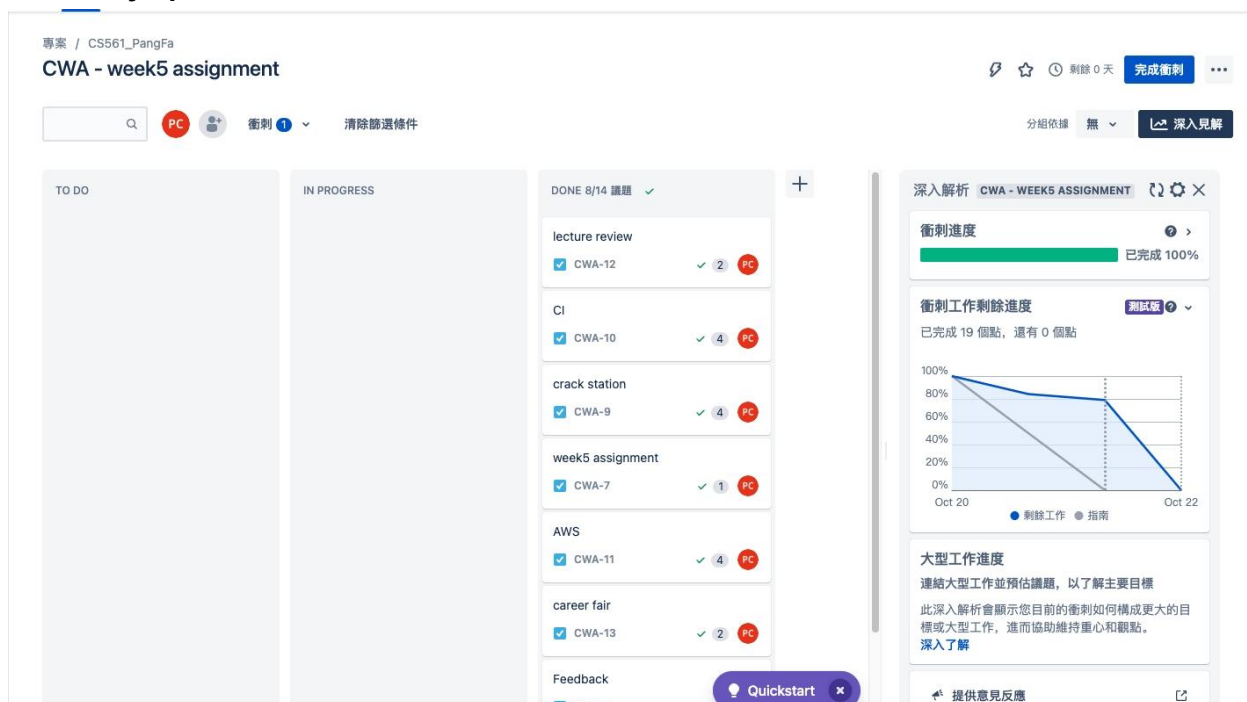


Part 1: My sprint in Jira:



Part 2: Feedback to you.

The screenshot shows a survey form from Oregon State University. The form is titled 'Oregon State University' and includes a logo. The text on the form reads: 'This survey is completely anonymous. Your input will help us better calibrate the course :)'. Below this text is a large empty box for the user to provide feedback. At the bottom of the form, it says 'We thank you for your time spent taking this survey. Your response has been recorded.' and 'Survey Powered By Qualtrics'.

Part 3: Security

TestCase1 :

input string =
"13fbd79c3d390e5d6585a21e11ff5ec1970cff0c8efd86fb78a56a5145ed7739dcb00c78581c53757a38d8cbd20d9932ba948efaa364bb62651d5ad4"

Expected result: "ktv"

Explanation: The input String is actually the hashed string, and the original text is "ktv", Therefore, the function should return "ktv" after cracking the string and it sure does.

```

final class MyLibraryTests: XCTestCase {
    func testEncryptUsingSha1() {
        let str = "a"

        // When
        let hashedstr = MyLibrary.encryptUsingSha1(from: str)

        // Then
        XCTAssertNotNil(hashedstr)
        XCTAssertEqual(hashedstr, "SHA1 digest: 86f7e437faa5a7fce15d1ddcb9eaaea377667b8")
    }

    func testCrack() {
        // Given
        let myLibrary = MyLibrary()
        let hashedpass =
            "13fbd79c3d390e5d6585a21e11ff5ec1978c8ff8e8fd86fb78a56a5145ed7739dc08c78581c53767a38d8cbd20d9932ba948efaa364bb62651d5ad4"

        // When
        let plainpassword = myLibrary.crack(password: hashedpass)

        // Then
        XCTAssertNotNil(plainpassword)
        XCTAssertEqual(plainpassword, "ktv")
    }
}

