

Coursework Report

Danni Xu

40332963@live.napier.ac.uk

Edinburgh Napier University - Advanced Web Technologies (SET09103)

Abstract

In modern life, the Internet has penetrated into every corner of our lives. We can find what we need from the Internet. Certainly, we can also use it to advertise the product. The purpose of this report is to introduce a prototype of a food company web application which was created by using python flask. It also used HTML, CSS, JavaScript, bootstrap, sqlite3 etc. Next, basic introduction, design architected, the functions and envisioned implementation of this application will be described.

Keywords – python, flask, sqlite3, food company, request, templates, static files

1 Introduction

The theme of this web application is Black Duck Zhou (a food company). This is a simple company website to introduce the company and advertise the products. The main purpose of this website is to introduce the development of the company, to promote its products and to collect feedback and comments from customers.

2 Design:

2.1 Layout Analysis:

Home page: includes Navigation bar, Carousel map and body content.

Navigation connects four parts, home page, product page, comment page and user login page. There is also a search bar on the last side of the navigation bar.

Carousel map: Shows the company's operating philosophy.

The main part introduces the development process of the company.

Products page: There is a pull-up button at the bottom of the navigation bar. After the click, the left side is an introduction to the food's workmanship and efficacy; the right side is the online purchase channel link. The body part is products display.

Comments page: This is a comment area that collects feedback from users on the product, where users can comment and see their comments below.

Login page: Users can login here, if login successfully, you can enter the home page, or still stay on the login page.

2.2 Technical Analysis:

The main technology used in the web-app: python-flask, Sqlite3, HTML, CSS, JavaScript, Bootstrap.

Using app. route to add more routes to the web-app, and write functions then add decorator for each one to make the function into routs, when a web-flask is running, that app will be accessible in a web browser on the exact network.

Using HTML, CSS, Bootstrap to decorate web pages, make the page more beautiful and make users have better use experience.

Using request, redirect, flash realizes the user login, prompt and the page skip. If you enter the correct username and password, you'll jump to the home page, or there will be a "Try Again" hint that appears below.

Using template inheritance allows each page to have the same navigation bar and footer. So that the whole website has integrity.

It also provides own error page, If you enter a path that doesn't exist, it will jump to the error page.

Using SQLite3 to connect to the database, and implement the comment function. Then, the source code which implemented the function will be displayed.

Listing 1: Example code

```

1 from flask import Flask, g, render_template, flash, request, ←
    session, redirect, url_for, abort
2 import sqlite3
3 import os
4
5 from flask import Flask
6
7 app = Flask(__name__)
8 app.config.from_object(__name__)
9 db_location = 'var/index.db'
10 app.secret_key = os.urandom(24)
11
12 def get_db():
13     db = getattr(g, 'db', None)
14     if db is None:
15         db = sqlite3.connect(db_location)
16         g.db = db
17     return db
18

```

```

19 @app.teardown_appcontext
20 def close_db(exception):
21     db = getattr(g, 'db', None)
22     if db is not None:
23         db.close()
24
25 def init_db():
26     with app.app_context():
27         db = get_db()
28         with app.open_resource('schema.sql', mode='r') as f:
29             db.cursor().executescript(f.read())
30         db.commit()
31
32 @app.before_request
33 def before_request():
34     g.db = get_db()
35
36 @app.teardown_request
37 def teardown_request(exception):
38     g.db.close()
39
40 @app.route('/comments')
41 def comments():
42     cur = g.db.execute('select title, text from entries order by id DESC')
43     entries = [dict(title=row[0], text=row[1]) for row in cur.fetchall()]
44     return render_template('comments.html', entries=entries)
45
46 @app.route('/comments/add', methods=['POST'])
47 def add_entry():
48     g.db.execute('insert into entries (title, text) values (?, ?)', [request.form['title'], request.form['text']])
49     g.db.commit()
50     flash('Submit successfully !')
51     return redirect(url_for('comments'))
52
53 if __name__ == "__main__":
54     init_db()
55     app.run(host='0.0.0.0')
56
57 }

```

Listing 2: to connect database

```

1 from index import init_db
2 init_db()

```

Listing 3: Create schema to define tables in the database

```

1 drop table if exists entries;
2 create table entries (
3     id integer primary key autoincrement,
4     title string not null,
5     text string not null
6 );

```

2.3 Style analysis:

The style of this website is succinct. In the home page and product page using yellow elements to stimulate the user's appetite, with pictures, stimulate the browser's desire to buy these products.

