

Software Degree

FULL STACK DEVELOPER

PAGE 01

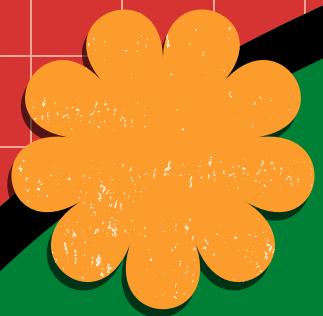
Portfolio



Gina Rubik



Morgan Stanley



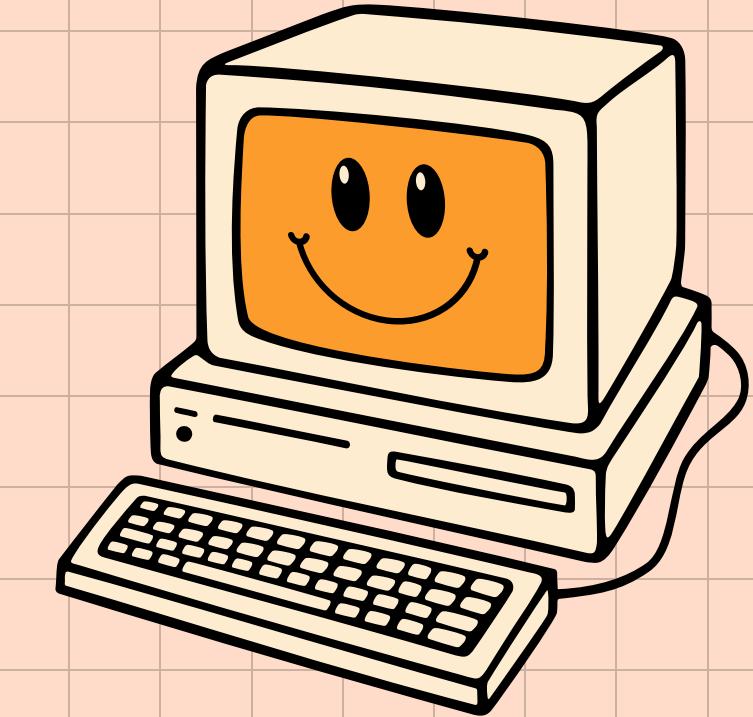
Introduction

"I know that two & two make four—and should be glad to prove it too if I could—though I must say if by any sort of process I could convert 2 & 2 into five it would give me much greater pleasure." —LORD BYRON 1813 letter to his future wife Annabella. Their daughter Ada Lovelace was the first programmer



About Me

PAGE 03



ginarubik - Overview

Code First Girls Software Development Full Stack
Creativity and Innovation - ginarubik



Welcome to my Visionary Portfolio! 🌟 My main interests are: full stack innovation, creative coding and sustainable solutions to problem-solving. 🌱 Trained by CFG in software engineering and web development. 🌈

<https://github.com/ginarubik>

CS Education

APRIL
2024-

Code First Girls

Intro to Coding
Intro to Python
Intro to Java
Intro to C #
Intro to Golang
Intro to Machine Learning

AUGUST
2023

MOOC Courses

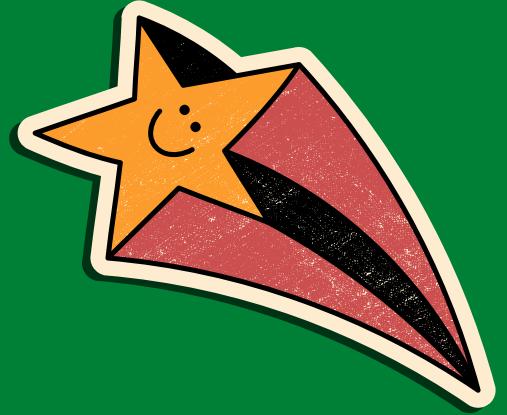
WebDevelopment
Interactive
Websites
Agile&Scrum
Git&Github

JAN 2024

Morgan Stanley

Software Degree:
Javascript
HTML/SQL
Python&Apps

Software Skills



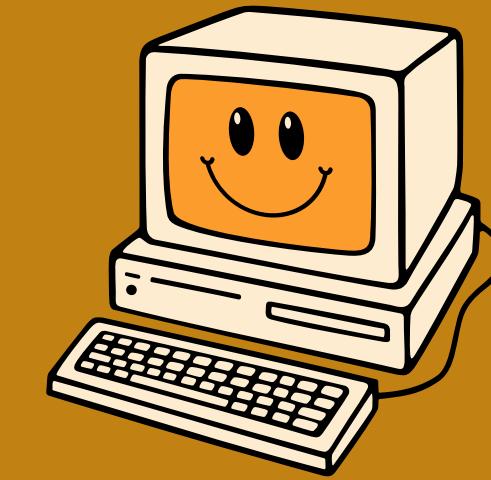
Front-End:
*Javascript, HTML, CSS
*Animation Design
*GIU
*Codepen.io
*OBS Studio
*Carbon Source Code Images

Back-End: Python3
PyMongo*Pandas*Flask*SQL
Jinja*Textblob*Plotly*MongoDB
Software Tools:Visual Studio Code*Visual Studio*Pycharm*Agile Scrum*Jira*Figma Unittesting*UAT Atlassian*Kanban*Debugging GIF API*Restful API
Webhooks*Zapier Dev Platform Git,Github Desktop*Canva Python Turtle*Tkinter Codewars*Hackerrank Google Colaboratory*InelliJ

