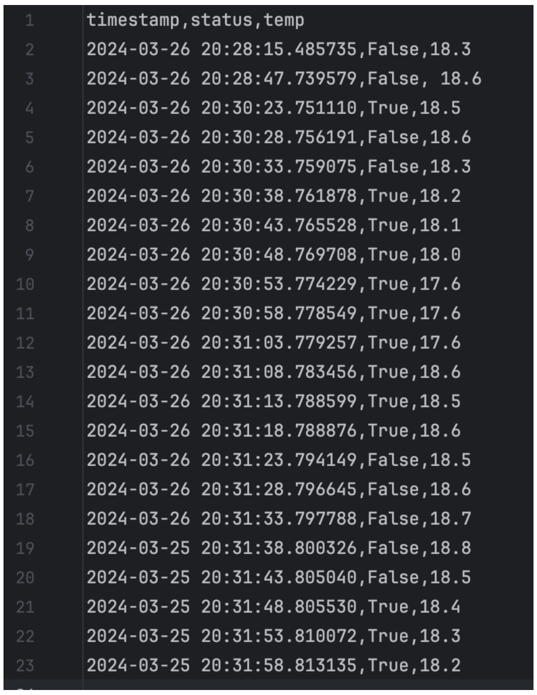
Analysing Data Using Pandas Data Frames

Sample Data

This csv file stores data about current temp and a status (on/off), for example: status of motion sensor or light.



```
timestamp,status,temp
2024-03-26 20:28:15.485735,False,18.3
2024-03-26 20:28:47.739579,False, 18.6
2024-03-26 20:30:23.751110,True,18.5
2024-03-26 20:30:28.756191,False,18.6
2024-03-26 20:30:33.759075,False,18.3
2024-03-26 20:30:38.761878,True,18.2
2024-03-26 20:30:43.765528,True,18.1
2024-03-26 20:30:48.769708,True,18.0
2024-03-26 20:30:53.774229,True,17.6
2024-03-26 20:30:58.778549,True,17.6
2024-03-26 20:31:03.779257,True,17.6
2024-03-26 20:31:08.783456,True,18.6
2024-03-26 20:31:13.788599,True,18.5
2024-03-26 20:31:18.788876,True,18.6
2024-03-26 20:31:23.794149,False,18.5
2024-03-26 20:31:28.796645,False,18.6
2024-03-26 20:31:33.797788,False,18.7
2024-03-25 20:31:38.800326,False,18.8
2024-03-25 20:31:43.805040,False,18.5
2024-03-25 20:31:48.805530,True,18.4
2024-03-25 20:31:53.810072,True,18.3
2024-03-25 20:31:58.813135,True,18.2
```

Website

The user can enter a date. It will filter the data stored in a csv to:

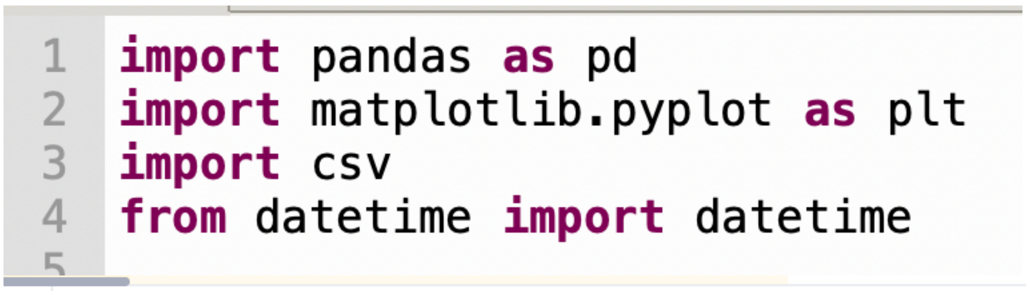
1. Display the percentage of the time the status = True on a specific day
2. Generate a chart to show temperatures on a specific date

activity.py

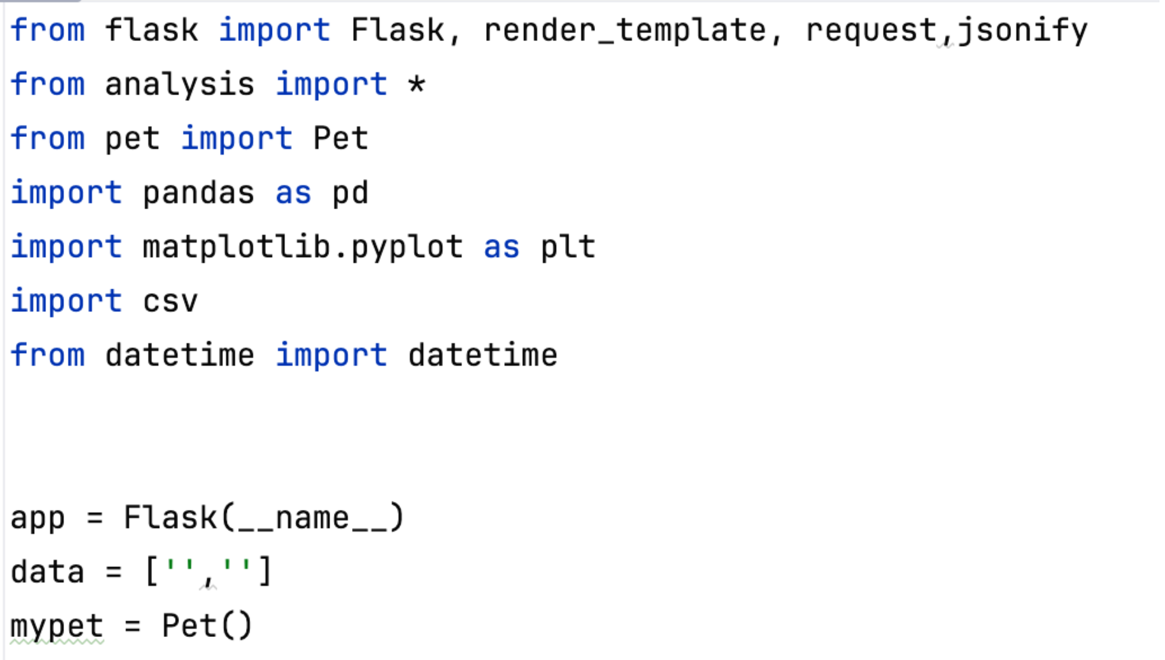
The code required to work out a percentage and charts from a pandas dataframe involved the following:

1. Convert csv into a pandas data frame (df)
2. Format the status column so that instead of True/False, it is changed to 1 or 0
3. Split the timestamp into a date column and time column
4. Filter the dataframe by data
5. Work out the percentage of how many times status = 1 during a specific time period (day).
6. Generate a chart for temp and status activity for a specific date and time period.
7. Display feedback if the % is above or below an expected %

Libraries:

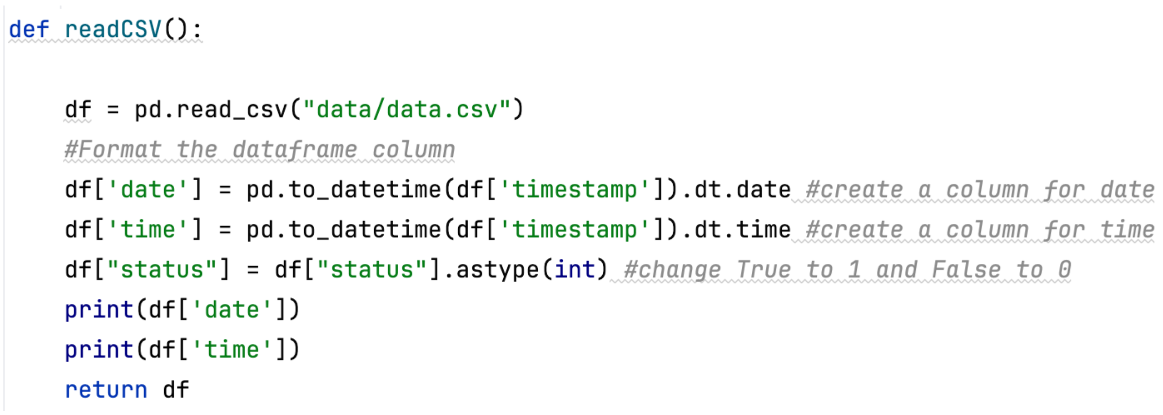


```python
import pandas as pd
import matplotlib.pyplot as plt
import csv
from datetime import datetime
```



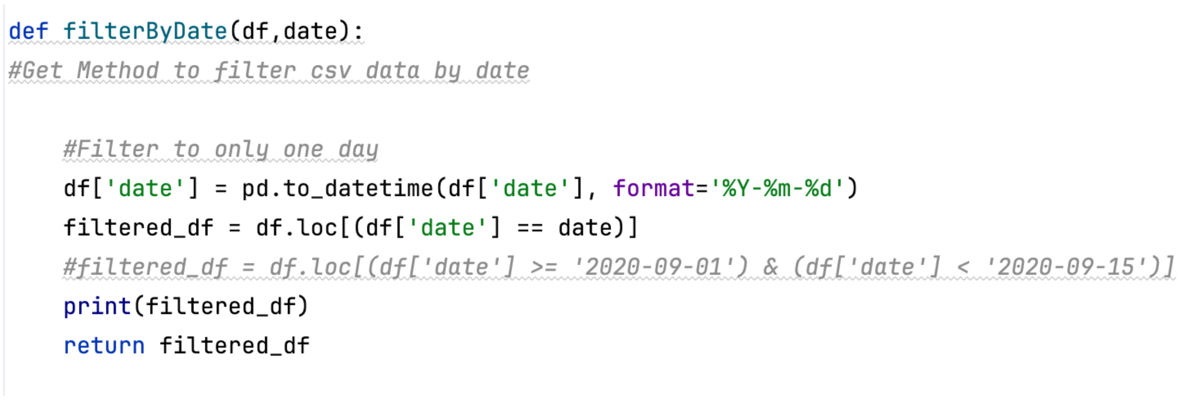
```python
from flask import Flask, render_template, request,jsonify
from analysis import *
from pet import Pet
import pandas as pd
import matplotlib.pyplot as plt
import csv
from datetime import datetime



app = Flask(__name__)
data = ['','']
mypet = Pet()
```

