CSL 2060 - Project

ReadMe File, Test Case Design and Automation scripts

for

Click to Order

An Online Food Ordering System

Team Members

Kshitiz(B19CSE111)

Barun(B19CSE020)

Readme File

Introduction

The software 'Click to Order 'is an online food ordering system that allows the customer to avail the IITJ Canteen services from the comfort of their browsers without the hassle of waiting in long queues and wasting a lot of time. The system helps to manage the working of the canteen more effectively and efficiently by computerizing meal ordering, billing and inventory control. It also allows the canteen to know about the details of the customer like number of people for a particular interval of time, their orders and accordingly the canteen can make arrangements to improve the customer experience. They can also calculate daily expenditure and profit.

The website is built using HTML, CSS and PHP with MySQL as the database to manage the data. HTML is used to display the basic static pages, CSS is used for styling. For managing the back end we used PHP. MySQL database contains all the data that we show on our webpages upon users request.

Functionalities:

- User:
    - **Food Menu and Categories -** Customers can order food from the food menu or go through individual categories available to select their food item they wish to order.
    - **Select Food Items** - Customers can select the desired food items from the menu on the basis of their choice or various categories available.
    - **Order Food** - Customers can order the food they wish to order. After placing the if the order is confirmed the "Congrats! Your Food Item is Ordered Successfully" message appears on the screen thereby confirming the order.

- Admin:
    - **Log-in** - The admin can login into the admin panel using his/her unique Username and Password. In order to maintain privacy the password entered is encrypted using the md5 hashing algorithm.
    - **Manage Admin** - The manage admin functionality allows to manage more than one admin.
    - **Manage Dashboard** - Admin can keep track of the number of food categories, menu, number of orders made along with the total revenue generated.
    - **Manage Food Items** - Admin can add new food categories and food items as per the availability of the raw materials provided to them by the supplier.
    - **Manage Order -** Admin can manage the order, change the status of the order as Ordered, On Delivery, Delivered and cancelled depending on the situation.

Technologies Used in the Project

- HTML - It is used to display the content on web pages.
- CSS - CSS is used to design the webpage.
- PHP - PHP is used as a backend technology. It connects the database with the front end.
- MYSQL - MYSQL is used to create Database for our system.
- XAMPP - The software can be locally deployed using XAMPP, an open source PHP server that allows local hosting along with Testing and is pre-installed with tools such as MySQL.

Steps to Run the Project

1. Download and install xampp.
2. Copy the project folder into xampp->htdocs folder.
3. Create a database on phpmyadmin using MYSQL database.
4. Open the browser and search http://localhost/clicktoorder to run the website.

Team Members

The project is developed by Barun (B19CSE020) and Kshitiz (B19CSE111).

Test Case Design

Code

1. Testing php-mySql connection

```php
<?php

$dbname = 'food-order';
$dbuser = 'root';
$dbpassword = '';
$dbhost = 'localhost';


$connect = mysqli_connect($dbhost, $dbuser, $dbpassword) or die("Unable to Connect to '$dbhost'");


mysqli_select_db($connect, $dbname) or die("Could not open the db '$dbname'");


if ($connect == True){
    echo "Connected Succesffully to the database";
}else{
    echo "Could not connect to the database server";
}

?>
```

2. Testing php mySQL pdo connection

```php
<?php


$dbhost = "localhost";
$dbusername = "root";
$dbpassword = "";
$dbname = "food-order";

try {
    $conn = new PDO("mysql:host=$dbhost; dbname=$dbname", $dbusername, $dbpassword);

    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    echo "Connected successfully";
    }

catch(PDOException $e)
    {
    echo "Connection failed: " . $e->getMessage();
    }

?>
```

