



Aditya Ambadkar

Python - 18th June's report

Submitted on Jun 24 2024 18:15:52 IST

**10.0 (26.3%)**

scored out of 38

**Failed**in the assessment (Cut-off score \geq 50%)**20**

problems attempted out of 20

**Severe Violation**

flagged by DoSelect proctoring engine

Test time analysis

**2h 0m 0s**

time taken for completion

**Jun 24 2024 16:08:04 IST**

test invite time

**Jun 24 2024 16:15:52 IST**

test start time

**Jun 24 2024 18:15:52 IST**

test end time

Performance summary

**10**

solutions accepted

**10**

solutions rejected

Proctor analysis

**1**

browser used

**3**

navigation violation

**4**

webcam violations

**0.23 mins**

test window violation

Webcam Violation - flagged by DoSelect Proctoring Engine due to below reasons

Total Frames Captured : 7188

Frames with Matching Faces | **7017**

Frames with Multiple Faces	147
Frames with Different Face	8
Frames with No Face	16

Total Frames Missing : 12

Webcam not detected	0
Test-taker closing the tab	0
Network Issues	0
Other factors*	12

Total Webcam Violations : 4

Set of 10 back-to-back Suspicious Frames** | **3**

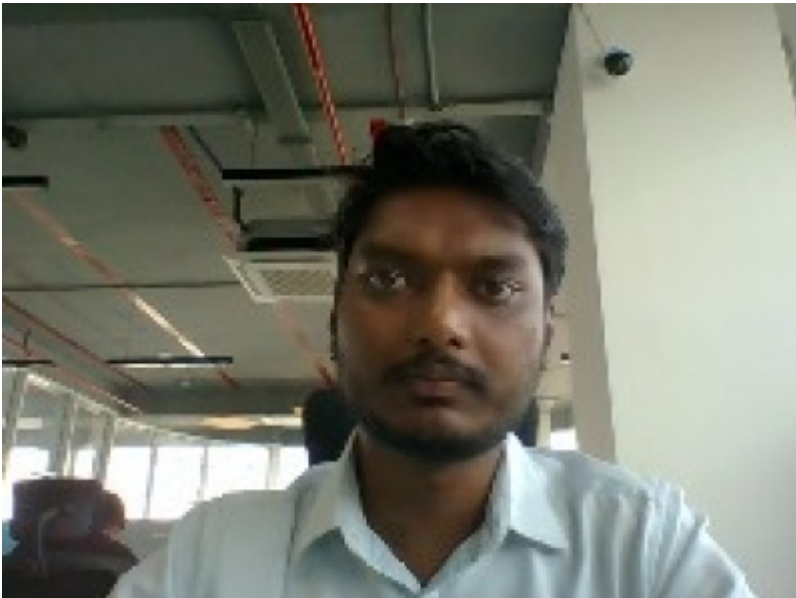
Set of 10 back-to-back Missing frames | **0**

Suspicious Frames**/Missing Frames detected in less than 5% of test duration | **1**

* Missing frames due to other factors such as test-taker's system issues etc

** Suspicious frames includes Multiple Faces, Different Faces and No Face

Identity Image



Solutions

Problem Name	Problem Type	Status	Score
Technical MCQ			10.0 / 18 (55.56%)
Python: Filter List	MCQ	ACCEPTED	1.0 / 1
Flask Decorators	MCQ	ACCEPTED	1.0 / 1
Predict the 'Exceptional' Output	MCQ	ACCEPTED	1.0 / 1
Flask Forms	MCQ	REJECTED	0.0 / 1
Abstract Models	MCQ	ACCEPTED	1.0 / 1
Connecting the lists	MCQ	REJECTED	0.0 / 1
Python: Instance	MCQ	REJECTED	0.0 / 1
Python: Multilevel	MCQ	ACCEPTED	1.0 / 1
Programming: Output of Python Class Methods	MCQ	REJECTED	0.0 / 1
Django: Serializer	MCQ	ACCEPTED	1.0 / 1
Logged Only	MCQ	REJECTED	0.0 / 1
Objects and classes	MCQ	ACCEPTED	1.0 / 1
Python: Deeply Nested	MCQ	REJECTED	0.0 / 1
Programming: Output of Python Class Method	MCQ	REJECTED	0.0 / 1
Regular Email id	MCQ	ACCEPTED	1.0 / 1
Referred	MCQ	REJECTED	0.0 / 1
Python: Default Dictionary Values	MCQ	ACCEPTED	1.0 / 1
Python: New Tuple	MCQ	ACCEPTED	1.0 / 1

