SR UNIVERSITY

School of Computer Science and Artificial Intelligence

Lab Assignment # 8.5

Program	**: B. Tech (CSE)**
Specialization	**: -**
Course Title	**: AI Assisted Coding**
Course Code	**: 23CS002PC304**
Semester	**II**
Academic Session	**: 2025-2026**
Name of Student	**: R.Akash Reddy**
Enrollment No.	**: 2403A51L30**
Batch No.	**51**
Date	**: 30/01/26**

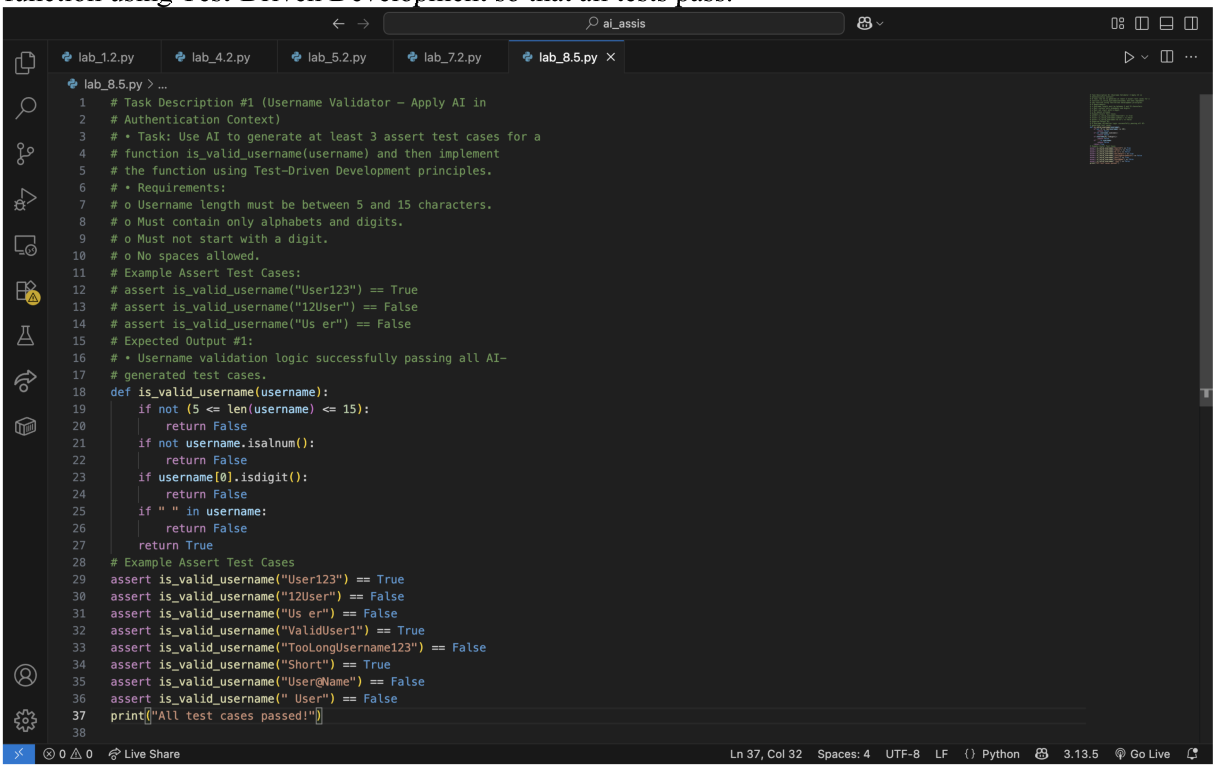
Submission Starts here

Screenshots:

Task Description #1 (Username Validator – Apply AI in Authentication Context)

Prompt:

Generate at least 3 assert-based test cases for a Python function is_valid_username(username) with the following rules: - Username length must be between 5 and 15 characters. - Only alphabets and digits allowed. - Username must not start with a digit. - No spaces allowed. Then implement the function using Test-Driven Development so that all tests pass.



