



Deployment on Heroko

Name: Abishek James

Batch Code: LISUM19

Date: 27 March 2023

Submitted to: Data Glacier

Submitted link: <https://github.com/abishekjames/Data-Glacier-intern-week4>

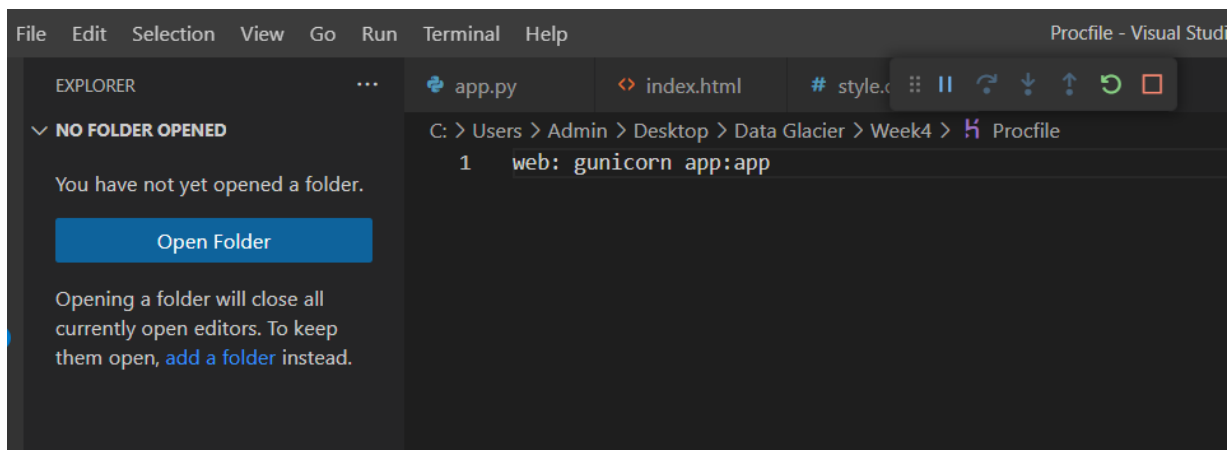
Deployment on Heroku

1) For deploying any model, the very important file is **requirements.txt**. In this all the libraries and dependencies are mentioned. Be careful while creating this file. Use below line to freeze(create) your requirement file in the end.

```
pip freeze > requirements.txt
```

2) Create **Procfile** for Heroku

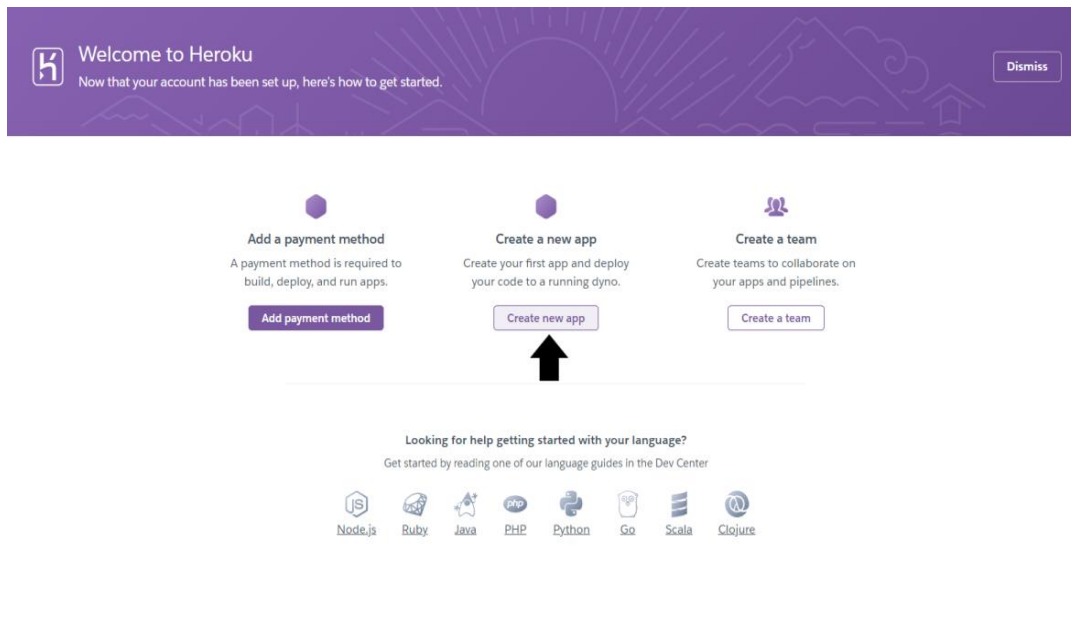
Write the name of the **app** according to the name given in **app.py**



3) Upload files to the GitHub repository.

