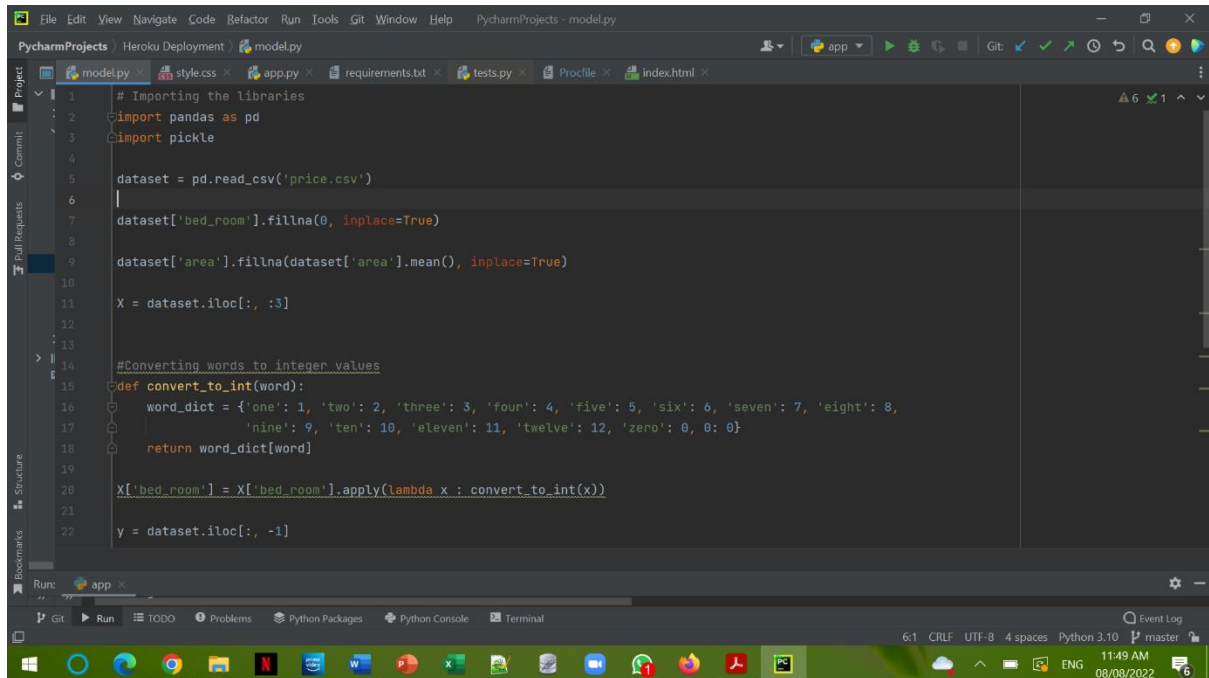


Name: Isha Panjwani

Batch Code: LISUM11

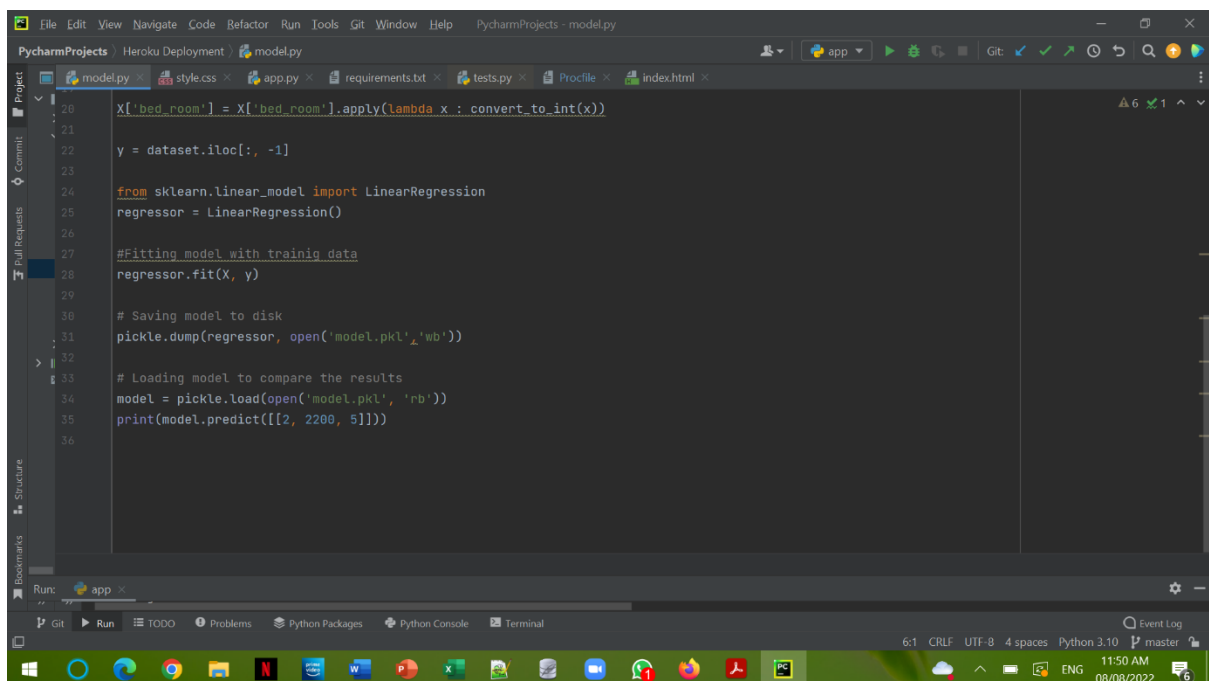
Submission Date: 08/08/2022

#model.py



This screenshot shows the first 22 lines of the Python script model.py in the PyCharm IDE. The code imports pandas and pickle, reads a CSV file named 'price.csv', and fills missing values in the 'bed\_room' and 'area' columns. It then selects the first three columns of the dataset and defines a function to convert words in the 'bed\_room' column to integers based on a predefined dictionary.

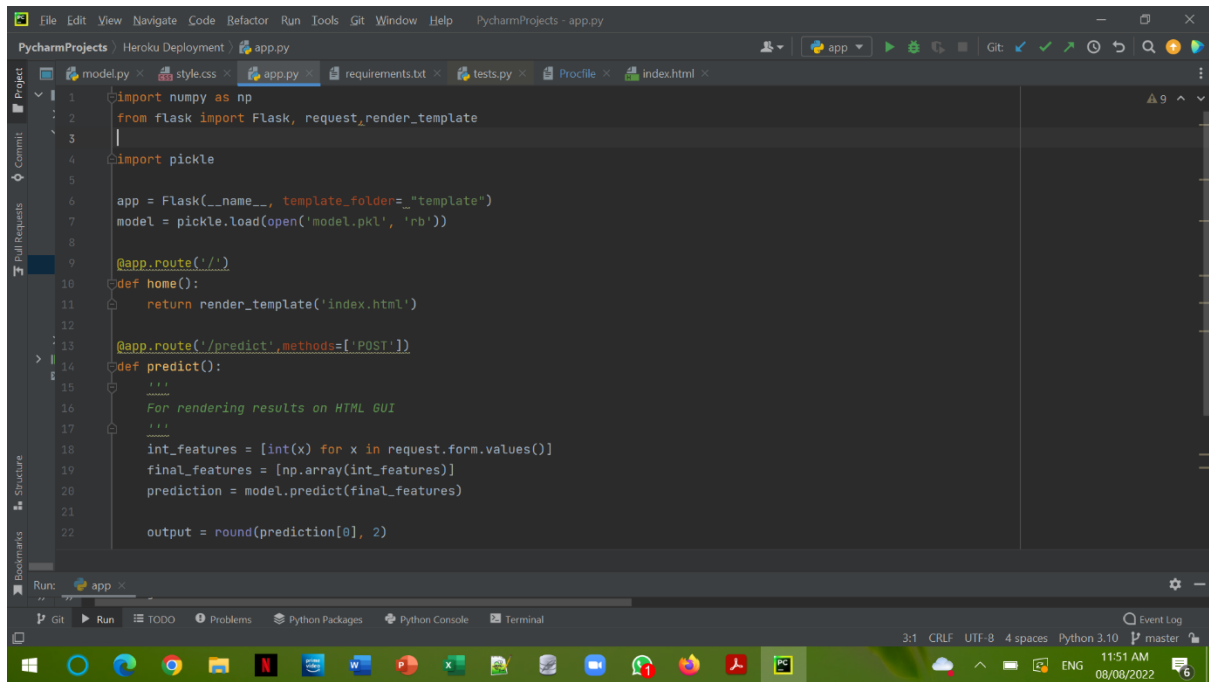
```
1 # Importing the libraries
2 import pandas as pd
3 import pickle
4
5 dataset = pd.read_csv('price.csv')
6
7 dataset['bed_room'].fillna(0, inplace=True)
8
9 dataset['area'].fillna(dataset['area'].mean(), inplace=True)
10
11 X = dataset.iloc[:, :3]
12
13
14 #Converting words to integer values
15 def convert_to_int(word):
16     word_dict = {'one': 1, 'two': 2, 'three': 3, 'four': 4, 'five': 5, 'six': 6, 'seven': 7, 'eight': 8,
17                 'nine': 9, 'ten': 10, 'eleven': 11, 'twelve': 12, 'zero': 0, 0: 0}
18     return word_dict[word]
19
20 X['bed_room'] = X['bed_room'].apply(lambda x : convert_to_int(x))
21
22 y = dataset.iloc[:, -1]
```



