

API Testing Report

Created By - Team 24 Team Members -

- 1. Shivang Kumar
- 2. Ajeet Kumar
- 3. Bharti Bansal
- 4. Durgesh Puri Goswami
- 5. Ghosh Anwesha Santosh
- 6. Shlok Sheth
- 7. Maitreya Shelare
- 8. Yogesh Kumar

Description

This document describes the results of the API testing for the Seek 2.0 API. The API has endpoints for retrieving Students, Courses, Weeks, Assignments, Questions and GenAI helper.


Test Cases

1. Student

|


Test	API endpoint	Description	Success
test_get_students()	/api/student/	Tests successful retrival of all students	✓
test_get_student()	/api/student/1/	Tests successful retrival of a single student	✓
test_get_courses	/api/student/courses/1	Tests successful retrival of all courses for a student	✓
test_get_score	/api/student/score/1	Tests successful retrival of score for a student	✓
test_submit	/api/student/submit/	Tests successful submission of a student's answer	✓

1.1 test_get_students()

- **API endpoint:** /api/student/
- **Input:** None
- **Expected Output:** All students are returned
- **Actual Output:** All students are returned
- **Status:** 
- **Remarks:** The test passed successfully
- **Examples:**

```
# True Scenario
response = client.get("/api/student/")
print(response)
assert response.status_code == 200
assert response.get_json() == mock_students
```


1.2 test_get_student()

- **API endpoint:** /api/student/1/
- **Input:** student_id
- **Expected Output:** A single student is returned
- **Actual Output:** A single student is returned
- **Status:** 
- **Remarks:** The test passed successfully
- **Examples:**

```
# True Scenario
response = client.get(f"/api/student/{student_id}")
assert response.status_code == 200
assert response.get_json() == mock_student

# False Scenario
response = client.get(f"/api/student/non_existing_id")
assert response.status_code == 404
assert response.get_json() == {"error": "Student not found"}
```

1.3 test_get_courses()


- **API endpoint:** /api/student/courses/1
- **Input:** student_id
- **Expected Output:** All courses for a student are returned
- **Actual Output:** All courses for a student are returned
- **Status:** 
- **Remarks:** The test passed successfully

- Examples:

```
# True Scenario
response = client.get(f"/api/student/courses/{student_id}")
assert response.status_code == 200
assert response.get_json() == {
    "courses": [course.to_json() for course in mock_courses]
}

# False Scenario
response = client.get(f"/api/student/courses/{non_existing_student_id}")
assert response.status_code == 404
assert response.get_json() == {"error": "Student not found"}
```

1.4 test_get_score()


- API endpoint: /api/student/score/1
- Input: student_id
- Expected Output: Score for a student is returned
- Actual Output: Score for a student is returned
- Status: 
- Remarks: The test passed successfully
- Examples:

```
# True Scenario
response = client.get("/api/student/score", json={"student_id": student_id, "assignment_id": a
assert response.status_code == 200
assert response.get_json() == {"score": mock_score}

# False Scenario
# 1. with non-existing student id
response = client.get("/api/student/score", json={"student_id": "non_existing_id", "assignment
assert response.status_code == 404
assert response.get_json() == {"error": "Student not found"}
# 2. with non-existing assignment id
response = client.get("/api/student/score", json={"student_id": student_id, "assignment_id": "
assert response.status_code == 404
assert response.get_json() == {"error": "Assignment not found"}
```

1.5 test_submit()






- API endpoint: /api/student/submit/
- Input: student_id, question_id, answer
- Expected Output: Answer is submitted successfully

- **Actual Output:** Answer is submitted successfully
- **Status:** 
- **Remarks:** The test passed successfully
- **Examples:**


```
# True Scenario
response = client.post("/api/student/submit", json={"student_id": "invalid_student_id", "assignment_id": "invalid_assignment_id"})
assert response.status_code == 404
assert response.get_json() == {"error": "Student not found"}

# False Scenario
response = client.post("/api/student/submit", json={"student_id": student_id, "assignment_id": assignment_id})
assert response.status_code == 404
assert response.get_json() == {"error": "Assignment not found"}
```

2. Course


Test	API endpoint	Description	Success
test_get_courses()	/api/course/	Tests successful retrieval of all courses	
test_get_course()	/api/course/1/	Tests successful retrieval of a single course	
test_get_students	/api/course/students/1	Tests successful retrieval of all students for a course	
test_get_weeks	/api/course/weeks/1	Tests successful retrieval of all weeks for a course	
test_get_assignments	/api/course/assignments/1	Tests successful retrieval of all assignments for a course	

2.1 test_get_courses()

- **API endpoint:** /api/course/
- **Input:** None
- **Expected Output:** All courses are returned
- **Actual Output:** All courses are returned
- **Status:** 
- **Remarks:** The test passed successfully
- **Examples:**

```
# True Scenario
response = client.get("/api/course/")
assert response.status_code == 200
assert response.get_json() == mock_courses
```


2.2 test_get_course()

- **API endpoint:** /api/course/1/
- **Input:** course_id
- **Expected Output:** A single course is returned
- **Actual Output:** A single course is returned
- **Status:** 
- **Remarks:** The test passed successfully
- **Examples:**

```
# True Scenario
response = client.get("/api/course/1")
assert response.status_code == 200
assert response.get_json() == mock_course
```

```
# False Scenario
response = client.get("/api/course/2")
assert response.status_code == 404
assert response.get_json() == {"error": "Course not found"}
```

2.3 test_get_students()


- **API endpoint:** /api/course/students/1
- **Input:** course_id
- **Expected Output:** All students for a course are returned
- **Actual Output:** All students for a course are returned
- **Status:** 
- **Remarks:** The test passed successfully
- **Examples:**

```
# True Scenario
response = client.get("/api/course/students/1")
assert response.status_code == 200
assert response.get_json() == {"students": mock_students}
```

```
# False Scenario
response = client.get("/api/course/students/2")
```

```
assert response.status_code == 404
assert response.get_json() == {"error": "Course not found"}
```


2.4 test_get_weeks()

- **API endpoint:** /api/course/weeks/1
- **Input:** course_id
- **Expected Output:** All weeks for a course are returned
- **Actual Output:** All weeks for a course are returned
- **Status:** 
- **Remarks:** The test passed successfully
- **Examples:**

```
# True Scenario
response = client.get(f"/api/course/weeks/{course_id}")
assert response.status_code == 200
assert response.get_json() == {"weeks": week_data}

# False Scenario
response = client.get(f"/api/course/weeks/{non_existing_course_id}")
assert response.status_code == 404
assert response.get_json() == {"error": "Course not found"}
```

2.5 test_get_assignments()

- **API endpoint:** /api/course/assignments/1
- **Input:** course_id
- **Expected Output:** All assignments for a course are returned
- **Actual Output:** All assignments for a course are returned
- **Status:** 
- **Remarks:** The test passed successfully
- **Examples:**

```
# True Scenario
response = client.get(f"/api/course/assignments/{course_id}")
assert response.status_code == 200
assert response.get_json() == {"assignments": assignment_data}

# False Scenario
response = client.get(f"/api/course/assignments/{non_existing_course_id}")
assert response.status_code == 404
assert response.get_json() == {"error": "Course not found"}
```

3. Week

|

Test	API endpoint	Description	Success
test_get_week()	/api/week/1/	Tests successful retrieval of a single week	✓
test_get_assignments	/api/week/assignments/1	Tests successful retrieval of all assignments for a week	✓
test_get_lectures	/api/week/lectures/1	Tests successful retrieval of all lectures for a week	✓

3.1 test_get_week()

