Elizabeth Thompson

Final Project Written Report

Introduction

The goal of my final project is to create a dashboard that produces instructions, called a pattern, for knitting a drop shoulder sweater based on the user's inputted characteristics.

In the knitting community, many users will write a pattern that outlines how to make a sweater and list it online for free or a small profit. Although there is an abundance of patterns available, each pattern is designed for one specific sweater, contains instructions for every size, and are not editable. This can be overwhelming for beginning knitters and cause difficulty finding a pattern that fits knitter's desired customizations. This dashboard allows knitters to create their own patterns without purchase with their exact sweater characteristics.

It is fully customizable for sizing, neckline types, sleeves, optional embellishments, and yarn type, but are limited to the styles that I know how to knit. They are as followed:

Sizes: Small, Medium, Large, X-Large
Neckline types: Boat neck, Crew neck, Mock neck, Turtleneck
Sleeve types: Straight, Tapered, Balloon, Flare
Embellishment types: Ribbed edges, Ribbed collar, Ribbed edges and collar, No embellishments

Methods

I used 4 different data sets to create this project, all created by me. The first is the small csv file called knit_data.csv that contains the dropdown options. The second is a size dictionary that contains the necessary sweater measurements by size. These numbers were found by researching standard sweater measurements and adjusting as I saw fit. The sources I referenced are in the bibliography. The third are the images of the sweater that dynamically change. I used Notability to draw every possible sweater variation and saved them to the folder called assets. I renamed each image so that the dashboard can pull the correct image based on the dropdown selections. The last form of data is the pattern itself. This pattern was written from my personal experience of knitting a drop shoulder sweaters and follows the methods I used. The sweaters I have made have not been made using patterns but instead using intuition and past experience in knitting. The knit_data.csv can be found in the data folder of the repository while the size dictionary and pattern are written into the SweaterPattern class in the pattern.py file.

The project is split into 3 files: dashboard.py, pattern.py, and main.py. The dashboard file contains the functions used to create the dashboard. The run_app function creates the dashboard and launches it in the terminal. The create_layout function defines and outlines the components of the dashboard and adds them to the dashboard's layout. Create_layout is called inside run_app for clarity in the py file. The update_image function dynamically updates the sweater image on the dashboard based on the selected dropdown options. The update_rib function displays the necessary rib inputs based on the embellishment selection. The update_edge function displays the edge input based on the embellishment selection. The update_collar function displays the collar input based on the embellishment and neckline selections. The update_size function displays the sizing inputs if the user chooses to enter their own measurements rather than using the stored sizes. The generate_pattern function calls the SweaterPattern class from the pattern.py file to create the pattern based on the selections made on the dashboard and downloads the pattern as a pdf in the patterns folder. The pattern.py file defines a sweater pattern class for writing the pattern based on the user's inputs. The constructor defines all instance attributes needed for the pattern based on the user's inputs. It also calculates the increase and decrease variables if they are needed for the pattern. I used Margaret Corrigan's "Technical Editing for Knitting Patterns" website when I first made a tapered sweater and used the knowledge I gained to write the patterns for the tapered and flared sleeves. The layout function contains the written pattern and uses if statements to differentiate between the user's inputs. The generate_pdf function turns the pattern into a pdf and defines the customization I made to the pdf. These customizations include color, font, spacing, bolding, and underlining. The main.py file calls the run_app function.

The dashboard is created using Dash from Plotly. I referenced their website often to understand how to properly use the library. I also used Flask to save and return the pattern file. I originally used Dash's serve_pdf function but was experiencing issues with the files downloading. After switching to Flask, the issues went away.

Demonstration

To test my calculations, I decided to compare an old handwritten pattern I made to one the dashboard would produce. I used the same measurements and characteristics that I chose when I knit the sweater and input them to the dashboard. This pattern is called Elizabeth's Sweater and is in the patterns folder in the repository. After checking the pattern against the calculations I made when knitting the sweater, I found that the results were extremely similar. There were a few discrepancies, but this is because I started to knit my stitches looser after calculating, so my measurements were underestimated.

I intend to use this dashboard to make my subsequent knitting patterns and plan to add more dropdown options as I learn them. I have so far created two sweater patterns that I am planning to make, one for a friend and one for myself. These are in the downloads folder as are called Ana’s Sweater and Slouchy Sweater.

Bibliography

https://forum.knittinghelp.com/t/rapid-decreasing-of-sleeve/147114

https://www.craftyarncouncil.com/standards/woman-size

https://tradlands.com/pages/sweater-size-guide?srsltid=AfmBOoo6fe5uSnoP24AKjMJ-S1q8c0tB4Tlwc9LF94ufrSfgGjdcReg9

https://www.midnightpurl.com/news-notes/taperedsleeves

https://dash.plotly.com/