Milestone 3

Summary of What We Have

Have working website that is up and running (using Python Anywhere)

- Sample landing page(project info)
- Sample gallery page: shows an image and mosaic view can be entered by clicking button.
- Mosaic feature is working: Mosaic can be downloaded,
- Started to combine front and back end

All of the above features worked as we would like them to at their current state during the demo.

Have a sample for the aesthetic view of the site

- Have improved form of website visual
  - About page
  - Gallery page with sample images to pick/queue from database or upload
  - Mosaic is loaded on page with option to download
- This doesn't interact correctly with file system but looks better

TA Questions/Answers (Bold Text is TA, Regular Text is Us):

Can you explain more about DB connecting whole working sites?
It is hosted using Flask and Python Anywhere. Different templates are set with render_template.

What are your DB connections? Have you implemented the database part?
Page loading and the image database that is running on the site.

Is your flask connected to that?
Not yet but we are working on it.

What are other features/next steps?
Be able to left click on the smaller images in the mosaic and open a popup menu.
The following options featured in the menu would include: "Learn More About (Selected) Image" and "Reshuffle" which makes the image chosen the mosaic

What is the most difficult part of the project?
The reshuffling option is the most difficult part; we are still working out ideas for it.

How are you planning to use the DB?
The ability to upload your own images and pull images is the biggest part of our database.

How are you manipulating the images?
We are currently using a photomosaic library in Python. We may have to repurpose it in the future and split it apart to modify it to fit our needs.

Is this why you stuck with flask?
Yes, the Python library is very good.

Great work guys, would've loved to see DB working but having a complete website is pretty cool.
The database is working, we just have not implemented it within the site. It's hosted on the same hosting service as the website, so it's not local.

Is the DB server running?
Yes.

Is the python app connected to DB/Credentials connected?
We still need to connect the app and the DB.

This is cool guys

We have a question, our commits to git aren't representative of who did what for this milestone, so how can we be graded more evenly?
In the milestone 3 document have who did what outlined for us to grade.

Suggestions:

- **Have a local version and then put it hosted**
  - **We could have local DB and connect it and it will give you more control and you will learn more**
- **Maybe have a community/contributor to post mosaics**
- **Implement a zoom function on the mosaic**

Milestone 3 Contribution Summary

Sophie and Elizabeth did work on the front end, made the templates for the home page, gallery page, and mosaic page. Sophie made the mosaic page used in the demo and Elizabeth made the home page used in the demo. Kevin made the gallery page used in the demo. Elizabeth made secondary improved versions of both the gallery and mosaic page that were also presented in the demo.

Kevin connected front end and back end elements, hosted the website through pythonanywhere with python and flask. Made a SQL database for images to be used in the project. Spent time modifying front end and back end elements to work together smoothly within the hosting service. Spent time writing scripts to work with image uploading into the database, and figuring out how to connect the database to the website's functions.

Sky, Jerry, and Tim made back end elements function with the photomosaic, and spent time looking into how to split up the mosaic image for functions coming later. Sky and Jerry found the initial photomosaic method and code that was the foundation of the function that was implemented in the demo. Tim then spent time looking through the library finding and testing other functions provided in the library to optimize the image output.

Issues Faced Throughout Development

Implementing the python algorithm to generate the mosaic images, and optimizing said algorithm to make images look better based on the given pool and size of the mosaic.

Figuring out how we are going to implement the reshuffle functionality which selects one of the smaller images in the mosaic and uses it as a new template for the mosaic.

Overcoming formatting challenges with html/css involving the placement of elements in specific positions to fit the overall feel and aesthetic of our site.

Hosting the website and making Flask work with it properly.

Creating a SQL database to hold images and implementing it into the functioning website.

Developing the popup menu that triggers the reshuffle function in python. Circles back to the overall challenge of getting front end components (HTML/CSS/JS) to interact with backend components (Python, DB etc.)