- **API endpoint:** /api/week/1/
- **Input:** week_id
- **Expected Output:** A single week is returned
- **Actual Output:** A single week is returned
- **Status:** ✓
- **Remarks:** The test passed successfully
- **Examples:**

```
# True Scenario
response = client.get("/api/week/1")
assert response.status_code == 200
assert response.get_json() == mock_week

# False Scenario
response = client.get(f"/api/week/{non_existing_week_id}")
assert response.status_code == 404
assert response.get_json() == {"error": "Week not found"}
```

3.2 test_get_assignments()

- **API endpoint:** /api/week/assignments/1
- **Input:** week_id
- **Expected Output:** All assignments for a week are returned
- **Actual Output:** All assignments for a week are returned
- **Status:** ✓
- **Remarks:** The test passed successfully
- **Examples:**

```

# True Scenario
response = client.get("/api/week/assignments/1")
assert response.status_code == 200
assert response.get_json() == {"assignments": mock_assignments}

# False Scenario
response = client.get(f"/api/week/assignments/{non_existing_week_id}")
assert response.status_code == 404
assert response.get_json() == {"error": "Week not found"}

```

4. Assignment

|

Test	API endpoint	Description	Success
test_get_assignment()	/api/assignment/1/	Tests successful retrival of a single assignment	✓

4.1 test_get_assignment()

- **API endpoint:** /api/assignment/1/
- **Input:** assignment_id
- **Expected Output:** A single assignment is returned
- **Actual Output:** A single assignment is returned
- **Status:** ✓
- **Remarks:** The test passed successfully
- **Examples:**

```

# True Scenario
response = client.get("/api/assignment/1")
assert response.status_code == 200
assert response.get_json() == mock_assignment
# False Scenario
response = client.get("/api/assignment/2")
assert response.status_code == 404
assert response.get_json() == {"error": "Assignment not found"}

```

5. Help

|

Test	API endpoint	Description	Success
test_chat()	/api/help/chat/	Tests successful chat with GenAI helper	✓

test_explain()	/api/help/explain/	Tests successful explanation of programming errors from GenAI helper	✓
test_feedback()	/api/help/feedback/	Tests successful feedback for incorrect questions to GenAI helper	

5.1 test_chat()

- **API endpoint:** /api/help/chat/
- **Input:** message - say ok
- **Expected Output:** Response with "ok" is returned
- **Actual Output:** Response with "ok" is returned
- **Status:** ✓
- **Examples:**

```
# True Scenario
data = {"message": message, "session_id": session_id}
response = client.post("/api/help/chat", json=data)
assert response.status_code == 200
response_json = response.get_json()
assert response_json["session_id"] == session_id
assert "ok" in response_json["response"].lower()
```


5.2 test_explain()

- **API endpoint:** /api/help/explain/
- **Input:** code - print("Hello, World!")
- **Expected Output:** Explanation for the code is returned
- **Actual Output:** Explanation for the code is returned
- **Status:** ✓
- **Examples:**

```
code = "print('Hello, World!')"
question = "What does this code print?"
data = {"code": code, "question": question}
response = client.post("/api/help/explain", json=data)
assert response.status_code == 200
response_json = response.get_json()
assert response_json["response"]
```

5.3 test_feedback()

- **API endpoint:** /api/help/feedback/

- **Input:** question_id, answers
- **Expected Output:** Feedback for incorrect answers is returned
- **Actual Output:** Feedback for incorrect answers is returned
- **Status:** 
- **Examples:**

```
assignment = [  
    {  
        "id": "1",  
        "question": "What is the capital of France?",  
        "answer": "Paris",  
        "correct": True,  
    },  
    {"id": "2", "question": "What is 2 + 2?", "answer": "5", "correct": False},  
]  
data = {"assignment": assignment}  
response = client.post("/api/help/feedback", json=data)  
assert response.status_code == 200  
response_json = response.get_json()  
assert response_json["1"] == "Correct"  
assert response_json["2"]
```

Conclusion

All the test cases passed successfully. The API is working as expected and is ready for deployment.