**COIT20245**

**Introduction to Programming**

**Assignment 2**

**Project Part 2 of 3**

**CONTEXT:**

**Task 1……………………………………………………………3-4**

**Task 2…………………………………………………………….4-6**

**Task3…………………………………………………………….7-9**

**Task 4……………………………………………………………9-10**

**Task 5…………………………………………………………..11-12**

**Task 6…………………………………………………………..12-14**

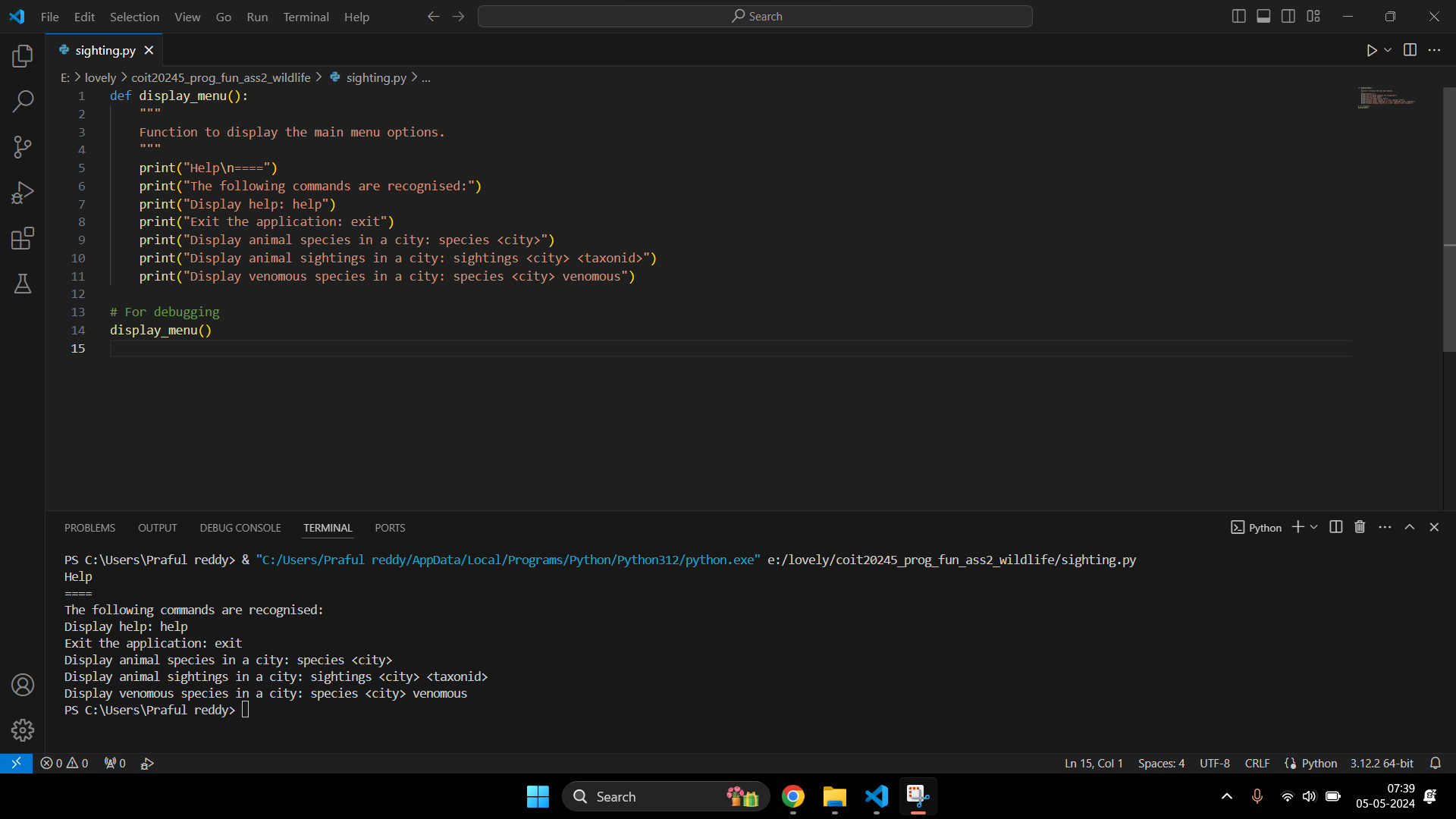
**Task 7…………………………………………………………..14-15**

**Task8……………………………………………………...……15-16**

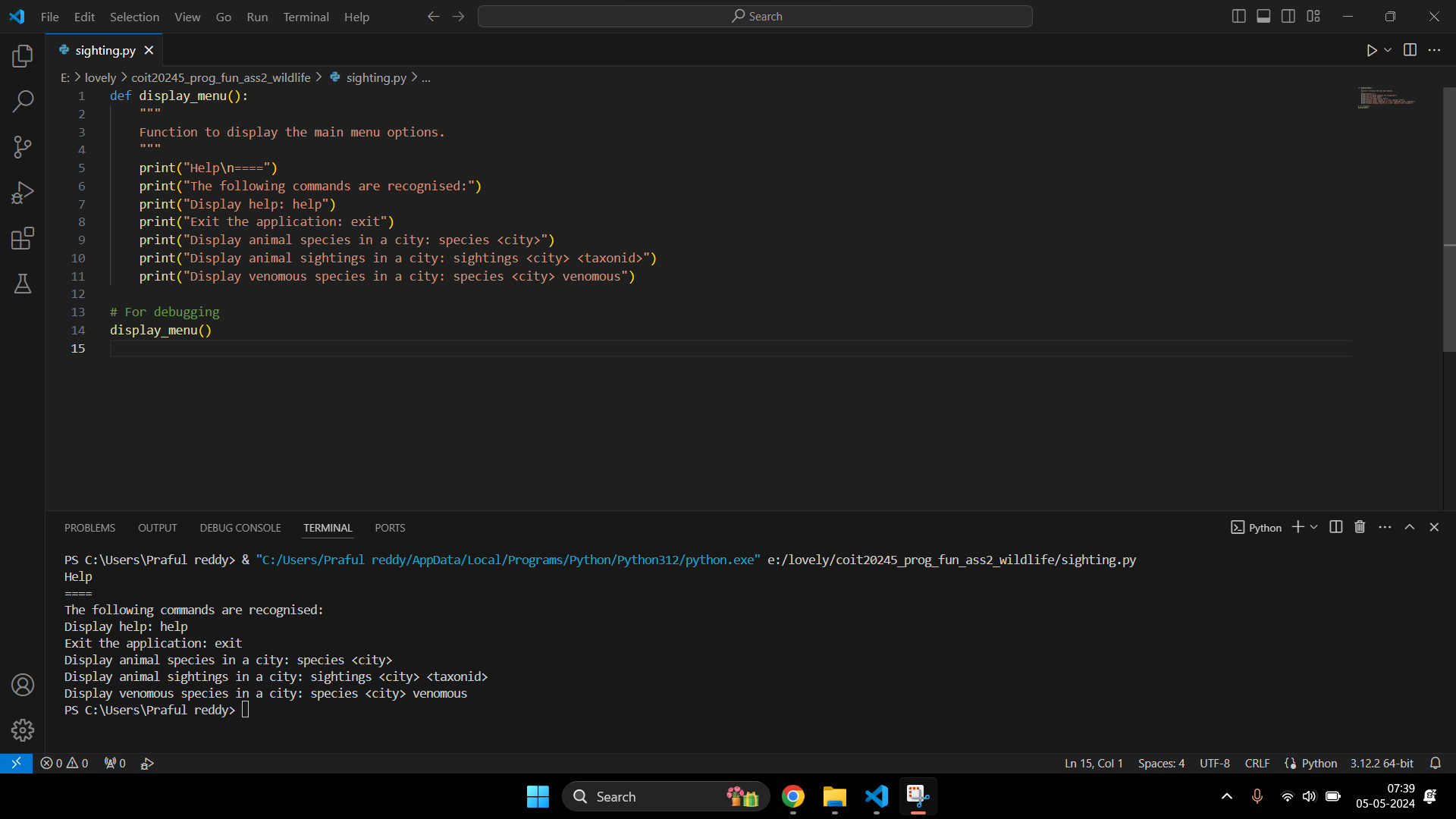
**Task1**

**Screenshots:**

The display\_menu() function that uses print to display the menu.



The display\_menu() function after debugging the code.



**About the Function used:**

> Function Name: display\_menu()

> Purpose: Function to display the main menu options.

> Parameters: None

> Returns: It returns the statements given in the ‘print()’ function.

> Exception: None.