IT Career Interests

- *AI Innovation&Research
- *Corporate Application Development
- *FullStack&Web Development
- *Project Management
- *Product Design
- *Education for Inclusion
- *Programming Documentation
- Finance and Wealth Management
- Health&Sustainability

Work Experience:



Content&Tutorial
Writing

Visual&Audio
Branding

Web&Logo
Design

Special Skills

- Studio Binder, Moho
- codepen.io JS, CSS, HTML
- flexbox, keyframes, left pill, pill text, fill mode

Animations:



Film Drawings

Roll the Dice

Pill Shape



Recipe APIs

A food recipe search program by ingredients, kcalories and weight, entered by user, interacting with Edamam and Tasty APIs, saved in Jupyter Notebook to document coding workflow

<https://github.com/ginaru/bik/Recipe-APIs>

```
D:\CFGDegree\Foundation1\Python\Intro to Python>python "Recipe APIs.py"
What ingredient should be in recipes? chicken
>> Recipe data stored in recipes.txt.
>> Recipe data stored in sorted_recipes.txt.
Approximately how much kcal must be in dish (type max amount): 100
Sorry, no dish found according to your parameters, please change your search parameters.
Data written to 'nutritionApiData.json' file.

Search results from Tasty API:
Low-Carb Avocado Chicken Salad
Total Calories: 250

Creamy Chicken Penne Pasta
Total Calories: 703

Asparagus-Stuffed Chicken Breast
Total Calories: 209

One-Pot Chicken Fajita Pasta
Total Calories: 862

Chicken & Veggie Stir-Fry
Total Calories: 272

Paprika Chicken & Rice Bake
Total Calories: 313

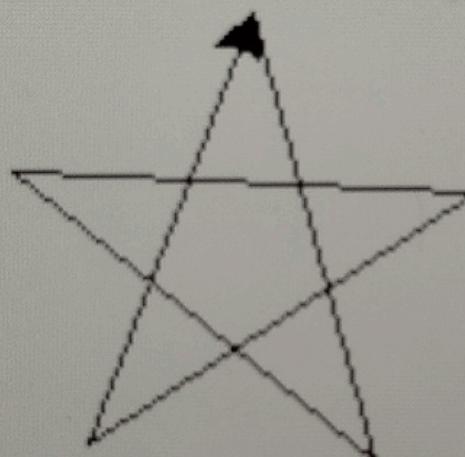
Hasselback Chicken
Total Calories: 332

Easy Butter Chicken
Total Calories: 799
```

