# House Rent Estimator-end to end ML project

After completing 3 machine learning projects, I wanted to do an end to end project to really understand the ML workflow outside the professional setup by myself. I wanted to make a project which would be relevant to our use. Keeping in mind all the articles that I have read since I started to learn ML and my initial web research, I decided to create a House Rent Estimator app.

## Getting Data

I was aware that finding an appropriate dataset for my project will be a challenge. I went on to find a dataset that had data on house rent of five cities of Brazil from Kaggle. I considered learning scraping data from web when I did not get any other producible dataset which will be relevant in the Indian setup. Again, with my research, I was not surprised to know that data scraping has its own world and will take me time to master. So, I decided to convert the dataset from Brazil into an Indian version.

The challenge in converting the dataset to an Indian version was to understand the conversion rate that I should use to convert Brazilian currency (Brazilian Real=₹14.xx) into an acceptable rent price that would be a fit in the Indian scenario. After some hit and trial and analysis of the price I converted the rent to ₹ which seemed to be okay.

The next decision to make was to choose the cities to replace the five Brazilian cities. I naturally went forward to choose the metro cities: Delhi, Mumbai, Kolkata, Chennai and Bangalore. I looked at the rent distributions and made a decision to replace each city. With the data I had, I was really skeptical to keep Mumbai as a city. The rent amounts did not even closely resonate with the real market values. For the time being, I decided to focus on modeling with the data on hand.

## Modeling

Unlike the other projects that I completed before this project, I was certain that for this project the main focus will be on deployment. I had not worked on deployment in any prior project in ML space before this project, so this was a natural decision. So, I went on with a simple Random Forest to make my model.

The new aspect after making the model was to convert it into a form that can be accessed by a web app to take inputs from a user and return a value. The random forest model that was developed with the data was converted into a pickle form using python module 'pickle'. *"Pickling"* is the process whereby a Python object hierarchy is converted into a byte stream". More details about pickling can be found from the official document from Python: [https://docs.python.org/3/library/pickle.html](https://docs.python.org/3/library/pickle.html)

This video (https://www.youtube.com/watch?v=2Tw39kZIbhs) by **sentdex** also well explains the concept of pickling.

## Creating Web App with Streamlit

In the web we find most of the available project apps are made by 'Flask' (a lightweight WSGI web application frameworkin Python) and deployed in 'Heroku' and few in 'AWS'. There are a lot of articles which can be found on the internet about the advantages of using Flask over Django (a fully featured Python web framework that can be used to build complex web applications). I also happened to look into some of the apps that were being made by ML rookies since I started my ML journey. I knew that I will learn Flask to make a web app when I will make this project. But during my research before this project I came across Streamlit. It is an open-source app framework for Machine Learning and Data Science teams to create data apps. It was launched in 2019. Here is an article by the cofounder, Adrien Treuille about Streamlit (**https://medium.com/streamlit/announcing-streamlits-21m-series-a-ae05daa6c885**). I read the documentations and visited the gallery of Streamlit website (https://www.streamlit.io/gallery). I was attracted by the gallery in the website and also to check the code to create them. They seemed to be simpler than the Flask codes I had previously seen. So, I decided I will go with Streamlit to make my first app.

The tutorials for using Streamlit to make a web app can be found in YouTube. I would suggest anyone to go through the tutorial playlist by JCharisTech & J-Secur1ty (https://www.youtube.com/watch?v=_9WiB2PDO7k&list=PLJ39kWiJXSixyRMcn3lrbv8xI8ZZoYNZU&index=1). I also found a **PDF** file by Streamlit which has all the features and resources to make an app with Streamlit. I must say that it was easy to build the app with Streamlit and did not take me much time as well. I ran and checked the app on local server before deploying it in Heroku.

I noticed that the rent estimates for the city of Mumbai was not doing justice to the Indian market value. I am sorry, but I had to replace Mumbai with Pune to make it work.

## Deployment in Heroku

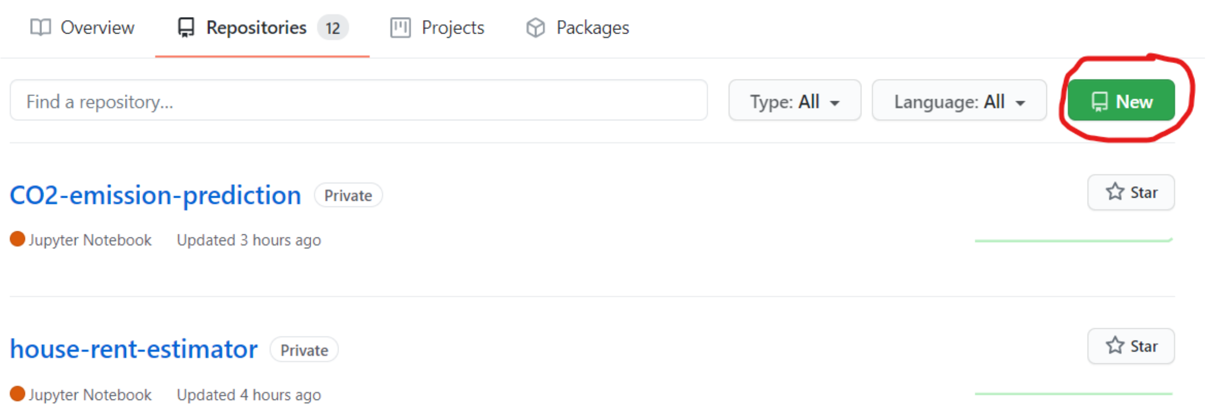
Requirements:

- GitHub account: To store the necessary files for deployment in a repository
- Heroku account: Heroku allows hobbyists to deploy upto 5 apps from their free account.
- Download and install Heroku CLI

```
https://devcenter.heroku.com/articles/heroku-cli
```

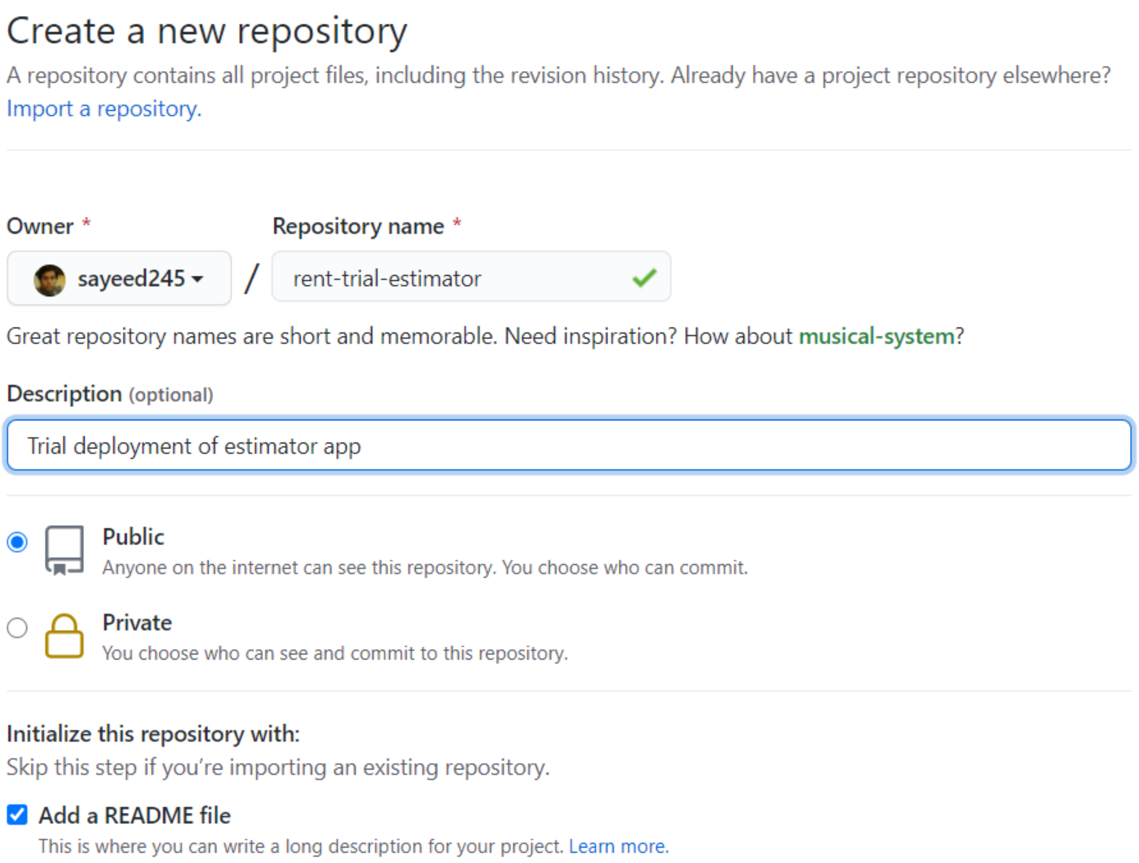
Steps:

- Create a new repository with an appropriate name(rent-trial-estimator) and a README.md file



CO2-emission-prediction Private
Jupyter Notebook Updated 3 hours ago

house-rent-estimator Private
Jupyter Notebook Updated 4 hours ago



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * sayeed245 / Repository name * rent-trial-estimator

Great repository names are short and memorable. Need inspiration? How about musical-system?

