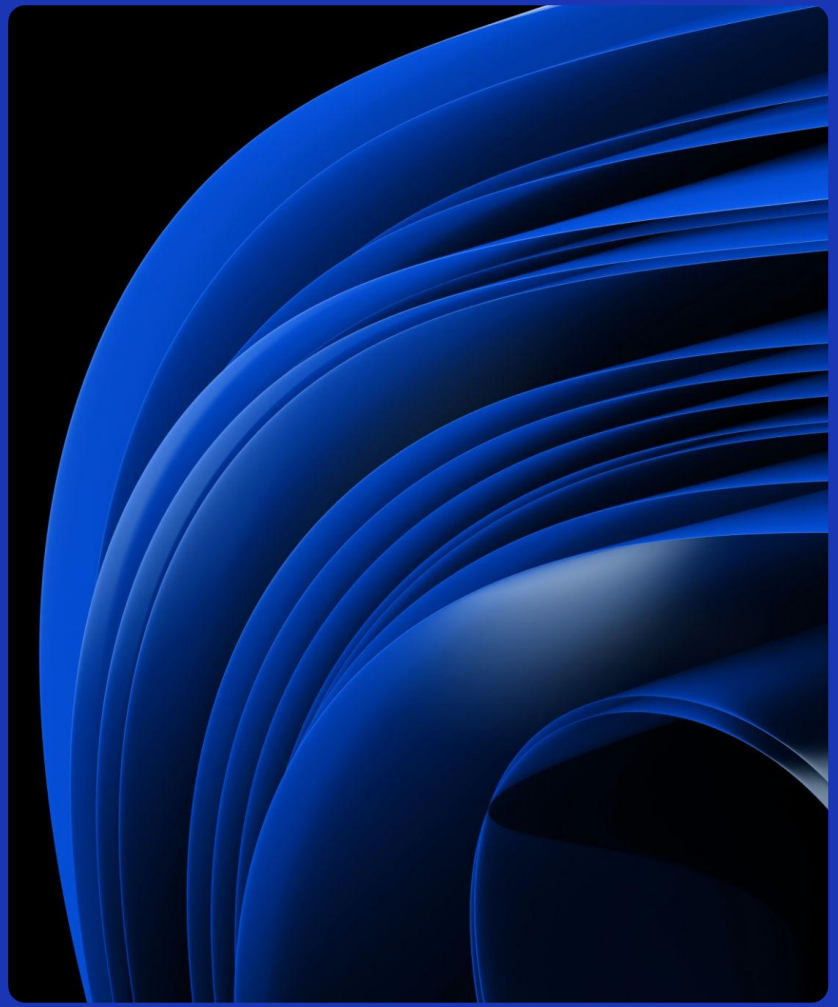
CYSE250

To-Do List

Using Python & Socket Programming

Tyler Walker
12/04/2024



Project Description

What it is:
This project involves the development of a task management server using Python, which allows users to securely manage their tasks over a network. The server handles user authentication and provides functionalities such as adding, viewing, removing, exporting, and importing tasks. User data is stored in a JSON file, ensuring that tasks persist across sessions. Priority levels are color coded: low = green, medium = yellow, high = red.

How it works:

- User connects to the server.
- Server authenticates the user with a username/password.
- Users can perform operations such as adding or deleting tasks.
- All communication is encrypted to ensure privacy.

Project Goal

- The primary goal of this project is to create a user-friendly task management application that can handle multiple users and their tasks securely. This includes implementing authentication, task management features, and ensuring data integrity through JSON file storage. The project aims to demonstrate how socket programming can be utilized to create a server-client architecture for managing tasks.

