Assignment 1: System Test Design and Postman

Course Name : **Software Testing and QA**
Course ID: CSE4495
Date: 25/12/2024

Submitted By

Group Name :Trinity

Group Member:

1. Tanvir Mahtab : 011183104
2. Md Shaif Shahriar Zidan : 011183099
3. Shuvro Sarker :

➔ /create

Choices

personnummer,name, courses_passed

Representative Values

- personnummer:
  - Valid: "990102-5678"
  - Invalid: "990101-1234" (duplicate)
  - Invalid: "990101-123" (malformed)
- name:
  - Valid: "Tanvir Mahtab"
  - Invalid: "" (empty)
  - Invalid: null (null value)
- courses_passed:
  - Valid: ["CSE1010", "CSE2020"]
  - Invalid: ["INVALID"] (invalid course ID)
  - Valid: [] (no courses passed)

Constraints

- ERROR:
  - If personnummer already exists, return {"error": "Personnummer already belongs to student <id>"}.

Test Cases

- Create Student with Valid Data
- Request Type: POST
  - URL: http://127.0.0.1:5000/create
  - Body:

```
{


    "name": "Tanvir Mahtab",


    "personnummer": "990102-1234",


    "courses_passed": ["CSE1010", "CSE2020"]


}
```

  - Assertions:
    - Status code is 201.
    - Response body contains the created student details.
    -

- Create Student with Duplicate Personnummer
  - Request Type: POST
  - URL: http://127.0.0.1:5000/create
  - Body:

```
{
```

```
    "name": "Tanvir Mahtab",
    "personnummer": "990102-1234",
    "courses_passed": ["CSE1010", "CSE2020"]
}
```

➢ Assertions:

- Status code is 400.
- Response body contains error message for duplicate personnummer.
-

❖ Create Student with Empty Name

➢ Request Type: POST

➢ URL: http://127.0.0.1:5000/create

➢ Body:

➢

➢

```
    "name": " ",
    "personnummer": "990102-1234",
    "courses_passed": ["CSE1010", "CSE2020"]
}
```

➢ Assertions:

- Status code is 400.
- Response body contains error message for empty name.

➔/update/{student_id}

Choices

- student_id
- name
- personnummer
- courses_passed

Representative Values

- student_id:
  - Valid: 2 (existing student ID)
  - Invalid : 0, -1, 999 (non-existing IDs)
- name:
  - Valid: "Tanvir Mahtab Updated"
  - Invalid: "" (empty)
  - Invalid: null (null value)
- personnummer:
  - Valid: "990101-1234" (must remain unchanged)
- courses_passed:
  - Valid: ["CSE1010", "DSE2020"]
  - Invalid: ["INVALID"] (invalid course ID)

Constraints

- IF: If student_id is valid, then the update should proceed.
- ERROR:If student_id < 1, return {"error": "Student ID <id> does not exist"}.

Test Cases:

- Update Student with Valid Data
  - Request Type: PUT
  - URL: http://127.0.0.1:5000/update/10
  - Body:

```
{
    "name": "Tanvir Mahtab Updated",
    "personnummer": "990101-1234",
    "courses_passed": ["CSE1010", "CSE2020", "CSE3030", "CSE4566"]
}
```

- Assertions:
  - Status code is 200.
  - Response body contains updated student details.
  -

- Update Student with Invalid ID
  - Request Type: PUT
  - URL: http://127.0.0.1:5000/update/999
  - Body:

```
    "name": "Tanvir Mahtab Updated",
    "personnummer": "990101-1234",
    "courses_passed": ["CSE1010", "CSE2020", "CSE3030", "CSE4566"]
}
```

  - Assertions:
    - Status code is 404.
    - Response body contains error message for non-existing student ID.
- Update Student with Empty Name
  - Request Type: PUT
  - URL: http://127.0.0.1:5000/update/10
  - Body:

```
{
    "name": "",
    "personnummer": "990101-1234",
    "courses_passed": ["CSE1010", "CSE2020", "CSE3030", "CSE4566"]
}
```

- Assertions:
  - Status code is 400.
  - Response body contains error message for empty name.

- Update Student with Changed Personnummer
  - Request Type: PUT
  - URL: http://127.0.0.1:5000/update/10
  - Body:

```
{
    "name": "Tanvir Mahtab Updated",
    "personnummer": "990101-123478",
    "courses_passed": ["CSE1010", "CSE2020", "CSE3030", "CSE4566"]
}
```

  - Assertions:
    - Status code is 400.
    - Response body contains error message for changing personnummer.

➔ /delete/{student_id}

- Choice: student_id

Representative Values

- student_id:
  - Valid: 5 (existing student ID)
  - Invalid: 0, -1, 999 (non-existing IDs)

Test Cases:

- Delete Student with Valid ID
  - Request Type: DELETE
  - URL: http://127.0.0.1:5000/delete/5
  - Assertions:
    - Status code is 200.
    - Response body confirms deletion.
- Delete Student with Invalid ID
  - Request Type: DELETE
  - URL: http://127.0.0.1:500 0/delete/1000
  - Assertions:
    - Status code is 404.
    - Response body contains error message for non-existing student ID.

