Test Migration to Cypress

Introduction:

The client has a web portal that connects mortgage borrowers to mortgage lenders. They had 1000+ UI Automation test cases (and some API test cases) in Java and Ruby. Maintaining both frameworks led to increased execution time and higher maintenance overhead. To streamline and modernize the testing process, we recommended migrating to a single, optimized framework using Cypress. We felt it would be in alignment with the client's goals of faster execution and reduced maintenance. Upon getting the required buy-in, we initiated the migration process.

Client Details:

Name: Confidential | **Industry:** Finserv | **Location:** USA

Technologies:

JavaScript, Cypress, Java, Ruby, Selenium, GitHub with Git Actions, Jira, IntelliJ, VS Code, TestRails, Notion

Test Migration to Cypress

Project Description:

The web platform of the client is quite innovative, meticulously designed to streamline and improve the mortgage process for both borrowers and lenders. For borrowers, it offers a user-friendly and intuitive interface to explore diverse mortgage options, compare rates from multiple lenders, and submit applications with ease. They can track their application status in real-time, upload necessary documents securely, and communicate directly with lenders, fostering transparency and reducing stress.

For lenders, the platform provides a powerful suite of tools to efficiently manage their leads, assess borrower-eligibility, and process applications. It offers a centralized dashboard for organizing client information, automating document collection, and facilitating secure communication. With advanced analytics and reporting features, lenders can gain valuable insights into market trends and optimize their offerings.

Ultimately, the platform aims to create a more transparent, efficient, and accessible mortgage ecosystem for all stakeholders.

Methodology followed:

We built a new framework from scratch and started migrating the test cases from Java/Ruby (UI and few API test cases) to Cypress. The migration was organized into multiple JIRA tickets, each representing a batch of test cases. We picked up tickets sequentially, implemented the tests, and raised PRs in GitHub upon completion. To ensure stability, each PR branch was executed multiple times using GitHub Actions with parameterized runs, verifying that the tests were not flaky. Cypress Cloud result links were attached to the PRs for review. Based on client feedback, PRs were occasionally refined through discussions before final approval and merging into the main branch.

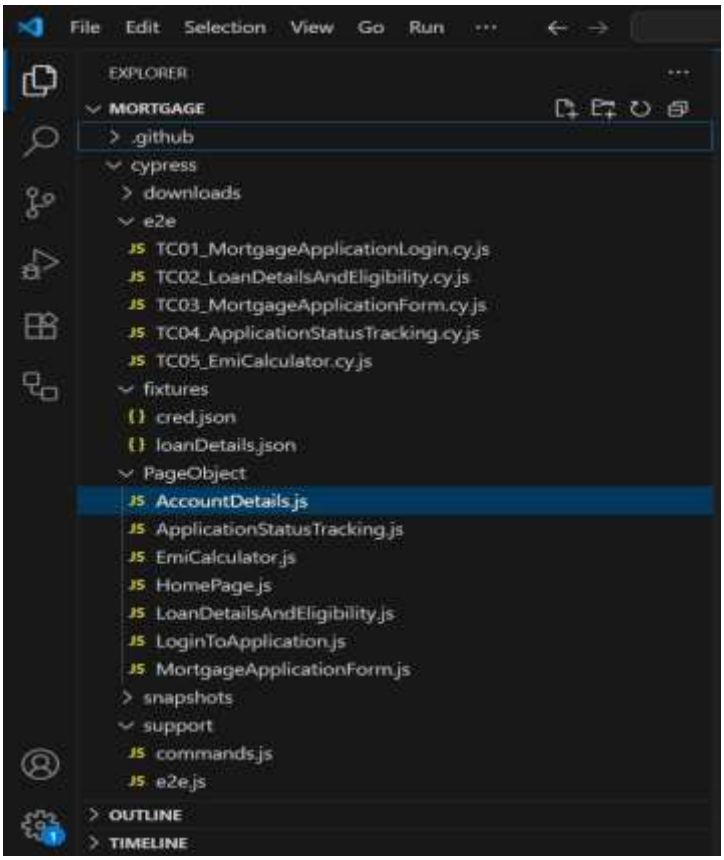
Apart from migrating test cases, we found some cases which were not covered in automation. After confirming with the client we created new tickets and corresponding test cases to address them. We also maintained written test cases actions job ran twice a day, actively monitoring test results in Cypress Cloud. If failures occurred, we analyzed the root cause—debugging and fixing issues related to automation code or logging bugs for

Test Migration to Cypress

unexpected UI/API behavior. The framework was designed to be simple and scalable, with reusable components built using the Page Object Model. We emphasized maintainability by adding meaningful comments, enforcing consistent code quality through linting, and running validations before every pull request.

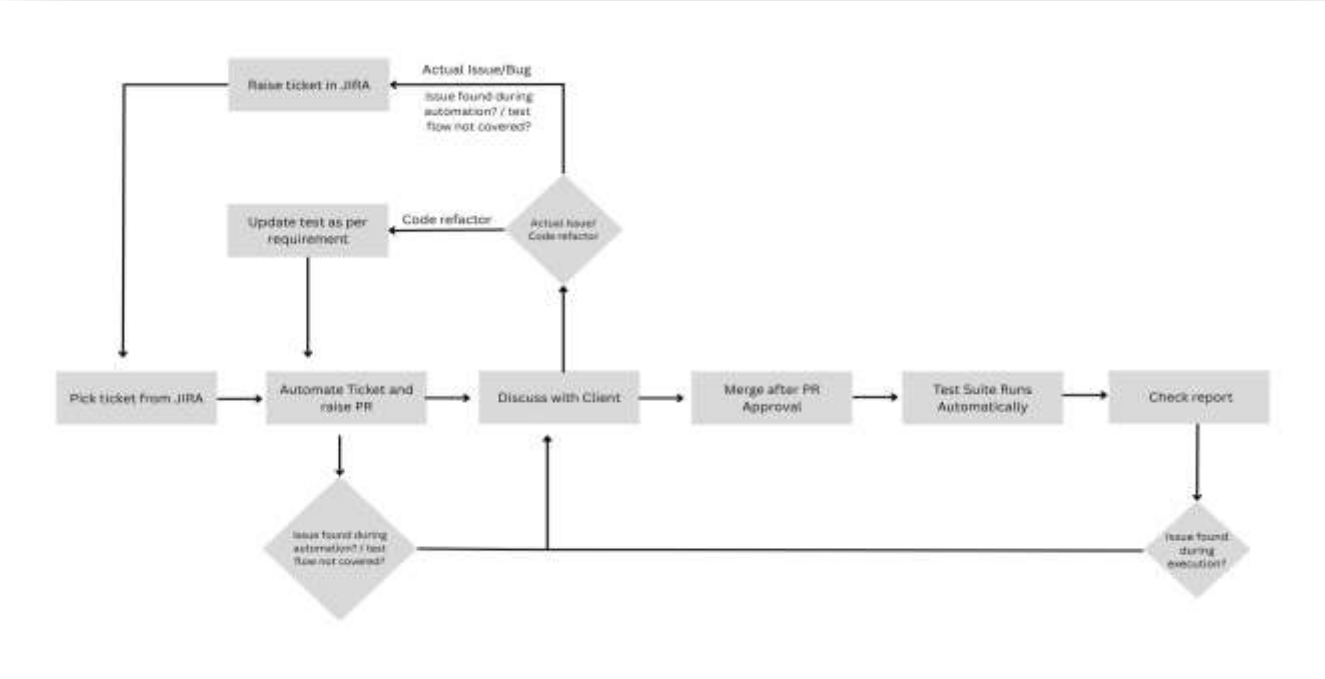
We held daily meetings with client to discuss on current/upcoming tasks and queries.

Project Structure:



Test Migration to Cypress

Flow Design:



Screenshots: