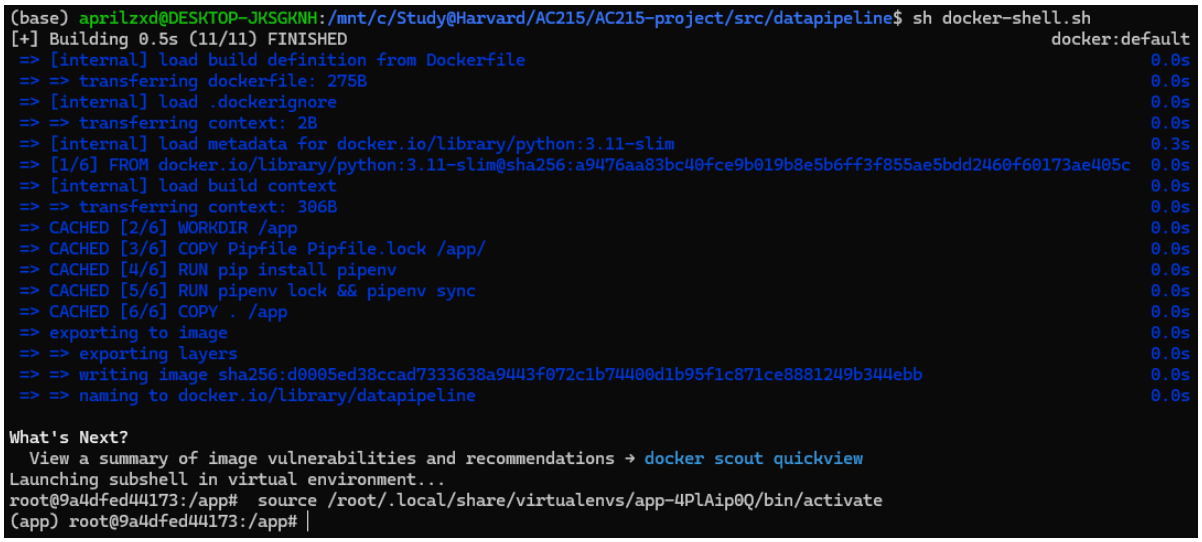
Milestone 2: Billionaire Secretary

Team Billionaires: Huandong Chang, Mingyuan Ma, Lance Lu, April Zhang

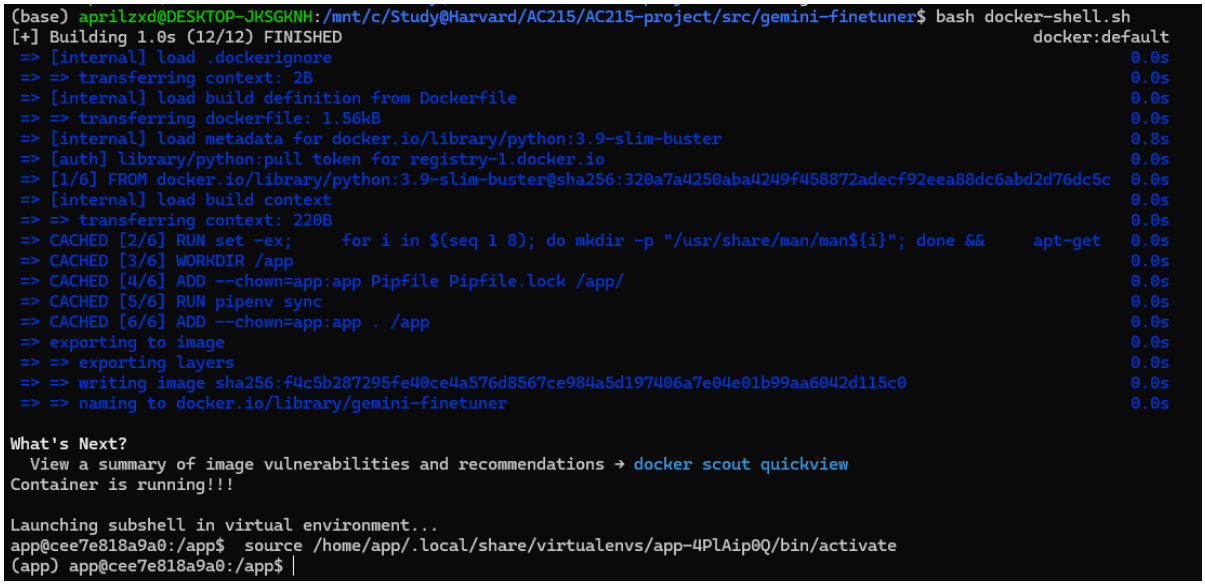
Virtual Environment Setup

Our virtual environment setup is included in the Dockerfile of each component. We used Pipenv with `Pipfile` and `Pipfile.lock` to manage dependencies.

