

Deploying a Python Flask Web App to Azure Using GitHub

Goal:

The aim of this project was to create a simple Python Flask web application, deploy it to Azure App Service, and integrate it with GitHub for continuous deployment.

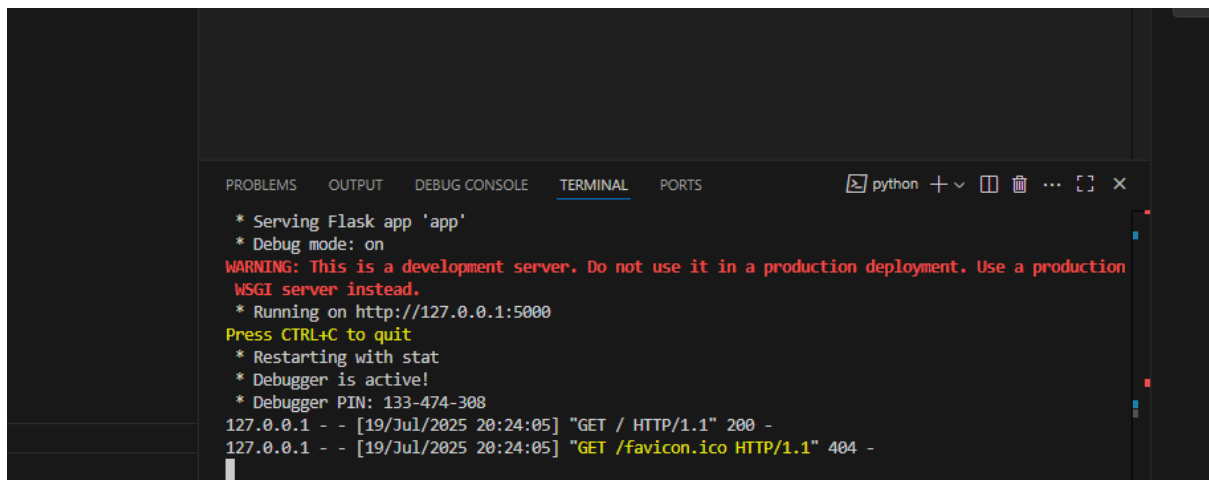
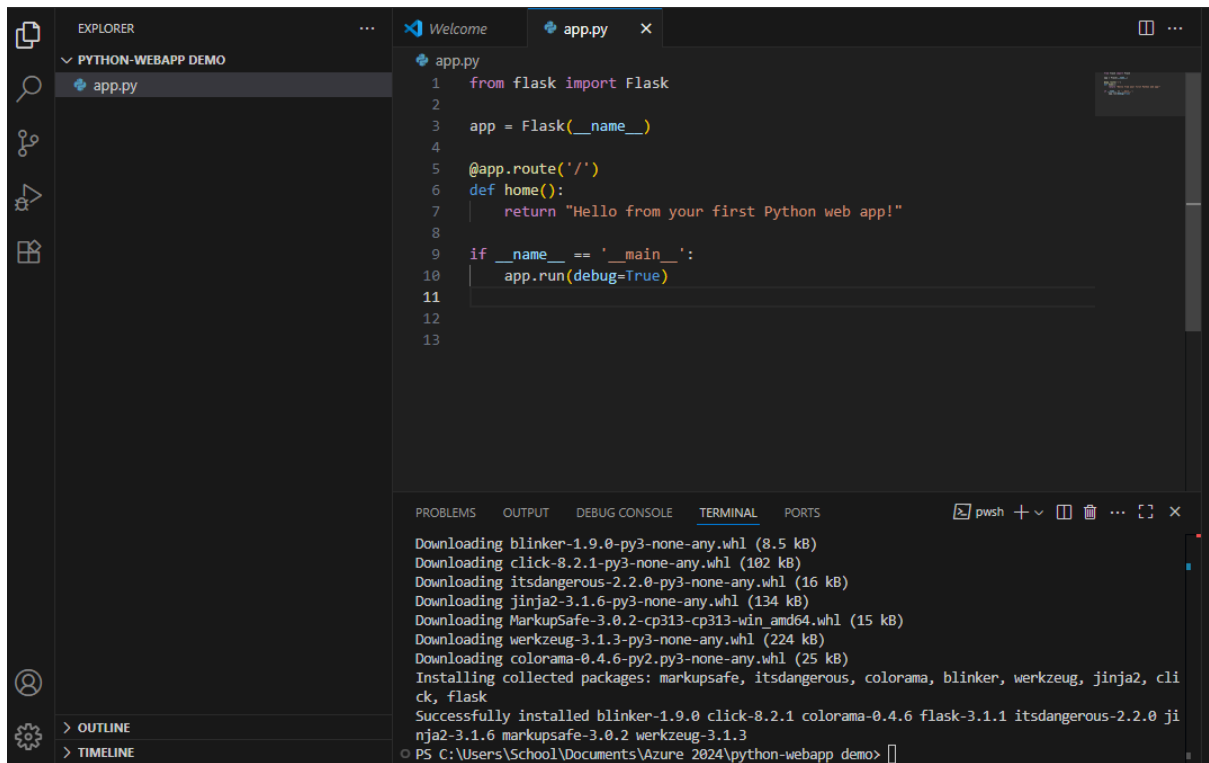
Step 1 - Creating the Flask App Locally

