Github Auditor

Team: So Far So Good

Group Members:

Chris Yang (leader)

Thomas Zhang (scribe)

Kyle Stubbs

Nico Wong

Edward Thai

Features

- **Create deployment pipeline on Heroku:** The web application should also be accessible outside local development, and we'll want to have two deployment environments; namely, **staging** for testing features in a production-like environment, and **prod** for the actual product we will be demoing in the future.
- **Setup webhook feature:** Replace the old polling mechanism with GitHub webhooks to allow our API to get information whenever new changes are made (i.e. new Dependabot alerts, outside collaborators added, code scanning alerts)
- **Connect frontend to backend:** The frontend should be obtaining information via our own internal API instead of directly from GitHub's API. We will need to implement additional internal API endpoints to support new features, such as receiving POST requests from GitHub webhooks
- **Update frontend to better display and organize security concerns:** Organize security alerts and display them in a manner that makes more sense to the user. Examples of this would be showing the specific Dependabot alerts rather than saying that there are unchecked dependabot alerts, or showing where the specific security concerns exist.

- **Setup Mailgun:** Users should be notified via email when events are detected and need to be informed. Examples of this include security vulnerabilities such as an API key in the codebase, or Dependabot alerts that are created.

Timeline

- □ Sprint 5
  - ☑ ~~Create deployment pipeline on Heroku~~
  - □ Enable webhooks for orgs/repos
  - □ Test Rails API to ensure that correct data is being collected
  - □ Test webhooks to see if code scanner works as expected
  - □ Implement Rails API endpoints
- □ Sprint 6
  - □ Replace frontend GitHub API calls with Rails API calls
  - □ Make frontend better display the list of security concerns
  - □ Learn which organization and repository details are important
  - □ Create easy-to-read sections that contain org/repo details
- □ Sprint 7
  - □ Setup Mailgun
  - □ Test Mailgun to ensure proper notification of security vulnerabilities, changes, and Dependabot alerts
  - □ Continue to spruce up UI design
  - □ Get user feedback on the web application
  - □ Use user feedback to improve functionality/appearance and fix possible bugs that may arise
  - □ Create multiple organizations with different security concerns to test the web app
- □ Sprint 8
  - □ Test website and finalize everything
  - □ Setup demos
  - □ Create and rehearse presentations