ASSIGNMENT - 8

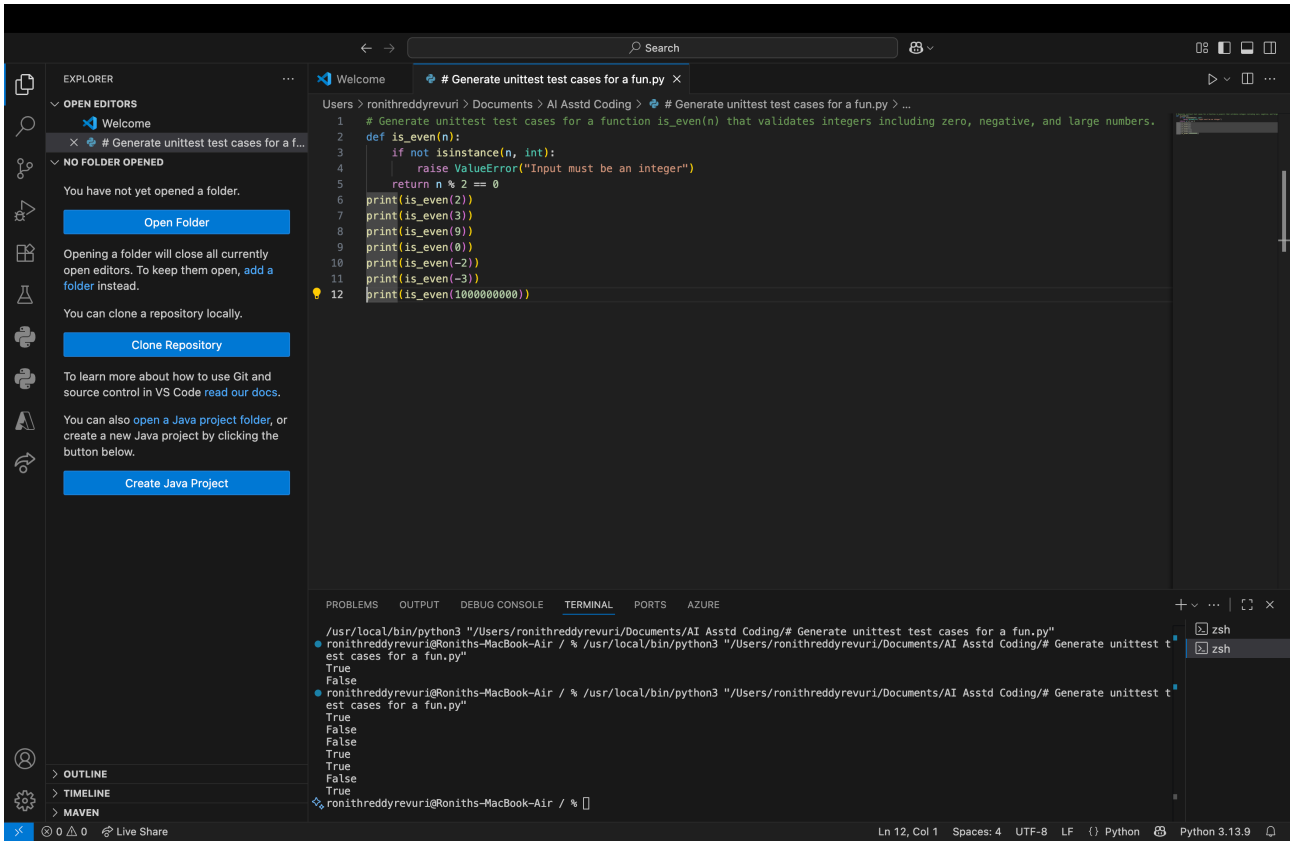
NAME : R.RONITH REDDY
HTNO : 2303A52280

TASK 1 –
TEST-DRIVEN DEVELOPMENT FOR EVEN/ODD NUMBER VALIDATOR

PROMPT USED :

 # GENERATE UNITTEST TEST CASES FOR A FUNCTION IS_EVEN(N) THAT VALIDATES INTEGERS INCLUDING ZERO, NEGATIVE, AND LARGE NUMBERS.