How to convert the csv file into a pandas data frame and create a date and time columns from the timestamp:



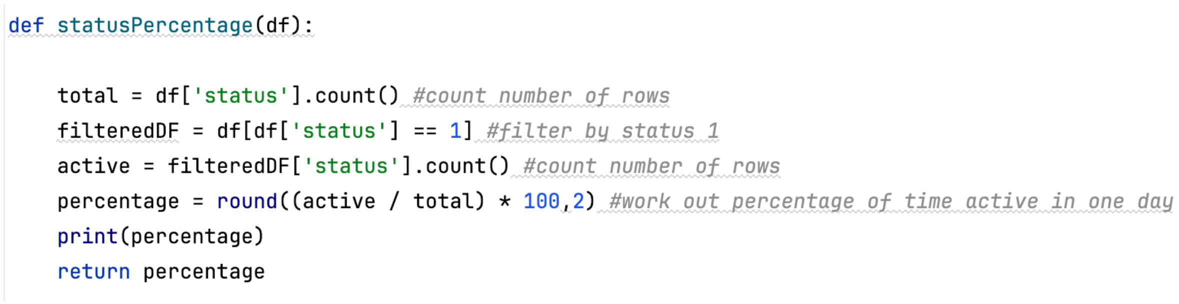
```python
def readCSV():

    df = pd.read_csv("data/data.csv")
    #Format the dataframe column
    df['date'] = pd.to_datetime(df['timestamp']).dt.date #create a column for date
    df['time'] = pd.to_datetime(df['timestamp']).dt.time #create a column for time
    df["status"] = df["status"].astype(int) #change True to 1 and False to 0
    print(df['date'])
    print(df['time'])
    return df
```

This code is used to filter the data frame by date and return a filtered df:



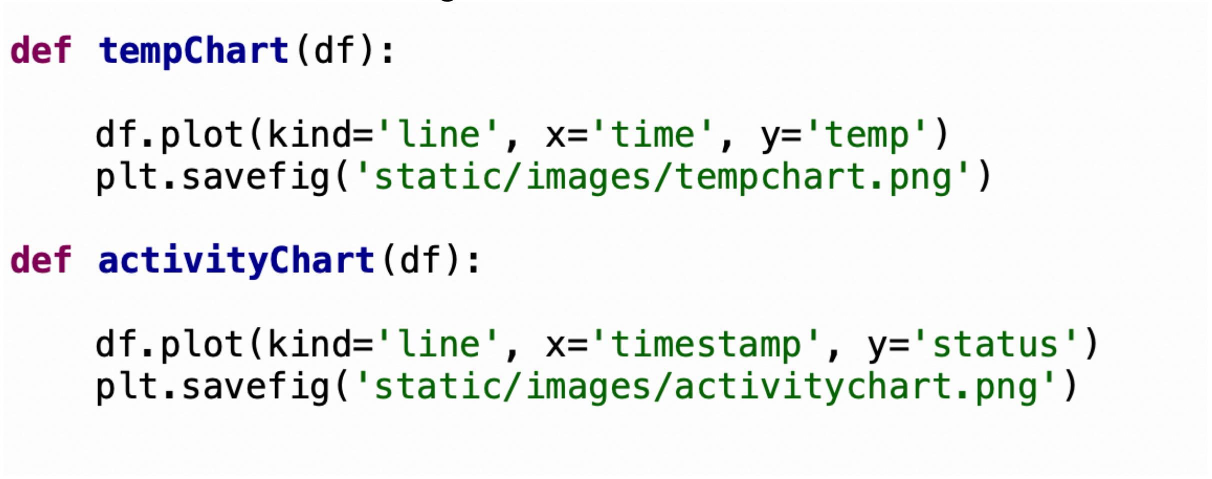
```python
def filterByDate(df,date):
#Get Method to filter csv data by date

    #Filter to only one day
    df['date'] = pd.to_datetime(df['date'], format='%Y-%m-%d')
    filtered_df = df.loc[(df['date'] == date)]
    #filtered_df = df.loc[(df['date'] >= '2020-09-01') & (df['date'] < '2020-09-15')]
    print(filtered_df)
    return filtered_df
```

This code is used to work out how often the status = 1 during specific day:



```python
def statusPercentage(df):

    total = df['status'].count() #count number of rows
    filteredDF = df[df['status'] == 1] #filter by status 1
    active = filteredDF['status'].count() #count number of rows
    percentage = round((active / total) * 100,2) #work out percentage of time active in one day
    print(percentage)
    return percentage
```

