Smart Waste Collection Route Planner - Ai based project

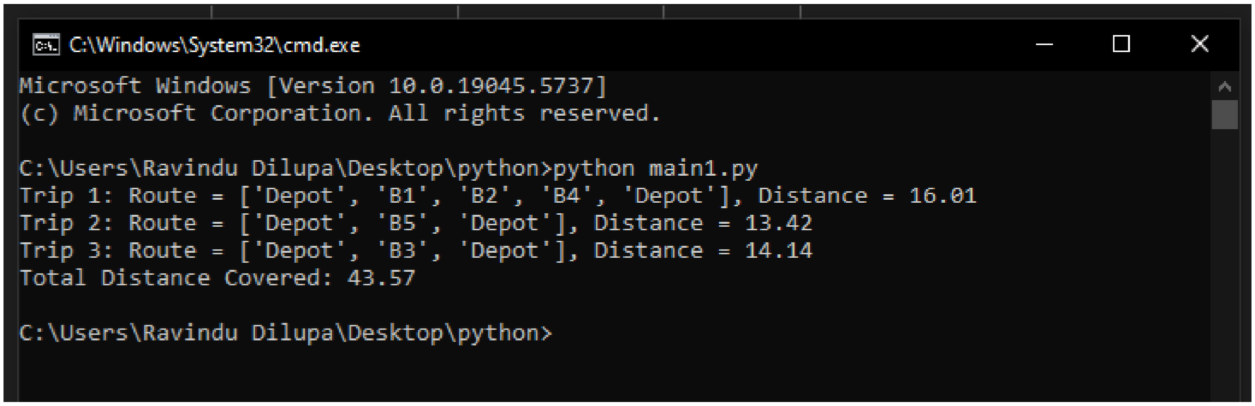
Name: M.K.R.D. Gunasekara

Temp No. 241065

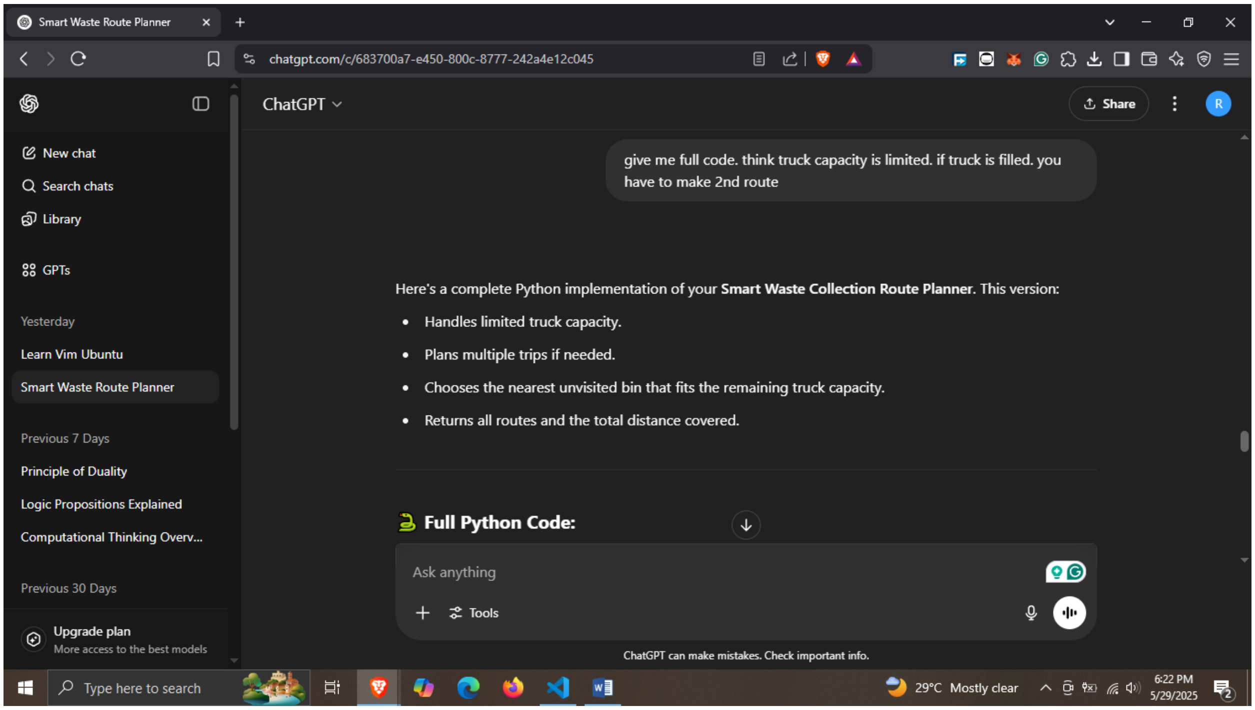
**************code***************

```
import math


# Sample data
bins = [
    {"id": "B1", "amount": 30, "location": (2, 3)},
    {"id": "B2", "amount": 20, "location": (5, 4)},
    {"id": "B3", "amount": 40, "location": (1, 7)},
    {"id": "B4", "amount": 10, "location": (6, 1)},
    {"id": "B5", "amount": 25, "location": (3, 6)},
]


truck_capacity = 60
depot_location = (0, 0)


# Distance function
def euclidean_distance(p1, p2):
    return math.sqrt((p1[0] - p2[0]) ** 2 + (p1[1] - p2[1]) ** 2)
```

```
# Find nearest bin that fits
def find_next_bin(current_location, remaining_capacity, bins, visited):
    candidates = [
        b for b in bins
        if b["id"] not in visited and b["amount"] <= remaining_capacity
    ]
    if not candidates:
        return None
    return min(candidates, key=lambda b: euclidean_distance(current_location,
b["location"]))


# Plan all trips
def plan_routes(bins, truck_capacity, depot_location):
    visited = set()
    all_routes = []
    total_distance = 0

    while len(visited) < len(bins):
        route = ["Depot"]
        distance = 0
        current_location = depot_location
        remaining_capacity = truck_capacity