I started by setting up a basic Flask application with two routes:

- Home route, displaying “Hello from your first Python web app”.
- About route, displaying “This is the about page”.

Screenshots:

1. Downloading and installing dependencies (Blinker 1.0 and Flask setup).
2. Running the app in debug mode locally.
3. Accessing the Home route in the browser.
4. Accessing the About page.
5. Updating routes to render HTML templates (home.html).



The image shows a Visual Studio Code editor window with a dark theme. The Explorer sidebar on the left shows a project named 'PYTHON-WEBAPP DEMO' with a file 'app.py'. The main editor area displays the code for 'app.py':

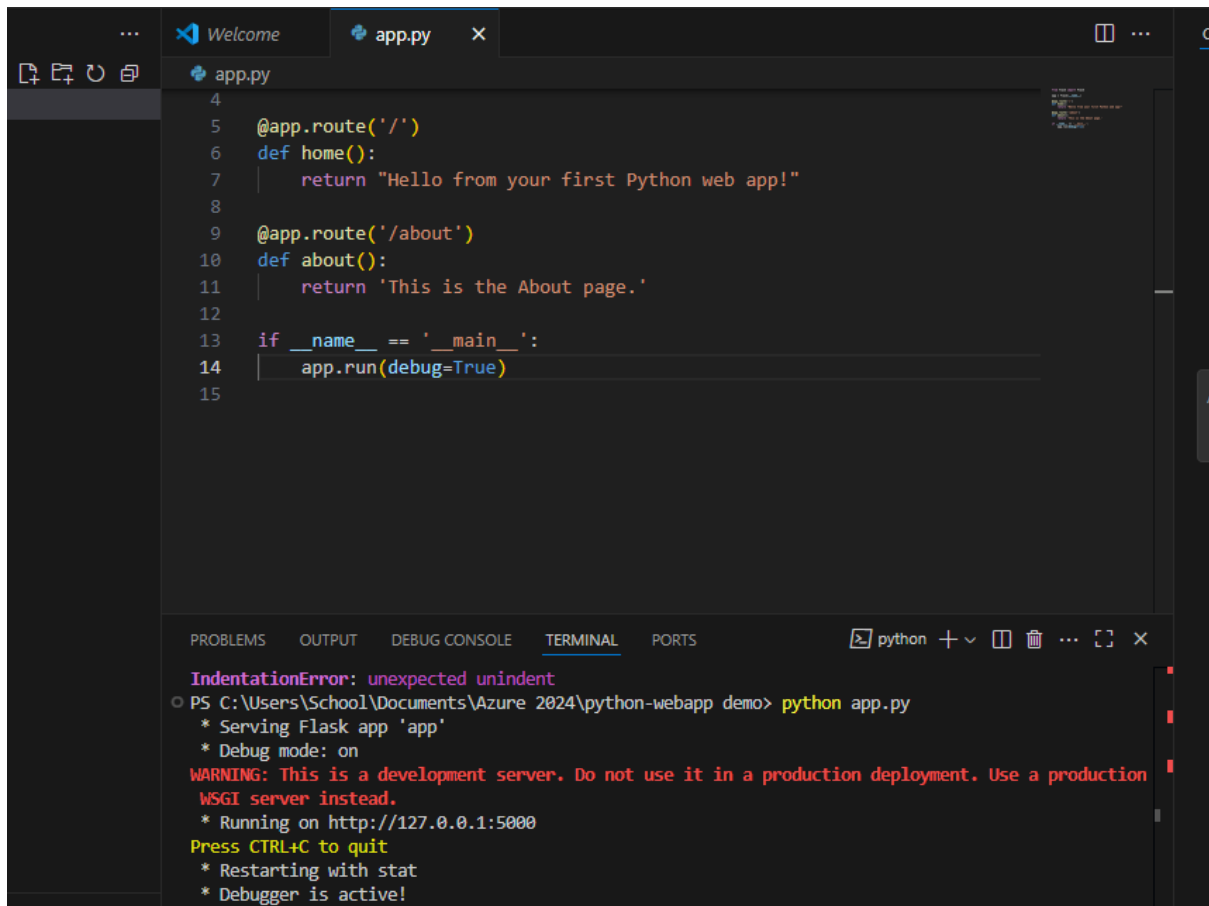
```
4
5 @app.route('/')
6 def home():
7     return "Hello from your first Python web app!"
8
9 @app.route('/about')
10 def about():
11     return 'This is the About page.'
12
13 if __name__ == '__main__':
14     app.run(debug=True)
15
```

Below the code editor is a panel with tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'PORTS'. The 'TERMINAL' tab is active, showing the output of running the application:

```
PS C:\Users\School\Documents\Azure 2024\python-webapp demo> python app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production
WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
```



Hello from your first Python web app!



The image shows a Visual Studio Code editor window with a file named `app.py` open. The code is a simple Flask web application with two routes: `home()` and `about()`. The `home()` route returns "Hello from your first Python web app!" and the `about()` route returns "This is the About page.". The application is run with `app.run(debug=True)`.

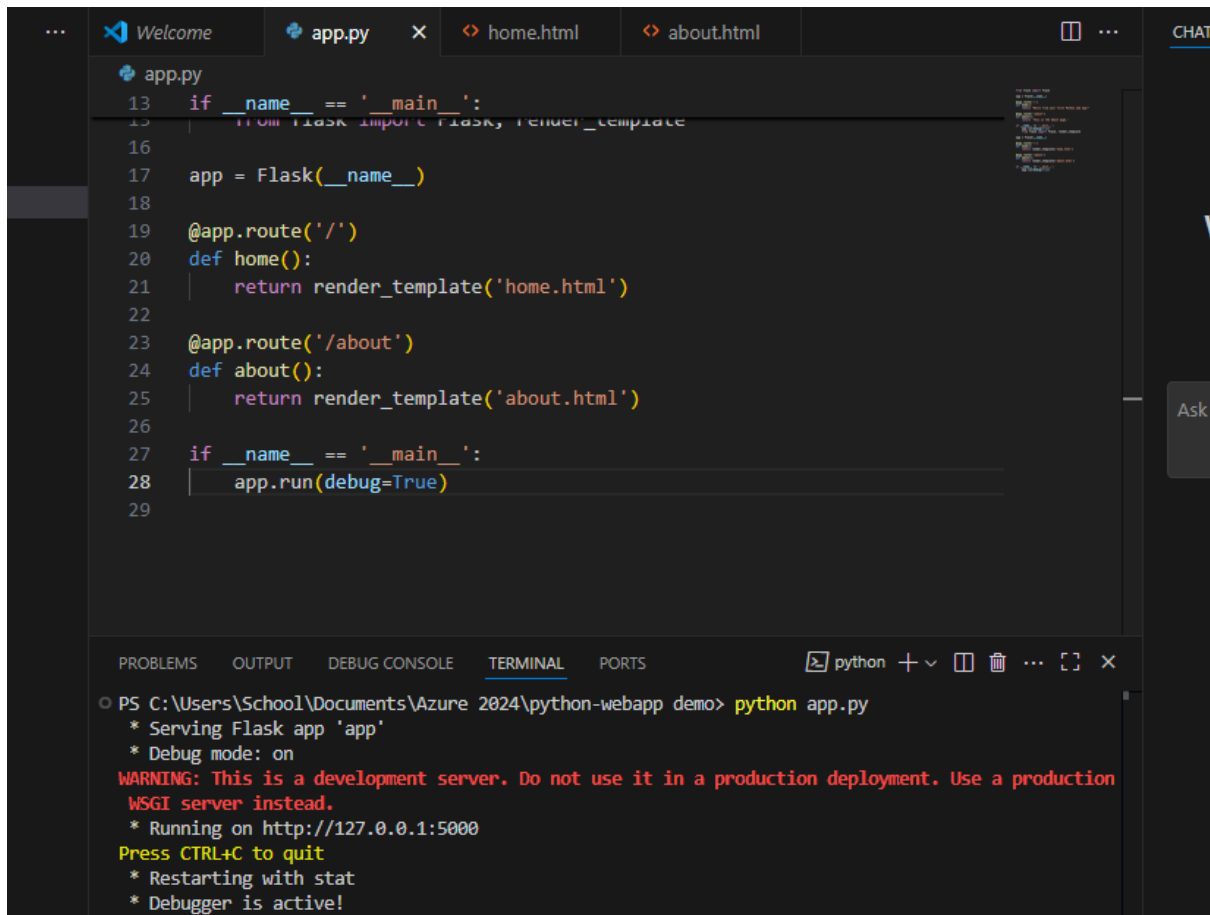
```
4
5 @app.route('/')
6 def home():
7     return "Hello from your first Python web app!"
8
9 @app.route('/about')
10 def about():
11     return 'This is the About page.'
12
13 if __name__ == '__main__':
14     app.run(debug=True)
15
```

Below the code editor, the TERMINAL panel is active, showing the output of running `python app.py`. It displays the Flask server starting in debug mode on `http://127.0.0.1:5000`. A warning message is shown: "WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead." The terminal also shows that the server is restarting with `stat` and that the debugger is active.

```
PS C:\Users\School\Documents\Azure 2024\python-webapp demo> python app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production
WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
```



This is the About page.

The image shows a Visual Studio Code editor window with a dark theme. The top bar shows several open files: 'Welcome', 'app.py', 'home.html', and 'about.html'. The 'app.py' file is active and displays the following Python code:

```
13 if __name__ == '__main__':  
14     from flask import Flask, render_template  
15  
16  
17 app = Flask(__name__)  
18  
19 @app.route('/')  
20 def home():  
21     return render_template('home.html')  
22  
23 @app.route('/about')  
24 def about():  
25     return render_template('about.html')  
26  
27 if __name__ == '__main__':  
28     app.run(debug=True)  
29
```

The bottom panel of the editor shows the 'TERMINAL' tab. It contains the output of running the command 'python app.py' in a PowerShell prompt. The output includes a warning about using a development server and the application running on http://127.0.0.1:5000.

```
PS C:\Users\School\Documents\Azure 2024\python-webapp demo> python app.py  
* Serving Flask app 'app'  
* Debug mode: on  
WARNING: This is a development server. Do not use it in a production deployment. Use a production  
WSGI server instead.  
* Running on http://127.0.0.1:5000  
Press CTRL+C to quit  
* Restarting with stat  
* Debugger is active!
```

Step 2 - Setting Up Azure Resources

I logged into Azure and created the following resources:

- Resource Group for the project.
- App Service Plan to host the app.
- Web App for deployment.

