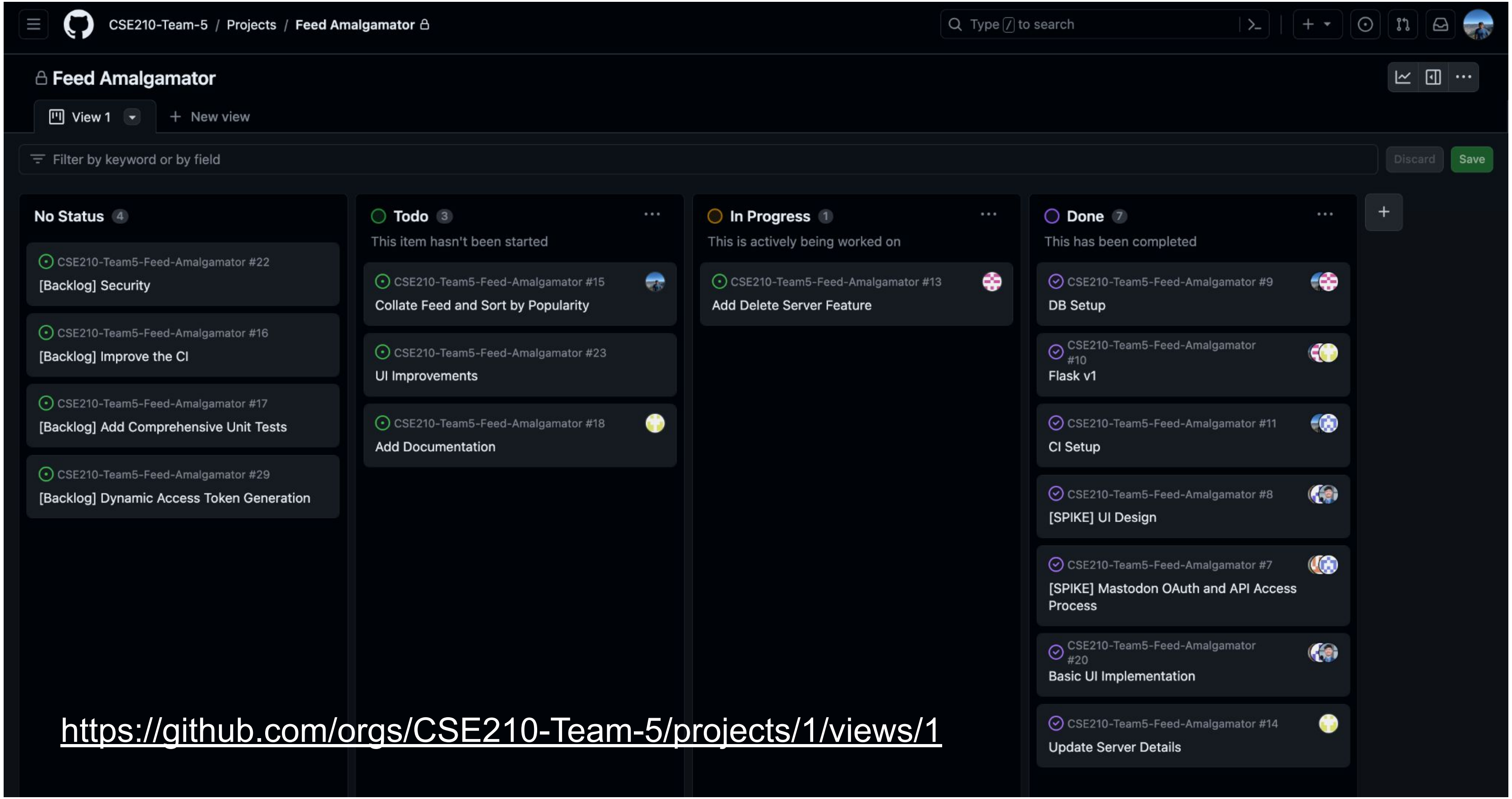
Team 5

Feed Amalgamator

Kanban Board



CSE210-Team-5 / Projects / Feed Amalgamator
Type / to search
Feed Amalgamator
View 1
New view
Filter by keyword or by field
Discard
Save

No Status 4

- CSE210-Team5-Feed-Amalgamator #22
  [Backlog] Security
- CSE210-Team5-Feed-Amalgamator #16
  [Backlog] Improve the CI
- CSE210-Team5-Feed-Amalgamator #17
  [Backlog] Add Comprehensive Unit Tests
- CSE210-Team5-Feed-Amalgamator #29
  [Backlog] Dynamic Access Token Generation

Todo 3

This item hasn't been started

- CSE210-Team5-Feed-Amalgamator #15
  Collate Feed and Sort by Popularity
- CSE210-Team5-Feed-Amalgamator #23
  UI Improvements
- CSE210-Team5-Feed-Amalgamator #18
  Add Documentation

In Progress 1

This is actively being worked on

- CSE210-Team5-Feed-Amalgamator #13
  Add Delete Server Feature

Done 7

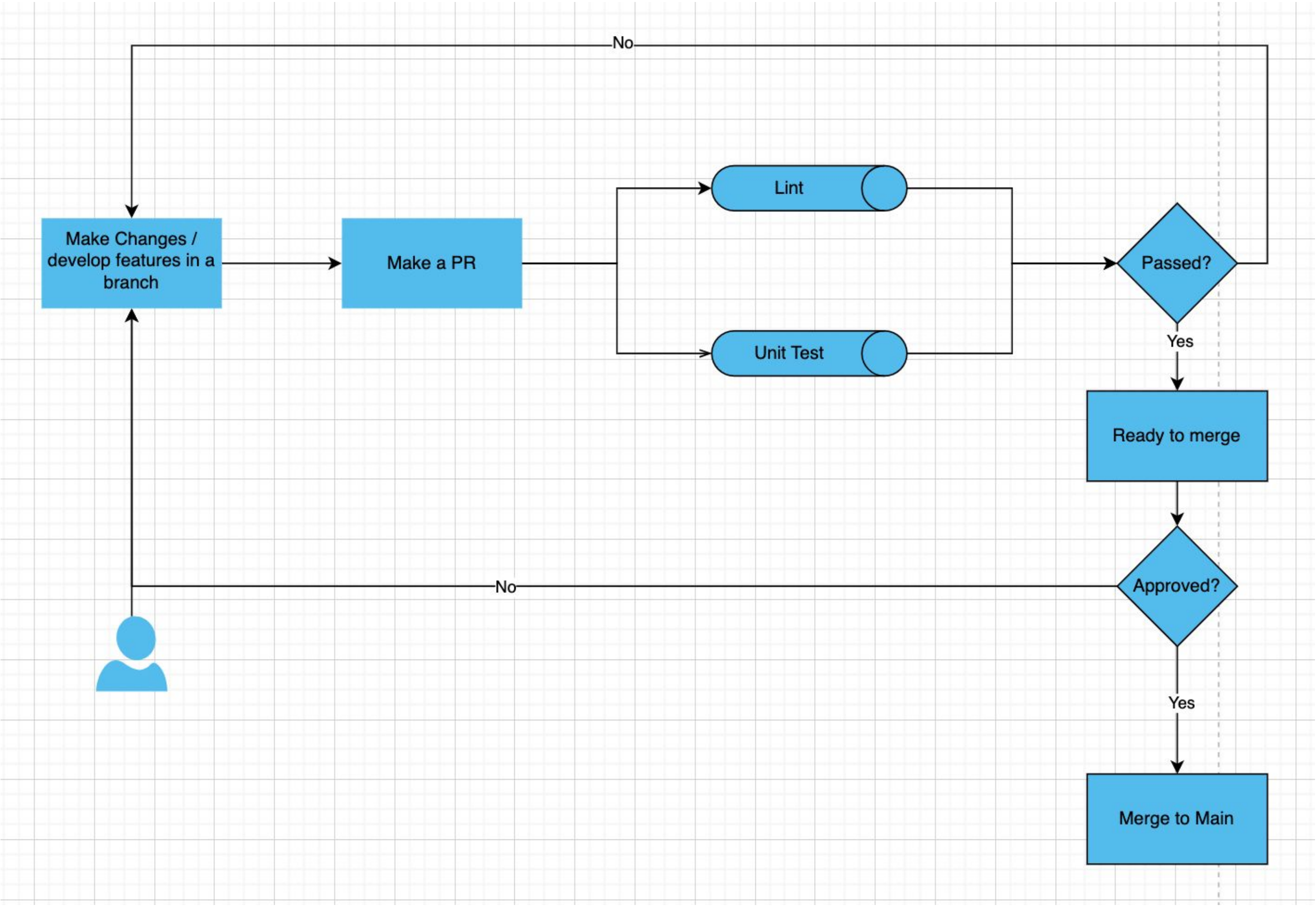
This has been completed

- CSE210-Team5-Feed-Amalgamator #9
  DB Setup
- CSE210-Team5-Feed-Amalgamator #10
  Flask v1
- CSE210-Team5-Feed-Amalgamator #11
  CI Setup
- CSE210-Team5-Feed-Amalgamator #8
  [SPIKE] UI Design
- CSE210-Team5-Feed-Amalgamator #7
  [SPIKE] Mastodon OAuth and API Access Process
- CSE210-Team5-Feed-Amalgamator #20
  Basic UI Implementation
- CSE210-Team5-Feed-Amalgamator #14
  Update Server Details

https://github.com/orgs/CSE210-Team-5/projects/1/views/1

CI

Linting is in place, WIP to integrate python unit tests to the pipeline.



