Dash App

How to run

On your terminal, create a new python environment.

```
conda create -n health_dss_app python=3.8
conda activate health_dss_app
```

On your terminal, install all package.

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

On your terminal, run the dash application via Bash.

```
sh run.sh
```

For development mode (with reload), run

```
sh run.sh development
```

How to dockerize

Make sure you have DockerHub account.

Make sure you have Docker Desktop installed locally. Open Docker Desktop and sign in with your DockerHub to enable Docker daemon to run on the background.

On your terminal, create a Docker image.

```
docker build -t your-dockerhub-username/project-name .
```

On your terminal, test run your built Docker image.

```
docker run -p 5000:8050 your-dockerhub-username/project-name
```

How to deploy

Deployment on GCP

1. Make sure you have a GCP account with some credit amount.
2. Go to Google Cloud Console > Create a New Project > Activate Cloud Shell.
3. Create a new project.
4. On the terminal, type `python3 -m venv .venv` to create a new Python environment.
5. On the terminal, type `. .venv/bin/activate` to activate your newly created environment.
6. Open Editor, add code for app.py, requirements.txt, Dockerfile.
7. On the terminal, install package via `pip install -r requirements.txt`.
8. On the terminal, enable services through GCP terminal: `gcloud services enable containerregistry.googleapis.com`.
9. On the terminal, enable permission via `gcloud auth configure-docker`.
10. On the terminal, build your docker image via `docker build -f Dockerfile -t gcr.io/your-project/test:test .`.
11. On the terminal, push your docker image via `docker push gcr.io/your-project/test:test`.
12. On the terminal, deploy your application via `gcloud run deploy dashapp --image gcr.io/your-project/test:test`.
13. Your application has been successfully deployed!

Deployment on Virginia Tech's Kubernetes Rancher

1. Consider reading through the Documentation - Cloud Quickstart.
2. Make sure you have DockerHub account and access to Virginia Tech's Kubernetes Rancher (cloud.cs.vt.edu).
3. Make sure you have Docker Desktop installed locally. Open Docker Desktop and sign in with your DockerHub to enable Docker daemon to run on the background.
4. On the terminal, build your docker image via `docker build -t your-dockerhub-username/project-name .`.
5. On the terminal, push your docker image via `docker push your-dockerhub-username/project-name`.
6. After granted access to cloud.cs.vt.edu via asking admin, create your Workload with the port that you set on Dockerfile. The name of the image for the server to pull should also be filled in here (such as your-dockerhub-username/project-name). Here, you can also define your domain name.
7. Now, create the ingress, and select your workload. Activate 1 pod to initialize and run your application.
8. Your application has been successfully deployed!

Feature TODOs

- Information dissemination and announcement / advisory creator with templates
- Public interface for public users
- Ability to call third-party apps (gmail, docs, editors, etc)
- Language support