```

Test Suite 'Selected tests' started at 2022-10-23 11:25:43.912
 Test Suite 'MyLibraryTests.xctest' started at 2022-10-23 11:25:43.913
 Test Suite 'MyLibraryTests' started at 2022-10-23 11:25:43.913
 Test Case '-[MyLibraryTests.MyLibraryTests testCrack]' started.
 plainpass = ktv
 Test Case '-[MyLibraryTests.MyLibraryTests testCrack]' passed (0.005 seconds).
 Test Suite 'MyLibraryTests' passed at 2022-10-23 11:25:43.919.
 Executed 1 test, with 0 failures (0 unexpected) in 0.005 (0.006) seconds
 Test Suite 'MyLibraryTests.xctest' passed at 2022-10-23 11:25:43.920.
 Executed 1 test, with 0 failures (0 unexpected) in 0.005 (0.007) seconds
 Test Suite 'Selected tests' passed at 2022-10-23 11:25:43.921.
 Executed 1 test, with 0 failures (0 unexpected) in 0.005 (0.009) seconds
 Program ended with exit code: 0

Testcase2 :

inputString = "CS561"

Expected Result: nil

Explanation: The input string is just a random string. It's not a hash string by SHA1.

Therefore, after passing it to the crack function, the function could not crack it and it returned nil.

```

final class MyLibraryTests: XCTestCase {
    func testEncryptUsingSha1() {
        let str = "a"

        // When
        let hashedstr = MyLibrary.encryptUsingSha1(from: str)

        // Then
        XCTAssertNotNil(hashedstr)
        XCTAssertEqual(hashedstr, "SHA1 digest: 86f7e437faa5a7fce15d1ddcb9eaaea377667b8")
    }

    func testCrack() {
        // Given
        let myLibrary = MyLibrary()
        let hashedpass = "CS561"

        // When
        let plainpassword = myLibrary.crack(password: hashedpass)

        // Then
        XCTAssertEqual(plainpassword, nil)
    }
}