4) To deploy the model on Heroku, create an account on **Heroku**. Please use python as programming language while setting up the account.

5) After that sign up and then **create a new app**



6) Enter the app name and choose the region

Create New App

App name

car-price-prediction-

car-price-prediction- is available

Choose a region

United States

Add to pipeline...

Create app Cancel



7) Connect your GitHub repository to Heroku.


The screenshot shows the Heroku dashboard for the application 'car-price-prediction-app'. The 'Deploy' tab is selected, showing options to add the app to a pipeline or a stage. Below this, the 'Deployment method' section shows three options: Heroku Git, GitHub (which is 'Connected' with a green checkmark), and Container Registry. The GitHub connection is highlighted with a yellow banner.


8) Deploy your branch

The screenshot shows the 'Automatic deploys' section of the Heroku dashboard. It includes a notification about changing the main deployment branch from 'master' to 'main'. Below this, there's a section to 'Enable automatic deploys from GitHub' with a dropdown menu set to 'main' and a checkbox for 'Wait for CI to pass before deploy'. At the bottom, the 'Manual deploy' section is visible, showing a dropdown menu set to 'main' and a 'Deploy Branch' button, which is highlighted by a large black arrow.

9)Deployment is done to Heroku and you will get the **URL**.

 Personal >  car-price-prediction-app

 abishekjames/Data-Glacier-intern-week4

 Open app More

Overview Resources Deploy Metrics Activity Access Settings

Activity Feed > Build Log ID 3f88be82-963f-4aff-81d6-3be5c409f595

```
Building wheel for jsonify (setup.py): started
Building wheel for jsonify (setup.py): finished with status 'done'
Created wheel for jsonify: filename=jsonify-0.5-py3-none-any.whl size=1538 sha256=8ce7ffe046d590263c9dae0243e8aeae6b1d03893bc374d53dfe71c24e308154
Stored in directory: /tmp/pip-ephem-wheel-cache_ro7e3jn/wheels/8b/0b/70/cd8a2f72ec6e8dbab2d7fffe3e8a545f4d152255cc7e8541f5
Building wheel for MarkupSafe (setup.py): started
Building wheel for MarkupSafe (setup.py): finished with status 'done'
Created wheel for MarkupSafe: filename=MarkupSafe-1.1.1-cp311-cp311-linux_x86_64.whl size=17169 sha256=890711d84baf57dd158a9e760a43ac39df98ffedc4cb33d8cb529a26a04494c2
Stored in directory: /tmp/pip-ephem-wheel-cache_ro7e3jn/wheels/42/2c/24/ee2a56da668805256e7c1b1d51ae9b177360788cabcd5b9c1
Successfully built jsonify MarkupSafe
Installing collected packages: wincertstore, jsonify, chardet, certifi, Werkzeug, urllib3, threadpoolctl, numpy, MarkupSafe, joblib, itsdangerous, idna, gunicorn, click,
scipy, requests, Jinja2, scikit-learn, Flask
Successfully installed Flask-1.1.2 Jinja2-2.11.2 MarkupSafe-1.1.1 Werkzeug-1.0.1 certifi-2020.6.20 chardet-3.0.4 click-7.1.2 gunicorn-20.1.0 idna-2.10 itsdangerous-1.1.0
joblib-1.2.0 jsonify-0.5 numpy-1.24.2 requests-2.24.0 scikit-learn-1.2.2 scipy-1.10.1 threadpoolctl-2.1.0 urllib3-1.25.9 wincertstore-0.2
-----> Discovering process types
Procfile declares types -> (none)
-----> Compressing...
Done: 104M
-----> Launching...
Released v4
https://car-price-prediction-app.herokuapp.com/ deployed to Heroku
```

Build finished

<https://car-price-prediction-app.herokuapp.com/>