

South East European University

Faculty of Contemporary Sciences and Technologies

Final Project

Flask Blog

Student: Mentor:

Alban Memishi Prof.Dr.Nuhi Besimi

Tetovo, June 2021

# Contents

[Contents 2](#_Toc74697463)

[List of figures 2](#_Toc74697464)

[1 Introduction 3](#_Toc74697465)

[2 USED TECHNOLOGIES FOR THIS DEVELOPMENT 4](#_Toc74697466)

[2.1 PyCharm 5](#_Toc74697467)

[3 How to use the app 5](#_Toc74697468)

[3.1 Project Structure 7](#_Toc74697469)

[4 Implementation of Authentication and CRUD 8](#_Toc74697470)

[4.1 CRUD 8](#_Toc74697471)

[5 Problem description 10](#_Toc74697472)

[6 Bibliography 11](#_Toc74697473)

# List of figures

[Figure 1: PyCharm 3](#_Toc74697516)

[Figure 2: Home Screen 4](#_Toc74697517)

[Figure 3 Account info 6](#_Toc74697518)

[Figure 4 New Post 7](#_Toc74697519)

[Figure 5 Example post added 7](#_Toc74697520)

[Figure 6: Login and Logout 8](#_Toc74697521)

[Figure 7 Create new post 9](#_Toc74697522)

[Figure 8: Edit and Delete methods 9](#_Toc74697523)

# Introduction

In this project I wanted to build something simple and that is a Blog page that will include all the necessary elements for a blog post.



Figure 1: PyCharm

This application first starts with the *home* page in which the users can see all the posts and also see by who it was posted.

The blog app shows all the posts from a different users and it also shows the date it was posted, in order for others to see the post they can click on the post and the post will be displayed one by one.

The app has the navigation links, it has the authentication part and also when the user is already logged in they can view their information and if they have a desire to change them they can do it freely. The application has all the CRUD operation (create, read, update, delete).

In this project I’ve used the PyCharm as my IDE for my development journey, the main language that I used her was Python and the framework was Flask.

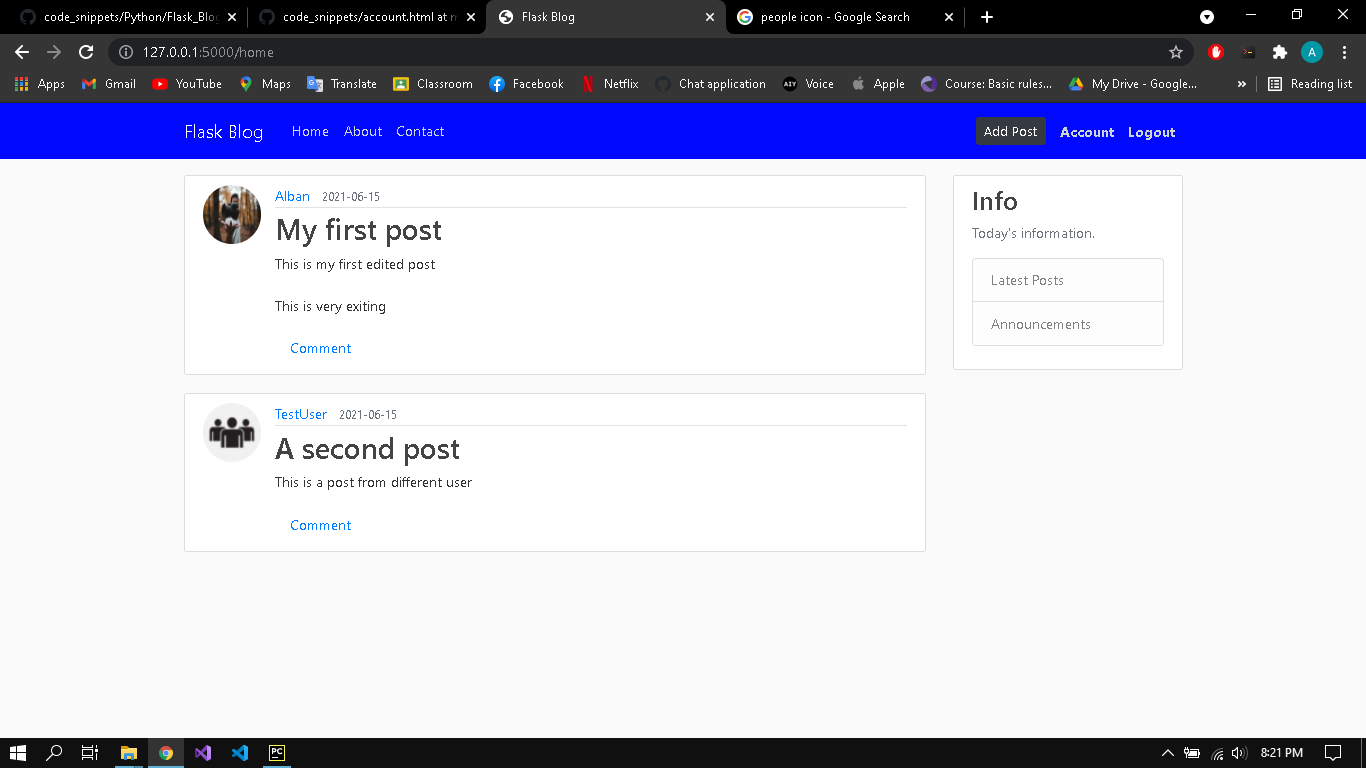


Figure 2: Home Screen

# USED TECHNOLOGIES FOR THIS DEVELOPMENT

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by metaprogramming and metaobjects (magic methods))Many other paradigms are supported via extensions, including design by contract and logic programming.

Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

Python's design offers some support for functional programming in the Lisp tradition. It has filter,mapandreduce functions; list comprehensions, dictionaries, sets, and generator expressions. The standard library has two modules (itertools and functools) that implement functional tools borrowed from Haskell and Standard ML.

The language's core philosophy is summarized in the document The Zen of Python (PEP 20), which includes aphorisms such as

Beautiful is better than ugly.

Explicit is better than implicit.

Simple is better than complex.

Complex is better than complicated.

Readability counts.

Rather than having all of its functionality built into its core, Python was designed to be highly extensible (with modules). This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications. Van Rossum's vision of a small core language with a large standard library and easily extensible interpreter stemmed from his frustrations with ABC, which espoused the opposite approach. [1]

## PyCharm

**PyCharm** is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains (formerly known as IntelliJ). It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as data science with Anaconda.

PyCharm is cross-platform, with Windows, macOS and Linux versions. The Community Edition is released under the Apache License, and there is also Professional Edition with extra features – released under a proprietary license. [2]

# How to use the app

As I said in the beginning, the first default view that the users will encounter is the home page. In the home page the users will see all the posts that other users have posted.

In the top there is the navigation bar where the users can navigate thru the app, in the left side there are the different links for Home, About and Contact, and in the right side there is the authentication part where the user can register or login.

The authentication part works in a simple way and that is as such, for instance the new users can register and create new account and with that account they can use it to log in and create post(s), what the user first sees is the Login and Register part. After the user is logged in the app, the option to add a new post has appeared next to authentication part together with the Account part and Logout part.

**Account part** this section allows the users to view their information and to edit them. This part has all the account information that the user provided us in the beginning. This looks like this

