Prototype Guide

Details about the Files and Folders:-

- **Cases :** This folder contains the appropriately named cases pdf with all the relevant documents, i.e. general details, interim orders and judgment.

- **History:** Contains the generative model chat environment files that needed to be loaded for the interactive chat.

- **Templates:** contains the html file for the flask.

- **admin.py:** this is the python code that is run in the local developer side machine, whenever he wants to update the database and create new chat environment for whenever new cases is raised in the High Court of Kerala.

- **App.py:** contains the main logic of the program along with flask infrastructure.

- **Case_update.pkl:** it is a case updatation package that can be uploaded in the server and used to update the chromaDB with the relevant new cases.

- **Chroma.sqlite3:** local persistent chromaDB running in cloud server environment.

- **Update_db.py:** This is the code to use the updatation package and update it on the cloud.

Steps to deploy and run:

Upload the relevant files into the GitHub.

Go to your EC2 instance(ensure minimum 4GB of RAM to be able to install ChromaDB).

Connect, setup the env, install all the packages and pull the repo from the git.

Use the following commands(also for updation):

Commands to Update

Kill the existing deployment server process(daemon):
```
pkill -f gunicorn
```
Go to the directory :
```
cd flask_test/
```
Pull the changes from the repo:
```
git pull origin
```
Go back to home directory:
```
cd ..
```
Copy files from local repo to home page for hosting :
```
cp -r flask_test/* .
```
Activate the Environment:
```
source venv/bin/activate
```
Run the database updation Script:
```
python update_db.py
```
Restart the deployment server
```
nohup gunicorn -w 5 -b 0.0.0.0:8000 app:app &
```

NOTE: Whenever new cases needs to be updated, in developer side with the relevant new documents run the admin.py, push the changes to the git and then run the above commands again.