Case Study

AI-Assisted Exploratory Testing Framework for a Leading K-12 EdTech Provider

Magic EdTech

The Client

The client is a leading K-12 education technology provider offering student information system solutions for school districts, administrators, teachers, students, and families.

The Challenge

The client wanted to modernize exploratory testing without losing the flexibility that made it valuable. While exploratory testing helped uncover hidden defects and real-world user journey issues, each session depended heavily on the individual tester's memory and personal notes. This made it difficult to trace what had been tested, reproduce findings, onboard new testers, or demonstrate coverage to stakeholders. The client needed a lightweight way to make exploratory testing more consistent, traceable, and evidence-driven without slowing testers down.

Critical Success Parameters

- ✓ Support dynamic exploratory test execution while maintaining consistency across sessions.
- ✓ Provide clear traceability of test actions, observations, screenshots, and findings.
- ✓ Enable AI-led reasoning, decision-making, execution support, and documentation.
- ✓ Configure Playwright MCP to perform real browser actions, including navigation, form entry, clicks, screenshots, and UI structure capture.
- ✓ Maintain a lightweight, file-based framework that can be adapted across different projects.
- ✓ Reduce manual documentation effort while improving the quality of exploratory testing reports.
- ✓ Keep testers in control of analysis, validation, and final decision-making.

Our Approach

- ✓ Set up an AI-assisted exploratory testing model using Claude Code and Playwright MCP.
- ✓ Configured Claude Code with MCP capabilities to communicate with Playwright and perform browser-based test actions.
- ✓ Created a lightweight framework structure with key files such as CLAUDE.md, project.md, session-prompt.md, test cases, session logs, screenshots, and UI snapshots.
- ✓ Defined AI instructions, application context, session-level testing strategy, and initial test scenarios to guide exploration.
- ✓ Exploratory sessions can be initiated with a single prompt, allowing Claude to read the context, identify test areas, execute flows, and document findings.
- ✓ Used Playwright MCP to perform real user actions in the browser, capture screenshots, and collect UI structure data.
- ✓ Logged exploratory testing activity into structured session files in real time for better visibility and traceability.

Key Result Highlights

Built a lightweight AI-assisted exploratory testing framework using Claude Code and Playwright MCP.

Improved consistency in exploratory testing through reusable session prompts and structured context files.

Enabled automated documentation of test execution, observations, screenshots, UI snapshots, and findings.

Reduced dependency on manual session notes and individual tester memory.

Improved traceability by capturing what was tested, how it was tested, and what was observed during each session.

Created a reusable framework that can be adapted across different projects without heavy tooling.

Helped testers focus more on analysis and decision-making while AI supported execution and documentation.