> Example Calls:

Help

====

The following commands are recognised:

Display help: help

Exit the application: exit

Display animal species in a city: species <city>

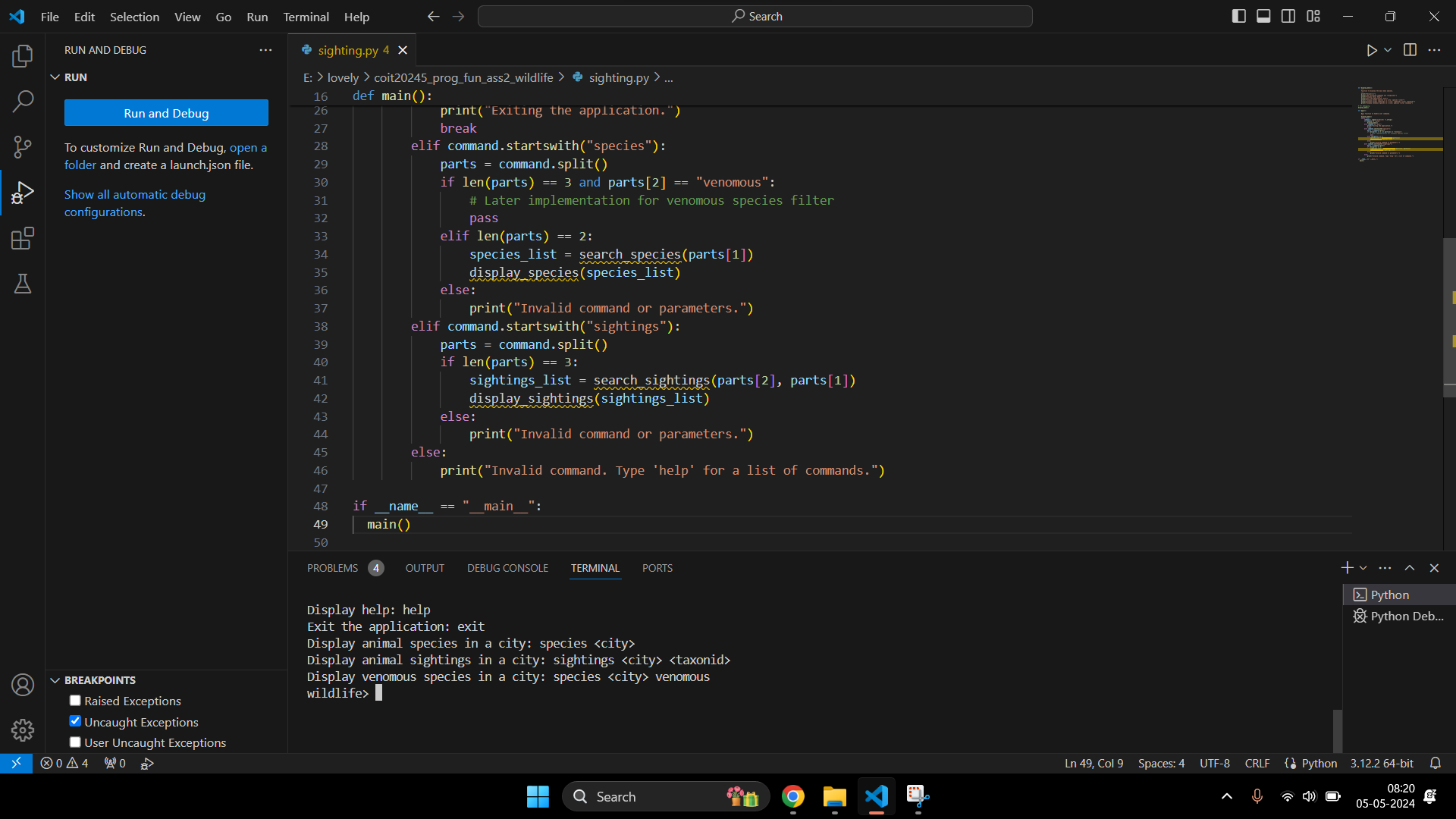
Display animal sightings in a city: sightings <city> <taxonid>

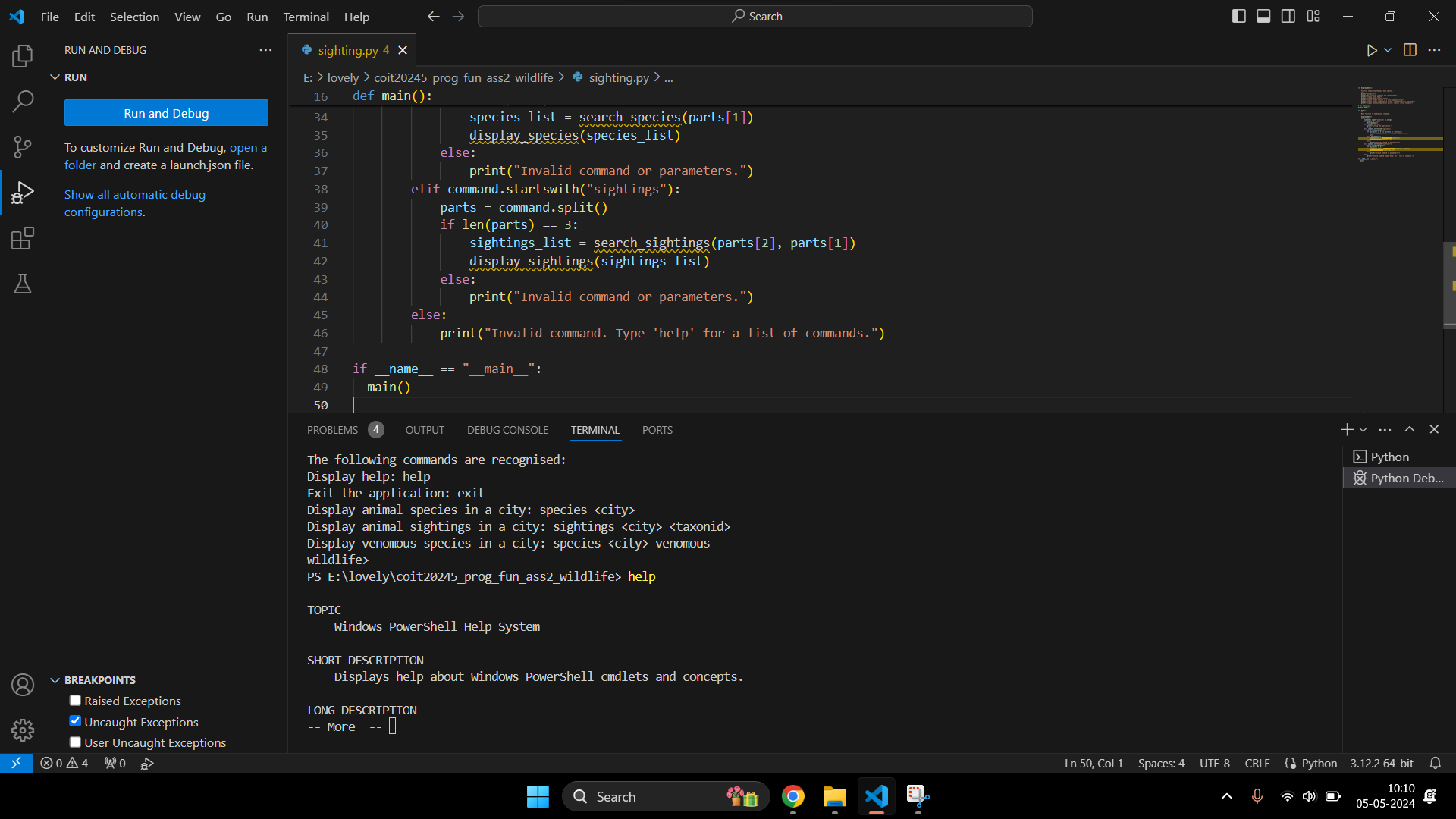
Display venomous species in a city: species <city> venomous

**Task 2**

**Screenshots:**

The main() function that displays the help menu and then repeatedly prompts the user to input their command.





Input: help

Output: Task 1 output will be displayed

Input: exit

Output: Exiting the application. (then the program terminates)

