

LEGO

Yuchen Zhang
40332915@live.napier.ac.uk
Edinburgh Napier University - Module Title (SET09103)

1 Introduction

The theme of web app is LEGO. About the theme, it is a type of building toy, there are colorful plastic building blocks that can be joined together easily to make a tower and more. It is very popular, there are millions of LEGO fans around the world. About the web app, it is design for recommended products and collect the opinion of user. Therefore, It published the new products information and divided products into different categories to introduced. There are five main pages such as 'homepage', 'products', 'contact us', 'sign up' and 'login'. The layout and style are concise and match the theme. People could have a good experience when they visit the web page, and it is easy to find their favorite products by the clear categories. Moreover, they can login and send message to company, leave their suggestions or problems.

2 Design

Based on the purpose of the web app, there are five pages were designed. There is the main content of each page will be showed.

Firstly, the homepage would show the introduction of LEGO and recommended few new products. Secondly, the 'products' page is the main part to introduce products. The third page is 'contact us', it was designed for user send their opinion or suggestions. Finally, the 'sign up' and 'login' pages are design for create a personal user experience.

Next, in order to know more design of this web app, this following part of report would show it in two aspects including page layout and function.

2.1 Page layout

The reasons that different modules were designed into these five pages would be showed, respectively.

The shared part is navigation bar, it is divided into six parts, including web icons, home button, products button, contact-us button, search box, sign-up and login button. All the button can link to their own page. In addition, the background of the 'products', 'sign up' and 'login' page is the same picture, the content of the picture is LEGO toys. It will bring a well visual feeling and also match the theme of web app.

In the 'home' page, there is a carousel to show some pictures about the LEGO toys. Then the following part a sub-field and a picture to introduce the LEGO. There is also a link to official website, it would let user get more information.

In addition, there is a list of few new products with a link button.

In the 'products' page, there is a list divided into three parts. The three parts are the classification of LEGO products, they are 'Car', 'Building' and 'Aircraft'. In the list, there are names and pictures of each products and people can link into another page to view more details.

In the 'contact us' page, there are few rows at the left part of layout to show the contact information of company. Moreover, at the right part of layout, there is a 'contact' header and a group of form include 'user name' form, 'email-address' form and 'message' box. Then with a 'submit' button, user could click it.

In the 'sign up' and 'login' page, there is a similar group of form with a click button at the centre part of layout. At the 'sign up' page, it included the 'user name' form, 'email-address' form, 'password' form and 'confirm password' form. At the 'login' page, it included 'email-address' form and 'password' form.

2.2 Function

About the function that this web app achieved will be discussed in this part. Firstly, this web app uses the templates, static files and bootstrap to design a well style and layout. Moreover, in the web app, people can browse different pages smoothly and there will be some reply when user make a wrong code. Next, there are some other functions and their codes will be showed, respectively.

In the 'contact us' page, user can enter their name, email-address and content of message. Then click the submit button to send the message. However, If they enter in the wrong information and click the submit button, they will still stay in the contact page, and there will be a sentence shows: "Please fill in all fields and check them." above the form to remind people. To achieve these functions, 'Flask-WTF', 'Flask-Mail' and some codes of 'Message Flashing', 'Configs' were used to creative this page. In the Contact From, it also used 'DataRequired', 'Email' built in 'validators' , And a own 'validators' to check the length of name. The main codes will be showed later.

In addition, at the 'login' page, people can enter their information which is requested. If they enter in the correct information and click the 'login' button, they will go to the home page, however, if they enter the wrong information, they will still stay in the 'login' page, and there will be a sentence shows: "Invalid credentials. Try Again." under the button to remind people.

Finally, the next part would show the main codes about the

above functions:

Listing 1: lego.py in python

```
1 from flask import Flask, render_template, request, url_for, ←
  redirect, flash
2 from flask_mail import Mail, Message
3 from forms import ContactForm
4
5 app = Flask(__name__)
6 app.secret_key = 'YourSuperSecretKey'
7
8
9 @app.route('/')
10 def root():
11     return render_template('home.html')
12
13 @app.route('/products/')
14 def products():
15     return render_template('products.html')
16
17 # add mail server config
18 app.config['MAIL_SERVER'] = 'smtp.gmail.com'
19 app.config['MAIL_PORT'] = 534
20 app.config['MAIL_USE_SSL'] = True
21 app.config['MAIL_USERNAME'] = 'zzzara07@gmail.com'
22 app.config['MAIL_PASSWORD'] = 'zmly1996'
23
24 mail = Mail(app)
25
26 @app.route('/contact', methods=('GET', 'POST'))
27 def contact():
28     form = ContactForm()
29     if request.method == 'POST':
30         if form.validate() == False:
31             flash('Please fill in all fields and check them')
32             return redirect(url_for('contact'))
33         else:
34             msg = Message("Message from your visitor" + form.name.data,
35                           sender='zzzara07@gmail.com',
36                           recipients=[request.form['email'],
37                                       'someOther@mail.com'])
38             msg.body = """
39             From: %s <%s>,
40             %s
41             """ % (form.name.data, form.email.data, form.message.data)
42             mail.send(msg)
43             return "Successfully sent message!"
44
45     return render_template('contact.html', form=form)
46
47 @app.route('/signup/')
48 def signup():
49     return render_template('signup.html')
50
51
52 @app.route('/login/', methods=['GET', 'POST'])
53 def login():
54     error=None
55     if request.method == "POST":
56         attempted_emailaddress = request.form['emailaddress']
57         attempted_password = request.form['password']
58
59         if attempted_emailaddress == "111@111" and ←
60             attempted_password == "password":
61             return redirect(url_for('root'))
62         else:
63             error = "Invalid credentials. Try Again."
64
65     return render_template('login.html', error=error)
66
67 if __name__ == "__main__":
68     app.run(host='0.0.0.0', debug=True)
69
70
71
72
73 }
```

Listing 2: forms.py in python

```
1 from flask_wtf import Form
2 from wtforms import StringField, TextAreaField, SubmitField, ←
  validators
3
4 def CheckNameLength(form, field):
5     if len(field.data) < 1:
6         raise ValidationError('Name must have more than 1 ←
  characters')
7
8 class ContactForm(Form):
9     name = StringField('Your Name:', [validators.DataRequired(), ←
  CheckNameLength])
10    email = StringField('E-mail address:', [validators.←
  DataRequired(), validators.Email('your@email.com')])
11    message = TextAreaField('Your message:', [validators.←
  DataRequired()])
12    submit = SubmitField('Submit')
13 }
```

3 Enhancements

The web app has a basic design and few functions at the moment. Compare to the original design, there are some features would be improved.

About contents and style. Firstly, the web app would add more sub-pages to make the collection be more complete, such as 'shop' page, personal homepage and some other sub-pages about each product. And users could get more detail information about LEGO. At the same time, it needs to add style that to make pages look nice.

About functions, the web app need to achieve the user registration system and complete the user login system, including connect the database and ensure the data is stored safety.

4 Critical Evaluation

About the features which I feel work well are the functions of 'contact' page and the 'login' page. In contrast, there are some features I feel work poorly. First, it is the content of the web app. I feel it needs to add more subpages. And the most regrettable is the full function of registration and login system did not achieve. The reason of that is I met some problems such as achieve the dynamic data entry and other steps, moreover, though I have research a lot and see some tutorial videos, I still do not have the enough ability to complete my codes about this function. I need keep learning this knowledge and try more times.

5 Personal Evaluation:

According to this coursework, I have learned some new knowledge, such as Flask-Mail and Flaks-WTForm. I feel I understand how to use python and Flask more clearly than before. And I understand more knowledge about using SQLite3 with Flask even though I did not achieve the function in this web app, I will keep learning it. In addition, I faced some challenges when I design the web app, such as the function of

'contact' page did not work, and the layout of group form did not become what I wished. I handled these challenges according to read and try to understand the workbook or other books, watch some videos to improve my knowledges, talk with my classmates or module leaders, and then try to correct codes. About my performed, I think I need more practice to improve my skills and make a reasonable arrangement of time to improve my self-directed study.

