

Testing the Student Management System

Tested By: Md. Tazveer Hossain Khan

Student Info

Test case ID	Request Type	URL	Body	Goal	Expected Output	Actual Output	Verdict
TC-01	GET	http://127.0.0.1:5000/student	N/A	Verify (200 OK) response when the request is sent, Ensure all the responses have filled up all the fields, and Ensure ID is always positive.	[{ "courses_passed": ["CSE2213", "CSE1110", "CSE1111", "CSE1114", "CSE1115", "CSE1116", "CSE2117", "CSE2118", "CSE3119", "CSE4010"], "id": 1, "name": "Abdul Jabbar", "personnummer": "920223-9999" }, { "courses_passed": ["CSE2213", "CSE1110", "CSE1111", "CSE1114", "CSE1115", "CSE1116", "CSE2117"], "id": 2, "name": "Abdel Ahmed", "personnummer": "990111-9999" }, { "courses_passed": [], "id": 3, "name": "Joel Stevensson ", "personnummer": "011030-9999" }, { "courses_passed": ["DSE1001",	[{ "courses_passed": ["CSE2213", "CSE1110", "CSE1111", "CSE1114", "CSE1115", "CSE1116", "CSE2117", "CSE2118", "CSE3119", "CSE4010"], "id": 1, "name": "Abdul Jabbar", "personnummer": "920223-9999" }, { "courses_passed": ["CSE2213", "CSE1110", "CSE1111", "CSE1114", "CSE1115", "CSE1116", "CSE2117"], "id": 2, "name": "Abdel Ahmed", "personnummer": "990111-9999" }, { "courses_passed": [], "id": 3, "name": "Joel Stevensson ", "personnummer": "011030-9999" }, { "courses_passed": ["DSE1001",	Pass

Test case ID	Request Type	URL	Body	Goal	Expected Output	Actual Output	Verdict
					<pre>"DSEE1002", "DSEE1003", "DSEE1004", "DSEE2105", "DSEE2116", "DSEE3217", "DSEE3218", "DSEE3239", "DSEE4100"], "id": 4, "name": "Yue Sakamoto", "personnummer": "980401-9999" }, { "courses_passed": ["DSEE1001", "DSEE1002", "DSEE1003", "DSEE3217"], "id": 5, "name": "Ivan Faknamovich", "personnummer": "951128-9999" }, { "courses_passed": ["CSE2213", "CSE1110", "CSE1111", "CSE1114", "CSE1115", "CSE1116", "CSE2117", "CSE2118", "CSE3119", "CSE4010", "DSEE1004", "DSEE2105"], "id": 6, "name": "Amy Pond", "personnummer": "020201-9999" }, { "courses_passed": ["CSE1115", "CSE4010", "DSEE1001", "DSEE1002", "DSEE1003", "DSEE1004", "DSEE2105", "DSEE2116", "DSEE3217",</pre>	<pre>"DSEE1002", "DSEE1003", "DSEE1004", "DSEE2105", "DSEE2116", "DSEE3217", "DSEE3218", "DSEE3239", "DSEE4100"], "id": 4, "name": "Yue Sakamoto", "personnummer": "980401-9999" }, { "courses_passed": ["DSEE1001", "DSEE1002", "DSEE1003", "DSEE3217"], "id": 5, "name": "Ivan Faknamovich", "personnummer": "951128-9999" }, { "courses_passed": ["CSE2213", "CSE1110", "CSE1111", "CSE1114", "CSE1115", "CSE1116", "CSE2117", "CSE2118", "CSE3119", "CSE4010", "DSEE1004", "DSEE2105"], "id": 6, "name": "Amy Pond", "personnummer": "020201-9999" }, { "courses_passed": ["CSE1115", "CSE4010", "DSEE1001", "DSEE1002", "DSEE1003", "DSEE1004", "DSEE2105", "DSEE2116", "DSEE3217",</pre>	