SR UNIVERSITY

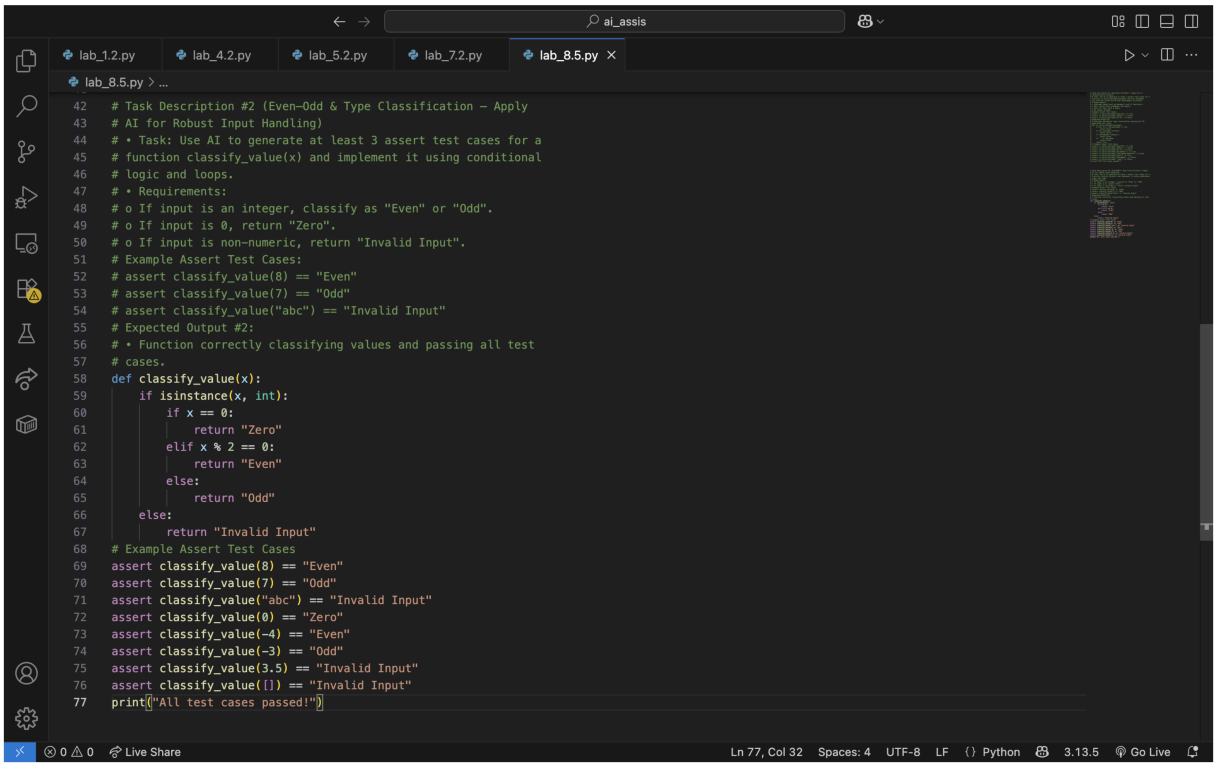
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                                                    Python

 /usr/local/bin/python3 /Users/akash/Desktop/ai_assis/lab_8.5.py
(base) akash@AKASHs-MacBook-Air ai_assis % /usr/local/bin/python3 /Users/akash/Desktop/ai_assis/lab_8.5.py
 All test cases passed!
(base) akash@AKASHs-MacBook-Air ai_assis %
```

Task Description #2 (Even–Odd & Type Classification – Apply AI for Robust Input Handling)

Prompt:
Generate at least 3 assert test cases for a Python function classify_value(x) and implement the function using conditional logic and loops with the following rules:

- If input is integer, return "Even" or "Odd".
- If input is 0, return "Zero".
- If input is non-numeric, return "Invalid Input".
Ensure all tests pass.




