Course Project

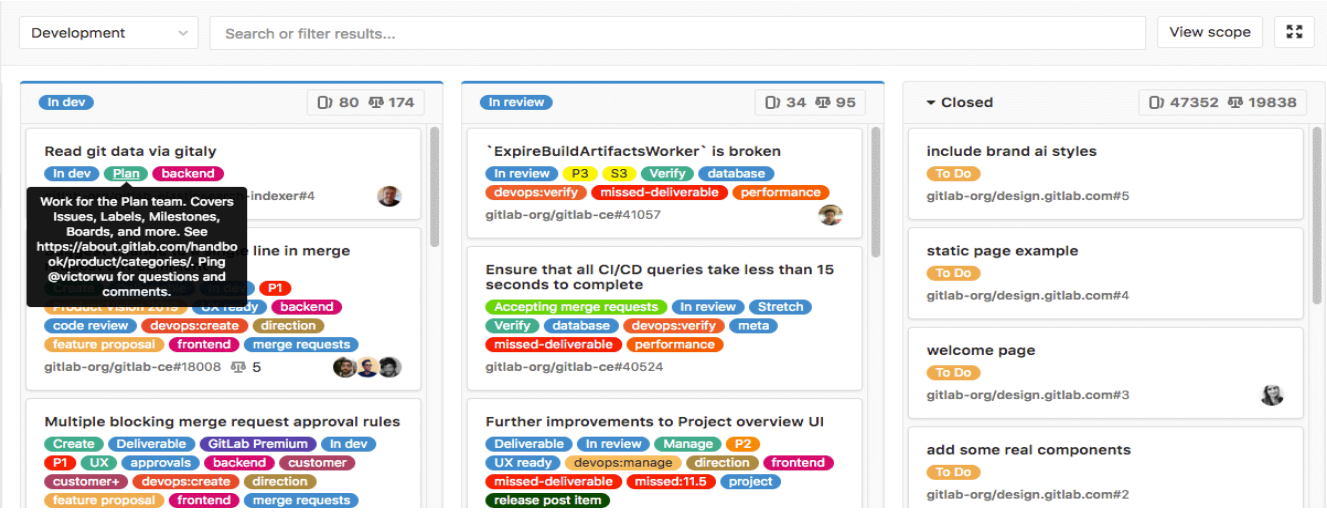
DevOps Software Development

Throughout this course you have learned about Agile Software development as well as built and deployed a number of Continuous Integration and Continuous Deployment tools. You will now be tasked to combine all of this knowledge into a working pipeline.

In Module 6, you will be assigned to a team to work with for the project, as teamwork is an essential part of real-world DevOps work. Teams will submit an Architecture Diagram showing the DevOps pipeline they have created, a Kanban project Management board showing assignment of tasks, security scans, and a video recording showing a live build of their solution.

Part 1: Use GitLab to create a Kanban Board (33% of the Grade)

Create various stories and issues and assign POCs within your team who will perform each work. Take multiple screenshots and show your team is going to split up the work. Example Kanban Board. (Please take 10+ screenshots to demonstrate your work):



Part 2: Create a Source Code Repository with Automated Code Quality Checks (33% of the Grade)

1. Previously you have used docker images to run each service individually. You will leverage docker-compose to create a .yml file that launches the following services:
    - GitLab Server - Source Code Repository
    - SonarQube – Code Quality

2. Once you have your GitLab Server and SonarQube up and running, you will want to make sure they integrate together. The following websites give examples of how you might perform this function. The goal being that any new code uploaded to GitLab is automated for code quality scanning.

- https://github.com/gabrie-allaigre/sonar-gitlab-plugin/
- https://medium.com/@speedforcerun/sonarqube-with-gitlab-ci-setup-stepbystep-dotnetcore-version-ee555d37d52e

3. Once you have the integration between GitLab completed you should download the source code of OWASP Juice Shop.

4. Record a video of you creating a Gitlab Repository for juice-shop, uploading your source code into GitLab, and walk us through the findings SonarQube provides about the juice-shop code. This video should show recommendations from a cyber perspective to a developer on how to secure their code/project.

Part 3: Building Docker Registry and Performing Automated Container Image Scanning (34% of the Grade)

1. Using the Gitlab Server you have previously created you will now enable a Container Registry.

2. Next you should create a Jenkins web server. Please note you will need to install Snyk on your Jenkins server to perform container scanning.

3. Next you will need to enable GitLab & Jenkins integration.

4. You should also download the container image for OWASP Juice Shop.

5. Upload your OWASP Juice Shop Container Image to GitLab.

6. Create a Jenkins Pipeline that will scan your juice-shop container for vulnerabilities with Snyk.
   - If a High or Critical vulnerability is found then break the pipeline build.
   - Example of how a Jenkins pipeline can break on vulnerabilities

7. Record a video of you uploading your juice-shop container image to Gitlab, then show how your code upload automates running your Jenkins pipeline. Show the output of your container security scan (i.e. Snyk) and provide results on if your scan broke/passed the build.

8. Please submit all of your custom code. Example Docker-Compose Files, Jenkins scripts, etc. Please note you may have to use a zip/rar to upload to Blackboard.

Congratulations you are now finished with the DevOps class. We appreciate your diligence and hope you have enjoyed learning cutting edge IT skills that help you in the workplace!