Coursework Report

Matthew Lynch
40322639@live.napier.ac.uk
Edinburgh Napier University - Advanced Web Technologies (SET09103)

Abstract

This project is introducing Matthew to a set of new skills including SQLite3, Flask, Python and Jinja2. The idea it to produce a website with database functions. He talks about how he planned, researched and thought about how colour and layout is important. The report lists the enhances that could be implemented and evaluate the project.

Keywords – Python, Flask, SQLite, HTML, JavaScript, CSS, Jinja2, Travel, Web-app

1 Introduction

The website that Matthew has produced is a web-app that allows users to create user account, allowing them to start adding wishes to their bucket list. This includes an admin, add and remove functions. This also includes standalone HTML pages such like the home and about pages.

2 Design

The web application was architect by using a series of stages including planning, research and implementation of the Flask features to develop to overall outcome.

2.1 Planning

Matthew started coursework 2 like coursework 1 in that he laid out all the directories as a visual concept. This helped him to plan were the files would go and the routes of the directories. This can be seen in Figure 1.

He then started planning the features he wanted to include in this web app. He wanted to include a login with sessions, register, database query and bucket list form.

2.2 Research

Matthew started by doing some research about what he wanted to base his web app on. He desired something simple and straightforward and wanted to base it off something that interested him. At this point, he descried to make a bucket list website. Users register and create an account, and this would allow account holders to make their own bucket list. He was inspired in doing this website by the ideas from looking at bucket list “blogs“[1] and “articles“[2] online.

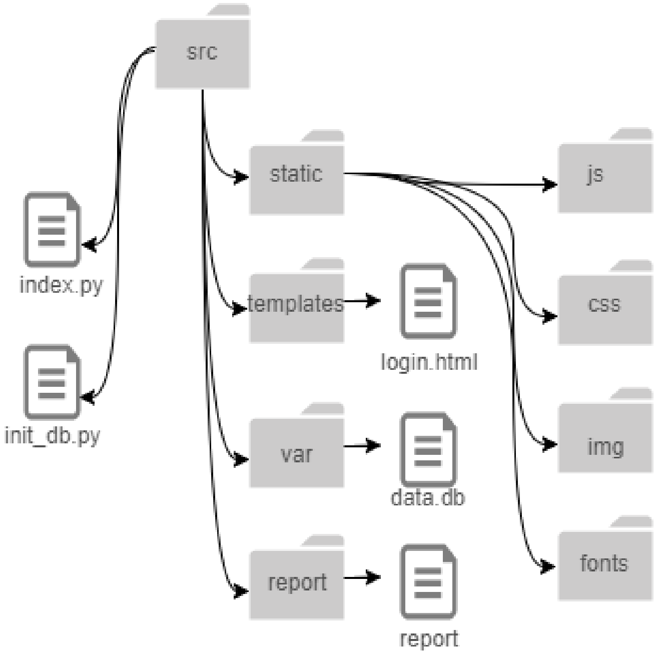

Figure 1: **Directory Hierarchy** Layout of Hierarchy

2.3 Colour Scheme and Layout

For the colour scheme Matthew desired to implement a simplistic white and blue scheme. This was chosen because he wanted it to represent traveling and seeking adventure. The font used is a simple but effective hand drawn web font. The colour scheme and layout are kept minimalistic to allow for legibility and usability. The layout is a full screen layout with a professional image taken from “Pixabay“[3]. This conveys an over look of professionalism and trustworthiness of the website.

2.4 Python Flask

The development of the website used python flask which is a microframework that is based on Jinja 2. It was beneficial for Matthew to use Flask as it allows for websites to be set up quickly with only a few lines of codes, as seen below. Flask allows for extensions however what Matthew wanted to implement did not require any extensions.

```
from flask import Flask, url_for, render_template, json, redirect, ↩
      request
app = Flask(__name__)

#The Homepage Route
@app.route("/")
def index():
  return render_template('index.html', title='Home')
```

Matthew decided that he would implement a 404 page, in case of any unexpected errors that users may find. This custom 404 page allows the user to re-direct back to the home page and would keep the page stylized for a consistent design across the website.

2.5 SQLite3

SQLite3 was used primarily as the websites functionally. Matthew wanted to implement a database within the website. The idea was to have two tables one for "user" and the other called "bucketlist". The users table would store the users name and password that the users imputed into the form. The users table would also fetch the data for when the user's login. The login and register function in the website is currently not working properly. This is due to the time constrains and the lack of knowledge in the sqlite3 language.

To login to the website the admin would only have access to the dashboard page, as seen below, which consists of the bucket list feature. This future works by the user adding details into the form and hitting submit. The details of their bucket list are then added in the dashboard and listed below. There are commented out attempts in the file index.py of the user login function.

```
if request.method == "POST":
        if request.form["username"] != "admin" or request.form["password"] != "admin":
            error = "Invalid Credentials. Please try again."
        else:
        session["username"] = True
        flash("Welcome! <username>")
        return redirect(url_for("dashboard"))
    return render_template("login.html", error=error, title="Login")
```

3 Enhancements

An enhancement that could be implemented is a working function of a user login. There is currently code commented out to prevent the index.py from running the debugging tool. The inclusion of a user login would then allow for users to have their own bucket list.

