Developer Setup Instructions

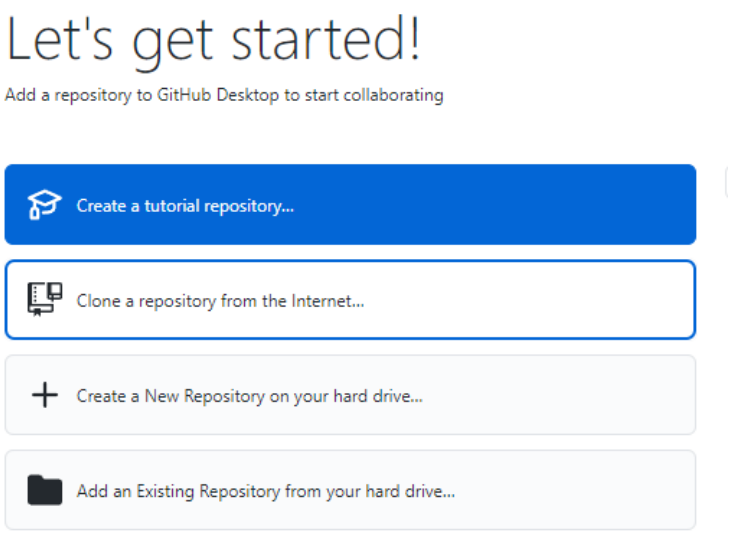
Windows Setup

It is recommended to install VSCode as your code editor and debugger.

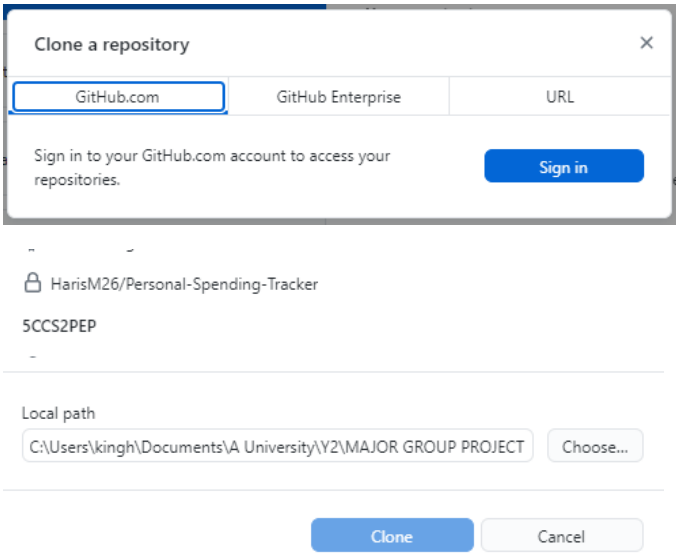
Also, install Github Desktop as it is easy to manage your repo using this app.

Clone Repository:

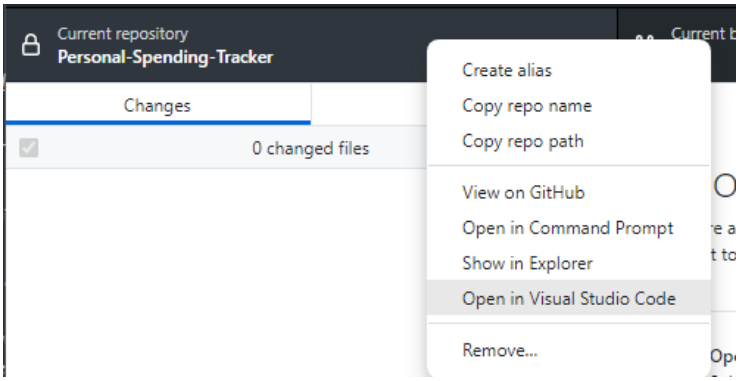
On Github Desktop:



Click “Clone a repository from the internet…” and sign into your github or github enterprise account, whichever is being used on the project.



After signing in, find the Personal-Spending-Tracker repo, choose a path of your choice, then click clone.



Right click Personal-Spending-Tracker and click “Open in Visual Studio Code”. VSCode will open.

Click on “Terminal -> New Terminal” at the top, a terminal will open.

In the terminal, type and run:

- **py -m venv void**
- **void\Scripts\activate.ps1**

Virtual environment is now activated.

Go to View -> Command Palette -> Click Python: Select Interpreter -> Click Python 3.10.6 (‘void’:venv)

Now type:

- **pip install -r requirements.txt**

Every time you fetch from main, execute the above command and always update requirements every time you install a package by typing:

- **pip freeze > requirements.txt**

You can check your py version using “py –version”.

API Keys:

- Visit NewsAPI
- Sign up to obtain a key
- Create a file in your root folder (where manage.py is located) and name it ‘.env’
- In your .env file, type: NEWS_API_KEY = your_key_here (no quotes)

Migrate the database, type:

- **py manage.py makemigrations**
- **py manage.py migrate**

Seed the development database with:

- **py manage.py seed**

Run all tests with:

- **py manage.py test**

You can create a superuser using:

- **py manage.py createsuperuser**

It will ask for an email and password. You can then login to the app or admin using this login.

Before running server, check app.py file and uncomment the code below to start scheduler:

scheduler.start()

Run the server:

- **py manage.py runserver**

➢

Linux Setup

Firstly, you need to add your ssh key to your github if you haven't already.

Then, navigate to the folder you would like to save the files.

Type:

$ git clone SSH/HTTP

NB: paste the ssh/http you copied from GitHub in the above instead of "SSH/HTTP"

Installing packages:

Do this every time you fetch from main, type:

$ pip3 install -r requirements.txt

Updating requirements:

Every time you install a package, update the requirements, type:

$ pip3 freeze > requirements.txt

Virtual Environment:

Navigate to root folder, which is "Personal-Spending-Tracker", type:

$ virtualenv foldername

NB: Add the name of your virtual environment folder (usually named 'venv' but you can choose whatever you want)

Activate virtual environment, type:

$ source foldername/bin/activate

Your virtual environment is now activated and you can run the commands:

- **python3 manage.py makemigrations**
- **python3 manage.py migrate (migrate to database)**
- **python3 manage.py createsuperuser (Create an admin user)**
- **python3 manage.py seed**
- **python3 manage.py test (Run your test cases)**
- **python3 manage.py runserver**

Where you want to debug on the command line while server is running, at a specific line, type:

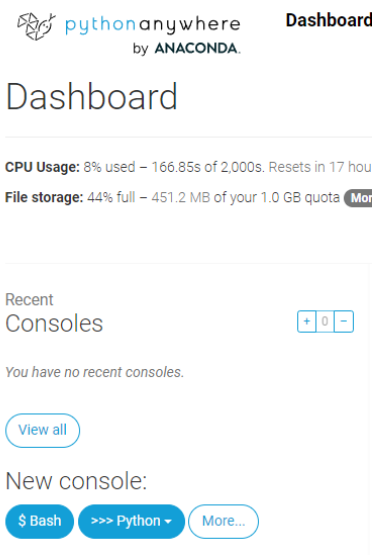
Import pdb; pdb.set_trace()

The app will stop running at that line and you can view local variables using locals() and global variables using globals(). You can run commands like SpendingTransaction.objects.all(), etc. To see the value of a variable, use “p variable”.

Deployment

To deploy you need to sign up to PythonAnywhere. You will obtain a free account with limited CPU usage and file storage.

After you’ve logged in and are on the Dashboard page, click “Bash” under “New console:”



In the bash console, type:

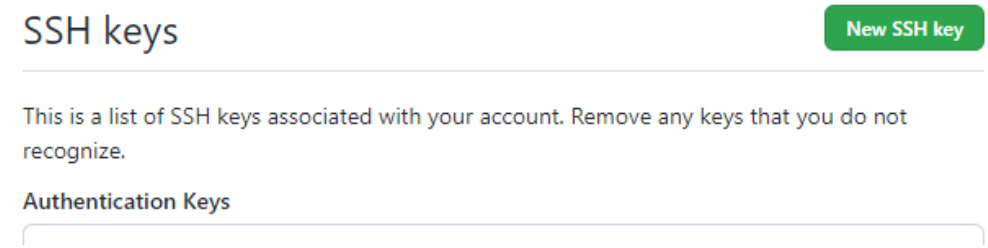
~$ `ssh-keygen -t rsa`

Press enter when asked which file to save the key.

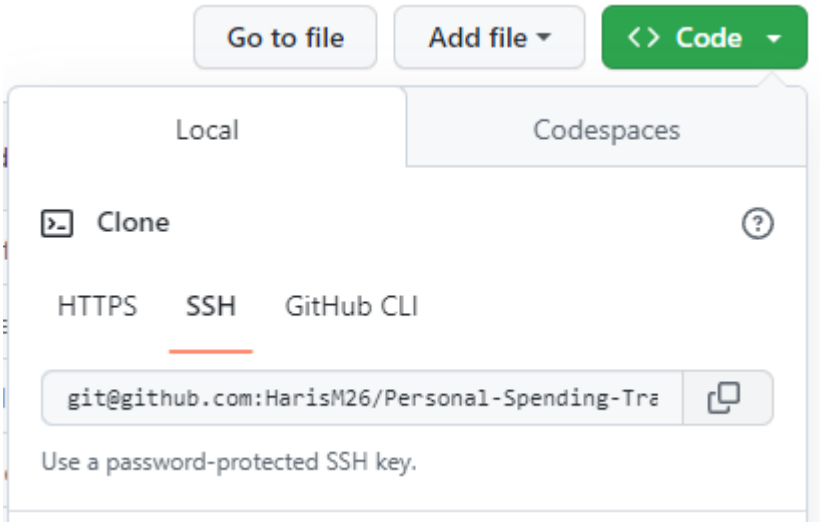
Enter a passphrase and confirm it or leave empty and press enter twice.

Then type: ~$ `cat ~/.ssh/id_rsa.pub`

This will output the public key to the console. Copy it and add it to your github ssh keys in settings on github.



Go to ssh as shown below and click to copy.



Go back to bash console and type:

~$ git clone git@github.com:HarisM26/Personal-Spending-Tracker.git

If it asks to enter a passphrase, enter the passphrase you created.

After the git repo has been cloned, type:

~$ `virtualenv venv --python="/usr/bin/python3.10"`

~$ `workon venv`

Virtual Environment is now loaded.

You need to install all packages, so type:

~$ `pip install -r requirements.txt`

Now you need to migrate:

~$ `python3 manage.py migrate`

Now once migration has finished, go to “Web” and add a new app.

Go next -> Click Manual Configuration -> Click Python 3.10- > Click next -> Back to Web.

On “Web”, scroll down until you see “Virtualenv”. Enter “venv” and click the tick button.

Under “Code”, click the WSGI configuration file.

```
# ++++++++++ DJANGO ++++++++++
# To use your own django app use code like this:
import os
import sys
#
## assuming your django settings file is at '/home/your_username/mysite/mysite/settings.py'
## and your manage.py is is at '/home/your_username/mysite/manage.py'
path = '/home/your_username/Personal-Spending-Tracker'
if path not in sys.path:
    sys.path.append(path)
#
os.environ['DJANGO_SETTINGS_MODULE'] = 'personal-spending-tracker.settings'
#
## then:
from django.core.wsgi import get_wsgi_application
application = get_wsgi_application()
```

Remove everything other than the DJANGO part of the file. The file should look as shown above.

Save and Click the Reload button.

Go back to “Web” and under “Static files” type:

URL: /static/

Directory: /home/your_username/Personal-Spending-Tracker/static

Click reload again and then open the link at the top of the page to open the web app.