

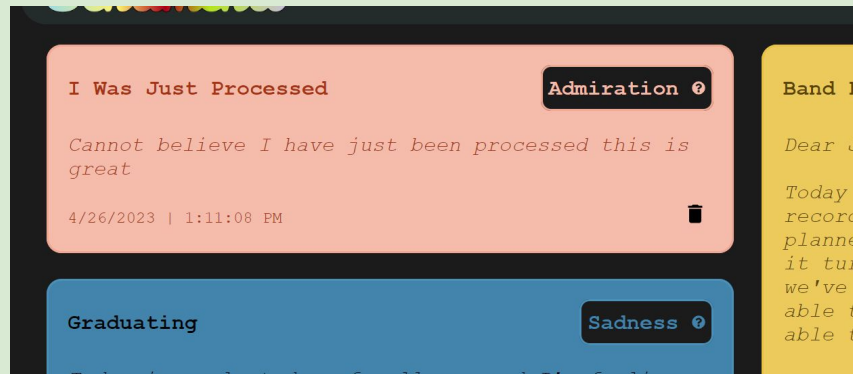
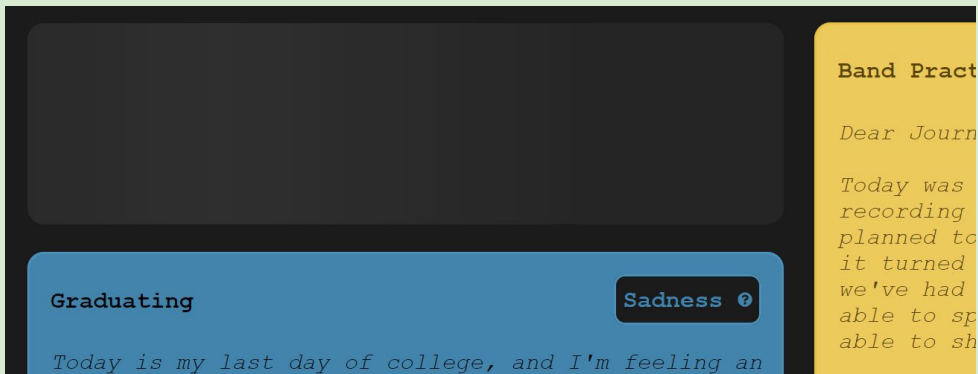
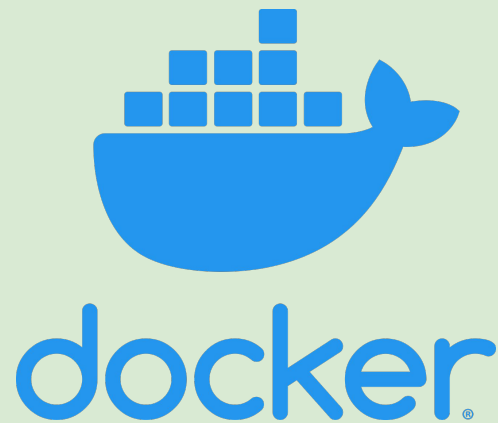
# ***Sentimento***

Final  
Presentation

Mario, Indré, Lupica

# Major Changes Overview

- Dynamic Color Feedback
- Back End - Front End Communication
- Docker!



# Front-End Progress In Depth

**Technologies** - HTML, CSS, JS, React JS, Vite JS

**As Of Today** - Sentimento now dynamically produces an associated mood/color on the submission of a new journal. Each entry now also provides a top three prevalent emotions breakdown with corresponding color and percentage. A login page is in the works, but unfortunately was not in a demo-able state for today

**What We'd Change** - More extensive design work before implementation (wireframes). In future projects, we would look into CSS frameworks to improve code quality and speed of development

# Back-End Progress In Depth

**Technologies** - Python, Flask, SpaCy, SQLite

**As Of Today** - Our script for emotion detection has been incorporated into our flask app. The frontend can now make a post request with a journal entry and the backend will process the entry and return back all the stats.

**In Progress** - We switched from SQLite to Postgres very late in the game. At different points throughout this project we've successfully done a few database operations, but none of which was ready for demo.

# What We Learned

- React
- Docker\*
- Python
- Researching and using resources/documentation
- Working with an API
- NLP
- Flask
- Just
- Pipenv

# Potential Future Plans

- Graphs/Trendlines
- Resources for emotions
  - Nervousness, anxiety, etc.
- Overview/Filter based on time
  - Week, Month, Year
  - Calendar view with each day with color, monthly (could make a new color)
- Weekly/Monthly/Yearly Updates
- Users choosing a color scheme
- Integration with other apps
- Prompts
- Reminders
- Goal Tracking
- Database functionality
- Separate builds for production and development
- Functional user accounts

# Live Demo

# Sentimento

