## **Endpoint Being Tested:** http://127.0.0.1:5000/topic_search

***Case:*** *Successful topic search query*

**Request Method:** POST

**Inputs:**

```
{
  "quest": "If-else statement"
}
```

**Expected Output:**

```
HTTP Status Code: 200 and JSON with 'results'
```

**Actual Output:**

```
HTTP Status Code: 200
JSON: {"search_query": "If-else statement", "results": [{"Lecture Link":
"https://www.youtube.com/watch?v=-
dBqiRCHbNw&list=PLZ2ps__7DhBb2cXAu5PevO_mzgS3Fj3Fs&index=22", "Lecture Title":
"Tutorial on if, else and else-if (elif) conditions", "Name Of Course":
"Programming in Python", "Week Number": "2"}, {"Lecture Link":
"https://www.youtube.com/watch?
v=FTX5wF_3J9Q&list=PLZ2ps__7DhBb2cXAu5PevO_mzgS3Fj3Fs&index=21", "Lecture Title":
"Introduction to the if statement", "Name Of Course": "Programming in Python",
"Week Number": "2"}, {"Lecture Link": "https://www.youtube.com/watch?
v=tDaXdoKfX0k&list=PLZ2ps__7DhBb2cXAu5PevO_mzgS3Fj3Fs&index=6", "Lecture Title":
"Variables and Literals", "Name Of Course": "Programming in Python", "Week
Number": "1"}]}
```

**Result:** Success

**Pytest Code:**

```
def test_topic_search_success(client):
    payload = {
        "quest": "If-else statement"
    }

    response = client.post("/topic_search", json=payload)
    data = response.get_json()
    expected_status = 200
    result = "Success" if response.status_code == expected_status and "results" in
data else "Failed"

    write_test_doc(
        title="***Case:*** *Successful topic search query*",
        endpoint="http://127.0.0.1:5000/topic_search",
        method="POST",
        inputs=json.dumps(payload, indent=2),
        expected="HTTP Status Code: 200 and JSON with 'results'",
        actual=f"HTTP Status Code: {response.status_code}\nJSON:
{json.dumps(data)}",
        result=result
    )

    assert response.status_code == 200
    assert "results" in data
    assert "search_query" in data
```

---

***Case:*** *Missing input field 'quest'*

**Request Method:** POST

**Inputs:**

```
{}
```

**Expected Output:**

```
HTTP Status Code: 400 and error message 'A query is required'
```

**Actual Output:**

```
HTTP Status Code: 400
JSON: {"Error": "A query is required"}
```

**Result:** Success

**Pytest Code:**

```
def test_topic_search_missing_input(client):
    payload = {}  # No 'quest' field

    response = client.post("/topic_search", json=payload)
    data = response.get_json()

    expected_status = 400
    result = "Success" if response.status_code == expected_status and "Error" in
data else "Failed"

    write_test_doc(
        title="***Case:*** *Missing input field 'quest'*",
        endpoint="http://127.0.0.1:5000/topic_search",
        method="POST",
        inputs=json.dumps(payload, indent=2),
        expected="HTTP Status Code: 400 and error message 'A query is required'",
        actual=f"HTTP Status Code: {response.status_code}\nJSON:
{json.dumps(data)}",
        result=result
    )

    assert response.status_code == 400
    assert data and "Error" in data
```

---

***Case:*** *Internal server error on topic search*

**Request Method:** `POST`

**Inputs:**

```
{
  "quest": null
}
```

**Expected Output:**

```
HTTP Status Code: 500 and error message
```

**Actual Output:**

```
HTTP Status Code: 500
JSON: {"Error": "Failed to conclude topic search"}
```

**Result:** `Success`

**Pytest Code:**

```
```

```
def test_topic_search_internal_server_error(client):
    payload = {
        "quest": None  # Will trigger type validation error
    }

    response = client.post("/topic_search", json=payload)
    try:
        data = response.get_json()
    except Exception:
        data = None

    expected_status = 500
    result = "Success" if response.status_code == expected_status else "Failed"

    write_test_doc(
        title="***Case:*** *Internal server error on topic search*",
        endpoint="http://127.0.0.1:5000/topic_search",
        method="POST",
        inputs=json.dumps(payload, indent=2),
        expected="HTTP Status Code: 500 and error message",
        actual=f"HTTP Status Code: {response.status_code}\nJSON:
{json.dumps(data)}",
        result=result
    )

    assert response.status_code == 500
    assert data and "Error" in data if isinstance(data, dict) else True
```