        while True:
            next_bin = find_next_bin(current_location, remaining_capacity, bins, visited)
            if not next_bin:
                break
            route.append(next_bin["id"])
            distance += euclidean_distance(current_location, next_bin["location"])
            remaining_capacity -= next_bin["amount"]
            current_location = next_bin["location"]
            visited.add(next_bin["id"])

        # Return to depot
        distance += euclidean_distance(current_location, depot_location)
        route.append("Depot")
        all_routes.append({"route": route, "distance": round(distance, 2)})
        total_distance += distance

    return all_routes, round(total_distance, 2)

# Run
routes, total = plan_routes(bins, truck_capacity, depot_location)

# Output
for i, trip in enumerate(routes, 1):
    print(f"Trip {i}: Route = {trip['route']}, Distance = {trip['distance']}")
print(f"Total Distance Covered: {total}")
```



```
Microsoft Windows [Version 10.0.19045.5737]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Ravindu Dilupa\Desktop\python>python main1.py
Trip 1: Route = ['Depot', 'B1', 'B2', 'B4', 'Depot'], Distance = 16.01
Trip 2: Route = ['Depot', 'B5', 'Depot'], Distance = 13.42
Trip 3: Route = ['Depot', 'B3', 'Depot'], Distance = 14.14
Total Distance Covered: 43.57

C:\Users\Ravindu Dilupa\Desktop\python>
```