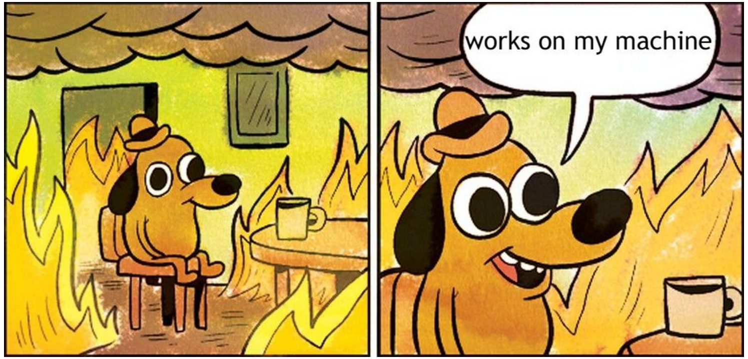
PDM

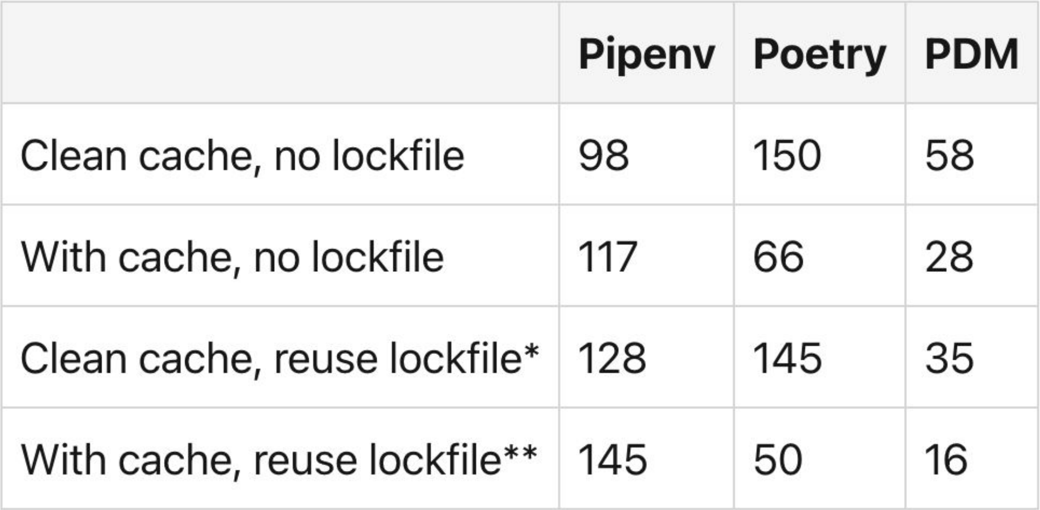
What is it?

- Python package and dependency manager

Why is it better?

- Better Performance
- Better dependency resolution.





	Pipenv	Poetry	PDM
Clean cache, no lockfile	98	150	58
With cache, no lockfile	117	66	28
Clean cache, reuse lockfile*	128	145	35
With cache, reuse lockfile**	145	50	16

[Source: https://dev.to/frostming/a-review-pipenv-vs-poetry-vs-pdm-39b4]

Flask - The Software Engineering Aspect

Flask offers several advantages that align with Software Engineering principles

- **Separation of Concerns** : Flask uses the Jinja2 templating engine, promoting the separation of business logic and presentation.
- **Modularity** : Developers can choose and integrate different components, libraries, and extensions based on their project requirements.
- **Widely Compatible** : Flask can be integrated with various databases, front-end frameworks, and other tools
- **Testing** : Flask provides built-in support for testing, allowing developers to write unit tests for their applications more easily

Demo

What’s going well

- Project was well scoped. Each feature is “bite sized”, allowing for work to be done incrementally. Goal for a viable product seems achievable

- Project members are involved and are able to provide feedback and discuss issues freely; psychological safety

- Everyone seems to be learning quite a lot!

Issues

- Members are generally inexperienced with using CI and adhering to it (merging multiple times per day), leading to some merge issues.

- Reliance on third party API that can be janky and have unclear conditions

- Code quality and style issues and differences - difficult to make code look like “one person wrote everything”

Questions?