**Test Suites and Test Cases**

**UI Test Suite (Cypress)**

**Test Case: Search Functionality**

* **Description**: Validate that the search functionality returns correct results.
* **Steps**:
  1. Navigate to <https://www.givelify.com>.
  2. Enter "Alfred Street Baptist Church" in the search bar.
  3. Verify that the search results include "Alfred Street Baptist Church".
* **Expected Result**: Search results are accurate and displayed within 3 seconds.

**Test Case: Search Performance**

* **Description**: Verify the search performance under load.
* **Steps**:
  1. Simulate multiple users performing searches concurrently.
  2. Measure the time taken to return search results.
* **Expected Result**: Search results are displayed within 3 seconds under load.

**API Test Suite (Postman)**

**Test Case: Valid Search Request**

* **Description**: Ensure the API returns correct results for a valid search request.
* **Request**: GET http://www.omdbapi.com/?apikey=your\_api\_key&t=A%20Star%20Is%20Born&y=2018
* **Expected Response**: Status 200, correct movie details.

**Test Case: Invalid Search Request**

* **Description**: Validate the API's response to an invalid search request.
* **Request**: GET http://www.omdbapi.com/?apikey=your\_api\_key&t=NonExistentMovie
* **Expected Response**: Status 200, error message indicating movie not found.

**Test Case: Missing API Key**

* **Description**: Check API response when API key is missing.
* **Request**: GET http://www.omdbapi.com/?t=A%20Star%20Is%20Born&y=2018
* **Expected Response**: Status 401, error message indicating missing API key.

**Performance Test Suite (JMeter)**

**Test Case: Load Test**

* **Description**: Simulate high traffic to test search performance.
* **Steps**:
  1. Set up JMeter with 10,000 virtual users.
  2. Perform search requests concurrently.
  3. Measure response times and throughput.
* **Expected Result**: Search results are returned within 3 seconds.

**Test Case: Stress Test**

* **Description**: Evaluate system behavior under extreme load.
* **Steps**:
  1. Gradually increase the number of users beyond 10,000.
  2. Monitor system performance and response times.
* **Expected Result**: System maintains performance, and response times remain within acceptable limits.