Sprint 1 Review Report
USSF GRC Controls - Kubernetes

Sprint Dates:

September 30th, 2024 - October 04th, 2024 - Sprint 1
October 07th, 2024 - October 11th, 2024 - Sprint 1 Retrospective and Sprint 2 Planning

Links to Important Resources:

Heroku App: app link
GitHub Repository: USSF-CSCE606
GitHub Project: USSF

Team Roles:

- **Scrum Master**: Aditya Gourishetty
- **Product Owner**: Shravan Bhat
- **Developers:** Duy Vu, Maitreya Niranjan, Medha Kaushika Podipireddi, Sahil Fayaz, Shravan Bhat, Vasudha Devarakonda

Sprint Goal:

The goal of Sprint 1 was to provide an initial prototype of the SaaS application having static views of multiple pages, to show it to the client so that they can get an understanding of the application that is going to be delivered. The login via SSO feature and finalizing the tools to be used in the backend to run the scan against the Image objects were also prioritized.

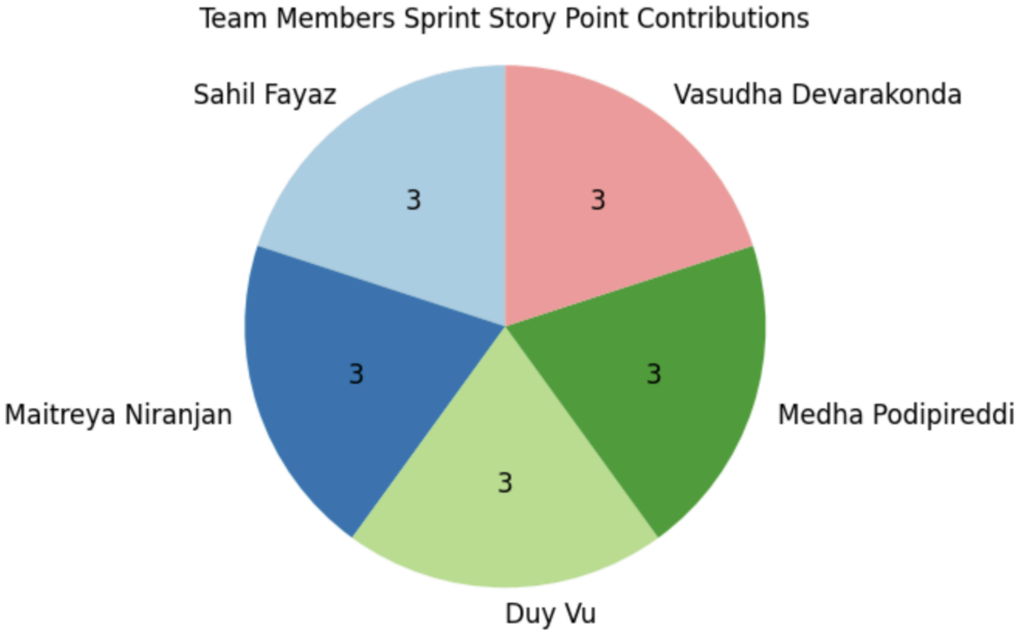
Sprint Summary and Achievements:

We were successfully able to achieve the sprint goal. We have a working prototype of the GRC scan application deployed on Heroku as well as the login feature through SSO enabled for the users of the application. We have also finalized using Trivy - an open-source security scanner as our backend tool to run the scan against the provided images. On the deployed application, currently, we have a home page, new image page, main image page, and report page, and can pull any public image and run the GRC scan for it. We have also achieved good code quality having used rubocop, rubycritic, and code coverage for rspec and cucumber test cases. All the committed stories for the sprint have been marked as done.