Hardware and Software Details

Hardware:

- Machine: MacBook Pro
- Processor: Intel Core i5
- RAM: 8 GB
- Storage: 256GB SSD

Software:

- Operating System: macOS Ventura
- Python Version: 3.10
- Libraries Used:
  - socket: for network communication
  - os: for interacting with the operating system
  - json: for handling data storage
  - datetime: for managing task deadlines
  - termcolor: for color-coding output in the terminal

Server Code

Server.py

```python
import socket
import os
import json
from datetime import datetime
from termcolor import colored

USER_DATA_FILE = "user_data.json"
if not os.path.exists(USER_DATA_FILE):
    with open(USER_DATA_FILE, 'w') as file:
        json.dump({}, file)

def load_user_data():
    with open(USER_DATA_FILE, 'r') as file:
        return json.load(file)

def save_user_data(data):
    with open(USER_DATA_FILE, 'w') as file:
        json.dump(data, file)

def handle_client(client_socket):
    user_data = load_user_data()


    client_socket.send("Enter username: ".encode())
    username = client_socket.recv(1024).decode()

    client_socket.send("Enter password: ".encode())
    password = client_socket.recv(1024).decode()

    if username in user_data and user_data[username]["password"] == password:
        client_socket.send(f"Welcome {username}!\n".encode())
    else:
        client_socket.send("Invalid username or password. Exiting...\n".encode())
        client_socket.close()
        return

    while True:

        client_socket.send("Choose an action: [1] Add Task [2] View Tasks [3] Remove Task [4] Export Tasks [5] Import Tasks [6] Quit\n".encode())
        action = client_socket.recv(1024).decode()

        if action == '1':
            client_socket.send("Enter task title: ".encode())
            title = client_socket.recv(1024).decode()
            client_socket.send("Enter task category: ".encode())
            category = client_socket.recv(1024).decode()
            client_socket.send("Enter task priority (high, medium, low): ".encode())
            priority = client_socket.recv(1024).decode()
            client_socket.send("Enter task deadline (YYYY-MM-DD): ".encode())
            deadline = client_socket.recv(1024).decode()
```