Problem Name	Problem Type	Status	Score
Coding			0.0 / 20 (0.00%)
Increase Array	Coding	REJECTED	0.0 / 10
Django: Book Collection	Project Based Backend	REJECTED	0.0 / 10

Technology used



Detailed Report

Section 1 : Technical MCQ

Problem 1 : Python: Filter List

MCQ

SCORE: 1

What is the output of the following code?

```
list(filter(lambda x: x > 5, [3, 4, 5, 6, 7]))
```

Solution

ACCEPTED

SCORE: 1.0 / 1

CANDIDATE'S ANSWER	CORRECT ANSWER
[6, 7]	
[3, 4, 5]	
[5,6,7]	
[4,5,6,7]	

Problem 2 : Flask Decorators

MCQ

SCORE: 1

You have a function that returns a dictionary. You want to return that dictionary as a JSON response without changing the function. So you write a decorator. Which of the following may represent your decorator?

Solution

ACCEPTED

SCORE: 1.0 / 1

```
def send_json_response(func):
    def wrapper(*args, **kwargs):
        value = func(*args, **kwargs)
        response = make_response(json.dumps(value))
        return func
    return send_json_response
```

CANDIDATE'S ANSWER

```
def send_json_response(func):
    def wrapper(*args, **kwargs):
        value = func(*args, **kwargs)
        response = make_response(json.dumps(value))
        return response
    return wrapper
```

CORRECT ANSWER

```
def send_json_response(func):
    def wrapper(*args, **kwargs):
        value = func(*args, **kwargs)
        response = make_response(json.dumps(value))
        return send_json_response
    return response
```

```
def send_json_response(func):
    def wrapper(*args, **kwargs):
        value = func(*args, **kwargs)
        return value
    return wrapper
```


Problem 3 : Predict the 'Exceptional' Output

MCQ

SCORE: 1

Select the correct output for the code given below :

```
try :  
    var1 = 4  
    if var1 < 4 :  
        var2 = var1/(var1-3)  
    print ("Value of var2 = ", var2)  
except(ZeroDivisionError, NameError):  
    print ("\nThe error that occurred was handled")
```

Solution

ACCEPTED

SCORE: 1.0 / 1

Value of var2 = 4

Value of var2 = 3

CANDIDATE'S ANSWER

CORRECT ANSWER

The error that occurred was handled

DivideByZeroError

Problem 4 : Flask Forms

MCQ

SCORE: 1

Assume the following URL, as a **POST** request to the flask server running on

```
http://localhost:5000/users?username=random_username
```

Which of the following can be used to access the URL parameter **username** in the view corresponding to the **/users** endpoint?

Solution

REJECTED

SCORE: 0.0 / 1

CORRECT ANSWER

```
request.args.get("username")
```

CANDIDATE'S ANSWER

```
request.form["username"]
```

```
request.form.get("username")
```

```
request.form.post("username")
```

Problem 5 : Abstract Models

MCQ

SCORE: 1

Why do we use abstract models in Django?

Solution

ACCEPTED

SCORE: 1.0 / 1

CANDIDATE'S ANSWER

To use the abstract model's properties in other models via inheritance.

CORRECT ANSWER

To automatically use caching on the model that is abstracted.

To generate a view and a form for the model automatically.

To create a table with same name as the abstract model in the database.

Problem 6 : Connecting the lists

MCQ

SCORE: 1

Let us say you have the Fibonacci series in two lists:

Fib1=[1,1,2,3,5,8]

Fib2=[13,21,34]

How will you **create a list** in python that has all the above elements in one list?

(Select all that apply)

Solution

REJECTED

SCORE: 0.0 / 1

CANDIDATE'S ANSWER

```
Fib1.append(Fib2)
```

CORRECT ANSWER

```
Fib2.extend(Fib1)
```

CANDIDATE'S ANSWER

CORRECT ANSWER

```
Fib1.extend(Fib2)
```

```
Fib1.cat()
```

Problem 7 : Python: Instance

MCQ

SCORE: 1

What is the difference between a **class method** and an **instance method** in Python OOP?

Solution

REJECTED

SCORE: 0.0 / 1

CORRECT ANSWER

A class method is called on the class itself, whereas an instance method is called on an instance of the class.

A class method can access instance variables, whereas an instance method cannot.

A class method is defined using the @classmethod decorator, whereas an instance method is not.

