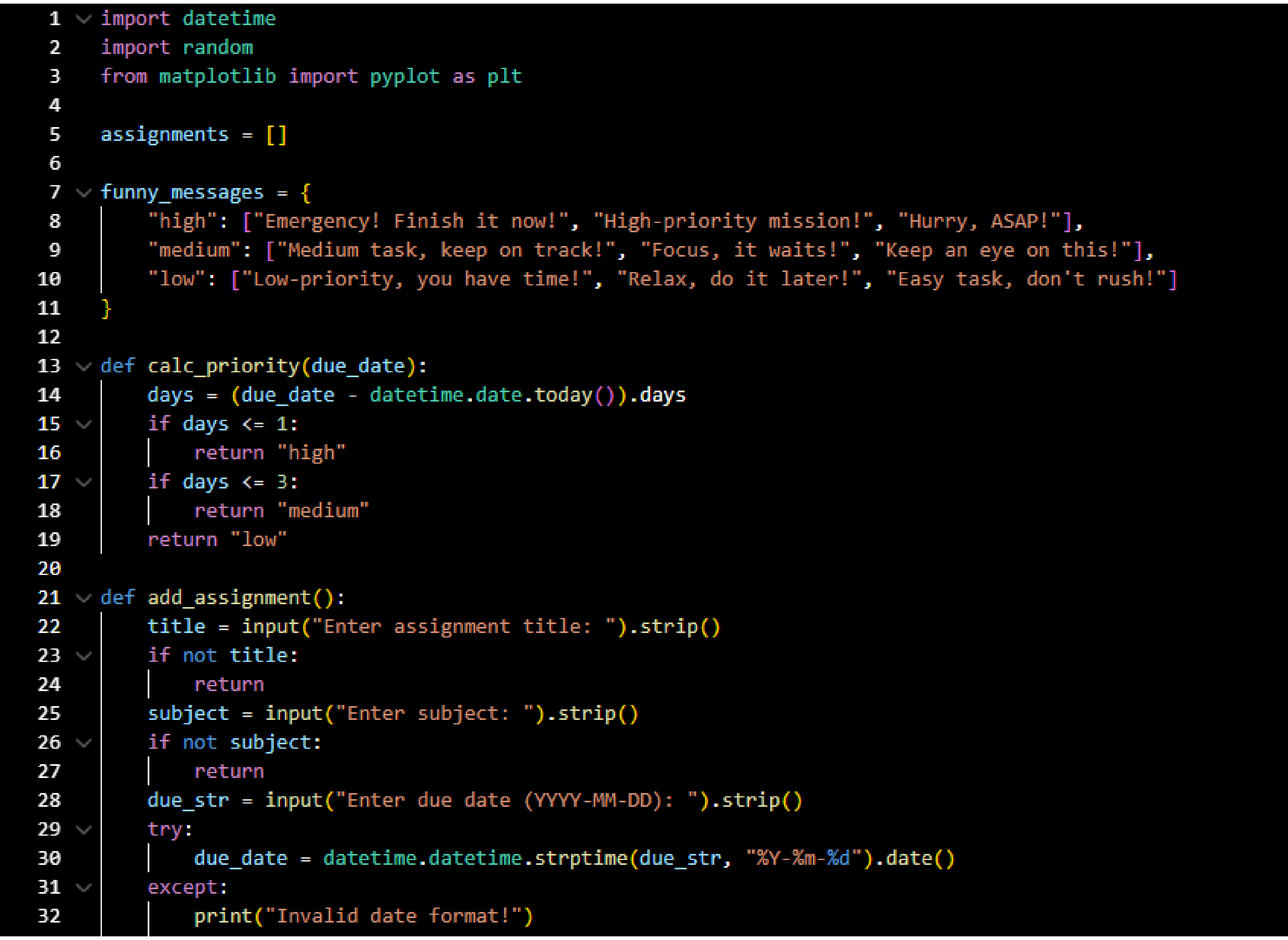
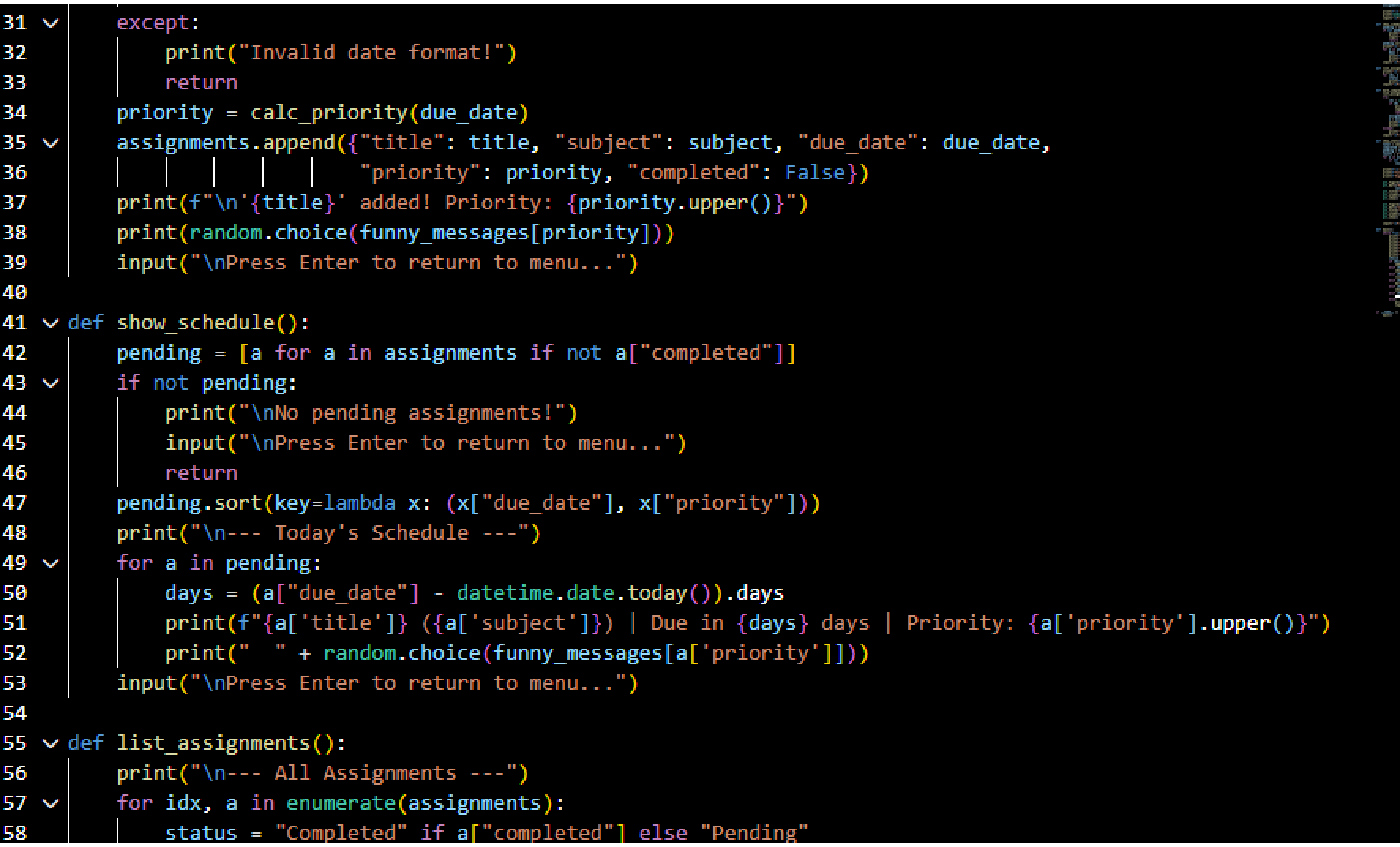
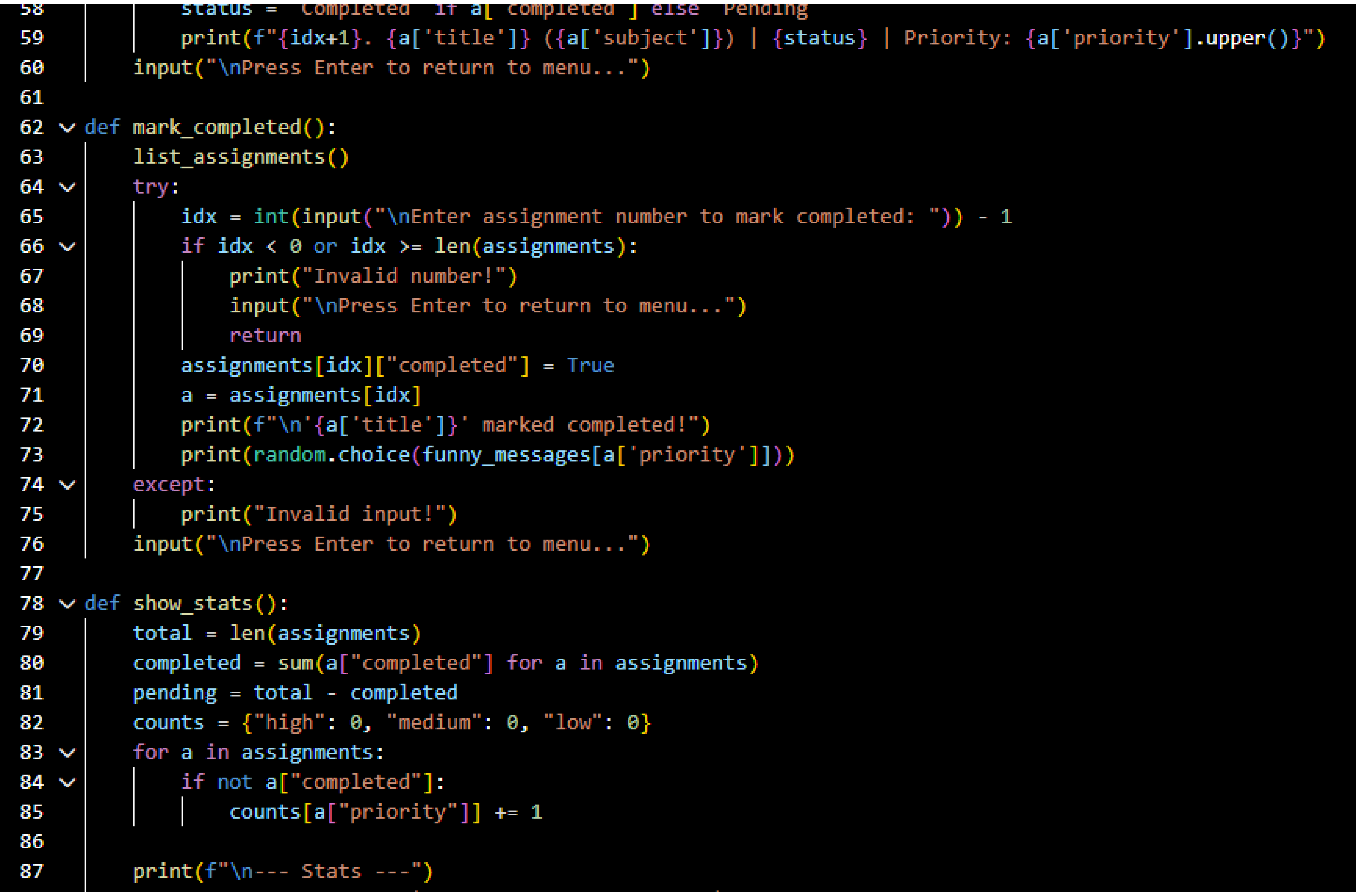


