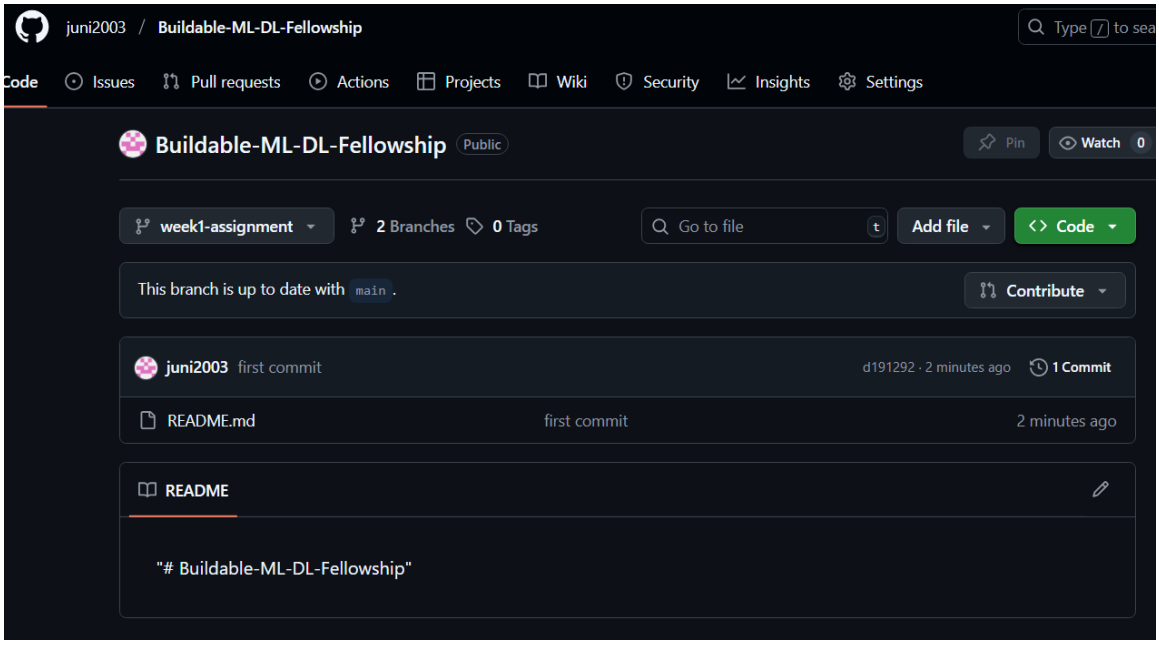
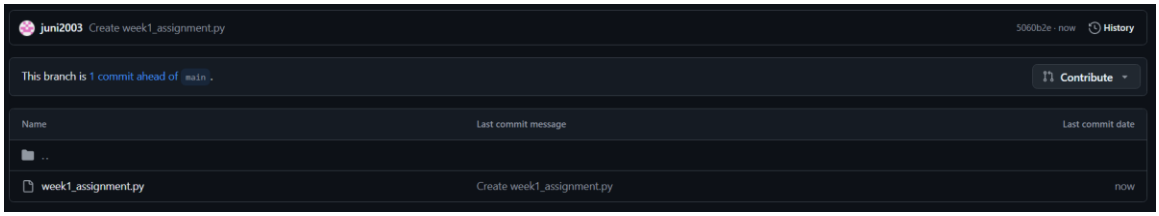
Report