This screenshot shows the continuation of the Python script model.py, lines 23 to 36. It imports LinearRegression from sklearn.linear\_model, fits the model with the training data, saves the model to a file named 'model.pkl', and then loads the model to make a prediction on a specific input.

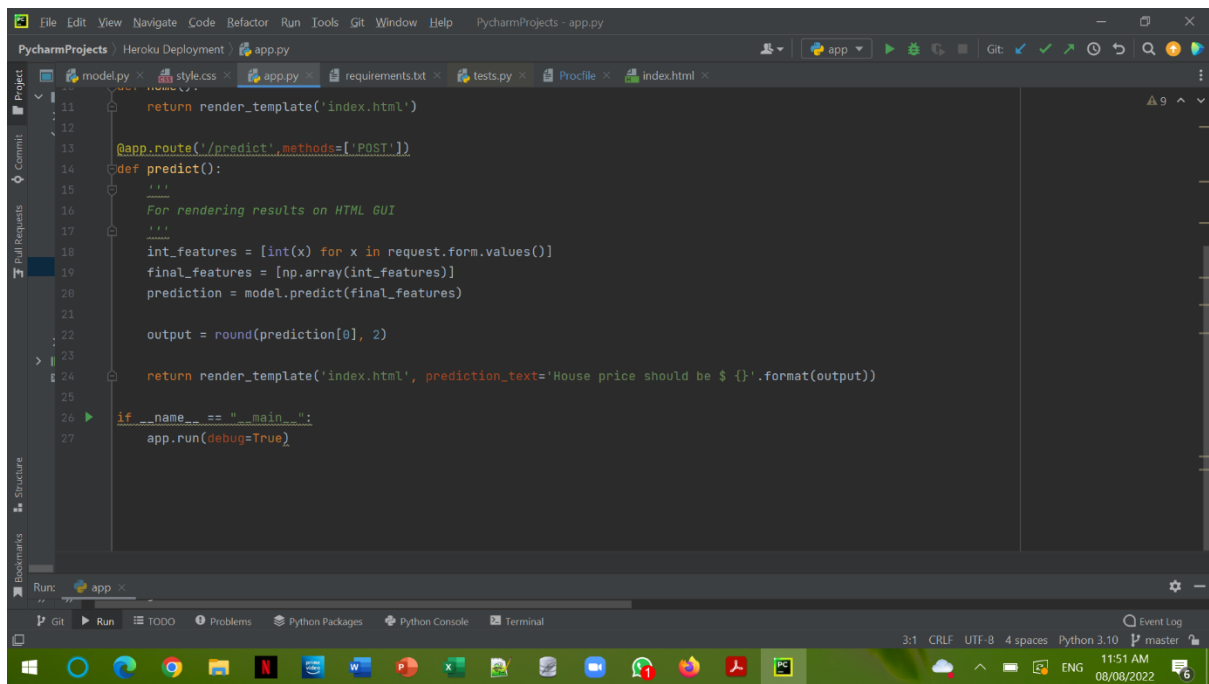
```
23
24 from sklearn.linear_model import LinearRegression
25 regressor = LinearRegression()
26
27 #Fitting model with trainig data
28 regressor.fit(X, y)
29
30 # Saving model to disk
31 pickle.dump(regressor, open('model.pkl','wb'))
32
33 # Loading model to compare the results
34 model = pickle.load(open('model.pkl', 'rb'))
35 print(model.predict([[2, 2200, 5]]))
36
```

#app.py



This screenshot shows the first 22 lines of the `app.py` file in the PyCharm IDE. The code imports `numpy` and `Flask`, loads a pre-trained model from `model.pkl`, and defines a `predict` endpoint that processes form data and returns a prediction rounded to 2 decimal places.

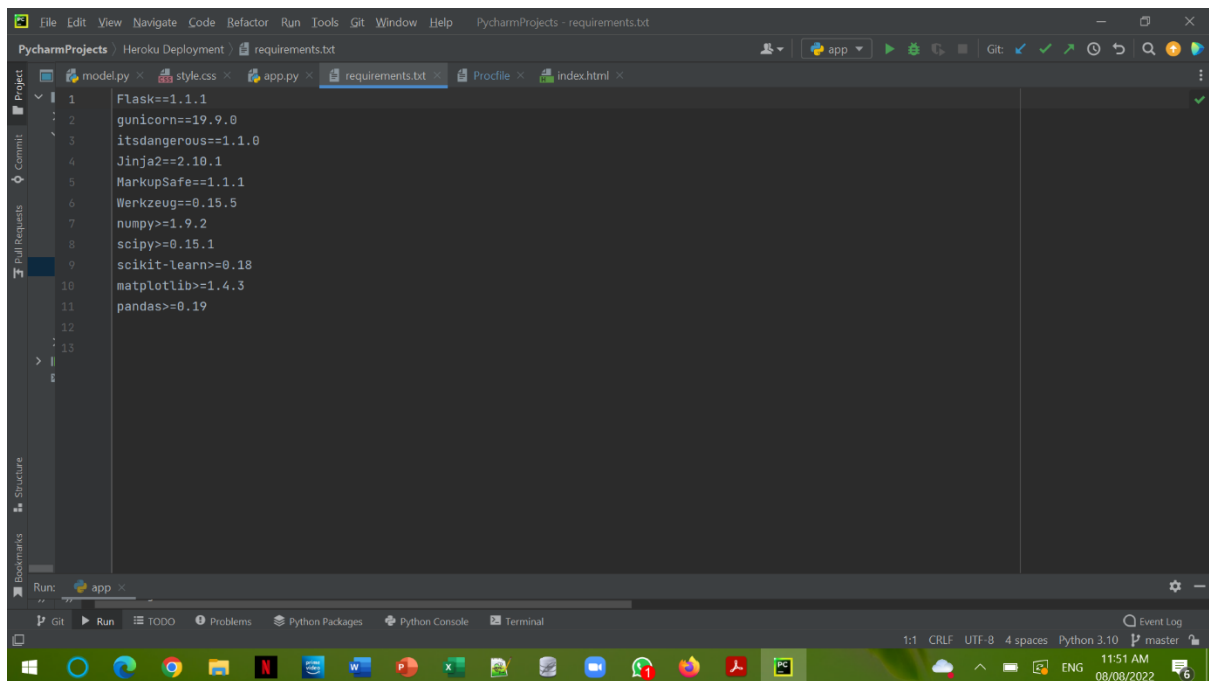
```
1 import numpy as np
2 from flask import Flask, request, render_template
3
4 import pickle
5
6 app = Flask(__name__, template_folder="template")
7 model = pickle.load(open('model.pkl', 'rb'))
8
9 @app.route('/')
10 def home():
11     return render_template('index.html')
12
13 @app.route('/predict', methods=['POST'])
14 def predict():
15     """
16     For rendering results on HTML GUI
17     """
18     int_features = [int(x) for x in request.form.values()]
19     final_features = [np.array(int_features)]
20     prediction = model.predict(final_features)
21
22     output = round(prediction[0], 2)
```



This screenshot shows the continuation of the `app.py` file, including the return statement for the `predict` function, the `if __name__ == '__main__':` block, and the `app.run()` call. The code is consistent with the first screenshot, showing the full implementation of the Flask application.

```
23     return render_template('index.html', prediction_text='House price should be $ {}'.format(output))
24
25
26 if __name__ == '__main__':
27     app.run(debug=True)
```

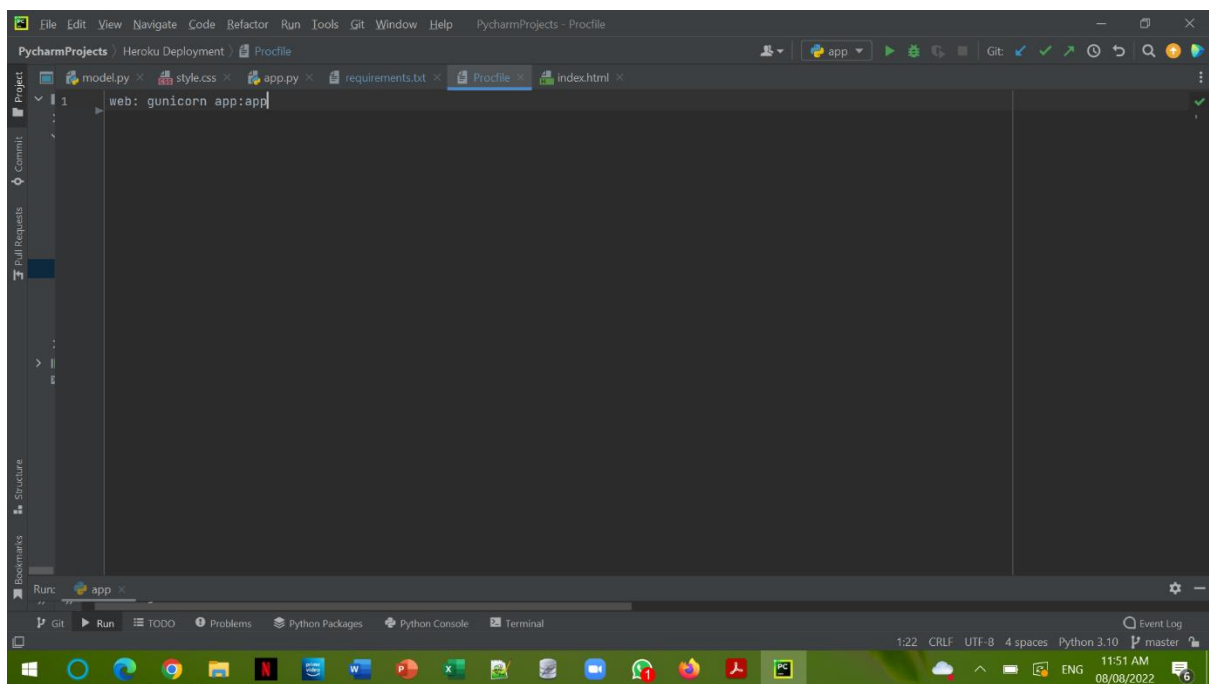
## #requirements.txt



A screenshot of the PyCharm IDE interface. The main editor window displays the contents of a file named 'requirements.txt'. The file contains a list of Python packages and their versions, separated by line numbers 1 through 13. The packages listed are Flask, gunicorn, itsdangerous, Jinja2, MarkupSafe, Werkzeug, numpy, scipy, scikit-learn, matplotlib, and pandas. The PyCharm interface includes a top menu bar, a left sidebar with project structure, and a bottom status bar showing the current file encoding and settings.

```
1 Flask==1.1.1
2 gunicorn==19.9.0
3 itsdangerous==1.1.0
4 Jinja2==2.10.1
5 MarkupSafe==1.1.1
6 Werkzeug==0.15.5
7 numpy>=1.9.2
8 scipy>=0.15.1
9 scikit-learn>=0.18
10 matplotlib>=1.4.3
11 pandas>=0.19
12
13
```

## #Procfile



A screenshot of the PyCharm IDE interface. The main editor window displays the contents of a file named 'Procfile'. The file contains a single line of text: 'web: gunicorn app:app'. The PyCharm interface includes a top menu bar, a left sidebar with project structure, and a bottom status bar showing the current file encoding and settings.

```
1 web: gunicorn app:app
```