Screenshots:

6. Azure Resource Group creation.
7. App Service Plan setup.
8. Web App created.
9. Default Azure Web App “running and waiting for content” page.



flask-rg

Resource group



How do I monitor this resource group?

Are there any a



Create



Manage view



Delete resource



Overview



Activity log



Access control (IAM)



Tags



Resource visualizer



Events



Settings



Cost Management



Monitoring



Automation



Help

Essentials

Resources

Recommendations

Filter for any field...

Type equals **all**



Showing 0 to 0 of 0 records.



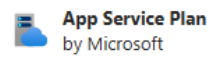
Show hidden types

Name ↑↓

Create App Service Plan ...

Basics Tags Review + create

Summary



Details

Subscription	Azure subscription 1
Resource Group	flask-rg
Name	flask-appserviceplan
Operating System	Linux
Region	South Africa North
SKU	Free
ACU	Shared infrastructure
Memory	1 GB memory

Create Web App ...

Basics Database Deployment Networking Monitor + secure Tags Review + create

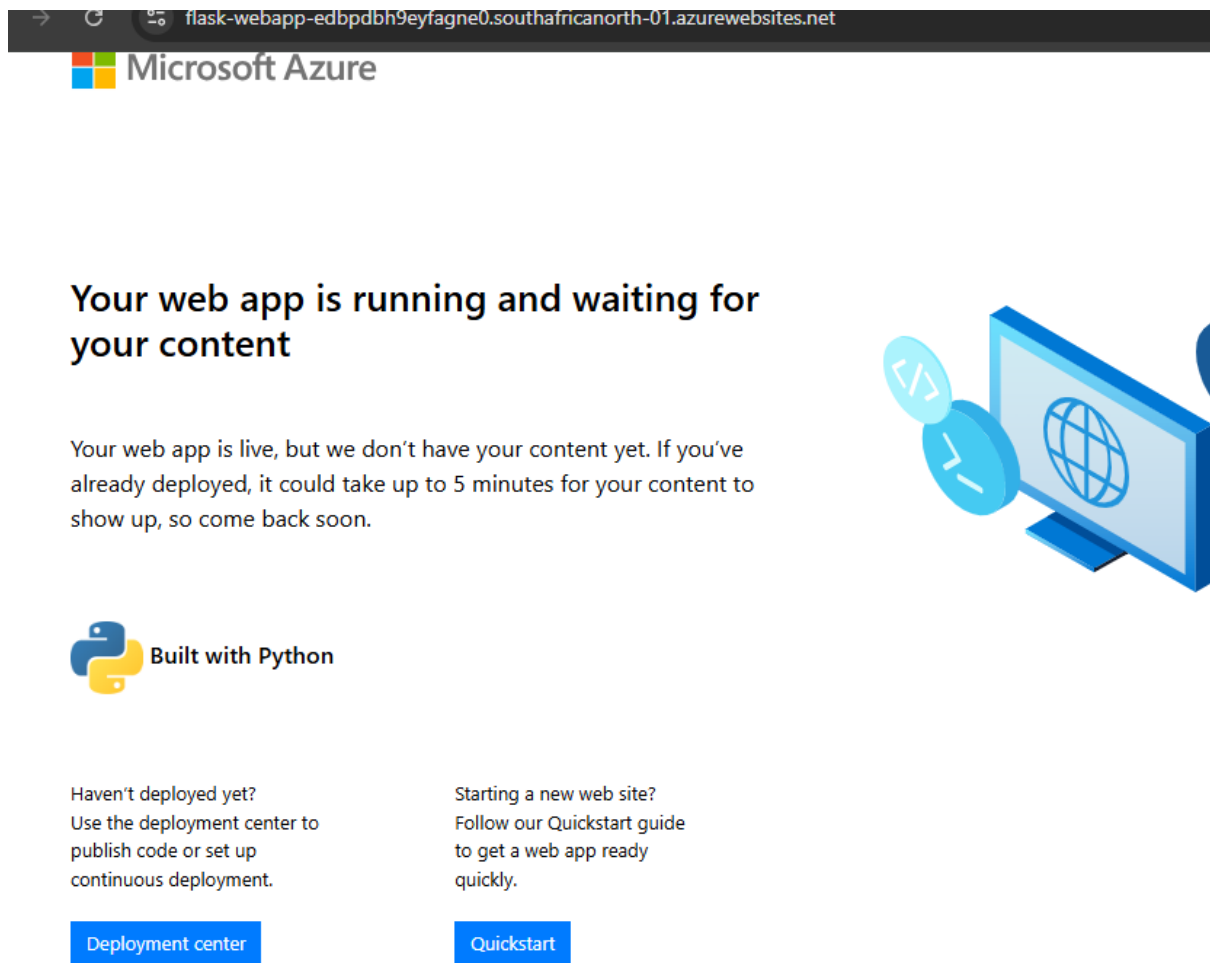
Summary



Basic authentication for this app is currently disabled and may impact deployments. Click to learn more.

Details

Subscription	a4fa4fb6-daa1-43d2-9d72-42a1674464b0
Resource Group	flask-rg
Name	flask-webapp
Secure unique default hostname	Enabled
Publish	Code
Runtime stack	Python 3.11



Step 3 - Configuring GitHub for Deployment

I initialized Git in the project folder and set my global configuration:

- Username and email for Git commits.
- Initial commit for the Flask app.

I then connected the local repository to GitHub:

- Added the remote URL for the repository.
- Pushed the project files to GitHub.

Screenshots:

10. Git config setup commands.
11. GitHub sign-in in the browser.
12. Adding the GitHub remote URL.
13. GitHub repository view showing project files (templates/, p.py, requirements.txt, startup.txt).

```