Group Project 1

Project 2

Draw Shapes



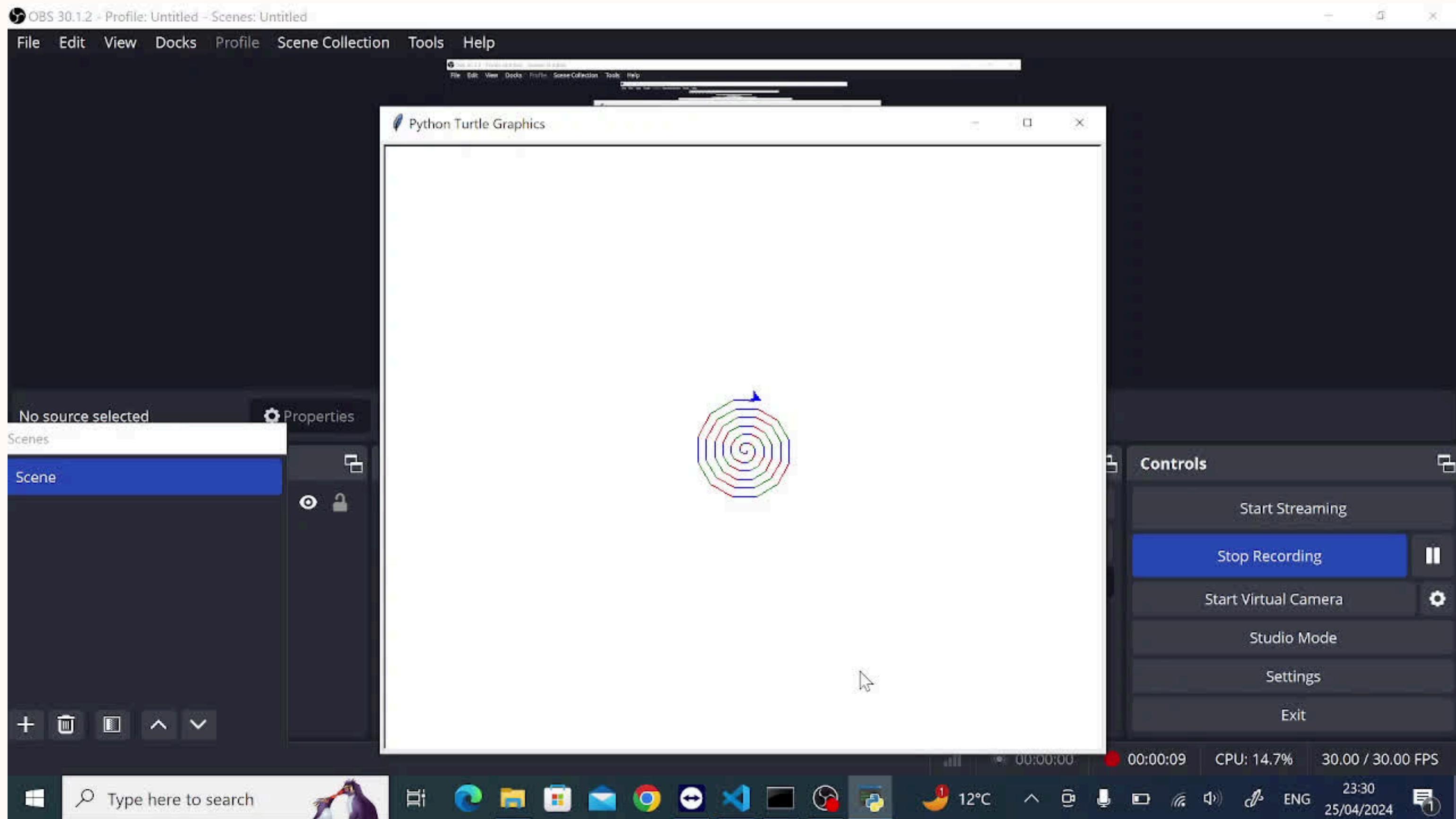
File Edit Selection View Go Run ... ← →

Restricted Mode is intended for safe code browsing. Trust this window to enable all features. [Manage](#)

draw star.py X

```
C: > Users > HP > Documents > CFG > draw star.py > ...
1 # Python program to draw star
2 # using Turtle Programming
3 import turtle
4 star = turtle.Turtle()
5
6 star.right(75)
7 star.forward(100)
8
9 for i in range(4):
10     star.right(144)
11     star.forward(100)
12
13 turtle.done()
```

- Python Turtle
Graphics -



File Edit Selection View Go Run ... ← → Search

□ □ □ □ -

draw_star.py create_shape.py star.py X

...

C:\> Users > HP > Documents > CFG > MOOC Intro to Python > star.py > ...

```
4     size = 99
5
6     arrow.color('red')
7     ### For creating a circle
8     for i in range(36):
9         arrow.right(10)
10        arrow.forward(9)
11
12    arrow.color('purple')
13    arrow.right(77)
14    # For creating a triangle
15    arrow.forward(size)
16    arrow.right(144)
17    arrow.forward(size)
18    arrow.right(144)
19    arrow.forward(size)
20    arrow.right(144)
21    arrow.forward(size)
22    arrow.right(144)
23    arrow.forward(size)
24
25    arrow.color('blue')
26    arrow.right(67)
27    arrow.forward(48)
28    arrow.right(90)
29    arrow.forward(104)
30    arrow.right(90)
31    arrow.forward(104)
32    arrow.right(90)
33    arrow.forward(104)
```

...