CANDIDATE'S ANSWER

A class method is used for object instantiation, whereas an instance method is used for object manipulation.

Problem 8 : Python: Multilevel

MCQ

SCORE: 1

Which of the following statements is correct about **multilevel inheritance** in Python?

Solution

ACCEPTED

SCORE: 1.0 / 1

Python does not support multilevel inheritance.

Python only supports single inheritance.

Python only supports multiple inheritances.

CANDIDATE'S ANSWER

Python supports multilevel inheritance.

CORRECT ANSWER

Problem 9 : Programming: Output of Python Class Methods

MCQ

SCORE: 1

Consider the following Python code:

```
class MyClass:
    def __init__(self, x):
        self.x = x
    def my_method(self):
        self.x += 5

obj1 = MyClass(10)
obj2 = MyClass(20)
obj1.my_method()
obj2.my_method()
print(obj1.x, obj2.x)
```

What would be the output of this code?

Solution

REJECTED

SCORE: 0.0 / 1

CANDIDATE'S ANSWER

10 20

CORRECT ANSWER

15 25

20 30

25 25

Problem 10 : Django: Serializer

MCQ

SCORE: 1

A **Django REST framework** is being used by a developer to create an API for a movie database. He wants to develop an endpoint that would let users search for movies by title and genre. He has a movie model with fields for title, genre, release date, and director.

For the developer to effectively handle this search capability, which serializer class should they use?

Solution

ACCEPTED

SCORE: 1.0 / 1

CANDIDATE'S ANSWER	CORRECT ANSWER
Custom Serializer with custom validation	
Custom serializer with Built-in text validation	
HyperlinkedModelSerializer	
SerializerMethodField within a ModelSerializer	

Problem 11 : Logged Only

MCQ

SCORE: 1

You want to add user authentication in a Django project such that one user is logged in only once. This should avoid two different persons logging into the system with the same login/password. Which of the following is a possible way to do this?

(Select all options that apply)

Solution

REJECTED

SCORE: 0.0 / 1

CANDIDATE'S ANSWER

checking in the django_session table before logging in the user

CORRECT ANSWER

checking in the django_session table after logging in the user

CORRECT ANSWER

shut down the browser and drop the cookies

Problem 12 : Objects and classes

MCQ

SCORE: 1

What will be the output on the successful execution of the below code?

```
class Character:

    travelspeed = 10
    def __init__(s, name, w, h, x, y):
        s.name = name
        s.w = w
        s.h = h
        s.x = x
        s.y = y

    def move(s, xamount, yamount):
        s.x += xamount
        s.y += yamount

character_0 = Character('char_0', 50, 100, 100, 100)
character_0.name
character_0.name= 'char_1'
character_0.name

character_0.move(50, 100)
print(character_0.x)
print(character_0.y)
```

Solution

ACCEPTED

SCORE: 1.0 / 1

CANDIDATE'S ANSWER	CORRECT ANSWER
150 200	
250 100	
100 150	
200 250	

Problem 13 : Python: Deeply Nested

MCQ

SCORE: 1

```
for i in range(3):  
    for j in range(2):  
        if j == 1:  
            break  
        print(i, j)  
    else:  
        print("End")
```

What will be the **output** of the following Python code snippet?

Solution

REJECTED

SCORE: 0.0 / 1

CORRECT ANSWER

```
0 0  
1 0  
2 0  
End
```

CANDIDATE'S ANSWER

```
0 0  
1 0  
2 0
```

```
End
```

```
0 0  
0 1  
0 2  
End
```

Problem 14 : Programming: Output of Python Class Method

MCQ

SCORE: 1

Consider the following Python code:

```
class MyClass:
    def __init__(self):
        self.x = 10
    def my_method(self):
        self.x += 5
obj = MyClass()
obj.my_method()
print(obj.x)
```

What would be the output of this code?

Solution

REJECTED

SCORE: 0.0 / 1

10

CANDIDATE'S ANSWER

5

CORRECT ANSWER

15

TypeError

Problem 15 : Regular Email id

MCQ

SCORE: 1

The following code is executed in Python.

