be add to cart after pressing the add to cart button. The selected product list data will be insert into database table called ‘cart’ and the selected product will be insert into the shopping cart page.



**Coding:**

**Python flask Coding:**

@app.route('/addProduct', methods=['GET','POST'])

def addProduct():

request.method == 'POST'

model = request.form.get("model")

color = request.form.get("color")

capacity = request.form.get("capacity")

price = request.form.get("price")

cursor = db.cursor()

# Insert data to database

sql = '''

INSERT INTO cart (Model, Color, Capacity, Price)

VALUES ('%s','%s','%s','%s')

'''

cursor.execute(sql%(model, color, capacity, price))

db.commit()

flash("Successfully added product to your cart","success")

return render\_template('message.html')

**HTML Coding:**

Coding:

<table id="product">

{% for row in rows %}

<form action="{{url\_for('addProduct')}}" method = "POST">

<tr>

<td>

<input type="text" value="{{row[0]}}" name="model" readonly>

</td>

<td>

<input type="text" value="{{row[1]}}" name="color" readonly>

</td>

<td>

<input type="text" value="{{row[2]}}" name="capacity" readonly>

</td>

<td>

<input type="text" value="{{row[3]}}" name="price" readonly>

</td>

<td>

<button type="summit">Add to cart</button>

</td>

**Submit Order**

The sumbit button will take action to submit the requird product to database table’customerorder’ and the cart data will be deleted once the customer order was sent to database.

**Python flask function**

**Coding**

request.method == 'POST'

model = request.form.get("model")

color = request.form.get("color")

capacity = request.form.get("capacity")

price = request.form.get("price")

cursor = db.cursor()

sql = ("SELECT \* FROM cart")

cursor.execute(sql)

db.commit()

rows = cursor.fetchall()

sql ='''

INSERT INTO customerorder(Model,Color,Capacity,Price)

VALUES ('%s','%s','%s','%s')

'''

cursor.execute(sql%(model, color, capacity, price))

db.commit()

flash("You order is being accepted , the bill will send to you through email.","success")

