

SIDDHARTH CHURIWAL

Case Studies

Github : [SiddharthChuriwal/CaseStudies \(github.com\)](https://github.com/SiddharthChuriwal/CaseStudies)

User Story for Header

As a user when I am successfully logged in to the website the user should be directed to the dashboard including the header. The header should contain the web page name, a button to upgrade and the drop-down menu including the options of About, Support, change Password and logout.

ACCEPTANCE CRITERIA: when the user is successfully logged in to the website. And have access to the dashboard page the user should be directed to the dashboard page which has a header which is included in every page of the website after the successful login. The header should include the web page navigated to (e.g. Dashboard) on the left top corner of the website. On the right side the header should include a drop-down menu header as the profile photo of the user with the user's full name. In the drop-down menu the options include About Support, Change Password and logout button.

On click on the About option from the drop-down menu the user should be displayed a pop-up showing Company Name, Version, Active Employees and Employees Terminated.

On click on the Support option from the drop-down menu the user should be directed to the Support page which has the basic information of the company and the customer support contact us link which should be hyperlinked to the website osupport@orangehrm.com.

On click on the Change Password option from the drop-down menu the user should be directed to the page asking for the changing the password of the current user logged in. Which has fields for current password, New Password and confirm password including the save and cancel buttons.

On click on the logout option from the drop-down menu the user should be logged out of the website and should be redirected to the login page.

The button of upgrade should be available to the left of the drop-down menu.

Priority 1

Effort 3

Test Case

1

Description: Validate the header content

Given the user is logged in to the website with the username: "Admin" and password "admin123"
And has the access to the dashboard page of the website.

Then the Header should display the web page name (e.g., Dashboard) on the left, an upgrade button, and a drop-down menu with the user's profile photo and full name on the right.

2

Description: Validate if The Drop-down menu is clickable

Given the user is logged in to the website with the username: "Admin" and password "admin123"
When the user clicks on the drop-down menu

Then the options About, Support, Change Password, and Logout should be displayed.

3

Description: Validate the about option in the drop-down menu

Given the user is logged in to the website with the username: "Admin" and password "admin123"

When the user clicks on the drop-down menu

And clicks on the "About" option.

Then the user should be able to view the pop-up which shows the Company Name, Version, Active Employees and Employees Terminated.

4

Description: Validate the support option in Drop-down menu

Given the user is logged in to the website with the username: "Admin" and password "admin123"

When the user clicks on the drop-down menu

And clicks on the "Support" option.

Then the user should be directed to Support page with company information and a contact link which is hyperlinked to the contact email (ossupport@orangehrm.com).

5

Description: Validate the change password option in Drop-down menu

Given the user is logged in to the website with the username: "Admin" and password "admin123"

When the user clicks on the drop-down menu

And clicks on the "Change Password" option.

Then the User is directed to the Change Password page, which contains fields for current password, new password, confirm password, and Save and Cancel buttons.

6

Description: Validate the change password option in Drop-down menu

Given the user is logged in to the website with the username: "Admin" and password "admin123"

When the user clicks on the drop-down menu

And clicks on the "Logout" option.

Then the User is logged out successfully and is redirected to the login page of the website.

User Story: Time at work section in Dashboard page

Description:

As a user, when I log in using valid credentials (username: Admin, password: admin123), then I should be landed on the dashboard page, and I should be able to see Time at work section.

ACCEPTANCE CRITERIA:

When the user is on the dashboard page, then Time at work section should be displayed.

The section should be in the form of container. The section should be present in the center with a little left offset. The Time at Work section container should have a heading - Time at Work with clock symbol present on the left side of the heading. Below heading - 'Time at work' logged in user image should be present in the form of circle. The image circle should be very small. On the right side of the profile image - 'Punched Out' subheading should be displayed. Punched out subheading should be written in small. Below 'Punched out' subheading Punched out should be displayed along with the timestamp at which logged in user was last punched out. Below user profile and punched out

detail – duration of punched in should be displayed. On the right side of the punched in time duration field, a clock symbol should be visible. On clicking on this clock symbol – time page should be displayed. On below of punched in time duration field – ‘This week’ text should be displayed. Below this text, this week duration should be displayed in the form (Month Day). Month should be abbreviated. On right of this week info – this week punched in should be displayed. Below this – for every day of this week punched in time should be displayed in the form of vertical bar chart.

Priority: 3

Effort: 4

Test Case 1: Display of “Time at Work” Section

- **Given:** The user is on the dashboard page.
- **When:** The page loads.
- **Then:** The “Time at Work” section should be displayed in the center with a slight left offset.

Test Case 2: Heading and Icon Display

- **Given:** The user is on the dashboard page.
- **When:** The “Time at Work” section is displayed.
- **Then:** The section should have a heading “Time at Work” with a clock symbol on the left side of the heading.

Test Case 3: User Profile Image Display

- **Given:** The user is on the dashboard page.
- **When:** The “Time at Work” section is displayed.
- **Then:** A very small circular profile image of the logged-in user should be displayed below the heading.
- image.

Test Case 4: “Punched Out” Timestamp Display

- **Given:** The user is on the dashboard page.
- **When:** The “Time at Work” section is displayed.
- **Then:** The timestamp of the last punch out should be displayed below the “Punched Out” subheading.

Test Case 5: Display of Punched In Duration

- **Given:** The user is on the dashboard page.
- **When:** The “Time at Work” section is displayed.

- **Then:** The correct duration of the punched-in time should be displayed below the user profile and punched-out details.

Test Case 6: Clock Symbol Functionality

- **Given:** The user is on the dashboard page.
- **When:** The user clicks on the clock symbol next to the punched-in duration field.
- **Then:** The user should be redirected to the time page.

Test Case 7: Display of This Week's Duration

- **Given:** The user is on the dashboard page.
- **When:** The "Time at Work" section is displayed.
- **Then:** The duration for the current week should be displayed in the format (Month Day), with the month abbreviated.

Test Case 8: Display of This Week's Punched In Time

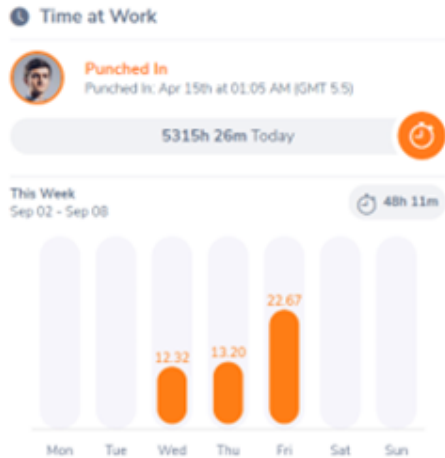
- **Given:** The user is on the dashboard page.
- **When:** The "Time at Work" section is displayed.
- **Then:** The total punched-in time for the current week should be displayed to the right of the week's duration.

Test Case 9: Vertical Bar Chart Displaying punched in time for every day of this week

- **Given:** The user is on the dashboard page.
- **When:** The "Time at Work" section is displayed.
- **Then:** A vertical bar chart showing the correct punched-in time for each day of the current week should be displayed below the week's duration

Test Case 10: Punched in time for today should match punched in time duration for today in vertical bar

- **Given:** The user is on the dashboard page.
- **When:** The "Time at Work" section is displayed.
- **Then:** Punched in time for today should match punched in time duration for today in vertical bar

Defect No.	1
Description	Punched In time duration detail for today is wrong
Steps to reproduce	<ol style="list-style-type: none"> 1. Open the edge browser 2. Go to https://opensource-demo.orangehrmlive.com/web/index.php 3. Login using valid credentials (username: admin, password: admin123) 4. View the Punched in time duration for today in 'Time at Work' section
Timestamp	06/09/2024 12:29
Severity	Low
Priority	High
Assignee	Dharya
Screenshot	 <p>The screenshot displays the 'Time at Work' interface. At the top, it shows a user profile and the status 'Punched In' for Apr 15th at 01:05 AM (GMT 5:5). Below this, a large bar indicates '5315h 26m Today' with a refresh button. A section titled 'This Week' (Sep 02 - Sep 08) features a bar chart showing time durations for each day of the week. The values are: Mon (0), Tue (0), Wed (12.32), Thu (13.20), Fri (22.67), Sat (0), and Sun (0). The total for the week is 48h 11m.</p>
Expected Results	Punched in time duration should match the punched in time for <u>today</u> (Friday) in vertical bar (22.67)
Actual Results	Punched in time duration is showing 5315h 26m which is not equal to punched in time for <u>today</u> (Friday) in vertical bar (22.67)

USER STORY FOR THE SEARCH BAR

As a user when I use the search bar of the website (OrangeHRM) to search for the available sections on the sidebar, then I should be able to search and find the section.

Acceptance Criteria

- The search bar should be prominently displayed at the top left corner of the website (OrangeHRM)
- The search bar should show the matching section on the sidebar

T E S T C A S E S

TC001

Scenario: Validation of the basic search functionality

Given the user is on the sidebar of the OrangeHRM website

When the user types something into the search bar

Then the search results should display the matching section of the sidebar

TC002

Scenario: Validation of the working of sidebar while searching

Given the user is on the sidebar of the OrangeHRM website

When the user types something into the search bar

Then the sidebar should show only the matching sections.

User Story For navigation Bar

As a user, when I login the website, I should be able to see a side navigation bar which should contain various categories which lead to their respective pages.

Acceptance Criteria:

When the user logs in they are able to see the side navigation bar on the website which contains the websites logo on the top and contains various categories. The categories should be

1. Search option
2. PIM option
3. Leave option
4. Time option
5. Recruitment option
6. My Info option
7. Performance option
8. Dashboard option
9. Directory option
10. Maintenance option
11. Claim option
12. Buzz option

All the options should have their respective logos on the left side. There should be an orange-colored arrow button on the sidebar which should be able to close and open the navigation bar. When the navigation bar is closed, only the logos of the respective options should be visible in orange color. All the logos should be clickable. All options in the nav bar should be white in color with grey colored text but when mouse pointer is brought near any option it should change to hand pointer and the color should be changed to orange.

TC 001

Scenario: Verify side navigation bar visibility after login

Given the user is on the login page.

When the user logs in.

Then the user should be able to see the side navigation bar on the website.

TC 002

Scenario: Verify website logo on the side navigation bar

Given the user is on the side navigation bar.

When the user logs in.

Then the website's logo should be visible at the top of the side navigation bar

AND the side navigation bar should contain the following categories with their respective logos:

Category	
Search	
PIM	
Leave	

Time	
Recruitment	
My Info	
Performance	
Dashboard	
Directory	
Maintenance	
Claim	
Buzz	

TC 003

Scenario: Verify orange-colored arrow button functionality

Given the user is on the side navigation bar

When the user clicks on the orange-colored arrow button

Then the navigation bar should close

And only the logos of the respective options should be visible in orange color

When the user clicks on the orange-colored arrow button again

Then the navigation bar should open

TC 004

Scenario: Verify clickable logos

Given the user is on the side navigation bar

When the user clicks on any logo

Then the user should be directed to the respective page

TC 005

Scenario: Verify hover effect on navigation options

Given the user is on the side navigation bar

When the mouse pointer is brought near any option

Then the pointer should change to a hand pointer

And the color of the option should change to orange

And all options in the nav bar should be white in color with grey colored text

User Story: Buzz Section

As a User when I want to post updates, share photos and videos, and view the most recent, most liked, and most commented posts in the Buzz section then I should be able to engage with my colleagues and stay updated with the latest activities and popular posts.

Acceptance Criteria:

Post Updates:

- The user should be able to type a message in the post input field.
- The user should be able to click on the post button.
- The user should see their post appear in the Buzz feed.

Share Photos:

- The user should be able to click on the share photos button.
- The user should be able to select photos from their device.
- The user should be able to write a description for the photos.
- The user should be able to click on the share button.
- The user should see the photos and description appear in the Buzz feed.

Share Videos:

- The user should be able to click on the share videos button.
- The user should be able to select videos from their device.
- The user should be able to write a description for the videos.
- The user should be able to click on the share button.
- The user should see the videos and description appear in the Buzz feed.

View Most Recent Posts:

- The user should be able to click on the most recent posts button.
- The user should see the most recent posts displayed in the Buzz feed.

View Most Liked Posts:

- The user should be able to click on the most liked posts button.
- The user should see the posts with the most likes displayed in the Buzz feed.

View Most Commented Posts:

- The user should be able to click on the most commented posts button.

1) POST SEARCH FUNCTIONALITY

As a user, when I am on the buzz section of the website (OrangeHRM) and I try to post anything, then I should be able to see my buzz.

Acceptance Criteria

- The user should be able to type the sentence on the search bar.

- The user should be able to click on the post button.
- The user should be able to view his post on the buzz feed.

TC -1

Scenario : Validate the user can successfully post a message

Given the user is on the Buzz section of the website

When the user types "Any Message" into the post input field

And the user clicks on the post button

Then the post with that message should be visible on the Buzz feed

Test Case 2: Empty Post

Scenario: Validate that the user is not able to put empty posts

Given the user is on the Buzz section of the website

When the user leaves the post input field empty

And the user clicks on the post button

Then an error message should be displayed

And the post should not be added to the Buzz feed

2) PHOTOS FUNCTIONALITY

Test Case 1: Successful Photo Share

Scenario: Validate that the users successfully shares a photo with a description

Given the user is on the Buzz section of the website

When the user clicks on the share photos button

And the user selects a photo from their device

And the user writes something in the description field

And the user clicks on the share button

Then the photo with the description should be visible on the Buzz feed

Test Case 2: Cancel Photo Share

Scenario: Validate the user can cancel the photo share process

Given the user is on the Buzz section of the website

When the user clicks on the share photos button

And the user selects a photo from their device

And the user clicks on the cancel button

Then the photo should not be posted on the Buzz feed

3) VIDEOS FUNCTIONALITY

Test Case 1: Successful Video Share

Scenario: Validate that the users can successfully shares a video with a description

Given the user is on the Buzz section of the website

When the user clicks on the share video button

And the user pastes a videos url

And the user writes something in the description field

And the user clicks on the share button

Then the video url with the description should be visible on the Buzz feed

Test Case 2: Cancel Video Share

Scenario: Validate that the users can successfully cancel a video share

Given the user is on the Buzz section of the website

When the user clicks on the share video button

And the user puts in an url in the video field

And the user clicks on the cancel button

Then the video url should not be posted on the Buzz feed

4) MOST RECENT POSTS FUNCTIONALITY

Test Case 1: View Most Recent Posts

Scenario: Validate that the user can view the most recent posts

Given the user is on the Buzz section of the website

When the user clicks on the most recent posts button

Then the most recent posts should be displayed under the most recent posts section

Test Case 2: Refresh Most Recent Posts

Scenario: Validate that the user sees the new posts(if any) after refreshing

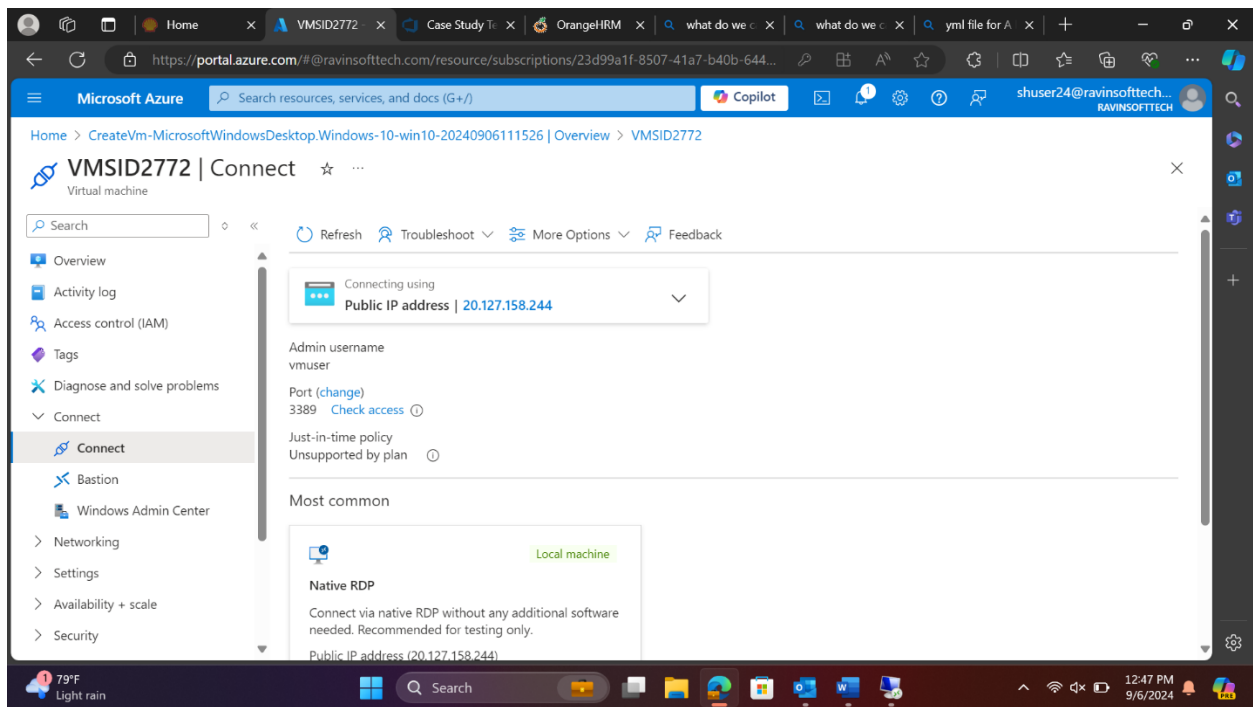
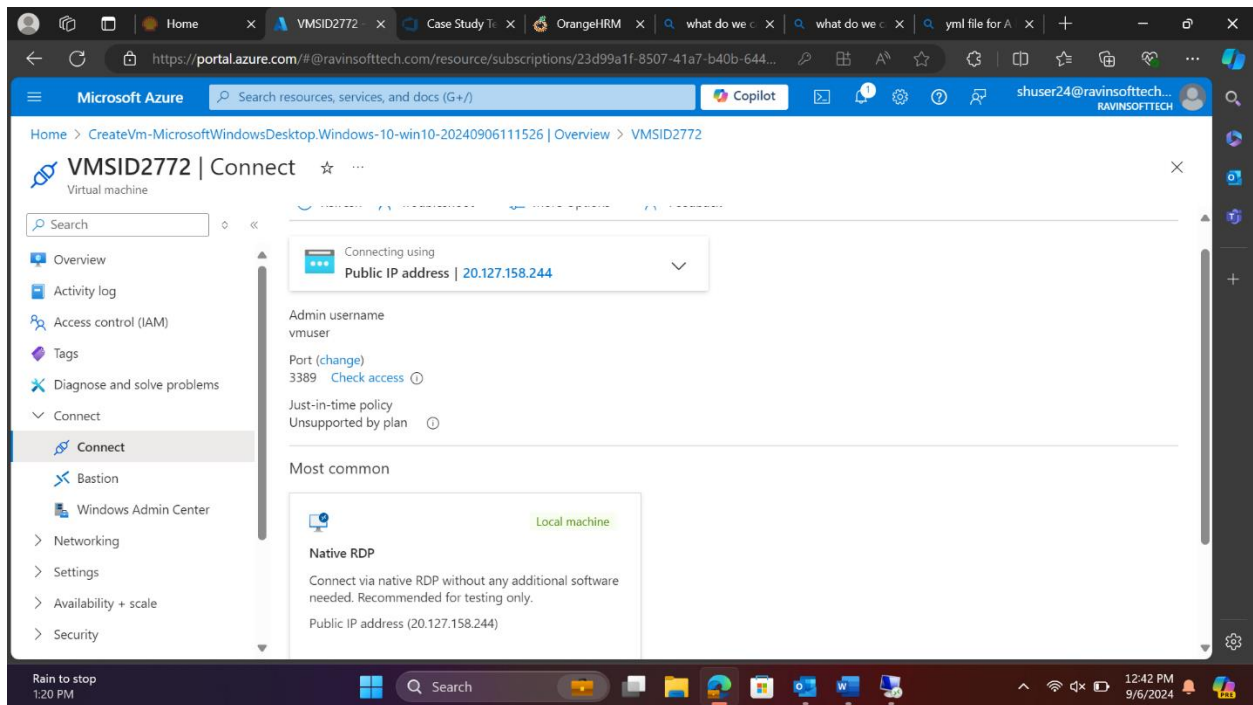
Given the user is on the Buzz section of the website

When the user clicks on the most recent posts button

And new posts have been added since the last view

Then the most recent posts section should display the latest posts

VM Creation



Azure DevOps Sprint Creation

The screenshot displays the Azure DevOps Sprints taskboard for the 'Case Study Team'. The interface is organized into four columns: 'To Do', 'Doing', and 'Done'. The 'To Do' column contains four items: '23 Create Reports option in PIM Page', '17 Create PIM Page', '16 Create Admin Page', and '15 My Info - Personal Details'. The 'Doing' column contains three items: '25 Writing Test Case', '26 Writing Bug Report', '27 Writing Test Case', '29 Writing Test Case', '30 Writing Bug Report', and '32 Writing Bug Report'. The 'Done' column contains three items: '28 Writing Bug Report', '31 Writing Test Case', and '31 Writing Test Case'. Each item is represented by a card showing its title, state (e.g., 'Unassigned', 'Doing', 'Done'), and a 'To Do' button. The left sidebar shows the 'Case Study' project with various views like Overview, Boards, Backlogs, and Sprints. The bottom status bar indicates the current temperature is 76°F and the weather is mostly cloudy.

Item ID	Title	State	Column
23	Create Reports option in PIM Page	Unassigned	To Do
17	Create PIM Page	Unassigned	To Do
16	Create Admin Page	Unassigned	To Do
15	My Info - Personal Details	Unassigned	To Do
25	Writing Test Case	Unassigned	Doing
26	Writing Bug Report	Unassigned	Doing
27	Writing Test Case	Unassigned	Doing
29	Writing Test Case	Unassigned	Doing
30	Writing Bug Report	Unassigned	Doing
32	Writing Bug Report	Unassigned	Doing
28	Writing Bug Report	Unassigned	Done
31	Writing Test Case	Unassigned	Done
31	Writing Test Case	Unassigned	Done

YML file Creation

The screenshot displays a web browser window with multiple tabs. The active tab is a GitHub repository page for 'SiddharthChuriwal / CaseStudies'. The URL in the address bar is 'https://github.com/SiddharthChuriwal/CaseStudies/blob/main/.github/workflows/blank.yml'. The repository's navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. On the left, a file explorer shows the repository structure, including a '.github/workflows' directory containing 'blank.yml'. The main content area shows the 'blank.yml' file with a commit message 'SiddharthChuriwal Create blank.yml' and a green checkmark. The file content is a GitHub Actions workflow configuration. The Windows taskbar at the bottom shows the search bar and several application icons. The system clock indicates 7:03 AM on 9/6/2024.

CaseStudies / .github / workflows / blank.yml

SiddharthChuriwal Create blank.yml ✓ 21f6820 · 51 minutes ago History

Code Blame 36 lines (29 loc) · 1.11 KB Raw Download Edit

```
1 # This is a basic workflow to help you get started with Actions
2
3 name: CI
4
5 # Controls when the workflow will run
6 on:
7   # Triggers the workflow on push or pull request events but only for the "main" branch
8   push:
9     branches: [ "main" ]
10  pull_request:
11    branches: [ "main" ]
12
13 # Allows you to run this workflow manually from the Actions tab
14 workflow_dispatch:
15
16 # A workflow run is made up of one or more jobs that can run sequentially or in parallel
```


WorkFlow Created Built and Rebuilt

This screenshot shows the GitHub Actions interface for the repository 'SiddharthChuriwal / CaseStudies'. The 'All workflows' tab is selected, displaying a list of workflow runs. The left sidebar contains navigation links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Under the 'Actions' section, there are links for 'All workflows', 'CI', 'Management', 'Caches', 'Attestations', and 'Runners'. The main area shows '3 workflow runs' with columns for Event, Status, Branch, and Actor. The runs listed are 'Update calci.java', 'Create calci.java', and 'Create blank.yml', all on the 'main' branch and completed successfully.

Event	Status	Branch	Actor
Update calci.java	Completed	main	SiddharthChuriwal
Create calci.java	Completed	main	SiddharthChuriwal
Create blank.yml	Completed	main	SiddharthChuriwal

This screenshot shows the details of a specific workflow run, 'Update calci.java #3', which is part of the 'CI' workflow. The 'build' job is selected, showing a list of steps that were executed successfully. The 'build' job summary indicates it succeeded 7 minutes ago in 4s. The steps listed are: 'Set up job', 'Run actions/checkout@v4', 'Run a one-line script', 'Run a multi-line script', 'Post Run actions/checkout@v4', and 'Complete job'. The 'Re-run all jobs' button is visible in the top right corner.

Step	Duration
Set up job	0s
Run actions/checkout@v4	1s
Run a one-line script	0s
Run a multi-line script	0s
Post Run actions/checkout@v4	0s
Complete job	0s

calculator java code basi x Basic Calculator Program x github - Search x Update calci.java - Siddh x java code with yaml file f x

https://github.com/SiddharthChuriwal/CaseStudies/actions/runs/10733894925

SiddharthChuriwal / CaseStudies

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Update calci.java #3

Summary

Jobs

- build

Run details

- Usage
- Workflow file

Triggered via push 7 minutes ago

SiddharthChuriwal pushed -> 0f01fc4 main

Status Success

Total duration 14s

Artifacts -

blank.yml

on: push

build 4s

Re-run all jobs

Type here to search

calculator java code basi x Basic Calculator Program x github - Search x Workflow runs - Siddhar x java code with yaml file f x

https://github.com/SiddharthChuriwal/CaseStudies/actions

SiddharthChuriwal / CaseStudies

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Actions

New workflow

All workflows

CI

Management

- Caches
- Attestations
- Runners

All workflows

Showing runs from all workflows

Filter workflow runs

2 workflow runs

	Event	Status	Branch	Actor
Create calci.java	CI #2: Commit 4c4fb0c pushed by SiddharthChuriwal	Success	main	40 minutes ago 17s
Create blank.yml	CI #1: Commit 21f6820 pushed by SiddharthChuriwal	Success	main	42 minutes ago 13s