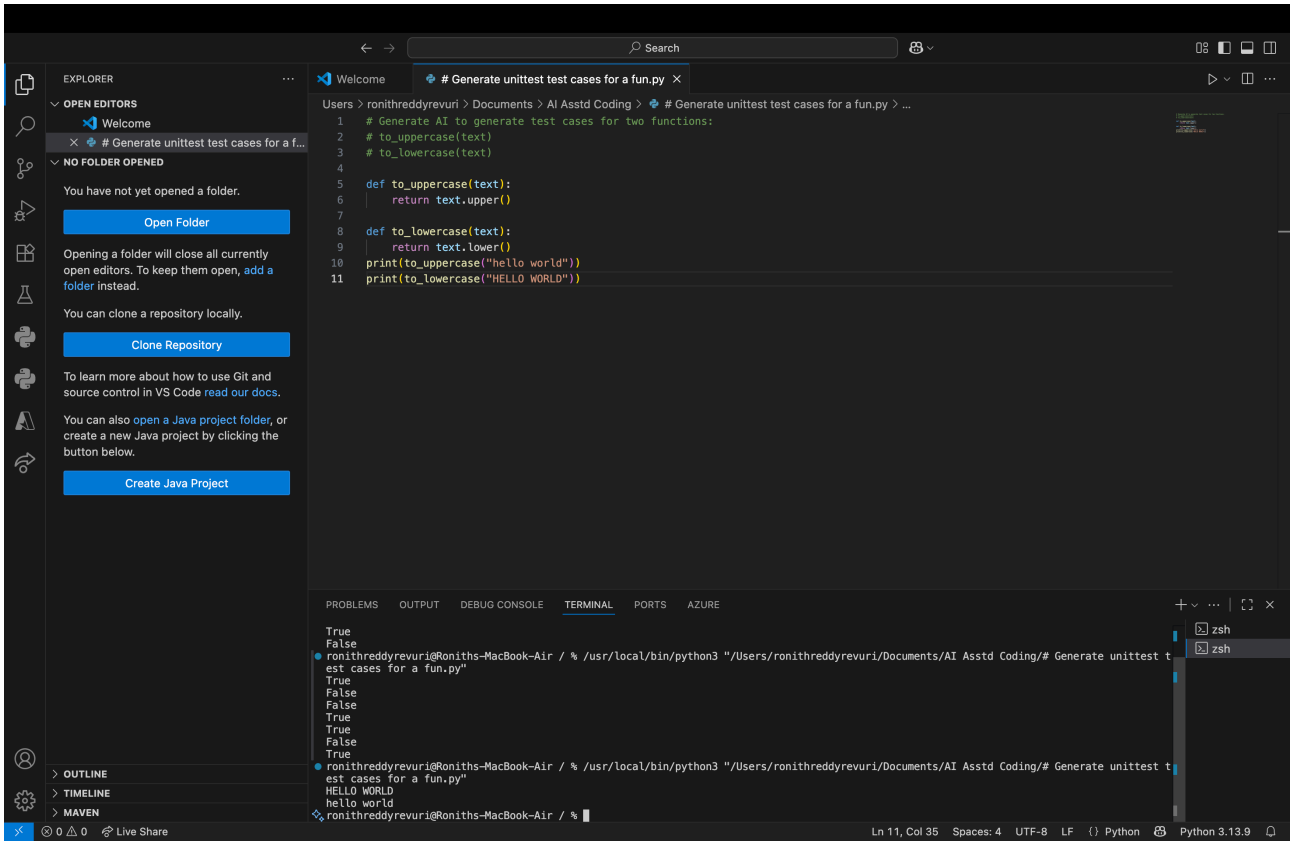
EXPLANATION :
THE FUNCTION USES THE MODULUS OPERATOR TO DETERMINE IF A NUMBER IS EVEN OR ODD. IN ORDER TO KEEP THE PROGRAM SAFE, IT ALSO MAKES SURE THE INPUT IS AN INTEGER.