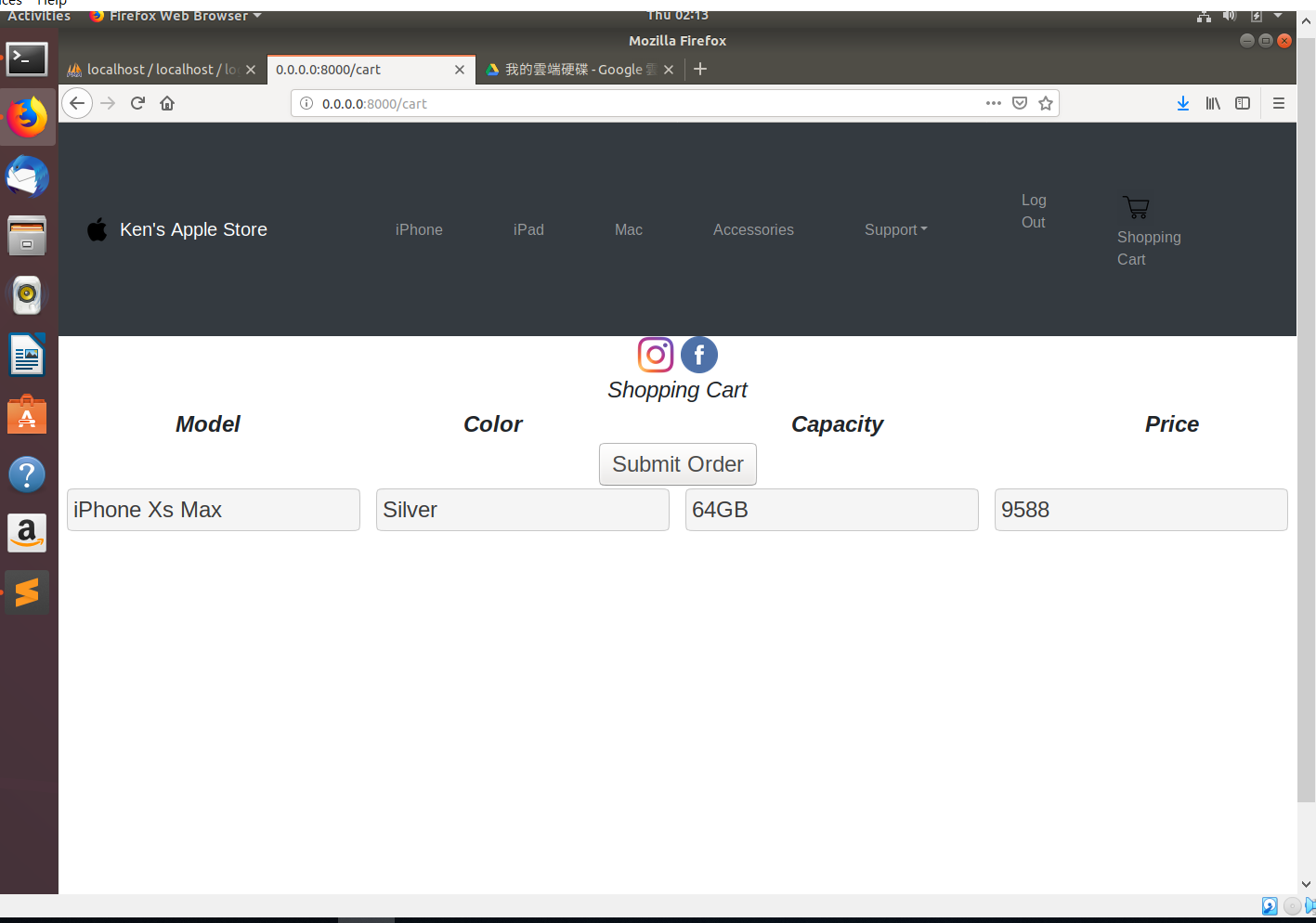
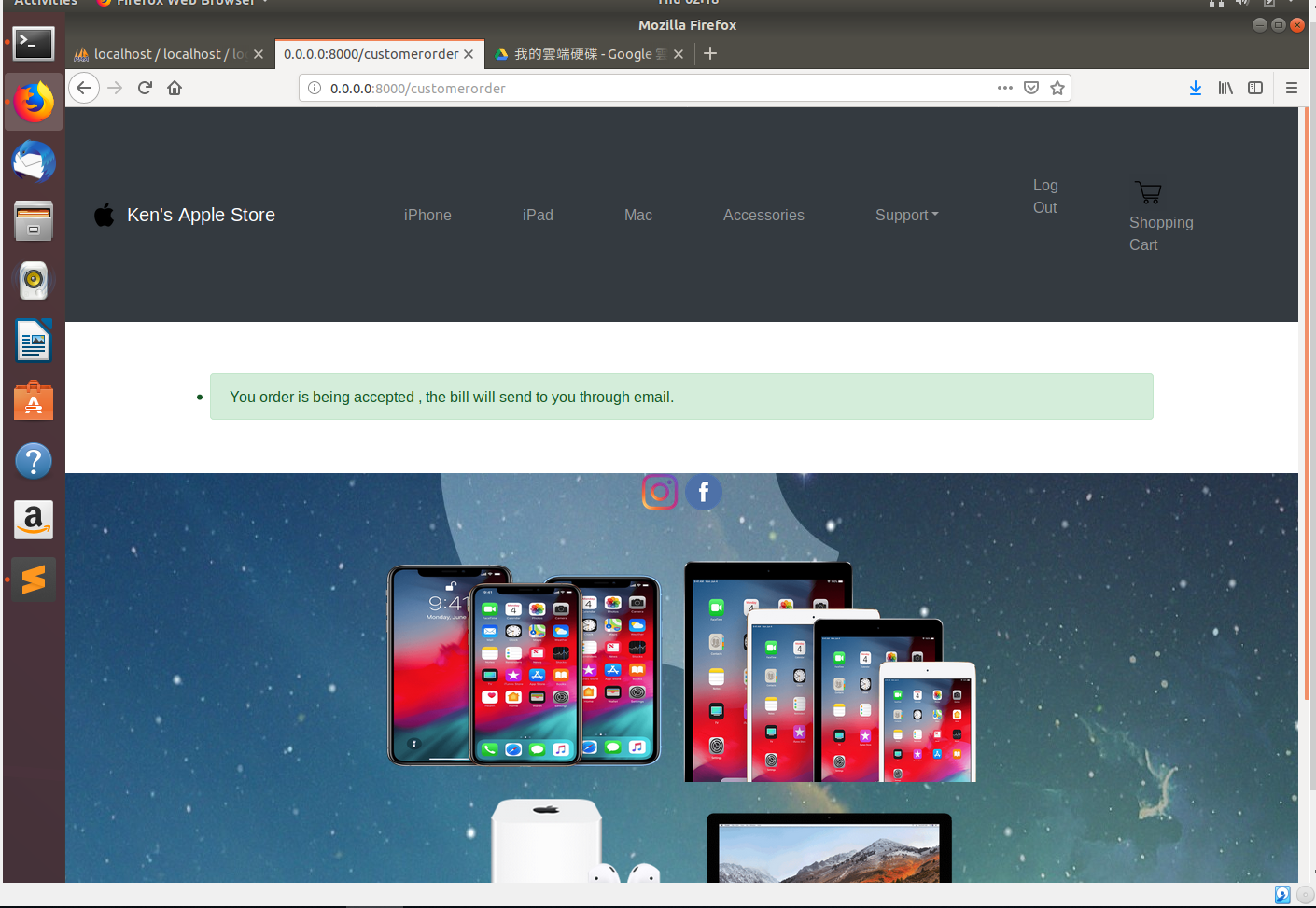
cursor = db.cursor()