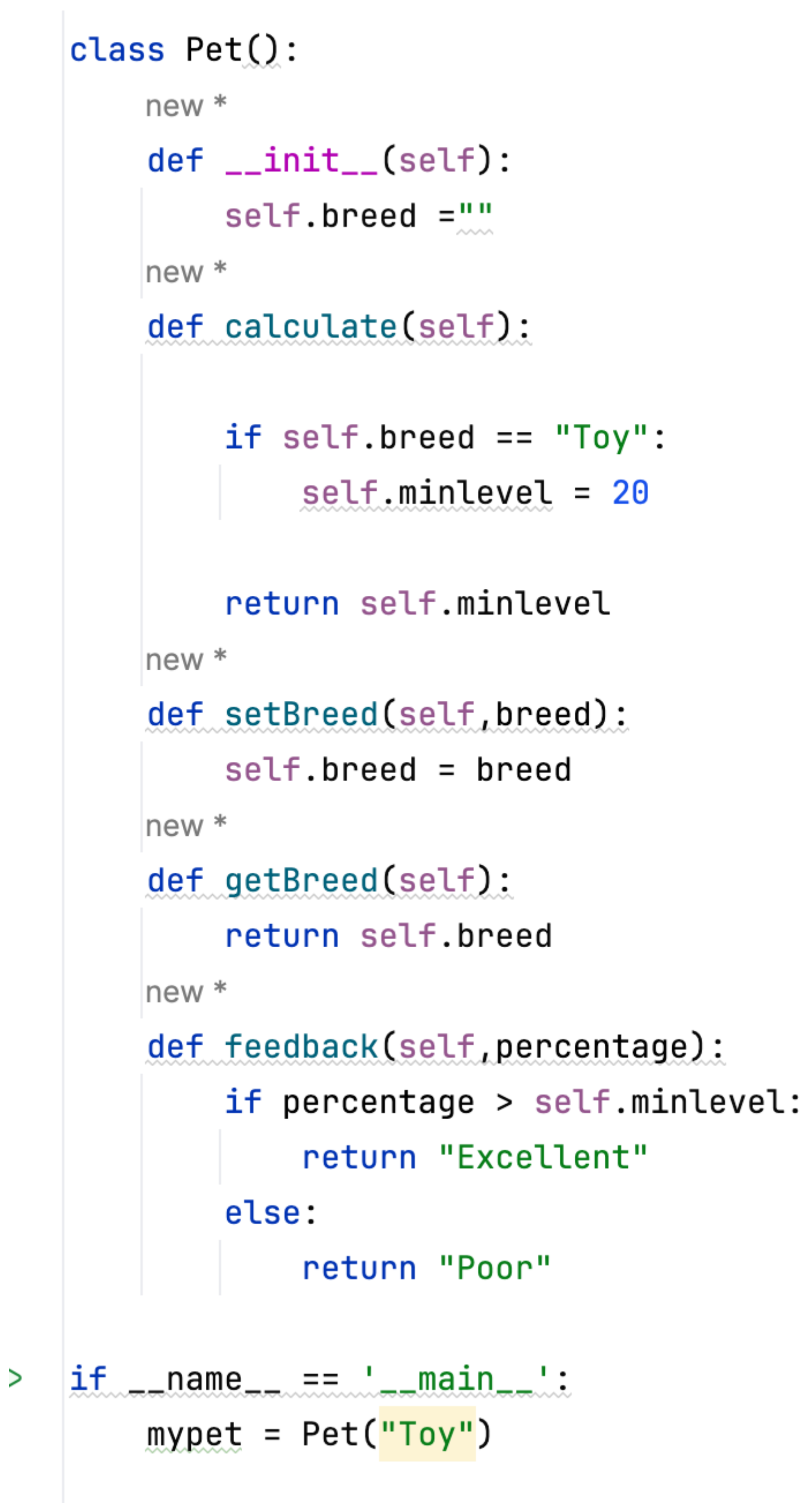
This is basic code for creating a chart from a data frame:



```python
def tempChart(df):

    df.plot(kind='line', x='time', y='temp')
    plt.savefig('static/images/tempchart.png')

def activityChart(df):

    df.plot(kind='line', x='timestamp', y='status')
    plt.savefig('static/images/activitychart.png')
```