Description (optional)

Trial deployment of estimator app

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

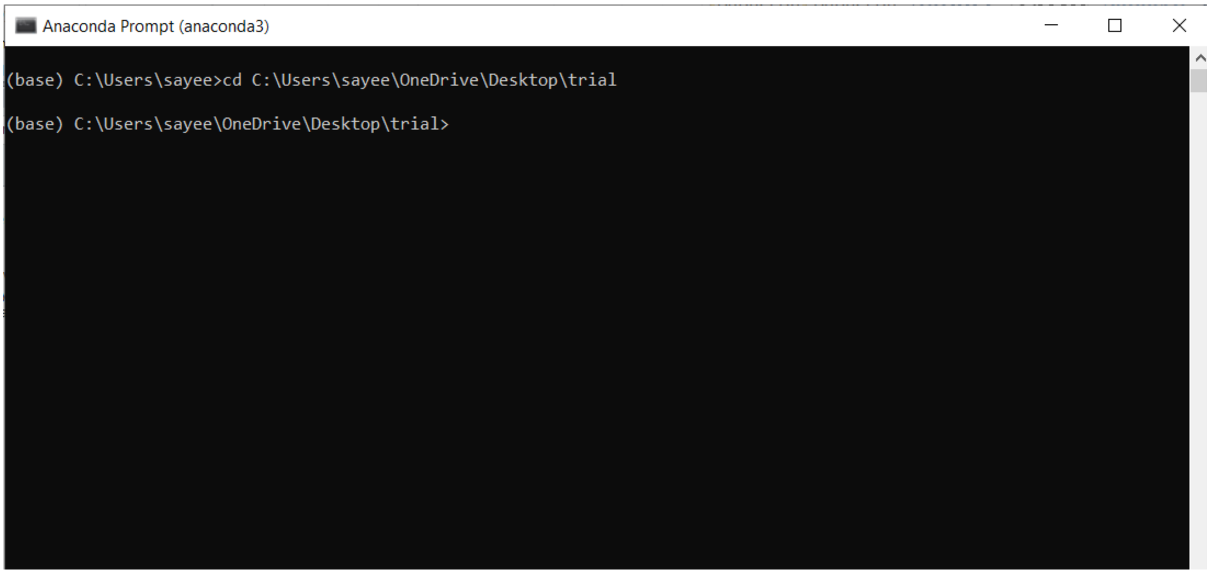
Initialize this repository with:
Skip this step if you're importing an existing repository.

☑ Add a README file
This is where you can write a long description for your project. Learn more.

- Create a folder on your local drive which will contain all the files for deployment
- Copy the <**address>** of this folder
- Open Command Prompt/Anaconda Prompt -> type cd -> Paste the <address>

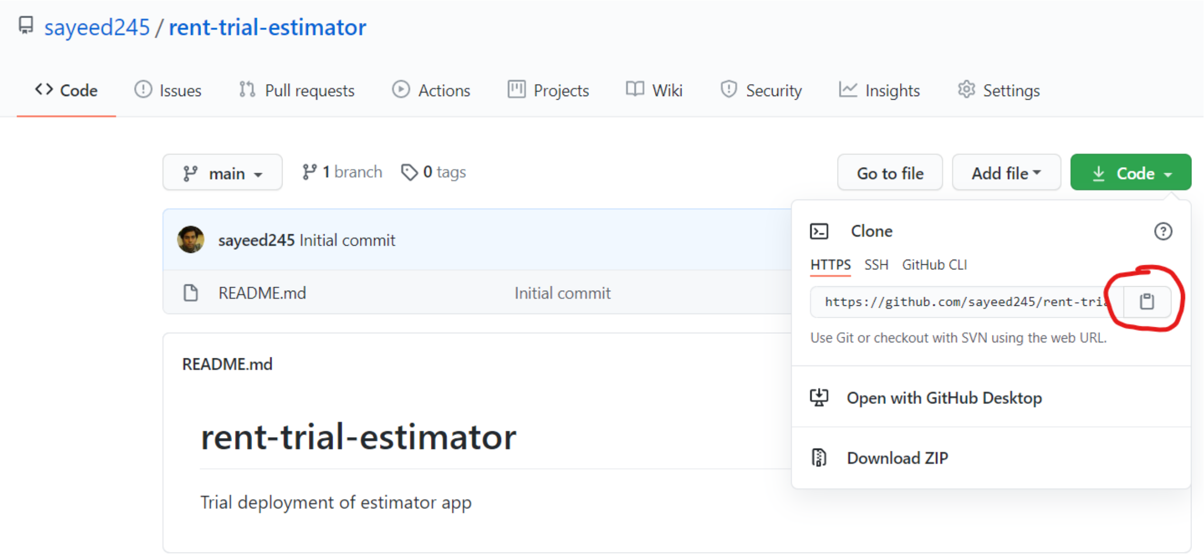
cd <address>

Enter



```
(base) C:\Users\sayee>cd C:\Users\sayee\OneDrive\Desktop\trial

(base) C:\Users\sayee\OneDrive\Desktop\trial>
```

- Go to GitHub and clone the repository. Click on the Code button and copy the link

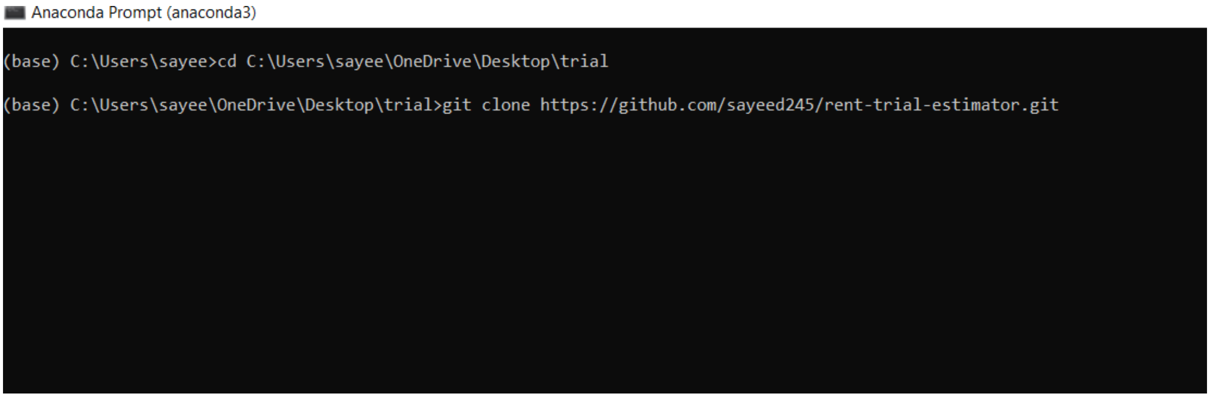


README.md

rent-trial-estimator

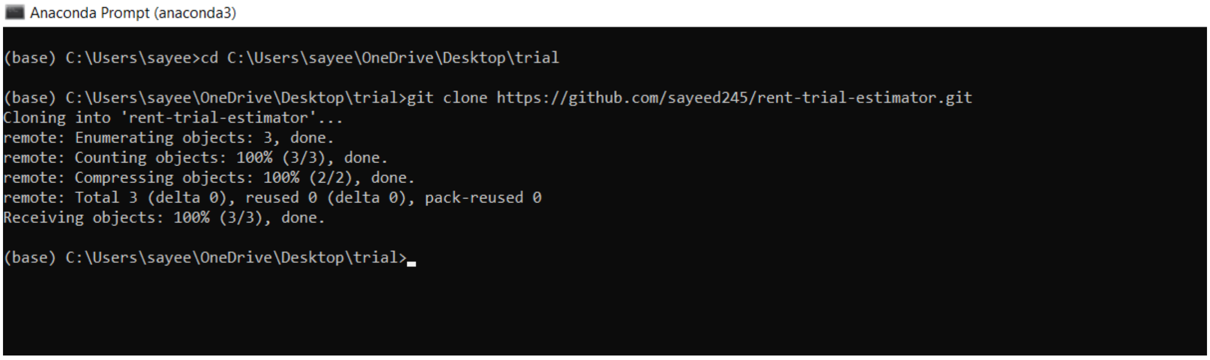
Trial deployment of estimator app

- Go to Command Prompt ->Type:   git clone -> paste the code



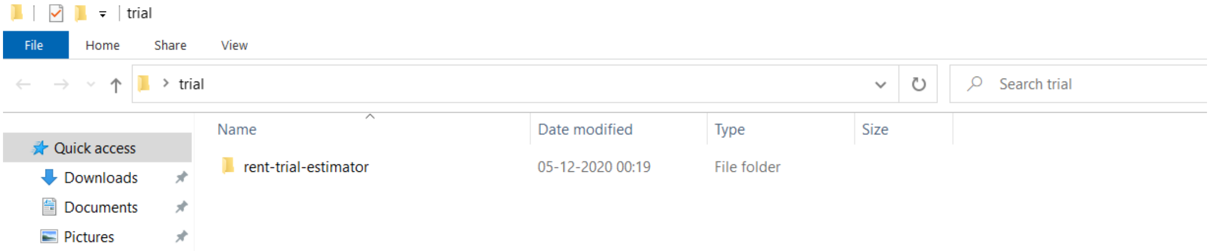
```
(base) C:\Users\sayee>cd C:\Users\sayee\OneDrive\Desktop\trial

(base) C:\Users\sayee\OneDrive\Desktop\trial>git clone https://github.com/sayeed245/rent-trial-estimator.git
```

After pressing ‘Enter’ the following screen will appear



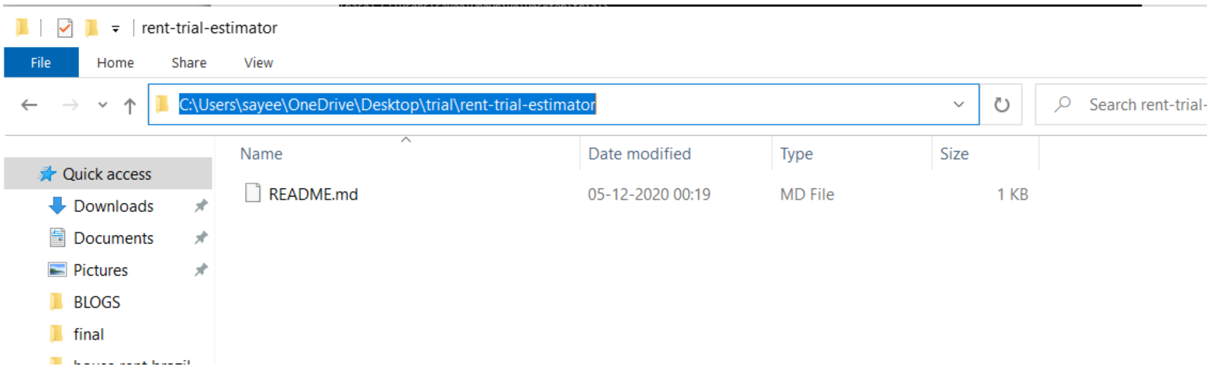
```
(base) C:\Users\sayee>cd C:\Users\sayee\OneDrive\Desktop\trial

(base) C:\Users\sayee\OneDrive\Desktop\trial>git clone https://github.com/sayeed245/rent-trial-estimator.git
Cloning into 'rent-trial-estimator'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

(base) C:\Users\sayee\OneDrive\Desktop\trial>_
```

- We will check the folder that we had made in step 1. We will find a new folder has been made.



| Name | Date modified | Type | Size |
|---|---|---|---|
| rent-trial-estimator | 05-12-2020 00:19 | File folder | |

- We will enter this folder and copy its address. We will find that the README.md file from the GitHub repository has been cloned to this folder



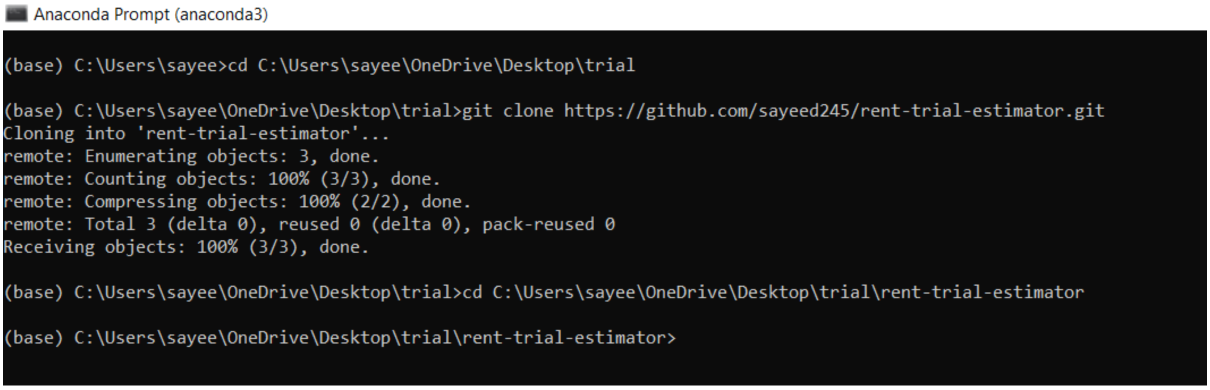
C:\Users\sayee\OneDrive\Desktop\trial\rent-trial-estimator

| Name | Date modified | Type | Size |
|---|---|---|---|
| README.md | 05-12-2020 00:19 | MD File | 1 KB |

- We will go to this folder in the Command Prompt as well.

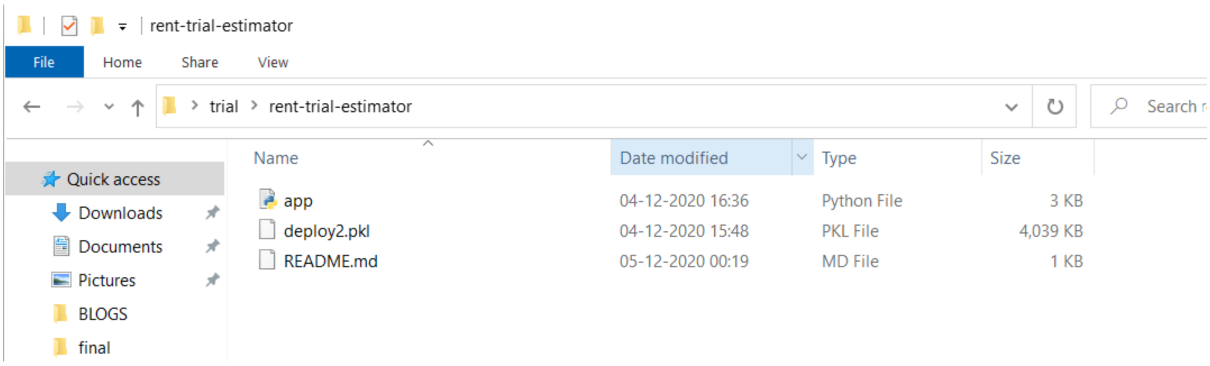
Type: cd -> Paste the address copied in step 8-> Enter

You will find the following:



```
(base) C:\Users\sayee>cd C:\Users\sayee\OneDrive\Desktop\trial

(base) C:\Users\sayee\OneDrive\Desktop\trial>git clone https://github.com/sayeed245/rent-trial-estimator.git
Cloning into 'rent-trial-estimator'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

(base) C:\Users\sayee\OneDrive\Desktop\trial>cd C:\Users\sayee\OneDrive\Desktop\trial\rent-trial-estimator

(base) C:\Users\sayee\OneDrive\Desktop\trial\rent-trial-estimator>
```

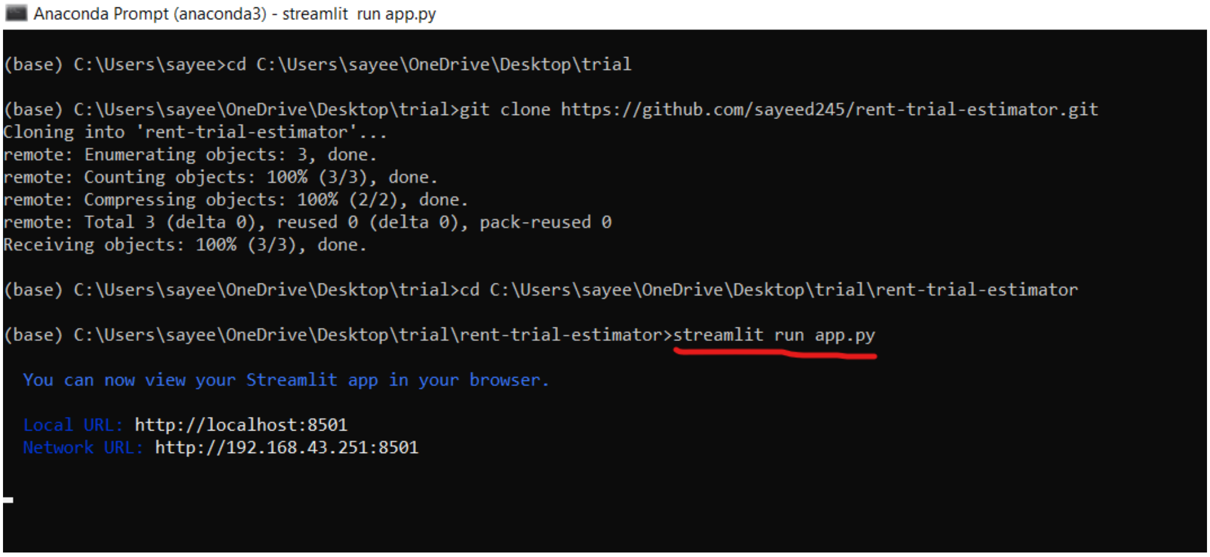
- Now copy the Python web app file that you created with streamlit along with the pickle file containing the model -> Paste them in the folder created in step 8



| Name | Date modified | Type | Size |
|---|---|---|---|
| app | 04-12-2020 16:36 | Python File | 3 KB |
| deploy2.pkl | 04-12-2020 15:48 | PKL File | 4,039 KB |
| README.md | 05-12-2020 00:19 | MD File | 1 KB |

Here, deploy2.pkl is the pickle file that contains my model and was used in the app.py to make the app.