3. Testing Database connection in php

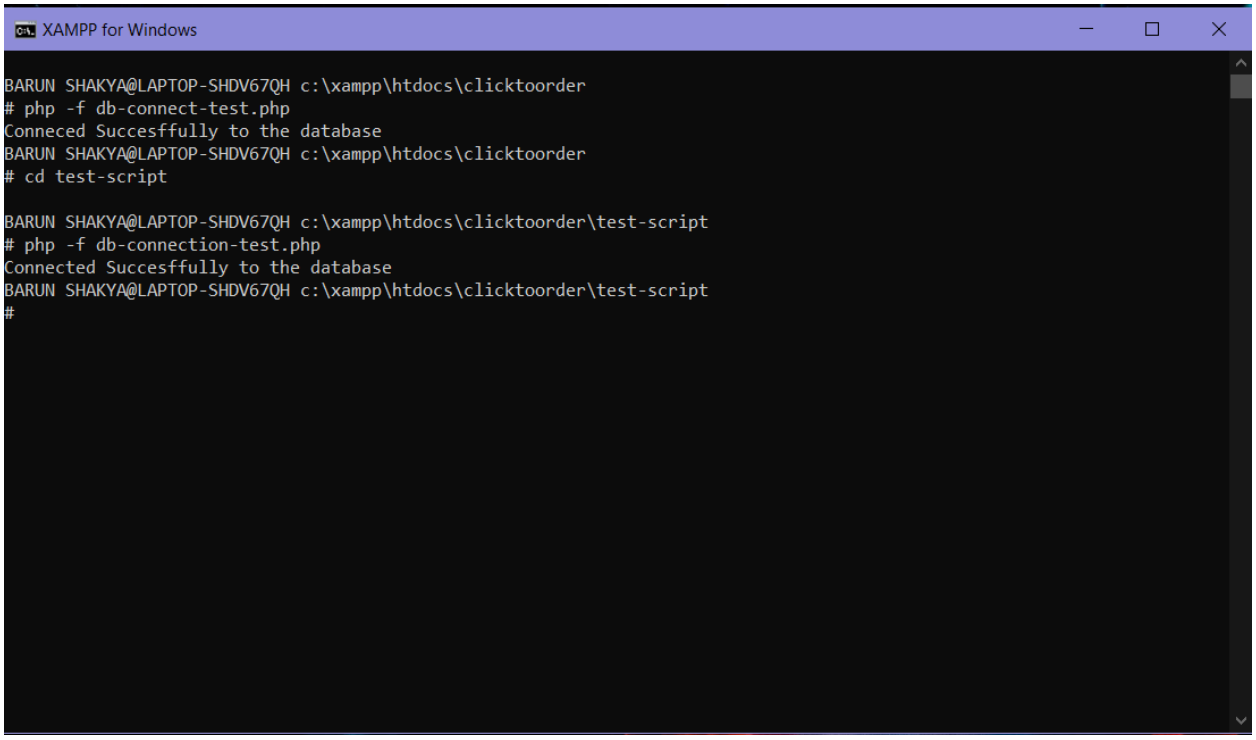
```php
<?php

    $server = "localhost";
    $username = "root";
    $password = "";
    $database = "food-order";

    $con = mysqLi_connect($server, $username, $password, $database);

    if(!$con){
        die ("Connection Terminated!". mysqLi_connect_error());

    }

    else {
        echo "Connected Succefully!";
    }

?>
```

4. Testing php connection to Database

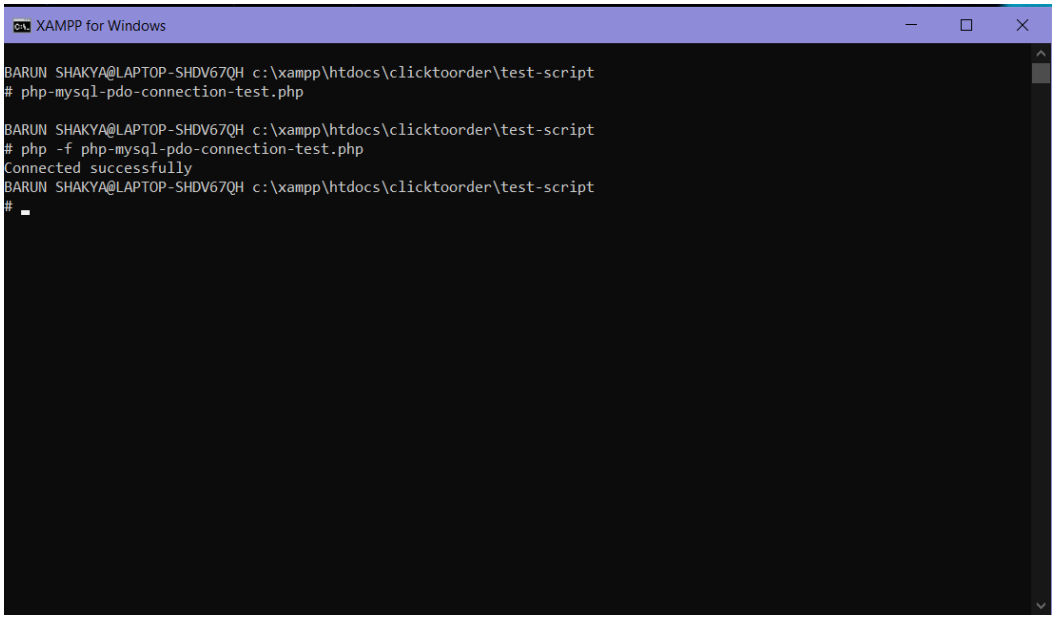
```php
<?php

$servername = "localhost";
$username = "root";
$password = "";


$conn = new mysqli($servername, $username, $password);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";


?>
```

Testing results screenshot

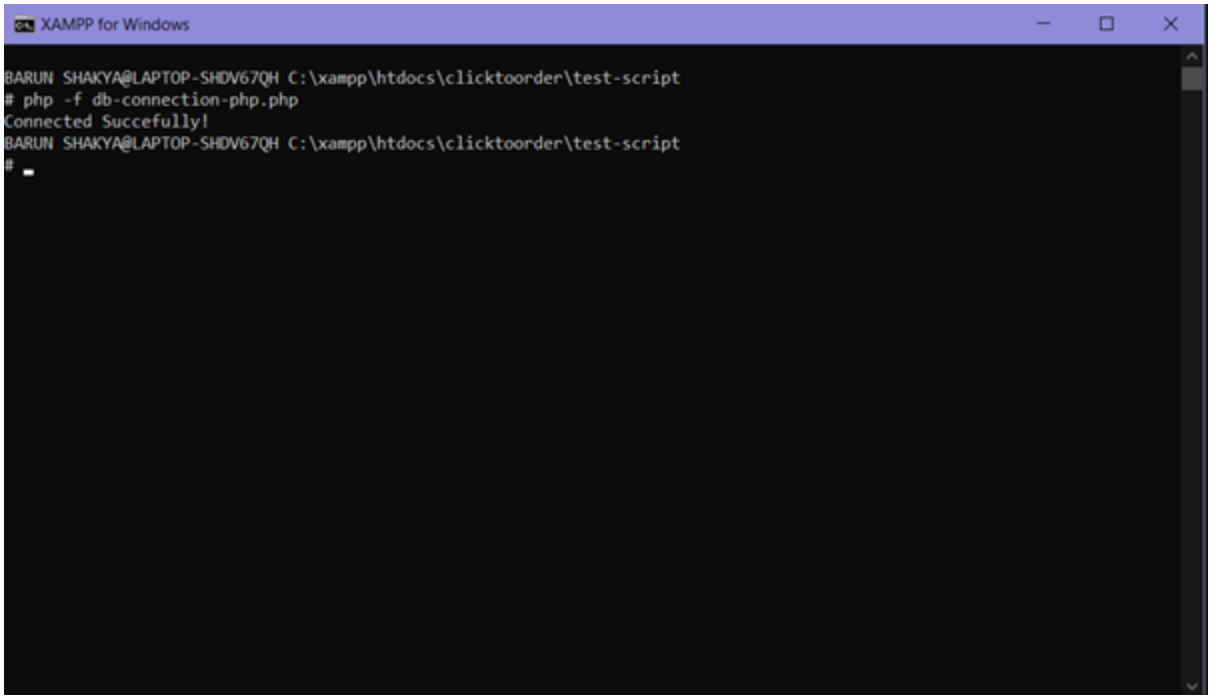
1. PHP - MySQL Connection



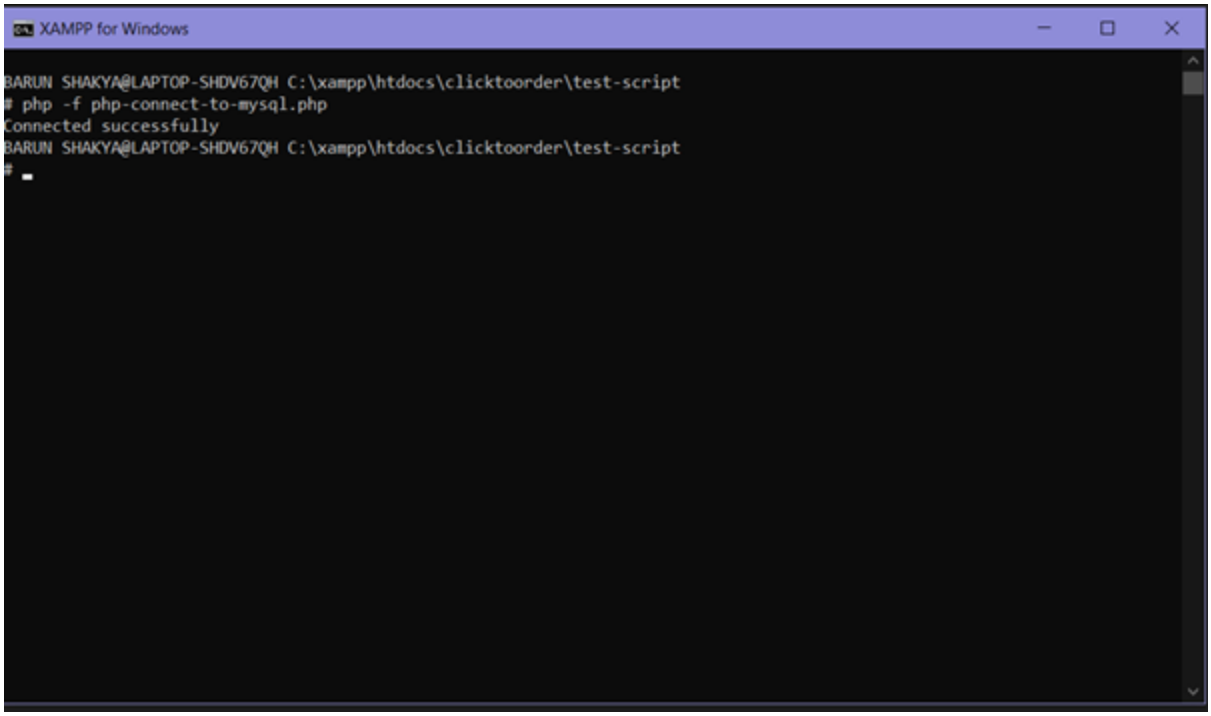
2. PHP MySQL pdo connection



3. Database connection in php



4. PHP Connection to Database

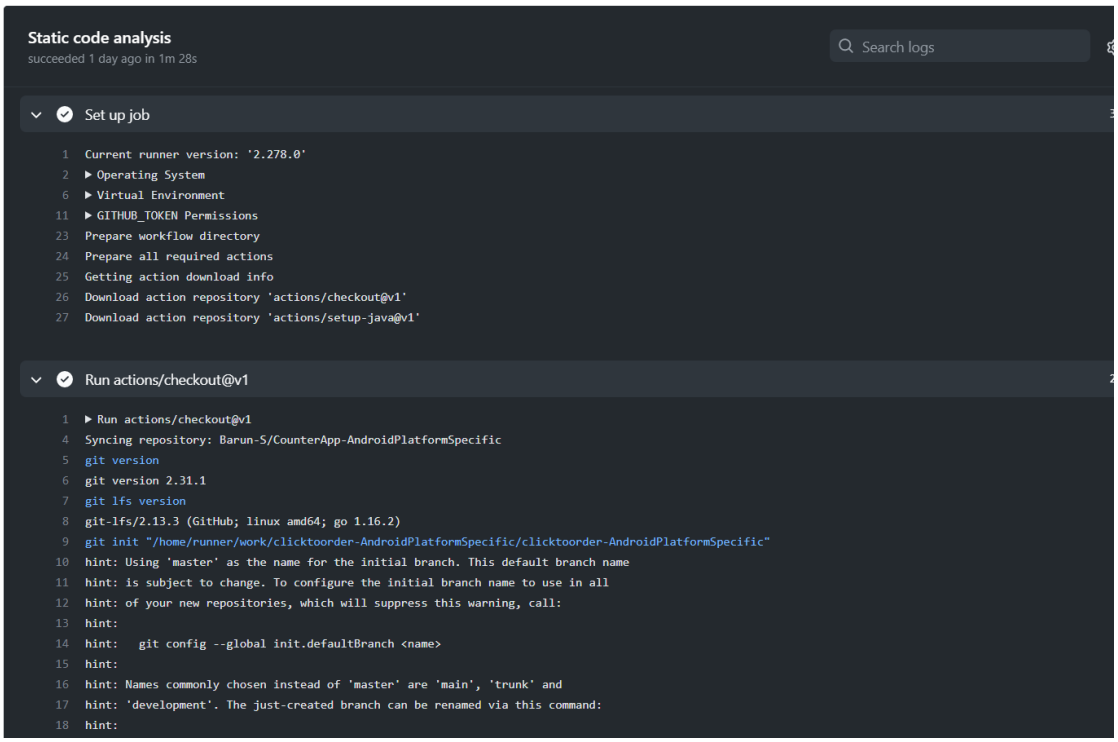


Automation Scripts

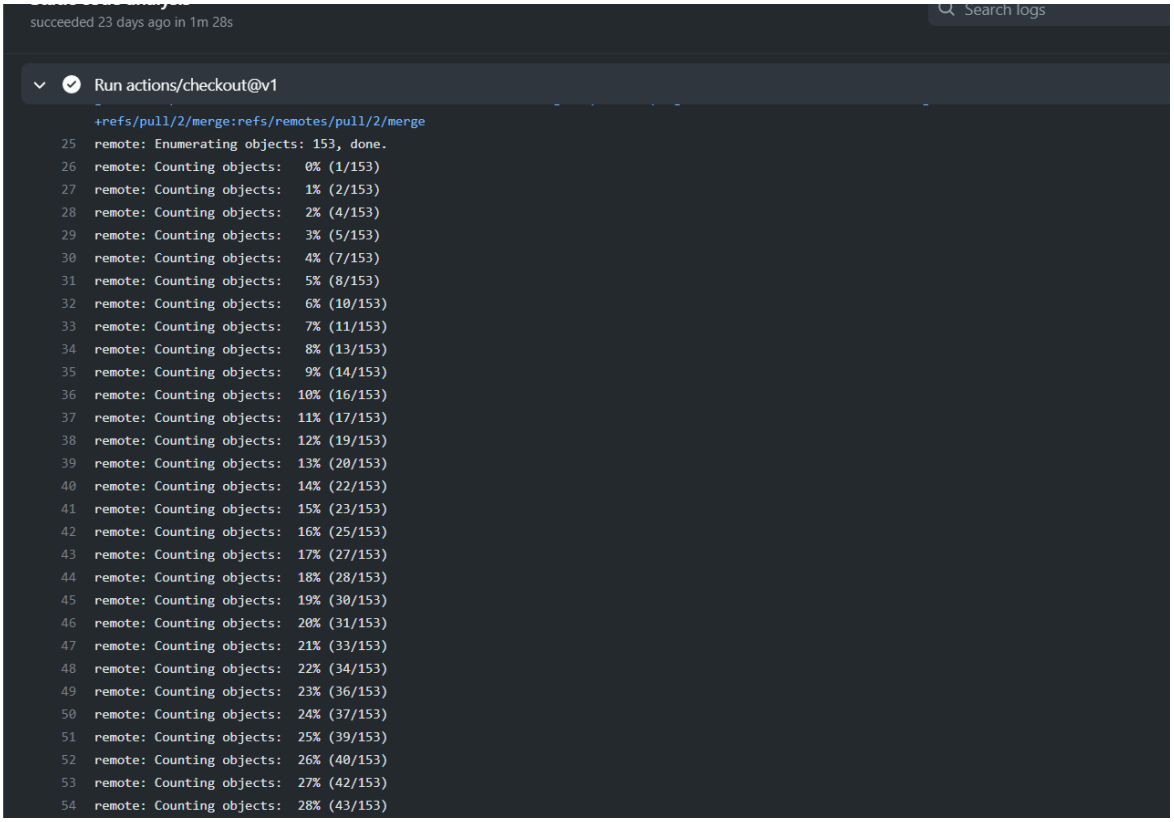
Screen Shots

1.

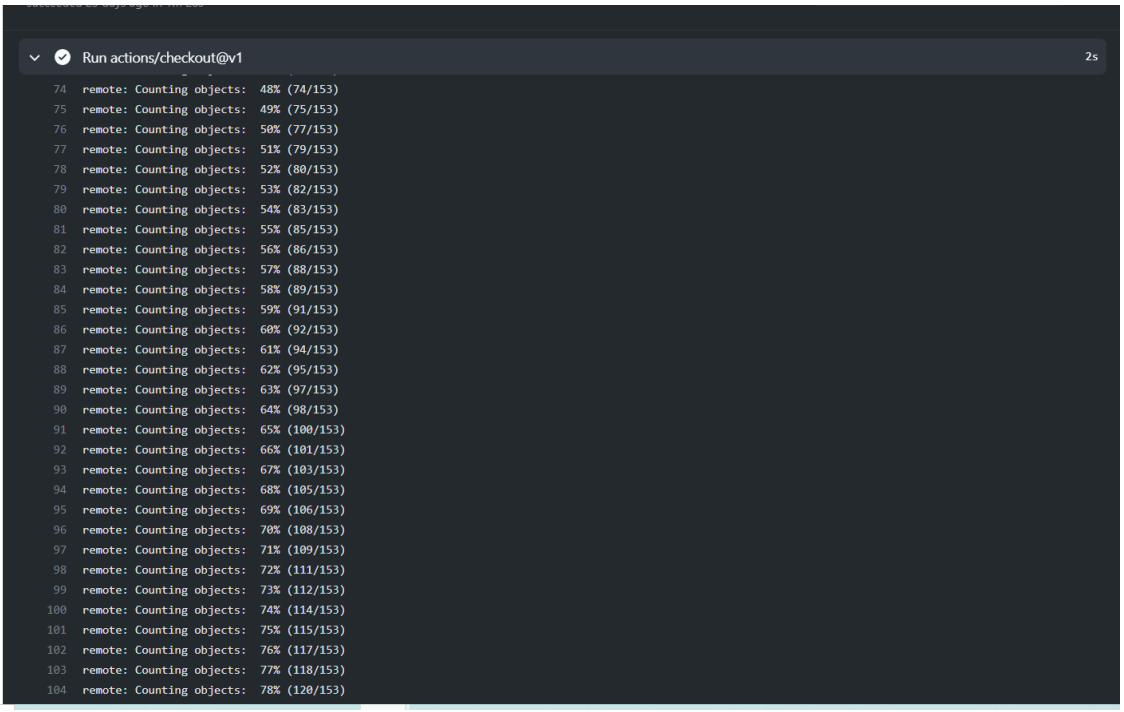
2.


Static code analysis
succeeded 1 day ago in 1m 28s
Search logs
Set up job
1 Current runner version: '2.278.0'
2 ▶ Operating System
6 ▶ Virtual Environment
11 ▶ GITHUB_TOKEN Permissions
23 Prepare workflow directory
24 Prepare all required actions
25 Getting action download info
26 Download action repository 'actions/checkout@v1'
27 Download action repository 'actions/setup-java@v1'
Run actions/checkout@v1
1 ▶ Run actions/checkout@v1
4 Syncing repository: Barun-S/CounterApp-AndroidPlatformSpecific
5 git version
6 git version 2.31.1
7 git lfs version
8 git-lfs/2.13.3 (GitHub; linux amd64; go 1.16.2)
9 git init "/home/runner/work/clicktoorder-AndroidPlatformSpecific/clicktoorder-AndroidPlatformSpecific"
10 hint: Using 'master' as the name for the initial branch. This default branch name
11 hint: is subject to change. To configure the initial branch name to use in all
12 hint: of your new repositories, which will suppress this warning, call:
13 hint:
14 hint: git config --global init.defaultBranch <name>
15 hint:
16 hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
17 hint: 'development'. The just-created branch can be renamed via this command:
18 hint:

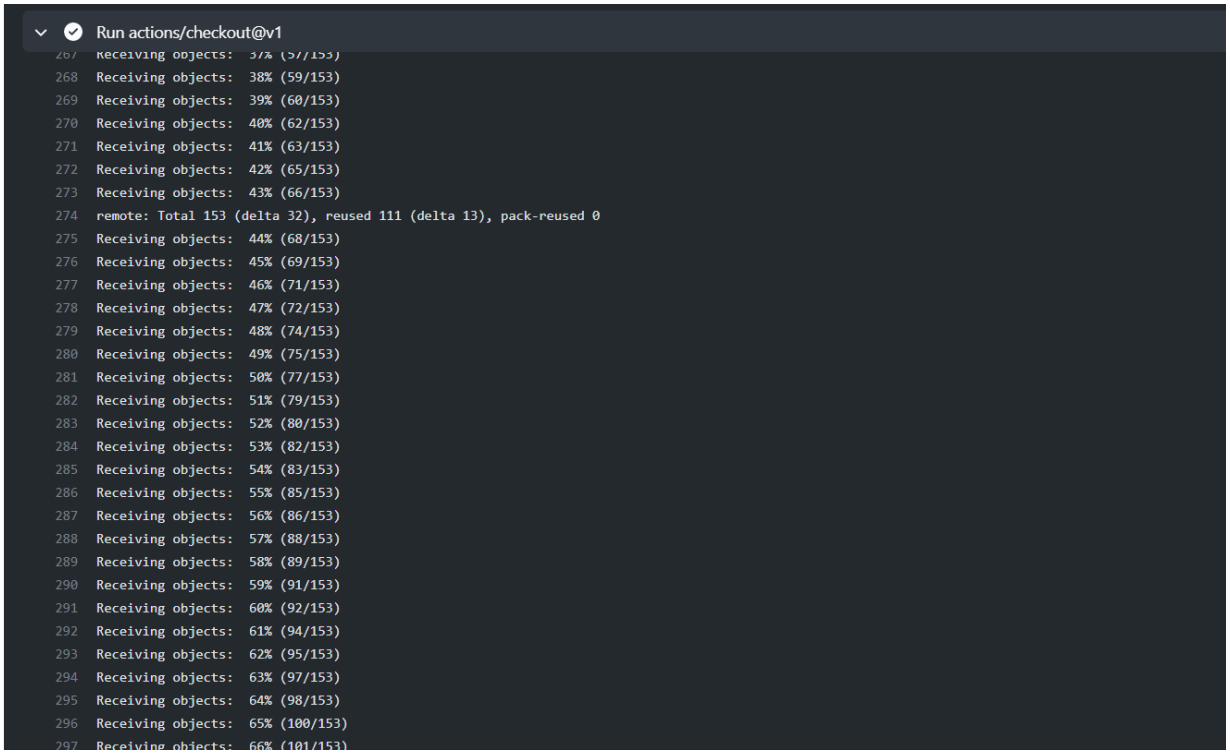
3.


succeeded 23 days ago in 1m 28s
Search logs
Run actions/checkout@v1
+refs/pull/2/merge:refs/remotes/pull/2/merge
remote: Enumerating objects: 153, done.
remote: Counting objects: 0% (1/153)
remote: Counting objects: 1% (2/153)
remote: Counting objects: 2% (4/153)
remote: Counting objects: 3% (5/153)
remote: Counting objects: 4% (7/153)
remote: Counting objects: 5% (8/153)
remote: Counting objects: 6% (10/153)
remote: Counting objects: 7% (11/153)
remote: Counting objects: 8% (13/153)
remote: Counting objects: 9% (14/153)
remote: Counting objects: 10% (16/153)
remote: Counting objects: 11% (17/153)
remote: Counting objects: 12% (19/153)
remote: Counting objects: 13% (20/153)
remote: Counting objects: 14% (22/153)
remote: Counting objects: 15% (23/153)
remote: Counting objects: 16% (25/153)
remote: Counting objects: 17% (27/153)
remote: Counting objects: 18% (28/153)
remote: Counting objects: 19% (30/153)
remote: Counting objects: 20% (31/153)
remote: Counting objects: 21% (33/153)
remote: Counting objects: 22% (34/153)
remote: Counting objects: 23% (36/153)
remote: Counting objects: 24% (37/153)
remote: Counting objects: 25% (39/153)
remote: Counting objects: 26% (40/153)
remote: Counting objects: 27% (42/153)
remote: Counting objects: 28% (43/153)

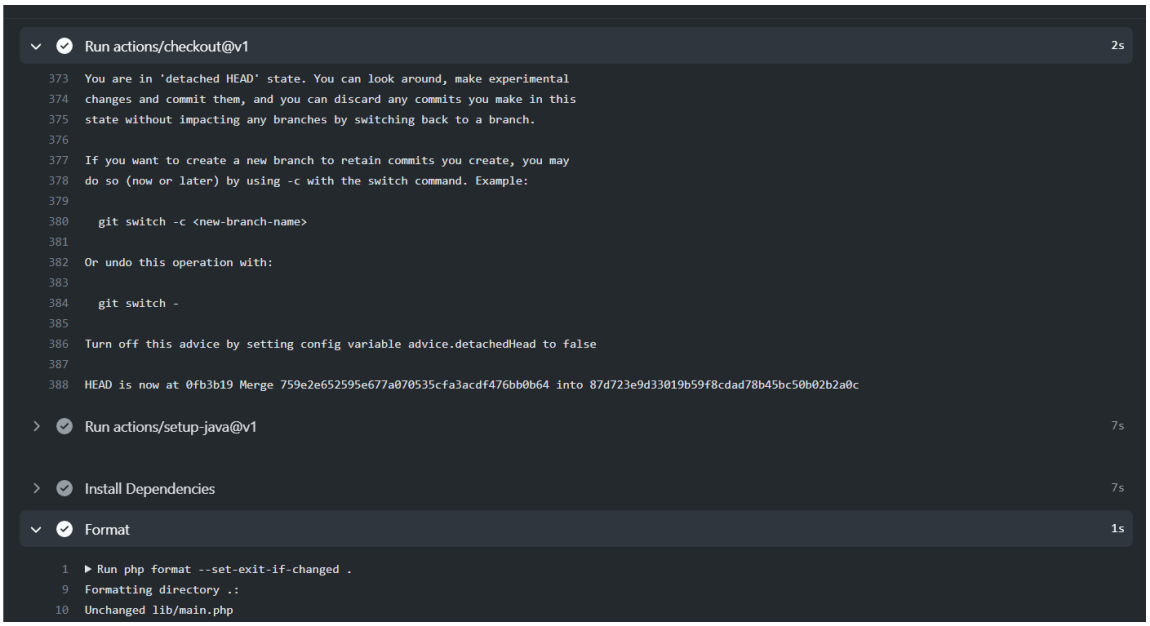
4.


Run actions/checkout@v1
2s
remote: Counting objects: 48% (74/153)
remote: Counting objects: 49% (75/153)
remote: Counting objects: 50% (77/153)
remote: Counting objects: 51% (79/153)
remote: Counting objects: 52% (80/153)
remote: Counting objects: 53% (82/153)
remote: Counting objects: 54% (83/153)
remote: Counting objects: 55% (85/153)
remote: Counting objects: 56% (86/153)
remote: Counting objects: 57% (88/153)
remote: Counting objects: 58% (89/153)
remote: Counting objects: 59% (91/153)
remote: Counting objects: 60% (92/153)
remote: Counting objects: 61% (94/153)
remote: Counting objects: 62% (95/153)
remote: Counting objects: 63% (97/153)
remote: Counting objects: 64% (98/153)
remote: Counting objects: 65% (100/153)
remote: Counting objects: 66% (101/153)
remote: Counting objects: 67% (103/153)
remote: Counting objects: 68% (105/153)
remote: Counting objects: 69% (106/153)
remote: Counting objects: 70% (108/153)
remote: Counting objects: 71% (109/153)
remote: Counting objects: 72% (111/153)
remote: Counting objects: 73% (112/153)
remote: Counting objects: 74% (114/153)
remote: Counting objects: 75% (115/153)
remote: Counting objects: 76% (117/153)
remote: Counting objects: 77% (118/153)
remote: Counting objects: 78% (120/153)

5.


Run actions/checkout@v1
Receiving objects: 37% (57/153)
Receiving objects: 38% (59/153)
Receiving objects: 39% (60/153)
Receiving objects: 40% (62/153)
Receiving objects: 41% (63/153)
Receiving objects: 42% (65/153)
Receiving objects: 43% (66/153)
remote: Total 153 (delta 32), reused 111 (delta 13), pack-reused 0
Receiving objects: 44% (68/153)
Receiving objects: 45% (69/153)
Receiving objects: 46% (71/153)
Receiving objects: 47% (72/153)
Receiving objects: 48% (74/153)
Receiving objects: 49% (75/153)
Receiving objects: 50% (77/153)
Receiving objects: 51% (79/153)
Receiving objects: 52% (80/153)
Receiving objects: 53% (82/153)
Receiving objects: 54% (83/153)
Receiving objects: 55% (85/153)
Receiving objects: 56% (86/153)
Receiving objects: 57% (88/153)
Receiving objects: 58% (89/153)
Receiving objects: 59% (91/153)
Receiving objects: 60% (92/153)
Receiving objects: 61% (94/153)
Receiving objects: 62% (95/153)
Receiving objects: 63% (97/153)
Receiving objects: 64% (98/153)
Receiving objects: 65% (100/153)
Receiving objects: 66% (101/153)

6.


Run actions/checkout@v1
2s
You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.
If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:
git switch -c <new-branch-name>
Or undo this operation with:
git switch -
Turn off this advice by setting config variable advice.detachedHead to false
HEAD is now at 0fb3b19 Merge 759e2e652595e677a070535cfa3acdf476bb0b64 into 87d723e9d33019b59f8cdad78b45bc50b02b2a0c
Run actions/setup-java@v1
7s
Install Dependencies
7s
Format
1s
Run php format --set-exit-if-changed .
Formatting directory .:
Unchanged lib/main.php

7.


10 Unchanged lib/main.php
11 Unchanged test/widget_test.php
Analyze
17s
Post Run actions/setup-java@v1
0s
1 Post job cleanup.
Complete job
0s
1 Cleaning up orphan processes