Another enhancement that could be included into the website is a registration feature. This would partner up with the login feature allowing users to sign up to the website. Enhancing the login with a forgot password function is also a feature that Matthew would incorporate with the login page. This would allow users to update their old forgotten password letting users use their account.

Social media is another feature Matthew would add to the website, allowing users to like our social media pages. Matthew would include a share button allowing users to share their bucket list with their friends on social media. Currently the website has a dashboard page, but the inclusion of a personal page would be great for users to update details such as their profile picture or the about me section.

4 Critical Evaluation

The features that went well were the use of flask features that were used in the development of the website. Matthew used jinja2 templating on the HTML pages which required less coding and reduced the receptiveness. Jinja2 allowed Matthew to create a base temple which includes the navigation bar, the title page, the footer and code to allow content to display as seen below.

```
{% extends "base.html" %}

{% block content %}

{% endblock content %}
```

The hierarchy allowed for users to access all the relevant pages. The hierarchy of the website is simple and mostly used with the login, sign in and dashboard pages.

The overall design and layout of the website work well as it allowed the user to view the content and use the features without getting overwhelmed or distracted by a bold colour scheme. This is also true with the websites layout as Matthew kept it free of clutter and used white-space throughout the website. This kept the website consistent and creates a brand and identity.

Finally, a feature that went successfully was the inclusion of a login session and admin. The session was added to prevent users from accessing the dashboard page from the URL. This means that only users that are logged can access the dashboard page. Due to the user login not working, admin was hard coded and allowed for the session to be created for admin.

Some features that Matthew felt didn't go well was the setup of the database. He had problems trying to set up the database with SQLite3. This meant when trying to query the database from the python script, several errors displayed with the debugging tool. This was also true for submitting data from the form. When Matthew added information into the form he was consistently receiving errors related to the insert user data query. Try to resolve these issues ending up using a lot of Matthews time. This resulted in Matthew restarting the setup of the database and trying to implement it into the website again. This again caused other issues making the user login not to add user details into the database.

Matthew also had issues with getting SQLite3 queries to link properly the corresponding rows in the database. There were also issues involving the authorisation of the user details, this resulted in many errors, causing the Matthew to spend a lot of his time trying to fix the issues with login. From this point Matthew decided to try and get a admin account working, which proved successful. This was done my hard coding it within the python app.

Matthew also tried to include hashing and salting to passwords in the database. This was done to try and protect user's passwords in-case of any of them gets leaked, exposing the users. Hashing was attempted in the python script but again with the issues related to the database it was not fully implement successfully. Lastly another issue Matthew faced was implementing the bucket list in the dashboard page. This was due to the database not recognising the corresponding rows in the python script. This again was left in the script partly finished. This was due to the limited time and knowledge that Matthew had to fix all the issues he faces with the login and database implementation.

5 Personal Evaluation

Overall, Matthew felt that the website could have been a lot more successfull, due to the number of issues he had implementing features using SQLite3. He feels that even thought the website has a lot of problems, these problems allowed him to learn even more about SQLite3 and databases, compared to if everything had worked without many issues. This has given him a better understand of databases and how they would integrate into a website. Next time he would try to plan his database and feature better rather that designing it during the development of the website.

Even though Matthew did not successfully manage to get user and the database to work he attempted to overcome this by using the coursework workbook that and online tutorial. He used online videos [4] to try and fix the login and sign up functions. On trying to fix the many errors that Matthew came across he tried to use Stack Overflow [5] to fix the issues. Matthew followed an online tutorial [6] to try and help him try to implement hashing to his passwords.

A method of overcoming the user login was to include a hard coding admin, this was so he could have some way to access the dashboard page a working on the bucket list function. Matthew did this looking at an online form [7].

In conclusion looking back and reflecting on the project Matthew would make sure he plans and tests his database first before continuing building the website. In this project Matthew setup, a static website first and then added forms before considering the tables and rows of the database. Looking back, he would do test runs of the database and then design the website around it which would have been an easier approach. Matthew also felt that a lot of time was spent focusing on the login feature and ended running out of time to improve the overall quality of the website. This was a good learning experience as it allows him to gain knowledge and a better understanding of how the SQLite works and practice for future projects.

References

[1] Kayla, “kaylablogs,” Aug. 2015.

[2] C. Morgan, “The bucket list: Have you made yours?,” May 2017.

[3] H. Braxmeier and S. Steinberger, “pixabay,” Nov. 2010.

[4] calicoJake, “Python/flask tutorial part 3 - login,” Feb. 2012.

[5] gallly, “flask-login not sure how to make it work using sqlite3,” May 2013.

[6] pythonprogramming.net, “Password hashing with flask tutorial,” Aug. 2013.

[7] ihoegen, “Flask-login-app-tutorial,” Dec. 2015.