Assignment 4 Report
Ayesha Akhtar and Jake Parrott

Design Choices/Changes from Assignment 3:

For this assignment, we followed the guidelines listed for the assignment to incorporate CRUD operations and use the express-validator and passport libraries to make our Chirpy App more robust. We first began by defining the new controllers and models and updating the existing ones to use req, res, flashes, and all the other commands the assignment guidelines specified. As of now, our application uses these calls to authenticate, validate, and pass error and success messages to the user. While our sessions and cookies have been added in, our flash messages and authentication checks help guide the user through the signup and signin pages to get to home. Additionally, we have chirpsController that handles all the chirp specifications and a usersController that does the same for the users. The main functions defined within these controllers are: show, delete, new, create, update, and redirectView.

To accommodate each of the functions, we defined a show, edit, index, and new ejs folder for both the chirp and users model. Though the names of these pages are self-explanatory, I'll give a quick overview: the show pages show a particular chirp and/or user, the edit pages allow the user to edit their chirp or profile, the index page lists all of the chirps made, and the new page allows the user to create a new chirp. With the index page, it is worthwhile to note that while the user can see each and every chirp, they can only edit and delete the chirps they themselves have made. Additionally, while we did make an index page for the user as well, it contains nothing because we updated our existing home and made a chirpsFeed.ejs file to tackle the posts displayed on the home page. We also did create a new.ejs file for the users model, but it also is not being used since we edited our signup page and that acts in the same manner as the new.ejs would. To see and test our newly added functions, you can find buttons in the navigation bar on the left side of the homepage. Our "Followers" table is displayed using a usersFeed.ejs file and this shows the profile listings of the user accounts in the database. When you click on a listing, it takes you to the user's profile page where you can view their information. A decent amount of our time was also spent either redoing the css for our existing and newly edited pages or making new css files for our new pages. Lastly, our home page's trending news feed is also currently displayed with accurate and real news articles; when you click on a title, it'll take you to the link.

Work Division and Future Plans:

In terms of how the workload was split, Jake focused on the controllers, making route paths, and writing Mongoose queries while Ayesha focused on incorporating the CRUD operation files and writing and updating the CSS. Jake also made the seed adjustments and Ayesha wrote the report. For our future plans, we would like to make the search bar usable, add hashtags, filter user homepage views with follows/follower logic a la Twitter, and to actually host this site somewhere. In our last report, our goals were to make the home page a lot more interactive and to specifically tackle the user followers table. We were able to reach and effectively complete this goal for the assignment. Additionally, we added in the option of updating posts; while this is not something that is currently allowed on Twitter, Chirpy allows its users to edit, update, and delete their posts. In addition to these options, we'd also like to eventually give the user the option to upload images; though we had hoped to do so for this assignment, we ran out of time. Even so, we do have the icons available and set up- it's just a matter of making them interactive!

How to Launch App:

To launch the app, you will need to clone the app into your machine. If you do this straight from the terminal, the command will be:

- git clone -b Assignment4 https://github.com/jparrott06/ChirpyApp.git
- From there, you can cd into that cloned file and start the program. Assuming you have nodemon and mongoDB installed (if not, install both of those first), use the following command lines:
- npm install (this will install the dependencies for the application)
- node seed (this populates the database with 4 dummy users)
- npm start (this starts running the application)

That should connect to the localhost and then you can go on your browser, type in “localhost:3000”, and begin exploring :)

[NOTE]:

- We have commented out the client-side validation for signup so it is easier to test our server-side validation with express-validator/flashMessages. We will obviously add it back in for the final iteration of this project.
- For users/edit we made certain design decisions to make select fields required. In future iterations, we’d like to have a separate route for changing information like password, DoB, email, where the user must enter and validate their current password to change it.