**205CDE Developing the Modern Web**

**Reflective Report**

**Student Name:**

**Wong Tsz Long**

**Student ID:**

**55202079**

**Due Date:**

**10th April 2019**

Programme Source code link:

GitHub:https://github.com/waltertsz/Wong-Tsz-Long/

Background: Walter’s Stationary is a stationary online shop which is selling some unique stationary to everyone.It can directly sell the products to the customers through the web.

In this project, I uesd Python, HTML, JavaScript,CSS and MySQL to complete the website. During the coding period, I have used different platform.It were Windows and Oracle VM VirtualBox in Linux Platform. Also I used Visual Studio Code and Sublime to edit it. Connecting MySQL database by XAMPP without any php code.

In this project, I have designed to use Template Inheritance to simplify the base layout in “base.html”. IIt can let the Nav-bar footer keep stay in every page which have the belows code.We may quote it in the front of the page

{% block head %}

and quote the bottom of the page.

{% endblock %}

In the other page, we only need to use the below codes to quote what you have designed at the base.html

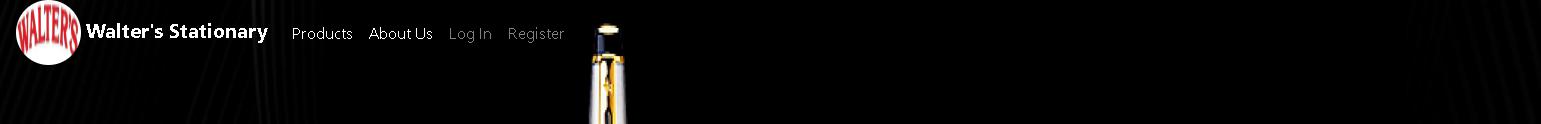
{% extends "base.html" %}

The others new details in different html page.We should code inside the statement

{% block main %}

xxxxxxxxx(new content)

{% endblock %}

On the nav-bar I have designed to use a colourless nav-bar with white text.It is because it can let the background photo fully show out and have a different style.

Because of the background is back colour so I have chose white text to show it clearly.

<nav class="navbar navbar-expand-sm bg-dark navbar-dark">

Here is the nav bar code.



Also to let the customer browse the products more convenient.The page have provided a search bar to let the customers search the product in the store.The below is the code to take action

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

def search(): #search function on the nav-bar , it can search by the products name.

form = OrderForm(request.form)

print('in search')

if 'search' in request.args:

print('in 2')

find = request.args['search']

cur = mysql.connection.cursor()

query\_string = "SELECT \* FROM products WHERE ptname LIKE %s ORDER BY pid ASC"

cur.execute(query\_string, ('%' + find + '%',))

This page required the customers to login to take order. So they should register an account

and login before they order a product. It required (Email,Username,Password,Address).

The below is the Register page



Here is the code:

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

def register():

form = RegisterForm(request.form)

if request.method == 'POST' and form.validate():

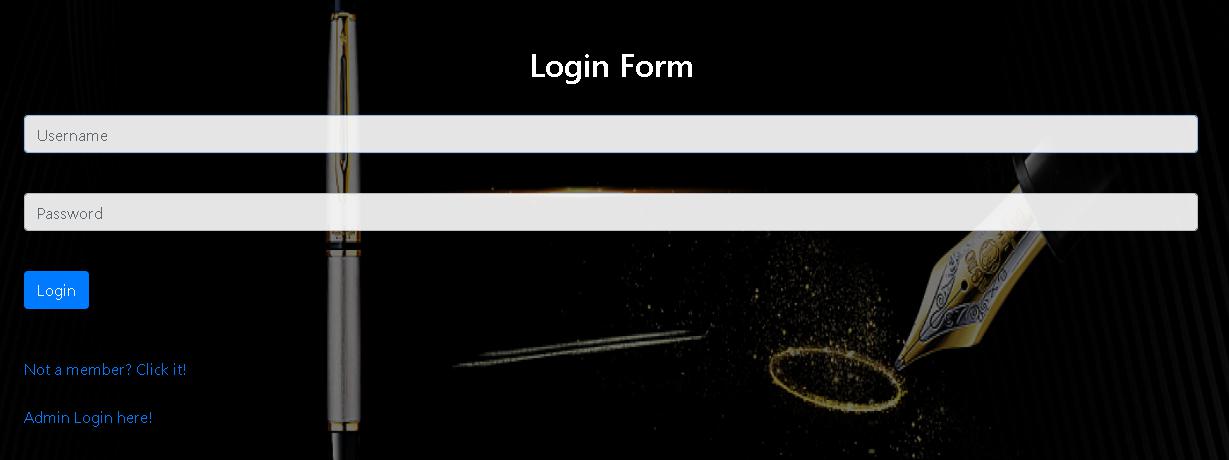
email = form.email.data

username = form.username.data

password = form.password.data

address = form.address.data

After register an account.We can go to the login page. Also the admin can login in the login page too.The admin login account and password is (admin,admin)



