

# SAFARI

Deployment link (Hosted in-pythonanywhere )

<https://melbinmartin.pythonanywhere.com/account/?next=/>

## How to Run Locally?

Step 1: Clone the github repository <https://github.com/melbinmartin/safari.git>

Step2: Install the packages from requirements.txt

Step3: Open the terminal from folder name Safari

Step4: Run the command python manage.py runserver

Step5: It will direct to browser with port number <http://127.0.0.1:8000/>

## Walk Through

- Sign Up If You are a new user
- After Signup - Login

The image displays three screenshots of the Safari web application interface, arranged in a 2x2 grid with the bottom-right cell empty. Each screenshot features a white form area on the left and an orange background area on the right.

- Top Left Screenshot:** The form area is titled "Log in" and contains fields for "username" and "Password". Below the password field is a link "Forgot your password?". The orange area on the right says "Hello, Friend!" and "Enter your personal details and start journey with us" with a "SIGN UP" button.
- Top Right Screenshot:** The form area is titled "Create Account" and contains fields for "Username", "Email", "First Name", "Last Name", "Password", and "Confirm Password". The orange area on the right displays "invalid credentials" and lists password requirements: "Your password can't be too similar to your other personal information.", "Your password must contain at least 8 characters.", "Your password can't be a commonly used password.", and "Your password can't be entirely numeric." It includes a "LOGIN" button.
- Bottom Left Screenshot:** The form area is titled "Create Account" and contains the same fields as the top right screenshot. The orange area on the right says "Welcome Back!" and "To keep connected with us please login with your personal info" with a "LOGIN" button.

## To check the Login


### Sample Credentials

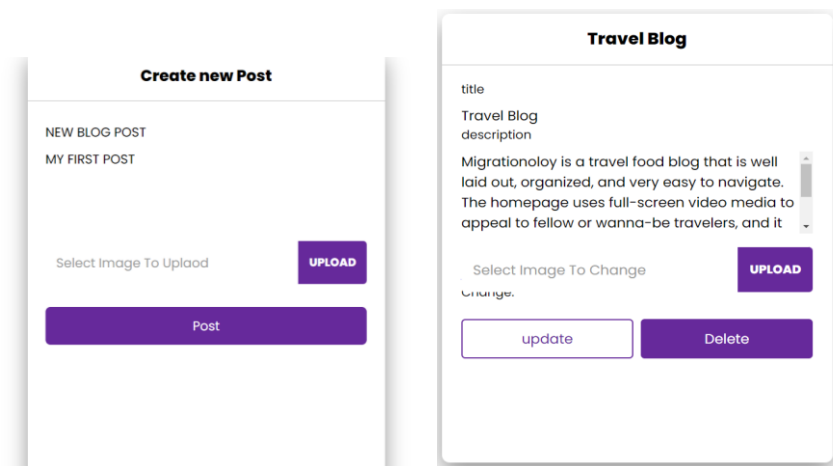
Username: alen

Password: Sherlock221b@

Username: nabeel4477

Password: Mnp44#blogs

- Once done it will take to profile update page fill the details if needed
- Click on Home
-  button in the corner You can add new Blog.
- Option to edit can be seen once the blog is posted.
- You can edit/update/delete the post.



**Create new Post**

NEW BLOG POST

MY FIRST POST

Select Image To Upload **UPLOAD**

**Post**

**Travel Blog**

title

Travel Blog

description

Migrationoloy is a travel food blog that is well laid out, organized, and very easy to navigate. The homepage uses full-screen video media to appeal to fellow or wanna-be travelers, and it

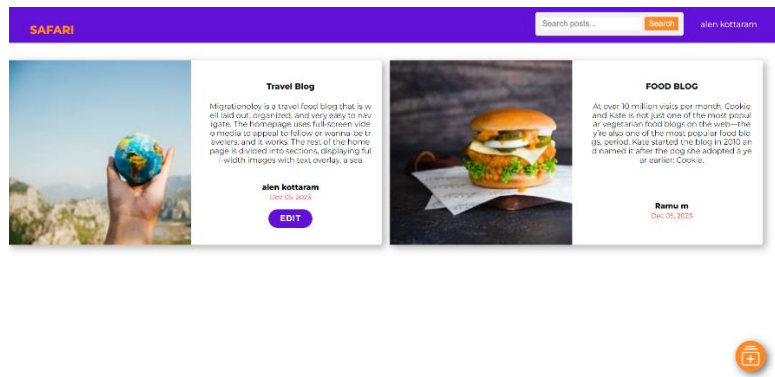
Select Image To Change **UPLOAD**

Change.

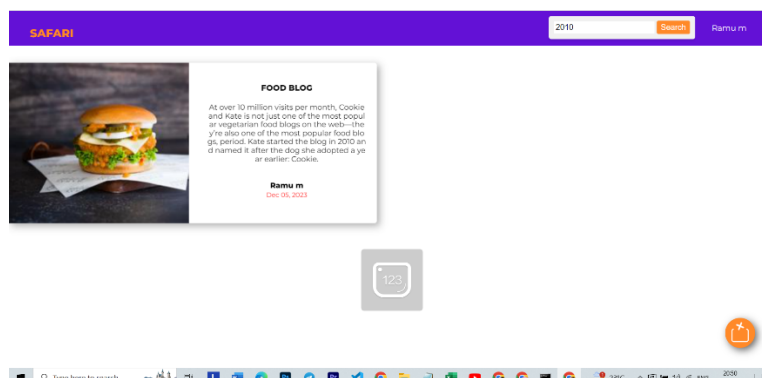
**update** **Delete**

Display :