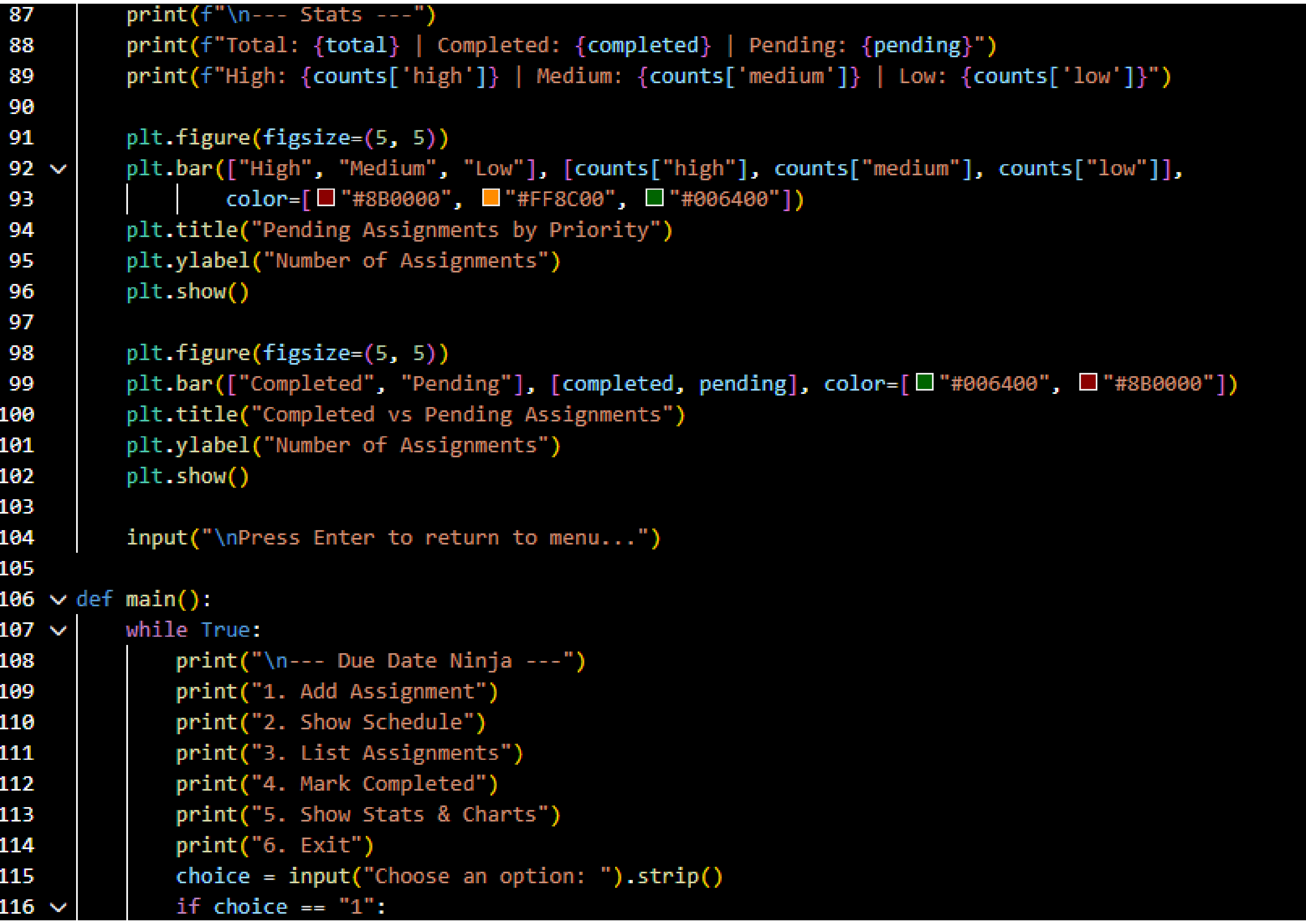
```python
import datetime
import random
from matplotlib import pyplot as plt

assignments = []

funny_messages = {
    "high": ["Emergency! Finish it now!", "High-priority mission!", "Hurry, ASAP!"],
    "medium": ["Medium task, keep on track!", "Focus, it waits!", "Keep an eye on this!"],
    "low": ["Low-priority, you have time!", "Relax, do it later!", "Easy task, don't rush!"]