⊗ 0 Δ 0 ⌂ 0 ↻

Ln 4, Col 10 Spaces: 4 UTF-8 CRLF { Python 3.12.3 64-bit



Type here to search



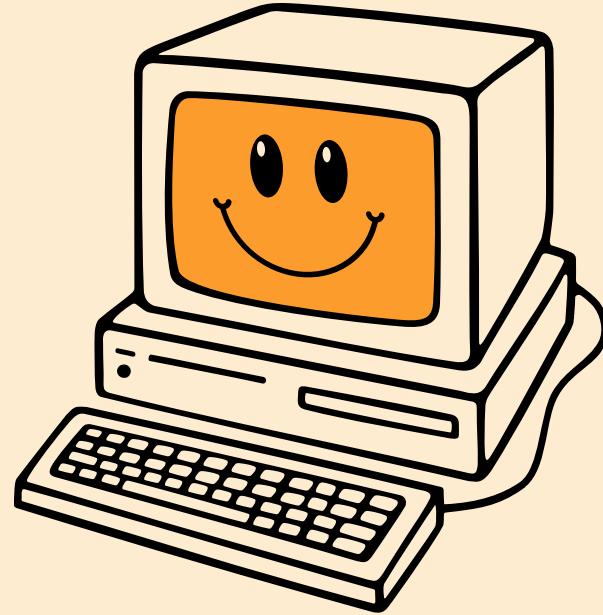
24°C



19:47

14/04/2024

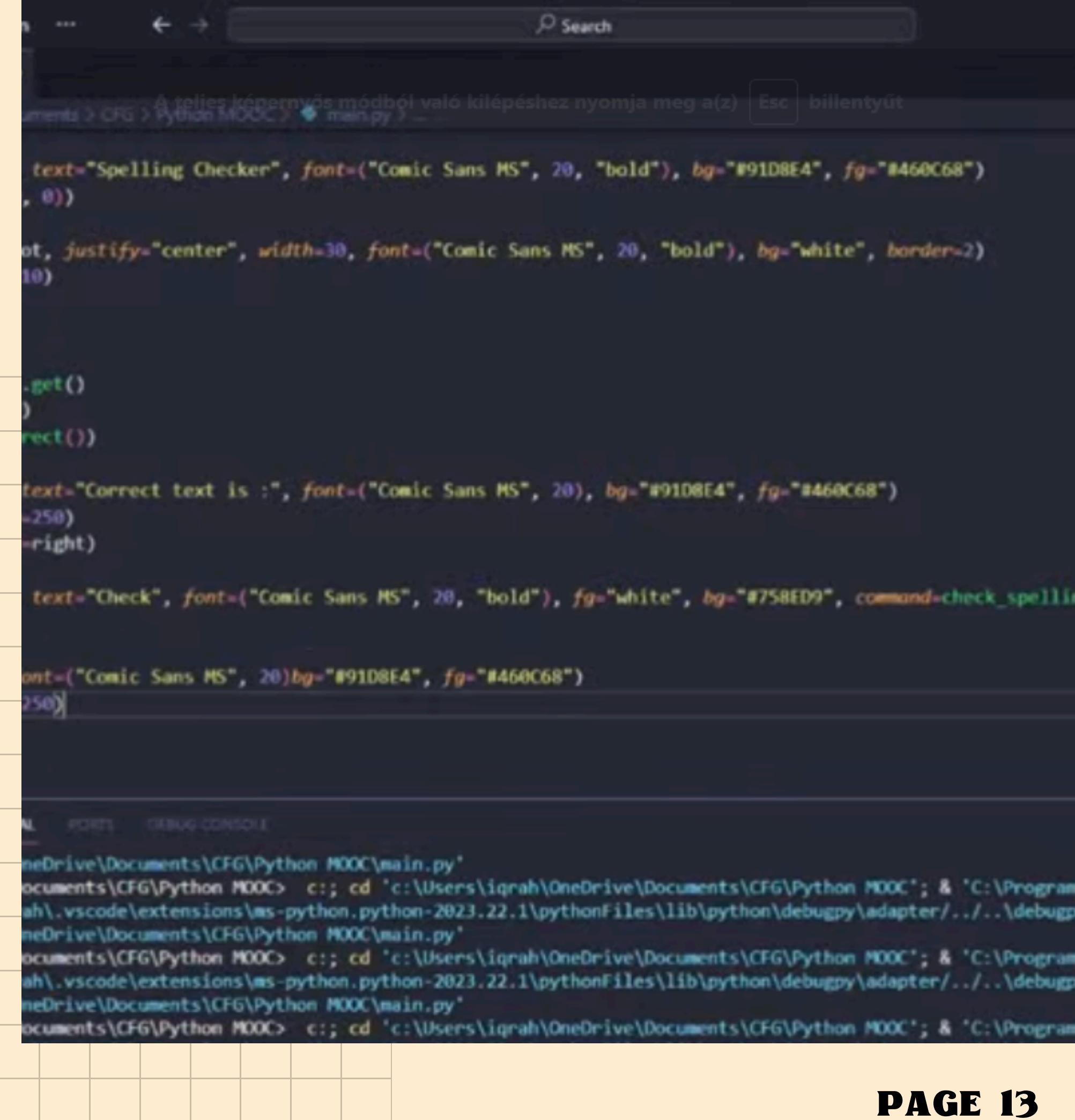
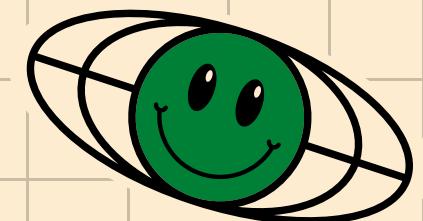
calculator Spell checker Simple Games



Python Tkinter, TextBlob and PIL

<https://github.com/ginarubik/SpellingChecker>
<https://github.com/ginarubik/calculator>

Project 3



A screenshot of a code editor window showing Python code for a graphical user interface (GUI) application. The code uses the Tkinter library and includes imports for `tk` and `ttk`. It defines a class `SpellingChecker` with methods like `check_spelling` and `check_text`. The code also includes a main loop and a configuration section at the bottom.

```
import tkinter as tk
from tkinter import ttk
from textblob import TextBlob
from PIL import Image, ImageTk
import os

class SpellingChecker:
    def __init__(self, master):
        self.master = master
        self.master.title("Spelling Checker")
        self.master.geometry("400x300")

        self.create_widgets()

    def create_widgets(self):
        self.text_label = tk.Label(self.master, text="Text to check:", font=("Comic Sans MS", 16))
        self.text_label.pack(pady=10)

        self.text_entry = tk.Text(self.master, height=10, width=40)
        self.text_entry.pack()

        self.check_button = tk.Button(self.master, text="Check", font=("Comic Sans MS", 16, "bold"), fg="white", bg="#758ED9", command=self.check_spelling)
        self.check_button.pack(pady=10)

        self.result_label = tk.Label(self.master, text="Correct text is : ", font=("Comic Sans MS", 16))
        self.result_label.pack(pady=10)

        self.result_text = tk.Text(self.master, height=10, width=40)
        self.result_text.pack()

    def check_spelling(self):
        text = self.text_entry.get("1.0", "end-1c")
        blob = TextBlob(text)
        corrected_text = blob.correct()
        self.result_text.delete("1.0", "end-1c")
        self.result_text.insert("1.0", corrected_text)

    def check_text(self):
        text = self.text_entry.get("1.0", "end-1c")
        blob = TextBlob(text)
        corrected_text = blob.correct()
        self.result_text.delete("1.0", "end-1c")
        self.result_text.insert("1.0", corrected_text)

    def on_closing(self):
        if tk.messagebox.askokcancel("Exit", "Are you sure you want to exit?"):
            self.master.destroy()

    def run(self):
        self.master.mainloop()

if __name__ == "__main__":
    root = tk.Tk()
    app = SpellingChecker(root)
    app.run()
```

The code editor interface shows the file path as `OneDrive\Documents\CFG\Python MOOC\main.py` and the status bar indicates `PORTS DEBUG CONSOLE`.



Magic Spell!

Enter a word:

A text input field containing the letter 'p'.A text input field next to the first one, currently empty.

Check my spelling!

You didn't enter a word!