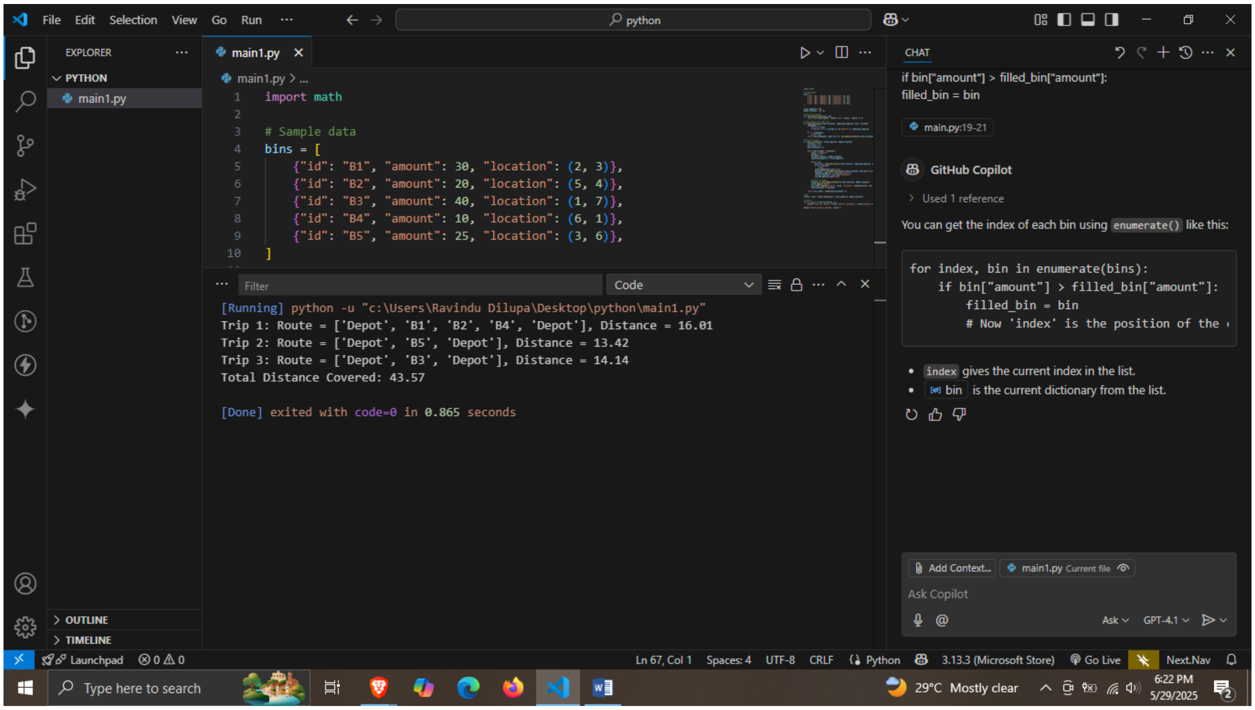

Smart Waste Route Planner
chatgpt.com/c/683700a7-e450-800c-8777-242a4e12c045
ChatGPT
Share
New chat
Search chats
Library
GPTs
Yesterday
Learn Vim Ubuntu
Smart Waste Route Planner
Previous 7 Days
Principle of Duality
Logic Propositions Explained
Computational Thinking Overv...
Previous 30 Days
Upgrade plan
More access to the best models
Ask anything
Tools
ChatGPT can make mistakes. Check important info.
Type here to search
29°C Mostly clear
6:22 PM
5/29/2025

give me full code. think truck capacity is limited. if truck is filled. you have to make 2nd route

Here's a complete Python implementation of your **Smart Waste Collection Route Planner**. This version:

- Handles limited truck capacity.
- Plans multiple trips if needed.
- Chooses the nearest unvisited bin that fits the remaining truck capacity.
- Returns all routes and the total distance covered.