Server Code

Server.py

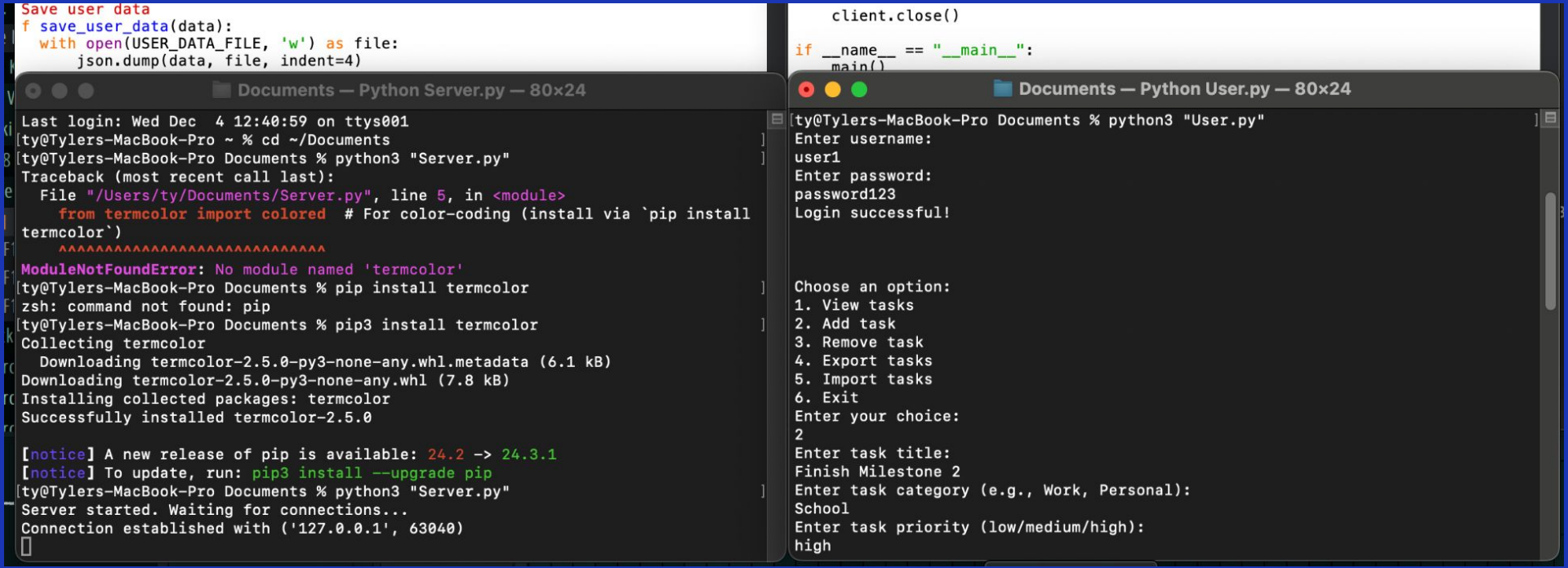
```
        elif action == '2':
            tasks = user_data.get(username, {}).get("tasks", [])
            response = "Your tasks:\n"
            for task in tasks:
                response += f"{task['title']} | Category: {task['category']} | Priority: {task['priority']} | Deadline: {task['deadline']}\n"
            client_socket.send(response.encode())

        elif action == '3':
            client_socket.send("Enter task title to remove: ".encode())
            title = client_socket.recv(1024).decode()
            tasks = user_data.get(username, {}).get("tasks", [])
            tasks = [task for task in tasks if task['title'] != title]
            user_data[username]["tasks"] = tasks
            save_user_data(user_data)
            client_socket.send(f"Task '{title}' removed successfully!\n".encode())

        elif action == '4':
            client_socket.send("Enter filename to export: ".encode())
            filename = client_socket.recv(1024).decode()
            with open(filename, 'w') as f:
                json.dump(user_data[username]["tasks"], f)
            client_socket.send(f"Tasks exported to '{filename}' successfully!\n".encode())

        elif action == '5':
            client_socket.send("Enter filename to import: ".encode())
            filename = client_socket.recv(1024).decode()
            with open(filename, 'r') as f:
                imported_tasks = json.load(f)
                user_data[username]["tasks"].extend(imported_tasks)
                save_user_data(user_data)
            client_socket.send(f"Tasks imported from '{filename}' successfully!\n".encode())

        elif action == '6':
            client_socket.send("Goodbye!\n".encode())
            break

    client_socket.close()

def main():
    server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server.bind(("127.0.0.1", 9999))
    server.listen(5)
    print("Server started. Waiting for connections...")

    while True:
        client_socket, addr = server.accept()
        print(f"Connection established with {addr}")
        handle_client(client_socket)

if __name__ == "__main__":
    main()
```