A Appendices

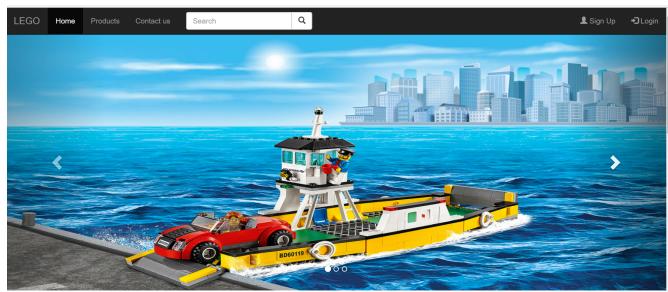


Figure 1: homepage -m

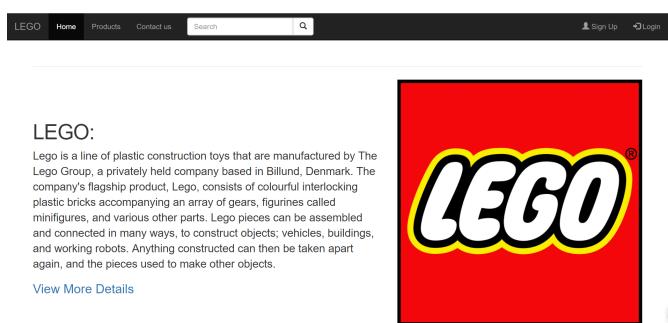


Figure 2: homepage1 -m

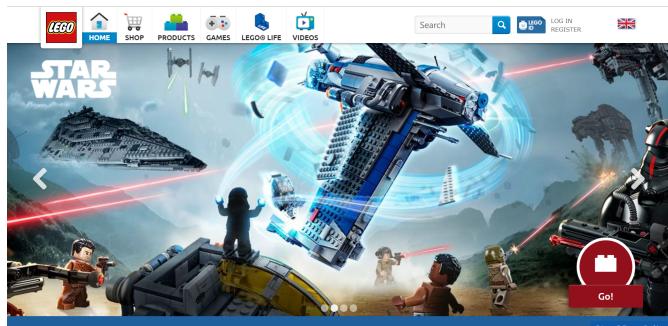


Figure 3: homepage-linkpage -m

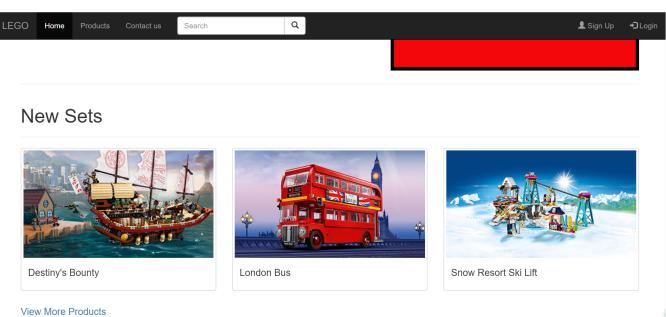


Figure 4: homepage2 -m

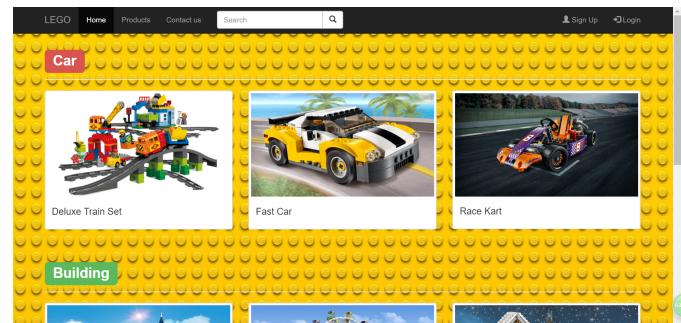


Figure 5: products -m

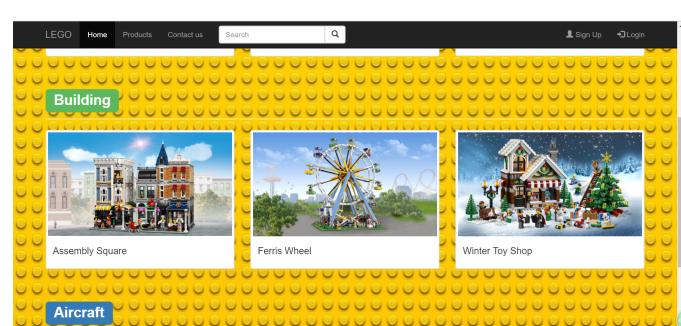


Figure 6: products1 -m

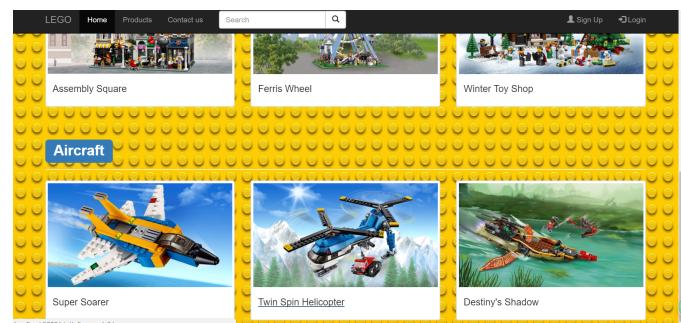


Figure 7: products2 -m

localhost:5000/contact

Contact us and we'll get back to you within 24 hours.

Edinburgh, UK
+00 15151515
myemail@something.com

CONTACT :

Your Name:

E-mail address:

Your message:

Submit

Figure 8: **contact -m**

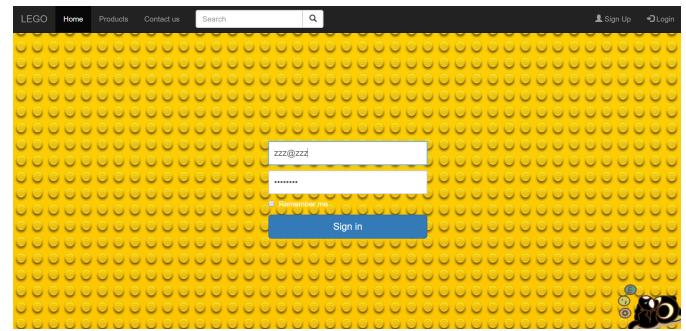


Figure 12: **login -m**

CONTACT :

Your Name:

E-mail address:

Your message:

Submit

Figure 9: **contact-test1 -m**

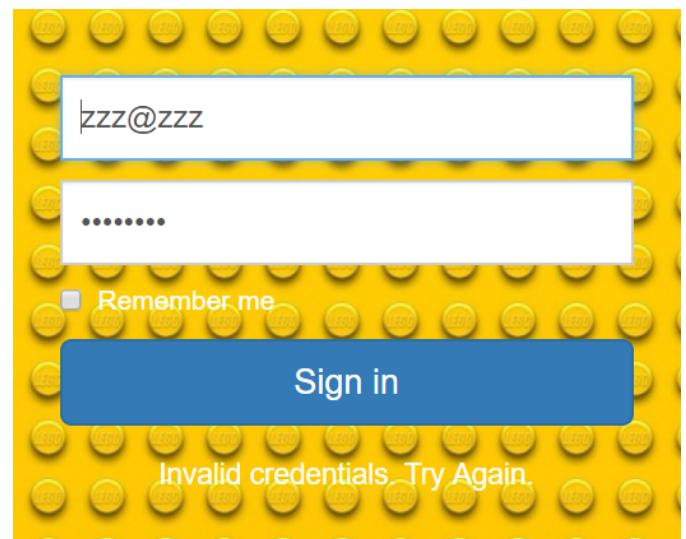


Figure 13: **login1 -m**

CONTACT :

- Please fill in all fields and check them

Your Name:

E-mail address:

Your message:

Submit

Figure 10: **contact-test2 -m**

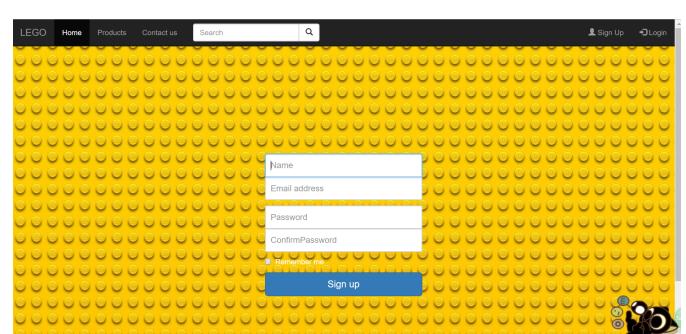


Figure 14: **signup -m**

CONTACT :

- Please fill in all fields and check them

Your Name:

E-mail address:

this is a message.

Your message:

Submit

Figure 11: **contact-test3 -m**