

BlogIt: A Flask WebApp Documentation

**By
Rohit Upadhyay**

Table of Contents

- **Introduction**
 - 1.1 Overview
 - 1.2 Features
 - Dependencies
- **Getting Started**
 - 2.1 Installation
 - 2.2 Configuration
 - 2.3 Run the Application
- **User Guide**
 - 3.1 Sign-up
 - 3.2 Login
 - 3.3 Create a Blog Post
 - 3.4 Publish a Blog Post
 - 3.5 View Blog Posts
- **Database (DynamoDB)**
 - 4.1 Schema
 - 4.2 Interacting with DynamoDB
- **Authentication**
 - 5.1 Password Encryption
 - 5.2 Flask-Session
- **Troubleshooting**
 - 6.1 Common Issues
 - 6.2 FAQs
- **Future Improvements**
- **License**

1. Introduction

- **Overview**

BlogIt is a Flask web application that allows users to sign up, log in, create, publish, and view blog posts created by other users. The application uses DynamoDB to store user login details and blog posts.

- **Features**

- User authentication (sign-up and login)
- Create and publish blog posts
- Like posts
- Filter According to number of likes or latest
- Search for posts
- View blog posts created by other users

- **Dependencies**

- Flask
- Flask-Session
- Boto3 (AWS SDK for Python)
- DynamoDB (Amazon DynamoDB)

2. Getting Started

- **Installation**

To install BlogIt and its dependencies, run the following commands:

```
pip install -r requirements.txt
```

- **Configuration**

Update the configuration file key-config.py with your AWS DynamoDB credentials.

- **Run the Application**

Run the Flask application using:

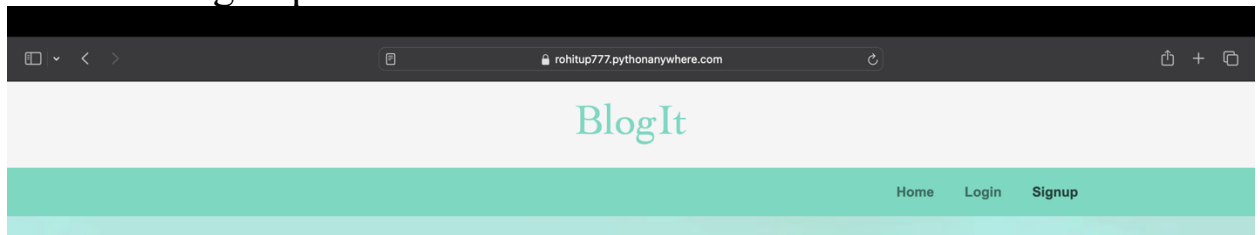
```
python3 app.py
```

Access the application at <http://localhost:7000> in your web browser. (Only if you want to run locally)
(Note: The application is already hosted using pythonanywhere.com, so all you have to do is to visit the website.

