

https://github.com/tyfa7215/3308_milestoneSubmissions

Project Features List

- User account
 - Scan logos and receive info on the specific logo
 - A way to save previous scans
 - Share scans and events
- Client account
 - Add new logos
 - How many people scan logos
- Image recognition
 - Google Cloud vision
 - Logo color recognition
- Web interface
 - Upload image for scanning
 - Direct to company websites and provides other information based on color and other aspects
- Mobile interface (Optional if time left)
 - Connection to the website in a mobile-friendly format
 - Focuses on paired down user focused interface
- System camera implementation
 - Allows user to use their system camera to take and upload a photo
- PostgreSQL database
 - store user information, image tags, and client information.
- Link Verification
 - Verify against malicious links

Requirements

- User Account
 - Functional Requirements
 - Clean interface
 - Login
 - Upload, scan, and save images
 - Non-Functional Requirements
 - Database interactions
 - Web authentication
- Client Accounts
 - Functional Requirements
 - Clean interface
 - Login
 - Upload logos
 - Shows statistics for scans of their logos
 - Non-Functional Requirements

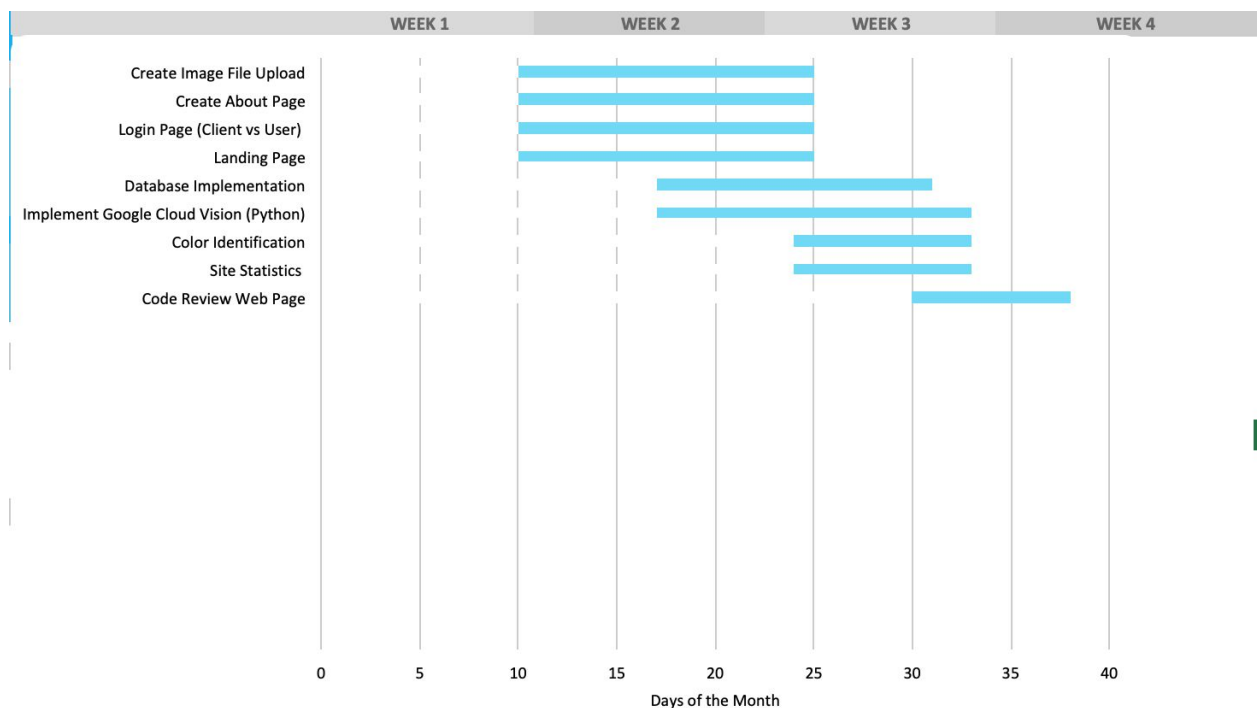
- Database interactions
 - Calculate and display analytics
 - Web authentication
 - Store client info
- Image recognition
 - Non-Functional Requirements
 - Image input/output with recognition software
 - Utilize a RESTful API implementation of Google Cloud Vision for logo recognition
 - Python OCR library for other image recognition features such as colors
- Web interface
 - Functional Requirements
 - Different web pages for different user functions
 - Upload button for uploading an image
 - Dashboard for clients to view logo stats
 - Non-Functional Requirements
 - Seamless script execution. The web interface would call each of the scripts for image analysis.
- Mobile interface
 - Functional Requirements
 - Users can use phone camera to upload a logo
 - Will be mobile friendly
 - Will have all the same elements as the web interface
 - Non-Functional Requirements
 - Ties in with web interface
- System camera implementation
 - Functional Requirements
 - Rather than just uploading existing image files to the website, the user can use the built in camera on their device to photograph and upload images.
 - Non-Functional Requirements
 - This would then need to be sent to the web interface to then be sent the image analysis scripts.
- PostgreSQL database
 - Functional Requirements
 - None
 - Non-Functional Requirements
 - A way of communicating with the database, all information and data will be kept here. For every display and function call, we can query the database. Either the script code or webserver will query the database to get information
- Link verification
 - Non-Functional Requirements

- Link verification that client link works and redirects to a trustworthy site prior to addition to database

Project Plan

Gantt Chart:

TASK NAME	START DATE	DAY OF MONTH*	END DATE	DURATION* (WORK DAYS)	DAYS COMPLETE*	DAYS REMAINING*	TEAM MEMBER	PERCENT COMPLETE
First Sample Project								
Create Image File Upload	10/10	10	10/24	15	0	15	FE Team	0%
Create About Page	10/10	10	10/24	15	0	15	FE Team	0%
Login Page (Client vs User)	10/10	10	10/24	15	0	15	FE Team	0%
Landing Page	10/10	10	10/24	15	0	15	FE Team	0%
Database Implementation	10/17	17	10/30	14	0	14	BE Team	0%
Implement Google Cloud Vision (Python)	17-Oct	17	1-Nov	16	0	16	BE Team	0%
Color Identification	10/24	24	11/1	9	0	9	BE Team	0%
Site Statistics	10/24	24	11/1	9	0	9	BE Team	0%
Code Review Web Page	10/30	30	11/6	8	0	8	Both	0%



High Level Team Assignments

- Front End
 - Yu Li
 - Jeremy Smolens
 - Kevin Sullivan
- Back End
 - Tyler Faragallah
 - Robert Meikle
 - Ian Jorquera