# Documentation: Flask User Registration System with SMS Verification

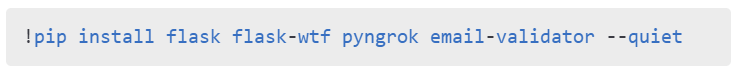
## How to Test the App

**Step-by-step testing guide:**

1. **Run the entire Python script** in a Google Colab notebook or local environment where all dependencies are installed.
2. **Check the console for the ngrok public URL** (e.g., https://xxxx-xx-xx-xx.ngrok.io).
   * Click or open it in a new browser tab.
3. **On the landing page:**
   * Fill out the registration form.
   * Make sure to accept the Terms of Service.
   * Submit the form.
4. **Check the console log for the mock SMS code** (shown in the browser for testing).
   * Enter the code on the verification page.
5. **After successful verification:**
   * You will be redirected to the Welcome page.
6. **Click Login** on the welcome or registration page:
   * Enter the registered username or email and password.
   * Submit and then enter the new verification code from console.
7. **You should now be logged in again and see the Welcome page.**
8. **To review stored users:**
   * Open /content/registered\_users.json to view user data in JSON format.
9. **Test invalid inputs:**
   * Try short passwords, incorrect confirmation, or duplicate usernames to verify error handling.

## Understanding the Code:

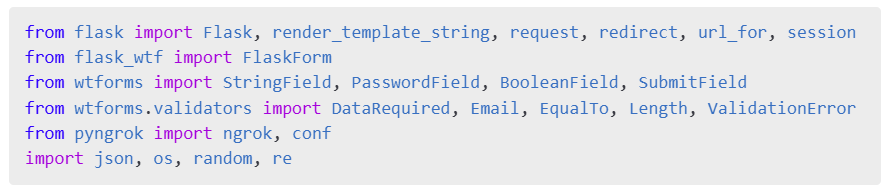
**1. Install Dependencies**



**Purpose:** Install required packages:

* flask: micro web framework for building the app
* flask-wtf: enables use of WTForms in Flask (form validation)
* pyngrok: expose local Flask server to the public via ngrok tunnel
* email-validator: ensures email inputs are valid

**2. Import Required Libraries**

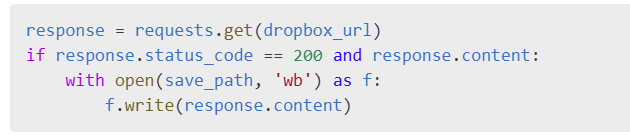
****

**Purpose:** Import Flask tools, form fields, validation tools, ngrok config, and Python standard libraries.

**Code Explanation of Imports**

* from flask import ... — Imports essential functions from Flask:
  + Flask: used to create the Flask app
  + render\_template\_string: renders inline HTML templates
  + request: accesses form data submitted by users
  + redirect: redirects the user to another route
  + url\_for: builds a URL for a given function
  + session: handles session data (like SMS code storage)
* from flask\_wtf import FlaskForm — Base class for forms using Flask-WTF integration with CSRF protection.
* from wtforms import ... — Form field types:
  + StringField: single-line text input (e.g., username)
  + PasswordField: masked input for passwords
  + BooleanField: checkbox (e.g., terms agreement)
  + SubmitField: submit button
* from wtforms.validators import ... — Validators for form field data:
  + DataRequired: ensures field is not empty
  + Email: checks if input is a valid email
  + EqualTo: confirms matching fields (e.g., password and confirm password)
  + Length: checks string length
  + ValidationError: used for raising custom validation messages
* from pyngrok import ngrok, conf — Used to connect and configure ngrok tunneling for local Flask server.
* import json, os, random, re — Standard Python libraries:
  + json: read/write JSON files (e.g., registered users)
  + os: file operations like checking if a file exists
  + random: generate verification code
* re: use regular expressions for password validation

**3. Download JSON File from Dropbox**

****

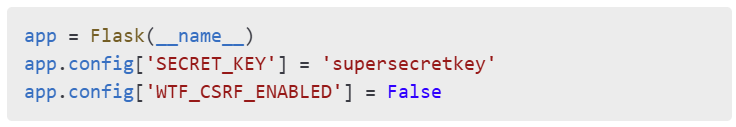
**Purpose:** Downloads registered\_users.json from Dropbox to store user registrations locally in Colab.

**4. Set ngrok Token**



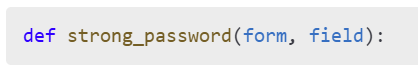
**Purpose:** Authenticates ngrok so it can expose the Flask server to the public.

**5. Initialize Flask App**



**Purpose:** Creates Flask application and sets up configuration like CSRF disabling (useful for development).

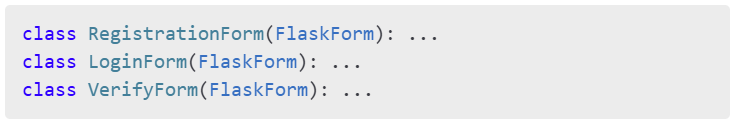
**6. Custom Password Validator Function**



**Purpose:** Custom logic to ensure the password is strong, checks:

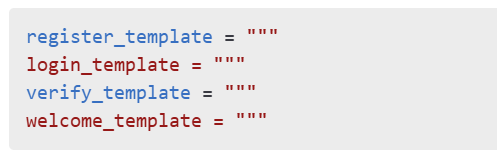
* length
* number
* special characters
* similarity with username or email

**7. Define WTForms**

****

**Purpose:** Define forms for user registration, login, and SMS code verification.

**8. Define HTML Templates**

****

**Purpose:** Contains inline HTML templates with Bootstrap styling for each page.

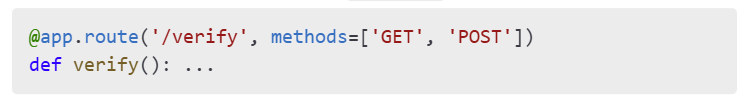
**9. Registration Route (/)**

****

**Purpose:** Handles new user registration:

* Validates form
* Checks if user/email already exists
* Sends SMS code
* Stores to pending\_user.json

**10. Verify Registration Route (/verify)**

**Purpose:** Validates SMS code:

* If valid, stores user in registered\_users.json
* Redirects to welcome page

**11. Login Route (/login)**

**Purpose:** Authenticates user login:

* Matches email/username and password
* Sends login verification code

**12. Login Verification Route (/verify-login)**



**Purpose:** Verifies login code:

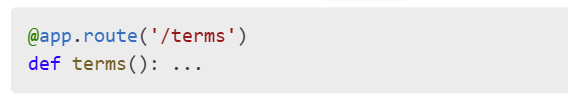
* If valid, redirects to welcome page

**13. Welcome Route (/welcome)**

****

**Purpose:** Displays welcome message with user name.

**14. Terms of Service Route (/terms)**

****

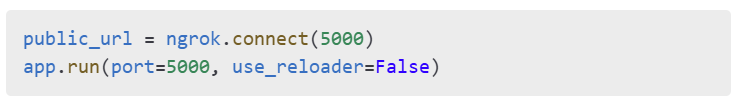
**Purpose:** Displays TOS agreement. Linked in registration form.

**15. SMS Code Sender Function**

****

**Purpose:** Mocks sending SMS for demonstration (prints code to console).

**16. Start Flask App with ngrok**

****

**Purpose:**

* Exposes Flask app via public ngrok URL
* Starts the Flask app for interaction