1. Manual code review
    a. Code is well documented, good variable names, efficient, no bad design smells, correct error handling:
    b. 10/10
    c. Their code meets all of the requirements in this section and everything is well documented in comments and in the design document
2. Manual test review
    a. Tests are complete (all APIs exposed to the frontend are tested, three main use cases are tested), errors and edge cases are thoroughly tested, correct assertions are used:
        i. 10/10
        ii. All endpoints are tested thoroughly and 3 main use cases are tested as well. The backend tests contain tests that account for errors and edge cases in the mocked tests.
    b. Tests implementation matches the requirements and design:
        i. 9/10
        ii. All tests correctly test out the happy paths and error cases thoroughly so it matches the requirements and design accurately.
    c. Tests are well-structured:
        i. 10/10
        ii. Tests are organized well, with each described group representing an endpoint so it is easy to view different test suites and individual tests.
    d. Tests are comprehensive (good coverage):
        i. 8/10
        ii. Code coverage is adequate, and any lines that weren't covered were justified completely in the writeup. Most main cases and main error/fail cases were covered as well.
    e. Non-functional requirements are tested well:
        i. 10/10
        ii. Non functional requirement tests are defined very clearly with good sources for test reasoning.
    f. All backend tests can be run automatically:
        i. 10/10
        ii. Jest tests are able to be run smoothly with an npm script provided
3. Automated code review
    a. Codacy runs with the required setup:
        i. 10/10
        ii. Codacy runs perfectly and is setup to scan the codebase after every commit
    b. All remaining Codacy issues are well-justified:
        i. 10/10
        ii. There were no Codacy issues that needed to be justified as all of them were fixed

4. Fault: report one major implementation issue you found in your peer-team app. The report should contain the details about the issue (with screenshots) and the severity of the issue

- Bug Report: Custom Marker Disappearing on Tab Switch

  Commit Hash: 25c05864ada8da6fbd19ee0eab303f878ab19540

  Steps to Reproduce:
  - Open the app and navigate to the Maps tab.
  - Create a custom marker on the map.
  - Verify that the marker is successfully created and visible.
  - Switch to the Home tab.
  - Return to the Maps tab.

  Expected Behavior:
  - The custom marker should persist and remain visible after switching tabs.

  Actual Behavior:
  - The custom marker disappears when returning to the Maps tab.