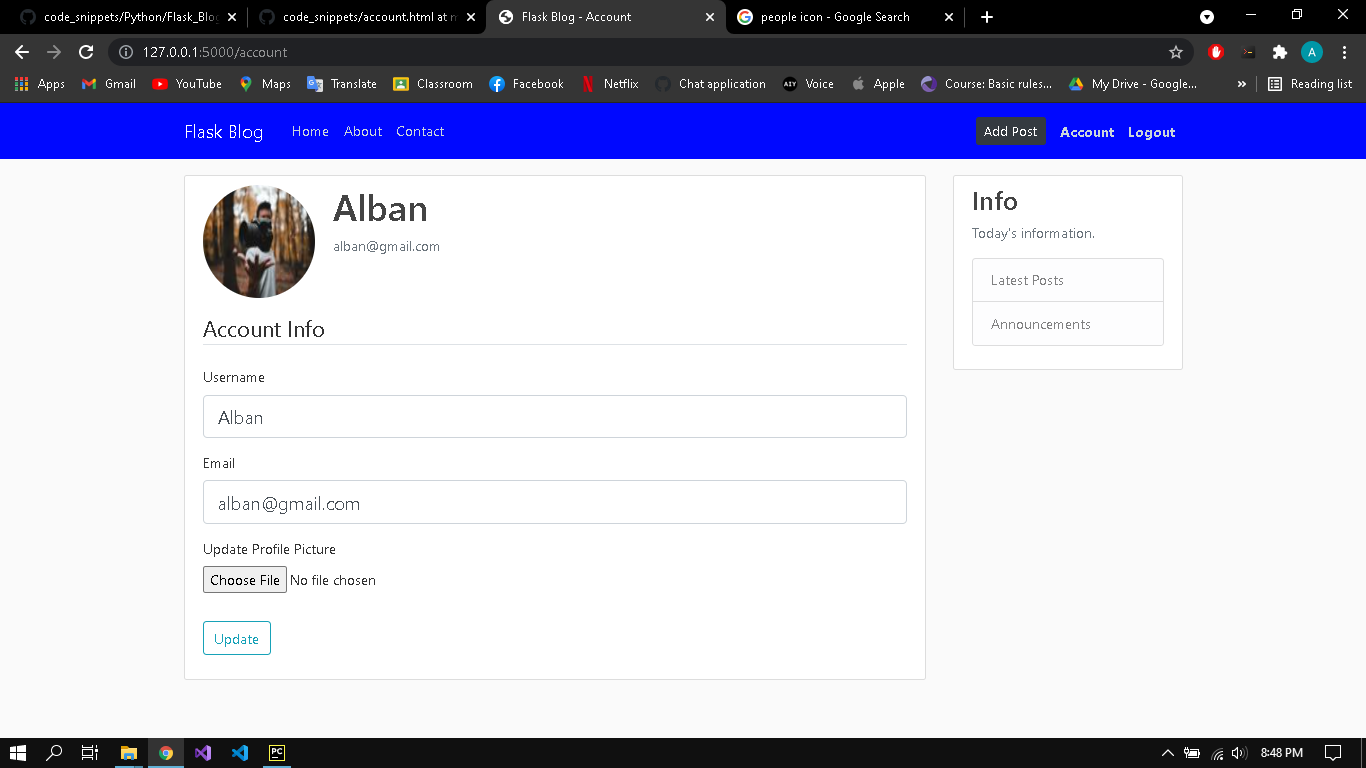


Figure 3 Account info

### As we can see clearly in the app after the user is logged in there is a button to add new post. (*Add Post*). To add new post is easy all you have to do is click on it and you will be displayed a form to add new post, you fill all the information needed and your post is done.