}

def calc_priority(due_date):
    days = (due_date - datetime.date.today()).days
    if days <= 1:
        return "high"
    if days <= 3:
        return "medium"
    return "low"

def add_assignment():
    title = input("Enter assignment title: ").strip()
    if not title:
        return
    subject = input("Enter subject: ").strip()
    if not subject:
        return
    due_str = input("Enter due date (YYYY-MM-DD): ").strip()
    try:
        due_date = datetime.datetime.strptime(due_str, "%Y-%m-%d").date()
    except:
        print("Invalid date format!")
```



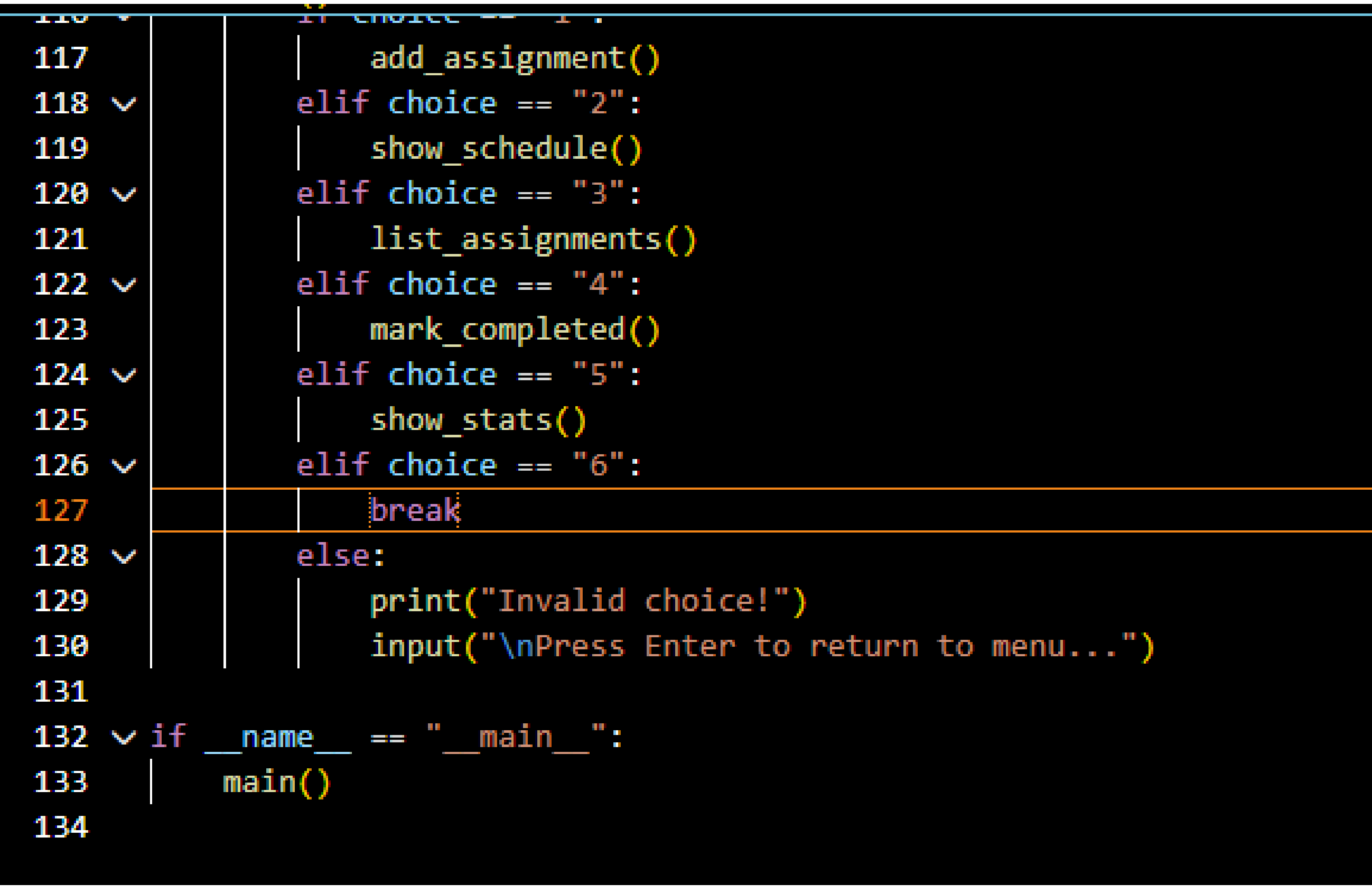
```python
    except:
        print("Invalid date format!")
        return
    priority = calc_priority(due_date)
    assignments.append({"title": title, "subject": subject, "due_date": due_date,
                        "priority": priority, "completed": False})
    print(f"\n'{title}' added! Priority: {priority.upper()}")
    print(random.choice(funny_messages[priority]))
    input("\nPress Enter to return to menu...")