Here is the login code:

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

def login():

form = LoginForm(request.form)

if request.method == 'POST' and form.validate():

# GEt user form

username = form.username.data

# password\_candidate = request.form['password']

password\_candidate = form.password.data

and the adminlogin code

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

def adminlogin():

form = LoginForm2(request.form)

if request.method == 'POST' and form.validate():

# GEt user form

username = form.username.data

# password\_candidate = request.form['password']

password\_candidate = form.password.data

It also have a validation with the database

here is the code:

if result > 0:

# Get stored value

data = cur.fetchone()

password = data['adpw']

uid = data['adid']

name = data['adname']

if password\_candidate == password:

session['adid'] = uid

session['adname'] = name

print('Admin :',session['adname'],' sucess login')

return redirect(url\_for('admin'))

else:

flash('Incorrect password', 'danger')

return render\_template('adminlogin.html', form=form)

else:

flash('Username not found', 'danger')

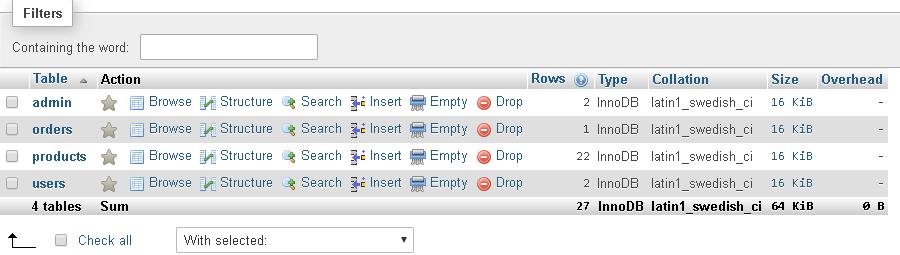
# Close connection

cur.close()

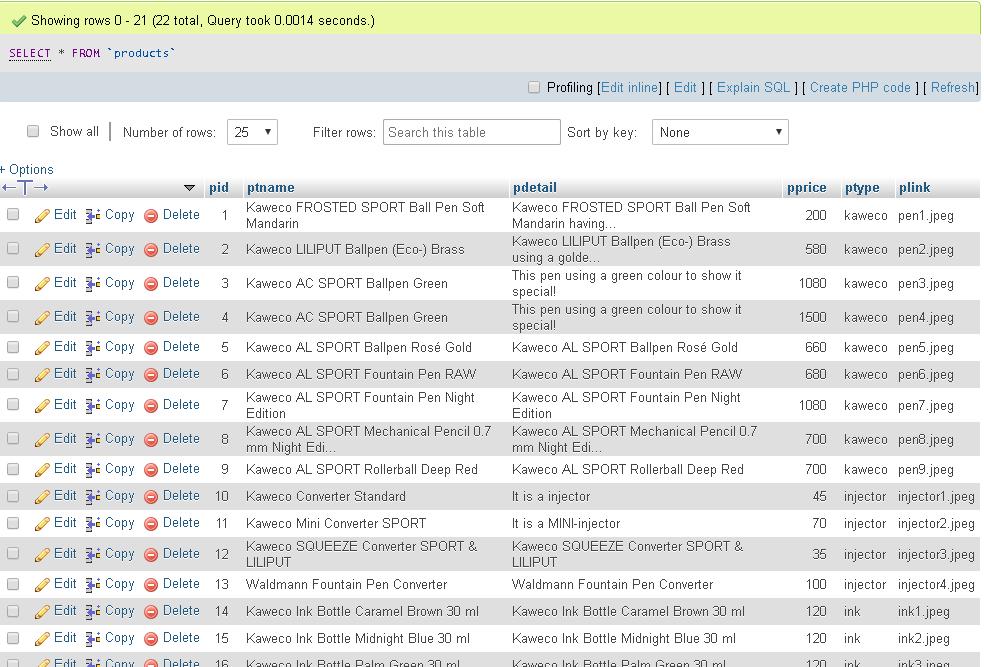
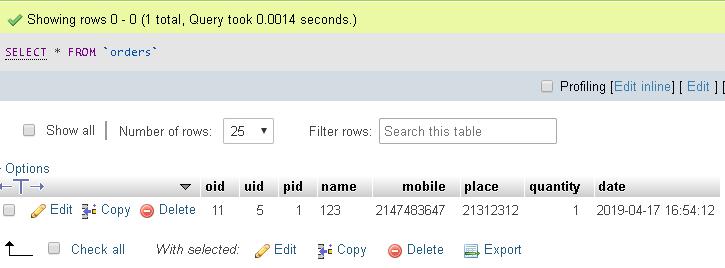
return render\_template('adminlogin.html', form=form)

return render\_template('adminlogin.html', form=form)

The below is the database table list.



It will store the data from the customers they register a account,ordered products and the product list. The below is the further information in the Table.



Here are lots of product.How can we show it?

I have used below codes to link with the data base

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

def store():

pdtypea = mysql.connection.cursor()

values = 'kaweco'

pdtypea.execute("SELECT \* FROM products WHERE ptype=%s ORDER BY RAND() ", (values,))

kaweco = pdtypea.fetchall()

pdtypeb = mysql.connection.cursor()

values = 'injector'

pdtypeb.execute("SELECT \* FROM products WHERE ptype=%s ORDER BY RAND() ", (values,))

injector = pdtypeb.fetchall()

pdtypeb = mysql.connection.cursor()

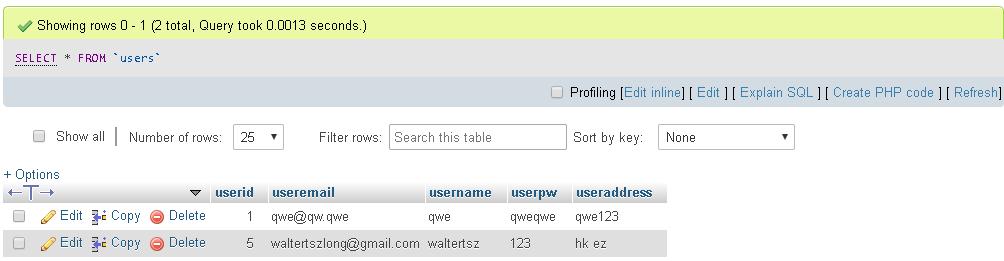
values = 'ink'

pdtypeb.execute("SELECT \* FROM products WHERE ptype=%s ORDER BY RAND() ", (values,))

ink = pdtypeb.fetchall()

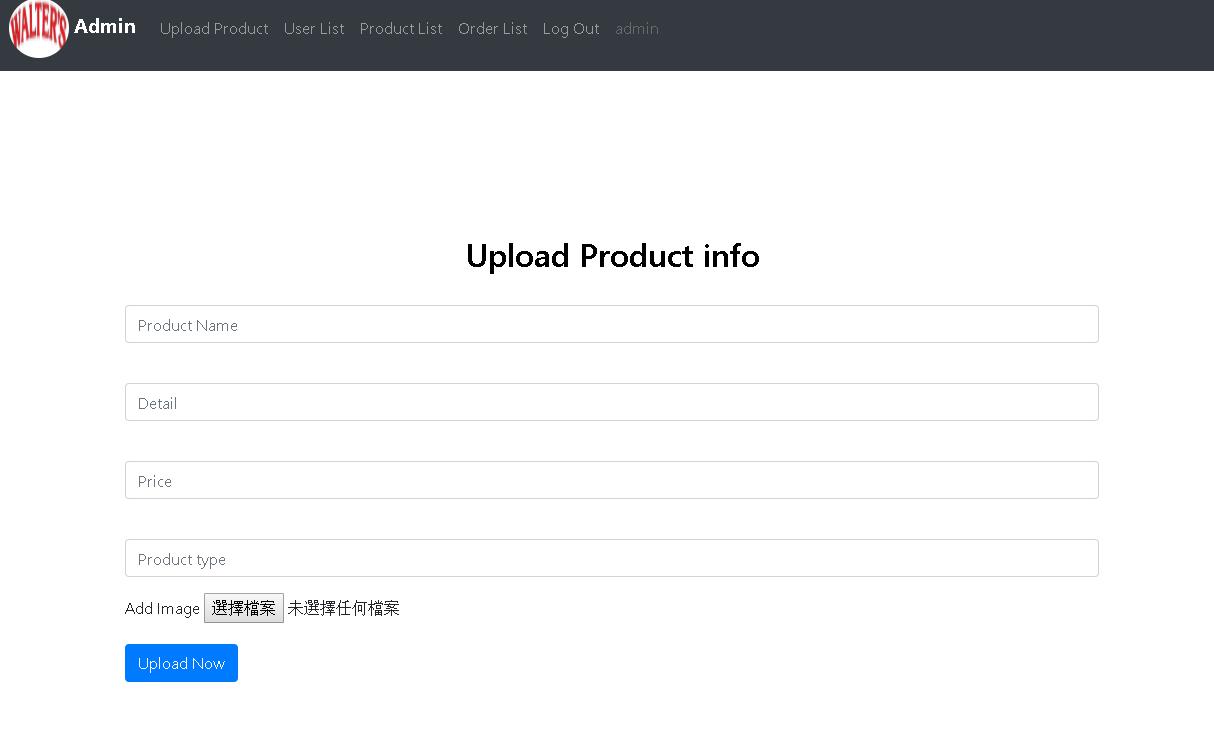
return render\_template ('store.html', kaweco=kaweco, injector=injector, ink=ink)

This code can get the data from the database and define it by type.



For the products, admin can also upload new product from the admin page.

The upload product info is the below picture.

The code here is the upload products function

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

def upload():

form = UploadProductForm(request.form)

if request.method == 'POST' and form.validate():

name = form.name.data

detail = form.detail.data

price = form.price.data

ptype = form.ptype.data

file = request.files['picture']

if name and detail and price and ptype and file:

pic = file.filename

photo = pic.replace("'", "")

picture = photo.replace(" ", "\_")

if picture.lower().endswith(('.png', '.jpg', '.jpeg')):

savepic = photos.save(file, folder=ptype)

if savepic:

curs = mysql.connection.cursor()

curs.execute("INSERT INTO products( ptname, pdetail, pprice, ptype, plink)"

"VALUES(%s, %s, %s, %s, %s)",

(name, detail, price, ptype, picture))

mysql.connection.commit()

flash('Product added successful', 'success')

return render\_template ('upload.html', form=form)

else:

flash('Picture not save', 'danger')

return render\_template ('upload.html', form=form)

else:

flash('File not supported', 'danger')

return render\_template ('upload.html', form=form)

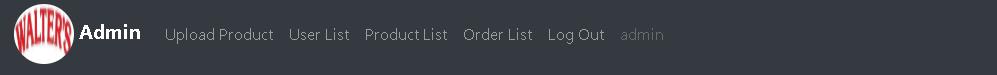
else:

flash('Missing Information', 'danger')

return render\_template ('upload.html', form=form)

return render\_template ('upload.html', form=form)

Admin can add products for the future. It can also set the price of the new product too.



The nav bar at admin page is also different.

It is black colour with white text.

Also it shows different kinds of function from the data base.

I.E. (User List, Product List ,Order List)

After Admin or customers finish their works.They can Logout to protect their account and privacy safety.

The Logout code:

@app.route("/logout")

def logout():

session.clear()

flash('Log out Success', 'success')

return render\_template ('index.html')

After Logout it will clear their information.

On reflection, I think I need to take more attention on the security problems.

Such as, we can access the admin page by the link after adding a /admin.

It is a serious security problems. It maybe occur some privacy problems.

Others people can watch the details of our company. It included the personal information of the customers.I think it should be improve next time.

On the database, I should create a better database for products,the product details are not really reflected how the products is good. Also, I think I should input more product type to make the website more attractive.

On the UI, I think it should be one page content to let the customers no need to press to much button to do their process. Also, it can get more animation on it to attract more customers to browse the page.I think it also can show which products is on sale at the homepage.Discount function is a really attractive funcion to all customers. It is because when the products on sales. It will inspire the customers should buy it immediately. It can help our shop to increase profits.

Finally, I hope I can have a chance to create a more gorgeous online shop for my own business.