Test case ID	Request Type	URL	Body	Goal	Expected Output	Actual Output	Verdict
					<pre>"DSE3218", "DSE3239", "DSE4100"], "id": 7, "name": "Rory Williams", "personnummer": "010203-9999" }, { "courses_passed": ["CSE1115", "CSE4010", "DSE1001", "DSE1004", "DSE2105", "DSE1004"], "id": 8, "name": "Steve Rogers", "personnummer": "0704-9999" }, { "courses_passed": ["CSE2213"], "id": 9, "name": "Natasha Romanov", "personnummer": "861231-9999" } }</pre>	<pre>"DSE3218", "DSE3239", "DSE4100"], "id": 7, "name": "Rory Williams", "personnummer": "010203-9999" }, { "courses_passed": ["CSE1115", "CSE4010", "DSE1001", "DSE1004", "DSE2105", "DSE1004"], "id": 8, "name": "Steve Rogers", "personnummer": "0704-9999" }, { "courses_passed": ["CSE2213"], "id": 9, "name": "Natasha Romanov", "personnummer": "861231-9999" } }</pre>	
TC-02	GET	http://127.0.0.1:5000/student/6	N/A	Check if a valid student ID returns (200 OK)	<pre>{ "courses_passed": ["CSE2213", "CSE1110", "CSE1111", "CSE1114", "CSE1115", "CSE1116", "CSE2117", "CSE2118", "CSE3119", "CSE4010", "DSE1004", "DSE2105"], "id": 6, "name": "Amy Pond", "personnummer": "020201-9999" }</pre>	<pre>{ "courses_passed": ["CSE2213", "CSE1110", "CSE1111", "CSE1114", "CSE1115", "CSE1116", "CSE2117", "CSE2118", "CSE3119", "CSE4010", "DSE1004", "DSE2105"], "id": 6, "name": "Amy Pond", "personnummer": "020201-9999" }</pre>	Pass
TC-	GET	http://127.0.0.1:5000/student/-1	N/A	Check that the invalid ID returns 404 not	<pre>{ "error": "Student ID -1 does not exist" }</pre>	<pre>{ "error": "Student ID -1 does not exist" }</pre>	Pass

Test case ID	Request Type	URL	Body	Goal	Expected Output	Actual Output	Verdict
03				found and shows an error message.	}	}	
TC-04	GET	http://127.0.0.1:5000/student/5000000	N/A	Check that the non-existing ID returns 404 not found and shows an error message	{ "error": "Student ID 5000000 does not exist" }	{ "error": "Student ID 5000000 does not exist" }	Pass
TC-05	POST	http://127.0.0.1:5000/create	{ "name": "John Ceena", "personnummer": ".", "850101-1234", "courses_passed": ["CSE1200", "DSE2400", "CSE3100"] }	If all the valid input is provided, then it will add the student to the system.	{ "courses_passed": ["CSE1200", "DSE2400", "CSE3100"], "id": 14, "name": "John Ceena", "personnummer": "850101-1234" }	{ "courses_passed": ["CSE1200", "DSE2400", "CSE3100"], "id": 14, "name": "John Ceena", "personnummer": "850101-1234" }	Pass
TC-06	POST	http://127.0.0.1:5000/create	{ "name": "Md Tazveer Hossain Khan", "personnummer": "950123-1234", "courses_passed": ["CSE1010", "CSE2010"] }	If a duplicate personnummer is found, it will not add the student in the system.	{ "error": "Personnummer already belongs to student 13" }	{ "error": "Personnummer already belongs to student 13" }	Pass
TC-07	POST	http://127.0.0.1:5000/create	{ "name": "Aungona", "personnummer": "1234", "courses_passed": ["CSE1010", "DSE2020"] }	If a personnummer is provided in wrong format (expected format:((string, format YYMMDD-NNNN), then this student can not be created.	{ "error": "Malformed personnummer 1234" }	{ "error": "Malformed personnummer 1234" }	Pass
TC-08	PUT	http://127.0.0.1:5000/update/11	{ "name": "B Dhrubo", "personnummer": ".", "990101-5678", }	If personnummer is match then it will change the existing name and update the new name.	{ "courses_passed": ["CSE4000"], "id": "11", "name": "B Dhrubo", "personnummer": "990101-5678" }	{ "courses_passed": ["CSE4000"], "id": "11", "name": "B Dhrubo", "personnummer": "990101-5678" }	Pass

Test case ID	Request Type	URL	Body	Goal	Expected Output	Actual Output	Verdict
			<pre>"courses_passed": ["CSE4000"]</pre>				
TC-09	PUT -	http://127.0.0.1:5000/update/11	<pre>{ "name": "B Dhrubo", "personnummer": "970101-5678" }</pre>	As personnummer changing is not allowed, it will throw an error message.	<pre>{ "error": "Changes to personnummer are not allowed." }</pre>	<pre>{ "error": "Changes to personnummer are not allowed." }</pre>	Pass -
TC-10	DELETE -	http://127.0.0.1:5000/delete/3	N/A	Check if a valid id is provide who haven't pass any course, then system can remove that user.	<pre>{ "deleted": "3" }</pre>	<pre>{ "deleted": "3" }</pre>	Pass -
TC-11	DELETE -	http://127.0.0.1:5000/delete/999	N/A	Check if an invalid id is provided, then system cannot remove that user.	<pre>{ "error": "Student ID 999 does not exist" }</pre>	<pre>{ "error": "Student ID 999 does not exist" }</pre>	Pass -
TC-12	DELETE -	http://127.0.0.1:5000/delete/5	N/A	Check if it is trying to delete an existing valid ID who passed any course, deletion will be rejected.	<pre>{ "error": "Student 5 cannot be deleted because they have associated courses" }</pre>	<pre>{ "deleted": "5" }</pre>	Fail -
TC-13	GET -	http://127.0.0.1:5000/program	N/A	Get all the courses required for a program existing with their IDs.	<pre>[{ "courses_required": ["CSE2213", "CSE1110", "CSE1111", "CSE1114", "CSE1115", "CSE1116", "CSE2117", "CSE2118", "CSE3119", "CSE4010"], "id": 1 },]</pre>	<pre>[{ "courses_required": ["CSE2213", "CSE1110", "CSE1111", "CSE1114", "CSE1115", "CSE1116", "CSE2117", "CSE2118", "CSE3119", "CSE4010"], "id": 1 },]</pre>	Pass -

Test case ID	Request Type	URL	Body	Goal	Expected Output	Actual Output	Verdict
					<pre>"courses_required": ["DSE1001", "DSE1002", "DSE1003", "DSE1004", "DSE2105", "DSE2116", "DSE3217", "DSE3218", "DSE3239", "DSE4100"], "id": 2 }</pre>	<pre>"courses_required": ["DSE1001", "DSE1002", "DSE1003", "DSE1004", "DSE2105", "DSE2116", "DSE3217", "DSE3218", "DSE3239", "DSE4100"], "id": 2 }</pre>	
TC-14	GET -	http://127.0.0.1:5000/program/1	N/A	Get all the required courses assigned under specific ID(Department)	<pre>{ "courses_required": ["CSE2213", "CSE1110", "CSE1111", "CSE1114", "CSE1115", "CSE1116", "CSE2117", "CSE2118", "CSE3119", "CSE4010"], "id": 1 }</pre>	<pre>{ "courses_required": ["CSE2213", "CSE1110", "CSE1111", "CSE1114", "CSE1115", "CSE1116", "CSE2117", "CSE2118", "CSE3119", "CSE4010"], "id": 1 }</pre>	Pass -
TC-15	GET -	http://127.0.0.1:5000/program/999	N/A	Get error if try to get required course with an ID which is not exist	<pre>{ "error": "Program ID 999 does not exist" }</pre>	<pre>{ "error": "Program ID 999 does not exist" }</pre>	Pass -
TC-16	GET -	http://127.0.0.1:5000/program/abc	N/A	Get an error if try to get the required course with an Invalid ID.	<pre>{ "error": "Program ID abc is not formatted correctly." }</pre>	<pre>{ "error": "Program ID abc is not formatted correctly." }</pre>	Pass -
TC-17	GET -	http://127.0.0.1:5000/finished/1/1	N/A	Check if a valid student with a valid department ID is currently eligible for graduation.	<pre>{ "completed_courses": 10, "status": true }</pre>	<pre>{ "completed_courses": 10, "status": true }</pre>	Pass -
TC-18	GET -	http://127.0.0.1:5000/finished/1/999	N/A	Check if a valid student with a department ID that does not exist is currently eligible for graduation.	<pre>{ "error": "Program ID 999 does not exist" }</pre>	<pre>{ "error": "Program ID 999 does not exist" }</pre>	Pass -

Test Case Explanation:

TC-01:

S-1: Get Student List

S-2: Get the student details of those who have valid person numbers and have passed some courses.