TASK - 2 :
TEST-DRIVEN DEVELOPMENT FOR STRING CASE CONVERTER

PROMPT :
 # GENERATE AI TO GENERATE TEST CASES FOR TWO FUNCTIONS:
 # TO_UPPERCASE(TEXT)
 # TO_LOWERCASE(TEXT)

EXPLANATION :
TEXT CAN BE CONVERTED TO LOWERCASE OR UPPERCASE USING THESE FUNCTIONS. ADDITIONALLY, THEY SAFELY HANDLE INCORRECT OR EMPTY INPUTS AND VERIFY THAT THE INPUT IS VALID.

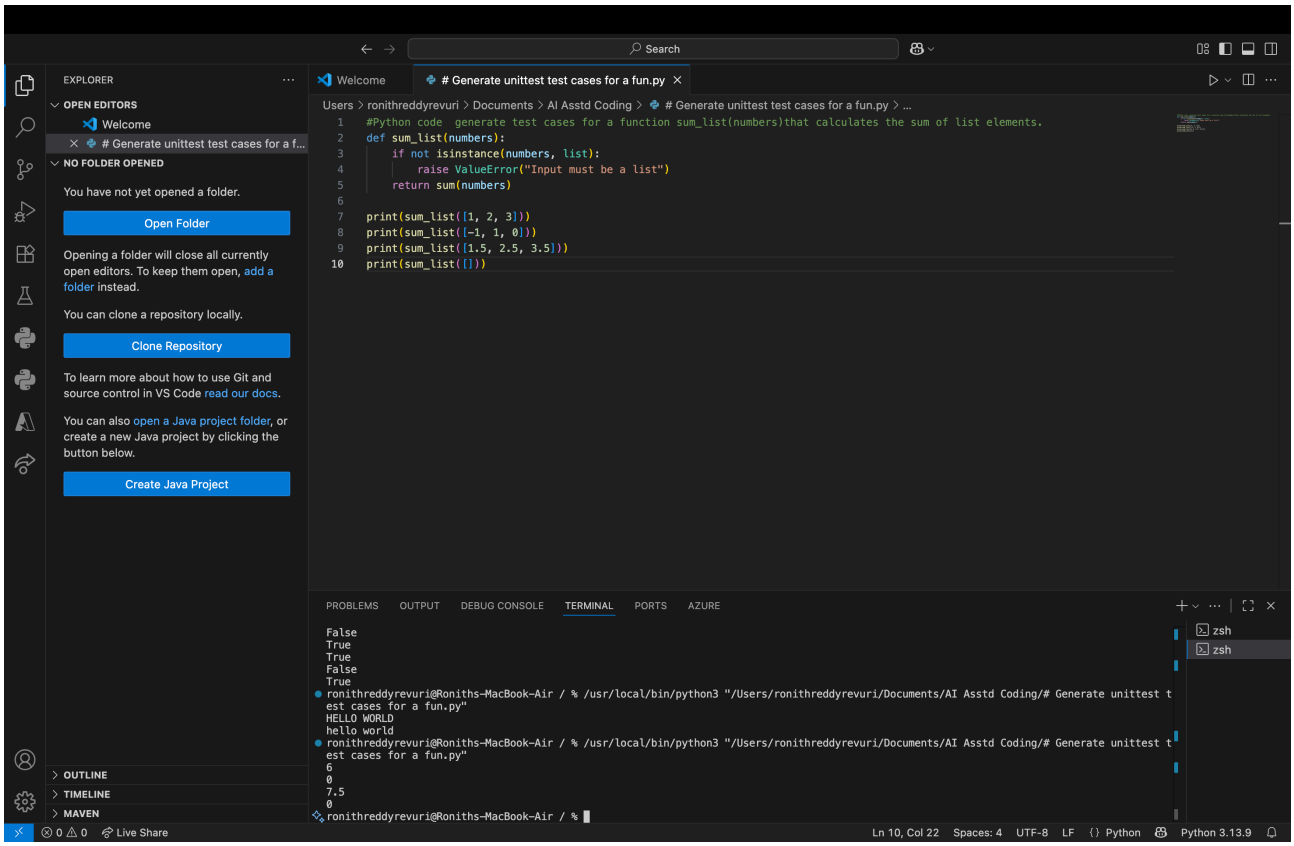


