Anvil Primer for HW1

It's one thing to be able to write code that you can use, it's a whole other task to write code that is useful for other people too. For extra credit on your midterm, I want you to build a webpage that allows a user to upload a csv file of x-y coordinate pairs, solves the TSP on those pairs using simulated annealing, displays the results in a graph, and allows the user to download a new csv with the path information in it.

This seems like it's a tall task, and it is! However, there is an online tool that makes this very manageable - https://anvil.works

Anvil is a python-based web development tool. It allows you to build dynamic web apps that run on python. It allows you to drag and drop elements onto a page, like WordPress or another gui based website builder, but then it lets you use python to modify what is displayed in those elements.

I have no stake in anvil. I found out about it this summer when I needed to quickly build a page for a startup I'm working with. The standard way to build a webpage with python is via flask or django. These both have a steep learning curve and require knowledge of html. Anvil on the other hand is very quick to learn and requires no html knowledge!

Anvil is free to use but the free version is limited: you can't use numpy, pandas or other similar packages. I have arranged for free academic licenses to the full version of anvil that will expire in July. Simply click this link: https://anvil.works/build#licence:DCBCE7098AG1673F710BE4E8 to get access to your license. Be sure to create your account with you @utexas.edu email address.

The user community for Anvil is large and the forums are actively monitored by Anvil employees to answer questions. If you have a question of how to do something, you'll likely be able to find the answer on their page.

Anvil has many blog posts that teach how to use their tool. Here are a few that may be helpful.

1. https://anvil.works/learn/examples/meter-feeder
2. https://anvil.works/blog/plotting-in-matplotlib
3. https://anvil.works/docs/media
4. https://anvil.works/learn/tutorials/jupyter-notebook-to-web-app
5. https://anvil.works/forum/t/how-to-add-multiple-pages-within-an-app/74/3
6. https://anvil.works/docs/client/python/alerts-and-notifications
7. https://anvil.works/docs/deployment/quickstart
8. https://anvil.works/docs/users

Your anvil webpage should:

1. Require a user to sign in with email and password
    a. Create an account for me with email: dan and password: Optimization1234
        i. I know dan is not an email address, but that's actually ok if you create the account for me!
    b. Do not allow users to create new accounts automatically.
    c. The rest of the page should only be accessible if a user is logged in.
2. Have a detailed description of how you built your NN, some pictures of well classified vs ill classified image, and analysis of why some images are classified poorly. Think of this as a medium article you would write about the mnist dataset and your network.
3. Allow the user to upload a csv file
    a. The csv file should have 28 rows and 28 columns, with no header row. Each entry should be the 0-255 pixel intensity.
    b. If the file uploaded is not a csv or if it is not in the correct format the webpage should alert the user that the file is not correct.
4. If the file works, the upload button and any description of what to upload should disappear.
5. Classify the image and display it.
6. Have a button that allows the user to start over: upload a new file and classify another image
7. Publish the app at the following url: MSBAoptim_{YOUR group number}.anvil.app
8. Make your site look nice, 50% of your grade will come from the appearance of your site.
9. Include a link to clone your anvil code in your submission.