def show_schedule():
    pending = [a for a in assignments if not a["completed"]]
    if not pending:
        print("\nNo pending assignments!")
        input("\nPress Enter to return to menu...")
        return
    pending.sort(key=lambda x: (x["due_date"], x["priority"]))
    print("\n--- Today's Schedule ---")
    for a in pending:
        days = (a["due_date"] - datetime.date.today()).days
        print(f"{a['title']} ({a['subject']}) | Due in {days} days | Priority: {a['priority'].upper()}")
        print("  " + random.choice(funny_messages[a['priority']]))
    input("\nPress Enter to return to menu...")

def list_assignments():
    print("\n--- All Assignments ---")
    for idx, a in enumerate(assignments):
        status = "Completed" if a["completed"] else "Pending"
```



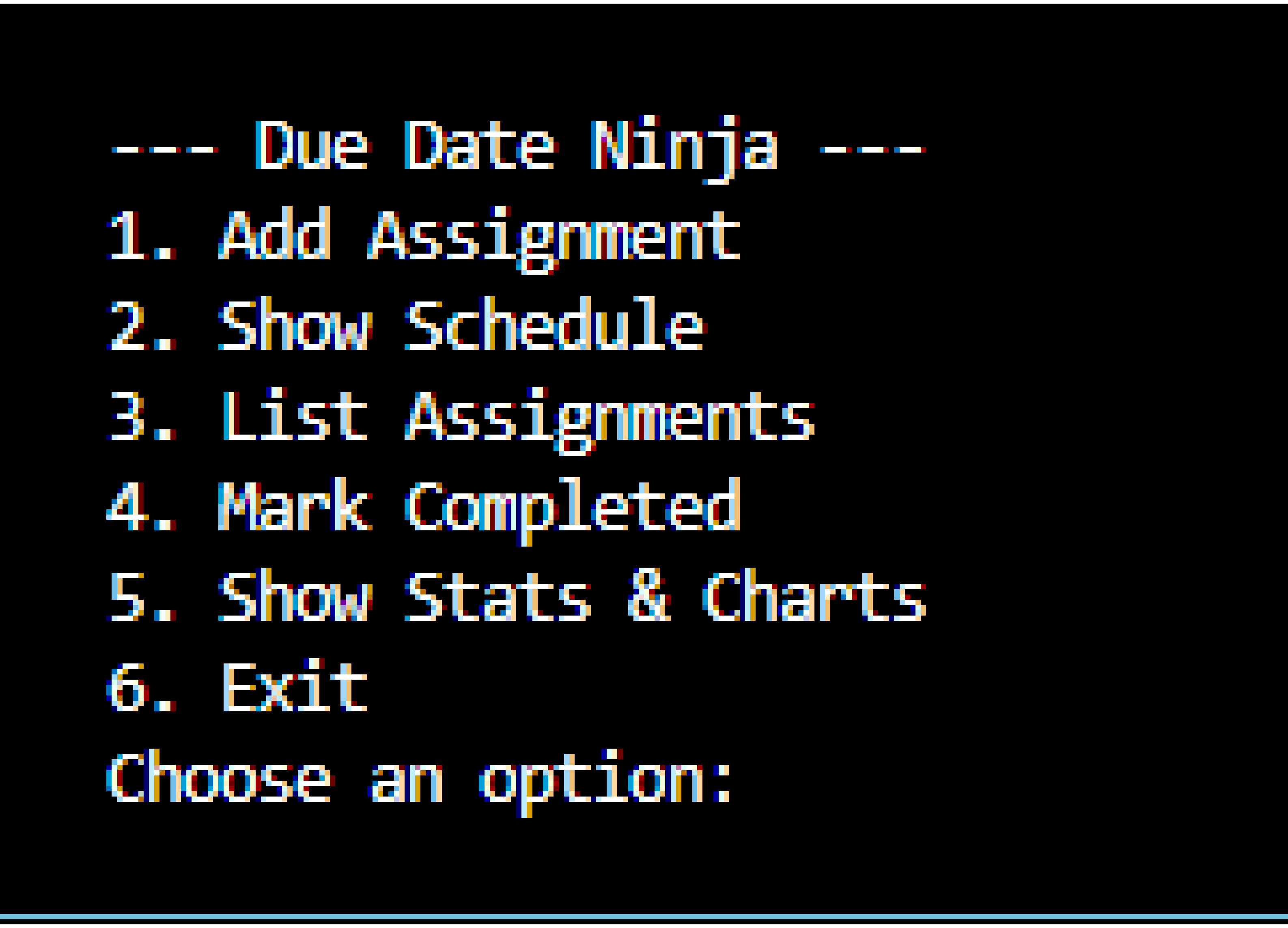
```python
        status = "Completed" if a["completed"] else "Pending"
        print(f"{idx+1}. {a['title']} ({a['subject']}) | {status} | Priority: {a['priority'].upper()}")
    input("\nPress Enter to return to menu...")

def mark_completed():
    list_assignments()
    try:
        idx = int(input("\nEnter assignment number to mark completed: ")) - 1
        if idx < 0 or idx >= len(assignments):
            print("Invalid number!")
            input("\nPress Enter to return to menu...")
            return
        assignments[idx]["completed"] = True
        a = assignments[idx]
        print(f"\n'{a['title']}' marked completed!")
        print(random.choice(funny_messages[a['priority']]))
    except:
        print("Invalid input!")
    input("\nPress Enter to return to menu...")

def show_stats():
    total = len(assignments)
    completed = sum(a["completed"] for a in assignments)
    pending = total - completed
    counts = {"high": 0, "medium": 0, "low": 0}
    for a in assignments:
        if not a["completed"]:
            counts[a["priority"]] += 1

    print(f"\n--- Stats ---")
```



```
    print(f"\n--- Stats ---")
    print(f"Total: {total} | Completed: {completed} | Pending: {pending}")
    print(f"High: {counts['high']} | Medium: {counts['medium']} | Low: {counts['low']}")

    plt.figure(figsize=(5, 5))
    plt.bar(["High", "Medium", "Low"], [counts["high"], counts["medium"], counts["low"]],
            color=[🟥"#8B0000", 🟧"#FF8C00", 🟩"#006400"])
    plt.title("Pending Assignments by Priority")
    plt.ylabel("Number of Assignments")
    plt.show()

    plt.figure(figsize=(5, 5))
    plt.bar(["Completed", "Pending"], [completed, pending], color=[🟩"#006400", 🟥"#8B0000"])
    plt.title("Completed vs Pending Assignments")
    plt.ylabel("Number of Assignments")
    plt.show()

    input("\nPress Enter to return to menu...")

def main():
    while True:
        print("\n--- Due Date Ninja ---")
        print("1. Add Assignment")
        print("2. Show Schedule")
        print("3. List Assignments")
        print("4. Mark Completed")
        print("5. Show Stats & Charts")
        print("6. Exit")
        choice = input("Choose an option: ").strip()
        if choice == "1":
```



```python
            if choice == "1":
                add_assignment()
            elif choice == "2":
                show_schedule()
            elif choice == "3":
                list_assignments()
            elif choice == "4":
                mark_completed()
            elif choice == "5":
                show_stats()
            elif choice == "6":
                break
            else:
                print("Invalid choice!")
                input("\nPress Enter to return to menu...")

if __name__ == "__main__":
    main()

```



```
--- Due Date Ninja ---
1. Add Assignment
2. Show Schedule
3. List Assignments
4. Mark Completed
5. Show Stats & Charts
6. Exit
Choose an option:
```