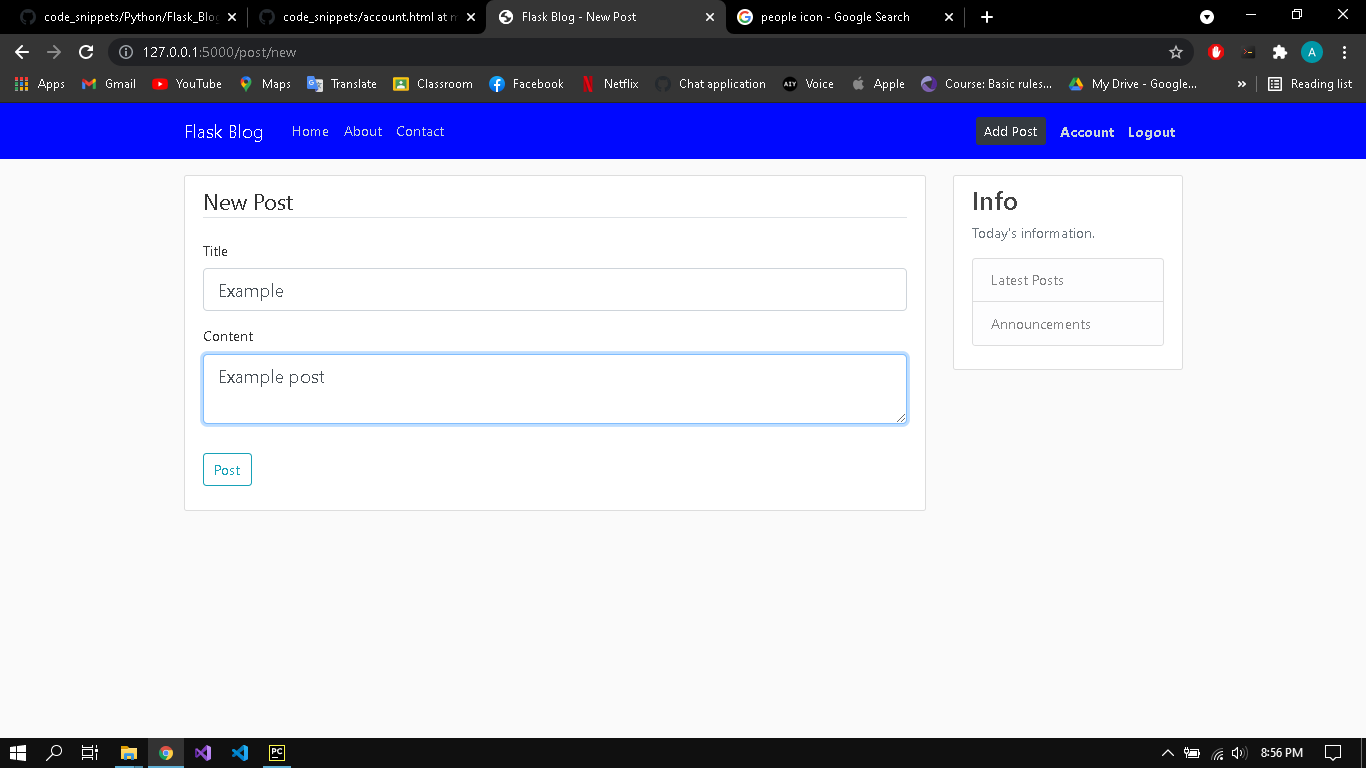


Figure 4 New Post

After the post is done you are redirected to the home page, in order for you to see the result. Take a look at the picture to see the effect of it.

Graphical user interface, text, application

Description automatically generated

Figure 5 Example post added

### Edit and delete your post: The users can view the content of the post by clicking on it. What is interesting here is that when the current user is logged in he can Update and Delete his posts, which means that this restricts them from editing others content and deleting.

Graphical user interface, text, application, email

Description automatically generated

## Project Structure

* **flaskblog –** is the folder where is stored the source code of each activity
* **static –** here we store all the css files and in *static* folder we have **profile\_pics** where we save the photos of every user profile each time they change it.
* **templates -** here we have all the .html files for each action.
* **\_\_init\_\_.py** – we save all the configuration
* **forms.py** – here we have all the methods for saving the forms as it should
* **routes**.**py** – in the routes.py file we have all functions that will handle all the user requests, like login, logout, CRUD.

# Implementation of Authentication and CRUD

In this part I will briefly explain the implementation of the most important concepts



Figure 6: Login and Logout

I’ve defined two functions here login and logout. How does each one of them works, I will explain briefly. In the login function the code starts by checking if the user is authenticated and if it is then it will redirect to the home page. If the user isn’t authenticated, which means if it does not have an account he creates one and after creating one the password is not saved as plaintext but as a hashed value. If the login was unsuccessfully than there will be a flash message saying “Please check email and password”

The logout is simple we only call the logout function and we redirect him to home page

## CRUD

The implementation of all the CRUD operations will be explained by this images:

Create: The Create method creates an instance form PostForm() (which inherits from FlaskForm) and check that if the form is on the validation process, creates and the object of the Post by getting the title, content and author, it adds to database and commits the changes.

Text

Description automatically generated

Figure 7 Create new post

Update (Edit): The post method checks if the same user is first authenticated, if not than he will have an abort. After validation the data will be updated successfully to the database else if our method is GET then we will display the data to the user.

Delete: The delete is almost the same as edit, the method accepts the id as the parameter and with that id we know exactly which post is going to be deleted. Than the changes are saved to database and the user is redirected to the home page.

Text

Description automatically generated

Figure 8: Edit and Delete methods

# Problem description

The idea was to create a fully functional blog posts, that except this fundamental things we have implemented I wanted to add more features to this project and one of the main features that I wanted to add was to add the comment section where different users can comment on the post and reply to each other, but because things went so fast during the lectures I only implemented this simple blog post with some minor features, hoping that in the future I will expand this project and make use of it

# Bibliography

|  |  |
| --- | --- |
| [1] | Wikipedia, https://en.wikipedia.org/wiki/Python\_(programming\_language). |
| [2] | Wikipedia, https://en.wikipedia.org/wiki/PyCharm. |