AutoGen QA Assignment

Key Problems

1. Frontend state change delays
2. Slow loading speeds
3. Unstable deployments/broken links

To test for the three issues reported and outlined above, we need to create both performance and functionality test cases. Testing should prioritize reliability and actionable feedback. Testing tools including Chrome DevTools, Lighthouse, and Playwright can be utilized to ensure tests are recorded, logged, and comparable between tests and testers.

Testing Tech Stack

Chrome DevTools
Reasoning:
- Built directly into Chrome, little setup required for testing to begin.
- Simple to use, does not require major training or onboarding for testers.

Playwright Framework
Reasoning:
- Language independent, can be used to test multiple projects that use varying tech stacks.
- Cross-platform support - Windows, Linux, macOS, etc
- Supports all major browsers and allows for mobile testing.
- Offers test parallelisation as a basic feature

Lighthouse
Reasoning:
- Open Source tool, can be run with Chrome DevTools for simplicity
- Provides a range of insights on web page and web app performance which can be analyzed to identify areas for improvements.
- Widely utilized, and various options available for sharing and viewing reports

Lighthouse CI
Reasoning:
- Can run automated tests on a defined schedule
- Great for preventing site regressions
- Integrates simply with GitHub to monitor changes and perform audits following each commit.

Test Cases:

Test Case ID: #AGN-STATE-001
Test Scenario: To test frontend state change times on AutoGen. Validate that UI state changes update within the acceptable performance threshold of <500ms across rendered components.
Test Steps:

- User navigates to https://autogen.nodeops.network/ - site will redirect to projects screen if user is already logged in. If not logged in, provide credentials and log in.
- User presses F12 button to access developer tools.
- User clicks Performance tab at the top of the tools screen.
- User presses the record button.
- User provides inputs to buttons and other elements of the app that trigger state changes based on inputs.
- User verifies state transitions after user actions (clicks, form submissions, navigation).
- Once the desired actions have been completed, user clicks "Stop" on the performance recording in DevTools window.
- User saves logs and exports traces as JSON file for later analysis.
- User repeats above steps 3-6 times and records variability of state change updates between tests.
- User records pass/fail result of test.
- Analysis of results to follow testing, with a focus on identifying any delayed state changes and the cause of the delay.

Prerequisites: AutoGen Account Credentials, Chrome browser installed, Playwright testing framework installed and tracing enabled.
Browser: Chrome v 141, Device: PC with Windows 11 24H2 OS
Test Data: state change load times
Expected/Intended Results: Full update to user input in <500 ms
Actual Results: TBD on test
Test Status – Pass/Fail: TBD on test

Alternative testing paths:

- Capture metrics using Playwright in addition to Chrome DevTools.
- Control additional variables, inlcuding clearing and disabling cache, setting consistent network profile in Chrome DevTools, and disabled extensions.
- Test using different browsers to compare results and determine if browser compatibility issues arise.
- Test using different operating systems including macOS, Linux, etc.

Duration:
This test case may have a range of duration expectations depending on the complexity of the deployment. Expected duration of this test case could range from 3-5 minutes if using Chrome DevTools, and extended to 5-8 minutes if also implementing testing via Playwright. Tests should be run following user reports on issues, and before each update.

Test Case ID: #AGN-LOAD-001
Test Scenario: To test loading speeds on AutoGen
Test Steps:

- User navigates to https://autogen.nodeops.network/ - site will redirect to projects screen if user is already logged in. If not logged in, provide credentials and log in.
- User presses F12 button to access developer tools.
- User clicks Performance tab at the top of the tools screen.
- User logs LCP value and clicks Record and Reload option.
- User compares and logs recorded LCP value and insights, save trace as JSON file for later analysis.
- User repeats above steps 3-6 times and calculates average LCP values.
- User records pass/fail result of test.
- Analysis of processes that are taking longer than expected, or to identify processes that can be simplified or made more lightweight to follow testing and create improvement plan. Examples could include long-running JS tasks, render-blocking scripts, large image payloads, and layout or paint bottlenecks.

Prerequisites: AutoGen Account Credentials, Chrome browser installed
Browser: Chrome v 141, Device: PC with Windows 11 24H2 OS
Test Data: Page and asset load times via LCP value.
Expected/Intended Results: LCP value of <2.5s
Actual Results: TBD on test
Test Status – Pass/Fail: TBD on test

Alternative testing paths:

- Use Lighthouse for additional insights, data collection, and improvement recommendations on load times.
- Compare additional data points beyond LCP, including TTFB, CLS, and Total Load Times. Compare to Google's Core Web Vitals performance guidelines.
- Control additional variables, inlcuding clearing and disabling cache, setting consistent network profile in Chrome DevTools, and disabled extensions.
- Test using different browsers to compare results and determine if browser compatibility issues arise.
- Test using different operating systems including macOS, Linux, etc.

Duration:
This test case is relatively lightweight, though should be done multiple times to confirm persistent issues. Duration of this test case could range from 1-5 minutes, depending on issues that arise. Testing should also be done following user issue reports to confirm issues and address any load time concerns.

Test Case ID: #AGN-STABILITY-001
Test Scenario: To test stability and link integrity
Test Steps:

- User navigates to https://autogen.nodeops.network/ - site will redirect to projects screen if user is already logged in. If not logged in, provide credentials and log in.
- User confirms page load and correct build version for testing unstable deployment.
- User presses F12 button to access developer tools.
- User checks DevTools console in Chrome for error logs - record any error logs.
    - Record any JS errors, failed asset loads, or other errors.
- After experiencing instability, user tests to see if refreshing page addresses any instability issues that have arisen in their session.
- If instability persists, user to try backoff retry pattern to confirm persistent error or if issue resolves with additional attempts.
- User tests all page links to ensure there are no 404 or 500 errors, or blank screens while following links.
- User tests edge cases of opening links in new tabs and new windows to confirm consistent behaviour.
- User records pass/fail result of test.
- User reviews notes to ensure all issues and behaviours are recorded for review.

Prerequisites: AutoGen Account Credentials, Chrome browser installed
Browser: Chrome v 141, Device: PC with Windows 11 24H2 OS
Test Data: Stability of deployment and link integrity.
Expected/Intended Results: Stable usage throughout test, and all links functioning correctly.
Actual Results: TBD on test
Test Status – Pass/Fail: TBD on test

Alternative testing paths:

- Use Lighthouse CI for automated audits for performance regression and build in alerts to be shared in internal team communications channels. Configure automated audits to occur on a schedule that makes most sense in alignment with the developer update pipeline.

Duration:
This is a more involved test procedure, but should be done manually before each major update or deployment. Duration of this test case could range from 3-10 minutes, depending on issues that arise. Testing should also be done following user issue reports to confirm bugs or issues, and to determine the path to correction. The need for manual execution can be minimized (but not eliminated) by implementing Lighthouse CI for automated auditing.

Recommend AutoGen UX improvements and Feature Updates:

The available interactions on the project templates page in the Marketplace section are not immediately obvious. Clicking the text opens a lightbox with more information on the template, but this is also covered as a tooltip through hovering.

I would recommend removing the "Deploy Template" button on the summary card, and only having it on the lightbox section for a couple of reasons. First, it makes it more obvious that users can click into each template for more information. Second, it prompts users to expand into the lightbox view, which can be better utilized to provide more detailed information about the template. Third, it simplifies the options available to the user on the marketplace landing page, and lets them explore the details of each template before feeling rushed to deploy one of the templates.

An interesting feature update might be the ability for users to add their own templates into the Marketplace. If this feature does exist already, I did not come across it. The Marketplace section is specified to be curated, but an option to submit a template for review would be great. It might also help to have some way to earn $NODE from contributing templates to the marketplace, which could be quantified by the number of unique users that deployed the template, or something similar.

On the main landing page for AutoGen, the testimonials section is great, but could be improved. It would be awesome if on hover, the testimonials stopped the side-scroll to make it easier for people to read. I would also suggest instead of taking users directly to the post, you pop up a lightbox with the post embedded with a button that more clearly takes the user to the social post. Right now if users want to reliably read any of the testimonials, they need to click, which takes them off site. It's great to have the confirmation that these are real people who have said good things about AutoGen, but it isn't immediately clear from the landing page that a click will take them off site to a social media post. Better to embed in a readable form, keep the users on the AutoGen page, and provide them a button there that clearly moves them to the post.

Returning users also don't have a clear way to login, as the only button on the landing page is "Sign Up". The login is available through that page, but could be clearer for returning users.

Sign Up and Login with Google/Github is a great feature, though it would be nice to add a blockchain native option like Sign In With Ethereum (https://login.xyz).