This model involves creating a web app which is able to take inputs and predict house Rent by using following parameters:

1. Latitude.
2. Longitude.
3. Gym.
4. Lift.
5. Swimming pool.
6. Negotiable.
7. Size of Property.
8. Age of Property.
9. Bathroom.
10. Cup Boards.
11. Floors.
12. Total number of floors.
13. Balconies.

[Data Source](https://www.kaggle.com/saisaathvik/house-rent-prices-of-metropolitan-cities-in-india)

* This project is an end-to-end implementation of a project where the goal is to predict house rent prices for various different parameters (Features).
* This project uses a machine learning model in the backend to predict the rent prices.
* This project uses a Random Forest Regressor model (which turned out to be the best model).
* The models were evaluated on the basis of:

Root Mean Squared Error(RMSE).

Running the project on local machine

Step 1 – Install all the required libraries.

import numpy as np

import pandas as pd

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import mean\_absolute\_error

from sklearn.ensemble import RandomForestRegressor

Step 2 – Import Required .CSV file.

housing\_train = pd.read\_csv('/content/train.csv') # import training file

housing\_test = pd.read\_csv('/content/test.csv')

Step 3 – Data Pre-Processing.

housing\_train.shape

housing\_train.describe()

housing\_train.columns

housing\_train.info()

Step 4 – Finding The correlation Metrics.

mport matplotlib.pyplot as plt

import seaborn as sns

from warnings import filterwarnings

filterwarnings(action='ignore', category=DeprecationWarning)

plt.figure(figsize=(24,13))

# mask is used to see only half heatmap i.e. lower or upper triangular part

d = housing\_train.drop("id",axis=1)

corr = d.corr()

mask = np.zeros\_like(corr, dtype=np.bool)

mask[np.triu\_indices\_from(mask)] = True

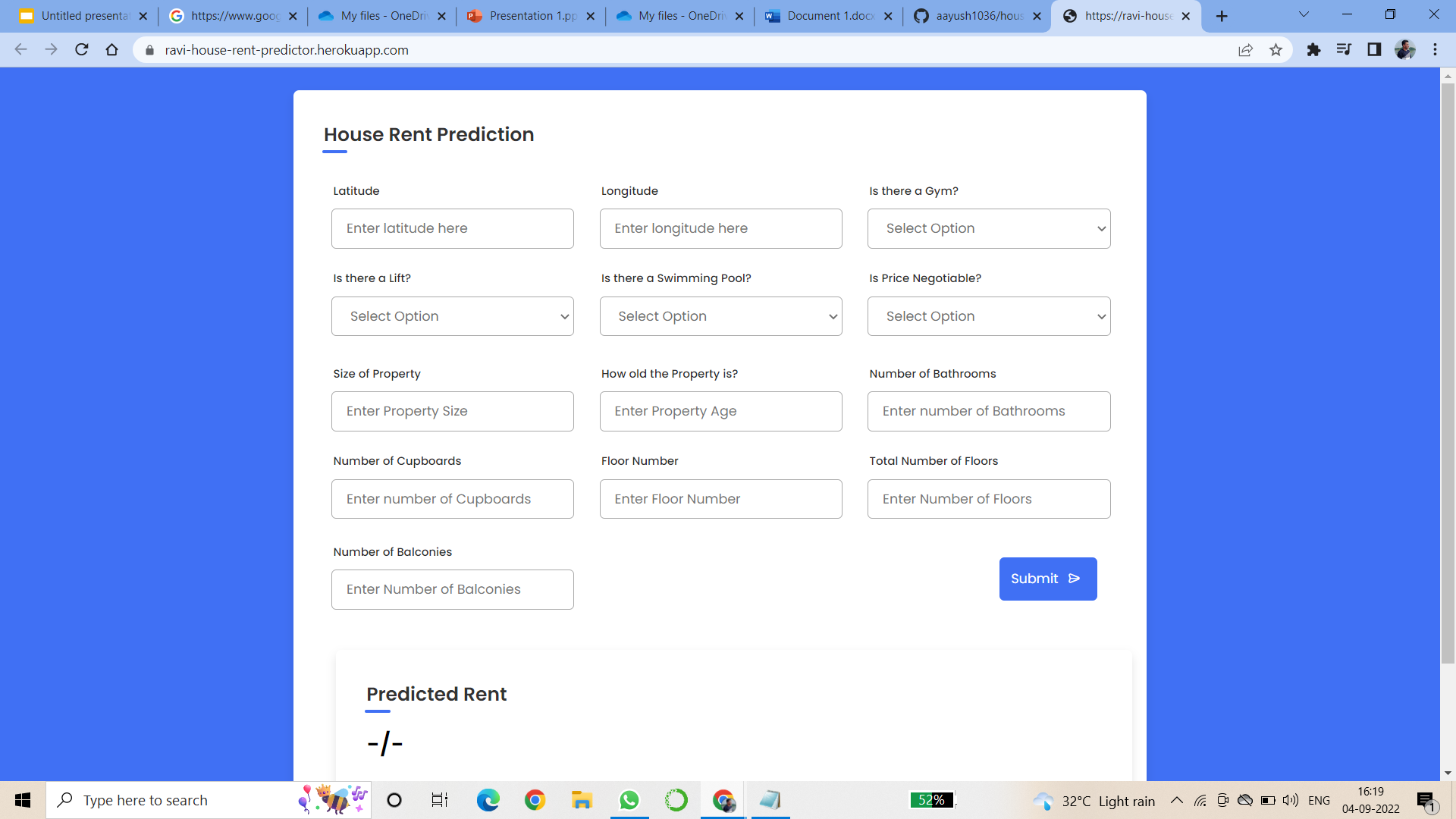
sns.heatmap(corr, annot=True, fmt=".2f", mask=mask)

Step 5 - Run the code

python Machineight Hackathon-House Rent prediction.ipynb

Index structure

Index(['id', 'type', 'locality', 'activation\_date', 'latitude', 'longitude', 'lease\_type', 'gym', 'lift', 'swimming\_pool', 'negotiable', 'furnishing', 'parking', 'property\_size', 'property\_age', 'bathroom', 'facing', 'cup\_board', 'floor', 'total\_floor', 'amenities', 'water\_supply', 'building\_type', 'balconies', 'rent'], dtype='object')

App structure  


In the app Structure, you seeing that we have 13 parameters to predict the house rent.