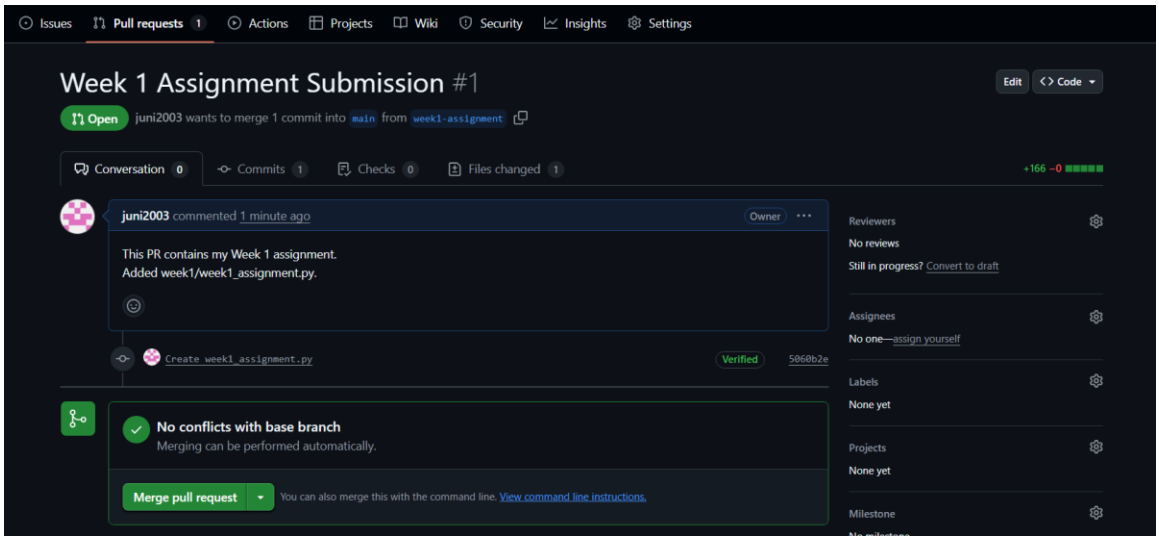
Junaid Mohi Ud Din

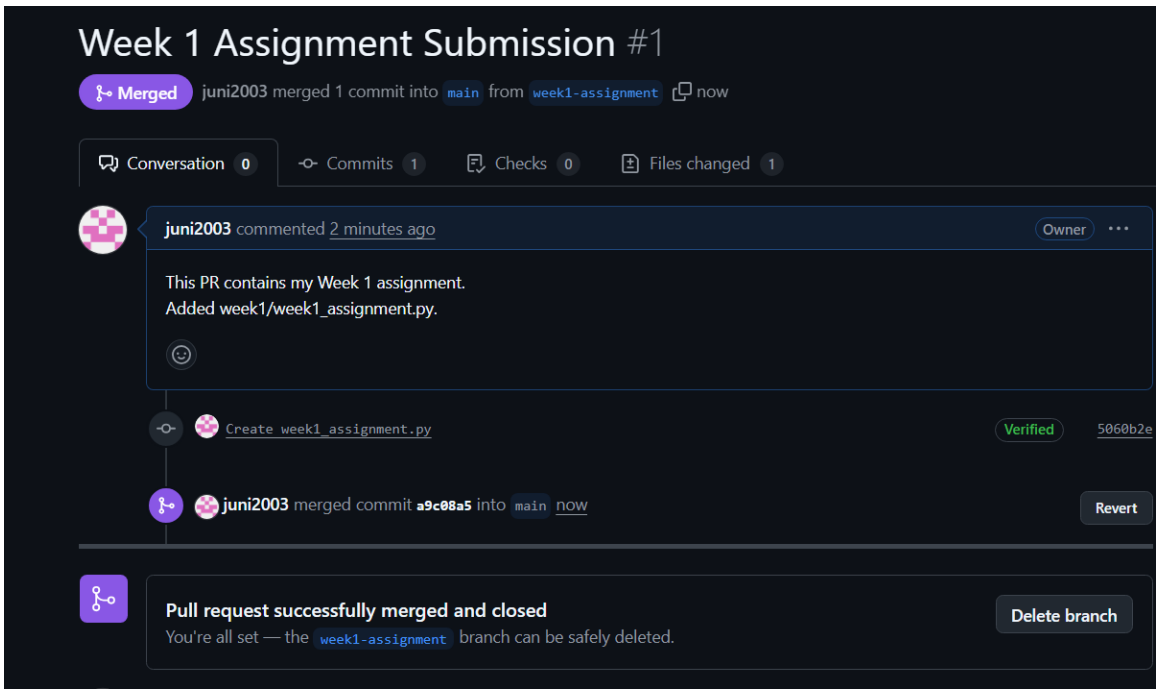
Question 1 – Git & GitHub (Version Control Basics)

Screenshots:





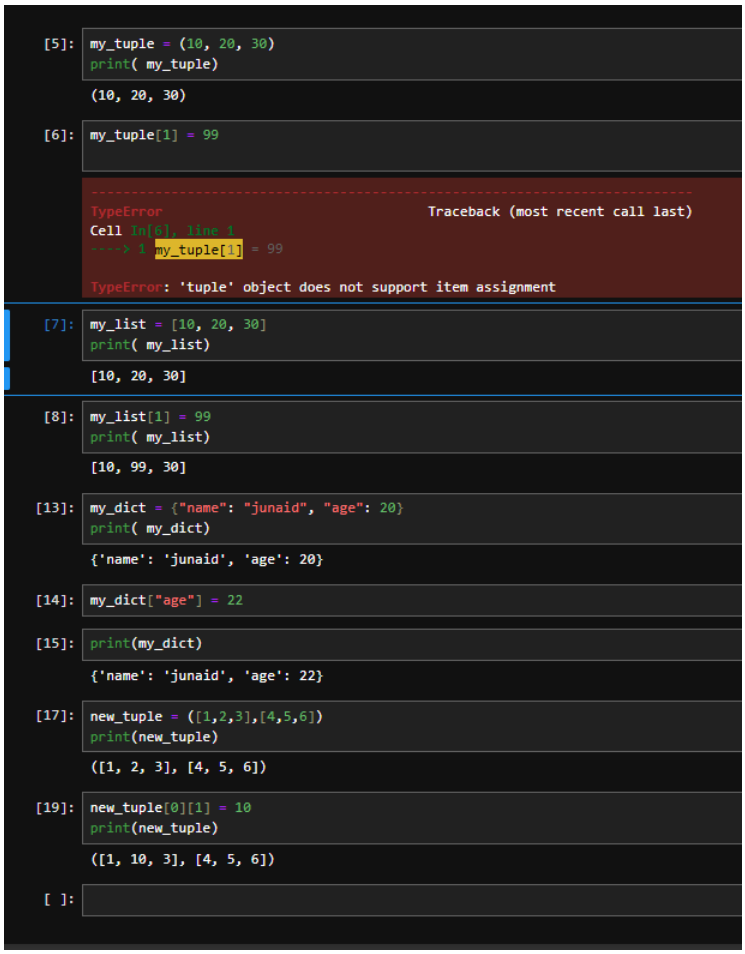




Question 2 – Mutable vs Immutable (Data Structures & Variables)

In this task, I performed different experiments in Python to understand the difference between mutable and immutable data types. Below is my explanation based on the outputs I observed.

Screenshot:



1. Tuple (Immutable)

I created a tuple with three elements (10, 20, 30). When I tried to change one of its elements, Python gave the following error:

TypeError Traceback (most recent call last)
Cell In[6], line 1
----> 1 my_tuple[1] = 99

TypeError: 'tuple' object does not support item assignment

This shows that tuples are immutable, meaning once created, their elements cannot be changed.

2. List (Mutable)

Next, I created a list with three elements [10, 20, 30]. When I changed the second element to 99, it worked successfully. The modified list became [10, 99, 30].

This proves that lists are mutable, and we can modify their elements after creation.

3. Dictionary (Mutable)

I created a dictionary with two key-value pairs: {'name': 'junaid', 'age': 20}. I then updated the value of 'age' to 22. This change was successful.

This shows that dictionaries are also mutable, and the

ir values can be updated by referencing their keys.

4. Tuple with Sub-lists

I created a tuple that contained two sub-lists: ([1, 2, 3], [4, 5, 6]). Although the tuple itself is immutable (I cannot replace the sub-lists), I was able to modify elements inside the sub-lists. For example, I changed the second element of the first list from 2 to 10, and it worked.

This is because while the tuple is immutable, the lists inside it are mutable and can be modified.

