**NXTFolio Refactor/Bug To-Do List:**

**Why does this need to be done?**

While refactoring code might seem like a waste of time since new features are not being added, it is a crucial part of software development, especially with legacy projects. This project dates back as early as 2017, making it quite old with respect to a typical class project. There are obvious sign of “patch work” littered throughout the application. These quick fixes make the code hard to read and ultimately causes the time needed to debug and add new features to become elongated. This process of writing bad code on top of other bad code causes future dev teams to take longer to pick up where previous dev teams have left off. This time of this transition process grows for each subsequent team.

We believe taking the time to refactor the code base properly will ultimately result in quicker development for our team and future dev teams as well. Below are a list of the most glaring parts of the code base which need to be addressed. These tasks range from tidying up code to design changes within the application. The ultimate goal is to make the code base as clean as possible and to do things the proper way so that when other teams pick up where we left off they should easily be able to familiarize themselves with the code and easily deploy it.

**Github Root Directory Files:**

* Need to inspect files in Github root directory and delete/move files. There are files such as keyfile, keyfile.pub, fashion.iml, database.yml, background\_creative.jpg that either need to be deleted or if they are needed by the application, need to be moved to better places in the application. Doesn’t make sense to have random pictures/authorization keys in the top directory.
* **Modify Readme:** Need to remove all the old instructions that no longer apply to how we setup the application.

**Login Functionality:**

* **Login logic:** Currently the login logic is very messy, and the password and email verification process is messy. When a user’s password format is incorrect it is very vague as to what is wrong.
* **Password Storage:** Passwords are currently stored in plain text which is a big security risk. If this app is to be open to public use we must ensure that passwords are not stored in plain text.
* **Third Party Auth:** Not sure if last team ever got google/facebook authentication working. If this is something still wanted, then we need to set this up with the proper API keys.
* **Email Input:** Weird that the general\_infos page asks for your email again. Not sure what will happen if you put a different email than when we signed up.
* In my (Lance) opinion we should refactor a large portion of the login process.

**Image Storage:**

* I thought the last team had the application working with S3 Bucket storage, but after reading the document closely, it says they never got it working. In fact, they said they had trouble with the app crashing after 24 to 48 hours of being deployed. I suspect this is because after Heroku would wipe the images contained in the tmp folder the app would crash.
* Need to configure app to work with Amazon S3 bucket and it needs to work with profile image uploads as well as gallery uploads.

**Views Refactoring:**

* **Unnecessary Files:** Some views seem to have a large amount of .html.erb files associated with them (See general info view). This makes it difficult to digest the code and understand where the actual code being used is. We should move these redundant files to a folder called something like “trash” to make sure we still have the files in case we delete something important.
* **Utilize Partials:** Need to make sure to utilize shared html partials as much as possible to keep views from being bloated. The show\_profile.html.erb view has the code for the navigation bar in the actual file rather than rendering the exact code used in shared/\_header.html.erb. This occurs in a few different places with the footer code as well.
* **Commented Out Code:** Throughout some of the views files there are large amounts of commented out code. We should probably delete this code. Some of that code is likely very old and if it’s not being utilized now then obviously it is not important and can be deleted.

**Controllers and Routing:**

* **White Page Routes:** When I try to go to certain routes I am displayed with white pages. I noticed this for the <http://nxt-folio.herokuapp.com/specific_model_list>, <http://nxt-folio.herokuapp.com/specific_photographer_list>, and <http://nxt-folio.herokuapp.com/specific_designer_list>.There are probably other routes that currently are not displaying anything as well. We need to either fix these controllers/views to work properly or we need to display 401, 403, 404, etc. pages.
* **Make Admin Feature:** On the general info controller page there is an option to make a user and admin. This only works if the user logged in is currently an admin. This seems like a weird feature to expose to every user. When I click on it, it just redirects me to the homepage. We should probably just flash them a message like “Can only make other users admin if you are an admin”. Not a big deal, just seems weird.

**Search Profile Page:**

* **Search Profile Interface:** I think this is already being worked on, but when scrolling down in the search page the interface becomes super glitchy. The whole flow of the page seems weird. Why do we have two different boxes for search and advanced search. Seems like they should be the same box with a toggled “Advanced Search” option. The footer on that page is also weirdly displayed on my browser (Firefox). The header also starts has bugs when I am on the search page such as the notifications and messages graphics disappear.
* **A lot of routes**: This might be for good reason for this, but there are a lot of routes for the search\_profile functionality. We need to go into these and make sure these are all needed and when they actually get used. When I am searching for users who are specific models/photographers/etc. It seems like we should just be able to query on the general\_infos table with the condition that the user is a model/photographer/etc. This might be the functionality that is wanted, but it seems like there is a cleaner method of condensing all these search types.

Text

Description automatically generated

**Gallery Page:**

* **Double Add Review Buttons:**

Text

Description automatically generated

* **View Picture Not Working:** This might be a bug which is caused by the apps inability to store images, but when “View Image” is selected when viewing a specific gallery, it does not work.

**Landing Page:**

* **Area Below Login/SignUp**: I believe there are supposed to be galleries that show up on the landing page. This did not start working until I viewed and rated someone’s gallery, then it showed up on the home page. We need to talk to Tito to discuss what the functionality of this portion of the page should be.
* **How it Works Link:** How it works link doesn’t work, just redirects to home page.

**Register Business:**

* **Business Same as User?:** When going to sign up a business after typing in the email and passwords the user is directed to the User sign up. If they are trying to register a business I think there should be a separate form to fill out. Additionally, if you sign up a business, should you be able to login as the business? This entire feature needs to be rethought.

**TESTING:**

* The most important part missing from the entire application is test code. Without tests we are blind to what changes break features within the application. The previous teams have attempted to write tests, but they are insufficient or nonexistent.