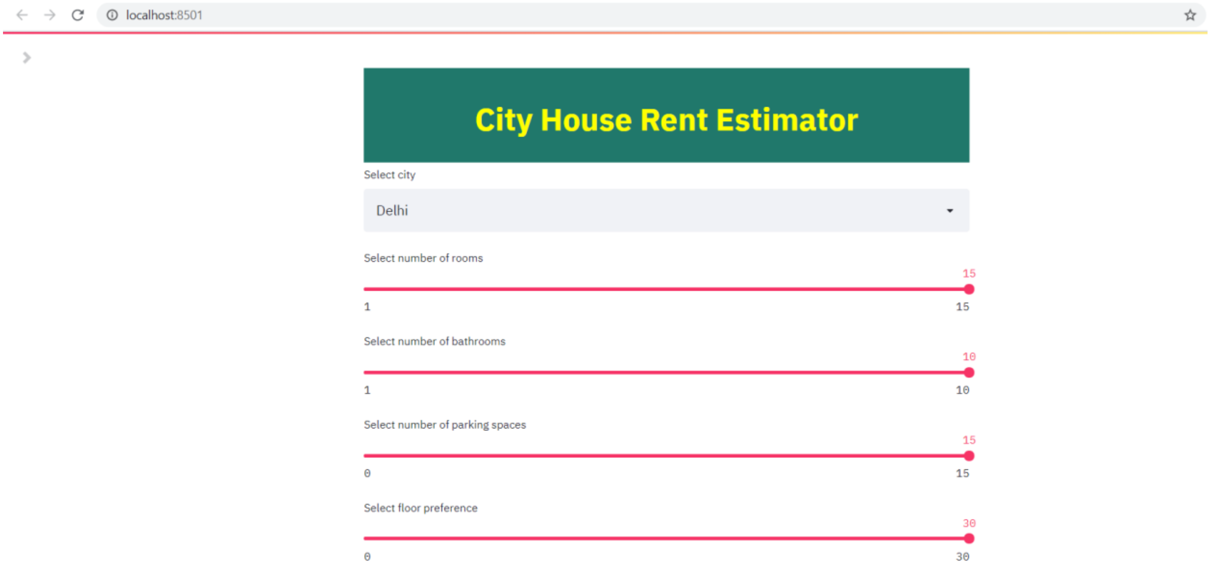
- It is better to name the app file as ‘app.py ‘ as it will be used in the following steps. Make sure whenever you name the files with their format like .py in ‘app’.py ,they do not come as a name ‘app.py’ or look like app.py.py. This should be kept in mind for all the other files that we will be making in the following steps
- You can check once if the app is running locally.
- In Command Prompt -> Type streamlit run app.py ->Enter



Anaconda Prompt (anaconda3) - streamlit  run app.py

```
(base) C:\Users\sayee>cd C:\Users\sayee\OneDrive\Desktop\trial

(base) C:\Users\sayee\OneDrive\Desktop\trial>git clone https://github.com/sayeed245/rent-trial-estimator.git
Cloning into 'rent-trial-estimator'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

(base) C:\Users\sayee\OneDrive\Desktop\trial>cd C:\Users\sayee\OneDrive\Desktop\trial\rent-trial-estimator

(base) C:\Users\sayee\OneDrive\Desktop\trial\rent-trial-estimator>streamlit run app.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://192.168.43.251:8501
```

A new tab with the app will open. We can check if its working which will ensure the appy is in the right folder and is ready to deployed.



localhost:8501

City House Rent Estimator

Select city

Delhi

Select number of rooms

15

1 15

Select number of bathrooms

10

1 10

Select number of parking spaces

15

0 15

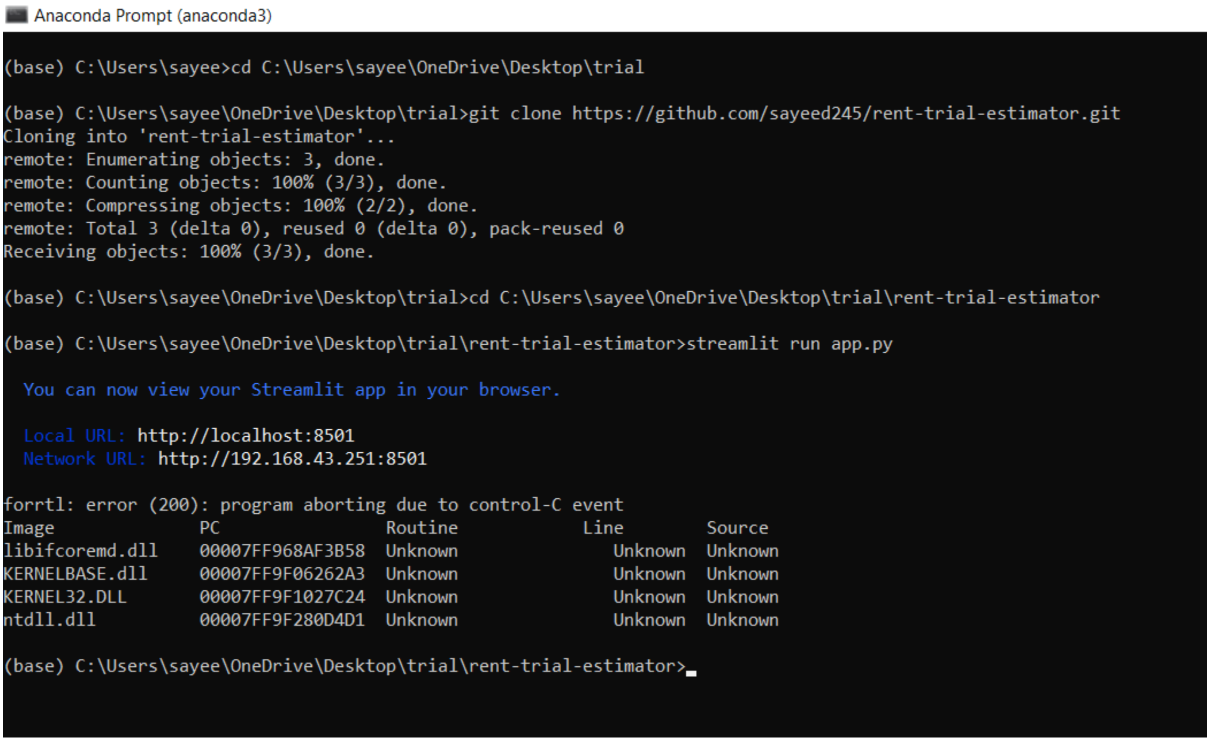
Select floor preference

30

0 30

- It is running alright for me
  In Command Prompt-> Press Cntrl+C ->Enter

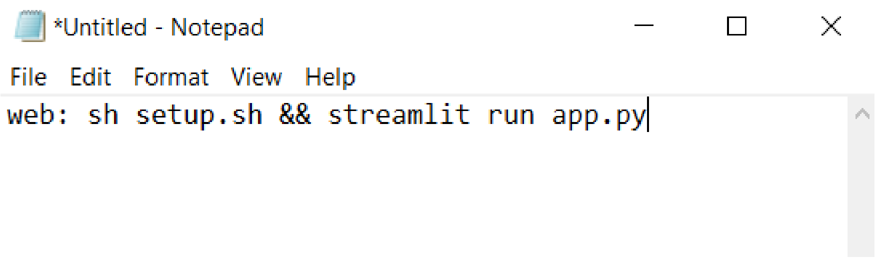
  to come out of the app



```
(base) C:\Users\sayee>cd C:\Users\sayee\OneDrive\Desktop\trial

(base) C:\Users\sayee\OneDrive\Desktop\trial>git clone https://github.com/sayeed245/rent-trial-estimator.git
Cloning into 'rent-trial-estimator'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

(base) C:\Users\sayee\OneDrive\Desktop\trial>cd C:\Users\sayee\OneDrive\Desktop\trial\rent-trial-estimator

(base) C:\Users\sayee\OneDrive\Desktop\trial\rent-trial-estimator>streamlit run app.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://192.168.43.251:8501

forrtl: error (200): program aborting due to control-C event
Image              PC                Routine            Line        Source
libifcoremd.dll    00007FF968AF3B58  Unknown               Unknown  Unknown
KERNELBASE.dll     00007FF9F06262A3  Unknown               Unknown  Unknown
KERNEL32.DLL       00007FF9F1027C24  Unknown               Unknown  Unknown
ntdll.dll          00007FF9F280D4D1  Unknown               Unknown  Unknown

(base) C:\Users\sayee\OneDrive\Desktop\trial\rent-trial-estimator>
```

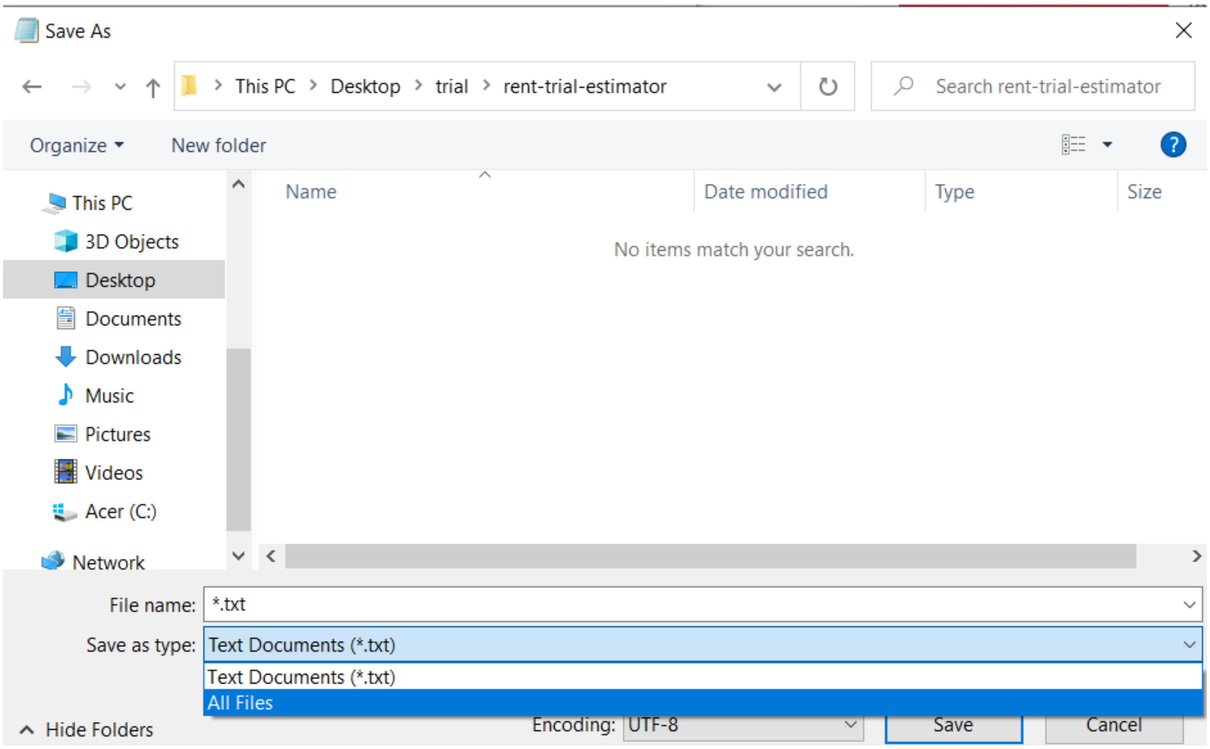
- Create a Procfile (requirement for Heroku deployment. This file contains the command which will run the app in Heroku environment
  Open Notepad
  Paste the following code:

```
web: sh setup.sh && streamlit run app.py
```



```
web: sh setup.sh && streamlit run app.py
```

You will have to save the file in the same folder as created in step 8
Make sure that the 'Save as type' is 'All Files'



No items match your search.

File name: *.txt

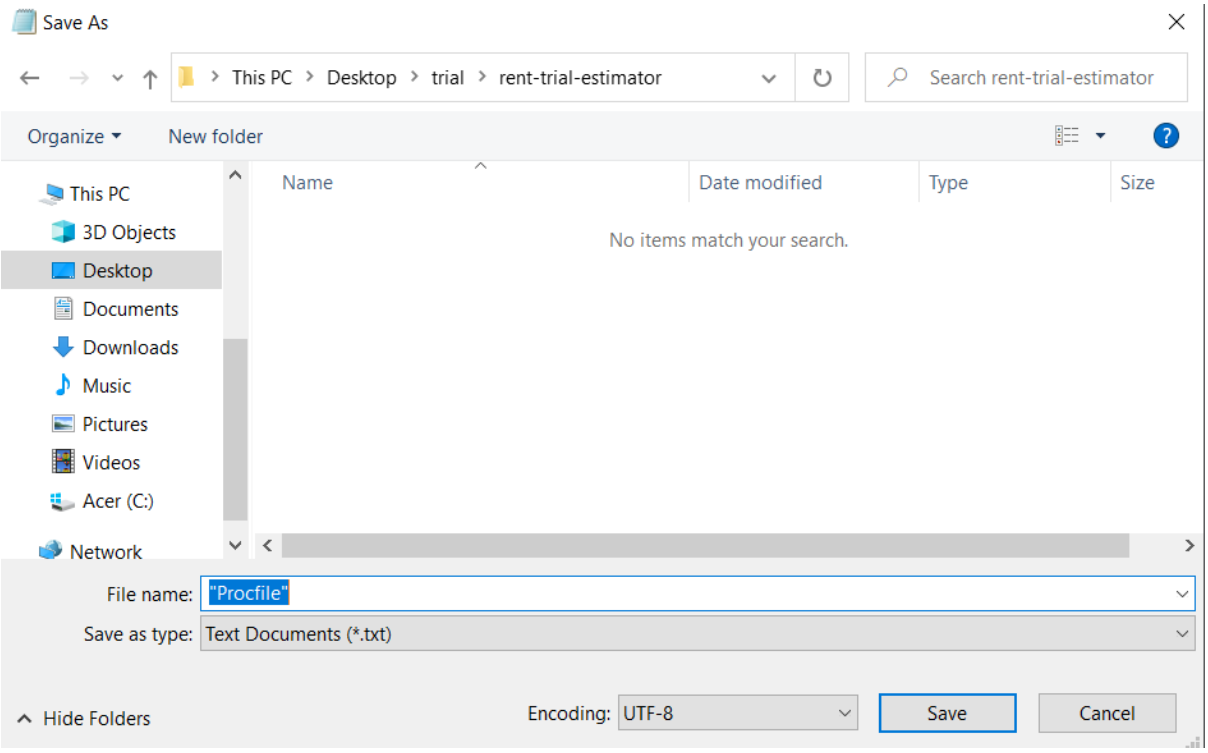
Save as type: Text Documents (*.txt)

Text Documents (*.txt)

All Files

Encoding: UTF-8

Save the name Procfile as “Procfile”. This will ensure that it is saved as a File without any extension.

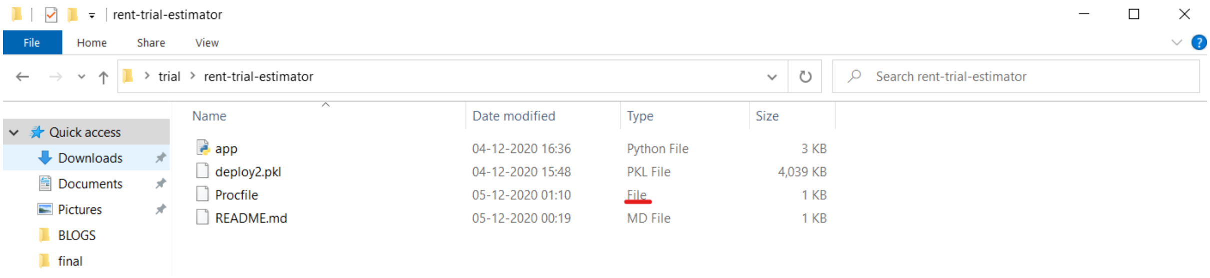


No items match your search.

File name: "Procfile"

Save as type: Text Documents (*.txt)

Encoding: UTF-8



| Name | Date modified | Type | Size |
|---|---|---|---|
| app | 04-12-2020 16:36 | Python File | 3 KB |
| deploy2.pkl | 04-12-2020 15:48 | PKL File | 4,039 KB |
| Procfile | 05-12-2020 01:10 | File | 1 KB |
| README.md | 05-12-2020 00:19 | MD File | 1 KB |

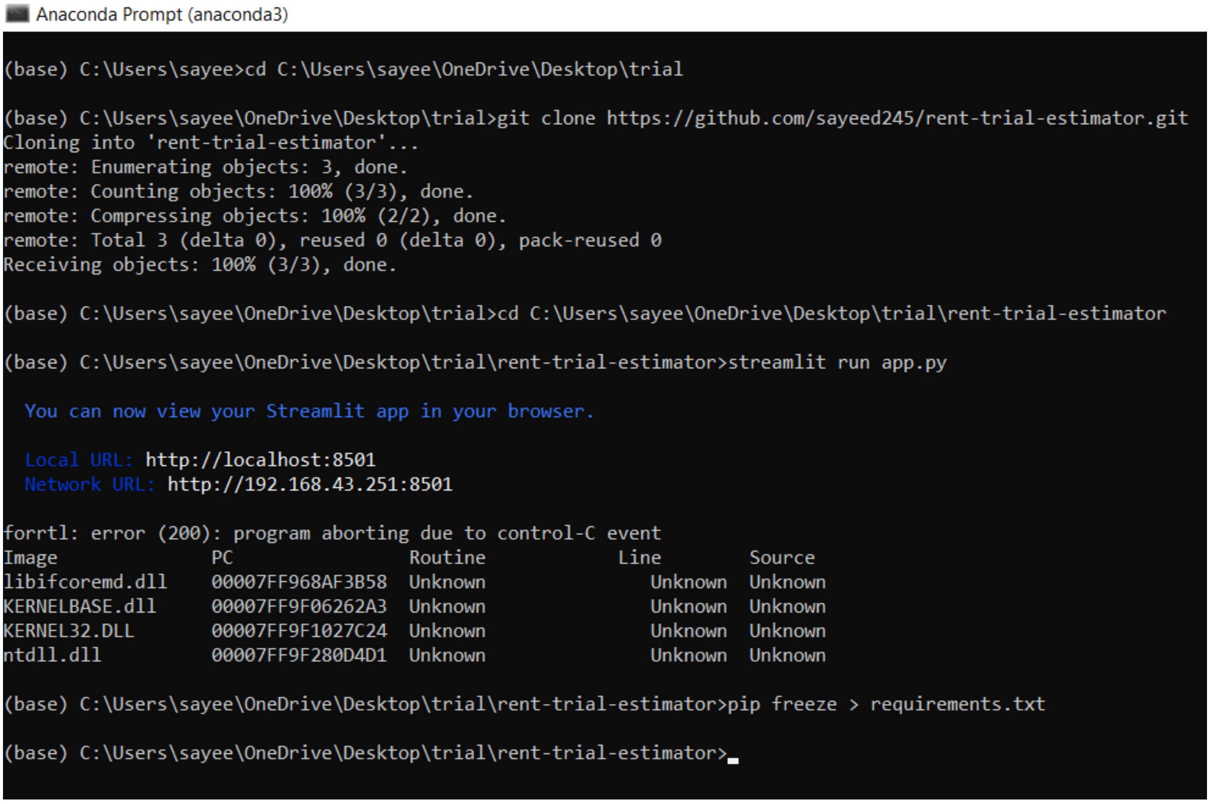
Procfile saved in File format

- Make a file called requirements.txt

  This will contain all the packages that was being used by you

  You can auto create this file by:

  Type: `pip freeze > requirements.txt`



```
(base) C:\Users\sayee>cd C:\Users\sayee\OneDrive\Desktop\trial

(base) C:\Users\sayee\OneDrive\Desktop\trial>git clone https://github.com/sayeed245/rent-trial-estimator.git
Cloning into 'rent-trial-estimator'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

(base) C:\Users\sayee\OneDrive\Desktop\trial>cd C:\Users\sayee\OneDrive\Desktop\trial\rent-trial-estimator

(base) C:\Users\sayee\OneDrive\Desktop\trial\rent-trial-estimator>streamlit run app.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://192.168.43.251:8501

forrtl: error (200): program aborting due to control-C event
Image              PC                Routine            Line        Source
libifcoremd.dll    00007FF968AF3B58  Unknown               Unknown  Unknown
KERNELBASE.dll     00007FF9F06262A3  Unknown               Unknown  Unknown
KERNEL32.DLL       00007FF9F1027C24  Unknown               Unknown  Unknown
ntdll.dll          00007FF9F280D4D1  Unknown               Unknown  Unknown

(base) C:\Users\sayee\OneDrive\Desktop\trial\rent-trial-estimator>pip freeze > requirements.txt

(base) C:\Users\sayee\OneDrive\Desktop\trial\rent-trial-estimator>_
```

We will find a requirements.txt file has been created in our folder(created on step 8)

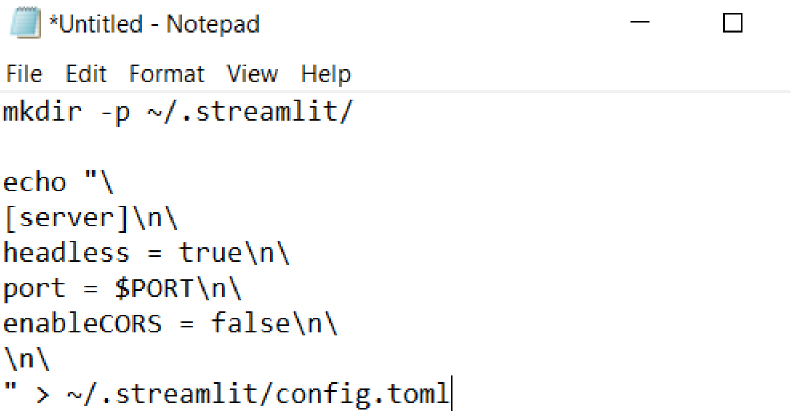
Open this file and check that you have the required packages that are needed to be installed in order to run the app. You can remove the extra packages that were added by: `pip freeze > requirements.txt.`

This will help you to avoid unnecessary complications that might arrive during deployment.

- Make a shell.sh file

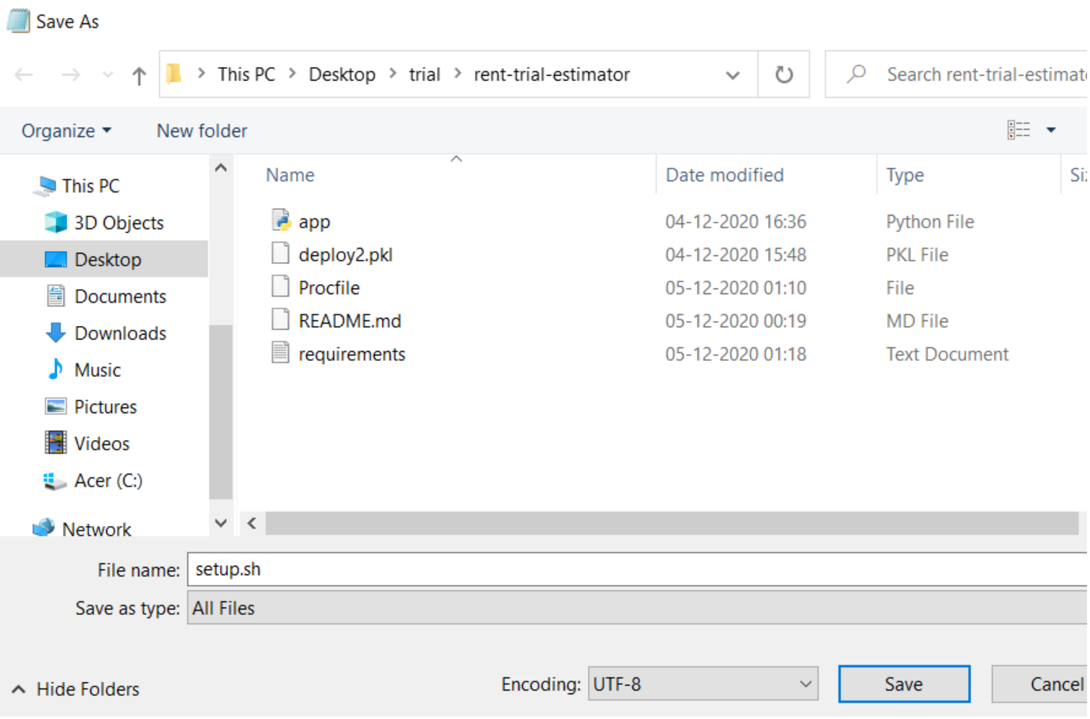
  Paste the following code:

```
mkdir -p ~/.streamlit/

echo "\
[server]\n\
headless = true\n\
port = $PORT\n\
enableCORS = false\n\
\n\
" > ~/.streamlit/config.toml
```



```
mkdir -p ~/.streamlit/

echo "\
[server]\n\
headless = true\n\
port = $PORT\n\
enableCORS = false\n\
\n\
" > ~/.streamlit/config.toml
```

Save the file as setup.sh

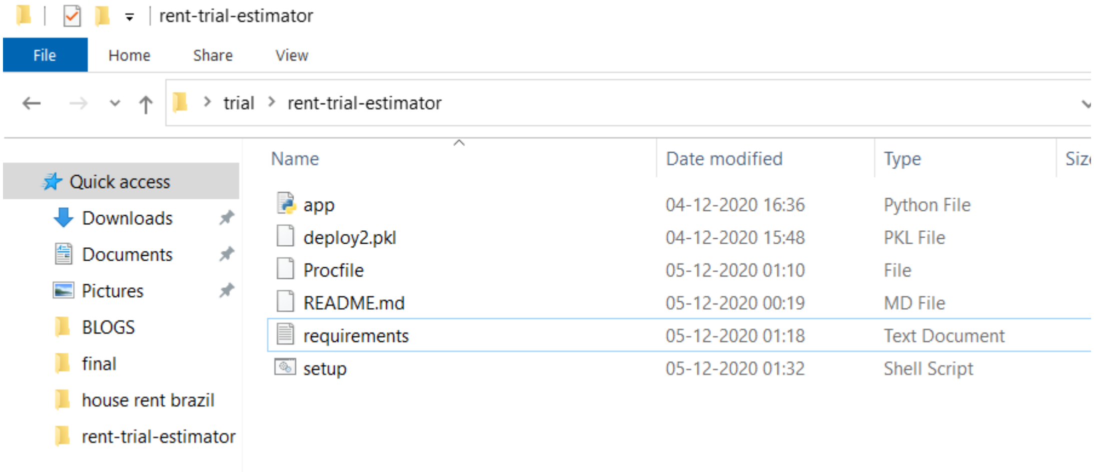


| Name | Date modified | Type | Si |
|---|---|---|---|
| app | 04-12-2020 16:36 | Python File | |
| deploy2.pkl | 04-12-2020 15:48 | PKL File | |
| Procfile | 05-12-2020 01:10 | File | |
| README.md | 05-12-2020 00:19 | MD File | |
| requirements | 05-12-2020 01:18 | Text Document | |