Display is based on Posted first based on date



- Search:
- Search is based on Title or Description



## Registration

Allow users to register on the platform using their email and password

Implement a robust authentication mechanism for secure user login

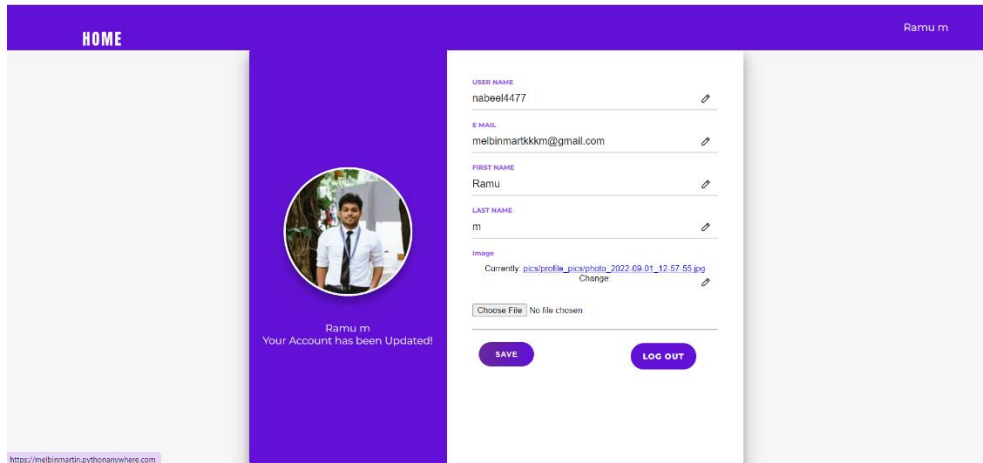
## Authentication

Sign in verification

Password cannot be related to Personal Details

Email id should be unique no same email can be used for logi

Search Functionality: Users can utilize the search bar to enter keywords or tags related to their interests. search functionality filters according to description and title of the blog post.



## TECHNOLOGIES AND FRAMEWORK

- HTML,CSS
- Javascript
- Python
- DJANGO
- SQLLite
- Cloud – PythonAnywere

## SECURITY

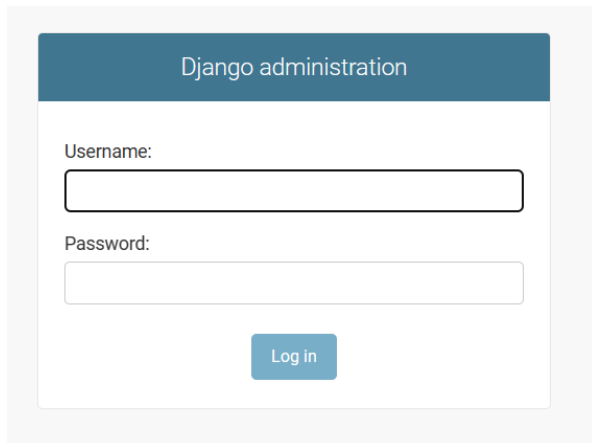
- Password validation
- Session manager
- Cross-Site Request Forgery

## ADDITIONAL FEATURES

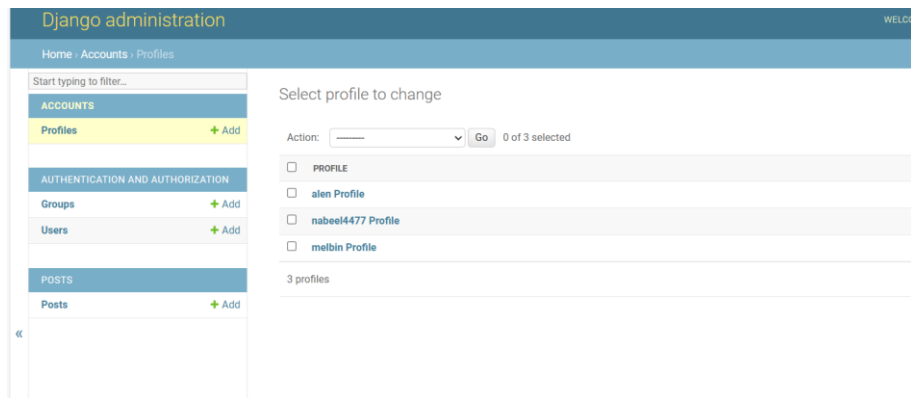
Enabling users to edit and update their profiles is a valuable feature that enhances user engagement and personalization on our blogging platform. This feature empowers users to manage and showcase their information, contributing to a more dynamic and personalized user experience.

## ADMIN MODULE

Django provide a default admin module



The image shows the Django administration login interface. It features a dark blue header with the text "Django administration". Below the header, there are two input fields: "Username:" and "Password:". A blue "Log in" button is positioned below the password field. The entire form is set against a light gray background.



The image shows the Django administration dashboard. The header is dark blue with "Django administration" on the left and "WELCOME" on the right. Below the header, there is a breadcrumb trail: "Home > Accounts > Profiles". On the left side, there is a sidebar with a search bar and a list of menu items: "ACCOUNTS" (with a sub-item "Profiles" and a "+ Add" link), "AUTHENTICATION AND AUTHORIZATION" (with sub-items "Groups" and "Users", each with a "+ Add" link), and "POSTS" (with a sub-item "Posts" and a "+ Add" link). The main content area is titled "Select profile to change" and contains a table with three rows: "Profile", "alen Profile", and "nabeel4477 Profile". Each row has a checkbox and a "Go" button. The table also shows "3 profiles" at the bottom.

## FUTURE ENHANCEMENT OR IMPROVEMENTS

**Two-Factor Authentication-** Implementing 2FA adds an extra layer of security for user logins.

**Personalized Recommendations-** Utilize algorithms to provide users with personalized blog recommendations

GitLink -

<https://github.com/melbinmartin/safari.git>