HoppyBrew Project Summary

Introduction

HoppyBrew is an open-source web application designed to assist beer brewing enthusiasts in managing their brewing recipes and brew logs. The project aims to provide a user-friendly and intuitive interface, facilitating the storage, retrieval, and management of brewing recipes and batches. This project was developed as part of a software engineering course, emphasizing practical implementation and real-world application.

Achievements

- **Database and API Implementation:** Successfully set up a PostgreSQL database and developed backend APIs using Python and FastAPI, facilitating CRUD operations for hops, fermentables, miscellaneous items, yeast, recipes, and batch instantiations.
- **Containerization:** Implemented Docker and Docker Compose to ensure seamless deployment and integration of backend, frontend, and database components.
- **Web Scraping:** Developed functionalities to import recipes from the Brewer Association and other sources using web scraping techniques.
- **BeerXML Support:** Implemented import and export functionalities for BeerXML files, enabling compatibility with other brewing software like Brewfather.
- **CI/CD Pipeline:** Established a CI/CD pipeline using GitHub Actions, automating testing, linting, and deployment processes.
- **Front-End Development:** Built a responsive front-end using Vue.js and Nuxt.js, allowing users to interact with the application seamlessly.

Challenges and Learnings

- **Version Control and Dependencies:** Faced issues with version control and dependencies due to inconsistent updates and tutorial discrepancies. Learned the importance of using database migration tools and maintaining consistent environments.
- **iSpindle Integration:** Decided not to implement iSpindle due to testing complexities and prioritization of other features.
- **Technical Skills:** Gained experience in using Vue.js, Nuxt.js, and SCRUM methodology. Developed comprehensive documentation using markdown and automated PDF conversion with Pandoc.
- **GitHub vs. GitLab:** Transitioned to GitLab for CI/CD but reverted to GitHub as per assignment requirements, realizing GitHub's ease of use and stronger CI/CD capabilities.
- **Docker and Docker Compose:** Learned that Docker Compose is not fully compatible with the Unraid system, requiring adjustments and workarounds.
- **Makefile Usage:** Utilized Makefile for automating processes, enhancing efficiency and consistency in the development workflow.

Technical Overview

- **Frontend:** Developed using Vue.js and Nuxt.js for a dynamic and responsive user interface.
- **Backend:** Built with Python and FastAPI, ensuring efficient and scalable API services.
- **Database:** Utilized PostgreSQL for robust data storage and management.

- **CI/CD:** Implemented CI/CD pipeline using GitHub Actions, ensuring automated testing and deployment.
- **Containerization:** Docker and Docker Compose used for containerization, facilitating smooth deployment and integration.

Conclusion

HoppyBrew is an operational web application that successfully meets the primary goals of managing brewing recipes and logs. While some features like iSpindle integration and advanced inventory tracking were left out due to time constraints, the project demonstrates a solid proof of concept with fully functional core features.

Future Steps

- **Complete iSpindle Integration:** Implement and test iSpindle data collection and integration.
- **Enhance Inventory Management:** Finalize the functionality to track inventory consumption.
- **Refine and Optimize:** Continuously improve the application based on user feedback and additional testing.
- **Expand Feature Set:** Introduce new features such as equipment profiles, water profiles, and mash profiles.

Lessons Learned

- **SCRUM Methodology:** The SCRUM approach provided dynamic agility, making it a valuable methodology for managing software development projects.
- **Markdown Documentation:** Using markdown for documentation and converting it to PDF using Pandoc proved to be effective, though challenging at times due to path and reference issues.
- **CI/CD with GitHub Actions:** GitHub Actions provided a robust platform for implementing CI/CD pipelines, streamlining the testing and deployment processes.
- **Docker and Docker Compose:** Gained significant experience with Docker and Docker Compose, despite compatibility challenges with the Unraid system.

Final Note

The project, including all code, documentation, and necessary files, is hosted on GitHub and is publicly accessible. The application is also cloud-hosted for direct access, fulfilling the project requirements. The detailed documentation provides comprehensive insights into the development process, technical decisions, and implementation procedures.

GitHub Repository:

- https://github.com/asbor/iu-project-software-engineering

Cloud Hosted Application:

- public.ecr.aws/m2z9f5m0/iu-project-backend
- public.ecr.aws/m2z9f5m0/iu-project-db
- public.ecr.aws/m2z9f5m0/iu-project-frontend

Note: The cloud-hosted application is currently not operational due to complications. Please refer to the GitHub repository for the latest codebase and documentation.