Objective

Design acceptance tests to ensure that the HackerNews API functions as expected and returns correct data.

Testing

In order to effectively test the API, first we defined the DUT (device under test) and the TE (test entity or framework) which will be used to test against the DUT to verify its functionality.

We verify that

- New functionality works as expected, and
- Existing functionality continues to work as expected, that is, there are no regressions

The DUT (Device Under Test) is the 3 APIs:

- Retrieving top stories with the Top Stories API
- Using the Top Stories API to retrieve the current top story from the Items API
- Using the Top Stories API to retrieve a top story, retrieve its first comment using the Items API

The TE (Test Entity or Framework):

- We built an application using the 3 APIs to support the test cases to verify the functionality of the APIs.

Types of Tests

- Functional or Feature Tests
    - API fields are present/not present and have correct and well formatted values returned.
    - Correct combinations of the fields are returned.
    - Input validation.
- High Availability
    - The API endpoints are highly available for regular or bursty traffic.

    - Note: Extensive stress testing is covered separately.
- Baseline Performance
    - API performance meets the accepted baseline.
    - Note: Extensive performance testing is covered separately.

Test Pass/Fail Criteria

- Functional or Feature Tests
    - Actual Result equals Expected Result.
    - Results are deterministic, that is, same output for the same input.
    - Accurate error responses are returned for error scenarios.
- High Availability Tests
    - The API endpoints are up for regular and bursty traffic.
- Baseline Performance Tests
    - API meets baseline performance for various input combinations.

Test Setup

- hn_app.py was implemented as an application that serves as the Test Framework/Test Entity. It used the requests library to communicate with the API endpoints.
- test_hn_app.py was implemented with tests.
- Pytest was used for running the tests.
- Conda was used for package and environment management.

test_hn_app

- Test APIs to access the hackernews API.
- Test APIs handle ConnectionError exceptions by returning {‘error’: exception string}

test_hn_app.py tests

- API reachable
- Top Stories
- Items
- Comments
- Input validation with valid/invalid input
- Continuous/bursty traffic
- Performance baseline

load test

- load.js
    - Grafana k6 open-source load testing tool for performance baseline testing

Test Summary

- Tests were designed following the spec https://github.com/HackerNews/API?tab=readme-ov-file#new-top-and-best-stories
    - Mandatory fields, field types
- All implemented tests except input validation passed.

Observed behavior for invalid input data

Example Item ID	Response Status Code	Response Data
4125#6477	200	JSONDecodeError
41256!477	200	None
41256@477	400	{'error': 'Invalid path: Invalid token in path'}

Feature Test Results

```
(bindu_take_home) himabinduthota@Himabindus-MacBook-Pro hackernews-api-test % pytest -v -s test_hn_app.py
============================= test session starts ==============================
platform darwin -- Python 3.12.4, pytest-7.4.4, pluggy-1.0.0 -- /opt/anaconda3/envs/bindu_take_home/bin/python
cachedir: .pytest_cache
rootdir: /Users/himabinduthota/together_ai/my_repo/hackernews-api-test
configfile: pytest.ini
collected 19 items

test_hn_app.py::test_api_reachable PASSED
test_hn_app.py::test_get_top_stories_status_code_200 PASSED
test_hn_app.py::test_get_top_stories_count_500 PASSED
test_hn_app.py::test_get_current_top_story_status_code_200 PASSED
test_hn_app.py::test_get_current_top_story_comments_field PASSED
test_hn_app.py::test_required_field_id PASSED
test_hn_app.py::test_type_field_story PASSED
test_hn_app.py::test_story_by_field_present PASSED
test_hn_app.py::test_story_by_field_not_empty PASSED
test_hn_app.py::test_get_story_with_non_integer_story_id_status_code_400_1 XFAIL
test_hn_app.py::test_get_story_with_non_integer_story_id_1 XFAIL
test_hn_app.py::test_get_story_with_non_integer_item_id_status_code_400_2 XFAIL
test_hn_app.py::test_get_story_with_non_integer_item_id_2 XFAIL
test_hn_app.py::test_comment_should_have_a_parent PASSED
test_hn_app.py::test_comment_should_have_a_text PASSED
test_hn_app.py::test_comment_should_have_type_comment PASSED
test_hn_app.py::test_comment_by_field_present PASSED
test_hn_app.py::test_comment_by_field_not_empty PASSED
test_hn_app.py::test_get_all_top_stories_and_iterate

 *** NOTE: This test takes about 2 mins to complete. To skip next time use -m "not burst" flag
PASSED

======================== 15 passed, 4 xfailed in 85.27s (0:01:25) ==============
(bindu_take_home) himabinduthota@Himabindus-MacBook-Pro hackernews-api-test %
```

Load Test Results

```
(bindu_take_home) himabinduthota@Himabindus-MacBook-Pro hackernews-api-test % k6 run load.j
s

          /\      |‾‾| /‾‾/   /‾‾/
     /\  /  \     |  |/  /   /  /
    /  \/    \    |     (   /   ‾‾\
   /          \   |  |\  \ |  (‾)  |
  / __________ \  |__| \__\ \_____/ .io

  execution: local
     script: load.js
     output: -

  scenarios: (100.00%) 1 scenario, 10 max VUs, 1m0s max duration (incl. graceful stop):
           * default: 10 looping VUs for 30s (gracefulStop: 30s)


running (0m30.6s), 00/10 VUs, 260 complete and 0 interrupted iterations
default ✓ [======================================] 10 VUs  30s

     data_received..................: 1.4 MB 47 kB/s
     data_sent......................: 41 kB  1.3 kB/s
     http_req_blocked...............: avg=5.53ms   min=1µs     med=6µs     max=153.53ms p(90)=8µs      p(95)=27µs
     http_req_connecting............: avg=891.54µs min=0s      med=0s      max=23.32ms  p(90)=0s       p(95)=0s
     http_req_duration..............: avg=170ms    min=53.41ms med=62.59ms max=2.38s    p(90)=178.83ms p(95)=247.39ms
       { expected_response:true }...: avg=170ms    min=53.41ms med=62.59ms max=2.38s    p(90)=178.83ms p(95)=247.39ms
     http_req_failed................: 0.00%  ✓ 0       ✗ 260
     http_req_receiving.............: avg=75.68µs  min=30µs    med=74.5µs  max=197µs    p(90)=106.09µs p(95)=115.04µs
     http_req_sending...............: avg=23.83µs  min=7µs     med=22µs    max=300µs    p(90)=32µs     p(95)=37µs
     http_req_tls_handshaking.......: avg=3.57ms   min=0s      med=0s      max=102.92ms p(90)=0s       p(95)=0s
     http_req_waiting...............: avg=169.9ms  min=53.35ms med=62.49ms max=2.38s    p(90)=178.75ms p(95)=247.21ms
     http_reqs......................: 260    8.49254/s
     iteration_duration.............: avg=1.17s    min=1.05s   med=1.06s   max=3.38s    p(90)=1.22s    p(95)=1.24s
     iterations.....................: 260    8.49254/s
     vus............................: 10     min=10    max=10
     vus_max........................: 10     min=10    max=10

(bindu_take_home) himabinduthota@Himabindus-MacBook-Pro hackernews-api-test %
(bindu_take_home) himabinduthota@Himabindus-MacBook-Pro hackernews-api-test %
```