

Testing Document
for
JAX Front-end Project

Version 1.1 approved

Authored by Kate Stenberg

Edited by Sarah Crane

CS5500 – Foundations for Software Engineering, Professor Gary Cantrell

Northeastern University, Roux Institute

April 4, 2025

Table of Contents

Table of Contents.....	2
Revision History.....	3
Team Signatures.....	3
Project Description.....	4
Testing Plan.....	4
Static Testing.....	4
Unit Tests.....	4
Integration Tests.....	5
Validation Tests.....	5
System Tests.....	5
References.....	6

Revision History

Name	Date	Reason For Changes	Version
Kate Stenberg	4/3/2025	Initial Draft	1.0
Sarah Crane	4/3/2025	Edits	1.1

Team Signatures

Name	Date
Sarah Crane	4/3/2025
Kate Stenberg	4/3/2025
Harshil Patel	4/3/2025
Prince Safo	4/4/2025

Project Description

The Jax Front-End project is an overhaul of the current GeneWeaver website interface. The GeneWeaver application program is currently usable directly through a console if a user downloads the code files, but for those unwilling or unable to operate that way, it also exists as a web application, which provides a convenient user interface for the tools and functionalities. However, the website is not without issues. Firstly, it does not provide functionality for all the analysis tools available in the code files. Secondly, it does not provide accessibility features such as ToolTip or text-to-speech options. Thirdly, the relations between pages are bulky in some parts and could be streamlined for a more efficient site layout. Finally, the design itself is somewhat dated and could be updated aesthetically.

Testing Plan

Testing will be done by all group members through assignment of various roles. Some testing will be handled by one or two people exclusively (validation testing, for example) while others will be carried out by all group members (system tests). Testing will be done at the end of the coding phase of the project. Bias will be avoided by using impartial checkers (sites that are designed to technically check things like accessibility and code consistency) and by using live test subjects chosen to mimic the user base of the app.

Static Testing

Static testing will be done both by the group (having other group members review code for clarity) and by using online code checkers for consistency and convention, as we are not experts in React and JavaScript. Between these methods, the code should be clear and consistent and pass conventions.

Static Testing Taskst:

- Each code file should be reviewed by 2 other group members for clarity.
- Code files should be run through an AI engine such as ChatGPT or Claude Enterprise for consistency and convention. [1, 2]
- Code files should be run through a code checker, such as Cloud Defense (<https://www.clouddefense.ai/tools/code-checker/javascript>). [3]

Unit Tests

Unit tests will be done as we progress. As this project comprises only an interface and does not involve any integration with back-end operations or algorithms (analysis tools or databases), unit testing will involve testing each individual page, making notes of those tests in a spreadsheet for shared testing documentation. This will include ensuring that the visual layouts load as desired across multiple devices.

Unit Testing Tasks:

- Test layout of each page on a laptop computer.
- Test layout of each page on a tablet.
- Test layout of each page on a phone.

Integration Tests

Integration tests for this project will be top down, and will chiefly involve ensuring navigation between pages is functional and efficient. This means testing the links on each page to make sure they are fully operational and lead to the correct destination. In our case, no other testing is needed, as this is a surface-level interface and does not interact with anything else.

Integration Testing Tasks:

- Test all links on each page to ensure functionality and document results in the testing spreadsheet.

Validation Tests

During validation testing, we will make sure that the project aligns with the project requirements we set out for ourselves and that it meets the needs of the client. We will also be making sure that the website meets accessibility standards; to do this we will use website tools that evaluate web pages for accessibility features. Using pagespeed.web.dev, we will aim for a score of “Good.” [4]

Validation Testing Tasks

- Meet with the client to review project achievements.
- Review SRS to ensure all requirements are met; log checks into the testing spreadsheet.
- Run pages through <https://pagespeed.web.dev/> and achieve a score of “Good.” [4]

System Tests

Our system tests will be using live testers. Since our system does not allow for any data input, we will primarily be soliciting tests from other students at the Roux. It will not be a random sample; we will be targeting people who reflect the target audience of GeneWeaver: students, educators, and scientists who are at least somewhat familiar with genomics. We will try for five testers (one through each group member). Each tester will be asked to sit down and complete a given set of tasks on the website without guidance from the tester. They will then complete a survey evaluating the ease of the tasks, the accessibility of the website, and their enjoyment in using it.

The testers will be asked to complete the following tasks:

- (1) Navigate to the search function to mimic search for a gene set.
- (2) Navigate to the analysis tools page.
- (3) Navigate to the documentation section.
- (4) Navigate to the user’s gene sets page.
- (5) Navigate to the user’s projects page.
- (6) Navigate to submit an issue with the website.

(7) Identify where the funding from this project comes from.

The testing survey will include the following questions:

- (1) On a scale of 1 to 5, with 1 being the most difficult and 5 being the easiest:
 - (a) How easy was task 1?
 - (b) How easy was task 2?
 - (c) How easy was task 3?
 - (d) How easy was task 4?
 - (e) How easy was task 5?
 - (f) How easy was task 6?
 - (g) How easy was task 7?
- (2) What is your level of education and field?
- (3) On a scale of 1 to 5, with 1 being the least enjoyable and 5 being the most enjoyable, how would you rate this site based on the following characteristics?
 - (a) Readability
 - (b) Clarity
 - (c) Navigability
 - (d) Aesthetic
 - (e) Accessibility

Testers will also note the time it takes for the subjects to complete each task. The target time will be under 60 seconds to complete each task. Also included in the test data will be tester name, environment, date and time, and other comments from the tester or subject. If possible, recordings should also be taken of the interviews, with the subject's permission.

System Testing Tasks

- Administer tests to test subjects and log results in the shared testing spreadsheet.

References

- [1] "ChatGPT," OpenAI, <https://chatgpt.com/> (accessed Apr. 3, 2025).
- [2] "Claude," Anthropic, <https://claude.ai/> (accessed Apr. 3, 2025).
- [3] "Best JavaScript Code Checker Tool Online," CloudDefense.AI, <https://www.clouddefense.ai/tools/code-checker/javascript> (accessed Apr. 3, 2025).
- [4] "About PageSpeed Insights," PageSpeed Insights - Google, <https://developers.google.com/speed/docs/insights/v5/about> (accessed Apr. 3, 2025).