fatal: unable to auto-detect email address (got 'School@LAPTOP-LRHFAHF0.(none)')
PS C:\Users\School\Documents\Azure 2024\python-webapp demo> git config --global user.name "Eunice"
PS C:\Users\School\Documents\Azure 2024\python-webapp demo> git config --global user.email "tshasumaeunice@gmail.com"
PS C:\Users\School\Documents\Azure 2024\python-webapp demo> git commit -m "Initial commit for flask app"
[master (root-commit) b32ae9f] Initial commit for flask app
5 files changed, 39 insertions(+)
create mode 100644 p.py
create mode 100644 requirements.txt
create mode 100644 startup
create mode 100644 templates/about.html
create mode 100644 templates/home.html
PS C:\Users\School\Documents\Azure 2024\python-webapp demo> █
```



The screenshot shows a terminal window with a GitHub sign-in dialog box overlaid. The dialog box is titled "Connect to GitHub" and has a close button (X) in the top right corner. It features the GitHub logo and the text "Sign in". Below this, there are two tabs: "Browser/Device" (which is selected) and "Token". Under the "Browser/Device" tab, there are two buttons: "Sign in with your browser" (in blue) and "Sign in with a code" (in grey). At the bottom of the dialog, it says "Don't have an account? [Sign up](#)".

The terminal window shows the following commands and output:

```
PS C:\Users\School\Documents\Azure 2024\python-webapp demo> git config --global user.name "Eunice"
PS C:\Users\School\Documents\Azure 2024\python-webapp demo> git config --global user.email "tshasumaeunice@gmail.com"
PS C:\Users\School\Documents\Azure 2024\python-webapp demo> git commit -m "Initial commit for flask app"
[master (root-commit) b32ae9f] Initial commit for flask app
5 files changed, 39 insertions(+)
create mode 100644 p.py
create mode 100644 requirements.txt
create mode 100644 startup
create mode 100644 templates/about.html
create mode 100644 templates/home.html
PS C:\Users\School\Documents\Azure 2024\python-webapp demo> █
```