Client Code

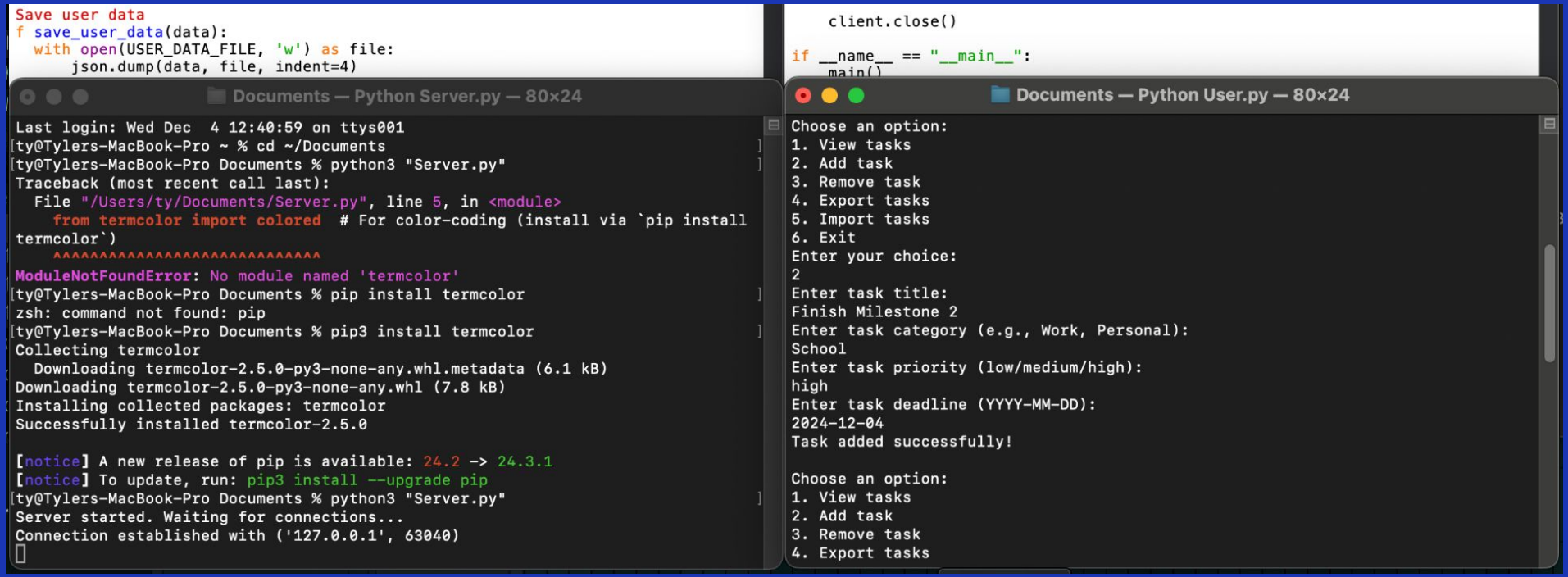
User.py

```python
import socket

def main():
    client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client.connect(("127.0.0.1", 9999))

    while True:

        server_message = client.recv(1024).decode()
        print(server_message)


        if "Goodbye!" in server_message:
            break


        user_input = input()
        client.send(user_input.encode())

    client.close()

if __name__ == "__main__":
    main()
```

