We will use a combination of two main testing methods.

The first will be manual testing, done by each member as they're programming and testing new features. This will help ensure both the functionality and frontend visuals are working as expected.

Since manual testing tends to miss certain edge cases (increasing with program complexity), we will also be implementing a second testing method, called 'Test-driven development,' or TDD. TDD's main focus is creating automated tests for a feature *before* programming the functionality. This helps with two core aspects of good development – ensuring added code meets requirements as it's being coded, and allowing automated testing, and therefore automated regression testing.

Essentially, TDD is used by creating and running a test function, which will call functions of different features with a certain input, and then compare the result with the expected output. It doesn’t need to know what process is being done on the inputs, only that the output matches what is expected. This is planned to be used for every non-void function.

To ensure continuous integration, we will use GitHub. Commits will be done by each dev at least once per development day. Features will be broken down into atomic tasks, allowing each part of the feature to be developed on its own branch. This then allows for frequent Pull Requests to update the central ‘develop’ branch, allowing us to regularly ensure that the constant additions to the code base mesh well together.