Project 4

IT WebApp for
Code First Girls

a user-interactive
educational web application
using HTML,CSS and
JavaScript: dice animation,
choice of 4 specialisations
and pop-up console

<https://codepen.io/Guru-GanaSaki/pen/mdamKbQ?editors=0110>

CFG Degree

Code First Girls 📚 Code First Girls 📚 Code First Girls 🎓 Code First Girls 🌟



Click

Specialisations:

1. Software Engineering 🧑
2. Full-Stack Developer 🧑
3. Data Career Pathway 🧑
4. Product Management 🧑

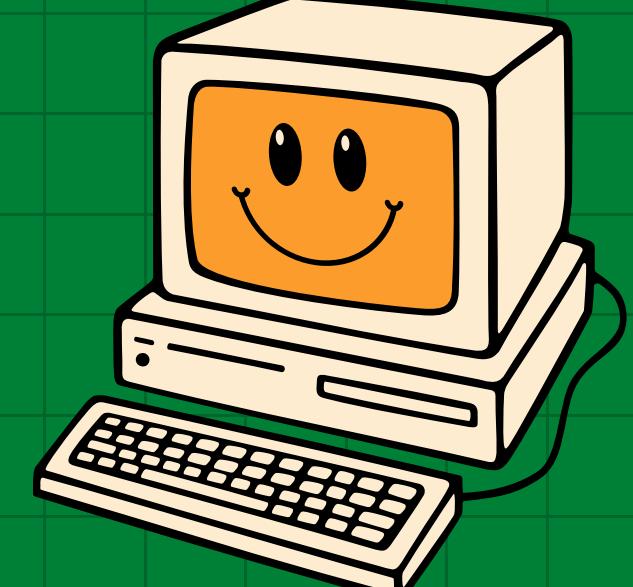
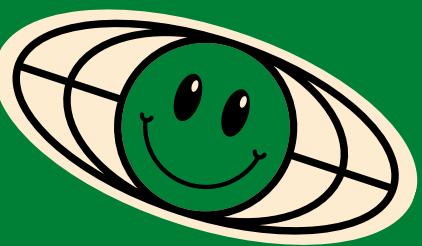
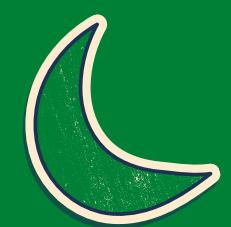
Select Your CFG Degree Track

Choose a Specialisation:

Enter Your Name



ISS International SpaceShip API



**Python console application
calculating distance from user
location, using Geopy Nominatim API
by mathematical Haversine formula
storing results in a JSON file**

https://github.com/ginarubik/ISS_SpaceShip_API

Project 5

```
Welcome to the International Space Station App!
Please enter your email: ginarubik@gmail.com
Enter a location (country or city): Hungary
The ISS is approximately 15328.60 kilometers away from Hungary
The ISS is currently flying over Russia

Do you want to check another location? ('n' to exit): Budapest
Enter a location (country or city): Budapest
The ISS is approximately 13858.49 kilometers away from Budapest
The ISS is currently flying over Russia

Do you want to check another location? ('n' to exit): yes
Enter a location (country or city): UK
The ISS is approximately 8440.45 kilometers away from UK
The ISS is currently flying over Russia

Do you want to check another location? ('n' to exit): yes
Enter a location (country or city): Hereford
The ISS is approximately 2353.74 kilometers away from Hereford
The ISS is currently flying over Russia

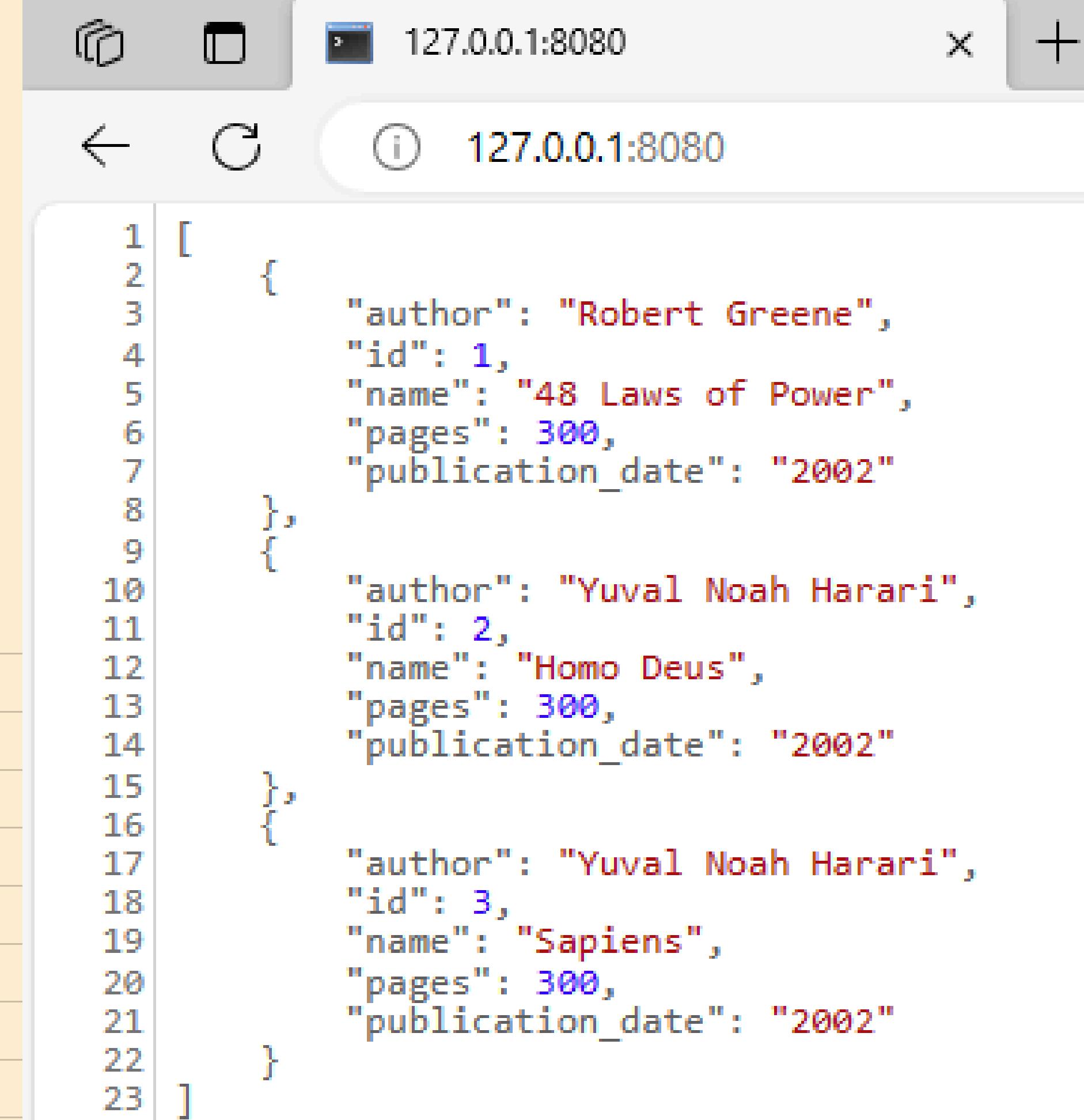
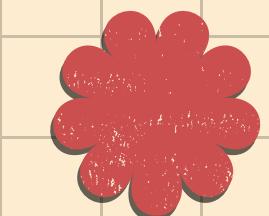
Do you want to check another location? ('n' to exit): n
Thank you for using the ISS App!
```