S-3: Student data with a valid ID

TC-02:

Choice: Student_id

Representative Value: 3

- **Valid ID**
- Negative ID
- Not existing ID

TC-03:

Choice: Student_id

Representative Value: 3

- Valid ID
- **Negative ID**
- Not existing ID

TC-04:

Choice: Student_id

Representative Value: 3

1. Valid ID
2. Negative ID
3. **Not existing ID**

TC-05:

Choice: name, personnuber, courses_passed

Representative Value: 3

1. Name: 3
 - **Valid** ("Tazveer Hossain Khan")

- Empty (" ") → **ERROR**
 - Malformed (12abc)
2. personNumber: 4
- Valid Format
 - Invalid Format
 - Duplicate Number
 - Empty (" ") → **ERROR**
3. Courses_passed: 4
- Valid Course Code
 - Invalid Course Code
 - Wrong Department
 - Empty (" ") → **ERROR**

Total Test case: 3 ERROR + 2*2*3 = 15

TC-06:

Choice: name, personnuber, courses_passed

Representative Value: 3

1. Name: 3
- Valid ("Tazveer Hossain Khan")
 - Empty (" ") → **ERROR**
 - Malformed (12abc)
2. personNumber: 4
- Valid Format
 - Malformed format
 - **Duplicate Number**
 - Empty (" ") → **ERROR**
3. Courses_passed: 4
- Valid Course Code
 - Invalid Course Code
 - Wrong Department
 - Empty (" ") → **ERROR**

Total Test case: 3 ERROR + 2*2*3 = 15

TC-07:

Choice: name, personnuber, courses_passed

Representative Value: 3

1. Name: 3

- Valid ("Tazveer Hossain Khan")
 - Empty (" ") → **ERROR**
 - Malformed (12abc)
2. personNumber: 4
- Valid Format
 - **Malformed format**
 - Duplicate Number
 - Empty (" ") → **ERROR**
3. Courses_passed: 4
- Valid Course Code
 - Invalid Course Code
 - Wrong Department
 - Empty (" ") → **ERROR**

Total Test case: 3 ERROR + 2*2*3 = 15

TC-08:

Choice: name, personnuber, courses_passed

Representative Value: 3

1. Name: 3
- **Valid ("B Dhrubo")**
 - Empty (" ") → **ERROR**
 - Malformed ("12abc")
2. personNumber: 1
- personnumber → **ERROR** (Changing personnumber is not allowed)
3. Courses_passed: 4
- Valid Course Code
 - Invalid Course Code
 - Wrong Department
 - Empty (" ") → **ERROR**

Total Test case: 3 ERROR + 2*3 = 9

TC-09:

Choice: name, personnuber, courses_passed

Representative Value: 3

1. Name: 3
- Valid ("B Dhrubo")
 - Empty (" ") → **ERROR**

- Malformed ("12abc")
- 2. personNumber: 1
 - **personnumber** → **ERROR** (Changing personnumber is not allowed)
- 3. Courses_passed: 4
 - Valid Course Code
 - Invalid Course Code
 - Wrong Department
 - Empty (" ") → **ERROR**

Total Test case: 3 ERROR + 2*3 = 9

TC-10:

IF constraints used here.

Choice: Student_id

Representative Value: 3

1. Valid ID
2. Negative ID
3. Not existing ID → **ERROR**

Total Test case: 1 ERROR + 2 = 3

-----**Used for Constraint**-----

S1: Valid Student ID

S2: Students who have valid ID, but no courses passed.

S3: ID:3, passed_courses=[] .

TC-11:

Choice: Student_id

Representative Value: 3

1. Valid ID
2. Negative ID
3. **Not existing ID** → **ERROR**

Total Test case: 1 ERROR + 2 = 3

TC-12:

Choice: Valid ID, Associated courses

TC-13:

1. Get all the courses for each department

Total Test case: 1

TC-14:

Choice: Program_id

Representative Value: 4

1. **Valid ID**
2. Negative ID
3. Not existing ID
4. Malformed → **ERROR**

Total Test case: 1 ERROR + 3 = 4

TC-15:

Choice: Program_id

Representative Value: 4

1. Valid ID
2. Negative ID
3. **Not existing ID**
4. Malformed → **ERROR**

Total Test case: 1 ERROR + 3 = 4

TC-16:

Choice: Program_id

Representative Value: 4

1. Valid ID
2. Negative ID
3. Not existing ID
4. **Malformed** → **ERROR**

Total Test case: 1 ERROR + 3 = 4

TC-17:

Choice: student_id, status

Representative Value:

1. Student_ID: 3
 - **Valid ID**
 - Negative ID

- Not existing ID
2. Status: 2
 - True
 - False

Total Test case: $3 \times 2 = 6$

TC-18:

Choice: student_id, status

Representative Value:

1. Student_ID: 3
 - Valid ID
 - Negative ID
 - **Not existing ID**
2. Status: 2
 - True
 - False

Total Test case: $3 \times 2 = 6$