GIF Search

## Description

So far you’ve gotten your hands dirty with HTML, CSS, and Python by learning what the FLASK framework is all about. Now it’s time to bring it all together in a practical way by building a GIF search page! **You will be working with a partner during the course of this project.**

Don’t worry, we’re not trying to build a Giphy competitor (not yet, anyway…), but we will be using [Tenor](https://tenor.com/) to get our GIFs. Through this project, you will get hands-on work with building a search page that uses basic routing to display content, and will allow you to get practice with integrating an API!

## Learning Outcomes

By completing this project, you should be able to…

1. Set up a development environment for building a FLASK-powered website
2. Use templates to quickly create layouts for your website
3. Build basic route logic that uses parameters
4. Integrate a simple API into your project
5. Style elements in your project beyond the default

## Schedule

From the day this project is assigned, **you will have 2 weeks to complete this project**. A sample daily outline is provided to assist you in planning your project.

**Important note:** “Day 01” refers to the first calendar day of the project being assigned, and subsequent days will follow this reference:

* **Day 01:** Project is introduced, partners are assigned
* **Day 02:** Both partners have their environment is set up, GitHub repo created, and are ready to begin coding the project
* **Day 05:** A home page has been built with a “Hello World” style message being displayed
* **Day 08:** Tenor API has been integrated, able to display a single GIF on the home page
* **Day 09:** Search bar has been added to the home page (search functionality not working)
* **Day 11:** Able to search Tenor for gifs based off of a search query
* **Day 13:** Page stylized to your choosing, final clean up/improvements done
* **Day 14:** Project submitted, you and your partner celebrate this milestone!

## Starter Code

The starter code for the project is located [here](http://github.com/Make-School-Labs/Gif-Search-Starter).

## Requirements

1. The page must use templates
2. The page must display GIFs (10 at the most)
3. GIFs should appear in a single vertical list
4. At the top of the page there should be a page title
5. Below the title there should be a search bar with a “Search” button near it (placement up to you, but needs to be on one side of the bar)
6. Users should be able to type a string into the search bar, press the search button, and be shown up to 10 GIFs related to the search query
7. GIFs should be displayed on a fresh load of the page, i.e. before a query has even been typed.
8. GIFs should only update once a user has pressed the “Search” button
9. If no GIFs could be found for the search term, display an error message saying that no GIFs could be found, and to try another search query
10. The following elements should have some custom styling (i.e. CSS rules) added to them:
    1. Page title
    2. Search Bar
    3. Search Button
11. All code must be commented with a description of what the code is doing, expected input, and expected output

### Stretch Requirements/Challenges (Optional)

* Add a [gitignore](https://git-scm.com/docs/gitignore) file and edit it so that “.DS\_Store” and “.env” won’t get tracked in Git. What else shouldn't be tracked?
* Center-align everything on the page
* Display the GIFs in a grid instead of a list
* Add a button that displays the top 10 trending GIFs on Tenor
  + Check the [documentation](https://tenor.com/gifapi/documentation#endpoints-trendinggifs)!
* Add a button that displays 10 random GIFs on Tenor
  + Check the [documentation](https://tenor.com/gifapi/documentation#endpoints-random)!
* Type-ahead: as the user types in the search box, the page is reloading the gifs to match the search query in real time (no longer needing to click the search button)

## Rubric

You can read the [project rubric here](https://docs.google.com/document/d/1u8zn_w9kQceK1y0f0F6QEWWgP8T7KRsQvQOIvlzyMi0/edit?usp=sharing). **You must get an average score of 2 to be considered passing.**

## Commit Requirements

**All projects will require a minimum of 10 commits (5 per partner minimum), and must take place throughout the entirety of the project**

* **Good Example:** 40+ commits throughout the length of the project, looking for a healthy spattering of commits each week (such as 3-5 per day).
* **Bad Example:** 10 commits on one day during the course and no others. Students who do this will be at severe risk of not passing the class.
* **Unacceptable Example:** 2 commits the day before a project is due. Students who do this should not expect to pass the class.

### Why are we doing this?

We want to encourage best practices that you will see working as a professional software engineer. Breaking up a project by doing a large amount of commits helps engineers in the following ways:

* It's much easier to retrace your steps if you break your project/product/code up into smaller pieces
* It helps with being able to comprehend the larger problem, and also will help with your debugging (i.e. finding exactly when you pushed that piece of broken code)
* It allows for a more streamlined, iterative communication in your team, as it's much easier to hand off a small change to someone (updating a function) than a huge one (changed the architecture of the project)

Through this requirement, we hope to encourage you to think about projects with an iterative, modular mindset. Doing so will allow you to break projects down into smaller milestones that come together to make your fully-realized solution.

## Resources/Tips

Additional resources that will help with this project, or that can be used as reference

* [Tenor API Quickstart](https://tenor.com/gifapi/documentation#quickstart)
* [How to use templates in FLASK](https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-ii-templates)
* [Jinja Documentation](http://jinja.pocoo.org/docs/2.10/templates/)
* Use the following code snippet to display your GIFs. Make sure to replace the TODO statements with actual code

<div class="tenor-gif-embed" data-postid="TODO" data-share-method="host" data-width="500px"

data-aspect-ratio="1.2266666666666666">

<a href="TODO"></a>

<a href="TODO"></a>

</div>

<script type="text/javascript" async src="https://tenor.com/embed.js"></script>