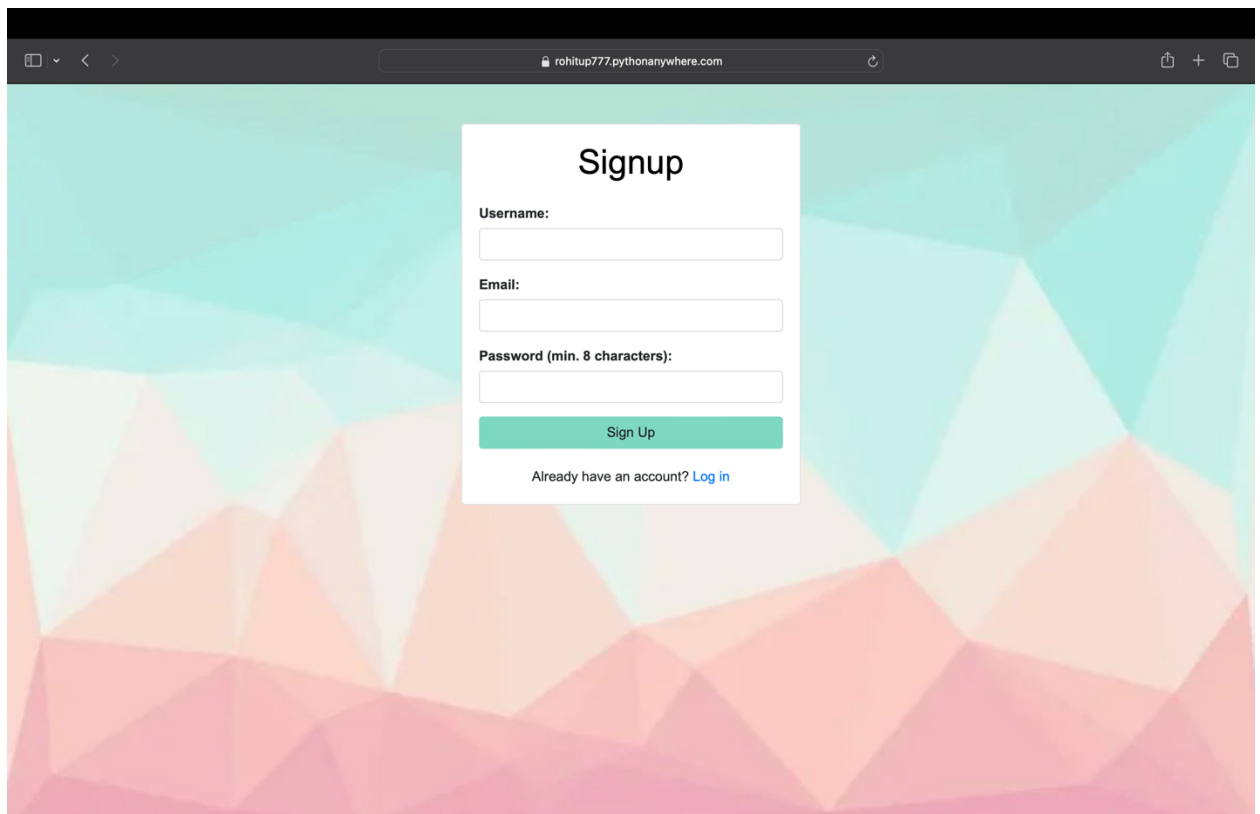
3. User Guide

- **Sign-up**

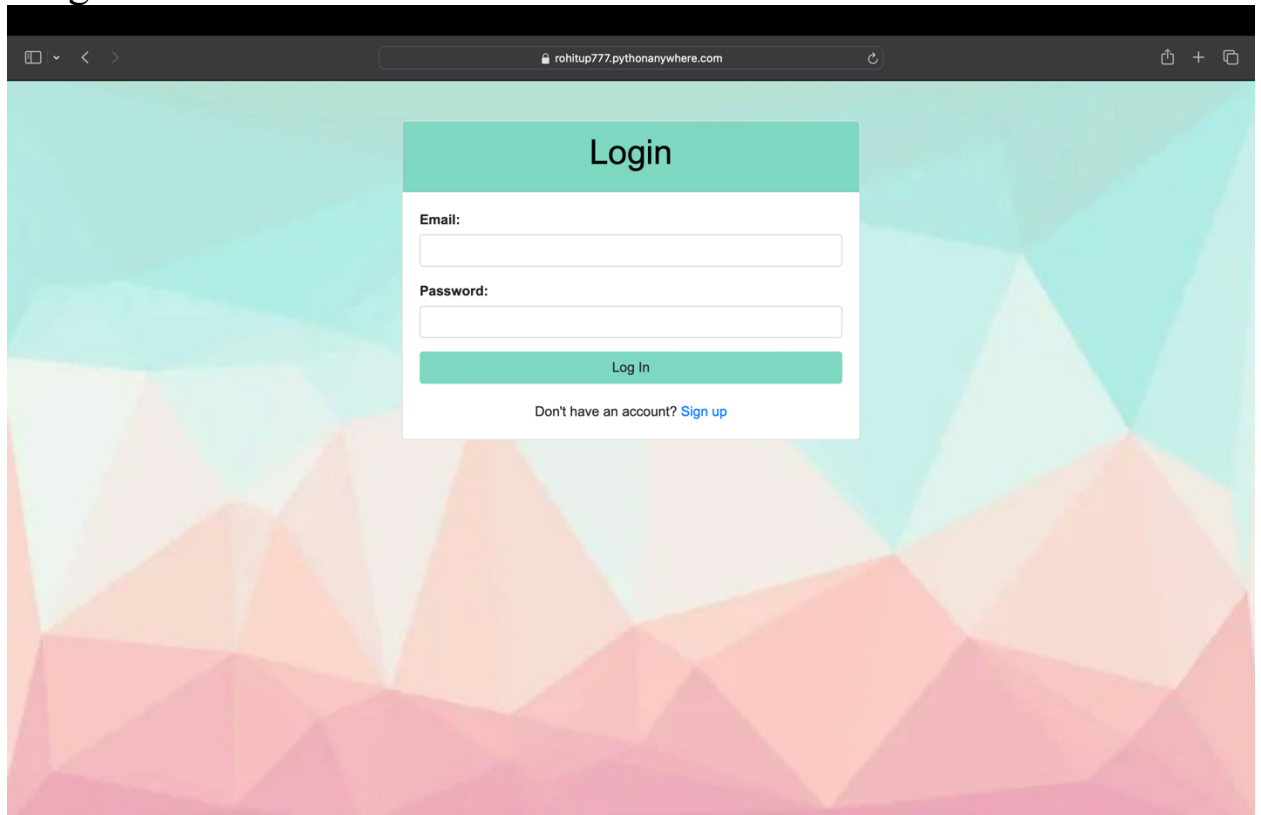
-Click on Sign-up button on the nav bar



-Enter the details



-Login with the entered details



rohitup777.pythonanywhere.com

Login

Email:

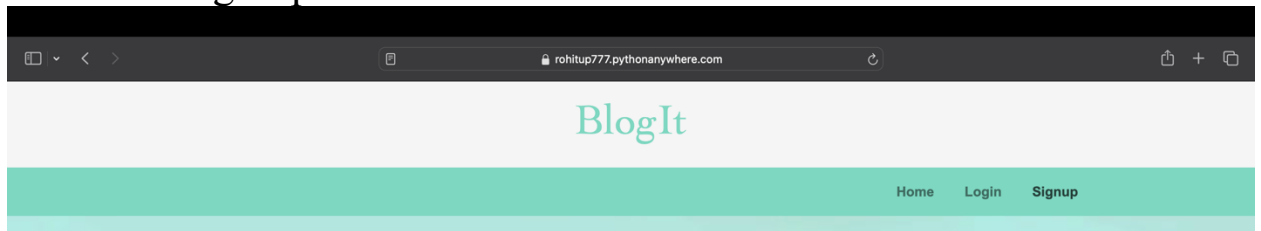
Password:

Log In

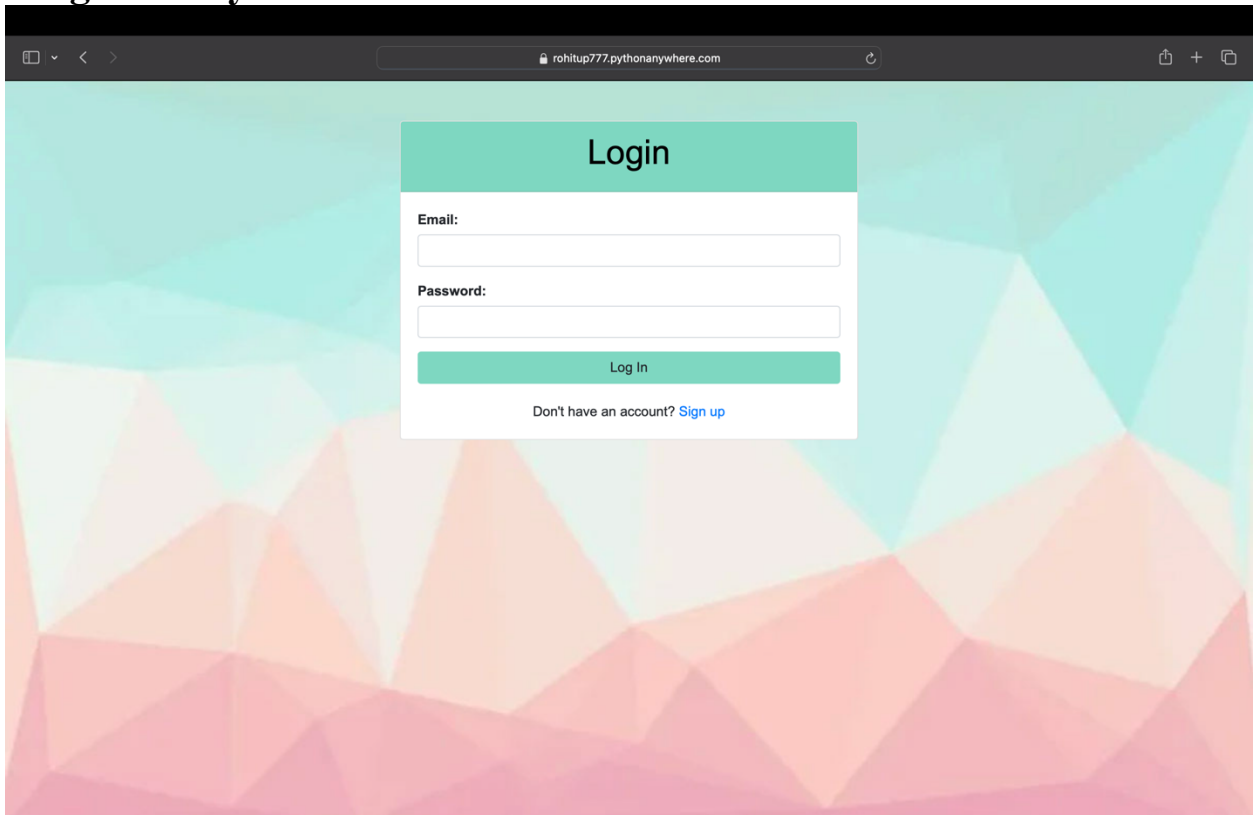
Don't have an account? [Sign up](#)

- **Login**

-Click on Sign-up button on the nav bar



-Login with your credentials



The screenshot shows a web browser window with a dark address bar displaying the URL `rohitup777.pythonanywhere.com`. The page features a vibrant, low-poly geometric background in shades of teal, orange, and pink. Centered on the page is a white login form with a teal header. The form includes fields for 'Email:' and 'Password:', a teal 'Log In' button, and a link for users without an account.

Login

Email:

Password:

[Log In](#)

Don't have an account? [Sign up](#)

- **Create a Blog Post**

The screenshot shows a web browser window with the URL `rohitup777.pythonanywhere.com`. The page is titled "BlogIt" and has navigation links for "Home", "Create Post", and "Logout". The main content area is titled "Create a New Post" and contains the following form fields:

- Title:** A text input field.
- Content (max 300 words):** A text area with a word count indicator showing "Words: 0/300".
- Tags (comma-separated):** A text input field.

Below the form fields are two buttons: a blue "Publish" button and a blue "Back to Home" link.

- **View Blog Posts**

The screenshot shows the "View Blog Posts" page in the BlogIt application. The page has a header with the "BlogIt" logo and navigation links for "Home", "Create Post", "Profile", and "Logout". Below the header is a search bar with a "Search" button and a "Latest" dropdown menu with an "Apply Filter" button. The main content area displays a list of blog posts. The first post is titled "GTA VI is coming 2025!" and is attributed to "by: tony@stark.com". The post text discusses the release of the Grand Theft Auto 6 trailer and the game's release date. The second post is titled "The Great Escape of Sir Fluffington" and is also attributed to "by: tony@stark.com".