## #index.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>ML API</title>
6 <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
7 <link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
8 <link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
9 <link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
10 <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
11 </head>
12
13 <body>
14 <div class="login">
15 <h1>Predict House Price</h1>
16
17 <!-- Main Input For Receiving Query to our ML -->
18 <form action="{{ url_for('predict')}}" method="post">
19 <input type="text" name="no_of_rooms" placeholder="Number of Rooms" required="required" />
20 <input type="text" name="area" placeholder="Area (in square feet)" required="required" />
21 <input type="text" name="house_age" placeholder="House Age" required="required" />
22
23 <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
24 </form>
25
26 <br>
27 <br>
28 {{ prediction_text }}
29
30 </div>
31
32 
33 </body>
34 </html>
```

#github repository <https://github.com/isha1912/Heroku-Deployment>

The screenshot shows a web browser window displaying the GitHub repository page for 'isha1912/Heroku-Deployment'. The repository is public and has 0 stars, 1 watcher, and 0 forks. The main content area shows a list of files and folders, including 'static', 'template', 'Procfile', 'app.py', 'model.pkl', 'model.py', 'price.csv', and 'requirements.txt'. The 'Procfile' file is highlighted, showing its commit history. The right sidebar contains sections for 'About', 'Releases', and 'Packages'. The bottom of the page features a Windows taskbar with various application icons and a system clock showing 12:07 PM on 08/08/2022.

isha1912 / Heroku-Deployment Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags

Go to file Add file Code

About

No description, website, or topics provided.

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

isha1912 Update Profile 3db1cfa 27 minutes ago 5 commits

static	Add files via upload	11 hours ago
template	Add files via upload	11 hours ago
Procfile	Update Procfile	27 minutes ago
app.py	Add files via upload	11 hours ago
model.pkl	Add files via upload	11 hours ago
model.py	Add files via upload	11 hours ago
price.csv	Add files via upload	11 hours ago
requirements.txt	Add files via upload	11 hours ago

Help people interested in this repository understand your project by adding a README. Add a README

12:07 PM 08/08/2022

## #Heroku Account

The screenshot shows the Heroku dashboard for the application 'price-prediction-for-home'. The 'Manual deploy' section is active, showing the deployment of the 'main' branch. The deployment process is complete, with all steps marked as successful: 'Receive code from GitHub', 'Build main 3db1cfa2', 'Release phase', and 'Deploy to Heroku'. A message at the bottom states 'Your app was successfully deployed.' with a 'View' button. The browser's address bar shows 'dashboard.heroku.com/apps/price-prediction-for-home/deploy/github'. The taskbar at the bottom includes icons for various applications and the system clock showing 12:09 PM on 08/08/2022.

Manual deploy

Deploy the current state of a branch to this app.

Deploy a GitHub branch

This will deploy the current state of the branch you specify below. [Learn more.](#)

Choose a branch to deploy

main

Deploy Branch

Receive code from GitHub

Build main 3db1cfa2

Release phase

Deploy to Heroku

Your app was successfully deployed.

[View](#)

This is a duplicate of the screenshot above, showing the same Heroku dashboard interface with the deployment of the 'main' branch completed successfully. The message 'Your app was successfully deployed.' and the 'View' button are visible at the bottom of the deployment section. The browser and taskbar details are identical to the first screenshot.

Manual deploy

Deploy the current state of a branch to this app.

Deploy a GitHub branch

This will deploy the current state of the branch you specify below. [Learn more.](#)

Choose a branch to deploy

main

Deploy Branch

Receive code from GitHub

Build main 3db1cfa2

Release phase

Deploy to Heroku

Your app was successfully deployed.

[View](#)

## #ML API- home page

The screenshot shows a web browser window with the URL `127.0.0.1:5000`. The page has a dark background and is titled "Predict House Price". It features three input fields: "Number of Rooms", "Area (in square feet)", and "House Age". Below these fields is a blue "Predict" button. At the bottom left, there is a logo for "Data Glacier" with the tagline "Your Deep Learning Partner". The browser's taskbar at the bottom shows various application icons and the system clock indicating 12:20 PM on 08/08/2022.

## #ML API- Predict

This screenshot shows the same "Predict House Price" interface, but now it displays the prediction result. Below the "Predict" button, the text "House price should be \$ 53728.61" is visible. The "Data Glacier" logo and tagline remain at the bottom left. The browser's taskbar at the bottom shows the system clock updated to 12:21 PM on 08/08/2022.