Project 6

Book API

- To simulate user interaction run the **main.py** for list of available books stored in MySQL database
- API handles the interactions between the client-side and the database, so books can be viewed, added, updated and deleted
- App can be expanded/customised by adding features like user authentication and additional endpoints

https://github.com/ginarubik/Book_API



A screenshot of a web browser window showing a JSON response. The URL in the address bar is 127.0.0.1:8080. The page content displays a list of three books in JSON format, each with an id, author, name, pages, and publication_date. The JSON is color-coded for readability.

```
1 [  
2 {  
3   "author": "Robert Greene",  
4   "id": 1,  
5   "name": "48 Laws of Power",  
6   "pages": 300,  
7   "publication_date": "2002"  
8 },  
9 {  
10   "author": "Yuval Noah Harari",  
11   "id": 2,  
12   "name": "Homo Deus",  
13   "pages": 300,  
14   "publication_date": "2002"  
15 },  
16 {  
17   "author": "Yuval Noah Harari",  
18   "id": 3,  
19   "name": "Sapiens",  
20   "pages": 300,  
21   "publication_date": "2002"  
22 }]  
23 ]
```

Group Project 7

Cinema Ticketing API

MySQL database stores information about movie titles, age restrictions, showtimes, prices, seat numbers and user names. The three API endpoints are for retrieving movie showings by age restrictions, adding new bookings, and updating existing bookings. The cinema_config module contains configuration information for connecting to the MySQL database

https://github.com/ginarubik/CineTix_API

```
← C ⓘ 127.0.0.1:8080  
[  
1  {  
2      "Age_restrictions": 16,  
3      "Movie_id": 1,  
4      "Movie_title": "57 Seconds",  
5      "Price": "5.50",  
6      "Seat_number": "17C",  
7      "Time": "20:30:00",  
8      "User_name": "John Doe"  
9  },  
10  {  
11      "Age_restrictions": 16,  
12      "Movie_id": 2,  
13      "Movie_title": "A Haunting in Venice",  
14      "Price": "5.99",  
15      "Seat_number": "11A",  
16      "Time": "20:00:00",  
17      "User_name": "Jane Smith"  
18  },  
19  {  
20      "Age_restrictions": 16,  
21      "Movie_id": 3,  
22      "Movie_title": "DogMan",  
23      "Price": "3.50",  
24      "Seat_number": "9B",  
25      "Time": "20:15:00",  
26      "User_name": "Mike Johnson"  
27  },  
28  {  
29      "Age_restrictions": 18,  
30      "Movie_id": 4,  
31      "Movie_title": "The Exorcist: Believer",  
32      "Price": "6.99",  
33      "Seat_number": "13F",  
34      "Time": "19:50:00",  
35      "User_name": "Emily Williams"  
36  },  
37  {  
38      "Age_restrictions": 16,  
39      "Movie_id": 5,  
40      "Movie_title": "The Nun II",  
41      "Price": "4.50",  
42      "Seat_number": "14D",  
43      "Time": "22:15:00",  
44      "User_name": "David Brown"  
45  },  
46  {  
47      "Age_restrictions": 18,  
48      "Movie_id": 6,  
49      "Movie_title": "The Equalizer 3"  
50  }
```