File name: setup.sh

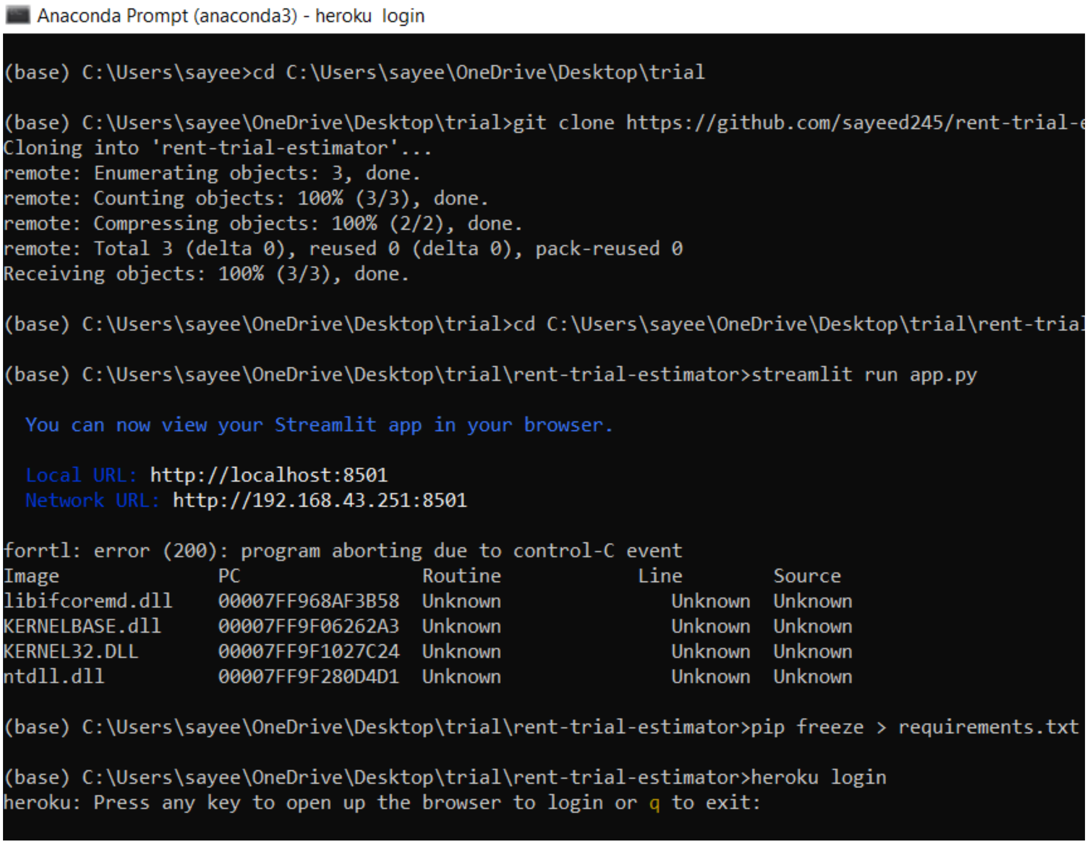
Save as type: All Files

Encoding: UTF-8



| Name | Date modified | Type | Siz |
|---|---|---|---|
| app | 04-12-2020 16:36 | Python File | |
| deploy2.pkl | 04-12-2020 15:48 | PKL File | |
| Procfile | 05-12-2020 01:10 | File | |
| README.md | 05-12-2020 00:19 | MD File | |
| requirements | 05-12-2020 01:18 | Text Document | |
| setup | 05-12-2020 01:32 | Shell Script | |

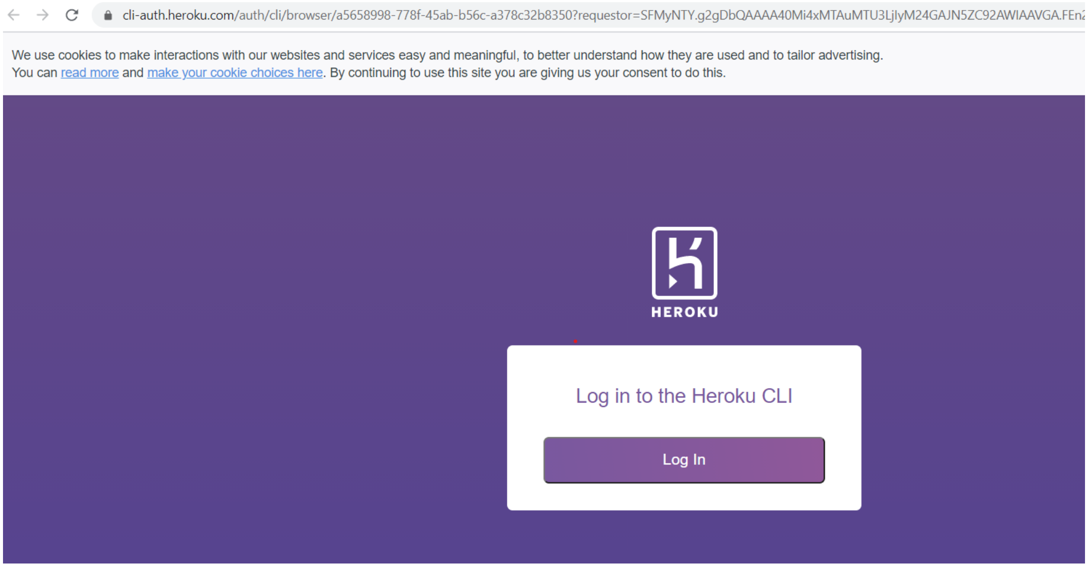
- Go to Command Prompt -> heroku login



```
(base) C:\Users\sayee>cd C:\Users\sayee\OneDrive\Desktop\trial

(base) C:\Users\sayee\OneDrive\Desktop\trial>git clone https://github.com/sayeed245/rent-trial-e
Cloning into 'rent-trial-estimator'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

(base) C:\Users\sayee\OneDrive\Desktop\trial>cd C:\Users\sayee\OneDrive\Desktop\trial\rent-trial

(base) C:\Users\sayee\OneDrive\Desktop\trial\rent-trial-estimator>streamlit run app.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://192.168.43.251:8501

forrtl: error (200): program aborting due to control-C event
Image              PC                Routine            Line        Source
libifcoremd.dll    00007FF968AF3B58  Unknown               Unknown  Unknown
KERNELBASE.dll     00007FF9F06262A3  Unknown               Unknown  Unknown
KERNEL32.DLL       00007FF9F1027C24  Unknown               Unknown  Unknown
ntdll.dll          00007FF9F280D4D1  Unknown               Unknown  Unknown

(base) C:\Users\sayee\OneDrive\Desktop\trial\rent-trial-estimator>pip freeze > requirements.txt

(base) C:\Users\sayee\OneDrive\Desktop\trial\rent-trial-estimator>heroku login
heroku: Press any key to open up the browser to login or q to exit:
```

Press any key to continue except q to continue

A window will open in the browser

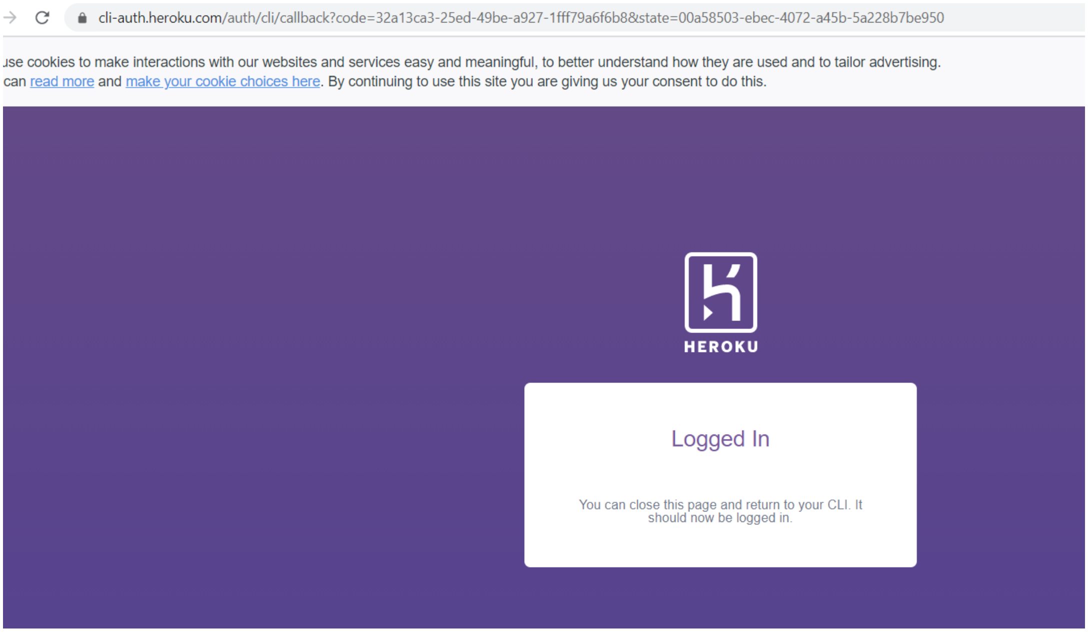


cli-auth.heroku.com/auth/cli/browser/a5658998-778f-45ab-b56c-a378c32b8350?requestor=SFMyNTY.g2gDbQAAAA40Mi4xMTAuMTU3LjIyM24GAJN5ZC92AWIAAVGA.FEn

We use cookies to make interactions with our websites and services easy and meaningful, to better understand how they are used and to tailor advertising. You can read more and make your cookie choices here. By continuing to use this site you are giving us your consent to do this.