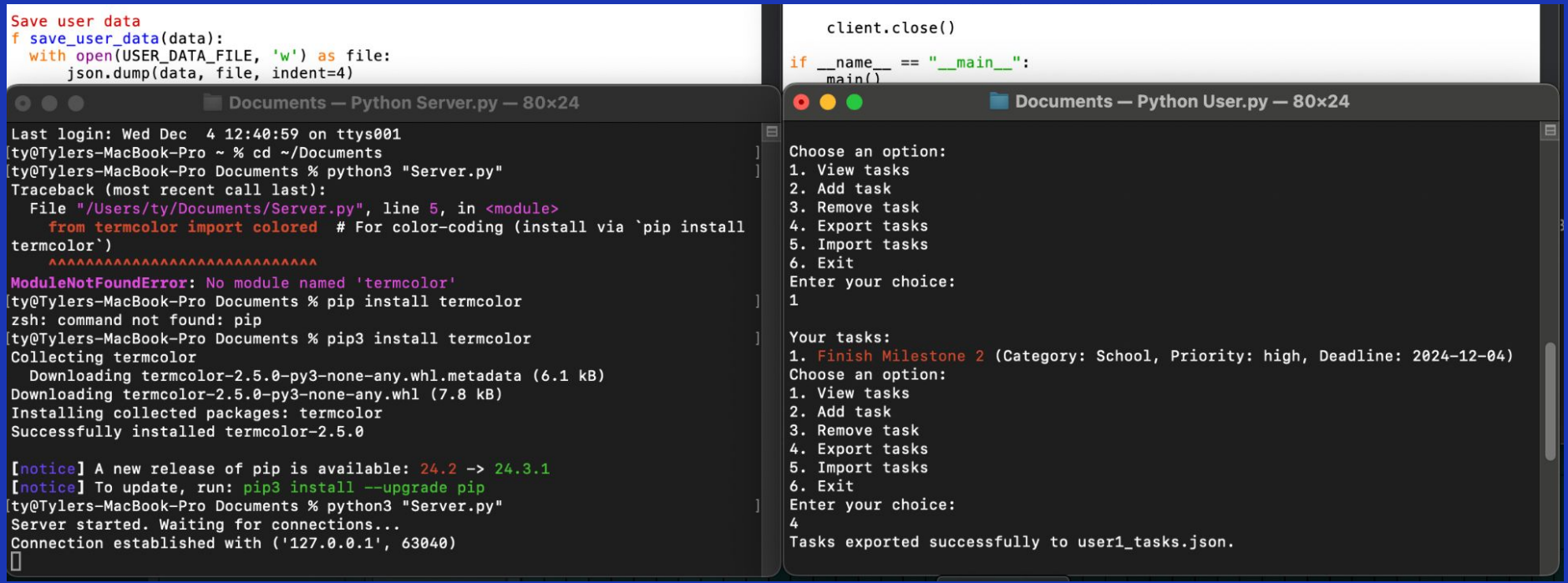
Outcome


Save user data
f save_user_data(data):
with open(USER_DATA_FILE, 'w') as file:
json.dump(data, file, indent=4)
client.close()
if __name__ == "__main__":
main()
Documents — Python Server.py — 80×24
Last login: Wed Dec 4 12:40:59 on ttys001
[ty@Tylers-MacBook-Pro ~ % cd ~/Documents
[ty@Tylers-MacBook-Pro Documents % python3 "Server.py"
Traceback (most recent call last):
File "/Users/ty/Documents/Server.py", line 5, in <module>
from termcolor import colored # For color-coding (install via `pip install termcolor`)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
ModuleNotFoundError: No module named 'termcolor'
[ty@Tylers-MacBook-Pro Documents % pip install termcolor
zsh: command not found: pip
[ty@Tylers-MacBook-Pro Documents % pip3 install termcolor
Collecting termcolor
Downloading termcolor-2.5.0-py3-none-any.whl.metadata (6.1 kB)
Downloading termcolor-2.5.0-py3-none-any.whl (7.8 kB)
Installing collected packages: termcolor
Successfully installed termcolor-2.5.0
[notice] A new release of pip is available: 24.2 -> 24.3.1
[notice] To update, run: pip3 install --upgrade pip
[ty@Tylers-MacBook-Pro Documents % python3 "Server.py"
Server started. Waiting for connections...
Connection established with ('127.0.0.1', 63040)
Documents — Python User.py — 80×24
[ty@Tylers-MacBook-Pro Documents % python3 "User.py"
Enter username:
user1
Enter password:
password123
Login successful!
Choose an option:
1. View tasks
2. Add task
3. Remove task
4. Export tasks
5. Import tasks
6. Exit
Enter your choice:
2
Enter task title:
Finish Milestone 2
Enter task category (e.g., Work, Personal):
School
Enter task priority (low/medium/high):
high