TASK 3 – TEST-DRIVEN DEVELOPMENT FOR LIST SUM CALCULATOR

PROMPT :
PYTHON CODE TO GENERATE TEST CASES FOR A FUNCTION SUM_LIST(NUMBERS)THAT CALCULATES THE SUM OF LIST ELEMENTS.

EXPLANATION :

ONLY NUMERIC VALUES FROM THE LIST ARE ADDED BY THE FUNCTION. IT HANDLES EMPTY LISTS SAFELY AND DISREGARDS NON-NUMERIC ELEMENTS.



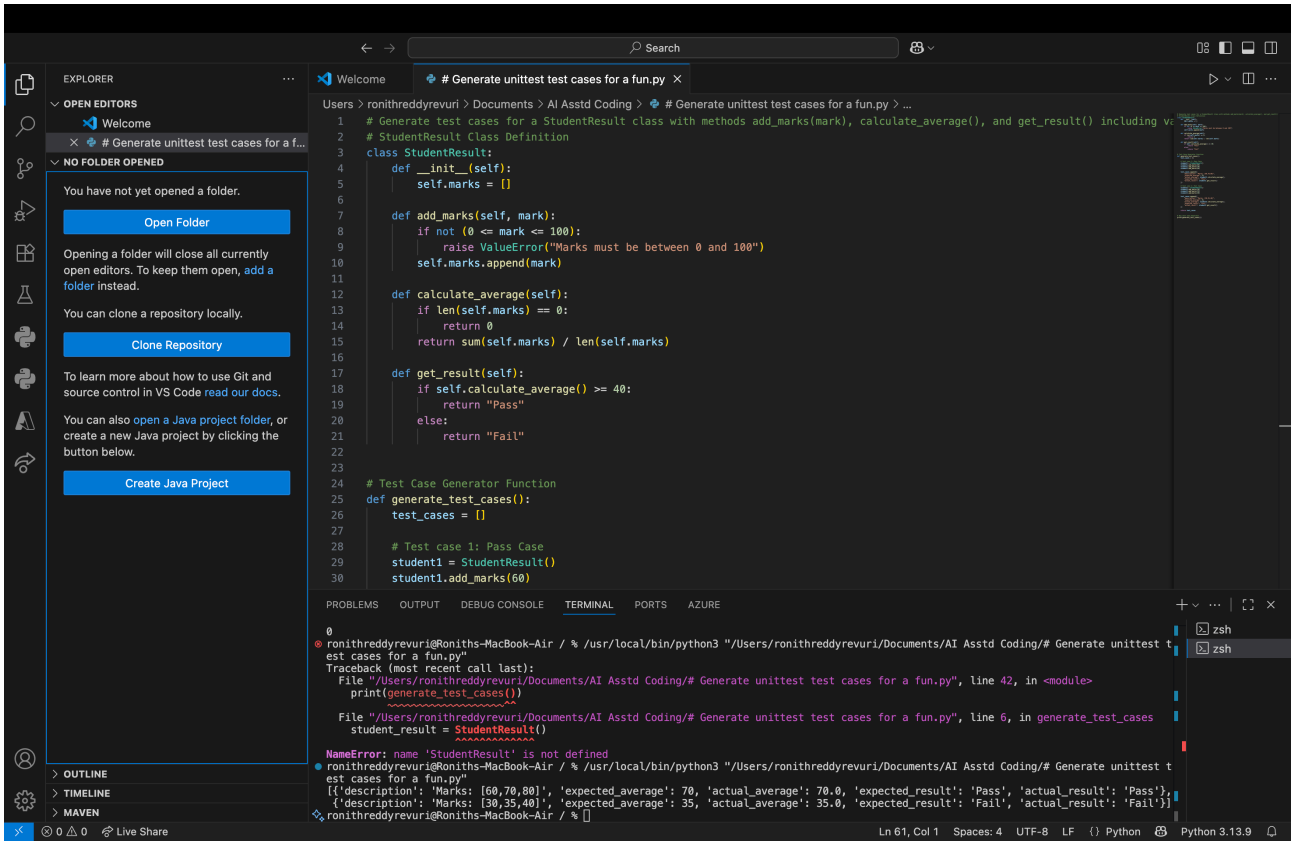
TASK 4 – TEST CASES FOR STUDENT RESULT CLASS

PROMPT :

GENERATE TEST CASES FOR A `STUDENTRESULT` CLASS WITH METHODS `ADD_MARKS(MARK)`, `CALCULATE_AVERAGE()`, AND `GET_RESULT()` INCLUDING VALIDATION RULES.

EXPLANATION :

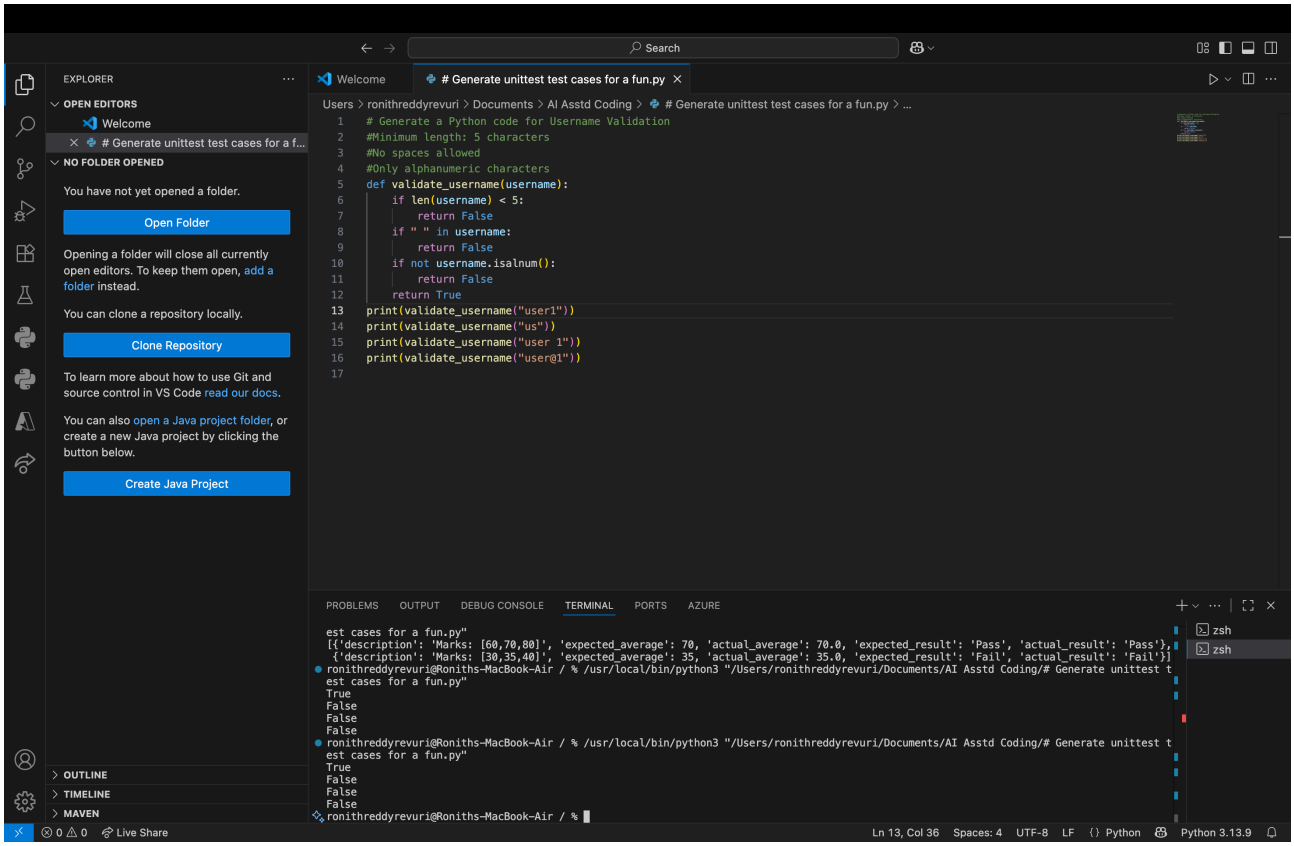
IN ORDER TO DETERMINE WHETHER A STUDENT PASSES OR FAILS, THE CLASS KEEPS TRACK OF THEIR GRADES AND COMPUTES THEIR AVERAGE. ADDITIONALLY, IT CONFIRMS THAT THE MARKS FALL WITHIN THE APPROPRIATE RANGE.