Full Python Code:



main1.py > ...

```python
import math

# Sample data
bins = [
    {"id": "B1", "amount": 30, "location": (2, 3)},
    {"id": "B2", "amount": 20, "location": (5, 4)},
    {"id": "B3", "amount": 40, "location": (1, 7)},
    {"id": "B4", "amount": 10, "location": (6, 1)},
    {"id": "B5", "amount": 25, "location": (3, 6)},
]
```

```
[Running] python -u "c:\Users\Ravindu Dilupa\Desktop\python\main1.py"
Trip 1: Route = ['Depot', 'B1', 'B2', 'B4', 'Depot'], Distance = 16.01
Trip 2: Route = ['Depot', 'B5', 'Depot'], Distance = 13.42
Trip 3: Route = ['Depot', 'B3', 'Depot'], Distance = 14.14
Total Distance Covered: 43.57

[Done] exited with code=0 in 0.865 seconds
```

if bin["amount"] > filled_bin["amount"]:
filled_bin = bin

main.py:19-21

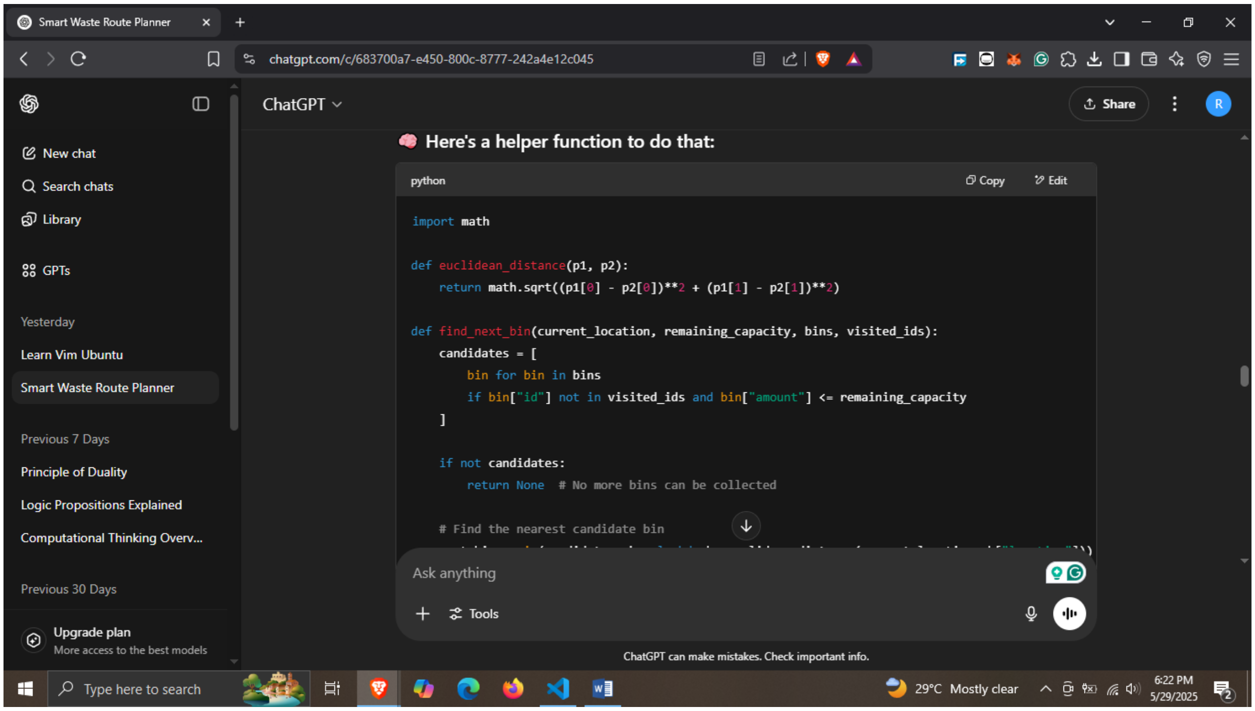
GitHub Copilot

Used 1 reference

You can get the index of each bin using `enumerate()` like this:

```python
for index, bin in enumerate(bins):
    if bin["amount"] > filled_bin["amount"]:
        filled_bin = bin
        # Now 'index' is the position of the
```

- `index` gives the current index in the list.
- `bin` is the current dictionary from the list.


Smart Waste Route Planner
chatgpt.com/c/683700a7-e450-800c-8777-242a4e12c045
ChatGPT
Share
New chat
Search chats
Library
GPTs
Yesterday
Learn Vim Ubuntu
Smart Waste Route Planner
Previous 7 Days
Principle of Duality
Logic Propositions Explained
Computational Thinking Overv...
Previous 30 Days
Upgrade plan
More access to the best models
python
Copy
Edit
Ask anything
Tools
ChatGPT can make mistakes. Check important info.
Type here to search
29°C Mostly clear
6:22 PM
5/29/2025

Here's a helper function to do that:

```python
import math

def euclidean_distance(p1, p2):
    return math.sqrt((p1[0] - p2[0])**2 + (p1[1] - p2[1])**2)

def find_next_bin(current_location, remaining_capacity, bins, visited_ids):
    candidates = [
        bin for bin in bins
        if bin["id"] not in visited_ids and bin["amount"] <= remaining_capacity
    ]

    if not candidates:
        return None  # No more bins can be collected

    # Find the nearest candidate bin
```

What software system did you choose to build, and why?

I chose to build a **Smart Waste Collection Route Planner**. This system addresses a real-world problem faced by urban waste management services. It was chosen because it combines optimization, environmental sustainability, and logistics. By efficiently planning waste collection routes, the system can reduce fuel consumption, minimize travel time, and prevent truck overloading, contributing to smarter city operations.

Which AI tools did you use during development, and how exactly did they help?

I used **ChatGPT** during the entire development process. It provided assistance in the following ways:

- Clarifying the problem requirements and helping define the input and output structure.
- Assisting in the design of the algorithm by explaining different approaches such as greedy selection and route optimization under constraints.
- Generating Python code to calculate Euclidean distance, filter bins by waste amount and proximity, and manage truck routing.
- Refining logic for handling multiple trips when the truck reaches its capacity.

ChatGPT functioned like an intelligent coding assistant that accelerated development while also helping me understand key programming concepts.

What parts of the system did you complete with AI assistance?

AI assistance was used to:

- Design the greedy algorithm that prioritizes larger bins first and selects the nearest one among ties.
- Implement functions such as `find_best_bin` and `plan_routes`, which form the core logic of the system.
- Write clean and readable Python code with logical structure and reusable components.
- Debug and refine the handling of multiple trips when the truck's capacity is exceeded.

Although I reviewed and modified the code where necessary, the AI played a significant role in creating a functional prototype.

What challenges did you face, and how did you solve them?

A key challenge was how to select the next bin in a way that balances waste amount and travel efficiency while staying within the truck's capacity. Initially, I only prioritized distance, which led to suboptimal waste collection. With the help of AI, I modified the algorithm to prioritize bins with the largest waste amounts that still fit within the remaining capacity. Another challenge was planning multiple trips. I addressed this by tracking visited bins and looping until all were collected, returning the truck to the depot at the end of each trip.

Do you believe AI tools make programming easier or reduce the need for deep technical knowledge? Explain your thoughts.

AI tools do make programming easier by reducing time spent on syntax, debugging, and initial algorithm design. They serve as a productivity booster and learning aid. However, they do not eliminate the need for technical knowledge. A solid understanding of algorithms, logic, and software structure is still necessary to evaluate and modify the AI-generated code correctly. In this project, I was able to critically assess and adjust the code generated by the AI to ensure it met the requirements.

Would you consider using AI tools in your future studies or professional tasks? Why or why not?

Yes, I would consider using AI tools in the future. They provide significant support for rapid prototyping, learning new concepts, and automating routine coding tasks. As long as the outputs are reviewed and validated, AI tools can enhance both academic and professional productivity. They are especially useful for exploring different approaches to solving a problem, saving time, and improving code quality.