



FIRST SEMESTER SESSION 2023/2024 (A231)
SKIG3013 MOBILE WEB PROGRAMMING

Assignment 2

Mode: Group

Marks: 23 – Refer to rubric below

Submission: Hardcopy & Softcopy (through Online Learning)

Exploring the Synergy: Deploying Python Flask Web App from Google Colab to GitHub

Introduction: The intersection of Google Colab and GitHub offers a unique platform for experimenting with Python Flask web applications and seamlessly deploying them. This case study documents the creation of a simple Flask web app in Google Colab and its subsequent deployment to GitHub. The objective is to illustrate the ease of prototyping and sharing web applications using these two platforms.

Project Scope: Create a basic Flask web app in Google Colab, deploy the app to GitHub, and explore the potential challenges and benefits of this workflow.

Methodology:

1. Setting Up the Flask App in Google Colab:

- Use the Colab environment to write and execute a basic Flask web app.
- Leverage the **flask-ngrok** package to expose the local development server to the internet.

```
python Copy code

!pip install flask-ngrok

from flask import Flask
from flask_ngrok import run_with_ngrok

app = Flask(__name__)
run_with_ngrok(app)

@app.route('/')
def home():
    return 'Hello, this is your Flask web application running in Google Colab'

if __name__ == '__main__':
    app.run()
```

2. Testing the App in Colab:

- Execute the Colab cell to run the Flask app.

- Access the provided ngrok link to ensure the app is working as expected.

3. Setting Up GitHub Repository:

- Create a new GitHub repository to host the Flask app.
- Copy the Flask app code to a new Python file (e.g., **app.py**) in the local development environment.
- Initialize a new Git repository and link it to the GitHub repository.

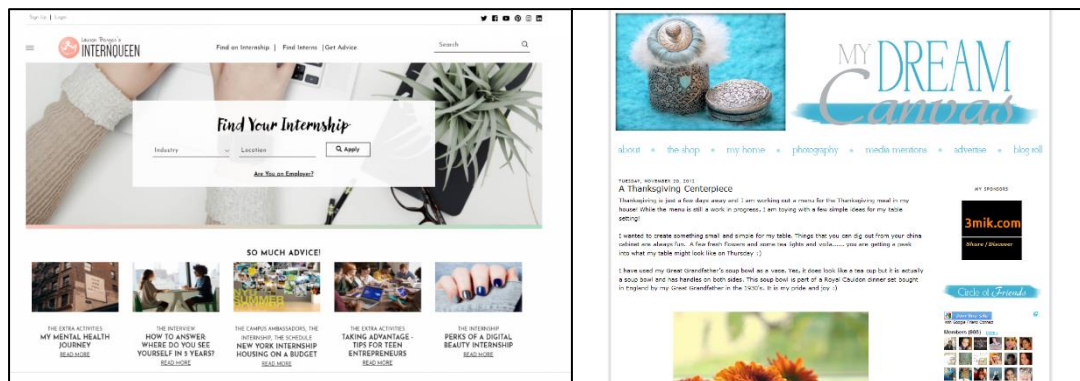
```
bash

git init
git add .
git commit -m "Initial commit"
git remote add origin your_github_repository_url
git branch -M main
git push -u origin main
```

4. Contents of Flask web app (includes adding Images, Routing, Input-Process-Output, Variables, Model-View-Controller component, and Templates Inheritance concepts):

- 1) Landing page
- 2) Main page
- 3) Hobby page
- 4) Portfolio page
- 5) Help or Contact page

* Use your imagination to fill the necessary information for this web site. Check these examples:



5. Continuous Integration with GitHub Actions:

- Create a GitHub Actions workflow (e.g., **.github/workflows/main.yml**) for continuous integration.
- Set up Python, install dependencies, and run tests (if applicable).

```
name: CI

on:
  push:
    branches:
      - main

jobs:
  build:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout repository
        uses: actions/checkout@v2

      - name: Set up Python
        uses: actions/setup-python@v2
        with:
          python-version: 3.8

      - name: Install dependencies
        run: |
          python -m pip install --upgrade pip
          pip install -r requirements.txt

      - name: Run tests
        run: |
          # Add tests if needed

      - name: Deploy to production
        run: |
          # Add deployment commands if needed
```

6. Deployment to GitHub Pages (Optional):

- Modify the GitHub Actions workflow to include deployment steps to GitHub Pages.
- Note that GitHub Pages primarily supports static websites, so consider using other services (e.g., Heroku) for dynamic Flask apps.

Outcome: The project demonstrates the seamless integration of Google Colab and GitHub for creating, testing, and deploying a Flask web application. The ngrok utility facilitates testing within Colab, while GitHub provides a version-controlled environment for collaborative development and continuous integration.

Conclusion: The synergy between Google Colab and GitHub presents a convenient workflow for web application development. While the limitations of Colab for long-term hosting are evident, the integration with GitHub allows for code sharing, collaboration, and integration with powerful CI/CD workflows. This case study highlights the potential of these platforms for rapid prototyping and collaborative development in the Python Flask ecosystem.

Rubrics of evaluation (FYI)

Table 2. Practical-skills performance rubric.

CLO,MQF	Task	Excellent (5)	Good (4)	Satisfactory (2-3)	Poor (0-1)	SCORE
CLO2, LOC3a	Content	Content is presented with excellent flow.	Content is presented with good flow.	Content is presented with some flow.	Content presented is hard to follow.	
	Overall Coding	Working with no error; neatly organized.	Working with few minor errors; somewhat neatly organized.	Working with major errors; code is readable.	Not working at all; code is still readable.	
	Process flow	Presented a clear description of its process; clear on how it is being done.	Presented the process; somewhat clear on how it is being done.	Process needs modification; unsure on how it is being done.	Not presented; little or no discussion.	
	Implementation	Implementation are presented and illustrated well; number of implementations is suitable.	Implementation are presented and illustrated but requires improvement; number of implementations is suitable.	Implementation are presented and illustrated but requires major modification; number of implementations should be improved.	Implementation is not presented.	
	Output	Output is correct.	Some output is incorrect; minor errors.	Most output is incorrect; major errors.	Output is wrong or no output.	
TOTAL						/25
% (TOTAL/25*23)						%