TASK - 5 : TEST-DRIVEN DEVELOPMENT FOR USERNAME VALIDATOR

PROMPT :
 # GENERATE A PYTHON CODE FOR USERNAME VALIDATION
#MINIMUM LENGTH: 5 CHARACTERS
#NO SPACES ALLOWED
#ONLY ALPHANUMERIC CHARACTERS

EXPLANATION:
THE FUNCTION VALIDATES A USERNAME BY CHECKING ITS LENGTH, ENSURING THERE ARE NO SPACES, AND ALLOWING ONLY ALPHANUMERIC CHARACTERS.


Search
EXPLORER
OPEN EDITORS
Welcome
Generate unittest test cases for a f...
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
Opening a folder will close all currently open editors. To keep them open, add a folder instead.
You can clone a repository locally.
Clone Repository
To learn more about how to use Git and source control in VS Code read our docs.
You can also open a Java project folder, or create a new Java project by clicking the button below.
Create Java Project
OUTLINE
TIMELINE
MAVEN
Welcome
Generate unittest test cases for a fun.py
Users > ronithreddyrevuri > Documents > AI Asstd Coding > # Generate unittest test cases for a fun.py > ...
1 # Generate a Python code for Username Validation
2 #Minimum length: 5 characters
3 #No spaces allowed
4 #Only alphanumeric characters
5 def validate_username(username):
6 if len(username) < 5:
7 return False
8 if " " in username:
9 return False
10 if not username.isalnum():
11 return False
12 return True
13 print(validate_username("user1"))
14 print(validate_username("us"))
15 print(validate_username("user 1"))
16 print(validate_username("user@1"))
17
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
AZURE
zsh
zsh
est cases for a fun.py"
[{'description': 'Marks: [60,70,80]', 'expected_average': 70, 'actual_average': 70.0, 'expected_result': 'Pass', 'actual_result': 'Pass'},
{'description': 'Marks: [30,35,40]', 'expected_average': 35, 'actual_average': 35.0, 'expected_result': 'Fail', 'actual_result': 'Fail'}]
ronithreddyrevuri@Roniths-MacBook-Air / % /usr/local/bin/python3 "/Users/ronithreddyrevuri/Documents/AI Asstd Coding/# Generate unittest t
est cases for a fun.py"
True
False
False
False
ronithreddyrevuri@Roniths-MacBook-Air / % /usr/local/bin/python3 "/Users/ronithreddyrevuri/Documents/AI Asstd Coding/# Generate unittest t
est cases for a fun.py"
True
False
False
False
ronithreddyrevuri@Roniths-MacBook-Air / %
0 0
Live Share
Ln 13, Col 36
Spaces: 4
UTF-8
LF
Python
Python 3.13.9

CONCLUSION :
THIS LAB HELPED IN UNDERSTANDING TEST-DRIVEN DEVELOPMENT USING AI TOOLS. BY WRITING TEST CASES FIRST AND THEN IMPLEMENTING THE CODE, WE ENSURED CORRECTNESS, VALIDATION, AND RELIABILITY. IT IMPROVED OUR UNDERSTANDING OF TESTING AND WRITING CLEAN, ERROR-FREE PROGRAMS.