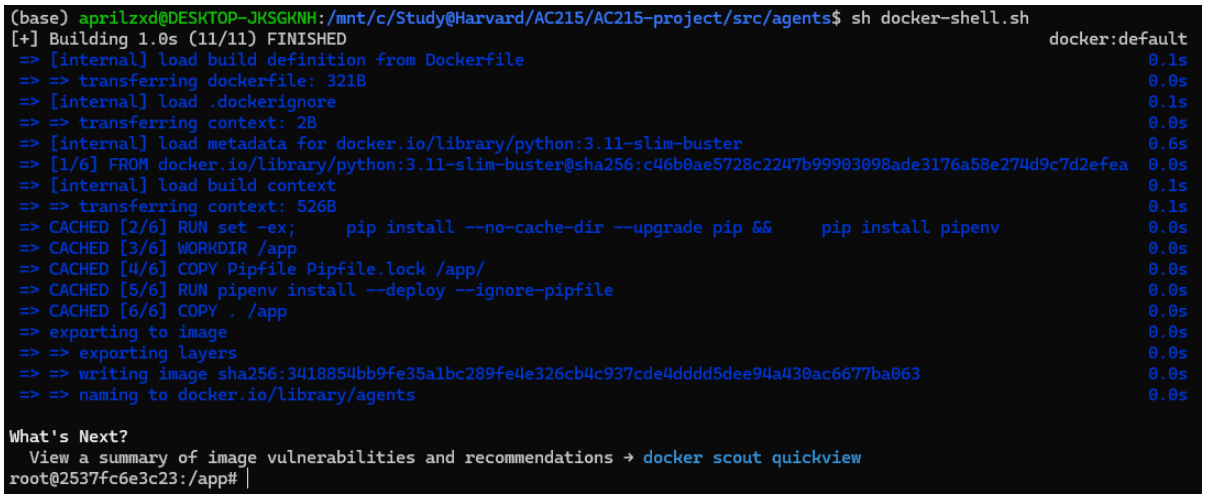
Data Pipeline



Gemini Finetuner



Agents



Containerized Components

As shown in the screenshots above, the data pipeline, Gemini fintuner, and agents components each run in their own container. The documentation and setup instructions for these containers are provided in the README file in our repository.

Versioned Data Strategy

Implement a data versioning strategy using tools like DVC or other suitable solutions. If feasible, this strategy should also be containerized to ensure portability and reproducibility of data processes.(Optional but recommended)
Please make sure to record your prompts as part of the data. For eg: If you are generating data using a LLM, please add the prompts and generated data as part of the dataset.
Deliverables:

- Documentation on the data versioning strategy chosen (e.g., DVC) and why
- A working containerized version of the data versioning pipeline (if applicable)
- Version control history showing tracked datasets, along with their respective versions, commits, and logs.

Teams Utilizing LLMs (Large Language Models)

Teams working with LLMs should implement a RAG setup. This setup should include data collection, chunking into appropriate sizes for processing, and the integration of a vector database. Teams should also fine-tune models and document the experimentation process.
Deliverables:

- A containerized RAG pipeline, including scripts for data chunking, vectorization, and integration with a vector database

Documentation of the fine-tuning process, including datasets used, hyperparameters, and models
Experiment logs showing model performance across different fine-tuning and RAG configurations.

Mock-up of the Application

A working prototype or mock-up of the final application that integrates all project components. Teams that have already submitted this in Milestone 1 should refine or extend their prototype based on feedback or new progress.
Deliverables:

An application mock-up or wireframe, including user interface elements and how the app will interact with back-end components

- **Use Branch 'milestone2'**: Commit your work for Milestone 2 to the branch named 'milestone2'.
- **Mock Submission in README**: For each milestone, we include a mock submission section in your README file.
- **Canvas Group**: Finally, please choose your group for Milestone 2 on Canvas, where grades will be posted.
- **Canvas Submission:** You will be submitting your Github commit hash for grading. (See below on how to find commit hash)