Story	Story Points	Sub Tasks	Sub Task Points	Assigned To	Status
Login to the application	4	Integrate Google OAuth SSO Provider by creating a new project in Google Developer Console. Implement SSO Login & Session Management	3	Sahil Fayaz	Done ✅
		Test and Validate SSO Functionality	1	Maitreya Niranjan	Done ✅
Home Page	2	Display the images already uploaded by the user along with a button to go to the “Main Image page”. Add a button that takes you to the “New Image page for GRC”.	2	Maitreya Niranjan	Done ✅
New Image Page for GRC	1	Create a form for image upload with necessary fields (e.g., image name, tags). Add a dummy upload button that will finally take you to the “Main Image Page”.	1	Duy Vu	Done ✅
Main Image page	2	Display uploaded image details on the main image page. Add a button (currently a dummy button) that links to the “Report Page”.	2	Duy Vu	Done ✅
Report page	6	Generate the raw output, parse it in a presentable way, and make a demo to pitch to the client and team.	3	Medha Podipireddi	Done ✅
		Figure out all possible solutions compatible with ruby and integrate the best possible one.	3	Vasudha Devarakonda	Done ✅

See the Appendix for story descriptions.

Team member contributions:



Duy Vu: Worked on the New Image page containing the form to create a new image and the Main Image page displaying details of a specific image for the application.

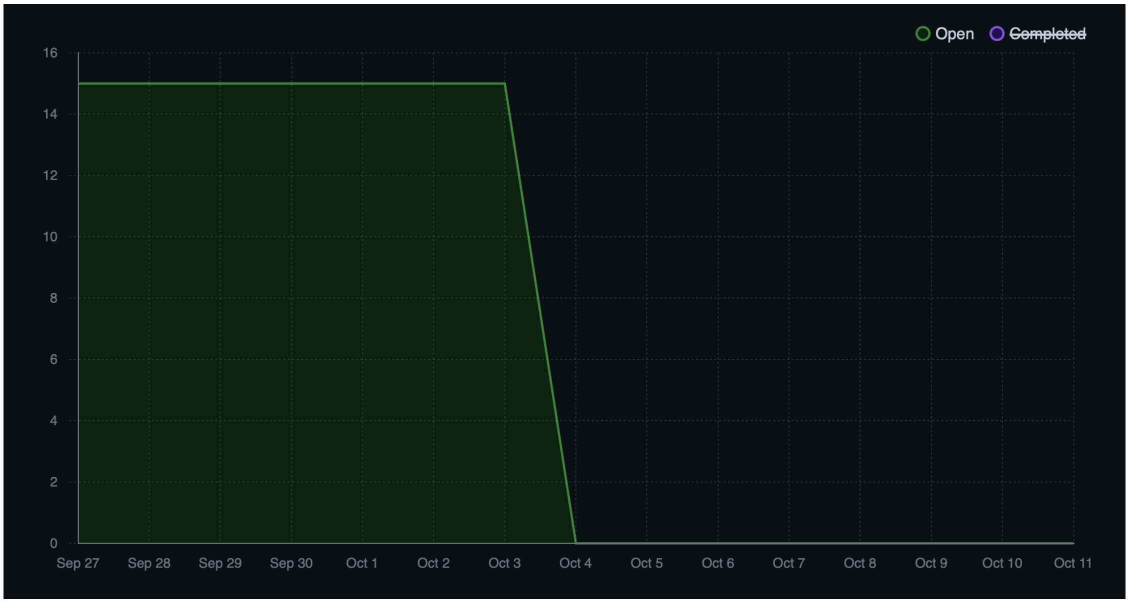
Maitreya Niranjan: Worked on the Home page containing the list of image objects, along with assisting in the Login page mechanism.

Medha Podipireddi: Worked on doing a POC for different tools to run the GRC scan and the view to display the scan results on the report page.

Sahil Fayaz: Worked on the Login Page and its mechanism for the SSO login via Google.

