



# PhotoPhonic

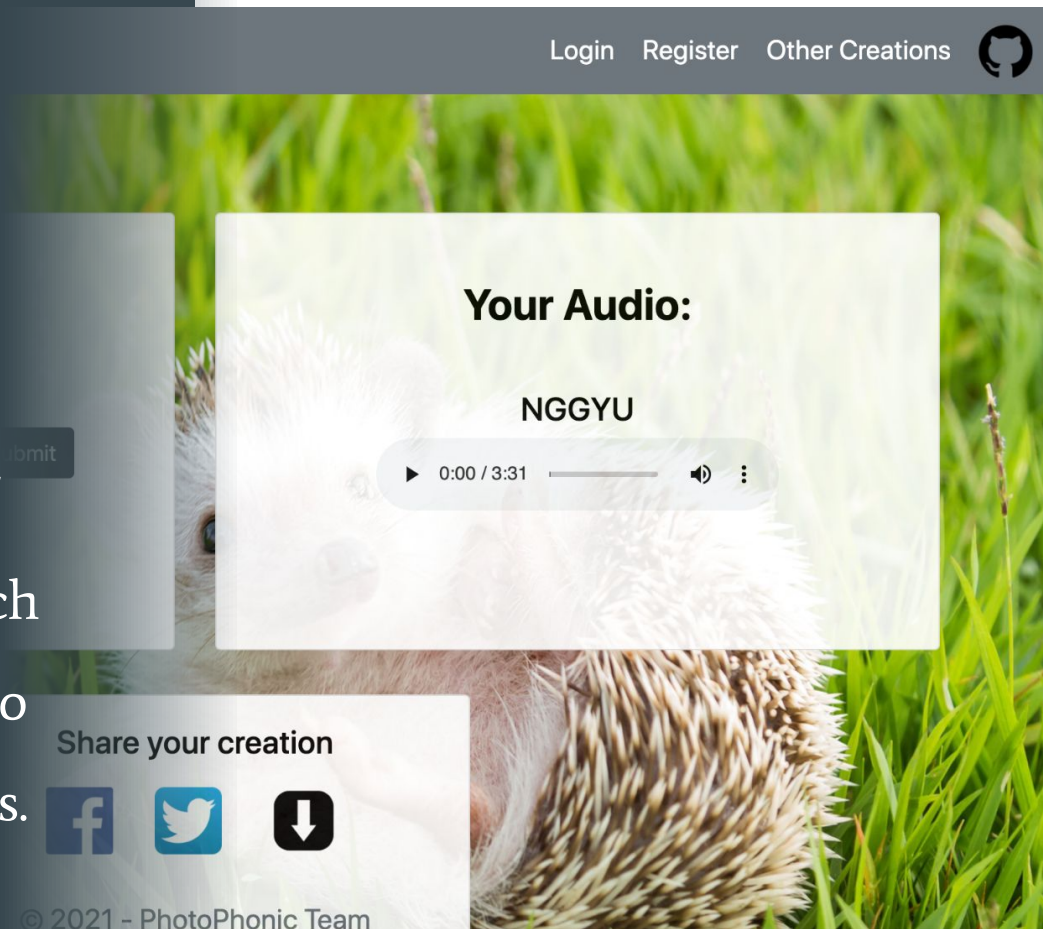


The PhotoPhonic Team: 013-6

Bella Barbieri, Spencer Stepanek, Dylan Ginsberg, Luke Pan, Trevor Aweida



PhotoPhonic is a *virtual  
synesthesia machine*, which  
artistically generates audio  
based on uploaded images.



# Management Tools



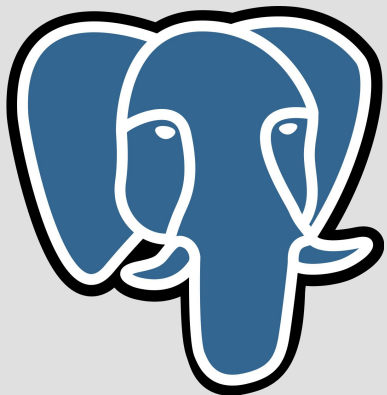
## Jira - Project Task Manager

- Convenient visuals to identify tasks at hand
- Using **scrum** methodology we completed about three 3-week sprints
- We rate it **9/10**

## GitHub - VCS

- Merge conflicts were common
- Generally extremely beneficial for keeping us up-to-date
- We rate it **10/10**

# Languages



## Python - Back-end, front-end

- Necessary for image processing
- Notable learning curve
- Well-documented
- Many packages used to replace tools typically in JS (Node, EJS)
- We rate it **7/10**

## PostgreSQL - Database

- Contains user and image data
- Manages image/audio UUID's
- Pain in the butt with Flask
- We rate it **8/10**

# Python Modules



## Jinja - HTML templating

- Almost identical in use to EJS
- We rate it **9/10**

## SQLAlchemy - Database interaction

- PostgreSQL query handler
- Does not like pre-initialized DB's
- We rate it **4/10**

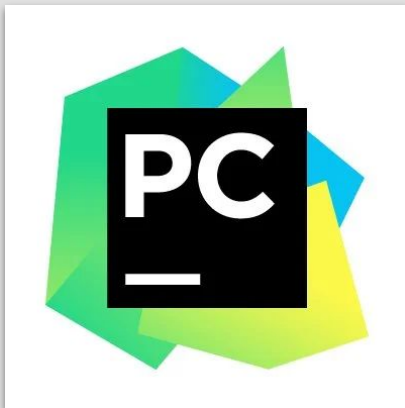
## OpenCV - Image Processing

- Super steep learning curve
- Very versatile
- Completely outfitted with image processing capability
- We rate it **9/10**

## Flask - Web Framework

- Imagine NodeJS for Python
- Maps very well to Node functionality
- Interesting learning curve
- We rate it **6/10**

# IDE's



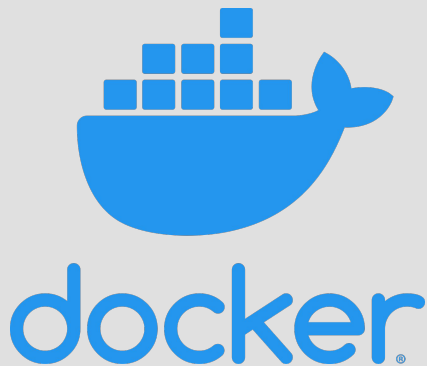
## Visual Studio Code

- Versatile IDE language-wise
- Lots of helpful plugins
- Good color coordination
- We rate it **10/10**

## PyCharm

- It does all the things
- *Excellent* python module management
- "Code With Me" allows for real-time collaboration with teammates
- We rate it **11/10**

# Miscellaneous



## **Docker** - General Testing

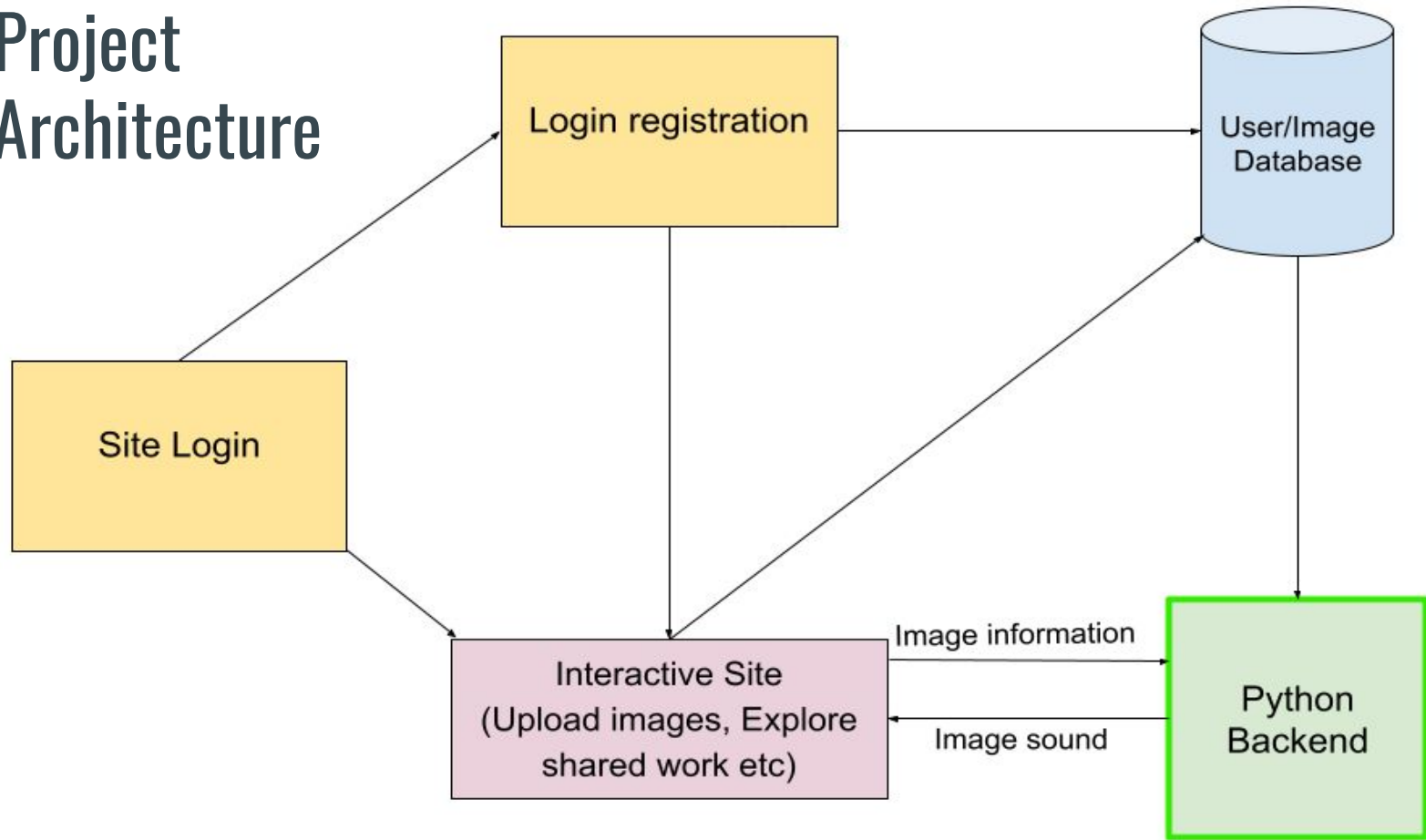
- Docker-Compose was tricky with Flask + PostgreSQL
- Starting containers is quick and easy
- Just doesn't work sometimes
- We rate it **6/10**

## **HTML, CSS (Bootstrap)** - Web-Page style

## **AJAX** - Dynamic page elements

## **Localhost** - Deployment

# Project Architecture





# Notes on Development

what was **easy** and  
what was **hard**

**Challenges** included:

- Linking PostgreSQL with Flask SQLAlchemy
- Developing a system for unique filenames to avoid updates to files getting caught in the cache
- Calculus for image analysis was required

Things that were **surprisingly easy**:

- Embedding audio and playback functionality into a webpage