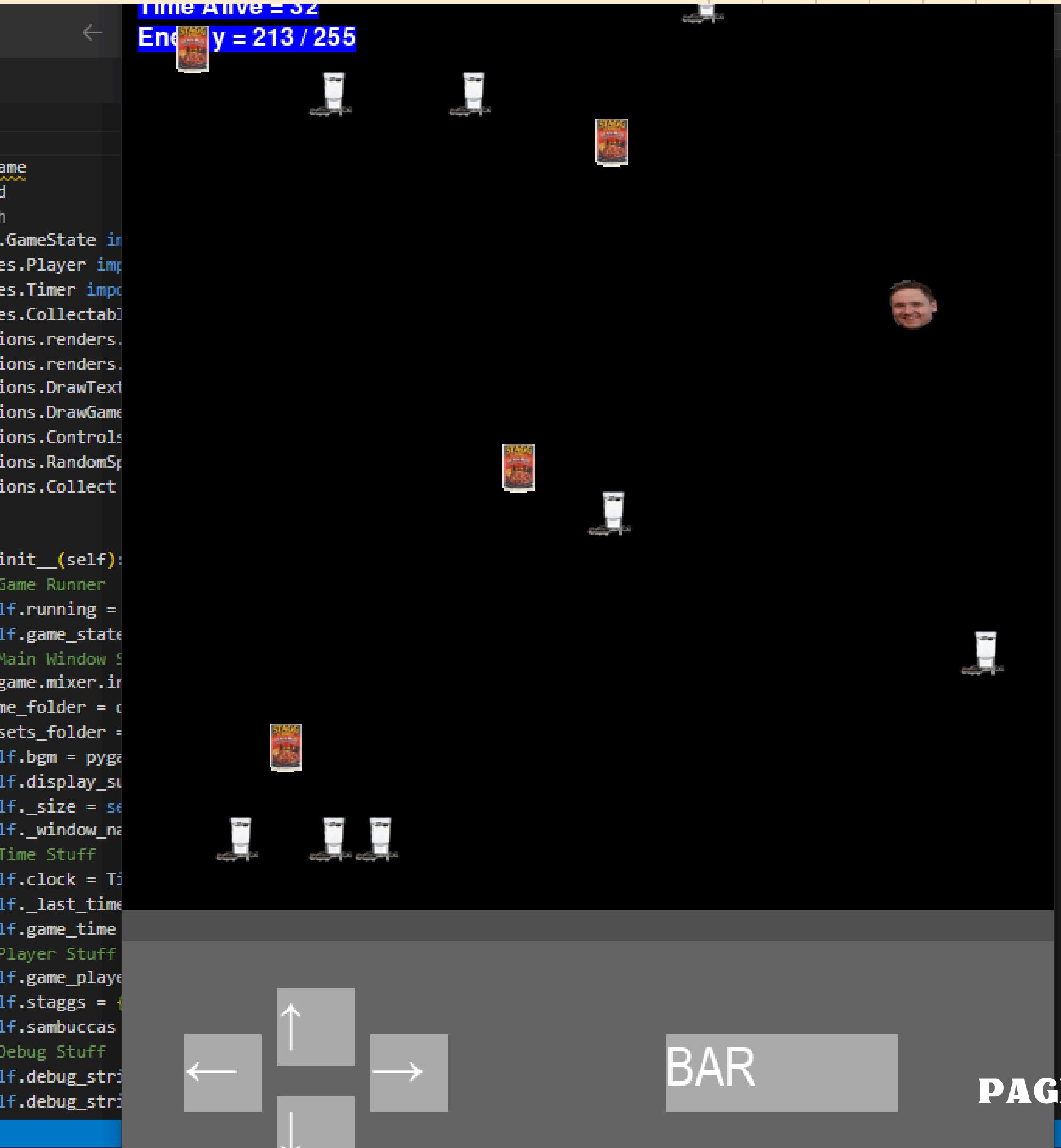
Project 8

Python Game

Py game with a simple menu:

- **when the player starts, the game_state switches to 'PLAY'. user controls a character to move around.**
- **the player can have an energy bar that decreases over time. collecting 'Stagg' items might increase energy, while 'Sambucca' items might decrease it**
- **consider writing unit tests (especially to refactor code into smaller functions) to ensure changes don't break existing functionality**