3 Enhancement:

It's just a prototype and need to add a lot of functionality and pages to improve. Firstly, for the home page, it can be added some videos to show their company and show process of producing these food. Secondly, for products page, At present there are only a few products, it can be listed all the products for users to choose and add links that customers can directly click links to buy on the above. Besides, if the

OpenSSL module installed, the payment function can be achieved. What is more, the search function can be started by connecting to the database. Then customers can quickly find what they need with the search box on the right side of the navigation bar. Thirdly, for the comments page, though it connected to database, it need to be decorated and make the page more attractive. Finally, the login page can be connected with database to store the information of users.

4 Critical:

4.1 Advantages:

Using SQLite3 implemented a simple comment area. So that users can publish their opinions or comments on the products in this region. Besides, Using template inheritance to make each page have the same navigation part, so that the site is unified and coherent. What is more, using bootstrap makes pages not particularly ugly and the most basic functions of Python flask are realized.

4.2 Disadvantages:

Firstly, the pages are simple and not beautiful enough. Secondly, Although it connected to the database, it only implements the most basic functionality. Due to the lack of comprehensive knowledge, more functions have not yet been realized. Thirdly, because of the limitation of time, I do not have enough time to explore the issue about data safe, so there is no website security aspect involved.

5 Personal evaluations:

Through this course and the coursework, I learned how to use Python flask to build a website, although it is still rudimentary. Then, I have a better understanding of the HTTP protocol, API and some other the basic concepts via practice. Besides, because of the first homework, the basic function of this website is more handy, the operation is more skilful this time. I met a lot of challenges in the process of doing this coursework. Because of lack of knowledge about database, it is difficult to understand. I browsed great mass of resource and watched a lot of Related videos to solve some problems I met in the process. Besides, I sought help from my friend and solved it smoothly. For example, after the code was run, it always showed that the database could not be found, and later, with the help of friends, I found the problem and solved it. Last but not least, it is a fantastic and challenging experience for me. Though the app application which I made is not perfect, I tried my best to do it. Besides, through this course and these two coursework, I learned how to use Python flask to make a simple web application and have learned some basic knowledge. What is more, it stimulates my interest in learning python, Even though the course is over, my study of Python will not stop.

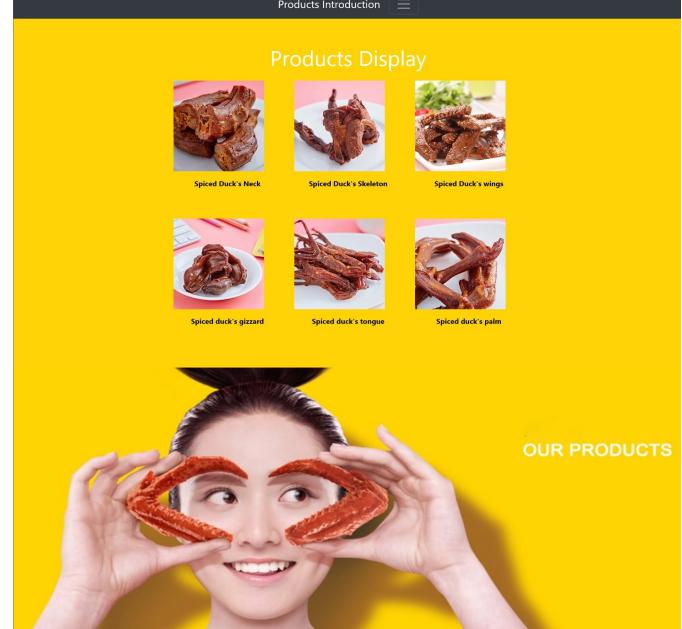
Appendix

Contents

1 Introduction	1
2 Design:	1
2.1 Layout Analysis:	1
2.2 Technical Analysis:	1
2.3 Style analysis:	2
3 Enhancement:	2
4 Critical:	2
4.1 Advantages:	2
4.2 Disadvantages:	2
5 Personal evaluations:	2

Screenshots

STRIVING
STRIVING FOR REJUVENATION OF CHINESE NATIONAL



SEARCH FOR GROWTH TIME MACHINE

SINCE THE EDGE
2002-2006

2002 -- the Zhou family set up the first "rich smell duck shop" in Wuhan.
2004 -- the Zhou family set up the second "black duck operation&department".
2005 -- apply for registered trademark.
2006 -- the establishment of Wuhan Black duck zhou proprietary company.

GROW UP
2007-2013

2007 -- the first processing factory was established in Wuhan.
2008 -- upgrade retail model to brand chain store with unique brand image.
2009 -- expanding to Hunan, launching vacuum packaging products.
2010 -- build online marketing channels to Guangdong.
2011 -- expanding to Shanghai and Jiangxi.
2012 -- expanding to Henan and Beijing, launching MAP products.
2013 -- was named "national agricultural products key leading enterprises".

DEVELOPMENTS
2014-2016

2014 -- expanding to Tianjin, Zhejiang and Jiangsu.
2015 -- expanding to Chongqing.
2016 -- expanding to Sichuan.

© 2017 Company, Inc. - BLACK DUCK ZHOU - [Back to top](#)

Figure 1: Home page - About Company

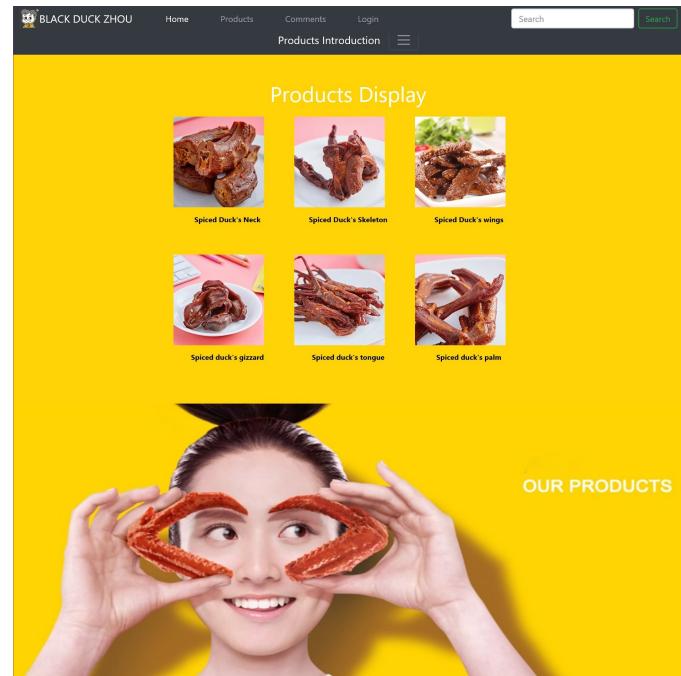


Figure 2: Products page - About Products

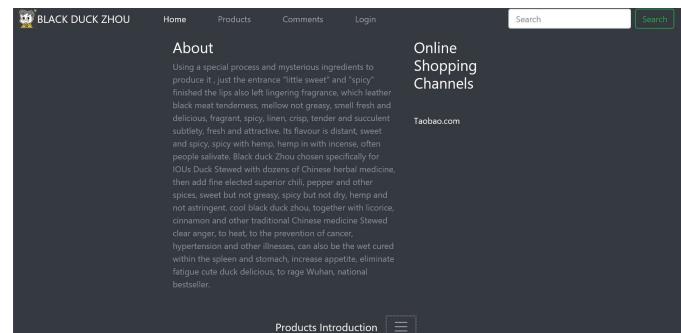


Figure 3: Products page - About Products

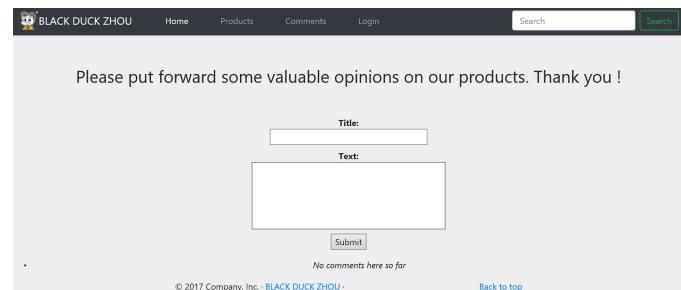


Figure 4: Comments page - Comment area

The screenshot shows a web page with a header containing links for Home, Products, Comments, and Login, along with a search bar. The main content area has a heading "Please put forward some valuable opinions on our products. Thank you !". Below this is a form with fields for "Title:" (containing "comment") and "Text:" (containing "Delicious!"). A "Submit" button is at the bottom of the form. A note "No comments here so far" is displayed above the form. At the bottom, there is copyright information and a "Back to top" link.

Figure 5: **Comments page** - Comment area

This screenshot is identical to Figure 5, showing the same comment form and the message "No comments here so far". However, it also includes a list of previous comments below the form, with one entry: ". comment Delicious!".

Figure 6: **Comments page** - Comment area

The screenshot shows a web page with a header containing links for Home, Products, Comments, and Login, along with a search bar. The main content area has a heading "Login". Below this is a form with fields for "User Name" and "Password", and a checkbox for "Remember me". A "Login in" button is at the bottom of the form. At the bottom, there is copyright information and a "Back to top" link.

Figure 7: **Login page**

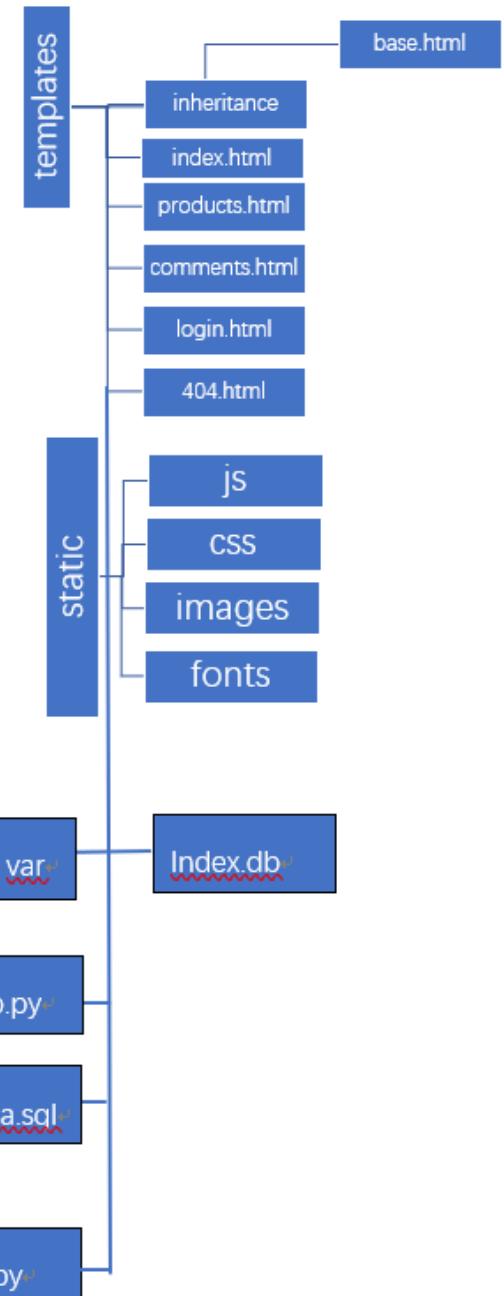


Figure 8: **File structure**