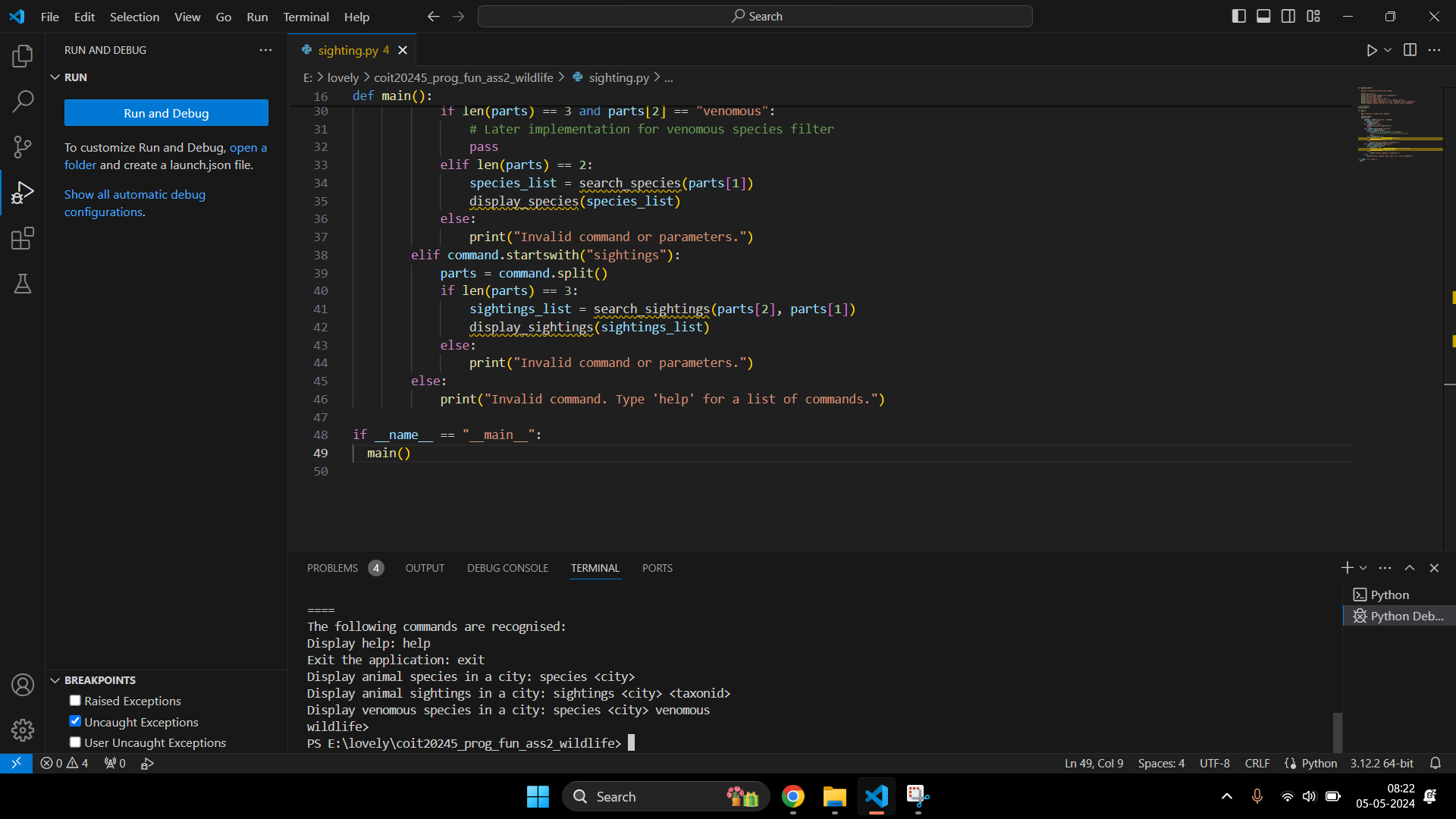
Input: species Cairns

Output: species listing will be displayed (Task3).

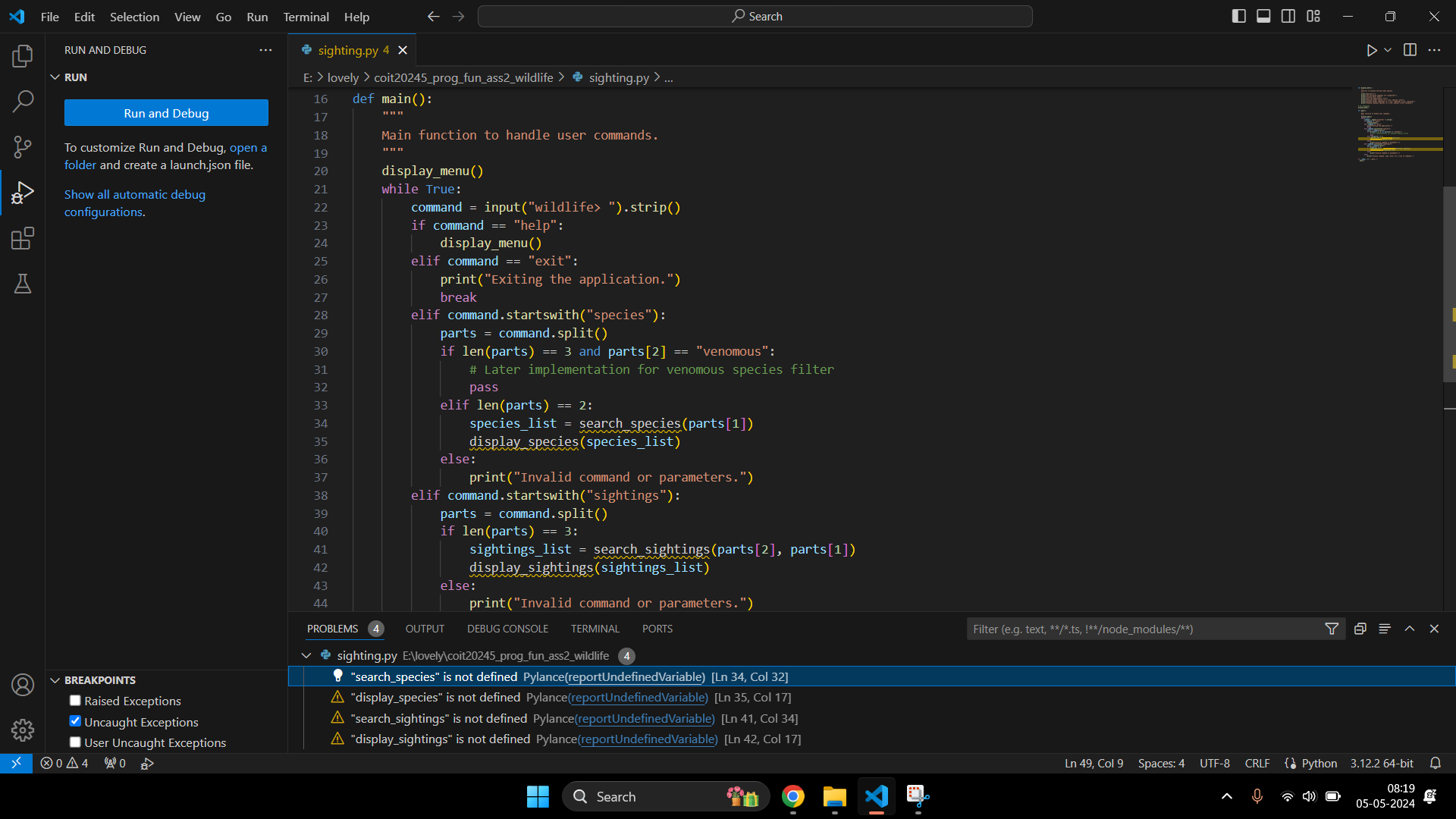
Input: sightings Cairns 1039

Output: sightings listing (Task 4)

The main() function that displays after debugging of code.



**Problems occurred:**



**About the Function used:**

> Function Name: main()

> Purpose: Function to create commands.

> Parameters: None

> Returns: None

> Exception: It excepts some on-defined functions such as search\_species, display\_species, search\_sighting and display\_sighting.

> Example Calls:

main()

wildlife> help

Displays help menu.