```

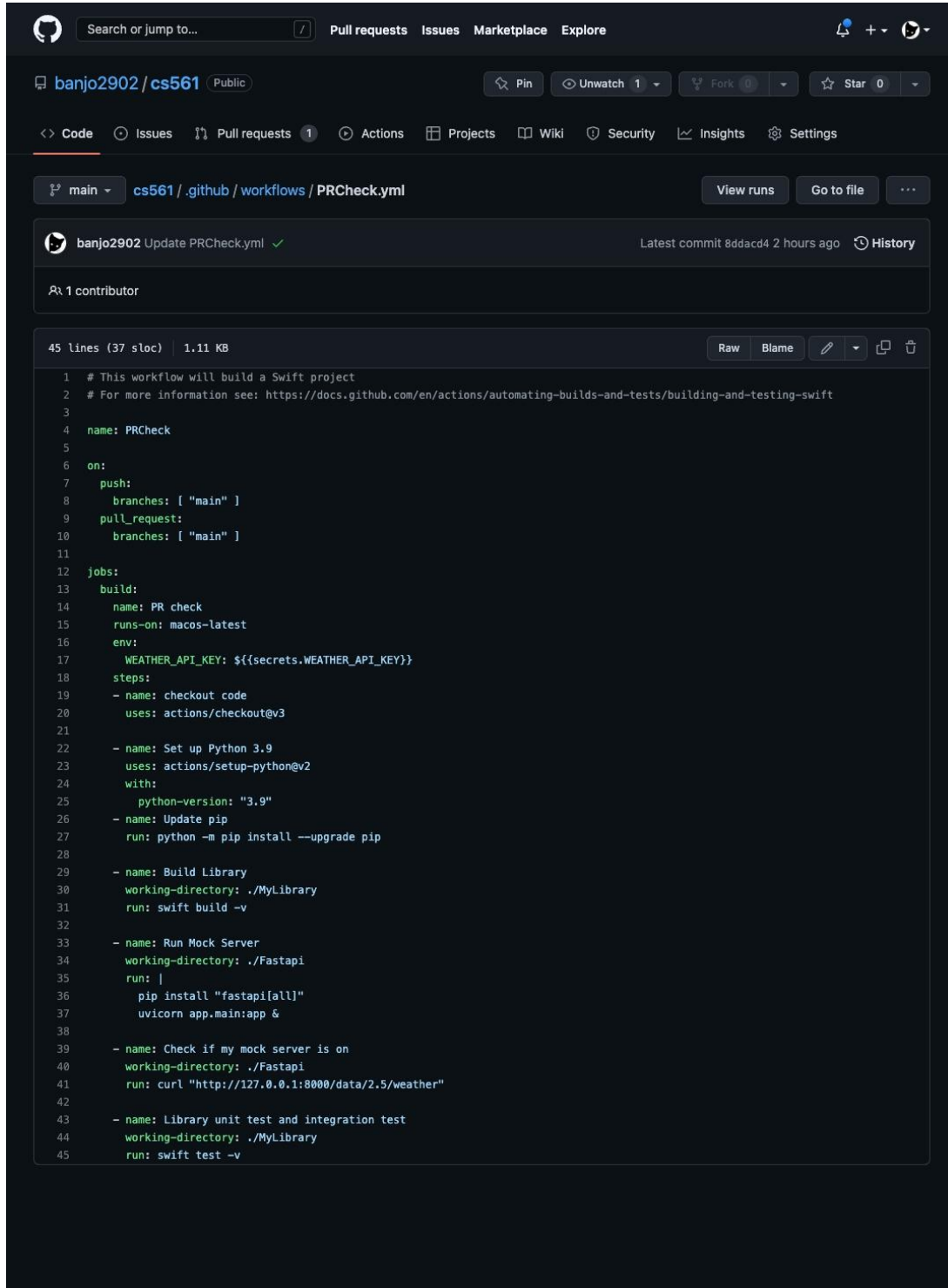
Test Suite 'Selected tests' started at 2022-10-23 11:58:27.782
 Test Suite 'MyLibraryTests.xctest' started at 2022-10-23 11:58:27.783
 Test Suite 'MyLibraryTests' started at 2022-10-23 11:58:27.784
 Test Case '-[MyLibraryTests.MyLibraryTests testCrack]' started.
 Test Case '-[MyLibraryTests.MyLibraryTests testCrack]' passed (0.005 seconds).
 Test Suite 'MyLibraryTests' passed at 2022-10-23 11:58:27.789.
 Executed 1 test, with 0 failures (0 unexpected) in 0.005 (0.006) seconds
 Test Suite 'MyLibraryTests.xctest' passed at 2022-10-23 11:58:27.789.
 Executed 1 test, with 0 failures (0 unexpected) in 0.005 (0.007) seconds
 Test Suite 'Selected tests' passed at 2022-10-23 11:58:27.791.
 Executed 1 test, with 0 failures (0 unexpected) in 0.005 (0.009) seconds
 Program ended with exit code: 0

Part 4: CI

Link to my repo: <https://github.com/banjo2902/cs561/pull/2>

Screenshots of how my CI works:

1. Github actions: build up PRCheck.yml



The screenshot shows the GitHub interface for the repository `banjo2902 / cs561`. The file `.github / workflows / PRCheck.yml` is selected, showing its content. The workflow is named `PRCheck` and is triggered on `push` to the `main` branch or a `pull_request` to `main`. The workflow includes several jobs: `build` (which runs a PR check), `Set up Python 3.9`, `Update pip`, `Build Library`, `Run Mock Server`, `Check if my mock server is on`, and `Library unit test and integration test`.

```
1 # This workflow will build a Swift project
2 # For more information see: https://docs.github.com/en/actions/automating-builds-and-tests/building-and-testing-swift
3
4 name: PRCheck
5
6 on:
7   push:
8     branches: [ "main" ]
9   pull_request:
10    branches: [ "main" ]
11
12 jobs:
13   build:
14     name: PR check
15     runs-on: macos-latest
16     env:
17       WEATHER_API_KEY: ${secrets.WEATHER_API_KEY}
18     steps:
19       - name: checkout code
20         uses: actions/checkout@v3
21
22       - name: Set up Python 3.9
23         uses: actions/setup-python@v2
24         with:
25           python-version: "3.9"
26
27       - name: Update pip
28         run: python -m pip install --upgrade pip
29
30       - name: Build Library
31         working-directory: ./MyLibrary
32         run: swift build -v
33
34       - name: Run Mock Server
35         working-directory: ./Fastapi
36         run: |
37           pip install "fastapi[all]"
38           uvicorn app.main:app &
39
40       - name: Check if my mock server is on
41         working-directory: ./Fastapi
42         run: curl "http://127.0.0.1:8000/data/2.5/weather"
43
44       - name: Library unit test and integration test
45         working-directory: ./MyLibrary
46         run: swift test -v
```

2. Using pull request to trigger CI

The screenshot shows a GitHub pull request titled "README content update #2". At the top, it indicates that user "banjo2902" wants to merge 1 commit into the "main" branch from the "pangfa/add-tests" branch. Below this, there are tabs for Conversation (0), Commits (1), Checks (0), and Files changed (2). A comment from "banjo2902" is visible, stating "No description provided." To the right, there are sections for Reviewers (No reviews), Assignees (No one—assign yourself), Labels (None yet), Projects (None yet), Milestone (No milestone), and Development (Successfully merging this pull request may close these issues). A "Merge pull request" button is present. Below the pull request details, there is a section for checks. It shows "Some checks haven't completed yet" with 1 in progress and 1 successful check. The checks listed are "PRCheck / PR check (pull_request)" which is in progress, and "GitGuardian Security Checks" which is successful. A message states "This branch has no conflicts with the base branch" and "Merging can be performed automatically." At the bottom, there is a "Write" tab and a "Preview" tab for comments.

3. Successfully passed the CI check (I set up the environment variable in Actions Secret for the weather api key, so it passes my integration test.)

The screenshot shows a GitHub Actions workflow run for the "PR check" job. The job has succeeded in 1m 49s. The workflow steps are listed as follows:

- Set up job
- checkout code
- Set up Python 3.9
- Update pip
- Build Library
- Run Mock Server
- Check if my mock server is on
- Library unit test and integration test
- Post Set up Python 3.9
- Post checkout code
- Complete job

Part 5. AWS: S3

The screenshot shows a web browser window displaying the DigitalCloud training interface. The browser's address bar shows the URL: <https://digitalcloud.training/courses/aws-certified-solutions-architect-associate-hands-on-labs/sections/section-7-amazon-simple-storage-service-s3-1hr-55m/>. The browser's tab bar shows several open tabs, including Gmail, YouTube, Cambridge Dictio..., Oregon State Univ..., Piazza - Ask Ans..., GPA Calculator | O..., LeetCode - The W..., My Training Cours..., Final Examination..., JPG to PDF Conve..., Graduate Scholar..., Search Classes, and How to Read the... The browser's toolbar shows various icons for navigation and search.

The DigitalCloud training interface is displayed in the browser window. The interface has a dark theme. The top navigation bar includes the DigitalCloud logo, Training, Pricing, Cheat Sheets, Free Resources, and About. The user's name, PangFa, is displayed in the top right corner. The left sidebar shows a list of sections and topics, including Week 5, How to submit the Cr..., Mandatory Ethical Res..., CTF - Collaborative In..., Syllabus - CS 561 - Co..., Week 5 - CS 561 - Com..., According to the U.S. F..., ctf training not answer..., CWA 看板 - Agile 看板..., Section 7: Amazon Simple..., hw5 - Google Drive..., hw5 - Google Docs..., and Men's Colosseum Black Dr... The main content area is titled "Section Content" and shows a list of topics for Section 7: Amazon Simple Storage Service (S3). The topics are listed with their completion status (green checkmark for complete, red X for incomplete). The topics are: Section 7—Introduction (complete), Amazon S3 Overview (complete), Amazon S3 Storage Classes (complete), [HOL] Create Amazon S3 Bucket (complete), IAM Policies, Bucket Policies and ACLs (complete), [HOL] Access Control Lists (ACLs) (complete), [HOL] Bucket and User Policy Practice (complete), S3 Versioning, Replication and Lifecycle Rules (complete), [HOL] Versioning and Replication (complete), and [HOL] Lifecycle Rules (complete). The right sidebar shows a list of sections and topics, including Section 1: Let's Get Started... (9 Lessons, complete), Section 2: AWS Identity an... (19 Lessons, complete), Section 3: Amazon Elastic... (30 Lessons, complete), Section 4: Elastic Load Bal... (24 Lessons, complete), Section 5: AWS Organizati... (11 Lessons, incomplete), Section 6: Amazon Virtual ... (27 Lessons, incomplete), Section 7: Amazon Simple ... (28 Lessons, complete), Section 8: DNS, Caching, ... (20 Lessons, incomplete), Section 9: Block and File S... (22 Lessons, incomplete), Section 10: Docker Contai... (17 Lessons, incomplete), Section 11: Serverless Appl... (21 Lessons, incomplete), Section 12: Databases and... (34 Lessons, incomplete), and Section 13: Deployment an... (19 Lessons, incomplete).