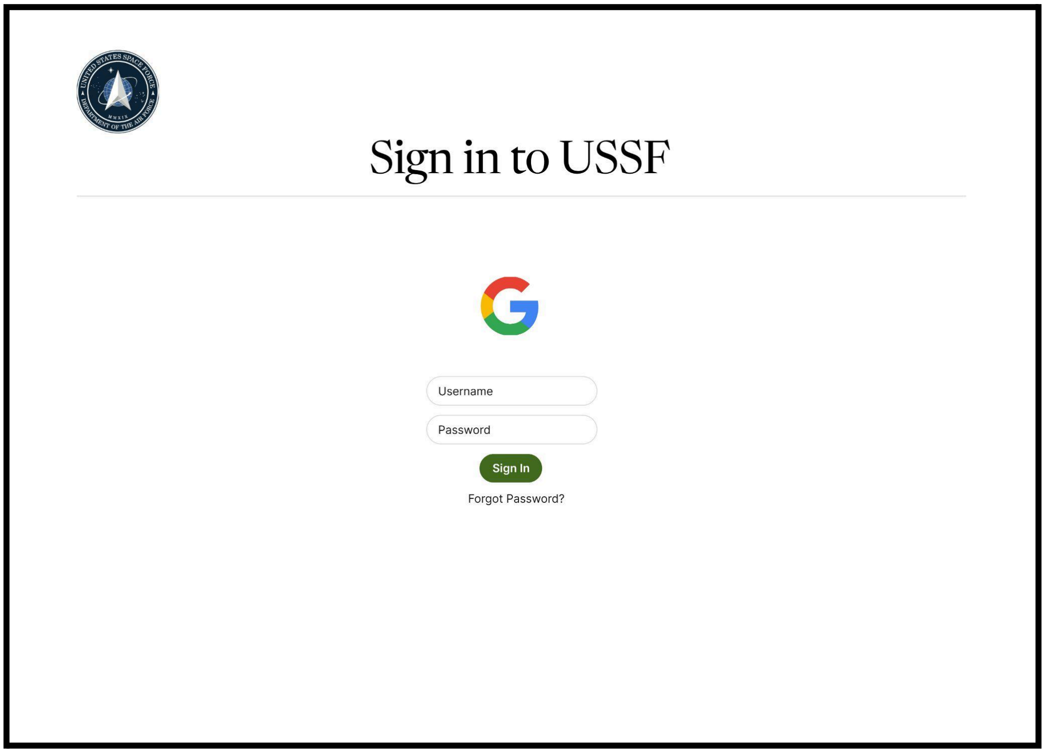
Vasudha Devarakonda: Worked on configuring the Trivy open-source image scan tool using static functions with Ruby on Rails for the project and streamlined the packaging and deployment of dependencies to Heroku.

Burn Down Chart:



The y-axis represents the story points committed for the sprint.

Design Diagrams and UI Mockups:



Sign in to USSF

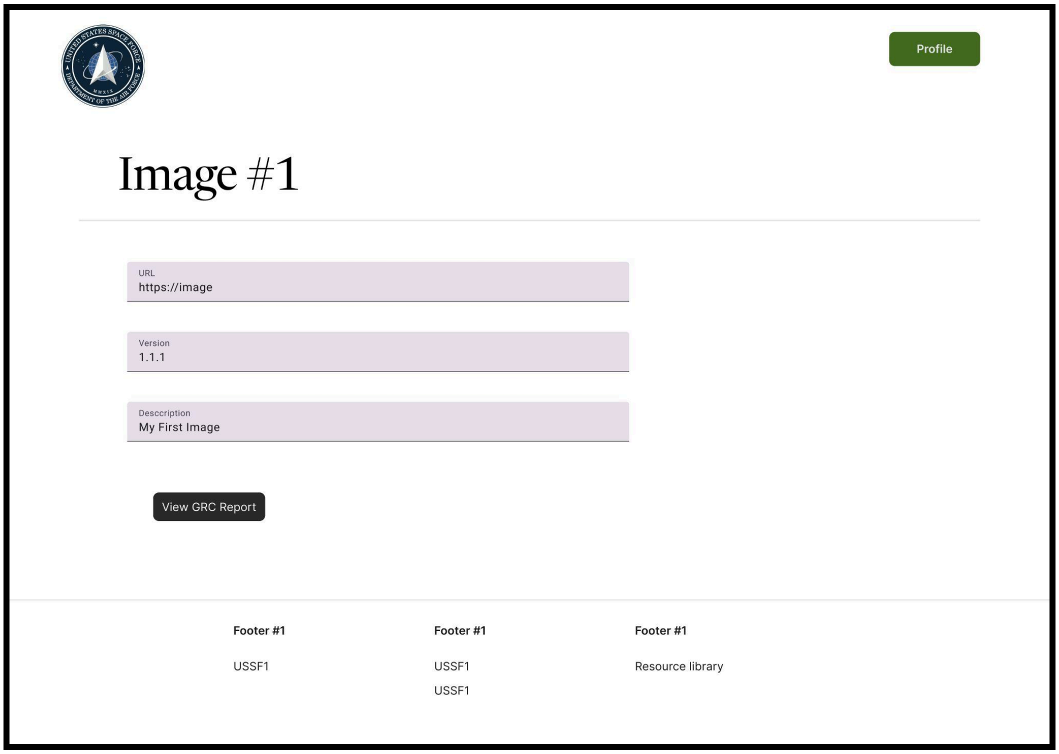
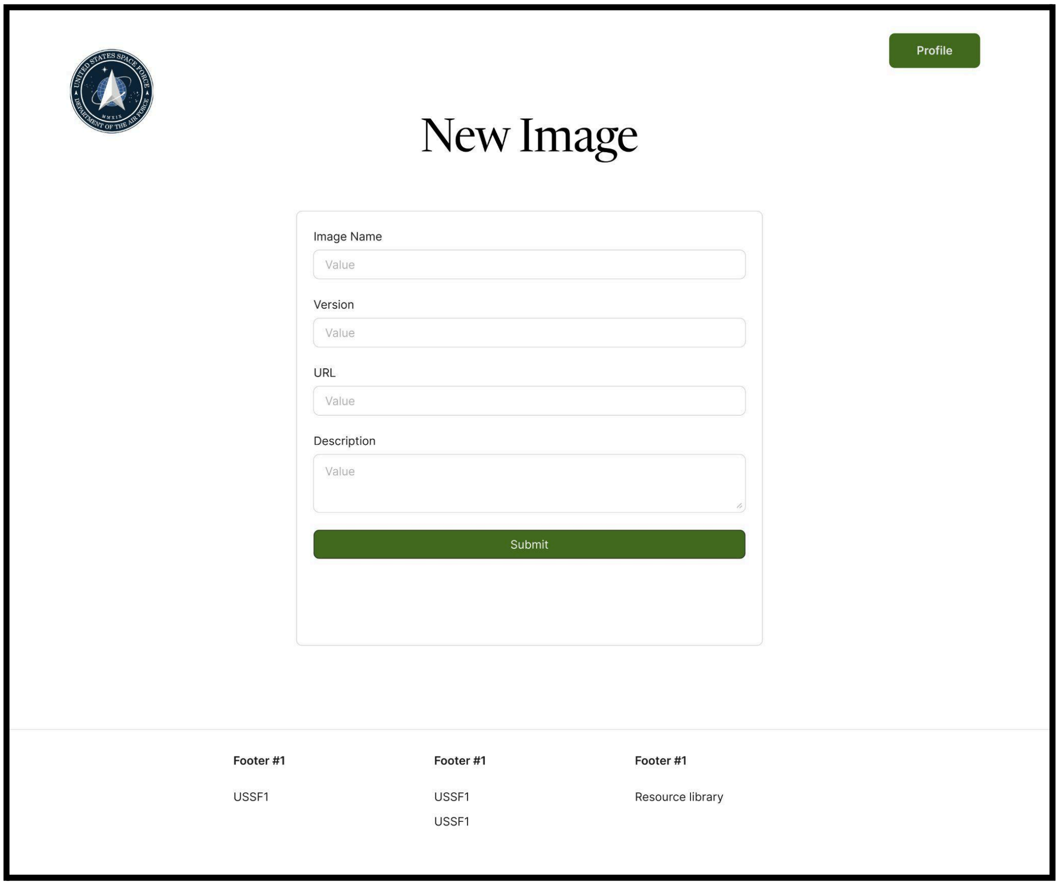


Image #1

URL
https://image

Version
1.1.1

Desccription
My First Image



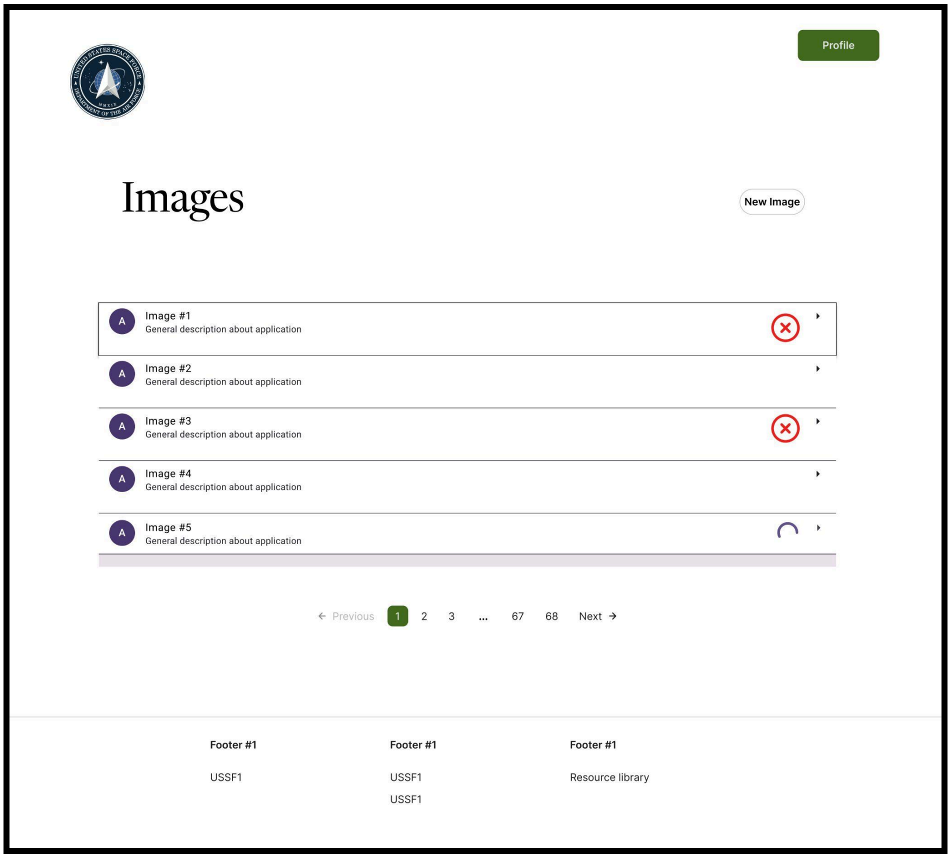
New Image

Image Name
Value

Version
Value

URL
Value

Description
Value



Profile

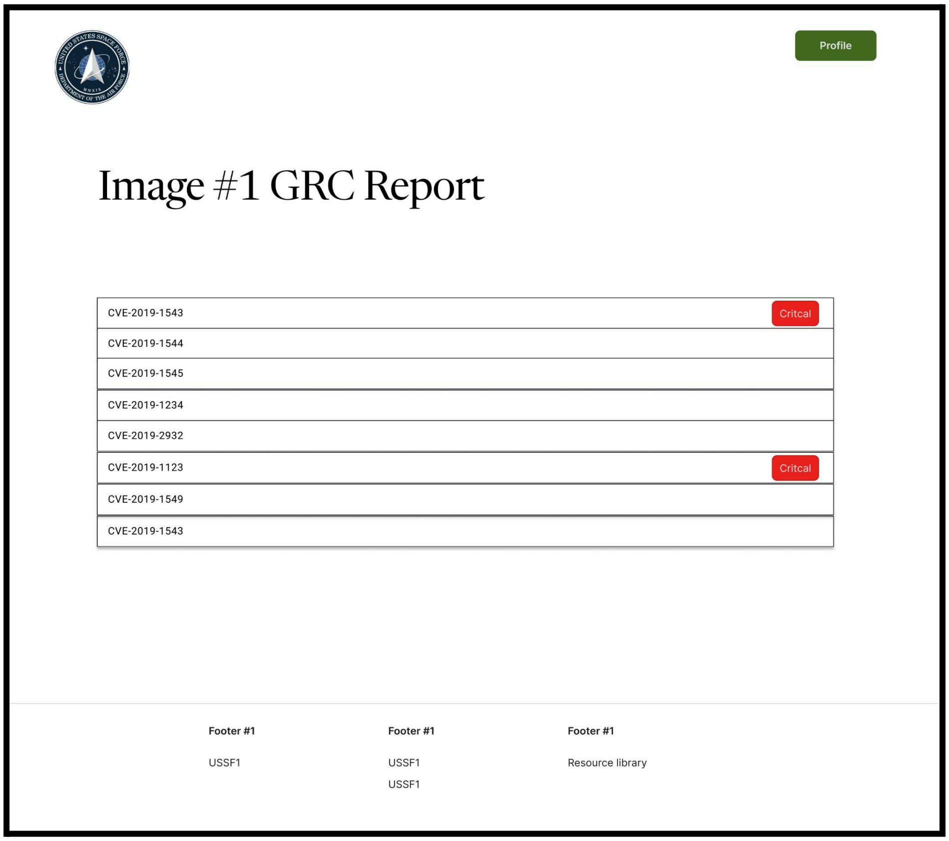
Images

New Image

- A Image #1 General description about application ⊗
- A Image #2 General description about application
- A Image #3 General description about application ⊗
- A Image #4 General description about application
- A Image #5 General description about application

← Previous 1 2 3 ... 67 68 Next →

Footer #1	Footer #1	Footer #1
USSF1	USSF1 USSF1	Resource library



Profile

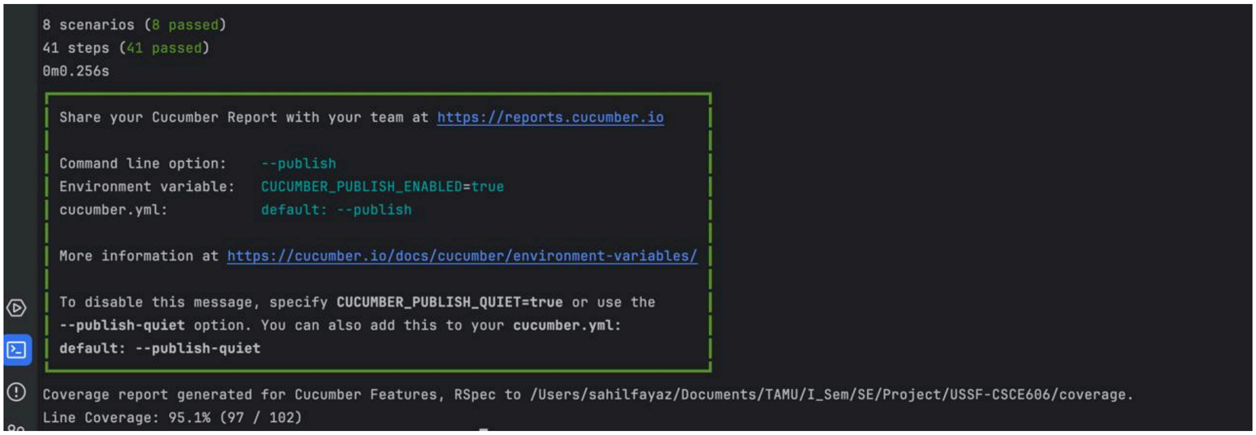
Image #1 GRC Report

CVE-2019-1543	Critcal
CVE-2019-1544	
CVE-2019-1545	
CVE-2019-1234	
CVE-2019-2932	
CVE-2019-1123	Critcal
CVE-2019-1549	
CVE-2019-1543	

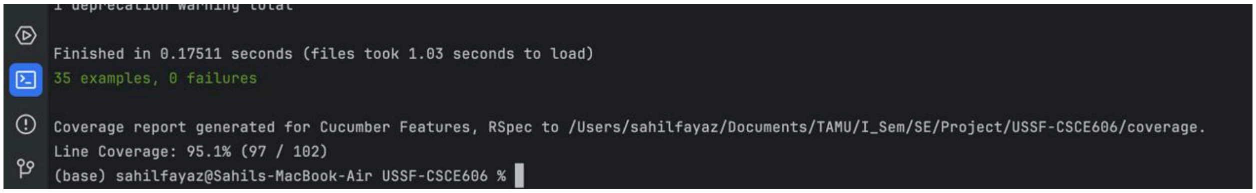
Footer #1	Footer #1	Footer #1
USSF1	USSF1 USSF1	Resource library

Code Evaluations:

Rspec and Cucumber Coverage:



```
8 scenarios (8 passed)
41 steps (41 passed)
0m0.256s
┌──────────────────────────────────────────────────────────────────────────────┐
│ Share your Cucumber Report with your team at https://reports.cucumber.io     │
│                                                                              │
│ Command line option:     --publish                                           │
│ Environment variable:    CUCUMBER_PUBLISH_ENABLED=true                       │
│ cucumber.yml:            default: --publish                                  │
│                                                                              │
│ More information at https://cucumber.io/docs/cucumber/environment-variables/ │
│                                                                              │
│ To disable this message, specify CUCUMBER_PUBLISH_QUIET=true or use the      │
│ --publish-quiet option. You can also add this to your cucumber.yml:          │
│ default: --publish-quiet                                                     │
└──────────────────────────────────────────────────────────────────────────────┘
Coverage report generated for Cucumber Features, RSpec to /Users/sahilfayaz/Documents/TAMU/I_Sem/SE/Project/USSF-CSCE606/coverage.
Line Coverage: 95.1% (97 / 102)
```

