Evaluation Document

- We have evaluated the application by measuring the response times of the server by changing the path length and giving both min and max as the elevation gains. This is achieved by keeping the source location fixed and changing the destination location farther and farther from the source.

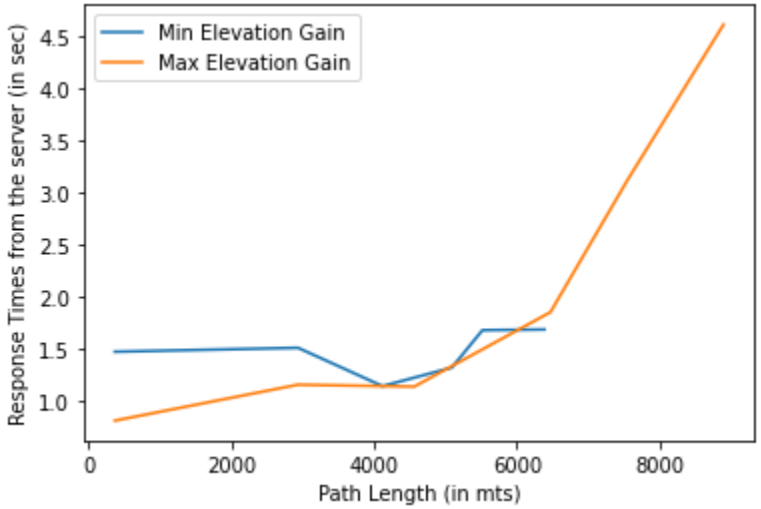


Fig.1 Response Times from the server vs Path Length

From Fig.1, it can be observed that the response times from the server are pretty low. For lower path lengths, the response time for min and max elevation gain is almost same and as path length increases, the response time for max increases as max elevation gain explores the paths keeping in mind the neighboring nodes with maximum elevation gain

- While doing this, we've also captured and analyzed how min and max elevation gain changes with path length.

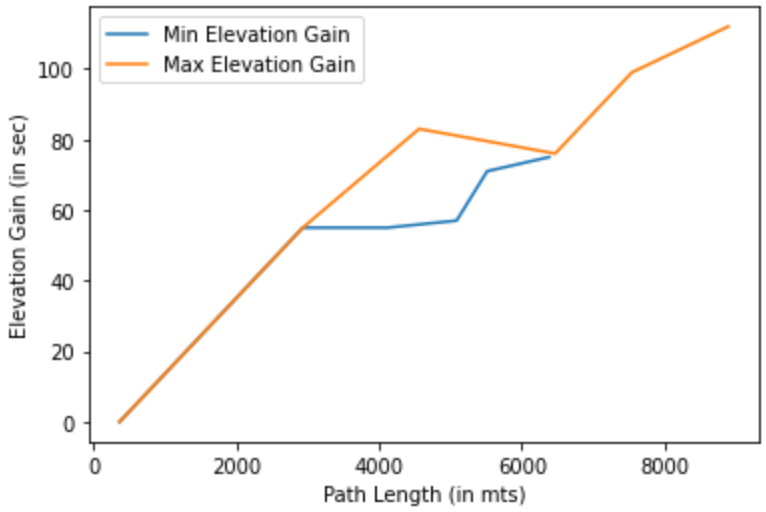


Fig.2 Elevation Gains vs Path Length

From Fig. 2, it can be observed that the elevation gains obtained from the application are almost the same when the path length is small as the possible number of paths between 2 closer points is less. As the path length increases, the elevation gain for max option is higher than min one as the number of paths increases and the option to choose min and max arises.

Testing:

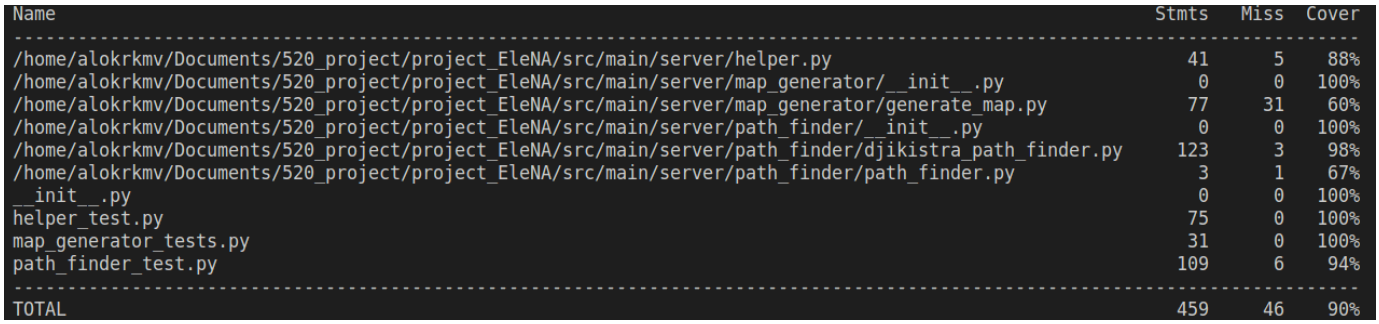
We have added an exhaustive test suite to make sure that our application performs as expected and all corner cases are covered properly. The following are covered by our application

- Unit testing :
  - Test edge cases
  - Validation checks on different inputs.
  - Tested PathFinder, MapProcessor and helper module individually.
- Manual Testing:
  - We tested our server using Postman using different locations and scenarios.
  - We tested edge cases and error situations manually as well.
- Integration test:
  - End-to-end testing for all components together
  - Testing frontend and backend together with different scenarios and edge cases.
  - We are using the integration test suite runner of Postman to run our integration tests.

- UI testing:
    - For UI testing, we made 2 frontends, and asked our peers to try our application and give feedback on the UI elements. We also asked them about our design choices and if they liked the way our UI looked. A friend of ours suggested the slider percentage changer which we changed after the feedback, it was a number field before.

Coverage:

The test coverage was calculated to be 90%.



Name	Stmts	Miss	Cover
/home/alokrkmv/Documents/520_project/project_EleNA/src/main/server/helper.py	41	5	88%
/home/alokrkmv/Documents/520_project/project_EleNA/src/main/server/map_generator/__init__.py	0	0	100%
/home/alokrkmv/Documents/520_project/project_EleNA/src/main/server/map_generator/generate_map.py	77	31	60%
/home/alokrkmv/Documents/520_project/project_EleNA/src/main/server/path_finder/__init__.py	0	0	100%
/home/alokrkmv/Documents/520_project/project_EleNA/src/main/server/path_finder/djikistra_path_finder.py	123	3	98%
/home/alokrkmv/Documents/520_project/project_EleNA/src/main/server/path_finder/path_finder.py	3	1	67%
__init__.py	0	0	100%
helper_test.py	75	0	100%
map_generator_tests.py	31	0	100%
path_finder_test.py	109	6	94%
TOTAL	459	46	90%