



COURSE SYLLABUS

Security Testing and Code Review for AI-Generated Web Applications

Test, attack and secure an AI-generated web application with automated end-to-end tests, multi-model AI code review and hands-on red-teaming of access control, injection, secrets and AI features.

\$1,000 per person

10 course hours (on-demand video and labs) plus 1 hour of coaching

On-demand asynchronous training with online support, plus 1 hour of coaching.

Security Testing and Code Review for AI-Generated Web Applications

Test, attack and secure an AI-generated web application with automated end-to-end tests, multi-model AI code review and hands-on red-teaming of access control, injection, secrets and AI features.

Credential type	Certificate, issued by We Can Code IT
Clock hours	11 (10 course hours plus 1 hour of coaching)
Price	\$1,000 per person
Time	10 course hours (on-demand video and labs) plus 1 hour of coaching
Delivery	On-demand asynchronous training with online support, plus 1 hour of coaching.
Credential	Certificate of completion from We Can Code IT
How you earn it	You earn the certificate by completing the labs and a capstone review package (passing test suite, findings log with test-proven fixes, and release decision), reviewed by your coach.
Who it is for	This course is for developers, citizen developers, QA staff, technical product owners, and entrepreneurs and founders who build or approve AI-generated web applications.
Prerequisites	AI-Assisted Web Application Development, or a working web app with a database and sign-in (a reference app is provided); No test-writing experience is needed; starter tests are provided, and you adapt and extend them
Requirements	Works on Windows, macOS or Linux; Node.js and Playwright (free), a code editor such as Visual Studio Code (free); A free GitHub account and access to the app's backend (Lovable Cloud or Supabase); Access to two different AI assistants (for example Claude and ChatGPT), following your employer's data privacy rules

About this course

AI app builders now scan for common security mistakes before you publish, but a scanner cannot tell whether your access rules match how your business should work. In this course you go past the scanners and test and secure the application itself: you run the built-in scans first, then write automated end-to-end tests, have two different AI models attack the code and argue over their findings, and red-team your own app for the most common web risks and for prompt injection in AI features. Every fix is proven by a test that failed before and passes after. This course is distinct from Implementing AI in Software Testing, which teaches AI-powered QA tools; here you test and secure the application you built.

What you will be able to do

- You will be able to write automated end-to-end tests in Playwright for an app's critical paths and their most likely failures.
- You will be able to run an adversarial code review in which two different AI models independently attack the same code, and reconcile their findings.
- You will be able to test a web application for broken access control, injection, exposed secrets and weak authentication using the OWASP Top 10.
- You will be able to test an AI feature for prompt injection and unsafe output, and configure defenses.
- You will be able to fix each finding test-first and produce a findings log and a release decision backed by passing tests.

Syllabus

Module	Hours
1. Failure modes in AI-generated code Code that runs but does not work, Common defects in generated code, What built-in security scans check and miss, Who reviews the reviewer. Lab: You will run Lovable's security scan on an AI-generated app, read its code and predict three failures the scan missed, then run the app to check your predictions.	1 h
2. Automated end-to-end tests Playwright setup, Critical paths and failure cases, Test data and test users, Writing the failing test first. Lab: Starting from provided Playwright tests, you will adapt them for sign-up, sign-in and your app's main task, and add one that fails today.	2 h
3. Adversarial AI code review Builder and reviewer as different models, Review prompts that demand vulnerabilities, Making reviewers argue, Separating real findings from noise. Lab: You will have two different AI models attack your app's code independently, then reconcile their findings into one list.	1.5 h
4. Red-teaming access control, injection and secrets OWASP Top 10, Broken access control and row-level security, SQL and script injection, Exposed keys in code and the browser, Authentication settings. Lab: You will sign in as one user and try to read or change another user's data, find exposed keys, and close every hole you find.	2 h
5. Red-teaming AI features Prompt injection, Untrusted input treated as data, Unsafe and leaking output, OWASP Top 10 for LLM applications. Lab: You will plant a hidden instruction in content your app's AI feature reads, see whether it obeys, and configure a defense.	1 h

6. Test-first fixes and release decisions

1 h

Reproducing a finding as a test, Fixing and retesting, Findings log, Release decision criteria.

Lab: You will fix your top three findings, each with a test that failed before the fix and passes after.

7. Capstone

1.5 h

Test suite, Adversarial review report, Red-team findings log, Release decision.

Lab: You will present your app's test suite, review report, findings log and release decision to your coach.

Coaching sessions

1 h

1 hour of coaching. Your coach reviews your capstone with you and answers questions as you work.

Teams pool their hours into group sessions, so a team of 6 shares 6 hours of group coaching.

Live online or in-person sessions are available for an additional fee. Email admissions@wecancodeit.org for details.

Provider

We Can Code IT, LLC, 26300 Cedar Rd, Suite 1105, Beachwood, OH 44122. Registered with the Ohio State Board of Career Colleges and Schools, Registration No. 15-03-2064T. Contact: admissions@wecancodeit.org. Syllabus version 2026-10-10.