

Spring Term Final Project

“GOOGLE KEEP”

TEAM MiLKShake

Kristin Lin, Levi Olevsky, Sonal Parab, Masha Zorin

DESCRIPTION

A recreation of Google Keep, an online post-it note taking application that allows a user to create and edit notes by typing, drawing, uploading an image or other things. The notes can be labeled or color coded the way the user chooses for better organization. Also, the notes can be arranged/rearranged on the dashboard.

COMPONENTS

HTML files

See 'Routes' section for details

CSS Stylesheets

Bootstrap's as well as our own

app.db

Database to hold information on users and notes taken. See 'DB Schema' section for details.

database.py

Functions that will interact with the database from the Flask app. Functions may include add_user, add_note, find_note, find_user, set_note_color etc.

app.py

Flask framework to connect the backend and the frontend.

notes.js

Helps user interact with notes present on the home page. Will use AJAX calls to update user's choices in the database. Will use d3 with svg to manage notes.

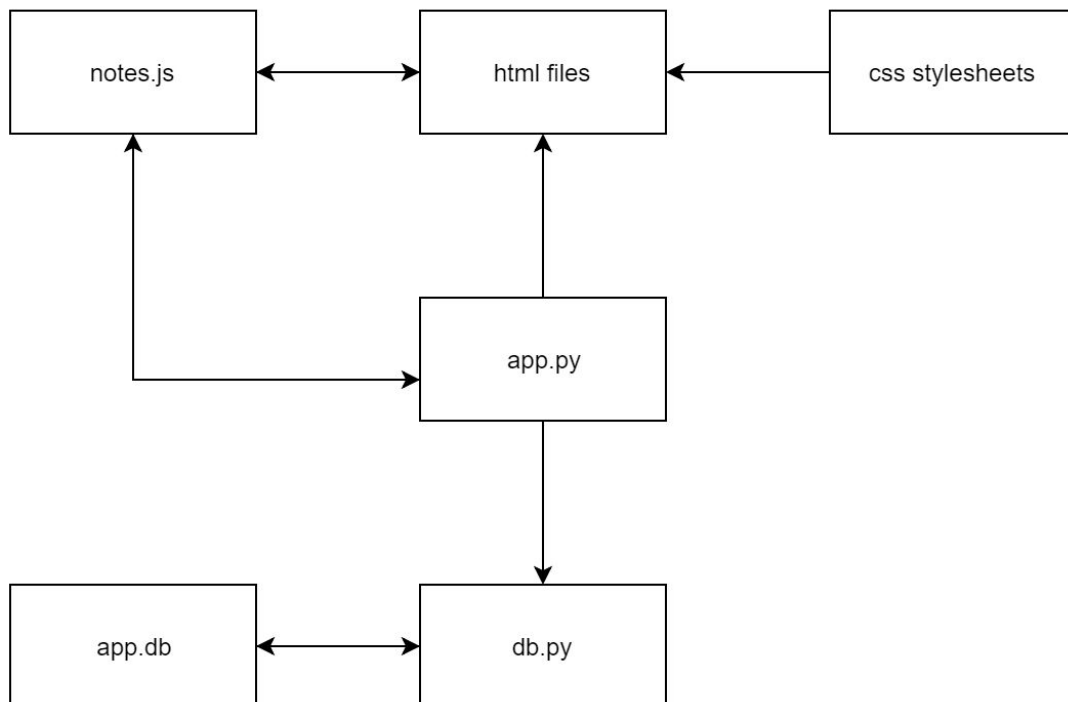
Possible Actions that Functions Must Cover:

- Create note
- Change note to checklist
- Change color of note
- Add images
- Drag/reorder notes
- Pin notes to top
- Archive notes (sends to archive page)
- Delete notes
- Select notes
- Label notes

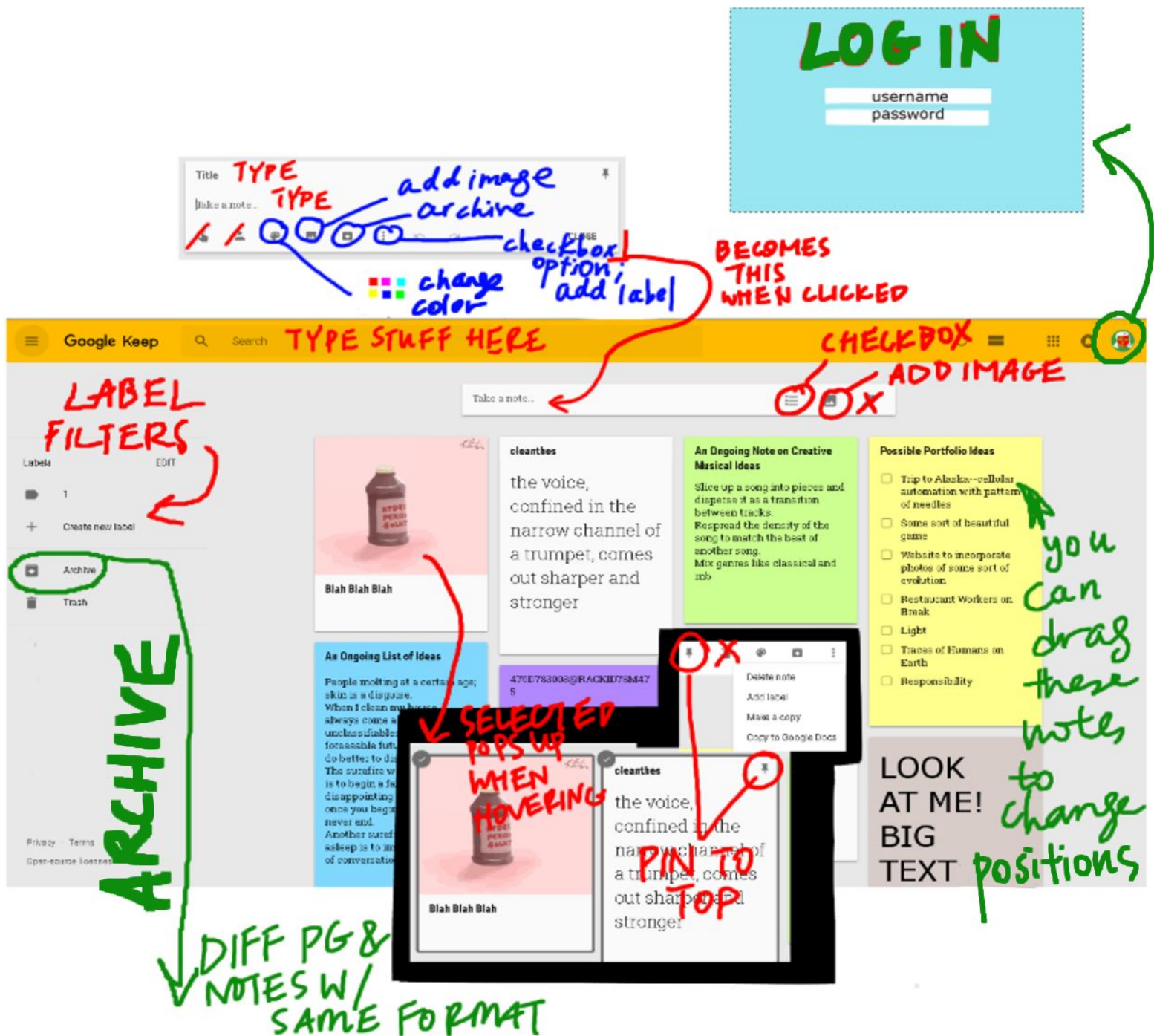
ROUTES

- **/**
The front page, which will display a description of the website, along with some basic information you need to know and maybe a sample arrangement of notes.
- **/login**
Login page for username and password input. Contains button to go to /signup.
- **/signup**
Page to create an account, with username, password, and confirm password. Contains button to go to /login.
- **/home**
Dashboard with all the notes
- **/search**
Filters notes. Will call database function to find notes containing keyword in title or content
- **/archive**
Dashboard with the notes that were archived

COMPONENT MAP



SITEMAP



DATABASE SCHEMA

users table

The users table will store the user's username, password.

notes table

The notes table will store the username, note id, note type (note, list), note content (list items separated by backslash, image file separated from message by backslash), whether it is pinned or not, order id (keeps track of what order notes are displayed in), color, whether it is archived or not, labels

labels table

The labels table will store the label name and note id (to keep track of labels of notes, and all labels in general)

note table

The note table will store the content of a note with type "note". It will store content, image (can be null), and note id.

list table

The list table will store the content of a note with type "list". It will store list item, item order, whether it is checked off, image (can be null), and note id.

reminders table

The reminders table will store the note id, the date, time, and repeats (once, daily, weekly, monthly, annually).

users	notes	labels	note	list	reminders
user *PRIMARY KEY (string)	note_id *PRIMARY KEY (int)	label_name (string)	note_id (int)	note_id (int)	note_id (int)
pass *encrypted (string)	user (string)	note_id (int)	content (string)	item_content (string)	date (string)
	note_type (string)		image (string - the link to image)	item_order (int)	time (string)
	pinned (bool)			checked (bool)	repeat (string)
	order_id (int)			image (string - the	

				link to image)	
	color (string)				
	archived (bool)				

ROLES:

- Project Manager & CSS: Kristin
- Database: Masha
- Flask App, HTML Templates & JS: Sonal
- More JS & Ajax: Levi

REACHES:

- Google Authentication instead of our login
- Add drawing
- Reminders
- Storing custom images the user wants to upload