

The Countries I Travelled

Silvia Balogova

40175139@napier.ac.uk

Edinburgh Napier University - Advanced Web Technologies (SET09103)

Keywords – Python, Flask, Web-app, API, Jinja2, login, SQLite3, travel, countries

1 Introduction

The Countries I Travelled is a web-app for sharing blog posts with other users who enjoy travelling. The users can browse the blog posts of other users which are available publicly. The web app provides registration for new users who would like to create their own blog posts.

The application stores the users personal information securely in its SQLite3 database.

The website was programmed with Python language and Flask micro-framework.

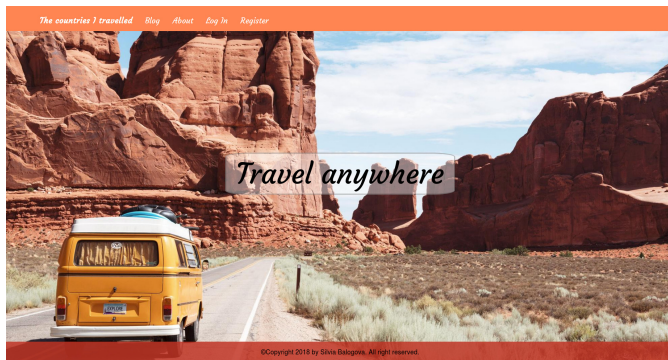


Figure 1: **The Countries I travelled** - The Screenshot of the main page

2 Design

The web-app was created with Python programming language, the Flask micro-framework, Jinja2, HTML and CSS. Many modules for Python and Flask were also installed and implemented.

First the nature and purpose of the web app were considered and then the URL Hierarchy was designed.

2.1 Python

The web app was programmed with Python3 programming language. Python3 and its extensive libraries were used to program the core of the web app.

2.2 Flask

Flask micro-framework was used to make web development faster and easier.

Flask was used to create routing for each Python function to access desired URL destination.

There were many Flask modules imported to help with the app functionality.

2.2.1 Routing

Flask provided tools for easy dynamically created routing. To minimise the repetition of code the hierarchy of page templates was build.

2.2.2 Flask SQLAlchemy

SQLite3 database was used to store the user username, email, and password. Flask's SQLAlchemy enabled the process of storing and retrieving data.

2.2.3 Flask Migrate

Flask-Migrate is an extension that handles SQLAlchemy database migrations for Flask applications using Alembic. The database operations are provided as command line arguments for Flask-Script.[1]

2.2.4 Flask Login

Flask-Login is a useful User session management tool for Flask applications. It handles the common tasks such as logging in, logging out, and remembering the users sessions over the time of their account existence.

Flask-Login was used with SQLite3 database system and Flask Migrate extension. The methods for user objects were implemented, and a callback to the extension capable of loading users from their ID was provided.

2.2.5 WTForms

WTForms library is a flexible forms validation and rendering for Python web development. The library was chosen to validate string, email, and password fields to confirm its validity. The library was also used to match the desired criteria such as fill all the necessary fields in the forms, and compare the equality of the passwords.

2.3 Blueprint

Flask Blueprint was used to organise and structure the collections of views, templates, static files, and init.py files. It expanded project into many files and modules to make it easier to see, read and maintain for future development.

The project was divided into further sections:

Project

= blog posts

= = init.py

- = = forms.py
- = = views.py
- = core
- = = init.py
- = = views.py
- = error pages
- = = handlers.py
- = static
- = = css files
- = = images
- = = profile pictures
- = templates
- = = html error pages
- = = other html templates
- = users
- = = init.py
- = = forms.py
- = = picture handler.py
- = = views.py
- = other python scripts
- = data.sqlite
- = app.py

2.4 Jinja2

The Jinja2, templating engine for Python, was used to take advantage of template inheritance. The web application includes a single base template that defines the structure for the rest of the pages. The majority of HTML code was added to the base template to avoid repetition. The subsequent child templates inherit the code of base template and add the unique content needed only for the specific page. The implementation allows consistent layout throughout the website and easy maintenance in the future.

2.5 Static files

The simple CSS file was created for *The countries I travelled* project. The CSS frameworks such as Bootstrap, and Bulma were considered but were not fit for this project. They often over complicate the process and get in the way of functionality and design.

2.5.1 Pillow

Pillow is the PIL (Python Imaging Library) for Python3 and above. The Python Imaging Library adds image processing capabilities to the Python interpreter. The PIL library was used to handle user profile pictures.

2.6 Error-handling

The web-app has a separate error message for the most common errors (404 and 403). There are many error codes the web-app might need to respond to but as the web is not publicly available they were not considered necessary. The users can benefit from error page informing about the type of the error and can further navigate to the webapp through the usual navbar.

3 Enhancements

3.1 Content

The *Countries I travelled* web application currently stores really little data. Only a few users were created for the testing purposes. There are currently two testing blog posts available upon opening the website.

4 Critical Evaluation

The main functionality of the *The Countries I travelled* was achieved. It is a working web app where users can view the blog posts of other authors, and after registering they can also create their own content. All main aspects of the web app are functional and are working well for small amount of users.

5 Personal Evaluation

The *Countries I travelled* project for the Coursework was the excellent opportunity to learn new skills and techniques used in modern web technologies.

This project was challenging and the learning curve was steep. The direct study of unknown Python, Flask and database elements and features took hours of time. There were many online materials read, and studied, mainly official documentation for Python3 and all the modules, libraries and extensions that were used. Time and resources was also dedicated to various tutorials at Udemy.com and YouTube.com.

Answers to many projects challenges and simple errors were discussed at forums such as Stack Overflow and FreeCodeCamp.

The background study of the security behind registration process, hashing passwords and storing users data should be also taken into account.

Overall, creating the *The Countries I Travelled* was an excellent chance to expand the knowledge about web applications, front end and back end development, data handling, user registration and so on.

Furthermore, this project is going to be a base for my personal web app in the future which is going to be released in upcoming months.

References

- [1] Flask-Migrate, "Flask-migrate," Nov. 2018.