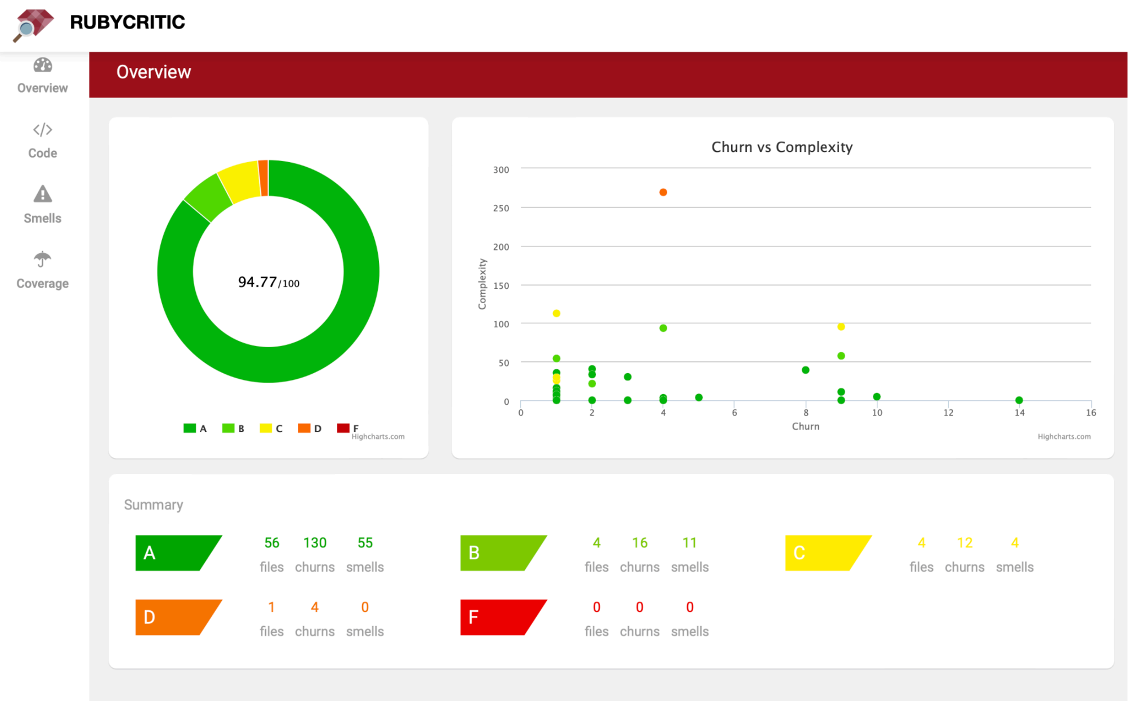


```
1 deprecation warning total

Finished in 0.17511 seconds (files took 1.03 seconds to load)
35 examples, 0 failures

Coverage report generated for Cucumber Features, RSpec to /Users/sahilfayaz/Documents/TAMU/I_Sem/SE/Project/USSF-CSCE606/coverage.
Line Coverage: 95.1% (97 / 102)
(base) sahilfayaz@Sahils-MacBook-Air USSF-CSCE606 %
```

RubyCritic:



Summary

A	56 files	130 churns	55 smells
B	4 files	16 churns	11 smells
C	4 files	12 churns	4 smells
D	1 files	4 churns	0 smells
F	0 files	0 churns	0 smells

Client Sprint MVP meeting information:

Date: 10/10/2024
Time: 18:00
Location: PETR 226
Attendees: Aditya Gourishetty, Maitreya Niranjan, Duy Vu, Prof. Shreyas Kumar, Medha Kaushika Podipireddi, Sahil Fayaz, Shravan Bhat, Vasudha Devarakonda

- The team demonstrated the deployed application to the client including the UI workflow and the different pages.
- The Login SSO mechanism and the use of Trivy - an open-source tool to run the GRC scan was communicated to the client.
- The client gave the following feedback:
    - A security analysis document for the project, including the external tools being used.
    - A product documentation including the limitations and the scope of the project.
    - The constraints and limitations should include the NIST800 controls we are committing to incorporate into the project. The team must analyze the controls Trivy is accommodating.
    - The client expects better aesthetics of the UI - A bigger Logo, better colors, and appearance overall.
    - The client also wants to have inline comments and documentation within the code for better code quality.
    - A Developers’ documentation or a more comprehensive README was also suggested.
- During the next Sprint MVP meeting, the USSF representatives will also be included.

Appendix:

Story Descriptions:

Feature: Login to the application

- **As a user** of the application
- **So that** I can log in to the application
- **I want** a login page with a login button through which I can log in via SSO.

Feature: Home Page

- **As a user** of the application
- **So that** I can view all my activities with all my images.

- **I want** a home page with a well-formatted list of all the images.

Feature: New Image Page for GRC

- **As a user** of the application
- **So that** I can add/upload a new image.
- **I want** a form where I can add details of the image.

Feature: Main Image Page

- **As a user** of the application
- **So that** I can view details of a created image
- **I want** a page where I can see the details of the image that I have created

Feature: Report page

- **As a user** of the application
- **So that** I can view the compliance report of the scan
- **I want** a report page that shows the results of the GRC scan(NIST).