Conclusion

From these experiments, I learned that immutable data types (like tuples) cannot be changed after creation, while mutable data types (like lists and dictionaries) can be modified. However, if an immutable structure (like a tuple) contains mutable elements (like lists), the inner mutable elements can still be changed.

Question 3– User information Dictionary

All questions screenshots are present in report and code in present in file week1_assignment.py.

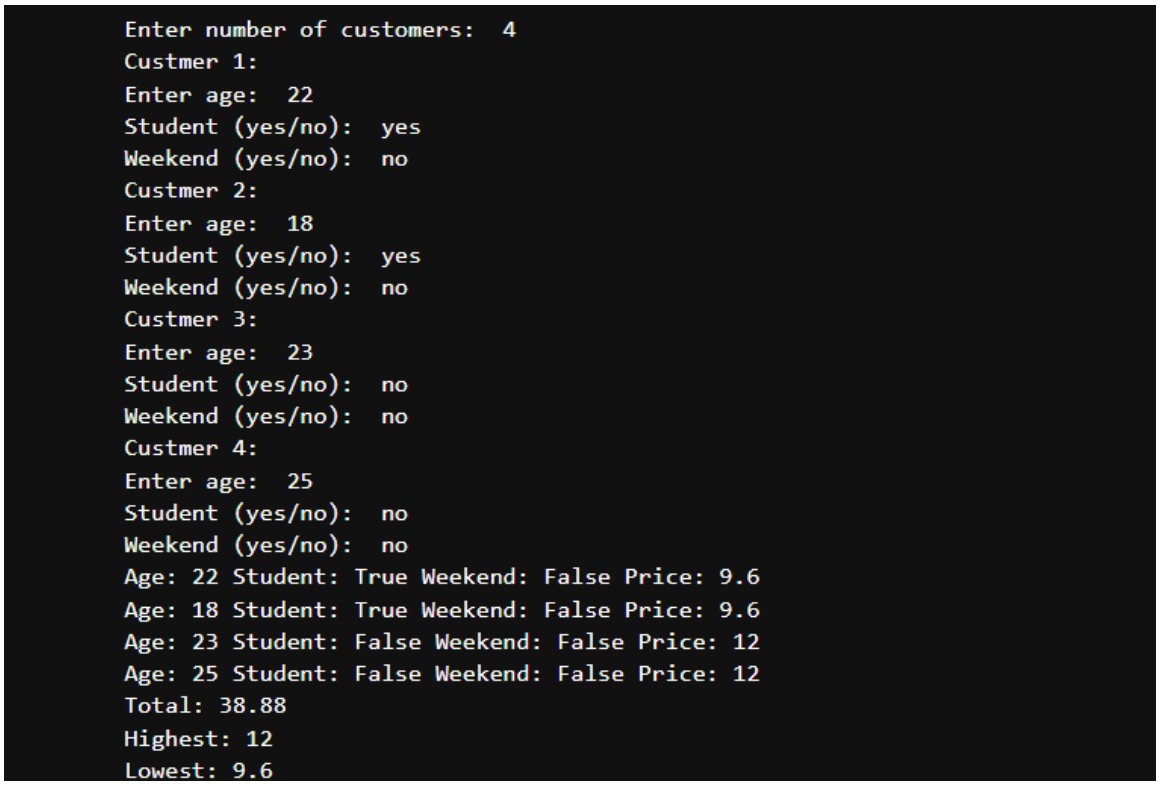
Screenshot:

[9]:
```python
# Question 3 - User Information Dictionary (Validation + Logic)

name = input("Enter name: ")

while True:
    age = int(input("Enter age: "))
    if 0 < age < 100: break


while True:
    email = input("Enter email: ")
    if "@" in email and "." in email and email[0].isalnum() and email[-1].isalnum():
        break

while True:
    fav = int(input("Enter favorite number (1-100): "))
    if 1 <= fav <= 100: break

user = {"name": name, "age": age, "email": email, "fav": fav}
print(f"Welcome {user['name']}! Your account has been registered with email {user['email']}.")
```

```
Enter name:  Junaid
Enter age:  200
Enter age:  101
Enter age:  999
Enter age:  22
Enter email:  junimax@gmialcom
Enter email:  juni.xatti@mgila.com#
Enter email:  juni.xatti@gmail.com
Enter favorite number (1-100):  200
Enter favorite number (1-100):  44
Welcome Junaid! Your account has been registered with email juni.xatti@gmail.com.
```

