### **2.1 Exploring the Cypress Project Framework**

Cypress provides a comprehensive framework for writing and running end-to-end tests for web applications. When you first set up Cypress, it creates a structured project directory to help you organize your tests and related files.

#### **2.1a Cypress Project Structure**

Here's a typical directory structure for a Cypress project:

my-cypress-project/

├── cypress/

│ ├── fixtures/

│ ├── integration/

│ ├── plugins/

│ └── support/

├── node\_modules/

├── cypress.json

├── package.json

└── README.md

1. **cypress/fixtures**: This directory is used to store test data files that you can use in your tests (e.g., JSON files).
2. **cypress/integration**: This is where your test files (spec files) go. Each test file should have the .js, .jsx, .ts, or .tsx extension.
3. **cypress/plugins**: You can extend or modify the default Cypress behaviour using plugins. This directory contains the index.js file where you can define and configure plugins.
4. **cypress/support**: This directory contains reusable support functions and commands. It includes the commands.js file for custom commands and the index.js file for global configurations.
5. **cypress.json**: This is the Cypress configuration file where you can set various options like base URL, viewport size, and timeouts.
6. **node\_modules**: This directory contains all the npm packages required for your Cypress project.
7. **package.json**: This file contains metadata about your project, including dependencies and scripts.