```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                                                    Python

 /usr/local/bin/python3 /Users/akash/Desktop/ai_assis/lab_8.5.py
(base) akash@AKASHs-MacBook-Air ai_assis % /usr/local/bin/python3 /Users/akash/Desktop/ai_assis/lab_8.5.py
 All test cases passed!
(base) akash@AKASHs-MacBook-Air ai_assis %
```

SR UNIVERSITY

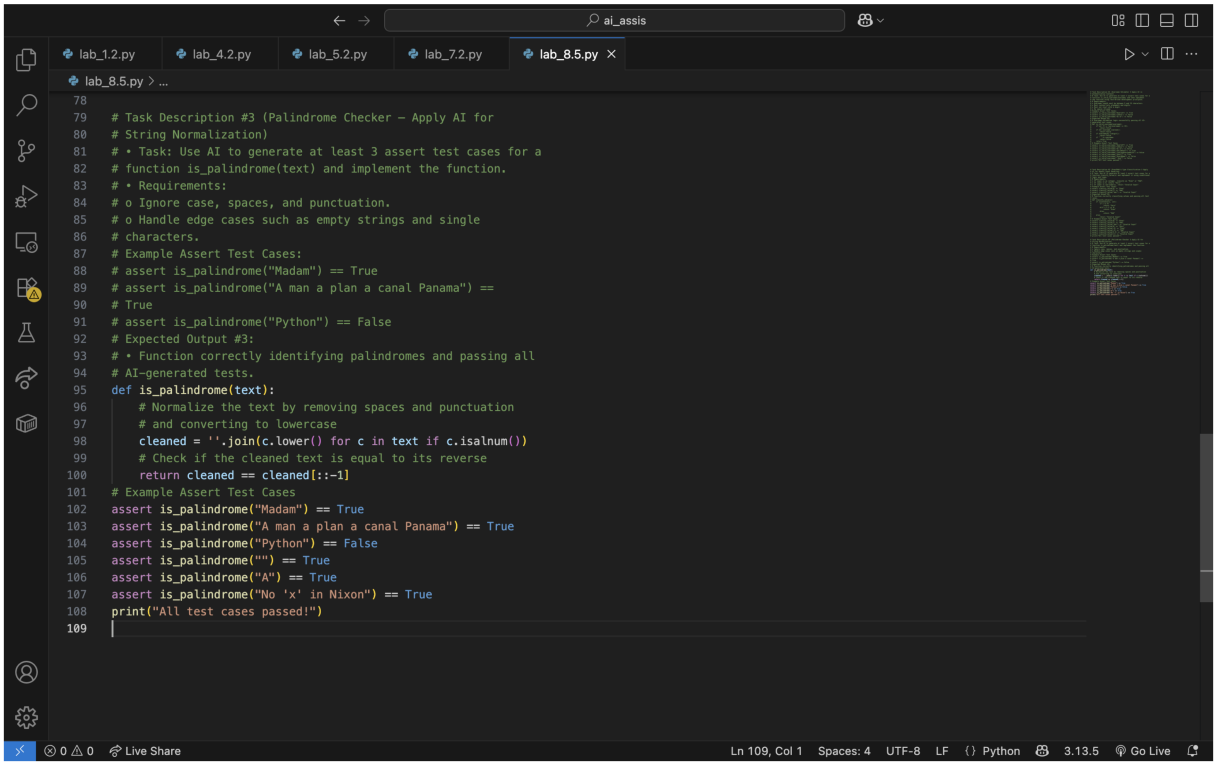
Code:

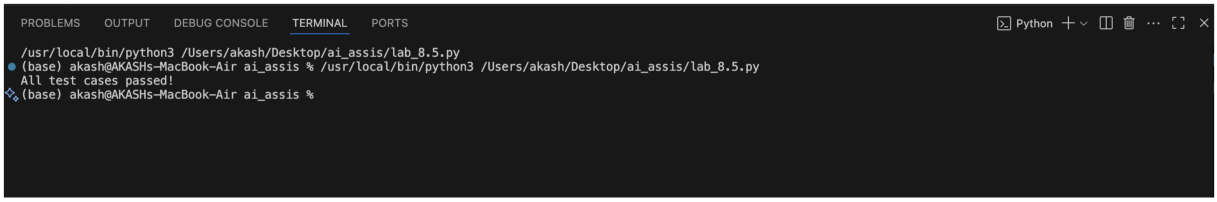
Task Description #3 (Palindrome Checker – Apply AI for String Normalization)

Prompt:

Generate at least 3 assert test cases for a function is_palindrome(text) and implement the function.

Requirements:
- Ignore case, spaces, and punctuation.
- Handle empty string and single characters.
- Ensure all tests pass.