https://github.com/ginarubik/Game_Development



BAR

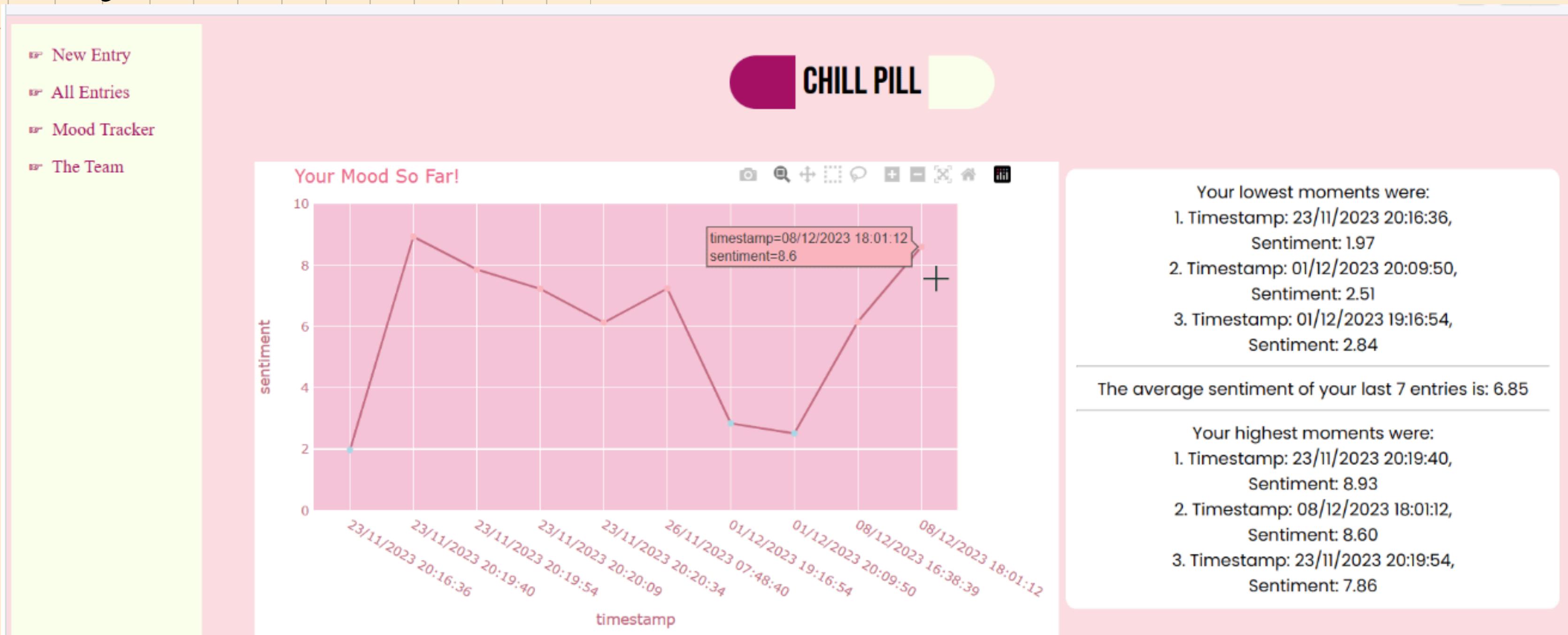
PAGE 18

Group Project 9

ChillPill Wellness Website

user journal entries of daily moods evaluated by AI-based API sentiment analysis and GIF affirmations displayed on mood-tracker graphs with pill animation creativity and team members' page.

https://github.com/ginarubik/Wellness_Website



B 2 Aea 698692 A 515 A 56 C 1 X • Chill Pill +

127.0.0.1:5000/submission?result=New+journal+entry+added!&id=657c72f12e6e4da91ff106ca&da 80% ⭐

New Entry

All Entries

Mood Tracker

The Team

CHILL PILL

JOURNAL ENTRY

15/12/2023, 15:38:25

I am feeling great today, I feel happy. I had a nice day and a nice cup of coffee. CFG classes were very interesting today and my project team were very helpful. We met our goals today.

SUBMIT

[Source Code](#)

127.0.0.1

Computer Tutorials

Python Foundation & Specialisation Topics

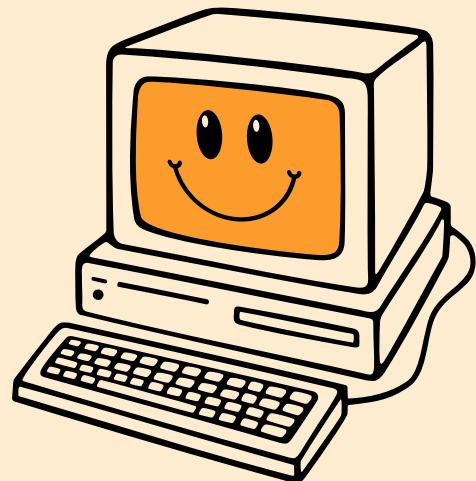
Problem Solving Logic

Developer Ethics

Program Documentation

https://github.com/ginarubik/Computer_Tutorials

Project 10



protected method (not recommended)
alert() # Output: Maintenance Alert: Schedule a service

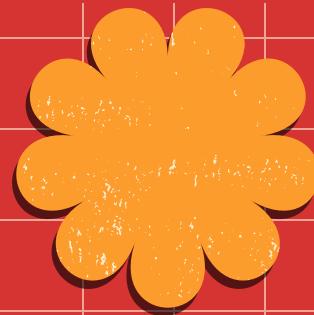
re: allows a class (subclass/derived class) to inherit attributes and methods from another class.
reusability and establishes an "is-a" relationship between classes.

```
class ElectricCar(Car):  
    def __init__(self, brand, model, battery_capacity):  
        super().__init__(brand, model)  
        self.battery_capacity = battery_capacity  
  
my_car = ElectricCar("Tesla", "Model S", "100 kwh")  
my_car.display_info() # Output: Tesla Model S
```

ism: allows objects of different types to be treated as if they were of the same type.

if):

Literature&Films

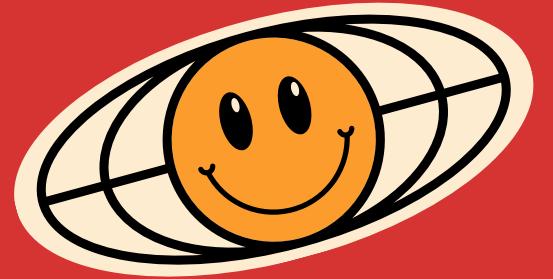


Computer Science Distilled (Ferreira Filho)
Grokking Algorithmns (Aditya Bhargava),
Introduction to APIs (Brian Cooksey)
The Daily Laws (Robert Greene)



The Man Who Knew Infinity(Mathematician
Ramanujan Biography- Feature Film)
Half Life(Documentary-Game Developers)





Thank You! Your Questions?

Connect with me!

[https://www.linkedin.com/in/gina-
rubik-25423923](https://www.linkedin.com/in/gina-rubik-25423923)