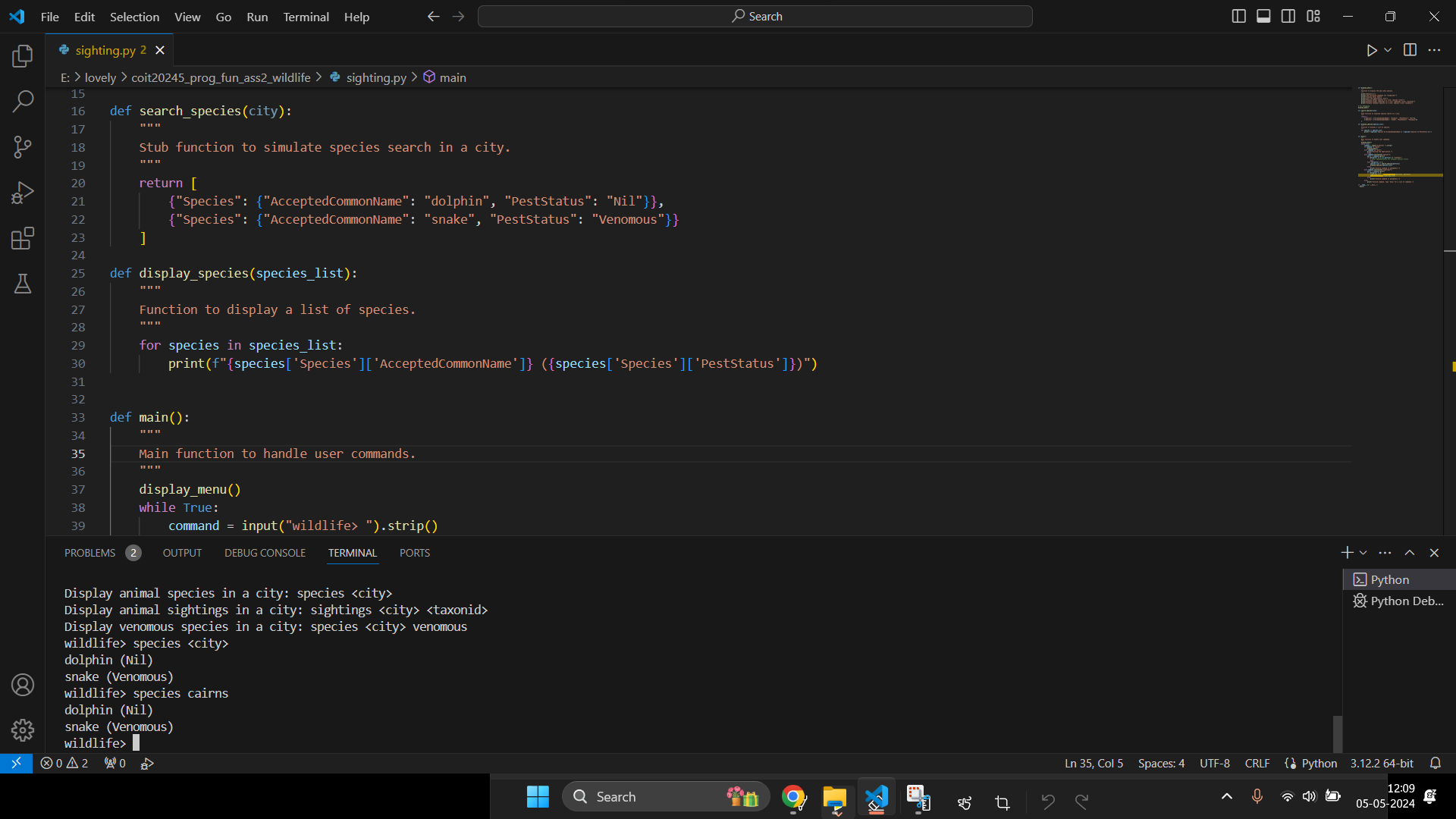
wildlife> exit

Exits the application

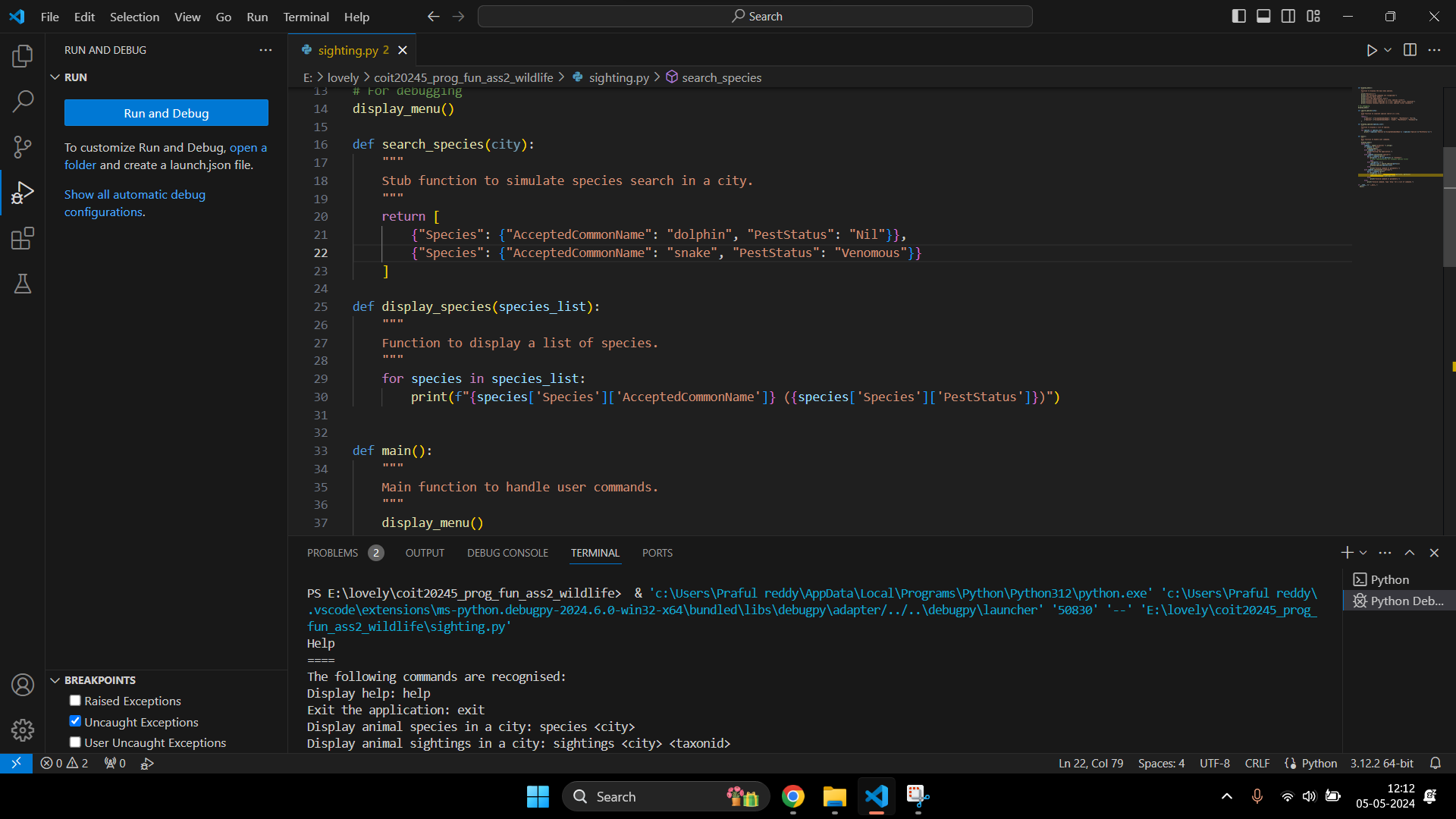
**Task 3**

**Screenshots:**

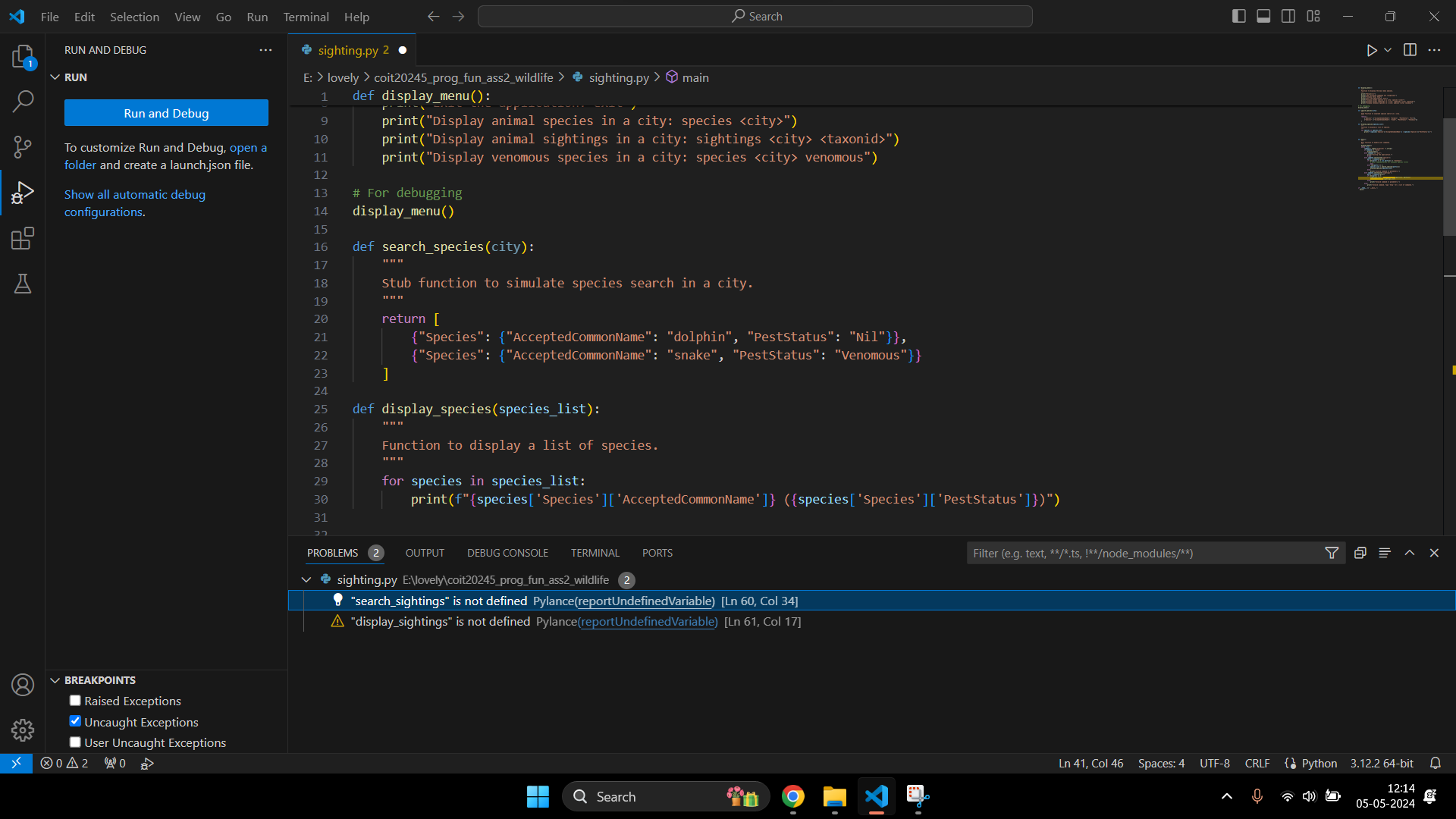
Update display\_menu() by adding a few functions.



Output after debugging the functions.



**Problems occurred:**

****

**About the Function used:**

> Function Name: search\_species(city)

> Purpose: Function to search different types of species in the city

> Parameters: city (str): The city where the species are to be searched.

> Returns: list: A list of dictionaries, each containing species information

> Exception: None

> Example Calls:

search\_species("Cairns")

[{"Species": {"AcceptedCommonName": "dolphin", "PestStatus": "Nil"}},

{"Species": {"AcceptedCommonName": "snake", "PestStatus": "Venomous"}}]

> Function Name: display\_species(city)

> Purpose: Displays a list of species to the user.

> Parameters: city (str): species\_list (list): A list of dictionaries containing species information.

> Returns: None

> Exception: None

> Example Calls:

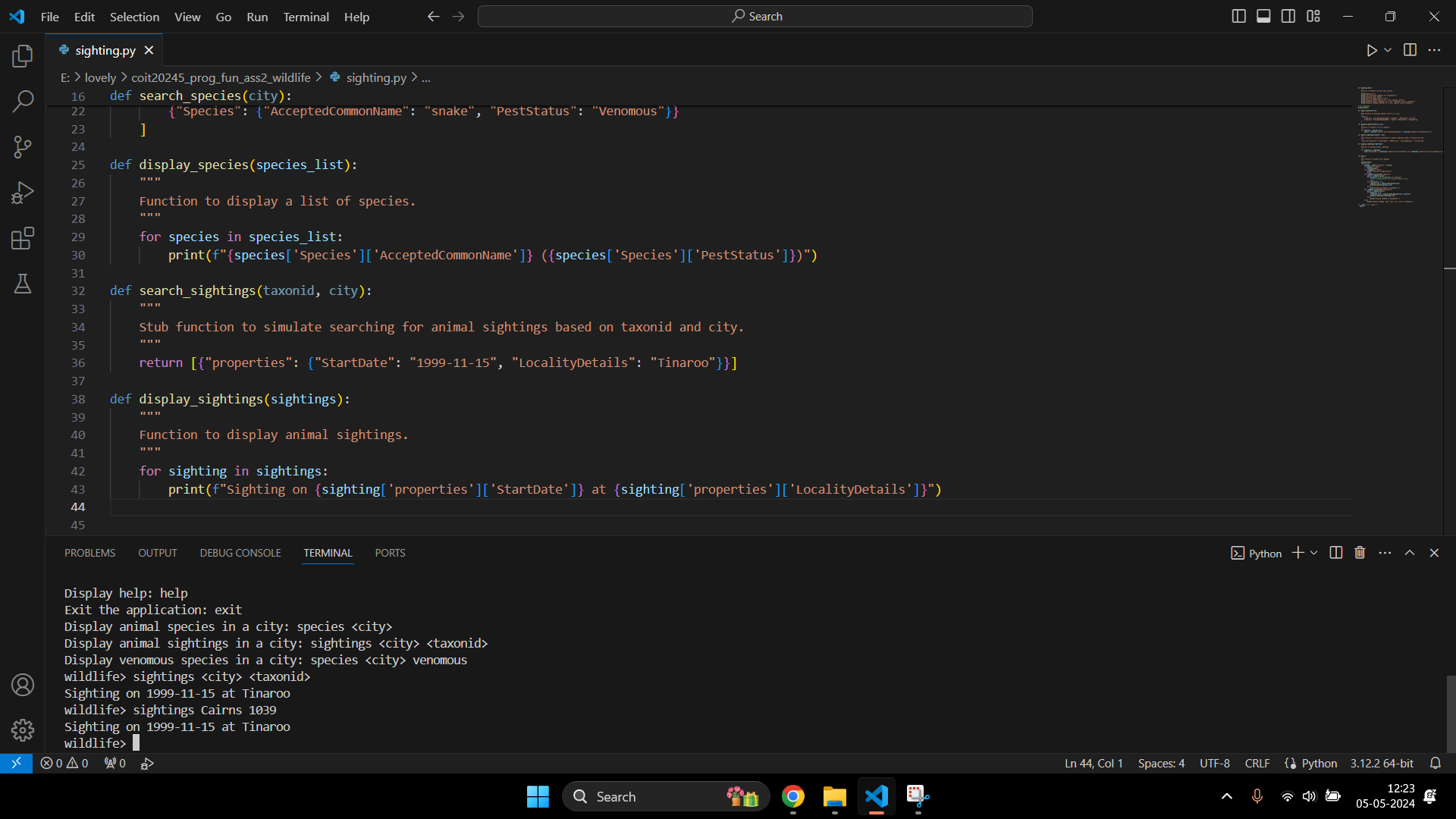
display\_species([{"Species": {"AcceptedCommonName": "dolphin", "PestStatus": "Nil"}}])

Species: dolphin, Pest Status: Nil

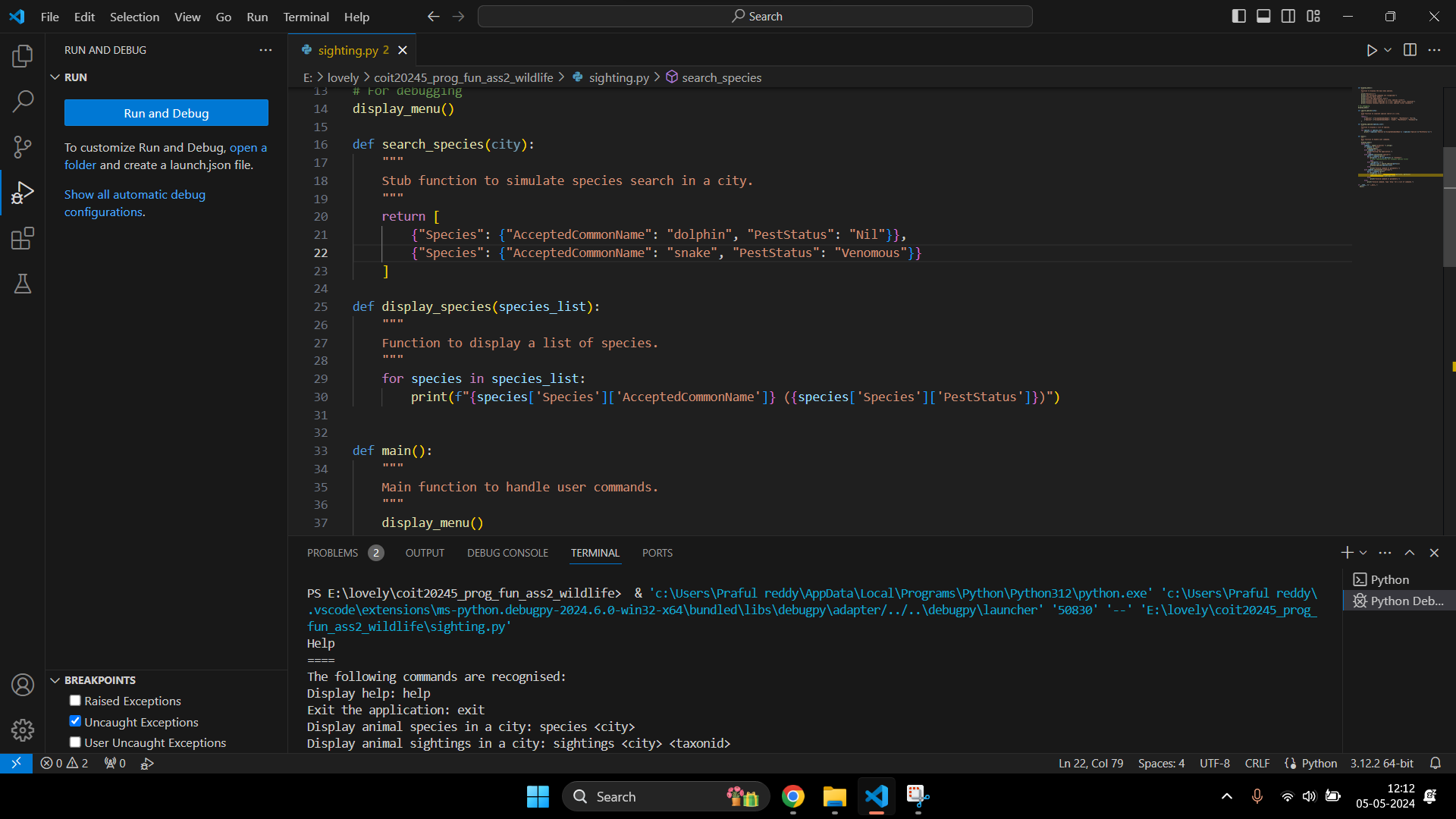
**Task 4**

**Screenshots:**

Update display\_menu() by adding a few functions.

****

Output after debugging the functions.

****

**About the Function used:**

> Function Name: search\_sightings(taxonid, city)

> Purpose: Searches for animal sightings based on TaxonID and city. Returns a stubbed list of sightings.

> Parameters: taxonid (int): The identifier for the species.

city (str): The city where the sightings are to be searched.

> Returns: list: A list of dictionaries, each containing sighting details.

> Exception: None

> Example Calls:

search\_sightings(1039 , " Cairns ")

[{"properties": {" StartDate ": " 1999-11-15", "Locality Details": "Tinaroo" }} ]

> Function Name: display\_sightings(sightings)

> Purpose: Displays a list of animal sightings.

> Parameters: sightings (list): A list of dictionaries containing sighting details.

> Returns: None

> Exception: None

> Example Calls:

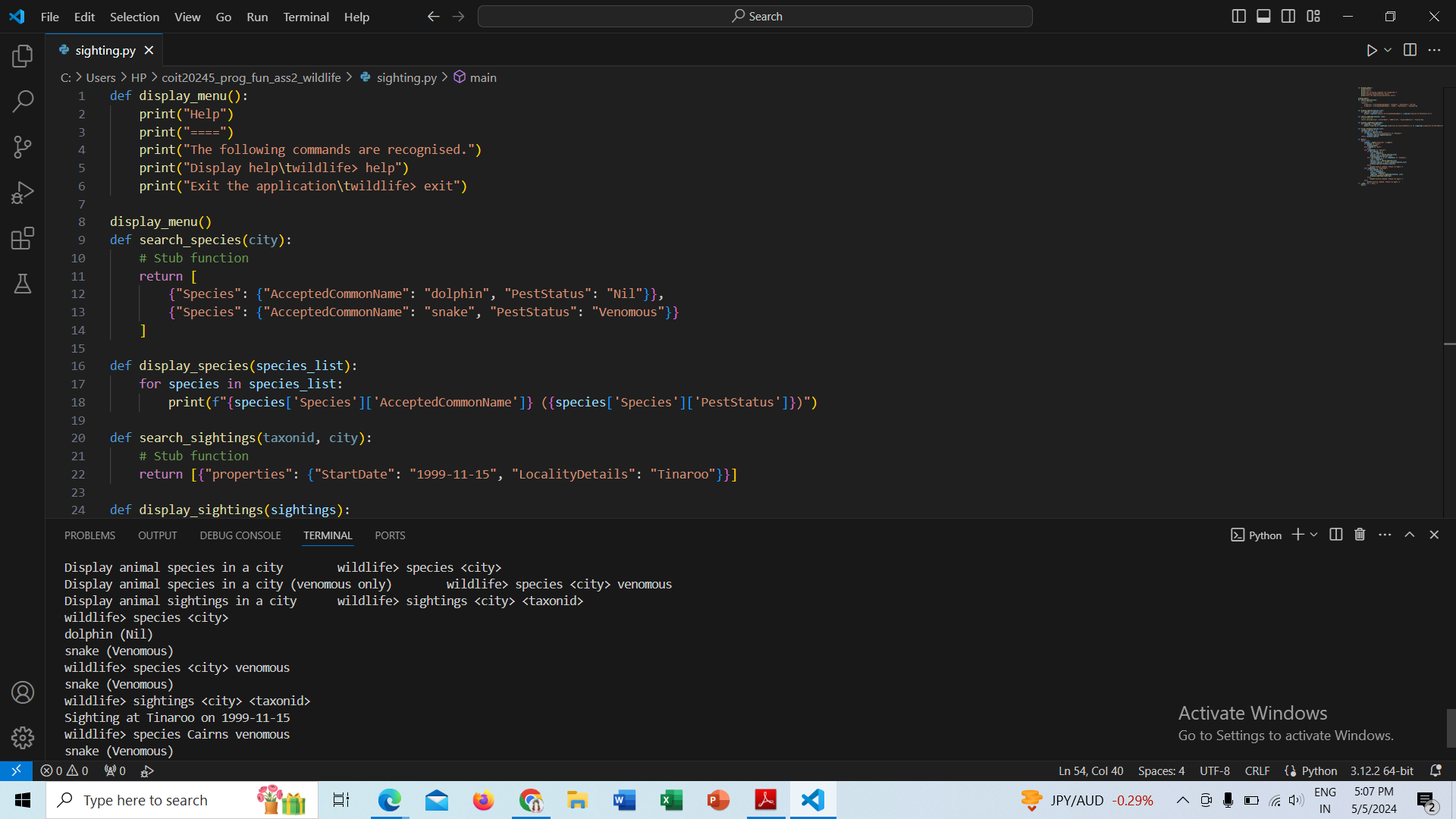
display\_sightings([{ "properties" : {"Start Date": "1999-11-15", "Locality Details": "Tinaroo"}}])

Sighting: Tinaroo, Date: 1999-11-15

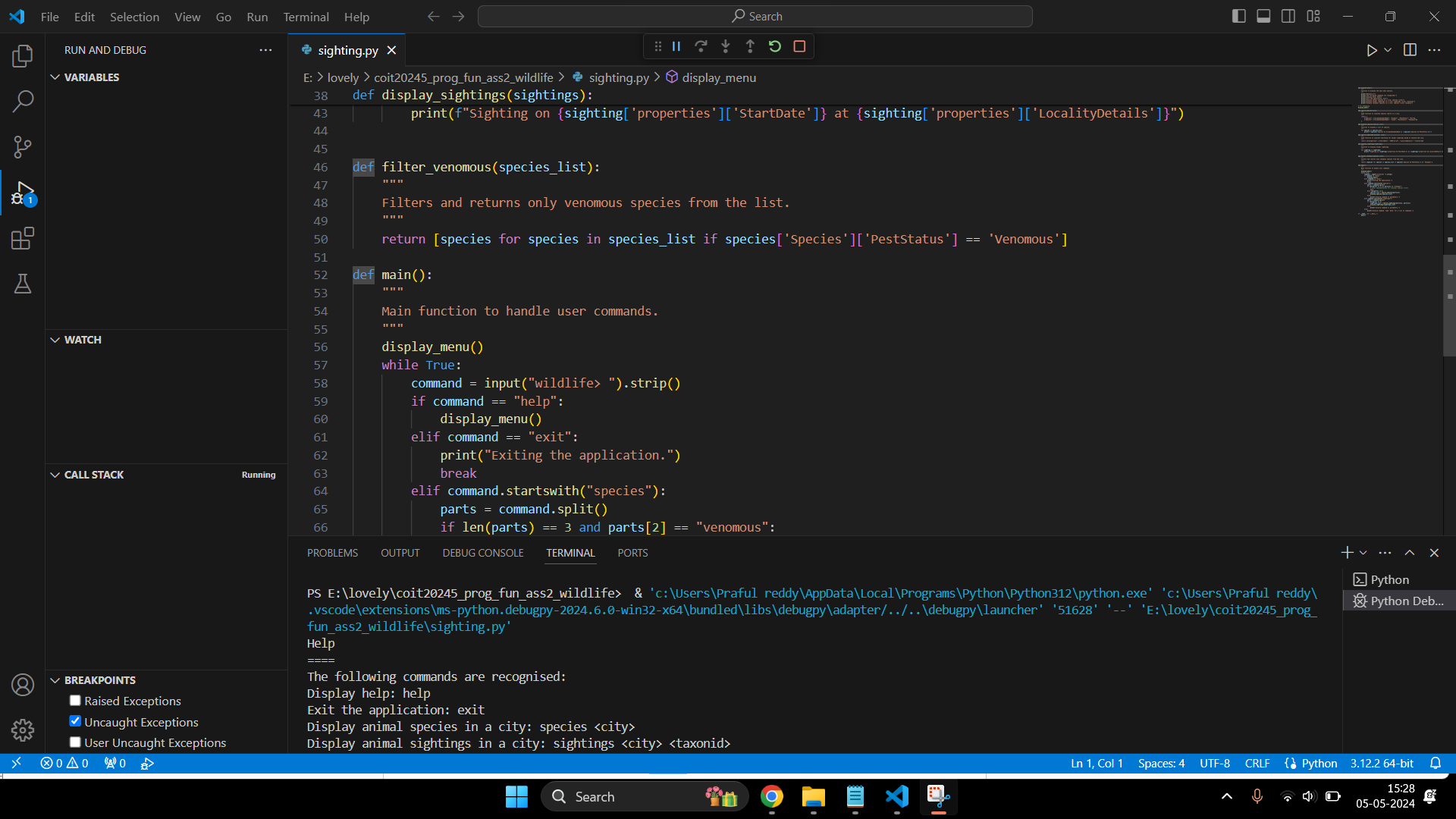
**Task 5**

**Screenshots:**

Update display\_menu() by adding a few functions.



Output after debugging the functions.



**About the Function used:**

> Function Name: filter\_venomous(species\_list)

> Purpose: Filters and returns only venomous species from a list of species.

> Parameters: species\_list (list): A list of dictionaries containing species details.

> Returns: list: A filtered list of dictionaries for venomous species only.

> Exception: None

> Example Calls:

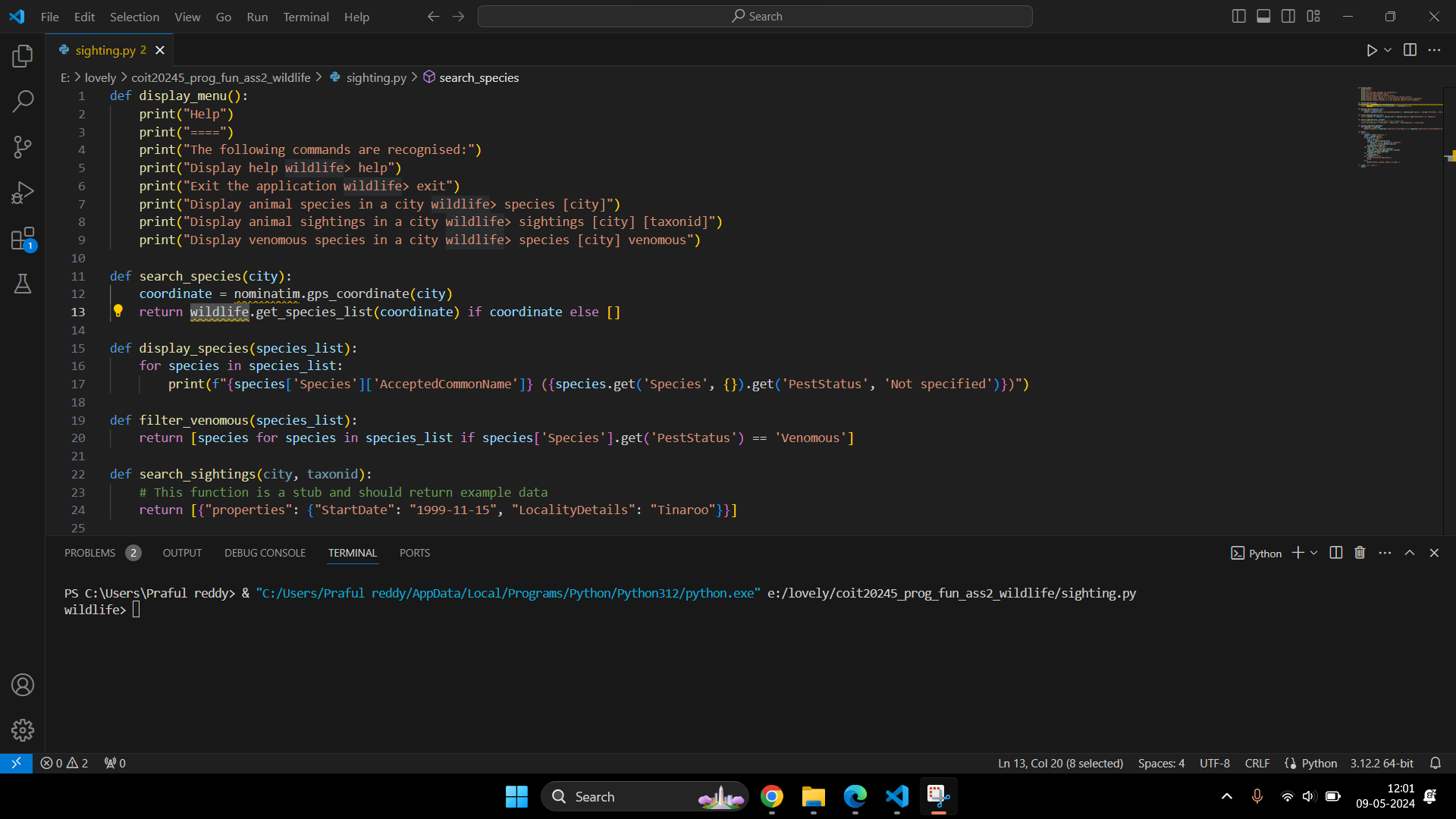
filter\_venomous([{"Species": {"AcceptedCommonName": "snake", "PestStatus": "Venomous"}}])

[{"Species": {"AcceptedCommonName": "snake", "PestStatus": "Venomous"}}]

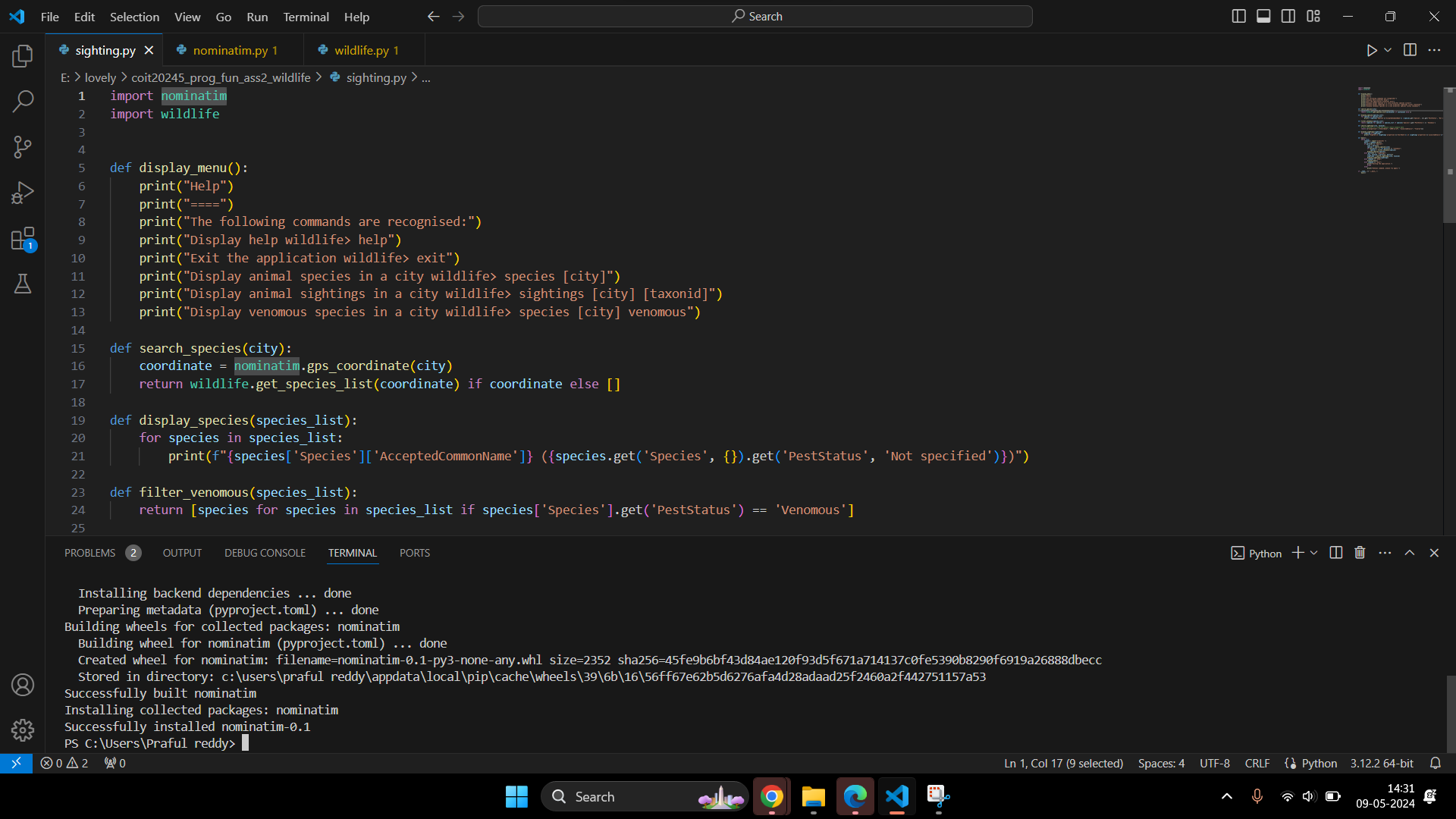
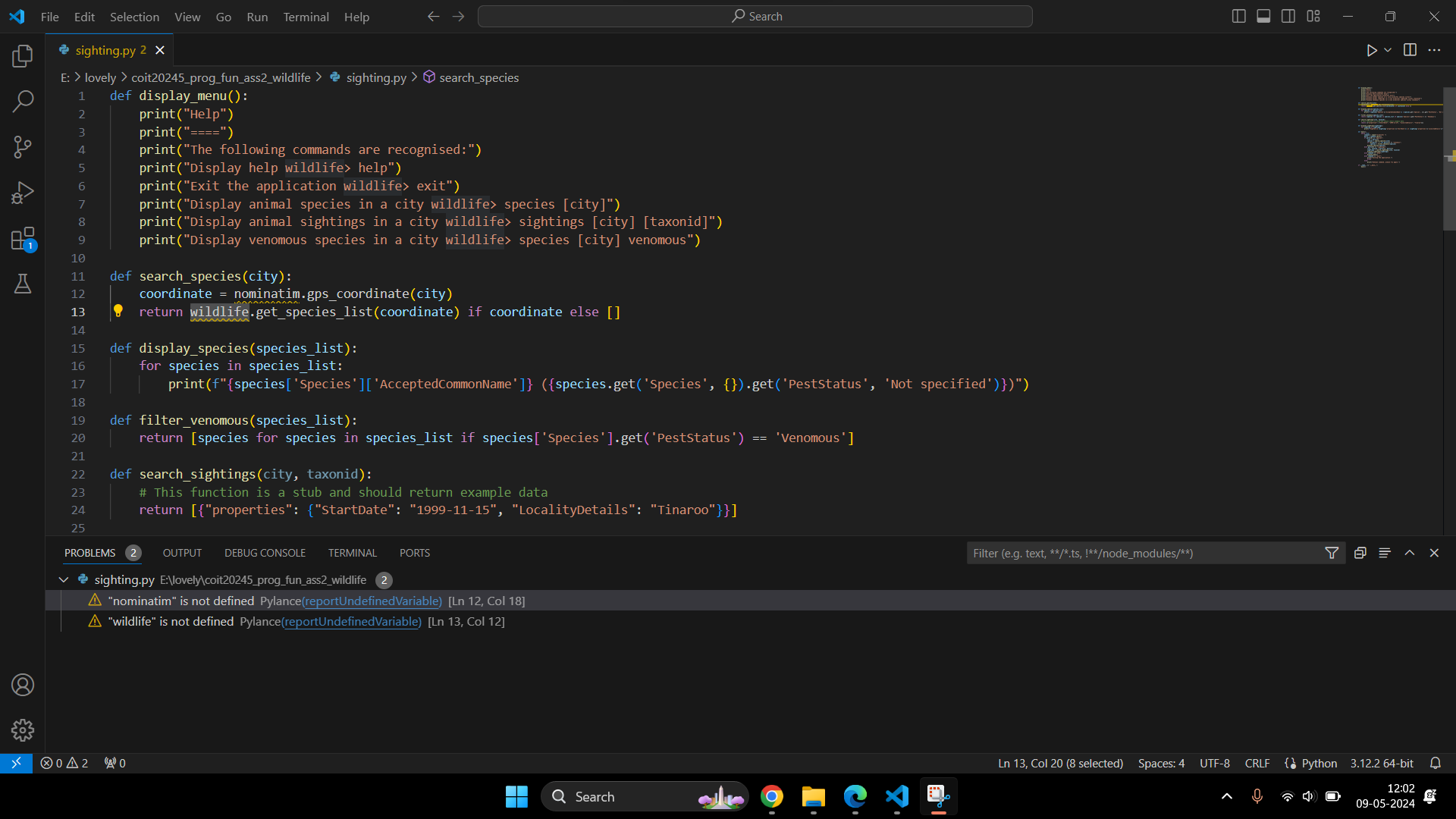
**Task 6**

**Screenshots:**

Updated search\_sightings(taxonid, city) function in sighting.py file.



Problems occurred while importing and updating the function.



> Function Name: gps(city)

> Purpose: Gets GPS coordinates for the given city name.

> Parameters: city (str): The city name.

> Returns: dict: The latitude and longitude of the city.

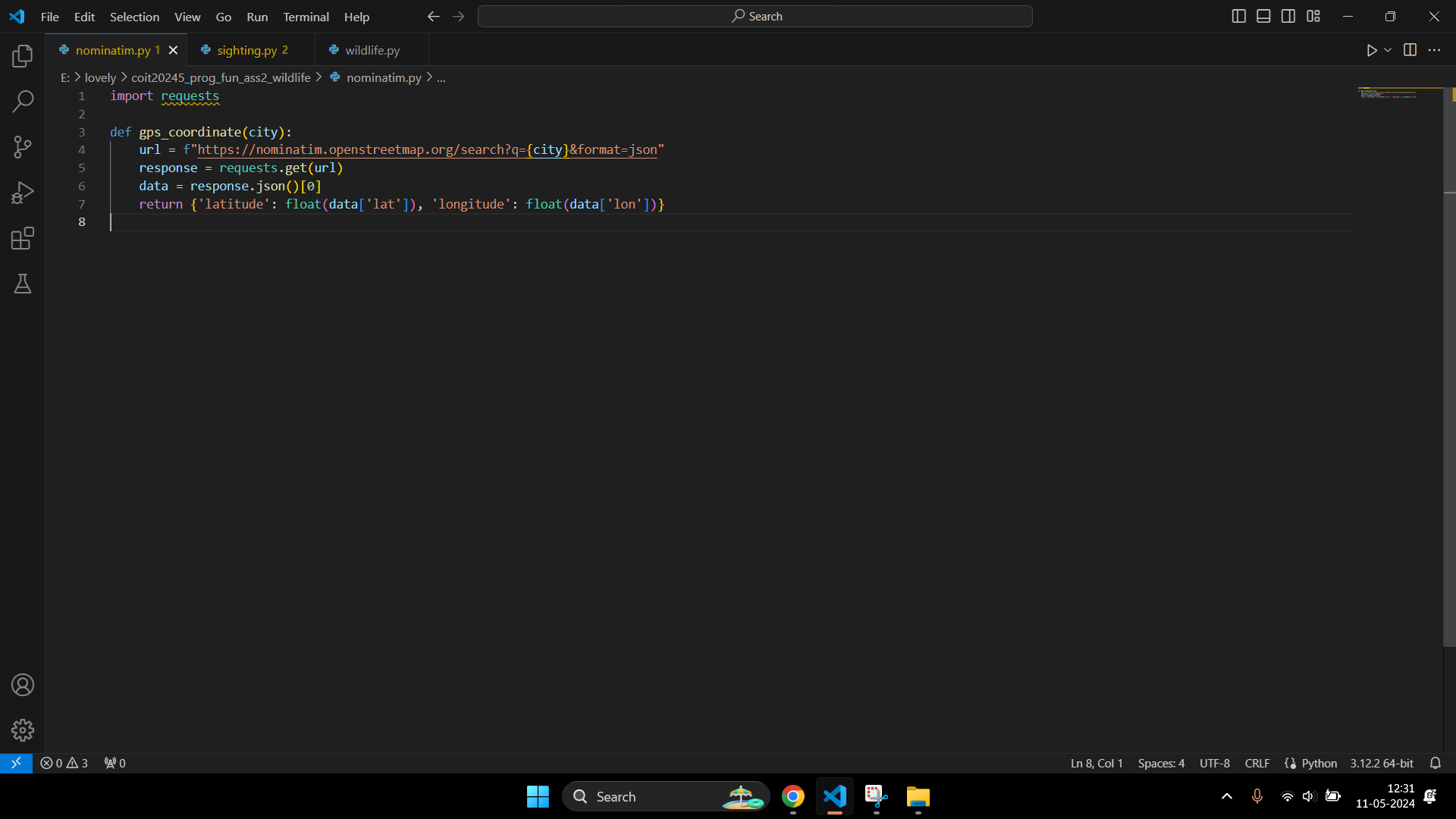
> Exception: None

> Example Calls: None

**Task 7**

**Screenshots:**

A new file nominatim.py has been created.



> Function Name: gps\_coordinate(city)

> Purpose: Fetches the latitude and longitude for a given city using the Nominatim geocoding service.

> Parameters: city (str): The city name to geocode.

> Returns: dict: A dictionary with keys 'latitude' and 'longitude'

> Exception: None

> Example Calls:

base\_url = 'https://nominatim.openstreetmap.org/search'

params = {'q': city, 'format': 'json'}

response = requests.get(base\_url, params=params)

data = response.json()

if data:

first\_result = data[0]

coordinates = {

'latitude': float(first\_result['lat']),

'longitude': float(first\_result['lon'])

}

return coordinates

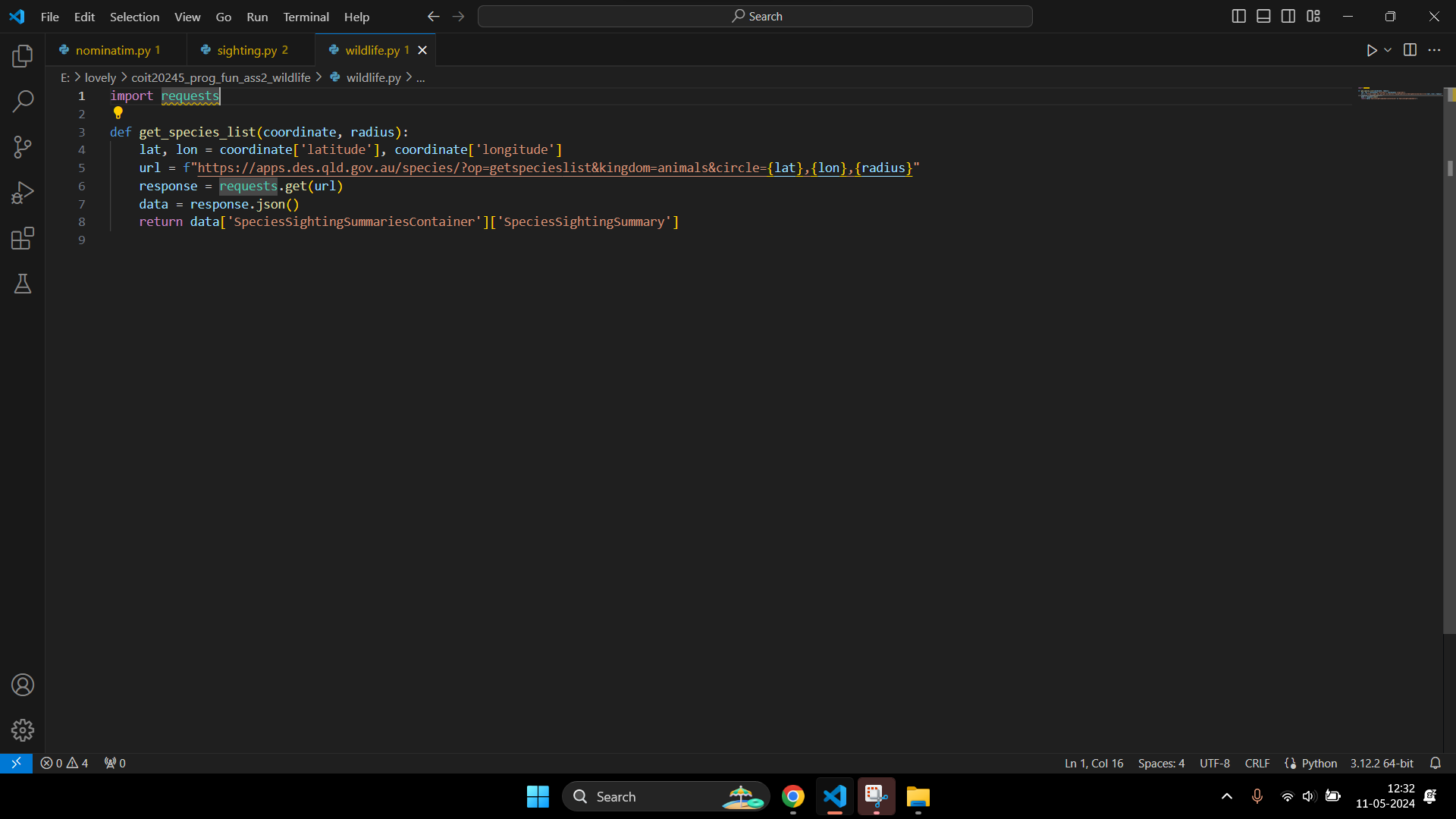
else:

return None

**Task 8**

**Screenshots:**

A new file wildlife.py has been created.

****

> Function Name: get\_species\_list(coordinate, radius)

> Purpose: Retrieves a list of species in an area defined by a circle with a given radius around a coordinate.

> Parameters: coordinate (dict): A dictionary with keys 'latitude' and 'longitude'.

radius (int): The radius of the search area in meters.

> Returns:list: A list of species dictionaries.

> Exception: None

> Example calls:

base\_url = 'https://apps.des.qld.gov.au/species/'

params = {

'op': 'getspecieslist',

'kingdom': 'animals',

'circle': f"{coordinate['latitude']},{coordinate['longitude']},{radius}"

}

response = requests.get(base\_url, params=params)

data = response.json()

return data["SpeciesSightingSummariesContainer"]["SpeciesSightingSummary"]