Fill in the blanks to fetch the **email-id** in **doselect.txt**.

```
import re
hand =open('doselect.txt')
for line in hand:
    line = line.rstrip()
    x = re.findall('_____', line)
    if len(x) > 0:
        print(x)
```

Solution

ACCEPTED

SCORE: 1.0 / 1

CANDIDATE'S ANSWER

[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}

CORRECT ANSWER

[A-Za-z0-9._%+-]+@[A-Za-z]+\.[A-Za-z]{2,3}

[A-Za-z0-9._%+-]+@[A-Za-z]+\.[A-Za-z0-9]{2,}

[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z0-9.-]{2,}

Problem 16 : Referred

MCQ

SCORE: 1

Why Django is mostly referred to as the **MTV** framework?

Solution

REJECTED

SCORE: 0.0 / 1

CANDIDATE'S ANSWER

Django prefers to use it's own logic in the implementation, the framework considers handling the Controller part of the MVC itself.

CORRECT ANSWER

Django prefers to position the interface or relationship between the data and contains everything related to data access and validation.

CANDIDATE'S ANSWER

Django prefers to the presentation layer that handles the presentation logic in the framework and basically controls what should be displayed and how it should be displayed to the user.

CORRECT ANSWER

Django prefers to uses the terminology Templates for Views and Views for Controller

Problem 17 : Python: Default Dictionary Values

MCQ

SCORE: 1

```
fruits = {"apple": 10, "orange": 5}  
quantity = fruits.get("mango", 0)
```

What will be the value of "quantity" after executing the code?

Solution

ACCEPTED

SCORE: 1.0 / 1

5

10

mango

CANDIDATE'S ANSWER

0

CORRECT ANSWER

Problem 18 : Python: New Tuple

MCQ

SCORE: 1

What will be the **output** of the following code?

```
myTuple = (1, 2, 3, 4, 5)
newTuple = tuple(map(lambda x: x * 2, myTuple))
print(newTuple)
```

Solution

ACCEPTED

SCORE: 1.0 / 1

CANDIDATE'S ANSWER

(2, 4, 6, 8, 10)

CORRECT ANSWER

(1, 2, 3, 4, 5)

(2, 4, 6, 8, 10, 12)

TypeError: 'tuple' object does not support item assignment

Section 2 : Coding

Problem 1 : Increase Array

CODING

SCORE: 10

Problem Statement

You are given an array of size n containing non-negative integers. You have to make sure that the adjacent values are not equal. You can increase the height by one in a single operation. The cost of the operation is b_i . Then, they are considered good.

Calculate the minimum cost to make an array good.

Input Format

- The first line contains an integer n .
- The next line contains 2 , which denotes the number of integers in the next n lines..
- The Next n lines contain 2 integers, h_i and b_i , where h_i is the value in the array and b_i is the cost of operation on h_i .

Constraints

- $1 \leq n \leq 10^6$
- $1 \leq h_i, b_i \leq 10^7$

Output Format

- Return the minimum cost to make the array good.

Evaluation Parameters

- **Sample Input**

```
3
2
2 3
2 10
2 6
```

- **Sample Output**

```
9
```

- **Explanation**

The two ways to make this array good are as follows:

1. Increment the 2nd occurrence of 2 in the array to 3. This will have a cost of 10.
2. Increment the 1st and 3rd occurrence of 2 in the array to 3. This will have a total cost of $3+6=9$.

The second option option has a lower cost of 9, so that will be our output.

Solution

REJECTED

SCORE: 0.0 / 10

Python 3

```

1  #!/bin/python3
2
3  import math
4  import os
5  import random
6  import re
7  import sys
8
9  #
10 # Complete the 'goodArray' function below.
11 #
12 # The function is expected to return an LONG INTEGER.
13 # The function accepts INTEGER 2D ARRAY arr as parameter.
14 #
15
16 def goodArray(arr):
17     # Write your code here
18
19
20
21 if __name__ == '__main__':
22     arr_rows = int(input().strip())
23     arr_columns = int(input().strip())
24
25     arr = []
26
27     for _ in range(arr_rows):
28         arr_item = list(map(int, input().rstrip().split()))
29         arr.append(arr_item)
30
31     result = goodArray(arr)
32
33     print(str(result))

```

Evaluation Details

Testcase #4 (weight:3)

Status	Failed
Execution time	0.67s
CPU	0s
Memory	6MB
Description	Runtime error.

Solution output

```

File "/code/source.py3", line 21
if __name__ == '__main__':
^
IndentationError: expected an indented block

```

Testcase #3 (weight:2)

Status	Failed
--------	--------

Execution time	0.65s
CPU	0s
Memory	7MB
Description	Runtime error.