GTA VI is coming 2025!
by: tony@stark.com

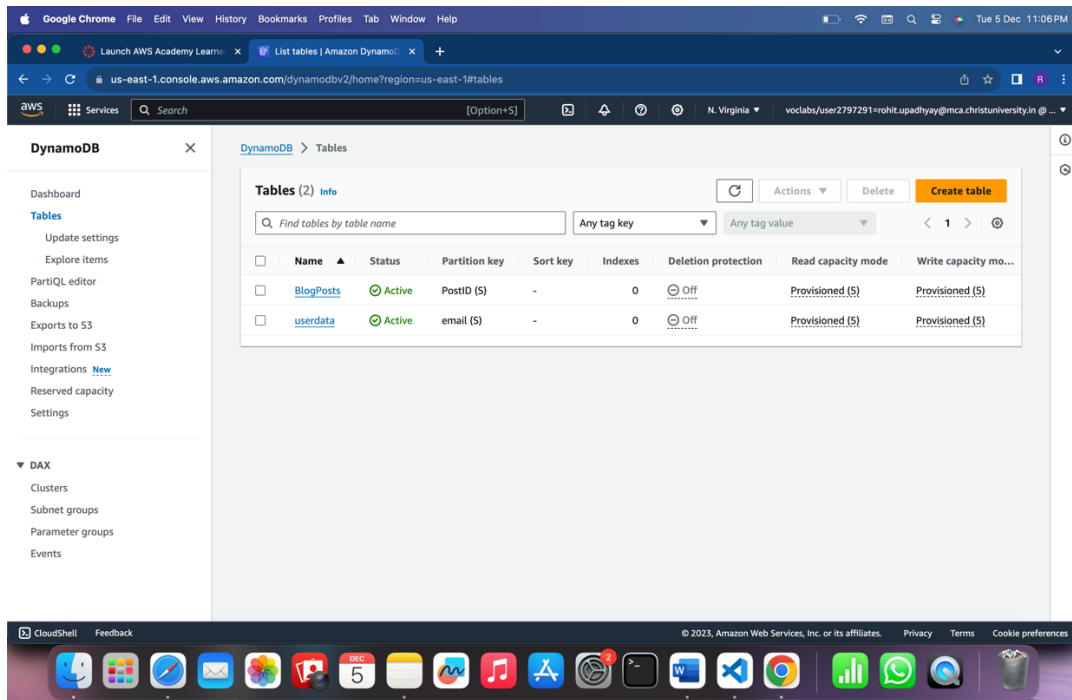
Rockstar Games finally released the trailer for its long-awaited Grand Theft Auto 6 on Tuesday, hours ahead of schedule thanks to a supposed trailer leak on social media platform X (formerly Twitter). As expected, the trailer quickly went viral on YouTube, with the video garnering over 11 million views in the 2-hour window since it was posted on the official Rockstar Games channel on YouTube. The GTA 6 trailer confirms many of the leaks that have been circulating around the open-world game, while a subsequent press release from Rockstar Games revealed that the game will be coming to PlayStation 5 and Xbox Series X and Series S in 2025. However, there was no update on when or if the game would be coming to PC, much to the dismay of many PC gaming enthusiasts. While confirming that the trailer for the upcoming game has indeed been leaked, Rockstar Games said on X: "Our trailer has leaked so please watch the real thing on YouTube" The trailer leak comes on the heels of a GTA 6 gameplay video going viral on TikTok, which gave a sneak peek at the highly anticipated gameplay and map of upcoming game, and sent the gaming community into a frenzy. The GTA 6 trailer introduces the series' first female protagonist, Lucia, who is seen in a prison at the start of the trailer. Later in the video, Lucia and her boyfriend are seen pulling off Bonnie and Clyde-style heists in Vice City. In a statement about GTA 6, Rockstar Games said: "Grand Theft Auto VI heads to the state of Leonida, home to the neon-soaked streets of Vice City and beyond in the biggest, most immersive evolution of the Grand Theft Auto series yet."

6

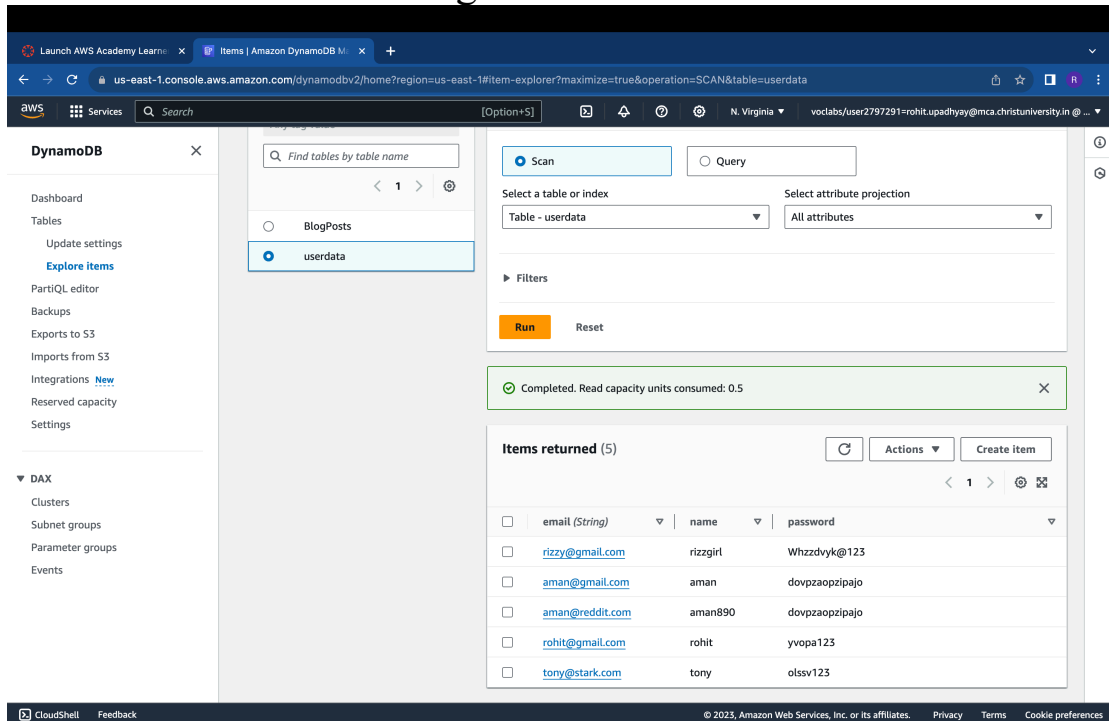
The Great Escape of Sir Fluffington
by: tony@stark.com

4. Database (DynamoDB)

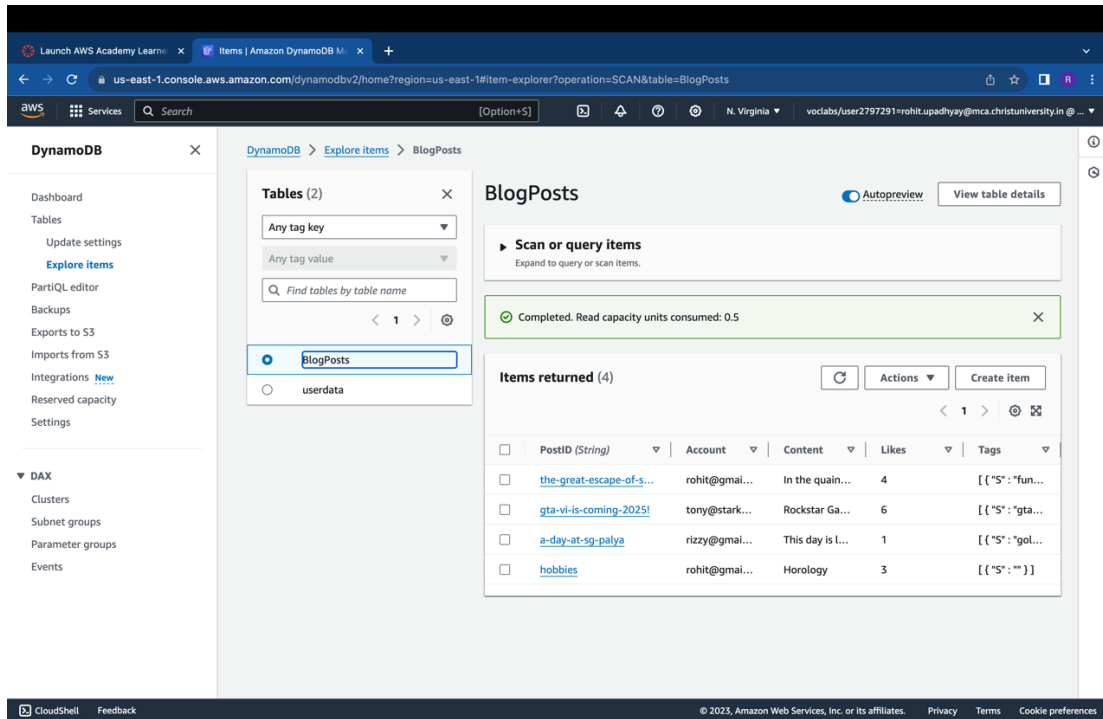
• Schema



-User Data table containing the account details



-BlogPosts table containing all the post information



5. Authentication

- **Password Encryption**

- **Alphabetic Encryption:**

The encrypt function is designed to encrypt text by shifting each alphabetic character by a specified amount (shift). It maintains the case (uppercase or lowercase) of the original character during encryption.

- **Modular Arithmetic for Circular Shift:**

To achieve the Caesar cipher-like shift, the function utilizes modular arithmetic. It calculates the new position of each character in the alphabet by taking the remainder after division by 26. This ensures a circular shift within the alphabet, allowing the encryption to loop back to the beginning after reaching 'z' or 'Z'.

- **Non-Alphabetic Characters Remain Unchanged:**

The function is designed to leave non-alphabetic characters unchanged during encryption. This ensures that spaces,

punctuation, and any other non-alphabetic symbols in the input text are preserved in the encrypted output. The conditional check using `isalpha()` helps identify and process only alphabetic characters.

- **Flask-Session**

- **Stateful Web Applications:**

- Flask sessions provide a way to persist user-specific data across multiple requests and responses in a web application. This is essential for maintaining stateful interactions, allowing the server to recognize and remember users as they navigate through different pages or perform various actions on the site.

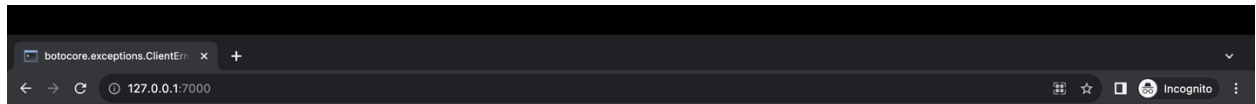
- **Secure Cookie-Based Implementation:**

- Flask sessions use a secure and tamper-resistant cookie-based implementation to store session data on the client side. The session data is encrypted and signed, preventing users from tampering with or manipulating their session information. This helps enhance the security of the web application by ensuring that the session data cannot be easily compromised or forged by malicious users.

6. Troubleshooting

- **Common Issues**

- The below error is caused when the AWS session credentials reset after 4 hours. The `key-config.py` file needs to be updated with the new session details.



ClientError

botocore.exceptions.ClientError: An error occurred (AccessDeniedException) when calling the Scan operation: User: arn:aws:sts::198724184674:assumed-role/voclabs/user2797291=rohit.upadhyay@mca.christuniversity.in is not authorized to perform: dynamodb:Scan on resource: arn:aws:dynamodb:us-east-1:198724184674:table/BlogPosts with an explicit deny in an identity-based policy

Traceback (most recent call last)

```
File "opt/anaconda3/lib/python3.9/site-packages/flask/app.py", line 2548, in __call__
    return self.wsgi_app(environ, start_response)
File "opt/anaconda3/lib/python3.9/site-packages/flask/app.py", line 2528, in wsgi_app
    response = self.handle_exception(e)
File "opt/anaconda3/lib/python3.9/site-packages/flask/app.py", line 2525, in wsgi_app
    response = self.full_dispatch_request()
File "opt/anaconda3/lib/python3.9/site-packages/flask/app.py", line 1822, in full_dispatch_request
    rv = self.handle_user_exception(e)
File "opt/anaconda3/lib/python3.9/site-packages/flask/app.py", line 1820, in full_dispatch_request
    rv = self.dispatch_request()
File "opt/anaconda3/lib/python3.9/site-packages/flask/app.py", line 1796, in dispatch_request
    return self.ensure_sync(self.view_functions[rule.endpoint])(**view_args)
File "Users/rohitupadhyay/Desktop/trying/app.py", line 83, in index
    response = table.scan()
File "opt/anaconda3/lib/python3.9/site-packages/boto3/resources/factory.py", line 580, in do_action
    response = action(self, *args, **kwargs)
File "opt/anaconda3/lib/python3.9/site-packages/boto3/resources/action.py", line 88, in __call__
    response = getattr(parent.meta.client, operation_name)(*args, **params)
File "opt/anaconda3/lib/python3.9/site-packages/botocore/client.py", line 415, in _api_call
    return self._make_api_call(operation_name, kwargs)
File "opt/anaconda3/lib/python3.9/site-packages/botocore/client.py", line 745, in _make_api_call
    raise error_class(parsed_response, operation_name)
```

botocore.exceptions.ClientError: An error occurred (AccessDeniedException) when calling the Scan operation: User: arn:aws:sts::198724184674:assumed-role/voclabs/user2797291=rohit.upadhyay@mca.christuniversity.in is not authorized to perform: dynamodb:Scan on resource: arn:aws:dynamodb:us-east-1:198724184674:table/BlogPosts with an explicit deny

• FAQs

○ Q: How do I create a new account on BlogIt?

Ans: To create a new account, go to the signup page and provide your name, email, and password. Click the "Sign Up" button to register. Make sure your email is unique and not already registered.

○ Q: Can I use special characters in my password during registration?

Ans: Yes, you can use special characters in your password. However, ensure that your password meets any specified requirements, and note that the password is case-sensitive.

○ Q: What should I do if I forget my password?

Ans: If you forget your password, there is currently no password recovery functionality. Consider creating a new account with a different email address.

○ Q: How can I create and publish a blog post?

Ans: After logging in, navigate to the "Create" page. Enter the title, content, and tags for your blog post and click "Publish." Your post will then be visible to other users.

○ **Q: Can I edit or delete a blog post after publishing it?**

Ans: No, the current version of BlogIt does not support editing or deleting blog posts after they have been published. Consider carefully reviewing your content before publishing.

○ **Q: How is user authentication handled in BlogIt?**

Ans: User authentication is managed through a Flask session. When you log in, your email is stored in the session, allowing you to stay authenticated as you navigate through the app.

○ **Q: What happens if I try to sign up with an email that already exists?**

Ans: If you attempt to sign up with an email that already exists in the system, you will be redirected to the login page. Ensure that your email is unique during the registration process.

○ **Q: Is there an API for interacting with BlogIt programmatically?**

Ans: The current version of BlogIt does not provide a public API. All interactions are done through the web interface.

○ **Q: How is user data stored in the DynamoDB database?**

Ans: User data, including login details, is stored in the DynamoDB 'userdata' table. Passwords are encrypted before being stored to enhance security.

○ **Q: Can I customize the appearance of my blog posts?**

Ans: The current version of BlogIt does not support customizing the appearance of blog posts. Blog posts are displayed in a standard format with the title, content, and tags.

7. Future Improvements

The following future improvements can be made if given more time:

- Better UI
- AWS hosting and Elastic Load Balancing to handle traffic
- Option to dislike posts
- Option to add pictures while posting
- Option to follow accounts

8. License

This project is licensed under the MIT License.