

Project Setup Documentation

Prerequisites

Before you begin, ensure you have the following installed on your local machine:

- Python 3.x
- pip (Python package installer)
- Git
- make sure to run sqllite-setup.py prior to this

Setting Up the Virtual Environment

It's recommended to use a virtual environment for Python projects to manage dependencies effectively. To set up a virtual environment:

1. Navigate to the project directory:

```
cd code
```

2. Create a virtual environment named `venv` (you can name it something else, but ensure it's included in the `.gitignore` to avoid pushing it to the repository):

```
python -m venv venv
```

3. Activate the virtual environment:

- On Windows:

```
.\venv\Scripts\activate
```

- On macOS and Linux:

```
source venv/bin/activate
```

Installing Dependencies

After activating the virtual environment, install the project dependencies:

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

This command reads the `requirements.txt` file from the project directory and installs all the necessary packages.

Setting Up the Database

If there is any database setup needed (like running a script to initialize tables), include those steps here. For example:

1. Ensure SQLite is installed on your machine (comes pre-installed with Python).
2. Navigate to the project root directory and run the following command:

```
'''  
python sql_setup.py  
'''
```

Running the Flask Application

1. Ensure your virtual environment is activated.
2. Set the environment variable for the Flask application:

- On Windows:

```
'''  
set FLASK_APP=app.py  
'''
```

- On macOS and Linux:

```
'''  
export FLASK_APP=app.py  
'''
```

3. Run the Flask development server:

```
'''  
flask run  
'''
```

4. Open a web browser and navigate to <http://127.0.0.1:5000/> to view the application.
5. To view the visualizations you would open <http://127.0.0.1:5000/data-visualization>

Notes

- Always activate the virtual environment (`venv`) before working on the project locally.
- After pulling new changes from the repository, run `pip install -r requirements.txt` to ensure your local environment matches the project's dependencies.
- Do not push changes to the `venv` directory, `hurriscan.db`, or any other files that contain sensitive information or personal configurations.
- Before committing changes to Git, ensure no sensitive information is included, and only necessary files are added.

Troubleshooting

Include common issues and their solutions here. For example:

- If you encounter errors while installing dependencies, try updating pip: `pip install --upgrade pip` and then rerun `pip install -r requirements.txt`.
- If the Flask app doesn't run, ensure the `FLASK_APP` environment variable is set correctly and points to `app.py`.