SR UNIVERSITY

Code:

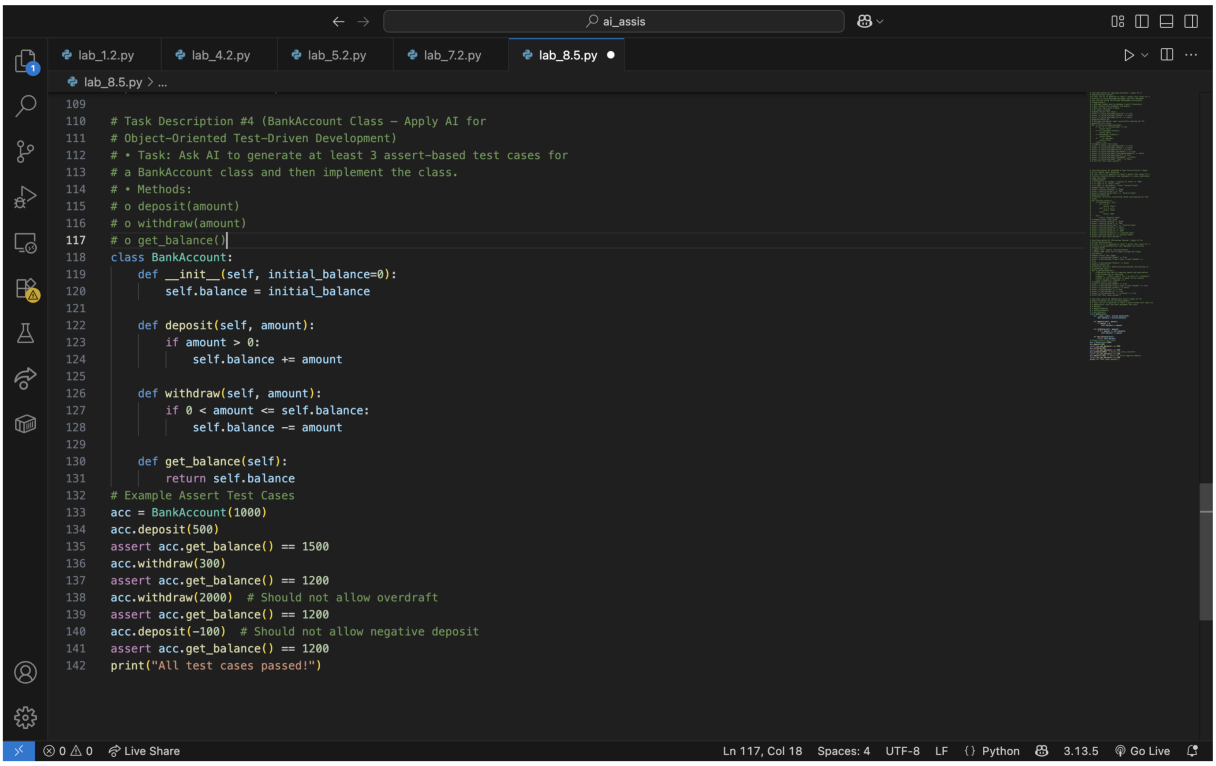
Task Description #4 (BankAccount Class – Apply AI for Object-Oriented Test-Driven Development)

Prompt:

Generate at least 3 assert-based test cases for a Python BankAccount class and then implement the class.

Methods required:
- deposit(amount)
- withdraw(amount)
- get_balance()

Ensure implementation passes all tests.



SR UNIVERSITY

Code:

Task Description #5 (Email ID Validation – Apply AI for Data Validation)

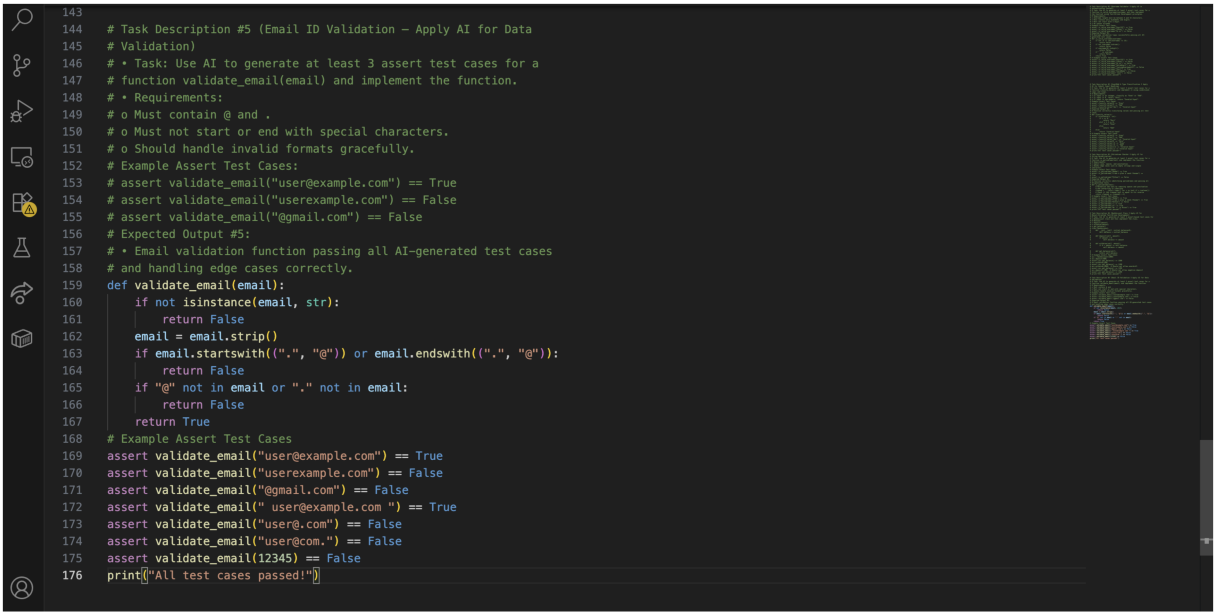
Prompt:

Generate at least 3 assert test cases for a function validate_email(email) and implement the function.

Rules:
- Must contain @ and .
- Must not start or end with special characters.
- Handle invalid formats gracefully.

Ensure all tests pass.

Code:



```
143
144  # Task Description #5 (Email ID Validation – Apply AI for Data
145  # Validation)
146  # • Task: Use AI to generate at least 3 assert test cases for a
147  # function validate_email(email) and implement the function.
148  # • Requirements:
149  # o Must contain @ and .
150  # o Must not start or end with special characters.
151  # o Should handle invalid formats gracefully.
152  # Example Assert Test Cases:
153  # assert validate_email("user@example.com") == True
154  # assert validate_email("userexample.com") == False
155  # assert validate_email("@gmail.com") == False
156  # Expected Output #5:
157  # • Email validation function passing all AI-generated test cases
158  # and handling edge cases correctly.
159  def validate_email(email):
160      if not isinstance(email, str):
161          return False
162      email = email.strip()
163      if email.startswith((".", "@")) or email.endswith((".", "@")):
164          return False
165      if "@" not in email or "." not in email:
166          return False
167      return True
168  # Example Assert Test Cases
169  assert validate_email("user@example.com") == True
170  assert validate_email("userexample.com") == False
171  assert validate_email("@gmail.com") == False
172  assert validate_email(" user@example.com ") == True
173  assert validate_email("user@.com") == False
174  assert validate_email("user@com.") == False
175  assert validate_email(12345) == False
176  print("All test cases passed!")
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                    Python

/usr/local/bin/python3 /Users/akash/Desktop/ai_assis/lab_8.5.py
(base) akash@AKASHs-MacBook-Air ai_assis % /usr/local/bin/python3 /Users/akash/Desktop/ai_assis/lab_8.5.py
All test cases passed!
(base) akash@AKASHs-MacBook-Air ai_assis %
```