CSCE606 Final Report

Project Members:

Saikrishna Chandhrasekhar
Sai Tejas Janjur
Jubey Garza
Ziyu Wang
Ruvail Shahzad
Christopher Muniz (Dropped)

Links:

Deployed app: https://nxtfolio.herokuapp.com/
Github: https://github.com/FashioNXT/NXTFolio
Pivotal Tracker: https://www.pivotaltracker.com/n/projects/2635646

Project Summary:

Our project is NXTFolio, a content-sharing, job-hiring, and advertising all-in-one social media platform for art and fashion creatives. The main stakeholder is FashioNXT. The client intends for the platform to be a social media app where artists can connect with each other and advertise themselves for work opportunities, similar to websites such as LinkedIn and Indeed. Through NXTFolio, a creator can establish and share their portfolio on anything arts and fashion related, showcasing their work to the world and advertising their talent. A client can use the website to find creators that meet their needs and establish contact for work-related inquiries.

This is a legacy project that went through many iterations before we took over. The client wanted us to focus on mobile view, CRM, recommendation system, and notifications for this iteration. Now, users can tag their portfolio galleries to receive personalized recommendations based on similarities to other portfolio galleries.

User Stories:

- **Sai Tejas Janjur**
    - Mobile View - Implemented code to ensure that the website's home page and sign-in/signup pages are fully optimized for mobile view. This involved adjusting CSS and HTML elements to adapt to different screen sizes, enhancing usability on smartphones and tablets.
    - The modifications include reconfiguring layout elements to stack vertically under smaller screen resolutions and ensuring that all interactive elements, such as buttons and form fields, are accessible and usable on touch screens. This story was given 4 points in sprint 1 and 2 points in spring 2 as it involved figuring out how the existing code worked and then changing it to fit in mobile view.

- Bottom Nav Bar - I developed a bottom nav bar that appears when the website is accessed via a mobile device. This nav bar allows users to easily switch between key pages of the application without having to revert to the main menu. The feature was given 2 points because it was relatively easy to implement in terms of code, as I had created the mobile view and was familiar with the code.
- Bug Fixes in Cucumber scripts - I worked on fixing bugs in the cucumber script and increasing the cucumber coverages. This feature was given 2 points as it involved debugging why the Selenium gem was failing, and adding code for new coverages was straightforward.

- **Sai Chandhrasekhar**
  - Fix login and signup issue: The views were error-prone because the application did not check whether a user was logged in before accessing instance variables. I assigned this story 2 points because it took time to understand and fix the problem.
  - Save user information in the edit profile feature: In the edit profile page for users, the user's country and state were not saved, so they had to re-enter them. This was inconvenient, so the view was changed to save the values. I assigned this story 2 points since it was moderately difficult to figure out.
  - Display travel information country: Users can add information about their destination when they travel. The previous teams forgot to display the destination country on their user profiles. I assigned this story 1 point since it was very easy to fix.
  - Improved RSpec coverage: The existing RSpec coverage was 28%. In order to meet the requirements, I worked on improving it to 90%, which I was able to do. I assigned this story 5 points since it took a long time to read through all the code for the models and controllers and write unit tests for them.
  - Create a tagging feature for galleries: The existing recommendation system showed every user the highest-rated galleries. To make the recommendations more personal, I implemented a feature allowing users to add tags to their galleries so they could receive recommendations based on their tags. I assigned this story 2 points since adding a new table to the Rails schema and properly integrating it with the existing tables was moderately difficult.
  - Improve the recommendation system: As described in the previous story, I changed the recommendation system to give personalized recommendations. The recommendation algorithm computes the Jaccard similarity score: https://en.wikipedia.org/wiki/Jaccard_index between a user's galleries and each of the other galleries in the system. The galleries are then displayed in order of most to least similar. This feature was completed. I assigned it 3 points since it took 2-3 days to implement the algorithm and test it.

- Email notifications for job postings: I set up a Rails mailer to email users when jobs they might be interested in are posted. I assigned this story 1 point since it was simple to implement.
- Email notifications for gallery ratings: I set up a Rails mailer to email users when one of their galleries is rated. Since it was simple to implement, I assigned this story 1 point.

- **Jubey Garza**
  - Change current navigation bar from dynamic to universal: The purpose of this was to make navigation less confusing. This change ended up not being made because it was found that a large amount of code was stored within the HTML, and thus, the change would be both non-trivial and problematic. This story was given 2 points as it was expected to be simple.
  - Re-discover the FashioNXT-CRM API feature for NXTfolio: There was no documentation for how this feature worked or how to use it. I rediscovered how to use this API and use it in the CRM tool. I added this information to the documentation for both NXTFolio and the CRM tool. This story has been completed. This story was given 3 points as it did not involve coding but was expected to take some time to figure out how the CRM tool worked and how the NXTfolio API functioned.
  - Add the "enabled" attribute to the user_login table: This allows a user's account to be either "enabled" or "disabled," which may prevent them from logging in. This change also needed to be made so that existing entries would be updated rather than just missing an attribute. This story has been completed. It was given 1 point as it was both simple and not expected to take long.
  - Add an enable and disable user feature to the FashioNXT-CRM tool user management page: This addition allows the CRM tool to enable or disable a specific user's account by calling an API specific to the web application it is currently viewing the users of. This story has been completed. It was given 1 point as it was simple to implement.
  - Add the "toggle_user_enabled" feature to the NXTfolio API: This API feature directs the app to enable or disable a specified user when the API is called with specific arguments. This story has been completed. It was given 1 point as it was simple to implement.
  - Add the delete user feature to the FashioNXT-CRM tool user management page: This addition allows the CRM tool to delete a specific user's account by calling an API specific to the web application it is currently viewing the users of. This story has been completed. It was given 1 point as it was simple to implement.
  - Add the "delete_user" feature to the NXTfolio API: This API feature directs the app to delete a specified user when the API is called with specific arguments. This story has been completed. This story was given 2 points because the database was

not set up to perform cascading deletes of user-related entries, making the task slightly tedious.
    - Bug Fixes in Cucumber scripts - Worked on fixing bugs in the cucumber script and also increasing the cucumber coverages. This feature was given 2 points as it involved debugging why the Selenium gem was failing, and adding code for new coverages was straightforward. This story was worked on with Tejas

- **Ziyu Wang**
    - Gallery example fix(1 pt) - Populate the test and release database with examples so the main page gallery is populated and presentable, 1 point for simplicity
    - Easier subcategory navigation(1 pt) - This story addresses the cumbersome ways for a user to navigate subcategories above the gallery, 1 point for simplicity
    - Make the home page gallery scroll with the page not in its scroll box(2 pt) - The gallery originally had its own scroll box on the main page along with the site scroll box, making navigation awkward and unintuitive. This story fixes that issue, assigning 2 points for the relatively simple coding involved
    - Legacy code refactoring (4 pt) - A semester-long, collaborative effort to bring the existing legacy code up to a new standard so it is more maintainable, assigned 4 points due to its large workload, necessary manpower, and degree of difficulty
    - Rspec coverage test improvement(4 pt) - Another collaborative effort between multiple team members to improve the Rspec test coverage for both legacy code and new features, assigned 4 points similar to legacy code refactoring
    - Notification(2 pt) - A simple notification system for when users interact with other users, held off for a long time due to prioritizing legacy code and test coverage, 2 points were assigned as the task was relatively simple
    - Cucumber test coverage improvement(2 pt) - Another collaborative effort between multiple team members to debug the cucumber tests and improve the coverage similar to Rspec, 2 points were assigned for the straightforward debugging process

Legacy Code:

To understand the existing code, we looked through it to understand what each file did and contacted the previous team to ask questions. We refactored the code to fix errors and improve the user experience. For example, several pages gave errors since there were no checks to ensure the user was logged in, which caused nil variable values. There were also visual issues, such as the default profile photo covering the follow button. The notification and DM icons were being implemented using old CSS files that no longer worked. Finally, there were inconveniences, such as the edit profile page not saving user location data and making them enter it again.

Team Roles:

Saikrishna Chandhrasekhar	Sai Tejas Janjur	Jubey Garza	Ziyu Wang	Ruvail Shahzad	Christopher Muniz
Sprint 1 SM Sprint 3 PM Sprint 2 & Sprint 4 Developer	Sprint 1 PM Sprint 3 SM Sprint 2 & Sprint 4 Developer	Sprint 2 PM Sprint 4 SM Sprint 1 & Sprint 3 Developer	Sprint 2 SM Sprint 4 PM Sprint 1 & Sprint 3 Developer		

Sprint Summaries:

- Sprint 1: We worked on understanding the legacy code and identifying bugs in the application.
- Sprint 2: We began working on new features such as CRM, a mobile view, and an improved recommendation system. Meanwhile, we continued working on improving test coverage and fixing bugs in the application.
- Sprint 3: We continued working on our features. We were able to integrate the CRM with NXTFolio, added a CRM feature to enable and disable users' accounts, successfully implemented the mobile view for the homepage, and implemented a tagging feature for galleries to aid with personalized recommendations. We also successfully improved RSpec coverage to 90%.
- Sprint 4: We finished the CRM and recommendation system features. Admins can delete users from the CRM portal. Users now receive personalized recommendations based on the tags they gave their galleries. We also worked on fixing bugs and improving the website's visual quality by using more aesthetically pleasing fonts.

Team Member Contributions:

Iteration	Sai Chandhrasekhar		Sai Tejas Janjur		Jubey Garza		Ziyu Wang		Ruvail Shahzad	
	Stories	Points	Stories	Points	Stories	Points	Stories	Points	Stories	Points
1	1	2	0	0	1	2	2	3	0	0
2	3	8	2	4	1	3	1	4	0	0
3	2	7	2	4	5	6	1	4	0	0
4	3	5	1	3	1	2	1	2	0	0
Total	**9**	**22**	**5**	**11**	**9**	**13**	**5**	**13**	**0**	**0**

Customer Meetings:

During our weekly meetings, we demonstrated whatever new work we had finished to our client and received feedback from him that would be iterated on. During these meetings, our client also directed us on what work they wanted to be done next.

February

- 6th - The client described what NXTFolio was and told us to look at other applications with similar functionality to see what features we could implement.
- 9th—The client received our feedback about the other websites and told us to get the app running on our local machines to test it and fix bugs, such as users' inability to sign up, sign in, and view the website.
- 13th - We showed the client our bug fixes, and he informed us of the features he wanted us to implement. We then assigned the features among ourselves-mobile view for Tejas, recommendation system for Chandhrasekhar, notifications for Ziyu, and CRM integration for Jubey.
- 20th - Most people didn’t show up, so we rescheduled the meeting for the 23rd.

- 23rd—We reported our progress to the client. Since RSpec and Cucumber coverage was very low, we decided to halt some of the feature work until we could improve coverage and maintainability.
- 27th - The client noticed bugs in NXTFolio and wanted us to fix them. The “My Profile” button and icons for notifications and DMs were not displaying properly on many pages. There were also some CSS issues, such as the follow button on user profiles being covered by the default profile picture.

March

- 5th - We reported our work to the client. Tejas reported his work on the mobile view, Chandhrasekhar reported his work on fixing bugs, and Jubey demoed the CRM for the client.
- 19th - Chandhrasekhar reported his work improving RSpec coverage, and Tejas reported his work on the mobile view.
- 22nd - Chandhrasekhar was told to go back to working on the recommendation system. Tejas reported his progress on the mobile view. Jubey was given the task of allowing admins to enable, disable, or delete users from the CRM app.
- 26th - Chandhrasekhar reported his plan for the recommendation system.

April

- 2nd - Most people didn’t show up, so we rescheduled the meeting for the 5th.
- 5th - Ziyu and Tejas took on the role of increasing Cucumber coverage.
- 9th - Chandhrasekhar reported his progress on the recommendation system. Jubey demoed the CRM features he had been working on and also took on the task of increasing Cucumber coverage.
- 12th - Chandhrasekhar demoed the improved recommendation system and was also assigned some new tasks, such as sending email notifications to users for gallery ratings and new job postings.
- 16th - Jubey and Tejas reported their progress on Cucumber, and Chandhrasekhar reported completing his assigned tasks.
- 22nd - The client assigned some finishing touches to Chandhrasekhar, such as improving the visual quality of the website.

BDD/TDD:

We used RSpec for TDD and Cucumber, Capybara, and Selenium WebDriver for BDD. Engaging in BDD and TDD improved our code by making us think about what our code ought to do before writing it and by testing whether it worked as expected. RSpec did not give us any

issues. We had trouble getting Cucumber to run but solved the issues by upgrading to newer Cucumber and Selenium WebDriver versions.

Configuration Management:

For each sprint, we independently develop and test feature branches. Once they are satisfactory, we create pull requests to merge them into the master branch for final quality assurance and release.

Production Release Issues:

As a legacy project, our app had a preexisting deployment and automated deployment set up for the GitHub master branch. We only encountered two issues with this. In the first sprint, we had some issues with accidentally deploying to master instead of a separate branch and then creating a pull request. This led to some bugs in the master branch that had to be resolved. The second issue came during deployment verification after sprint 4. When another team tried to deploy our app, we realized there actually were not any instructions on the readme about how to do this, only information that the app was set up to deploy using AWS. This was resolved quickly, and proper steps were added to set up a deployment manually.

Tools/Gems:

We used GitHub for repo management and sharing; we took over the project repo from the previous developers. We used CodeClimate alongside tools such as Simplecov and Rubocop for code coverage and maintainability rating. These tools are great for legacy code refactoring and for writing Rspec and Cucumber tests. For each sprint, we used PivotalTracker to keep track of our progress on user stories. The project is developed using Ruby on Rails. To set up and test the project locally, we used Docker to set up containers, as it is very fast and convenient. For project deployment, we used Heroku and AWS. Several APIs, such as Google Geocoding and OpenAI, are used for some features.

Repo Content:

Codebase and Directory Structure -

- The repo includes folders like ‘.github/workflows’ for CI/CD processes using GitHub Actions. The ‘app’ folder contains the application logic. The ‘bin’ holds executable scripts, and ‘config’ contains files for the Rails application environment.
- Specific directories like ‘db’ for database migrations and schema, ‘documentation’ folders contain all the reports written by our team and the previous teams that have worked on this project, and ‘features’ for development files with cucumber.

Development and Development Tools -

- The repository integrates several tools for development and testing. Docker is used to containerize the application, ensuring consistency across various environments. Ruby on Rails is the primary framework, supported by PostgreSQL for database management.
- Continuous development is handled via ‘travis.ylm’ and GitHub Actions within ‘.github/actions,’ ensuring that each commit and pull request is automatically tested and deployed.

Key Files and Scripts -

- Files like ‘gemfile’ and ‘gemfile.lock’ dictate the Ruby gem dependencies required for the project.
- Script files like ‘setup.sh’ in the install folder facilitate setting up the development environment, especially on different environments including configuration for docker.

Presentation/Demo Videos

https://www.youtube.com/watch?v=Ksh4hbkm5rI