/finished/{student_id}/{program_id}

Choices

- Choice: student_id
- Choice: program_id

Representative Values

- student_id:
  - Valid: 1 (existing student ID)
  - Invalid: 0, -1, 999 (non-existing IDs)
- program_id:
  - Valid: 1 (existing program ID)

Test Cases

- ❖ Check Graduation Status with Valid IDs
  - ➢ Request Type: GET
  - ➢ URL: http://127.0.0.1:5000/finished/1/1
  - ➢ Assertions:
    - ■ Status code is 200.
    - ■ Response body indicates whether the student is ready to graduate.
- ❖ Check Graduation Status with Invalid Student ID
  - ➢ Request Type: GET
  - ➢ URL: http://127.0.0.1:5000/finished/999/1
  - ➢ Assertions:
    - ■ Status code is 404.
    - ■ Response body contains error message for non-existing student ID.
- ❖ Check Graduation Status with Invalid Program ID
  - ➢ Request Type: GET
  - ➢ URL: http://127.0.0.1:5000/finished/1/999
  - ➢ Assertions:
    - ■ Status code is 404.
    - ■ Response body contains error message for non-existing program ID.

➔ /student/{student_id}

Choices

- Choice: student_id

Representative Values

- student_id:

- Valid: 2 (existing student ID)
- Invalid: 999 (non-existing ID)

Constraints

- IF: If student_id is valid, then the student details should be returned.
- ERROR:If student_id < 1, return {"error": "Student ID <id> does not exist"}.

Test Cases

- ❖ Get Student Details with Valid ID
  - ➢ Request Type: GET
  - ➢ URL: http://127.0.0.1:5000/student/2
  - ➢ Assertions:
    - ■ Status code is 200.
    - ■ Response body contains student details.
- ❖ Get Student Details with Invalid ID
  - ➢ Request Type: GET
  - ➢ URL: http://127.0.0.1:5000/student/999
  - ➢ Assertions:
    - ■ Status code is 404.
    - ■ Response body contains error message for non-existing student ID.

➔ /program/{program_id}

Choices

- Choice: program_id

Representative Values

- ❖ program_id:
  - ➢ Valid: 1 (existing program ID)
  - ➢ Invalid: 999 (non-existing ID)

Constraints

- ❖ IF: If program_id is valid, then the program details should be returned.

Test Cases

- ❖ Get Program Details with Valid ID
  - ➢ Request Type: GET
  - ➢ URL: http://127.0.0.1:5000/program/1
  - ➢ Assertions:
    - ■ Status code is 200.
    - ■ Response body contains required courses for the program.
- ❖ Get Program Details with Invalid ID
  - ➢ Request Type: GET
  - ➢ URL: http://127.0.0.1:5000/program/10000
  - ➢ Assertions:
    - ■ Status code is 404.
    - ■ Response body contains error message for non-existing program ID.

➔/finished/{student_id}/{program_id}

Choices

- Choice: student_id, program_id

Representative Values

- ❖ student_id:
  - ➢ Valid: 1 (existing student ID)
  - ➢ Invalid: 0, -1, 999 (non-existing IDs)
- ❖ program_id:
  - ➢ Valid: 1 (existing program ID)
  - ➢ Invalid: 0, -1, 999 (non-existing IDs)

Constraints

ERROR: If student_id < 1, return {"error": "Student ID <id> does not exist"}.

SINGLE: Testing with valid student_id and program_id

Test Cases

- ❖ Check Graduation Status with Valid IDs
  - ➢ Request Type: GET
  - ➢ URL: http://127.0.0.1:5000/finished/1/1
  - ➢ Assertions:
    - ■ Status code is 200.
    - ■ Response body indicates whether the student is ready to graduate (e.g., {"status": true, "completed_courses": 10}).
- ❖ Check Graduation Status with Invalid Student ID
  - ➢ Request Type: GET
  - ➢ URL: http://127.0.0.1:5000/finished/999/1
  - ➢ Assertions:

        - Status code is 404.
        - Response body contains error message for non-existing student ID (e.g., {"error": "Student ID 999 does not exist"}).
- Check Graduation Status with Invalid Program ID
    - Request Type: GET
    - URL: http://127.0.0.1:5000/finished/1/999
    - Assertions:
        - Status code is 404.
        - Response body contains error message for non-existing program ID (e.g., {"error": "Program ID 999 does not exist"}).
- Check Graduation Status with Both Invalid IDs
    - Request Type: GET
    - URL: http://127.0.0.1:5000/finished/999/999
    - Assertions:
        - Status code is 404.
        - Response body contains error message for non-existing student ID