Design Choices/Changes from Assignment 2:

For this assignment, we followed the guidelines listed for the assignment to incorporate EJS and Express into the Chirpy App. We first began by placing our files into their proper folder locations. We created a controllers, models, views, and a public folder; within the views, each of our pages that were in HTML format were converted into EJS format. Additionally, we added a few other EJS view pages such as error, layout, and index; while the index page is the first page that is rendered when the app is launched and gives the user the option to login or signup, the layout page contains the elements that are included on multiple pages. For our app, the layout includes the header, navigation bar, and footer but in some pages, some of these elements are hidden. For example, the navigation bar is hidden on the signup page since realistically, a user should not be able to navigate through the site if they have not created an account. The error page outputs an error message when the application crashes or faces a similar error. Our public folder consists of all of the css documents, javascript documents, and images. There were not a lot of changes here except that we needed to change the pathing of the ejs folders to their corresponding images and css and js scripts with their new locations. The models folder contains a single file (user.js) which is the user schema that contains all the information needed for a user to create an account. This file specifies the type of answer the user has, the expected input and other important notes such as whether it is required and the maxlength of certain fields. The last folder is the controllers folder which contains the usersController, homeController, and errorController. The usersController deals mainly with outputting error messages when fields are not filled properly and rendering to the correct pages from signup and signin. The homeController page deals with additional page rendering when certain functions are called from main.js; for example, when the showIndex function takes the user to the index page and this first

occurs when the app is launched. The errorController considers the different types of errors a user may face (404, 500) and outputs a proper message depending on the error; this looks specifically at generic and specific errors. Finally, in the main part of our app, we have main.js and seed.js. Main.js connects the database, sets the port, and is used to navigate through the pages. It uses different get and post functions that make it easy for the user to complete both the login and signup forms and to check the entered form inputs to determine whether they meet the necessary requirements. The seed.js contains a few dummy users to test the database with.

Work Division and Future Plans:

In terms of how the workload was split, Jake worked on setting up the controllers and database system while Ayesha worked on the EJS conversion and fixing the issues that were made from the conversion. We worked together on building the user.js file and tackling any responsivity issues. Jake also used the bcrypt library to hash and salt the passwords for the seed.js users and worked to make the header option in the layout.ejs file something that could be changed with each page, while Ayesha wrote the report. For our future plans, we would like to make the home page a lot more user interactive. As of now, the users listed for the followers page are not real pages and the 'follow' button is strictly there for styling. We would like to make those buttons render actual actions and for the search bar to be usable as well. We also want to give the user the option to make posts and create an authentication method for users using a token. For this assignment, a lot of our work was focused on making the signup and login pages more authenticated and act like true social media pages. For the future, we'd like to do the same with the home page and use CRUD operations to help us with this task.

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 "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 :)