Solution output

```
File "/code/source.py3", line 21
if __name__ == '__main__':
^
IndentationError: expected an indented block
```

Testcase #6 (weight:4)

Status	Failed
Execution time	0.67s
CPU	0s
Memory	8MB
Description	Runtime error.

Solution output

```
File "/code/source.py3", line 21
if __name__ == '__main__':
^
IndentationError: expected an indented block
```

Testcase #5 (weight:3)

Status	Failed
Execution time	0.65s
CPU	0s
Memory	7MB
Description	Runtime error.

Solution output

```
File "/code/source.py3", line 21
if __name__ == '__main__':
^
IndentationError: expected an indented block
```

Testcase #1 (sample)

Status	Failed
---------------	--------

Execution time	0.68s
CPU	0s
Memory	7MB
Description	Runtime error.

Input

```
3
2
2 3
2 10
2 6
```

Expected output

```
9
```

Testcase #2 (weight:1)

Status	Failed
Execution time	0.65s
CPU	0s
Memory	8MB
Description	Runtime error.

Solution output

```
File "/code/source.py3", line 21
if __name__ == '__main__':
^
IndentationError: expected an indented block
```

Testcase #8 (weight:5)

Status	Failed
Execution time	0.69s
CPU	0s
Memory	8MB
Description	Runtime error.

Solution output

```
File "/code/source.py3", line 21
if __name__ == '__main__':
^
IndentationError: expected an indented block
```

Testcase #7 (weight:5)

Status	Failed
Execution time	0.65s
CPU	0s
Memory	7MB
Description	Runtime error.

Solution output

File "/code/source.py3", line 21

```
if __name__ == '__main__':
```

^

IndentationError: expected an indented block

Problem 2 : Django: Book Collection

PROJECT BASED BACKEND

SCORE: 10

Django API for User and Book Management

Implement a simple REST API to manage a collection of user and book records.

Each user and book record is a JSON entry with the following keys:

User

- id: The unique ID of the user. (Integer)
- username: The username of the user. (String)
- first_name: The first name of the user. (String)
- last_name: The last name of the user. (String)
- email: The email of the user. (String)

Book

- id: The unique ID of the book. (Integer)
- user: The ID of the user who created the book record. (Integer)
- isbn_number: The ISBN number of the book. (String)
- title: The title of the book. (String)
- author: The author of the book. (String)
- published_date: The published date of the book. (Date)
- pages: The number of pages in the book. (Integer)
- genre: The genre of the book. (String)
- language: The language of the book. (String)

API Endpoints

POST request to /users/

- Creates a new user record.
- Expects a JSON user object without an id property as a body payload.
- Adds the given user object to the collection of user records and assigns a unique integer id to it. The first created user record must have id 1, the second one 2, and so on.
- The response code is 201, and the response body is the created user record.

Example request body:

```
{
  "username": "john_doe",
  "email": "john@example.com",
  "first_name": "John",
  "last_name": "Doe"
}
```

Example response:

```
{
  "id": 1,
  "username": "john_doe",
  "email": "john@example.com",
  "first_name": "John",
  "last_name": "Doe"
}
```

GET request to /users/

- Returns a collection of all user records.
- The response code is 200, and the response body is an array of all user records.

Example response:

```
[
  {
    "id": 1,
    "username": "john_doe",
    "email": "john@example.com",
    "first_name": "John",
    "last_name": "Doe"
  },
  {
    "id": 2,
    "username": "jane_doe",
    "email": "jane@example.com",
    "first_name": "Jane",
    "last_name": "Doe"
  }
]
```

GET request to /users/<id>/

- Returns all the books for the user with the given id.
- The response code is 200, and the response body is the user record with the given id.

- If the user record with the given id does not exist, the response code is 404 and the response body is a json `{'message': 'User not found'}`.

```
[
  {
    "id": 2,
    "isbn_number": "129ISBN348",
    "title": "Treasure Island",
    "author": "Robert Louis Stevenson",
    "published_date": "2012-01-23",
    "pages": 135,
    "genre": "Adventure",
    "language": "English",
    "user": 1
  },
  {
    "id": 3,
    "isbn_number": "1234567890",
    "title": "Dune",
    "author": "Frank Herbert",
    "published_date": "2009-05-21",
    "pages": 456,
    "genre": "Fiction",
    "language": "English",
    "user": 1
  }
]
```

POST request to /books/

- Creates a new book record.
- Expects a JSON book object without an id property as a body payload.
- Adds the given book object to the collection of book records and assigns a unique integer id to it. The first created book record must have id 1, the second one 2, and so on.
- The response code is 201, and the response body is the created book record.