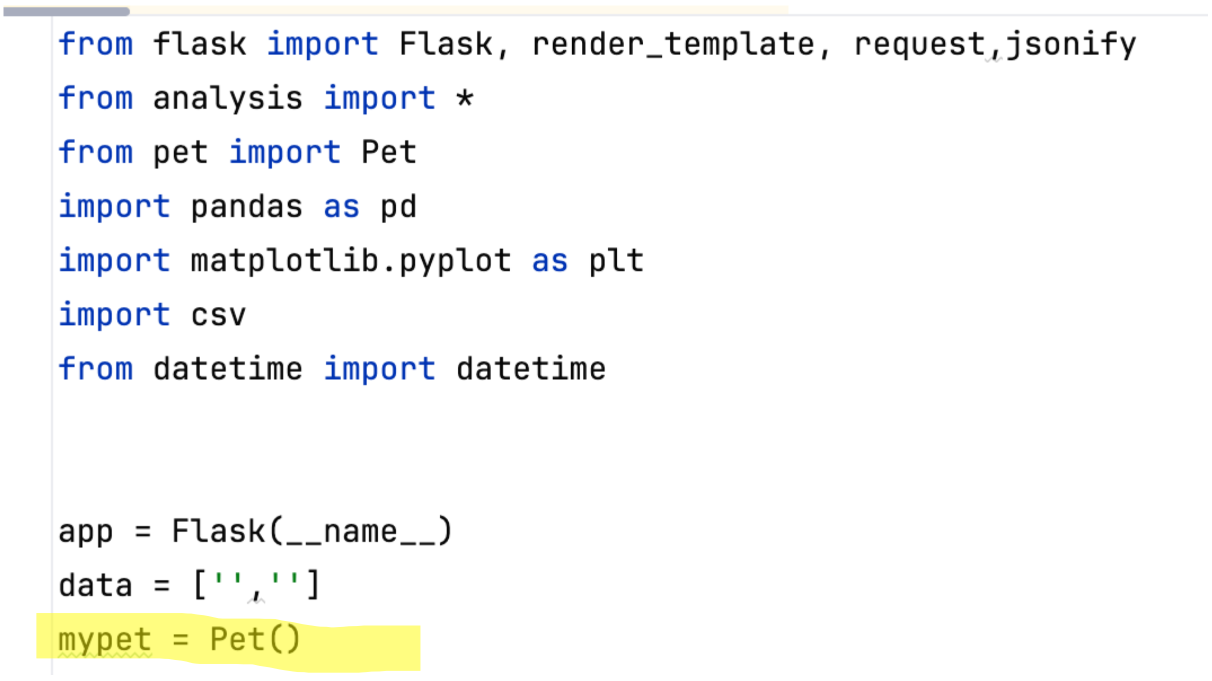
Pet class

This class is used to create a pet object that stored data about a pet's breed and will return how much % activity this dog breed should have per day.

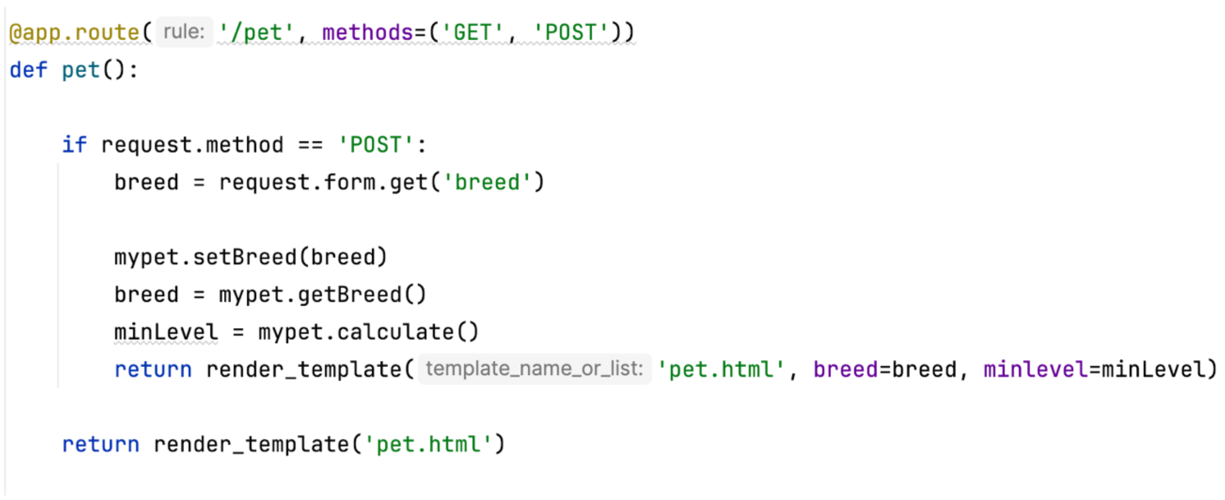


```
class Pet():
    new *
    def __init__(self):
        self.breed =""
    new *
    def calculate(self):

        if self.breed == "Toy":
            self.minlevel = 20

        return self.minlevel
    new *
    def setBreed(self,breed):
        self.breed = breed
    new *
    def getBreed(self):
        return self.breed
    new *
    def feedback(self,percentage):
        if percentage > self.minlevel:
            return "Excellent"
        else:
            return "Poor"

if __name__ == '__main__':
    mypet = Pet("Toy")
```

