

Photo Database

Utilizing Python, Flask, and SQL databases

John L. McGraw

Northern Michigan University

December 2021

Introduction

For my senior project I developed a web application implementing a photo database. I utilized technologies such as Python, Flask, and MySQL along with numerous libraries. At its core my web app enables individuals to create a user account allowing them to upload, share edit, and download images. Some features and functionalities are as follows:

- Implemented a user login system. Multiple users can be logged in at any given time and photos are linked to which user uploaded them.
- Sends a verification email to a user specified email to activate account.
- Allows users to upload images in a range of supported formats, storing them in a MySQL database.
- Allows users to “Like” and “Dislike” photos, displaying the number of each next to the image updating in real time.
- Allows users the ability to edit photos in numerous ways – users can apply filters, transform and transpose images, along with other functionality.
- Once uploaded users can redownload images in varying quality, resolution, and file size. Lower quality - smaller file sizes.

The ambition for this came mainly from wanting to learn new technologies and how to make them interact with each other on a real-world level. I never experienced any solid web

development before and decided this was a perfect opportunity to learn a bunch of new things that I might use in the real world.

Technologies in Use:

Flask

Flask is a web microframework, developed in Python, which allows development of web applications. There are numerous libraries written for flask which grant the ability to do some intriguing things. Flask natively supports URL routing, which I used extensively throughout my project. The majority of my code was written to be used within flask routes.

MySQL

MySQL is a database management system. My project uses multiple tables held within a single database. Tables are used within my project to enable user login (passwords hashed in sha256). To hold user data such as username, password, user id, email and verification status. MySQL tables also used to implement my “like and dislike” feature and yet another table holds the image file data.

Python 3

My project is mainly developed using Python 3. I chose Python for a collection of reasons. The main reason being the rate at which its growing in popularity. I previously had little to no experience with Python and decided using it for my senior project would get me

valuable experience. I thought having knowledge of one of the most popular programming languages would make me a more marketable candidate when entering the workforce in the coming months.

Challenges

Front End

One of the most challenging aspects I faced during this project didn't really have much to do with coding at all. However, I definitely struggled with the front end and user interface aspect of things. You could say its due to my lacking of artistic nature. I have no doubt in my mind that it's a respectable skill, one that I yet to possess. Before starting this project, I had heard the development terms "front end", "backend" and "full stack". However, due to having no web development experience before starting I didn't really know what they were. I am by far, more confident in my skills concerning backend development than front end. Making it so the user could easily interact with my project, arguably the most important aspect of the whole thing, was difficult. To make it a little easier on myself I used Bootstrap, which is a front-end CSS framework. It made designing things a lot less painful.

Databases

When creating the tables within my database I had to think outside the box. For example, the table that held the likes and dislikes for the images. For a while I thought about ways to somehow track which photos were liked or disliked and by who. I eventually came up with creating a database which consisted of four columns. These four columns held the values

of the image id, the user id, and a column for likes and for dislikes. When a user likes an image it creates a new entry by executing the following SQL statement "INSERT INTO likesDislikes(image, likes, user_id, dislikes) VALUES(%s,1,%s,0)" . By structing my table this way I'm creating an entry for every single like or dislike and connecting it to an image and user. The only downside is this can cause a large database table when scaled.

Vulnerabilities

Searching for vulnerabilities within my code was another difficult task. Knowing my project was going to be exposed to substantial user input I knew it would need to be more protected than code I normally write. Researching and thinking of different ways I could wreak havoc on my databases so I could implement safeguards was an intriguing task. I had to figure out different ways my databases were susceptible to SQL injection attacks and what I needed to do to eliminate the risks. I had to learn about "sanitizing" user input and how to properly use it to protect against attacks. This also include not only sanitizing inputted user data through forms but also checking names of images they uploaded.

Error Checking

Predicting mistakes users might make was again another challenge. As the one developing and creating the project I know how things are supposed to work, and that's the problem. What makes sense and seems intuitive to me might very well not be to a user. This can easily cause problems to arise. Thinking of every way a user might accidentally break my project is tedious and time consuming, let alone coming up with a solution for each case.

Something as simple as trying to upload a non-image file (even though it's clearly a Photo Database...) would cause some pretty serious issues before I created a safeguard against it.

Learning Curve

While I ran into quite a few challenges along the way, the amount I needed to learn was definitely the hardest part of the project overall. Needing to learn a new programming language and how to implement it caused a slow start to the project. Not only was learning Python a task in itself but learning the libraries that allowed me to do so many cool things was a challenge as well. Two of the libraries I used which required a bit of learning were PIL (Python Image Library) and cv2 (OpenCV-Python). These two libraries are what allowed me to implement most of image editing functionality. Because of the way I stored my images, as a long blob in a MySQL database, it created a few extra steps for me that not many others encounter. Below is an example how I would fetch an image from the database, add the desired affect, and return it.

```

@edit_routes.route('/greyscale/<int:id>')
@login_required
def greyscale(id):
    mydb = MySQLdb.connect(
        host=cred.host,user=cred.user,password=cred.password, database=cred.database)
    mycursor = mydb.cursor()
    mycursor.execute("select image_file from images where image_id = %s",(id,));
    img= mycursor.fetchall()
    mycursor.close()
    mydb.close()

    try:
        img = img[0][0]
        img=Image.open(BytesIO(img))
    except IndexError:
        return render_template("no-file.html",port = port)
    except UnidentifiedImageError:
        delete(id)
        return '<h1>Corrupted Image! <meta http-equiv="refresh" content="3;url=http://euclid.nmu.edu:'+str(port)+'/'+str(current_user.username)+'gallery"/> </h1>'

    if img.mode in ("RGBA", "P"):
        img = img.convert("RGB")
    imgBytes = io.BytesIO()
    img = ImageOps.grayscale(img)
    img.save(imgBytes, format='JPEG')
    imgBytes.seek(0)
    return send_file(imgBytes,attachment_filename='greyscale-'+str(id)+'.JPEG')

```

What I Enjoyed Most

This project carried with it many enjoyable attributes. One of the things I enjoyed the most however was implementing real world things. Things that you would actually see implemented by other applications. One example of this was the email verification feature. Adding this to my program made it feel like my project was really doing something, it was interacting with other things. If I had to chose one thing that my application did that was my favorite it would have to be this.

As previously mentioned, I learned a lot during this project. Knowing how much I learned is what I enjoyed second best. I learned skills that have a good chance of being used in a future career. MySQL and Python are two widely used skills in the real world and now I have at least an intermediate understanding of them.

Conclusion

Overall this project was a huge learning experience for myself and I had fun doing it. In all honesty I don't plan on stopping where I'm at. I've been having a great time adding new features and will continue to do so for experience and enjoyment. Some things I would like to add in the future are

- SMS verification
- SMS password reset
- Ajax implementation for image searching
- Allowing you to "follow" other users

Overall, I would give this project a medium difficulty rating. It took a number of things working together to come out with the finished project. I implemented all the features listed in my proposal and then some. While I can always add upon what I currently have I'm happy with how everything turned out thus far.