Example request body:

```
{
  "user": 1,
  "isbn_number": "1234567890",
  "title": "Dune",
  "author": "Frank Herbert",
  "published_date": "2009-05-24",
  "pages": 465,
  "genre": "Fiction",
  "language": "English"
}
```


Example response:

```
{
  "id": 1,
  "user": 1,
  "isbn_number": "1234567890",
  "title": "Dune",
  "author": "Frank Herbert",
  "published_date": "2009-05-24",
  "pages": 465,
  "genre": "Fiction",
  "language": "English"
}
```

GET request to /books/

- Returns a collection of all book records.
- The response code is 200, and the response body is an array of all book records ordered by their ids in increasing order.

Example response:

```
[
  {
    "id": 1,
    "user": 1,
    "isbn_number": "1234567890",
    "title": "Dune",
    "author": "Frank Herbert",
    "published_date": "2008-05-24",
    "pages": 465,
    "genre": "Fiction",
    "language": "English"
  },
  {
    "id": 2,
    "user": 1,
    "isbn_number": "0987654321",
    "title": "The Power of your subconscious mind",
    "author": "Joseph Murphy",
    "published_date": "2004-04-15",
    "pages": 200,
    "genre": "Self-help book",
    "language": "English"
  }
]
```

PATCH request to /books/<id>/

- Updates a book record with the given id.
- Expects a JSON book object without an id property as a body payload.
- On success, the response code is 200, and the response body is the updated book record.
- If there is no book with the given id in the collection, the response code is 404 with an error message `{ 'message': 'Book not found' }`.
- If no id is provided, the response code is 400 with an error message `No id provided`.

Example request body:

```
{
  "title": "Updated Book Title"
}
```

Example response for a valid book:

```
{
  "id": 1,
  "user": 1,
  "isbn_number": "1234567890",
  "title": "Updated Book Title",
  "author": "Author Name",
  "published_date": "2023-01-01",
  "pages": 100,
  "genre": "Fiction",
  "language": "English"
}
```

Example response for an invalid book:

```
{
  "message": "Book not found"
}
```

Example of no id provided:

```
{
  "message": "Book id is required"
}
```

DELETE request to /books/<id>/

- Deletes a book record with the given id.

- If the book exists, then delete the book with give id and the response code is 404.
- If there is no book with the given id, the response code is 404 with an error message Book not found.
- If no id is provided, the response code is 400 with an error message Book id is required.

Example response for an invalid book:

```
{
  "message": "Book not found"
}
```

Example of no id provided:

```
{
  "message": "Book id is required"
}
```

Solution

REJECTED

SCORE: 0.0 / 10

To view the candidate workspace, Please login with your credentials

Evaluation Details

[Download Scoring Logs](#)

[Download JUnit/XUnit XML 1](#)

✗ test_delete_without_id | Django TestCases ⊗ REJECTED Score: 0.0/4.69

✗ test_delete_with_invalid_id | Django TestCases ⊗ REJECTED Score: 0.0/4.69

✗ test_invalid_method | Django TestCases ⊗ REJECTED Score: 0.0/4.69

✗ test_get_all_books | Django TestCases ⊗ REJECTED Score: 0.0/4.69

✗ test_patch_without_id | Django TestCases ⊗ REJECTED Score: 0.0/4.69

✗ test_patch_with_invalid_id Django TestCases	⊗ REJECTED	Score: 0.0/4.69
✗ test_update_book Django TestCases	⊗ REJECTED	Score: 0.0/4.69
✗ test_create_user Django TestCases	⊗ REJECTED	Score: 0.0/4.69
✗ test_invalid_method Django TestCases	⊗ REJECTED	Score: 0.0/4.69
✗ test_get_user_with_id Django TestCases	⊗ REJECTED	Score: 0.0/4.69
✗ test_user_with_invalid_id Django TestCases	⊗ REJECTED	Score: 0.0/4.69
✗ test_invalid_user Django TestCases	⊗ REJECTED	Score: 0.0/4.69
✗ test_create_book Django TestCases	⊗ REJECTED	Score: 0.0/4.69
✗ test_user_withput_id Django TestCases	⊗ REJECTED	Score: 0.0/4.69
✗ test_delete Django TestCases	⊗ REJECTED	Score: 0.0/4.69
✗ test_create_invalid_data Django TestCases	⊗ REJECTED	Score: 0.0/4.69