Question 4– Cinema Ticketing System

Screenshot:



Question 5– Weather Alert System

Screenshot:

[18]:

```
#Question 5 - Weather Alert System

def weather_alert(temp_c, cond):
    if temp_c < 0 and cond == "snowy":
        msg = "Heavy snow alert! Stay indoors."
    elif temp_c > 35 and cond == "sunny":
        msg = "Heatwave warning! Stay hydrated."
    elif cond == "rainy" and temp_c < 15:
        msg = "Cold rain alert! Wear warm clothes."
    else:
        msg = "Normal weather conditions."

    temp_f = (temp_c * 9/5) + 32
    temp_k = temp_c + 273.15
    return msg, round(temp_f, 2), round(temp_k, 2)

t = float(input("Enter temperature in Celsius: "))
c = input("Enter condition (sunny, rainy, snowy, etc.): ").lower()

m, f, k = weather_alert(t, c)
print("Msg:", m)
print("Fahrenheit: ", f)
print("Kelvin: ", k)
```

```
Enter temperature in Celsius:  38
Enter condition (sunny, rainy, snowy, etc.):  sunny
Msg: Heatwave warning! Stay hydrated.
Fahrenheit:  100.4
Kelvin:  311.15
```

Question 6– Sales Analytics

Screenshot:

```
Enter number of days:  3
Need at least 5
Enter number of days:  4
Need at least 5
Enter number of days:  5
Enter sale (Day 1):  94
Enter sale (Day 2):  98
Enter sale (Day 3):  76
Enter sale (Day 4):  64
Enter sale (Day 5):  63
Highest sales day: 98.0
Lowest sales day: 63.0
Median sales: 76.0
```

Question 7– E-commerce Inventory Management

Screenshot:

```
Items: {'pen': 10, 'book': 8, 'bag': 5, 'marker': 12, 'register': 6}
Enter item:  book
Enter qty:  22
Not enough stock for book
Enter item:  book
Enter qty:  5
Enter item:  pointer
Enter qty:  2
Item not found
Inv: {'pen': 10, 'book': 3, 'bag': 5, 'marker': 12, 'register': 6}
Max: marker 12
Min: book 3
```

Question 9– Conceptual Understanding

1. **Explain the difference between AI, Machine Learning, Deep Learning, and Data Science with real-world examples.**

AI (Artificial Intelligence): AI is about making machines act smart like humans. It covers many areas like problem solving, decision making, and natural language. AI is the big field under which ML and DL come.

Example: Siri or Alexa answering questions.

Machine Learning (ML): A part of AI where machines learn from data and improve. It works on patterns in data like it creates pattern based on data it has given, and the more data it has, the better it performs.

Example: Netflix recommending movies based on what you watched.

Deep Learning (DL): A special type of ML using neural networks with many layers. It is good for images, speech, and large datasets but needs a lot of computing power.

Example: Self-driving cars recognizing traffic signs.

Data Science: About collecting, cleaning, and analyzing data to make decisions. It combines statistics, coding, and visualization to understand and use data properly before working on it like create graphs see outliners etc.

Example: A company using sales data to predict which product will sell more.

2. Explain the difference between deep copy and shallow copy.

Shallow Copy: Makes a copy of an object, but if it has other objects inside, it only copies references (not full copies). Example: Copying a list of lists, changes in inner lists affect both.

Deep Copy: Makes a full copy of everything, including inner objects. Example: Copying a list of lists, changes in inner lists don't affect the original.

3. Explain Git branching and why it is important in collaborative development.

Git Branching: Branching means making a separate line of development in Git. You can work on a feature or bug fix without touching the main project.

Importance: In teamwork, branches allow different people to work on different features safely. Later, they merge into the main branch. Example: One student works on login feature, another works on payment feature, both on separate branches.