205 Developing the modern web: reflective report

**THE FOOL COMP**

Name: Lam Yat Long

Student number: 5713 0051

GitHub link: <https://github.com/TFJoKer122/reassessfinal.git>

Video :<https://youtu.be/OJ7zWVLtf04>

# Content Page

[Content Page 2](#_Toc140503737)

[Introduction 3](#_Toc140503738)

[HTML 4](#_Toc140503739)

[CSS 8](#_Toc140503740)

[CSS coding example: 9](#_Toc140503741)

[JavaScript 11](#_Toc140503742)

[Javascript code example for the cart: 11](#_Toc140503743)

[Flask and python 14](#_Toc140503744)

[Flask code import 14](#_Toc140503745)

[Flask code for connect to sql database: 14](#_Toc140503746)

[Flask code for render template: 15](#_Toc140503747)

[SQL 17](#_Toc140503748)

[Reflection 20](#_Toc140503749)

# Introduction

In this project, I run a shop call TheFoolComp to sell the computer stuff in my website. I make this website since I love playing computer games and using some expensive setup. So I want more people to use it since it feels better to play game when you compare to others.

I used serval thing that I learn from the course in below.

[HTML](#_59d53di06oll)

[CSS](#_igsnrmahkcbv)

[JavaScript](#_10lct4swe9ru)

[Flask and Python](#_548uhrdycc1f)

[SQL](#_ffkt0asiz5g3)

Within the conclusion, the report will be examining why the site is built and how the website will be created within the future.

# HTML

HTML is a markup language that is used to create and structure the content of web pages and applications. Also, it is fundamental to every website. The webpage and application can be connected to the World Wide Web. We can use HTML tags, such as headings, links, and images, that are used to define the structure and content of a webpage.

In my website, all pages are using HTML to write. The main website serves as a homepage to connect all other pages. The header of my webpage "The Fool Comp" contains an icon and title and path to other pages. In the body section, some divs are used to contain the photos. We need to use hyperlink for the path to go to other pages and hyper link are stored in the div using the ""{{ url\_for('') }}"" tag. When the div is clicked, the subpage will open. At the lower part of the webpage, I add a return to top button for user to back quickly to the top.

一張含有 文字, 螢幕擷取畫面, 字型, 設計 的圖片

自動產生的描述

<body>

<div class="form-container">

<form method="POST" action="">

<h1>Welcome to TheFoolComp.</h1>

<h3>login now</h3>

{{ form.hidden\_tag() }}

{{ form.username }}

{{ form.password }}

{{ form.submit }}

<p></p>

<a href="{{ url\_for('register') }}">don't have an account? register now</a>

</form>

</div>

</body>

一張含有 文字, 螢幕擷取畫面, 平面設計, 圖形 的圖片

自動產生的描述

<section class="home" id="home">

<div class="content">

<h3>Wecome to TheFoolComp.</h3>

<span>Here we sale the popular esports mouse ,keyboard and

Headset/earphone

</span>

<p>Hope u can find the things you need.</p>

<a href="#" class="btn">shop now</a>

</div>

</section>

一張含有 文字, 螢幕擷取畫面, 平面設計, 圖形 的圖片

自動產生的描述

<button class="open-button" onclick="openForm()"><a href="#" class="fas fa-shopping-cart"></a></button>

<div class="form-popup" id="myForm">

<form action="/action\_page.php" class="form-container">

<section class="shopcart">

<h2 class="section-header">CART</h2>

<div class="cart-items">

</div>

<div class="cart-total">

<strong class="ctotal">Total</strong>

<span class="ctotal-price">$0</span>

</div>

<button class="btn-purchase" type="button" id="myBtn"><a class="fas fa-shopping-cart"></a>buy</button>

<button type="button" class="btn cancel" onclick="closeForm()">Close</button>

</section>

</form>

一張含有 文字, 字型, 標誌, 符號 的圖片

自動產生的描述

function scrollFunction() {

if (document.body.scrollTop > 20 || document.documentElement.scrollTop > 20) {

mybutton.style.display = "block";

} else {

mybutton.style.display = "none";

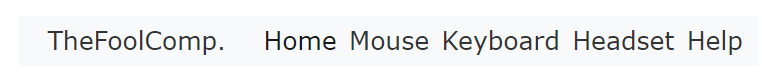
}

}

# CSS

CSS is a language that enhances the visual appeal of a webpage. It works with HTML and XML. CSS is a tool that to change and upgrade the way a website looks by changing its layout, typography, color scheme, and other visual elements. It enables developers to maintain by making sure that all web pages on a site look the same. It determines the design of a page. It is unclear what is written on it. Therefore, a developer can apply the same style to multiple pages or even the entire website. CSS helps developers create and control web page elements, fonts, colors, and animations. It allows for the customization of HTML elements with style code, resulting in a visually distinct appearance. Rules for style include selectors and statements. Selectors lead to choose the parts of a webpage we want to make design on it, while declarations describe 100% that equal to how the parts look like we want. CSS can be added to an HTML document with "style" tag or by linking to a CSS file in the static file when we use flask with the "link" tag(example :<link rel="stylesheet" type="text/css" href="{{ url\_for('static', filename='css/style.css') }}">). If you keep CSS and HTML separate, web developers can create web pages and applications that are easier to manage and scale. In my webpage, I separate the CSS into 2 file which is loginstyle.css and style.css. loginstle.css is the CSS file for the login and out page. The style.css is for the main webpage for showing. And also, my website also use CSS inside the html file like the return to top button.

And I also use bootstrap into the apps for style. which mainly in frontpage



一張含有 文字, 字型, 白色, 螢幕擷取畫面 的圖片

自動產生的描述

<nav class="navbar navbar-expand-lg bg-light">

<div class="container-fluid">

<a class="navbar-brand" href="{{ url\_for('program') }}">TheFoolComp<span>.</span></a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNav">

<ul class="navbar-nav">

<li class="nav-item">

<a class="nav-link active" aria-current="page" href="{{ url\_for('program') }}">Home</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{{ url\_for('mouse') }}">Mouse</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{{ url\_for('keyboard') }}">Keyboard</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{{ url\_for('headset') }}">Headset</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{{ url\_for('contaxt') }}">Help</a>

</li>

</ul>

</div>

</div>

</nav>

## CSS coding example:

|  |  |
| --- | --- |
| .open-button {  background-color: #555;  color: white;  padding: 16px 20px;  border: none;  cursor: pointer;  opacity: 0.8;  position: fixed;  bottom: 23px;  left: 28px;  width: 280px;  } | .form-container form .error-msg{  margin:10px 0;  display: block;  background: crimson;  color:#fff;  border-radius: 5px;  font-size: 20px;  padding:10px;  } |
| .shopcart{  top:0;left:0;right:0;  background: #e20505;  padding:2rem 9%;  display: flex;  justify-content: space-between;  z-index: 1000;  box-shadow : 0 .5rem 1rem rgba(0,0,0,.1);  } |  |

# JavaScript

JavaScript, a programming language created by Netscape in 1995, has picked up gigantic ubiquity on the web. JavaScript is a widely used programming language. It can be the engagement of web pages by empowering clients to associated with them within the web. The page reacts to client intelligent, such as clicking or writing. It can change the substance on the page without utilizing any devices or assets. It too guarantees that the data entered shapes is exact. JavaScript can modify the appearance of a webpage. The content structure is composed of HTML or XML. JavaScript can modify the appearance and behavior of a webpage based on client activities or other occasions. JavaScript can be added to HTML using the `<script>` tag to link the javascript file. On my webpage, the shopping cart function is located at the bottom of the page. This allows customers to easily click on the button and calculate the total price of the items in their cart. The function also calculates the total price by adding up the prices of all the items in the cart.

## Javascript code example for the cart:

function addItemToCart(prodname, price, Cimage) {

var cartRow = document.createElement('div')

cartRow.classList.add('cart-row')

var cItem = document.getElementsByClassName('cart-items')[0]

var cartItemNames = cItem.getElementsByClassName('cart-item-prodname')

for (var i = 0; i < cartItemNames.length; i++) {

if (cartItemNames[i].innerText == prodname) {

alert('This item is already added to the cart!')

return

}

}

var cartRowContents = `

<div class="cart-item cart-column">

<img class="cart-item-image" src="${Cimage}" width="100" height="100">

<span class="cart-item-prodname">${prodname}</span>

</div>

<span class="cart-price cart-column">${price}</span>

<div class="cart-quantity cart-column">

<input class="cartnuminput" type="number" value="1">

<button class="btn rbutt" type="button">REMOVE</button>

</div>`

cartRow.innerHTML = cartRowContents

cItem.append(cartRow)

cartRow.getElementsByClassName('rbutt')[0].addEventListener('click', removeitem)

cartRow.getElementsByClassName('cartnuminput')[0].addEventListener('change', numchange)

}

function updcart() {

var cartItemContainer = document.getElementsByClassName('cart-items')[0]

var cartRows = cartItemContainer.getElementsByClassName('cart-row')

var total = 0

for (var i = 0; i < cartRows.length; i++) {

var cartRow = cartRows[i]

var priceElement = cartRow.getElementsByClassName('cart-price')[0]

var qel = cartRow.getElementsByClassName('cartnuminput')[0]

var price = parseFloat(priceElement.innerText.replace('$', ''))

var quantity = qel.value

total = total + (price \* quantity)

}

total = Math.round(total \* 100) / 100

document.getElementsByClassName('ctotal-price')[0].innerText = '$' + total

}

# Flask and python

Flask simplifies the creation of web applications using Python. In 2010, Armin Ronacher accomplished the feat. Purpose: Designing a user-friendly platform to create website. This object has a simple design. The Flask framework is a tool for developing web applications with Python. WSGI enables compatibility with Apache, Nginx, and Gunicorn servers. It is often import SQLAlchemy to simplify data retrieval and streamline template creation. Flask offers a plethora of tools for developing web applications. The server manages testing, URLs, requests, and responses. This software helps to create web forms, utilize cookies and sessions, and facilitate file uploads. Flask is a simple and versatile framework. The framework is flexible and can be easily customized for a variety of web applications. Flask can easily to set up by the beginner. File we need to set up (/) to set up the main page of out webpage after you input the all the pages into the app.py, you are almost done the outline of the Flask. In Flask we need to set up a file call static to put in our ,css .js and phot for the webpage for the webpage to read. In the import part, I import many things inside it. For example flask\_bcrypt is for encrypting and decrypting the information of the password of the database. If the hacker hack the database they need time to encrypt it first. This will make us to shut down the server first to prevent the leak of customer information.

## Flask code import

from flask import Flask, render\_template, url\_for, redirect

from flask\_sqlalchemy import SQLAlchemy

from flask\_login import UserMixin, login\_user, LoginManager, login\_required, logout\_user, current\_user

from flask\_wtf import FlaskForm

from wtforms import StringField, PasswordField, SubmitField

from wtforms.validators import InputRequired, Length, ValidationError

from flask\_bcrypt import Bcrypt

## Flask code for connect to sql database:

class User(db.Model, UserMixin):

id = db.Column(db.Integer, primary\_key=True)

username = db.Column(db.String(20), nullable=False, unique=True)

password = db.Column(db.String(80), nullable=False)

class RegisterForm(FlaskForm):

username = StringField(validators=[

InputRequired(), Length(min=4, max=20)], render\_kw={"placeholder": "Username"})

password = PasswordField(validators=[

InputRequired(), Length(min=8, max=20)], render\_kw={"placeholder": "Password"})

submit = SubmitField('Register')

def validate\_username(self, username):

existing\_user\_username = User.query.filter\_by(

username=username.data).first()

if existing\_user\_username:

raise ValidationError(

'That username already exists. Please choose a different one.')

## Flask code for render template:

@app.route('/')

def program():

return render\_template('program.html')

@app.route('/mouse')

def mouse():

return render\_template('mouse.html')

@app.route('/keyboard')

def keyboard():

return render\_template('keyboard.html')

@app.route('/headset')

def headset():

return render\_template('headset.html')

@app.route('/contaxt')

def contaxt():

return render\_template('contaxt.html')

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)

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

def register():

form = RegisterForm()

if form.validate\_on\_submit():

hashed\_password = bcrypt.generate\_password\_hash(form.password.data)

new\_user = User(username=form.username.data, password=hashed\_password)

db.session.add(new\_user)

db.session.commit()

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

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

# SQL

SQL stands for Structured Query Language. SQL is a common way of organizing and controlling data in databases that have links between different pieces of data. SQL could be a instrument that creates it simple to form and adjust tables, which are utilized to organize data. SQL is a helpful tool to ask questions about information in tables from various databases. It can be utilized in both little and enormous databases that are utilized by enormous companies. Many common database systems in the business environment, such as MySQL, PostgreSQL, Oracle, and Microsoft SQL Server, can work with SQL. SQL is a useful language that can be used to manage data in many ways. Knowing how to work with databases and data-driven applications is crucial for individuals who work with them. In in assessment, I use .db file as my database .In the website the database mainly work on the login system. When user create user account with username and password in the register page, it will create the data in the database. When we create the data in the database, the password will be created and encrypted. When the users want to login, the flask will read the data from the database and decrypt the password and let the user login.

In the future, we can link more database into the product, and user information. Like we can add buying history, saving their shopping cart. This can increase the user experience when they always use our website. We can also import our product item into the database to check how many we have sell and which one is popular.

一張含有 文字, 名片, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

SELECT id, username, password

FROM user;

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 字型, 設計 的圖片

自動產生的描述

Example of importing table

import sqlite3

conn = sqlite3.connect('product.db')

c = conn.cursor()

c.execute("""CREATE TABLE request(

username varchar(20),

LName text,

email text,

Subject text

)""")

conn.commit()

conn.close()

# Reflection

In the web project, I found that write web is difficult and interesting. But I still do something not good in writing the website. In the database, I found that it is very hard to connect to online database like XAMPP. So I only connect to a .db file as my database. And database is the hardest part in the web. Since all the element in the web should connect to the database. But it is very hard to do it when compared to HTML, CSS. And the way to do it between window and ubuntu is different. Some of the code work on window but not work on ubuntu. This will increase the work time in this part. Like I use the SQLite3 in window to show the database, but I can’t use it in the ubuntu. This makes me need to use window to present my database in the video. Talk to Ubuntu, I also faced some bug in Ubuntu. I don’t know why my screen of ubuntu can’t enlarge it and it will keep so small.

In the future, I need to increase more element of database on the website. In the current, the product and report form are still not connected to database. Second, the web should connect to online database. The data should auto update to another device which using my web. And I should put the product in the database to check the number of product.

In JavaScript, I should connect the shopping cart into the flask and database tot let them use it. this can let them save the shopping cart when they switch webpage.

In CSS and bootstrap, I should have a better design to absorb customers. The design should increase more technology and esports feels.

In adding pages, I think a customer detail page should be created. To let them check their information, shop history and editing their profile or password.