Cookify: The Social Cooking App

CSCI 3308 Final Project: Group 014-07

by: Ryan Greer (@doublergreer), Lili Hailemariam (@liliHmariam), Vladimir Hristovski (@Vladimir6812), Alex Reid (@AlexReid144), Hannah Yoon (@hannahyoon0105)

Project Description

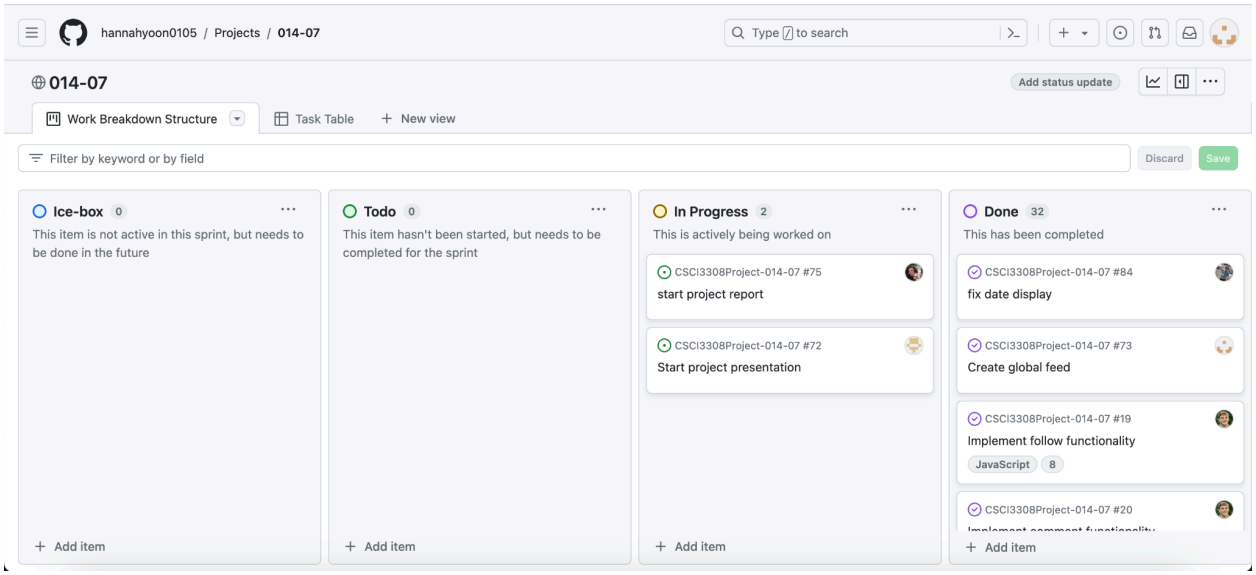
Cookify is a social platform that provides an all-in-one social cooking experience. Cookify aims to centralize the online cooking space; currently, recipes are hosted on various third-party websites, with varying credibility. Culinary influencers don't have a central place to post content, and usually link their recipes to third-party sites. This prevents the development of a platform! Cookify is trying to create the standard for culinary social media!

The app provides familiar social media functionality such as posting, following/unfollowing, and liking/commenting on posts. Users can register and view all posts that users have made on the platform. They can also create their own posts of original recipes and repost other users' recipes!

The key features that Cookify offers are the reposting feature and the recipe view. When a user posts an original recipe, other users can recreate it. Each repost will be tied to the original recipe, meaning that original creator will have increased visibility and potential of going viral! Further, recipes are viewed within the app. This allows users to host their recipes on a central platform that viewers can trust. This provides a comprehensive experience that centralizes the digital culinary landscape, empowering both creators and enthusiasts alike.

Project Tracker

Link to project tracker: https://github.com/users/hannahyoon0105/projects/1



Video

Link: Video Link

VCS

Link to GitHub repository: Github Repository

Contributions

Ryan Greer:

- Created recipe and user page frontend (HTML, Handlebars)
- Implemented user and recipe GET endpoints and functionality (NodeJS, PostgreSQL)
- Improved date formatting across platform, including post date, comment dates, and recipe post dates (NodeJS, HTML, Handlebars)
- Improved styling across global/home page (CSS, HTML)
- Contributed to like count implementation in home and global feed (PostgreSQL, Handlebars)
- Recorded and committed minutes for all TA meetings
- Contributed to project presentation

Lili Hailemariam:

- Created message, title, and footer partials (HTML, Handlebars)
- Contributed to project presentation

Vladimir Hristovski:

- Created insert statements to create test data (PostgreSQL)
- Worked on implementing functionalities such as like and post functionality (JS)
- Contributed to the presentation
- Contributed to defining the minimal acceptance criteria for users
- Improved look of profile view, to include user posts in a single container (NodeJS, PostgreSQL, HTML, Handlebars)

Alex Reid:

- Worked on the front end layout of the home page (HTML/CSS)
- Implemented the functionality of the like and comment buttons on home page and user page so the data was properly updated or removed from the database and the displayed data accurately reflected what the user wanted to see (NodeJS, ExpressJS, Handlebars, PostgreSQL)

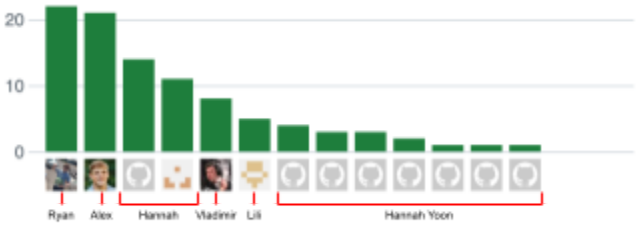
- Implemented following functionality for users on both the Account page and Home page so users could successfully follow or unfollow other accounts and the database reflected those changes (NodeJS, Handlebars, HTML)
- Created the login and logout functionality (HTML, Handlebars, NodeJS)

Hannah Yoon:

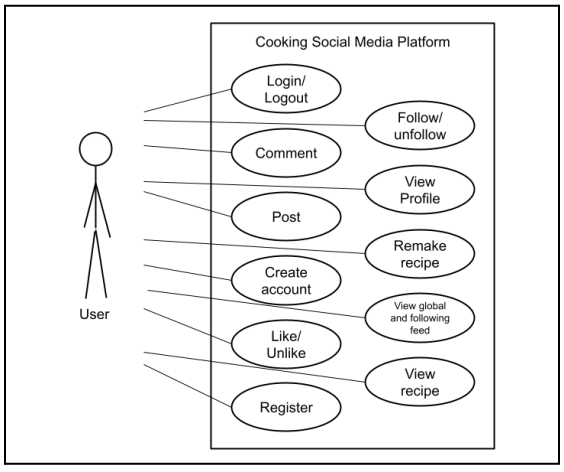
- Initialized database and managed containers (Docker, PostgreSQL)
- Created navigation bar and implemented user session management (HTML, Handlebars, NodeJS)
- Developed create post and repost UI and functionality (HTML, Handlebars, NodeJS, PostgreSQL)
- Improved aesthetics and user friendliness of home page, profile view, and recipe page. This included creating a heart button for liking posts, implementing scrollable comment section, improving button visuals, making username on posts clickable, making recipes and on recipe page clickable. (HTML, Bootstrap, Handlebars)
- Implemented like count on posts, followers/following counts on user page, and like/repost count on recipes (NodeJS, PostgreSQL)
- Created global feed (HTML, NodeJS, Handlebars)
- Formatted and stylized all presentation slides

The team's commits are described below:

Excluding merges, **13 authors** have pushed **93 commits** to main and **96 commits** to all branches. On main, **39 files** have changed and there have been **7,135 additions** and **8 deletions**.

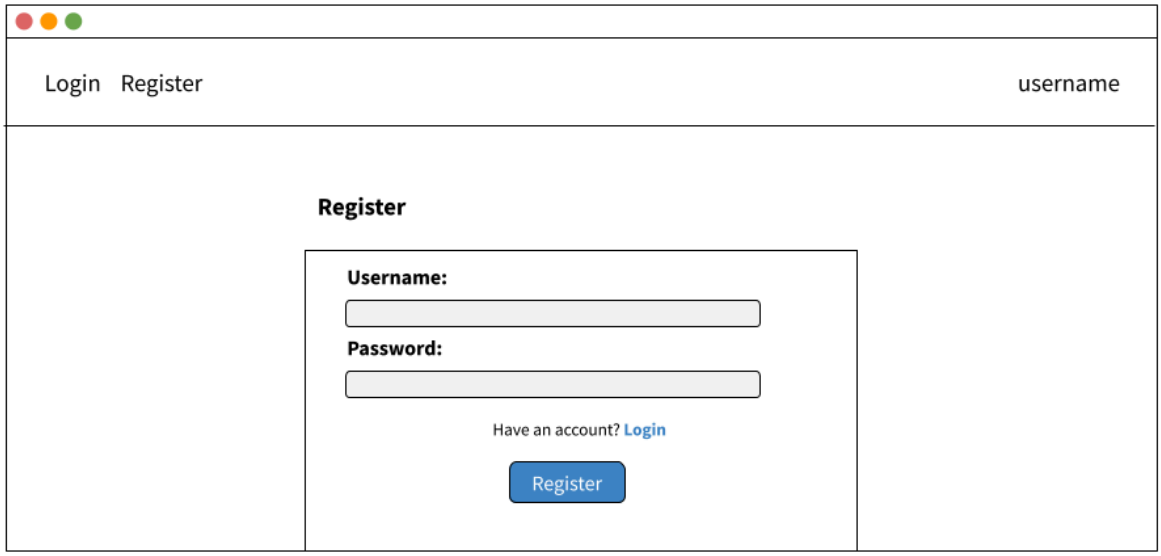


Use Case Diagram

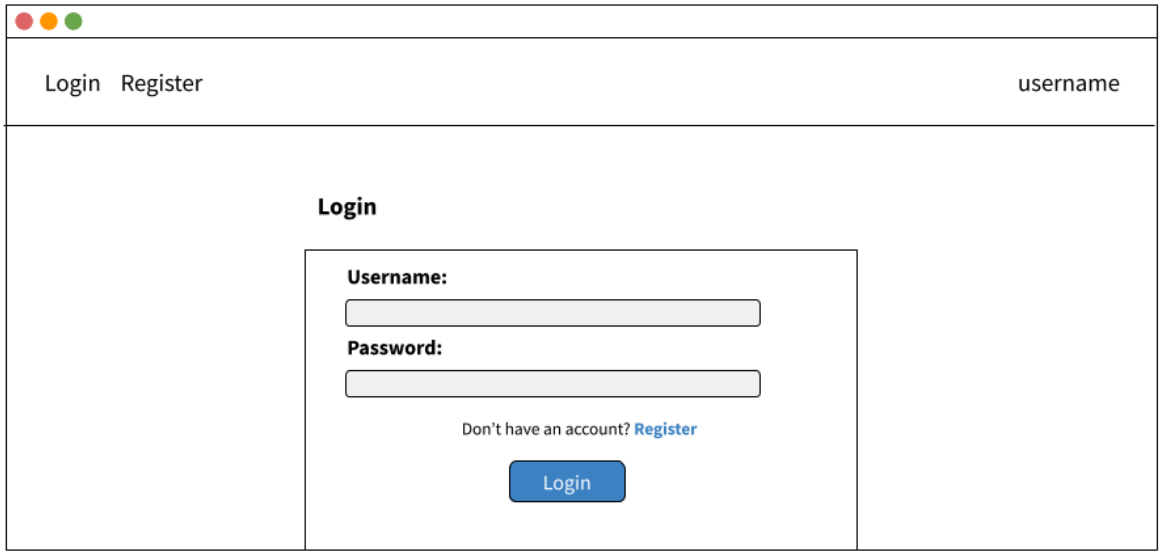


Wireframes

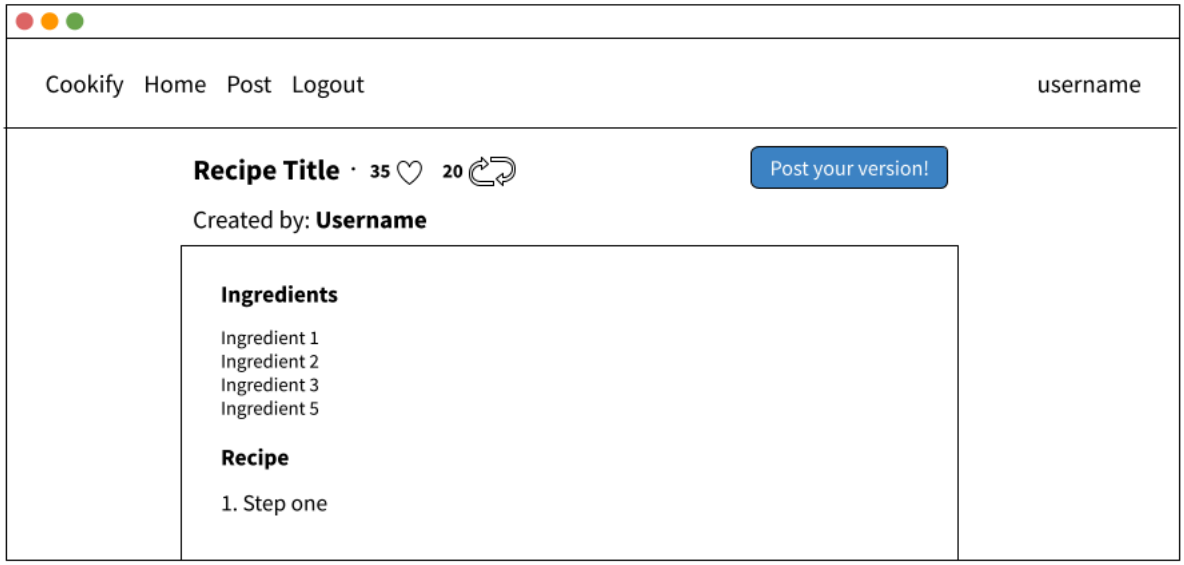
Register page:



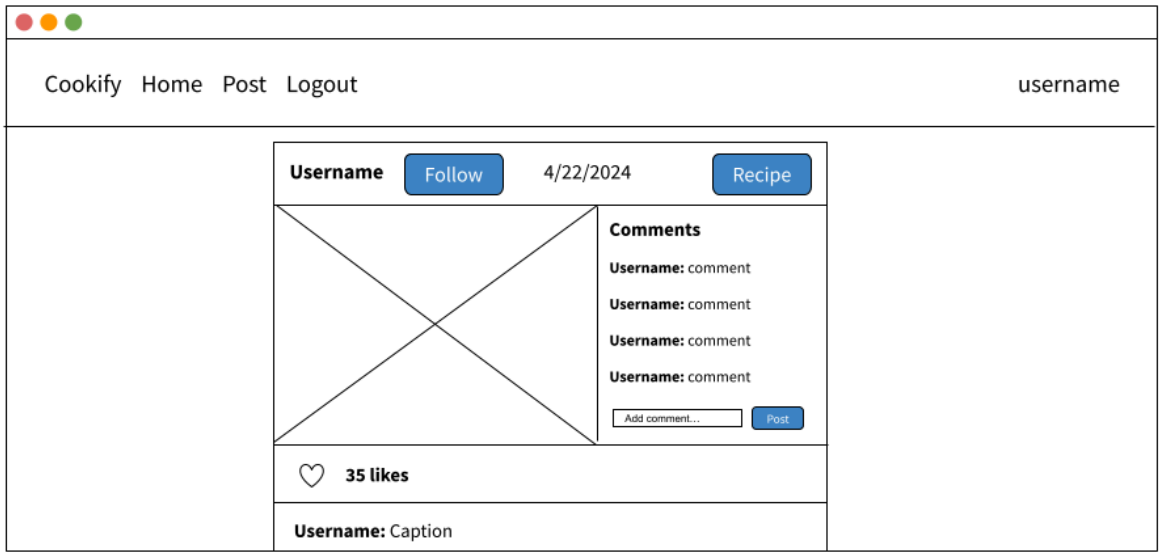
Login page:



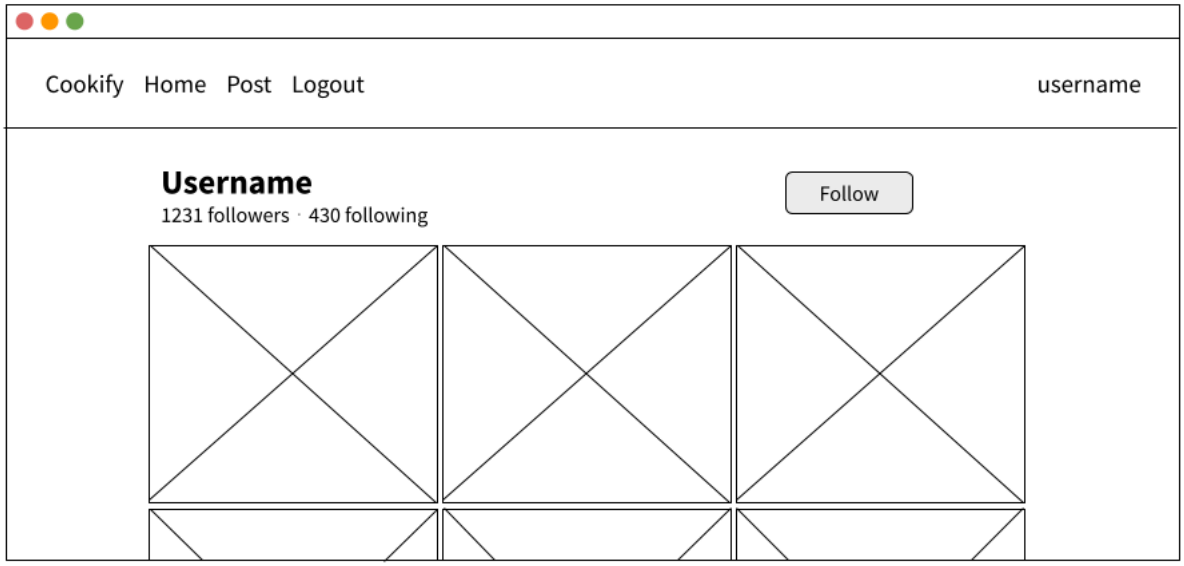
Recipe view:



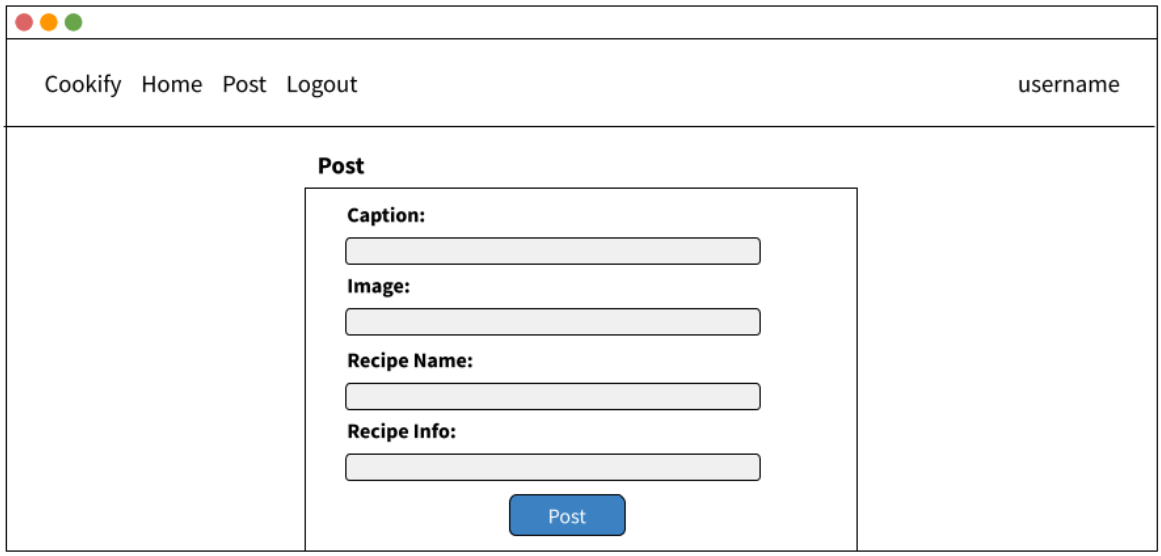
Main feed:



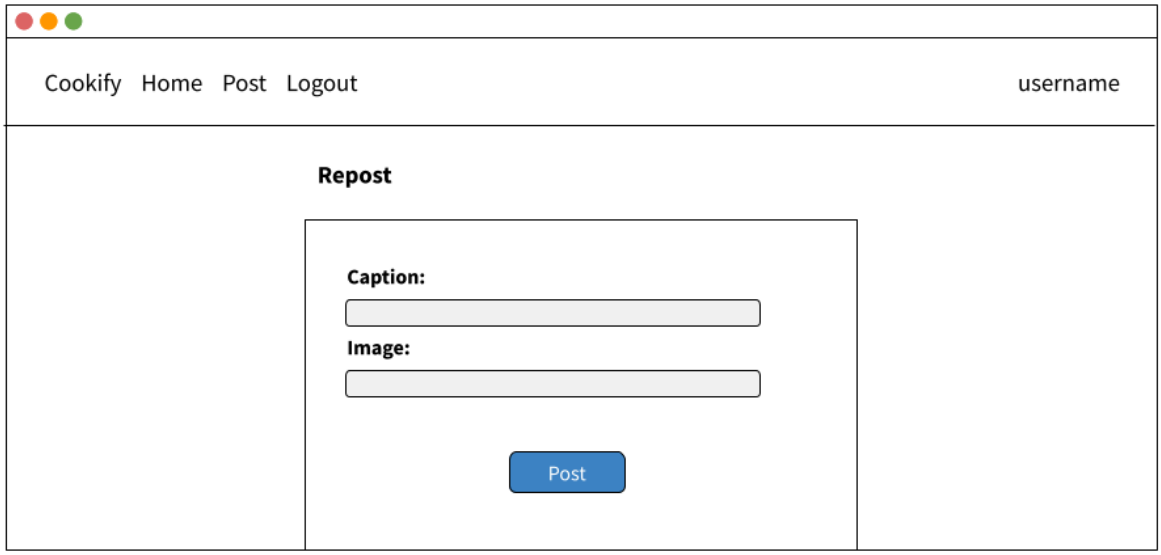
User view:



Post:



Repost:


Cookify Home Post Logout
username
Repost
Caption:
Image:
Post

Test Results

We asked a non-CS student peer to test our application. The following use cases were tested with their respective observations:

Use case 1: Login

Description: The user should be able to login with correct credentials. User login will fail if the user provides incorrect username or password. The form provides the user with specific feedback about the error.

Observations: Test user behavior is consistent with the use case. Created a login, and used the same credentials to login to the application. No deviation from expected actions, thus no changes were made to the product.

Use case 2: Like Post

Description: User should be able to press the like button to like an unliked post. User should be able to press the like button of a liked post to unlike the post.

Observations: Test user behavior is consistent with the use case. User scrolled the feed, found a post they enjoyed and pressed the like button to like the post. User couldn't notice the difference between the like and unlike button, which directed us to change the like button. At the time of testing, the like button was just a button with the word "Like" or "Unlike" based on the state of the post. The like button was changed to a heart, where a filled in heart indicates a liked post, and an empty heart shape indicates a post that has yet to be liked.

Use case 3: Create Post

Description: User should be able to navigate to the post page through the navigation bar. User should be able to input the details of their post and create a post. The user should be able to see the post populated on the Cookify global page.

Observations: After being prompted to create a post, the test user navigated to the post button in the navigation bar with ease. The button directed them to the post UI, which they filled out. They were confused about what the difference was between 'post content' and 'recipe content.' This caused us to separate the post UI to two sections, post information and recipe information. We changed the verbiage to be more familiar to social media users, by saying "caption" instead of "post content." After pressing the 'Submit' button, the user saw their post populate on the global Cookify page!

Use case 4: View user/follow/unfollow

Description: User should be able to click on users from their feed and view their profile and associated posts. User should be able to follow the user on the feed and also in the profile view.

Observations: The user scrolled through their feed and found a user that they wanted to explore further. They tried to click on the user's handle on the post caption, not at the top of the post card as we expected. This was due to familiarity from TikTok/Instagram for navigating to user profiles in this way. This caused us to make the username in the caption also clickable. After being redirected to click on the user at the top of the post card, the user scrolled the profile. They clicked the follow button, which correctly followed the user. They then clicked the follow button again (which was turned to the 'Following' button), which correctly unfollowed the user.

Deployment

Link: http://recitation-14-team-07.eastus.cloudapp.azure.com:3000/home

To access the app, navigate to the link above. If you would like to use a test login, you can use the following credentials:

User: user1
Password: password123

Otherwise, you can receive the same functionality by registering a new user. The Home page will initially be empty due to having no users that are followed, but you can follow users on the global Cookify page to populate the Home page.