Log in to the Heroku CLI

Log in and once you are logged in close the window

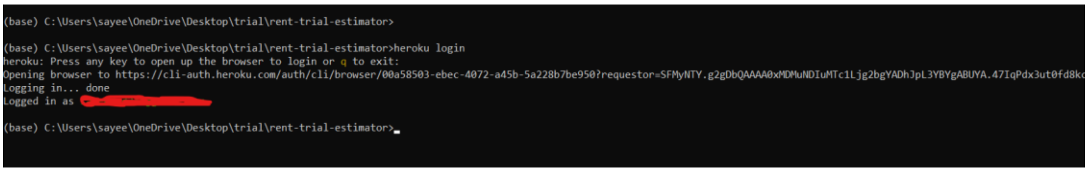


cli-auth.heroku.com/auth/cli/callback?code=32a13ca3-25ed-49be-a927-1fff79a6f6b8&state=00a58503-ebec-4072-a45b-5a228b7be950

ıse cookies to make interactions with our websites and services easy and meaningful, to better understand how they are used and to tailor advertising. can read more and make your cookie choices here. By continuing to use this site you are giving us your consent to do this.

Logged In

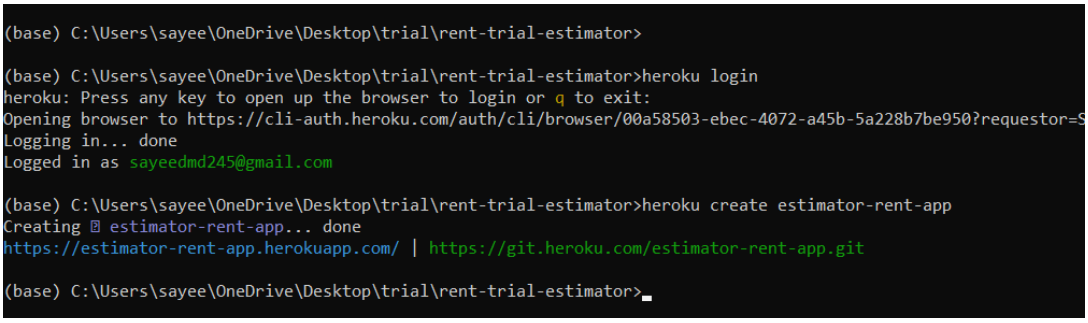
You can close this page and return to your CLI. It should now be logged in.



```
(base) C:\Users\sayee\OneDrive\Desktop\trial\rent-trial-estimator>

(base) C:\Users\sayee\OneDrive\Desktop\trial\rent-trial-estimator>heroku login
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/00a58503-ebec-4072-a45b-5a228b7be950?requestor=SFMyNTY.g2gDbQAAAA0xMDMuNDIuMTc1Ljg2bgYADhJpL3YBYgABUYA.47IqPdx3ut0fd8kc
Logging in... done
Logged in as [redacted]

(base) C:\Users\sayee\OneDrive\Desktop\trial\rent-trial-estimator>
```

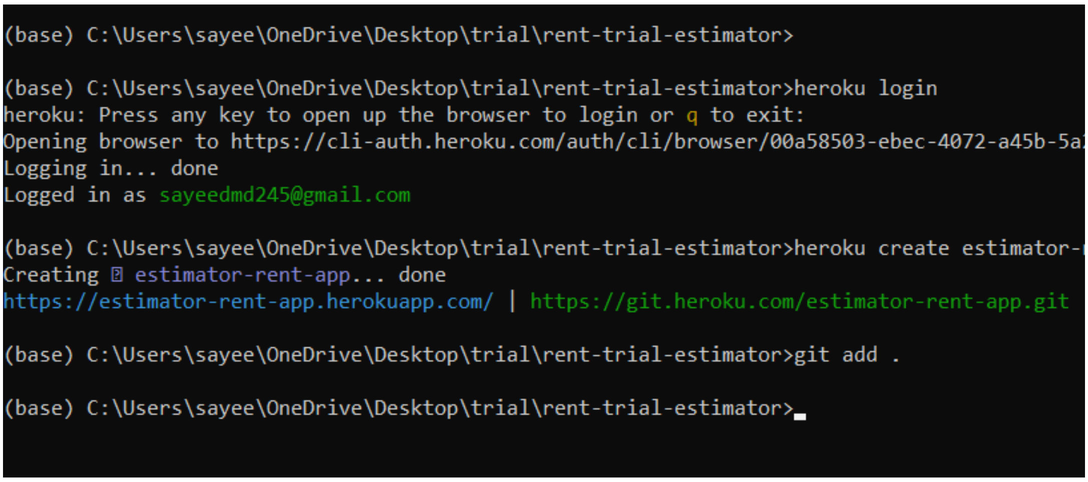
In Command Prompt ->heroku create estimator-rent-app

I give the name estimator-rent-app. You can give any other name. If your given name is not available use this command to give a name until it is taken.



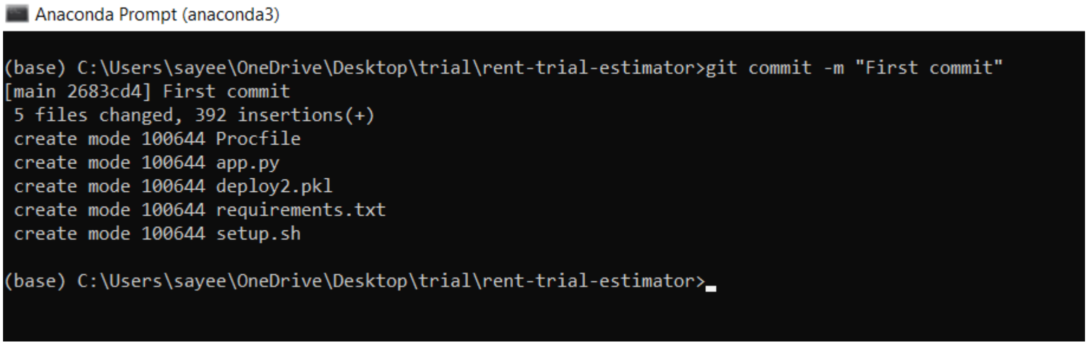
```
(base) C:\Users\sayee\OneDrive\Desktop\trial\rent-trial-estimator>

(base) C:\Users\sayee\OneDrive\Desktop\trial\rent-trial-estimator>heroku login
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/00a58503-ebec-4072-a45b-5a228b7be950?requestor=S
Logging in... done
Logged in as sayeedmd245@gmail.com

(base) C:\Users\sayee\OneDrive\Desktop\trial\rent-trial-estimator>heroku create estimator-rent-app
Creating □ estimator-rent-app... done
https://estimator-rent-app.herokuapp.com/ | https://git.heroku.com/estimator-rent-app.git

(base) C:\Users\sayee\OneDrive\Desktop\trial\rent-trial-estimator>_
```

Type:  git add . ->Enter



```
(base) C:\Users\sayee\OneDrive\Desktop\trial\rent-trial-estimator>

(base) C:\Users\sayee\OneDrive\Desktop\trial\rent-trial-estimator>heroku login
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/00a58503-ebec-4072-a45b-5a
Logging in... done
Logged in as sayeedmd245@gmail.com

(base) C:\Users\sayee\OneDrive\Desktop\trial\rent-trial-estimator>heroku create estimator-
Creating □ estimator-rent-app... done
https://estimator-rent-app.herokuapp.com/ | https://git.heroku.com/estimator-rent-app.git

(base) C:\Users\sayee\OneDrive\Desktop\trial\rent-trial-estimator>git add .

(base) C:\Users\sayee\OneDrive\Desktop\trial\rent-trial-estimator>_
```

Type: git commit –m “First commit” ->Enter



```
(base) C:\Users\sayee\OneDrive\Desktop\trial\rent-trial-estimator>git commit -m "First commit"
[main 2683cd4] First commit
 5 files changed, 392 insertions(+)
 create mode 100644 Procfile
 create mode 100644 app.py
 create mode 100644 deploy2.pkl
 create mode 100644 requirements.txt
 create mode 100644 setup.sh

(base) C:\Users\sayee\OneDrive\Desktop\trial\rent-trial-estimator>_
```

With my deployment I was getting an error

heroku[router]: at=error code=H14 desc="No web processes running"

I read the documentation in (https://devcenter.heroku.com/articles/error-codes#h14-no-web-dynos-running) and found that I have to add another command in command prompt

In command prompt-> heroku ps:scale worker 1

Running heroku ps:scale web=1 scales your app to one running dyno, basically meaning you have one server running your app currently.

Now use the link to view your app and share with the world.

(https://house-rent-estimate-app.herokuapp.com/)