App.py code for pet.html

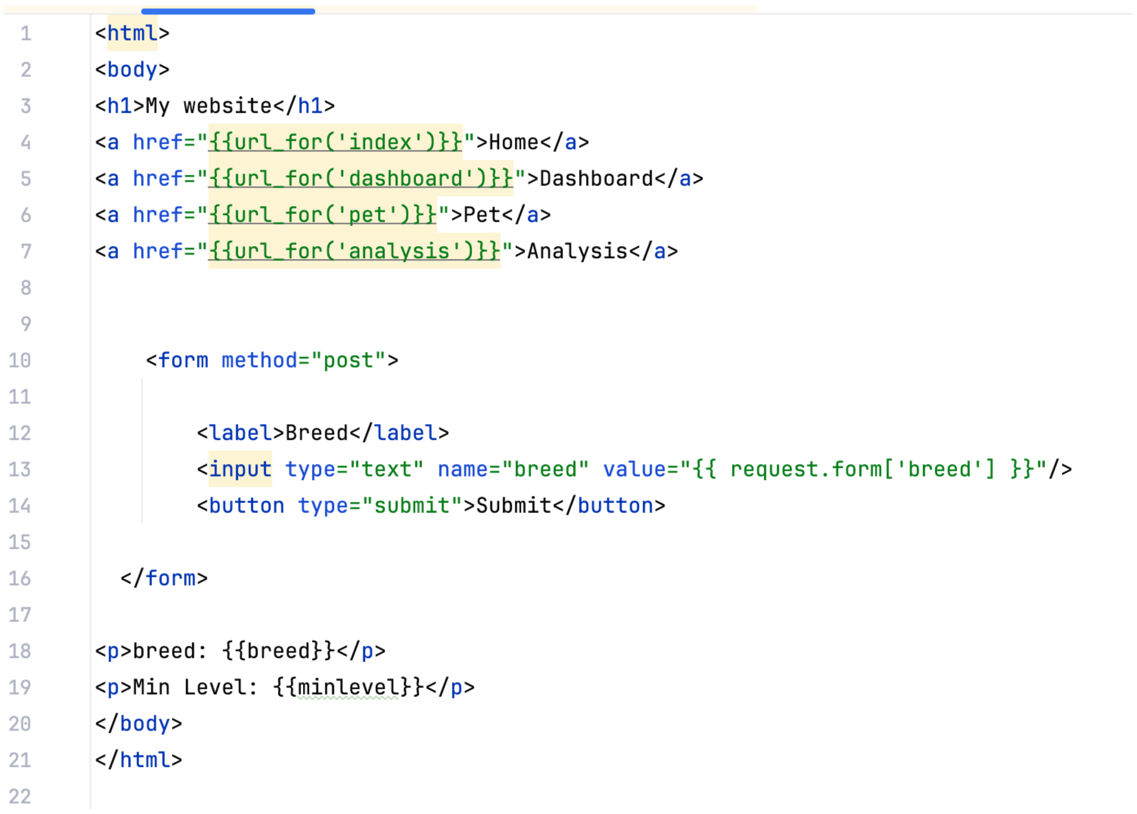


```python
from flask import Flask, render_template, request,jsonify
from analysis import *
from pet import Pet
import pandas as pd
import matplotlib.pyplot as plt
import csv
from datetime import datetime


app = Flask(__name__)
data = ['','']
mypet = Pet()
```

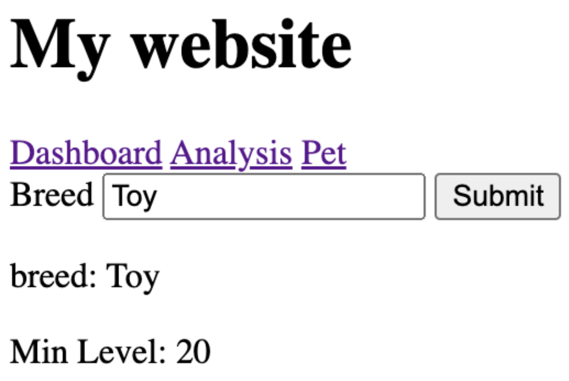


```python
@app.route('/pet', methods=('GET', 'POST'))
def pet():

    if request.method == 'POST':
        breed = request.form.get('breed')

        mypet.setBreed(breed)
        breed = mypet.getBreed()
        minLevel = mypet.calculate()
        return render_template('pet.html', breed=breed, minlevel=minLevel)

    return render_template('pet.html')
```

Pet.html



```html
<html>
<body>
<h1>My website</h1>
<a href="{{url_for('index')}}">Home</a>
<a href="{{url_for('dashboard')}}">Dashboard</a>
<a href="{{url_for('pet')}}">Pet</a>
<a href="{{url_for('analysis')}}">Analysis</a>


    <form method="post">

        <label>Breed</label>
        <input type="text" name="breed" value="{{ request.form['breed'] }}"/>
        <button type="submit">Submit</button>

  </form>

<p>breed: {{breed}}</p>
<p>Min Level: {{minlevel}}</p>
</body>
</html>
```



My website

Dashboard Analysis Pet
Breed Toy Submit

breed: Toy

Min Level: 20

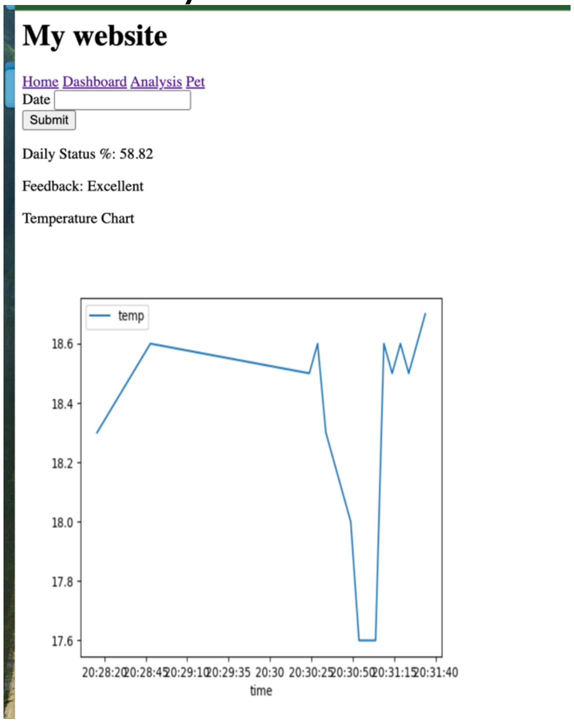
Flask code for the analysis web page:

app.py

```python
from flask import Flask, render_template, request,jsonify
from analysis import *
from pet import Pet
```

```python
@app.route( rule: '/analysis', methods=('GET', 'POST'))
def analysis():

    if request.method == 'POST':
        date = request.form.get('date')

        df = readCSV()
        filteredDF = filterByDate(df,date)
        tempChart(filteredDF)
        activityChart(filteredDF)

        percent = statusPercentage(filteredDF)
        feedback = mypet.feedback(percent)

        return render_template( template_name_or_list: 'analysis.html', tempchart='static/images/tempchart.png',
                               activitychart='static/images/activitychart.png',
                               percent=percent,
                               feedback=feedback)
    return render_template('analysis.html')
```

HTML - analysis.html



My website

Home Dashboard Analysis Pet

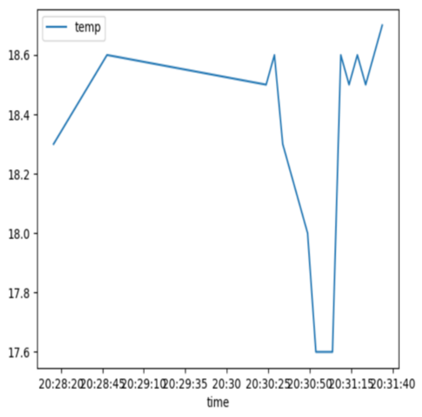
Date

Submit

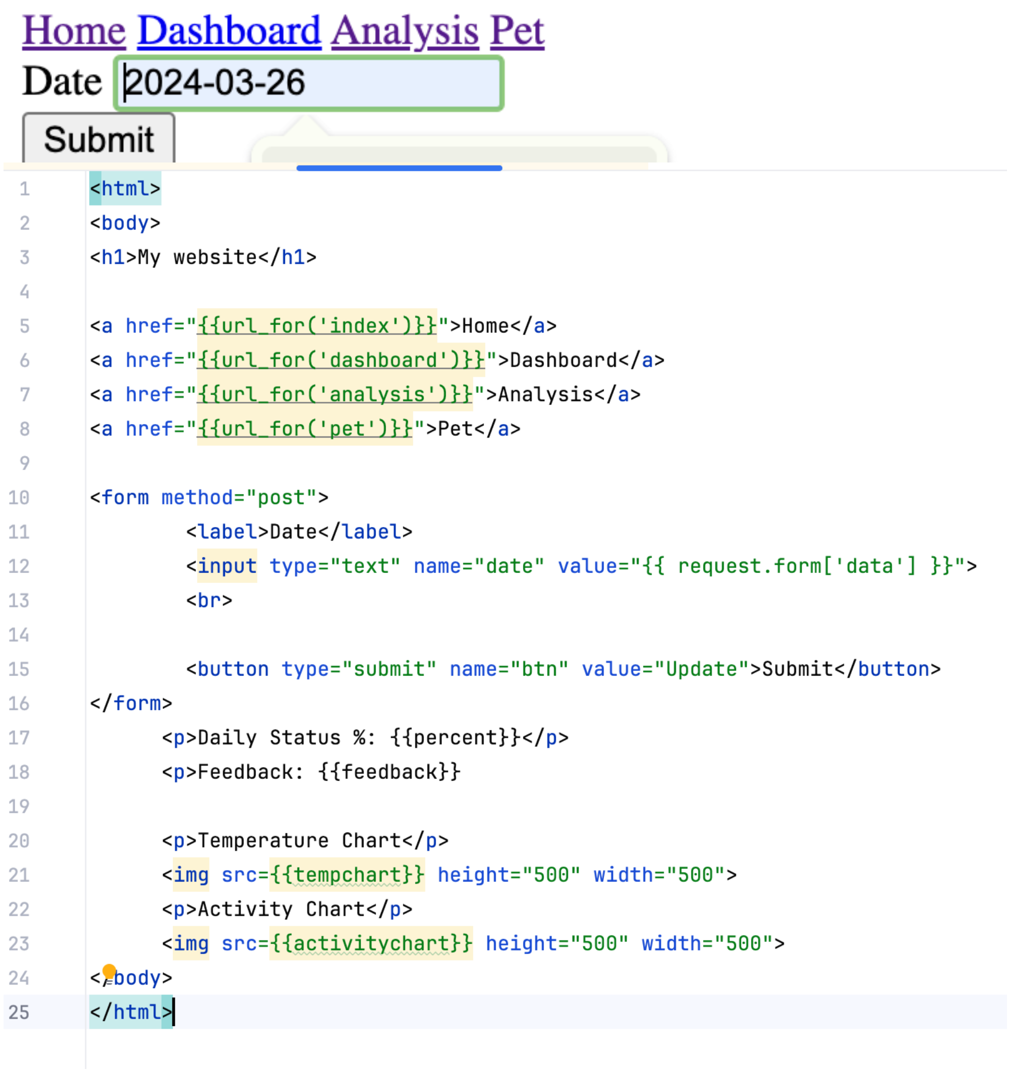
Daily Status %: 58.82

Feedback: Excellent

Temperature Chart



Enter a date that is in the CSV file, for example:

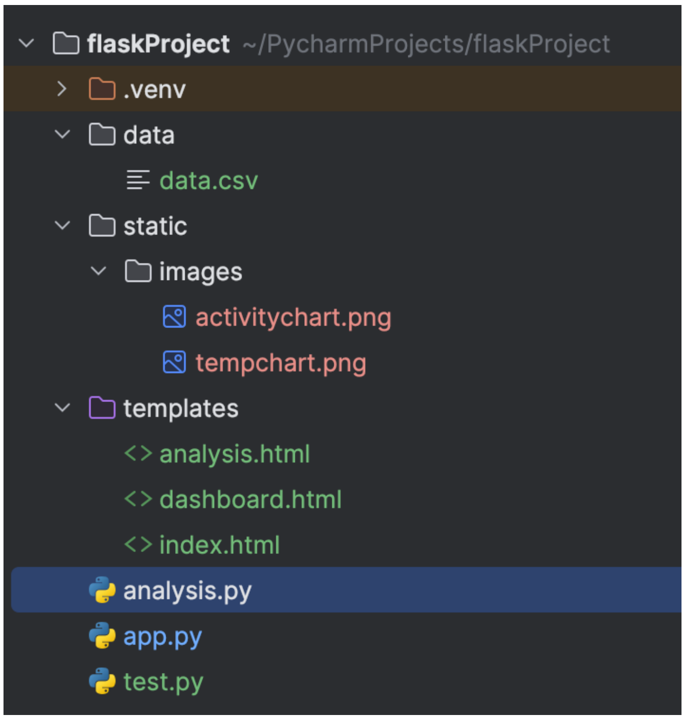


```
<html>
<body>
<h1>My website</h1>

<a href="{{url_for('index')}}">Home</a>
<a href="{{url_for('dashboard')}}">Dashboard</a>
<a href="{{url_for('analysis')}}">Analysis</a>
<a href="{{url_for('pet')}}">Pet</a>

<form method="post">
        <label>Date</label>
        <input type="text" name="date" value="{{ request.form['data'] }}">
        <br>

        <button type="submit" name="btn" value="Update">Submit</button>
</form>
      <p>Daily Status %: {{percent}}</p>
      <p>Feedback: {{feedback}}

      <p>Temperature Chart</p>
      <img src={{tempchart}} height="500" width="500">
      <p>Activity Chart</p>
      <img src={{activitychart}} height="500" width="500">
</body>
</html>
```

Final app code

```
from flask import Flask, render_template, request,jsonify
from analysis import *
from pet import Pet
import pandas as pd
import matplotlib.pyplot as plt
import csv
from datetime import datetime


app = Flask(__name__)
data = ['','']
mypet = Pet()
@app.route('/')
def index():
    return render_template('index.html')

@app.route('/pi', methods=['POST'])
def pi():
    # getData
    pico_data = request.json
    print(f'Value on server {pico_data}')
    data[0] = pico_data['status']
    data[1] = pico_data['temp']

    return 'done'

@app.route('/dashboard')
def dashboard():

    return render_template('dashboard.html')

@app.route('/analysis', methods=('GET', 'POST'))
def analysis():

    if request.method == 'POST':
        date = request.form.get('date')

        df = readCSV()
        filteredDF = filterByDate(df,date)
        tempChart(filteredDF)
        activityChart(filteredDF)
        percent = statusPercentage(filteredDF)
        feedback = mypet.feedback(percent)

        return render_template('analysis.html',
tempchart='static/images/tempchart.png',

activitychart='static/images/activitychart.png',
                               percent=percent,
                               feedback=feedback)
    return render_template('analysis.html')

@app.route('/pet', methods=('GET', 'POST'))
def pet():

    if request.method == 'POST':
        breed = request.form.get('breed')
        mypet.setBreed(breed)
        breed = mypet.getBreed()
        minLevel = mypet.calculate()
        return render_template('pet.html', breed=breed,
minlevel=minLevel)

    return render_template('pet.html')

#Get data
@app.get('/update')
def update():

    return jsonify(temp=data[0],status=data[1])


if __name__ == '__main__':
    app.run(debug=True, port="80", host='0.0.0.0')
```

File Structure



```
flaskProject ~/PycharmProjects/flaskProject
  .venv
  data
    data.csv
  static
    images
      activitychart.png
      tempchart.png
  templates
    analysis.html
    dashboard.html
    index.html
  analysis.py
  app.py
  test.py
```