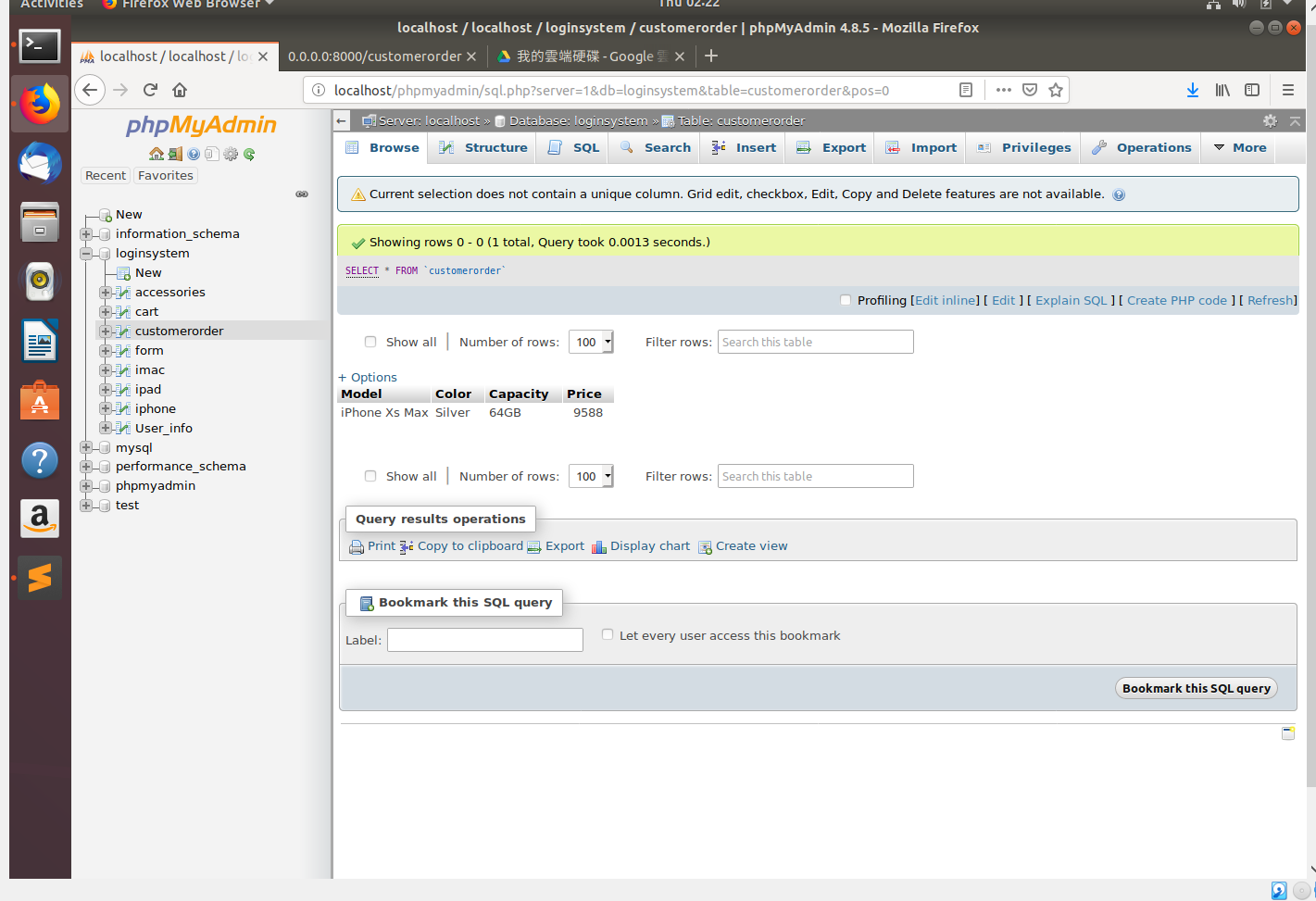
sql = ("DELETE FROM cart")

cursor.execute(sql)

db.commit()

return render\_template('index.html',rows = rows)



**Log Out Function**

Log out simply clear the session of the page.