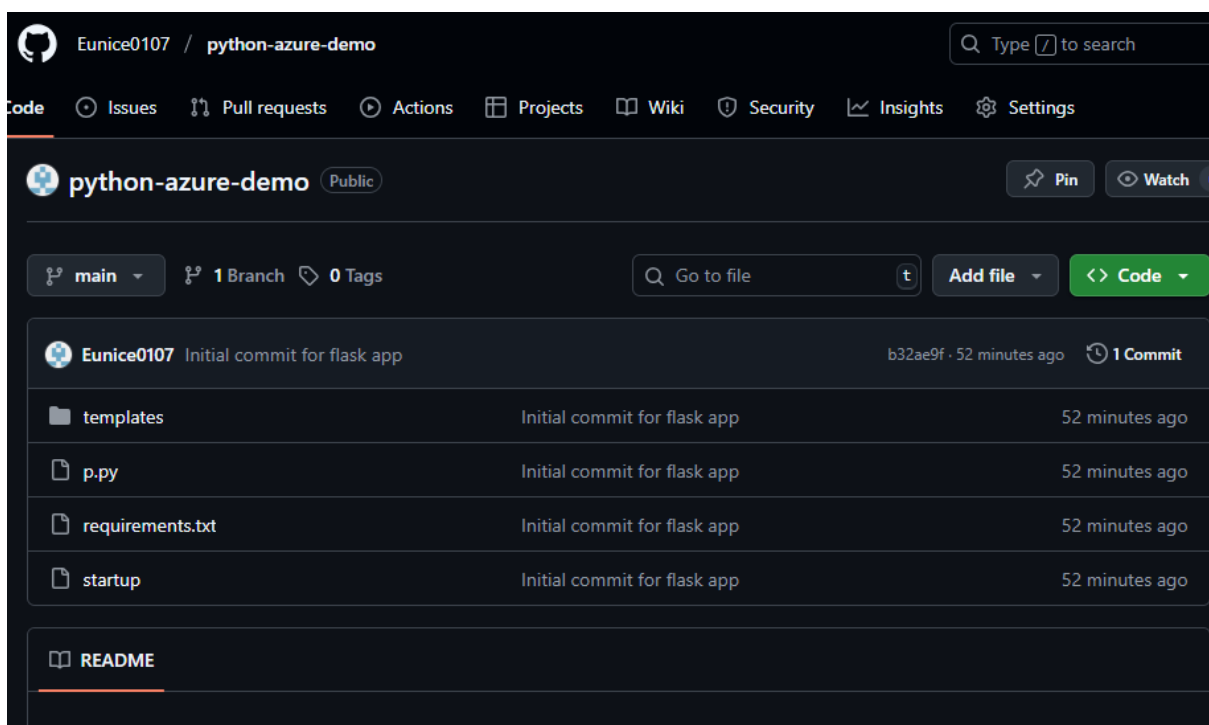
Below the dialog box, the terminal shows the following commands and output:

```
PS C:\Users\School\Documents\Azure 2024\python-webapp demo> git push -u origin main
does not appear to be a git repository
head from remote repository.

you have the correct access rights
y exists.
PS C:\Users\School\Documents\Azure 2024\python-webapp demo> git remote add origin https://github.com/Eunice0107/python-azure-demo.g
PS C:\Users\School\Documents\Azure 2024\python-webapp demo> git branch -M main
PS C:\Users\School\Documents\Azure 2024\python-webapp demo> git push -u origin main
```



```
PS C:\Users\School\Documents\Azure 2024\python-webapp demo> git remote add origin https://github.com/Eunice0107/python-azure-demo.git
PS C:\Users\School\Documents\Azure 2024\python-webapp demo> git branch -M main
PS C:\Users\School\Documents\Azure 2024\python-webapp demo> git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (8/8), 883 bytes | 55.00 KiB/s, done.
Total 8 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Eunice0107/python-azure-demo.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\School\Documents\Azure 2024\python-webapp demo>
```



Step 4 - Deploying to Azure

Azure was linked to the GitHub repository for automatic deployment.

Once the code was pushed, Azure built and deployed the app.

Screenshots:

14. Azure build and deployment process.

15. Final confirmation - “Hi Python Developer, your app is up and running” in the browser.