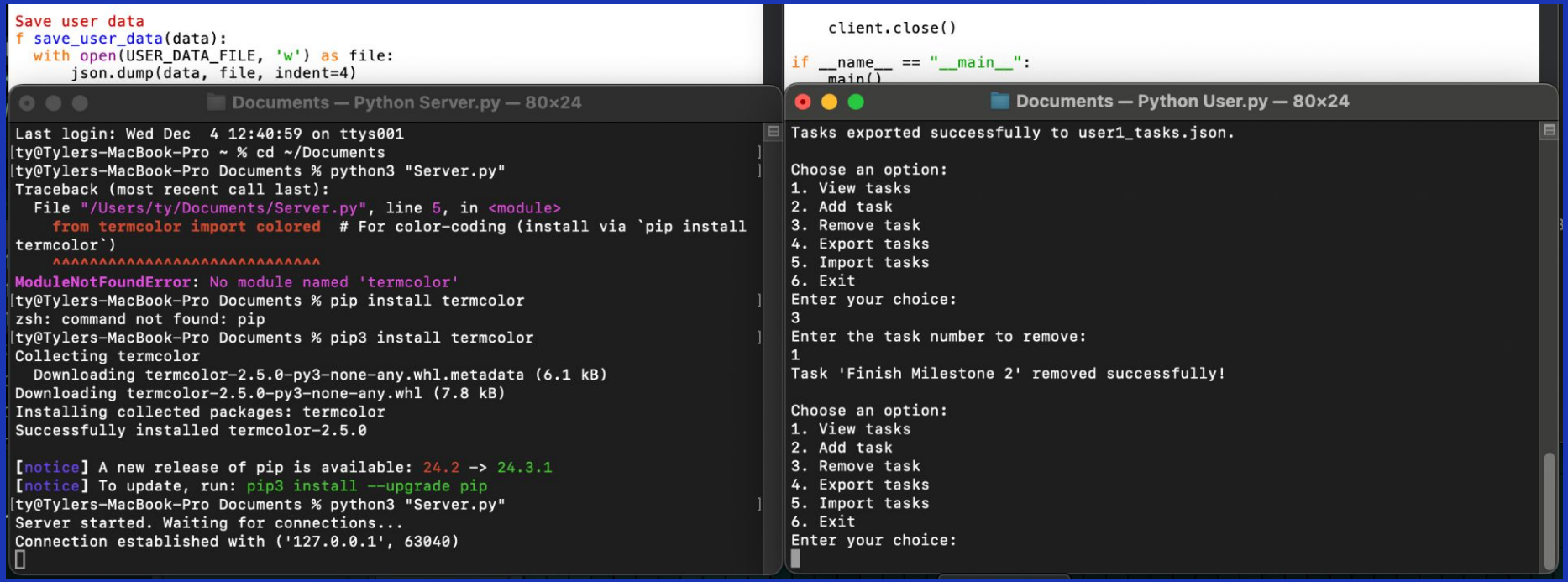
Outcome 2


Save user data
f save_user_data(data):
with open(USER_DATA_FILE, 'w') as file:
json.dump(data, file, indent=4)
client.close()
if __name__ == "__main__":
main()
Documents — Python Server.py — 80×24
Last login: Wed Dec 4 12:40:59 on ttys001
[ty@Tylers-MacBook-Pro ~ % cd ~/Documents
[ty@Tylers-MacBook-Pro Documents % python3 "Server.py"
Traceback (most recent call last):
File "/Users/ty/Documents/Server.py", line 5, in <module>
from termcolor import colored # For color-coding (install via `pip install termcolor`)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
ModuleNotFoundError: No module named 'termcolor'
[ty@Tylers-MacBook-Pro Documents % pip install termcolor
zsh: command not found: pip
[ty@Tylers-MacBook-Pro Documents % pip3 install termcolor
Collecting termcolor
Downloading termcolor-2.5.0-py3-none-any.whl.metadata (6.1 kB)
Downloading termcolor-2.5.0-py3-none-any.whl (7.8 kB)
Installing collected packages: termcolor
Successfully installed termcolor-2.5.0
[notice] A new release of pip is available: 24.2 -> 24.3.1
[notice] To update, run: pip3 install --upgrade pip
[ty@Tylers-MacBook-Pro Documents % python3 "Server.py"
Server started. Waiting for connections...
Connection established with ('127.0.0.1', 63040)
Documents — Python User.py — 80×24
Choose an option:
1. View tasks
2. Add task
3. Remove task
4. Export tasks
5. Import tasks
6. Exit
Enter your choice:
2
Enter task title:
Finish Milestone 2
Enter task category (e.g., Work, Personal):
School
Enter task priority (low/medium/high):
high
Enter task deadline (YYYY-MM-DD):
2024-12-04
Task added successfully!
Choose an option:
1. View tasks
2. Add task
3. Remove task
4. Export tasks

Outcome 3


Save user data
f save_user_data(data):
with open(USER_DATA_FILE, 'w') as file:
json.dump(data, file, indent=4)
client.close()
if __name__ == "__main__":
main()
Documents — Python Server.py — 80×24
Last login: Wed Dec 4 12:40:59 on ttys001
ty@Tylers-MacBook-Pro ~ % cd ~/Documents
ty@Tylers-MacBook-Pro Documents % python3 "Server.py"
Traceback (most recent call last):
File "/Users/ty/Documents/Server.py", line 5, in <module>
from termcolor import colored # For color-coding (install via `pip install termcolor`)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
ModuleNotFoundError: No module named 'termcolor'
ty@Tylers-MacBook-Pro Documents % pip install termcolor
zsh: command not found: pip
ty@Tylers-MacBook-Pro Documents % pip3 install termcolor
Collecting termcolor
Downloading termcolor-2.5.0-py3-none-any.whl.metadata (6.1 kB)
Downloading termcolor-2.5.0-py3-none-any.whl (7.8 kB)
Installing collected packages: termcolor
Successfully installed termcolor-2.5.0
[notice] A new release of pip is available: 24.2 -> 24.3.1
[notice] To update, run: pip3 install --upgrade pip
ty@Tylers-MacBook-Pro Documents % python3 "Server.py"
Server started. Waiting for connections...
Connection established with ('127.0.0.1', 63040)
Documents — Python User.py — 80×24
Choose an option:
1. View tasks
2. Add task
3. Remove task
4. Export tasks
5. Import tasks
6. Exit
Enter your choice:
1
Your tasks:
1. Finish Milestone 2 (Category: School, Priority: high, Deadline: 2024-12-04)
Choose an option:
1. View tasks
2. Add task
3. Remove task
4. Export tasks
5. Import tasks
6. Exit
Enter your choice:
4
Tasks exported successfully to user1_tasks.json.

Outcome 4


Save user data
f save_user_data(data):
with open(USER_DATA_FILE, 'w') as file:
json.dump(data, file, indent=4)
client.close()
if __name__ == "__main__":
main()
Documents — Python Server.py — 80×24
Last login: Wed Dec 4 12:40:59 on ttys001
ty@Tylers-MacBook-Pro ~ % cd ~/Documents
ty@Tylers-MacBook-Pro Documents % python3 "Server.py"
Traceback (most recent call last):
File "/Users/ty/Documents/Server.py", line 5, in <module>
from termcolor import colored # For color-coding (install via `pip install termcolor`)
ModuleNotFoundError: No module named 'termcolor'
ty@Tylers-MacBook-Pro Documents % pip install termcolor
zsh: command not found: pip
ty@Tylers-MacBook-Pro Documents % pip3 install termcolor
Collecting termcolor
Downloading termcolor-2.5.0-py3-none-any.whl.metadata (6.1 kB)
Downloading termcolor-2.5.0-py3-none-any.whl (7.8 kB)
Installing collected packages: termcolor
Successfully installed termcolor-2.5.0
[notice] A new release of pip is available: 24.2 -> 24.3.1
[notice] To update, run: pip3 install --upgrade pip
ty@Tylers-MacBook-Pro Documents % python3 "Server.py"
Server started. Waiting for connections...
Connection established with ('127.0.0.1', 63040)
Documents — Python User.py — 80×24
Tasks exported successfully to user1_tasks.json.
Choose an option:
1. View tasks
2. Add task
3. Remove task
4. Export tasks
5. Import tasks
6. Exit
Enter your choice:
3
Enter the task number to remove:
1
Task 'Finish Milestone 2' removed successfully!
Choose an option:
1. View tasks
2. Add task
3. Remove task
4. Export tasks
5. Import tasks
6. Exit
Enter your choice: