# Weekly Report: Week 2 (Ending 27th September 2024) - Continuous Integration Setup with GitHub

## Objectives

This week, the focus was on embedding Continuous Integration (CI) practices into our development workflow using GitHub. The primary objectives were to configure GitHub Actions, implement automated build processes, and integrate automated testing alongside code quality checks.

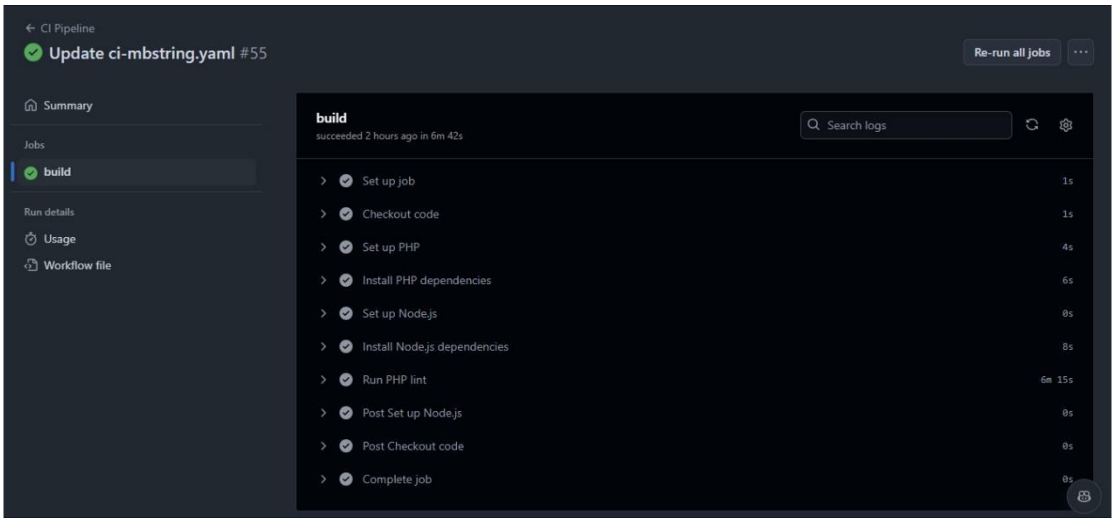
## CI Tool Configuration

GitHub Actions was selected as the CI tool for its seamless integration with our repository, allowing us to automate workflows directly within our version control system.

## Configuration Steps

- **Job setup completed**: The CI pipeline job has been defined and integrated into GitHub Actions.
- **Checkout code setup done**: The repository code is checked out automatically, ready for testing and building.
- **PHP environment configured**: The specified PHP version is in place for executing scripts and running tests.
- **PHP dependencies installed**: Composer was used to install all required PHP dependencies.
- **Node.js environment configured**: Node.js has been set up with the necessary version for JavaScript-related tasks.
- **Node.js dependencies installed**: All required Node.js packages were installed via npm/yarn.
- **PHP lint executed**: PHP code was successfully linted, ensuring quality and consistency.
- **Post Node.js setup completed**: Additional configuration tasks after Node.js installation were handled.

- **Post-checkout code tasks done**: Necessary post-checkout operations, such as configuring environment variables, were executed.
- **Job completed**: The CI pipeline successfully ran all tasks, ensuring the stability of the codebase.



```
build
succeeded 2 hours ago in 6m 42s

Set up job                      1s
Checkout code                   1s
Set up PHP                      4s
Install PHP dependencies        6s
Set up Node.js                  0s
Install Node.js dependencies    8s
Run PHP lint                6m 15s
Post Set up Node.js             0s
Post Checkout code              0s
Complete job                    0s
```

## Challenges Encountered

- **Dependency conflicts**: Issues arose between different package versions.
- **Poor communication during commits**: This led to uncoordinated changes and documentation.
- **Environment differences**: Discrepancies between development and CI environments.
- **Merge conflicts**: Some issues emerged during branch merges.
- **Configuration issues**: Initial difficulties configuring certain components.
- **Lack of communication**: Hindered smooth collaboration at times.

**TEAM COORDINATION**

- **Byansi Anthony** added PHPUnit with a phpunit.xml file to organize unit and integration tests. He also integrated Laravel Dusk for browser-based testing.
- **Joseph Sendi** and **Devote Boniface** primarily worked on creating the CI/CD environment, installing software such as PHP and its dependencies, configuring PHP lint, and performing code refactoring.

## Configuration File

The CI workflow is managed using the ci.yml configuration file, which defines the pipeline structure.

## Automated Testing and Code Quality

### Testing Frameworks

- **PHPUnit** was integrated for unit testing.
- **Laravel Dusk** was added for integration testing, ensuring robust application performance.

### Code Quality Integration

- **PHP Lint** was implemented to enforce coding standards, promoting consistency across the project.

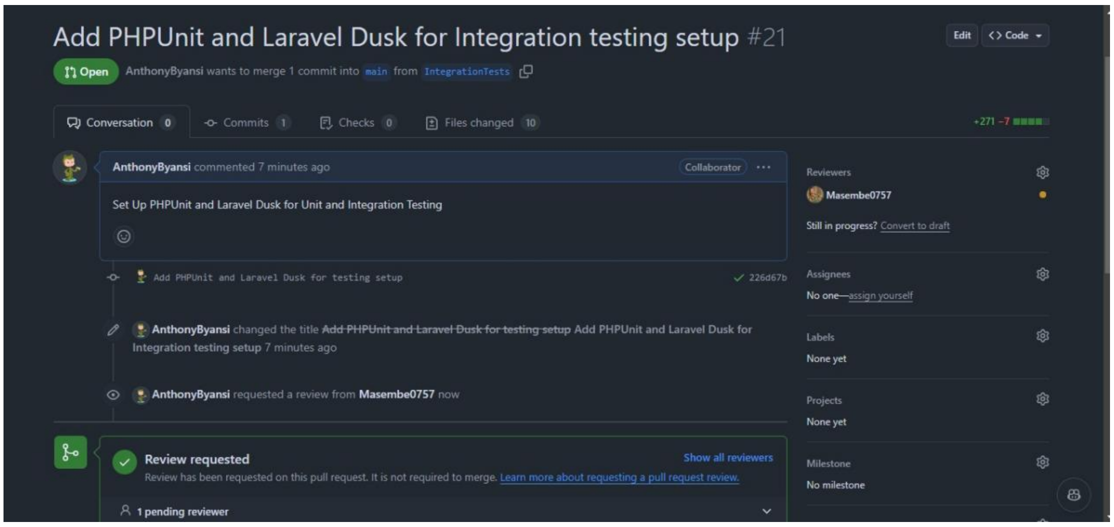
## Code Integration

As part of the CI/CD setup, code integration played a crucial role in ensuring that changes from multiple contributors were merged smoothly and tested automatically. The setup enforced a consistent integration process where every new code push or pull request triggered the CI pipeline. This allowed us to:

- Automatically **run tests** on the integrated code to catch any issues early.
- **Resolve merge conflicts** before they could impact the main branch, ensuring that only stable and tested code was merged.

- **Maintain code quality** through automated linting and testing, providing immediate feedback to the team when integration issues arose.
- **Reduce manual intervention** by automating the build and test processes, speeding up the integration of new features or bug fixes.

By integrating these steps into the CI/CD pipeline, we were able to continuously deliver updates without interrupting the development flow, while maintaining stability and code integrity.



Add PHPUnit and Laravel Dusk for Integration testing setup #21

Open AnthonyByansi wants to merge 1 commit into `main` from `IntegrationTests`

Conversation 0 · Commits 1 · Checks 0 · Files changed 10 · +271 −7

AnthonyByansi commented 7 minutes ago · Collaborator

Set Up PHPUnit and Laravel Dusk for Unit and Integration Testing

`Add PHPUnit and Laravel Dusk for testing setup` ✓ 226d67b

AnthonyByansi changed the title ~~Add PHPUnit and Laravel Dusk for testing setup~~ Add PHPUnit and Laravel Dusk for Integration testing setup 7 minutes ago

AnthonyByansi requested a review from Masembe0757 now

Review requested · Show all reviewers
Review has been requested on this pull request. It is not required to merge. Learn more about requesting a pull request review.

1 pending reviewer

Reviewers: Masembe0757
Still in progress? Convert to draft

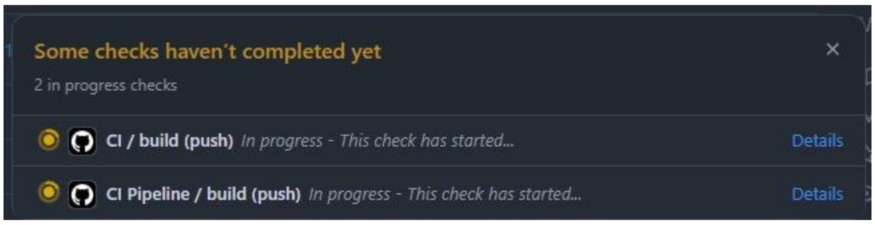
Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

Milestone: No milestone

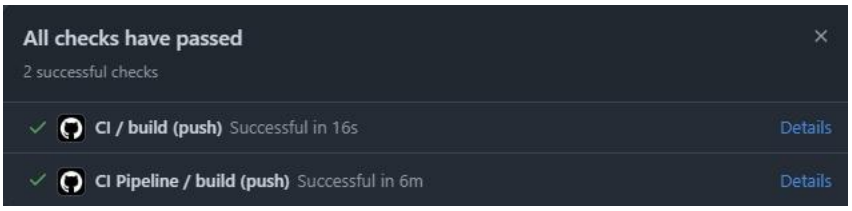
## BUILD WORKFLOW



Some checks haven't completed yet
2 in progress checks

CI / build (push) *In progress - This check has started...* Details

CI Pipeline / build (push) *In progress - This check has started...* Details



All checks have passed
2 successful checks

✓ CI / build (push) Successful in 16s Details

✓ CI Pipeline / build (push) Successful in 6m Details

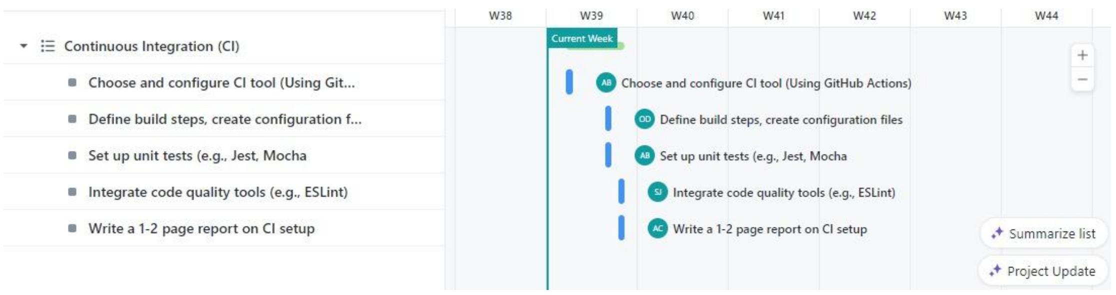
## ClickUp Setup

### Task Management

**Tasks were organized in ClickUp for streamlined workflow management:**

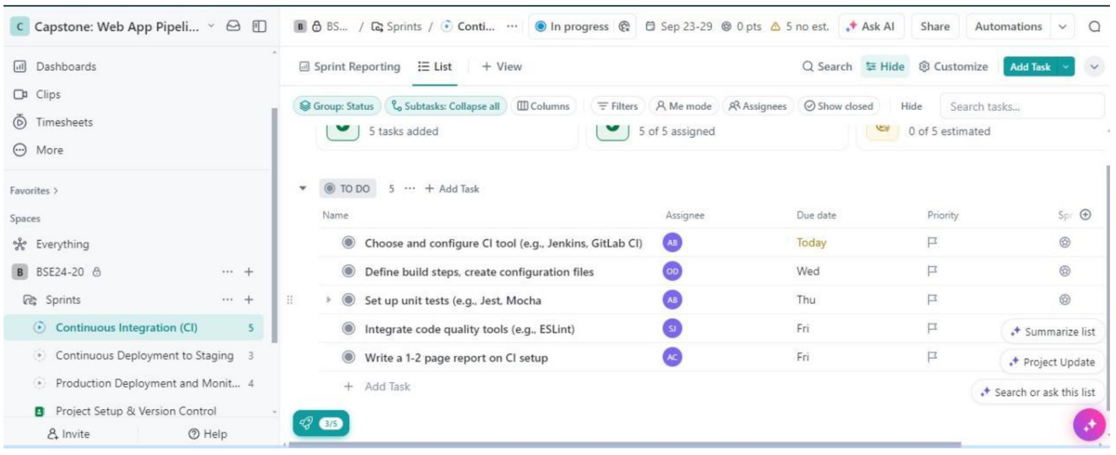
- **Definining build steps & Creating configuration files**: Assigned to **Devote Boniface Ochieng**.
- **Setting up Unit and Integration Tests**: Assigned to **Byansi Anthony**.
- **Setting Up Code Quality Checks & Integrating tools**: Assigned to **Sendi Joseph**.
- **Week 2 Report Writing**: Assigned to **Aparo Cecilia**.

Each task was documented with deadlines and progress tracking, ensuring accountability and transparency.

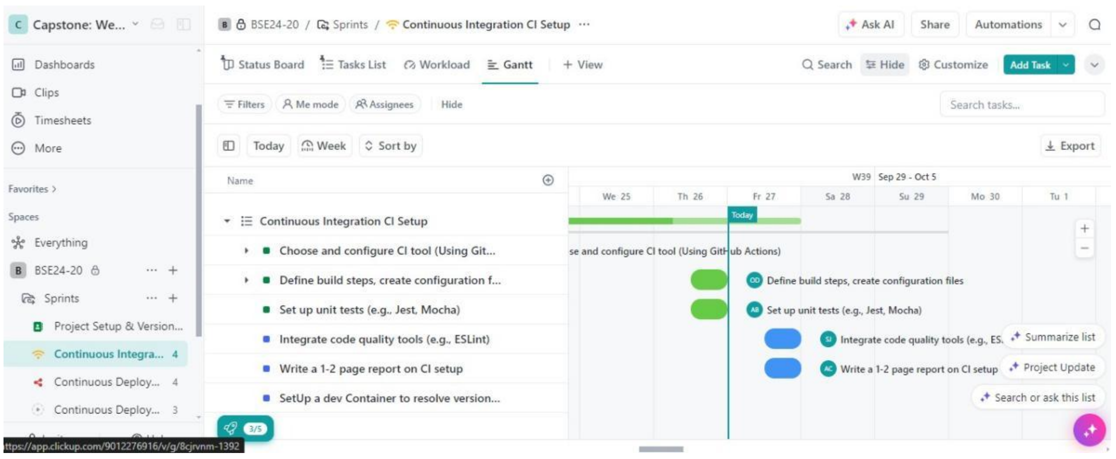


Continuous Integration (CI)
- Choose and configure CI tool (Using Git...
- Define build steps, create configuration f...
- Set up unit tests (e.g., Jest, Mocha
- Integrate code quality tools (e.g., ESLint)
- Write a 1-2 page report on CI setup

AB Choose and configure CI tool (Using GitHub Actions)
OD Define build steps, create configuration files
AB Set up unit tests (e.g., Jest, Mocha
SJ Integrate code quality tools (e.g., ESLint)
AC Write a 1-2 page report on CI setup



| Name | Assignee | Due date | Priority |
|---|---|---|---|
| Choose and configure CI tool (e.g., Jenkins, GitLab CI) | AB | Today | |
| Define build steps, create configuration files | OD | Wed | |
| Set up unit tests (e.g., Jest, Mocha | AB | Thu | |
| Integrate code quality tools (e.g., ESLint) | SJ | Fri | |
| Write a 1-2 page report on CI setup | AC | Fri | |



Continuous Integration CI Setup
- Choose and configure CI tool (Using Git...
- Define build steps, create configuration f...
- Set up unit tests (e.g., Jest, Mocha)
- Integrate code quality tools (e.g., ESLint)
- Write a 1-2 page report on CI setup
- SetUp a dev Container to resolve version...

se and configure CI tool (Using GitHub Actions)
OD Define build steps, create configuration files
AB Set up unit tests (e.g., Jest, Mocha)
SI Integrate code quality tools (e.g., ES.
AC Write a 1-2 page report on CI setup

## Deliverables

- A fully configured CI pipeline with GitHub Actions, running automated builds and tests on code push events.
- Automated testing and code quality checks integrated into the CI workflow.
- Updated ClickUp tasks reflecting the progress of Week 2 activities.

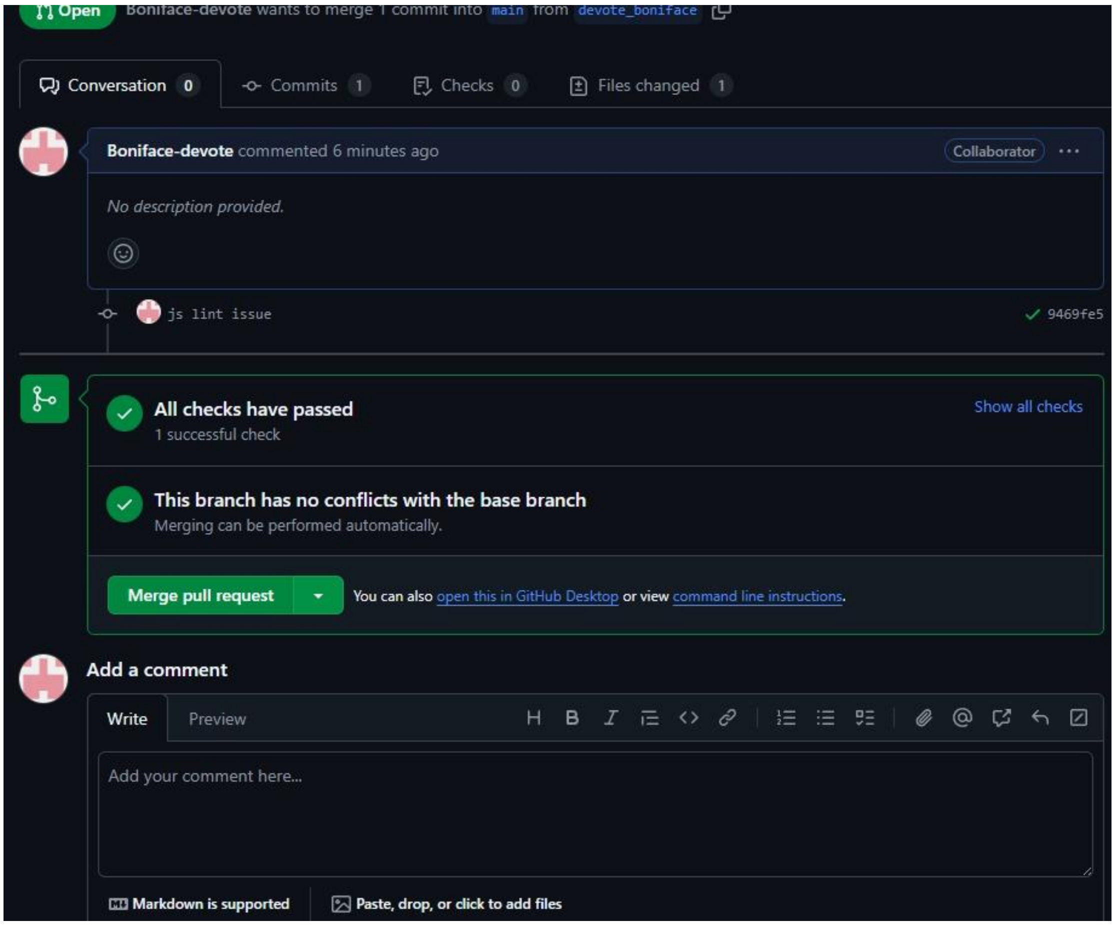
## Outcomes

The setup of CI using GitHub Actions has significantly improved our development process, yielding the following benefits:

- **Enhanced code quality** through automated linting and testing.
- **Increased efficiency** with immediate build notifications and error reporting.
- **Improved team collaboration** through structured task management in ClickUp.

## Conclusion

The successful implementation of the CI pipeline with GitHub Actions marks a significant milestone in our development practices. Moving forward, we will continue refining our CI/CD processes and exploring further automation opportunities to enhance productivity and streamline our workflow.



Open Boniface-devote wants to merge 1 commit into main from devote_boniface

Boniface-devote commented 6 minutes ago Collaborator

*No description provided.*

js lint issue ✓ 9469fe5

All checks have passed
1 successful check

This branch has no conflicts with the base branch
Merging can be performed automatically.

You can also open this in GitHub Desktop or view command line instructions.

Add a comment