Title of the Project

Neighborly Connect

Team Members

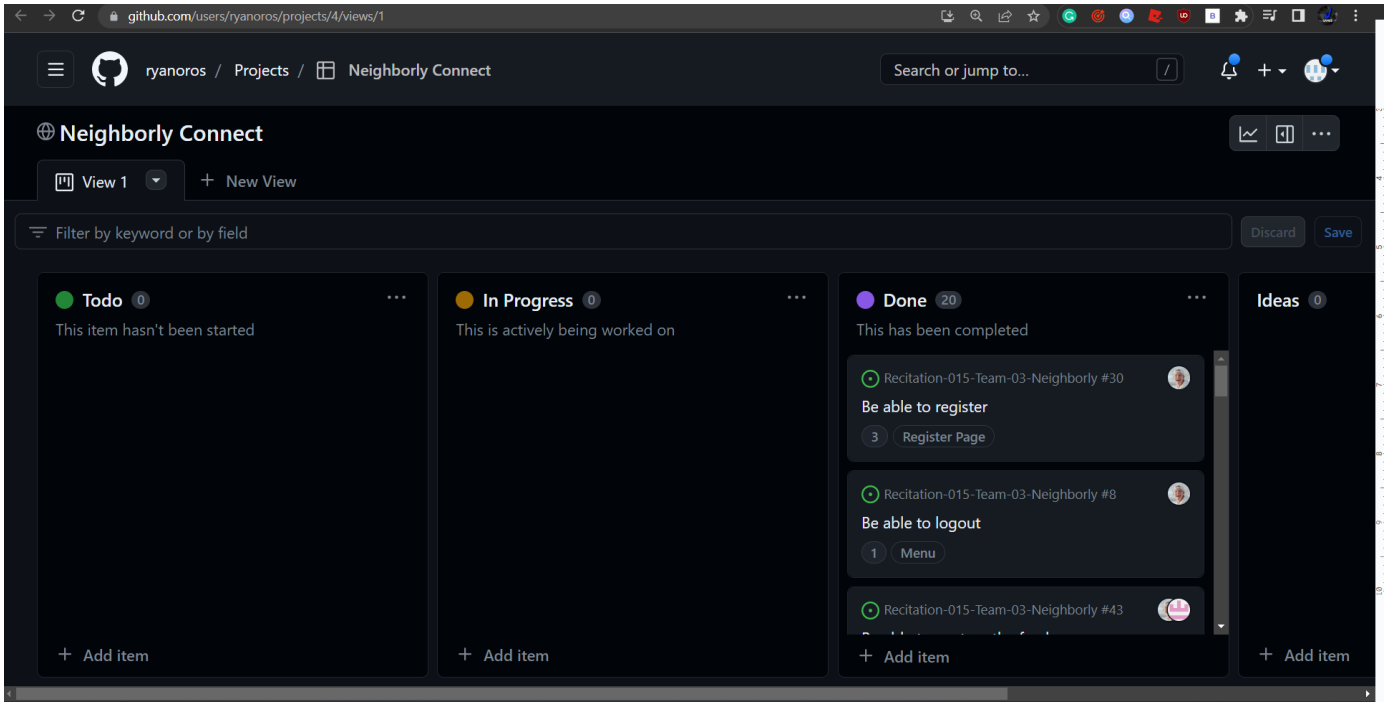
Name	Github	Email
William Johnson	wijo9385	wijo9385@colorado.edu
James Vu	JamesVu34	javu5882@colorado.edu
Ryan Oros	ryanoros	ryor7056@colorado.edu
Evan Lesnefsky	EvanLesnefsky	evle2740@colorado.edu
Warren Fu	warren-fu	wafu8567@colorado.edu
Juno Park	JunoPark-01	jupa3922@colorado.edu

Project Description

Neighborly Connect allows you to introduce yourself to your neighbors before making a final decision on moving. It also allows you to find real estate listings in your desired areas as well as find communities within the region that have the same interests/jobs as yourself. It will also allow you to review places you've stayed and make yourself available to roommates allowing potential tenants to apply and be reviewed by who currently lives there. Our goal at Neighborly Connect is to empower people to find their dream home in a community that shares their interests and occupations, by providing a seamless web app that fosters meaningful connections with potential roommates and neighbors.

Project Tracker

Link to GitHub Project Board: https://github.com/users/ryanoros/projects/4/views/1



Video

Link to Video:
https://drive.google.com/file/d/1X1MkirmgU--hghMfheoKgZvEdDSlsFMY/view?usp=sharing

VCS

Link to GitHub Repository: https://github.com/ryanoros/Recitation-015-Team-03-Neighborly

Contributions

Evan: Made several improvements to our web application. Enhanced the listing page by improving its appearance and added an "apply" button at the bottom for user convenience. The login form now displays a message upon incorrect password entry and prevents unauthorized access. The login and register pages were also updated for better visual appeal. User profiles have been made more versatile by adding majors and jobs. Created release notes with tags, added team meeting log notes, and designed a professional-looking presentation that shared with everyone. Submitted milestones and debugged the login/register features while also updating user story names.

Ryan: Wrote API calls for the feed page for adding posts, adding comments to posts, and liking/disliking posts. Wrote queries so only posts from your neighborhood are shown and added an option to filter only your posts. formatted the feed page using HTML, wrote test cases for

login and register, filled out slides for the project presentation, updated the project board and user stories, debugged and found errors in code, and created the logo for our brand.

James: Added interests table to database and made apis to get and post interests from the interests page. Also, I made the interests page UI. Helped with the presentation, updated the project board and fixed some of the user stories, and helped debug code.

Juno: Worked on fixing SQL queries in API calls, implementing the update address feature, and completing the process of saving addresses. Added feed deleting and editing functionalities for users to delete or update their own feeds. Re-designed main commits to fit the new main branch and updated the database. Debugged codes within profile API calls and queries, formatted the profile page base skeleton, added release notes code within the .git for release.yml, fixed database issues, reinitialized SERIAL call for primary key to avoid insert.sql conflicts, updated project board, and assisted with the presentation.

Will: Updated database with new features like API calls for neighborhood ID based on address, profile reformatting with interests section, integration of Mapbox API for pins on maps, and the ability to add pictures to listings. Improved queries with more efficient create.sql, debugging of api calls and queries, worked on feed page voting capabilities as well as commenting on posts, finding other api calls for neighborhood ID, user case diagrams, fixing merge conflicts, editing storyboards, and implementing options for pictures and questions in listings. Applications page showing listings you have applied to as well as applications that have applied to your listing. Listing application process choosing questions and the interests implementation of the profile page. Assisted with the presentation.

Warren: Worked on fixing the realty API, eventually led to the switch to the Zillow API. Used the Zillow API to find the listing on Zillow and linked that to the listing application page. Updated the old test cases to work with the new code and changed a lot of the functionality. Created a feature to update a listing after you created it. Helped update the storyboard and user stories, worked on the final presentation slideshow, debugging errors, fixing merge conflicts, Made the original database structure on draw.io.

Use Case Diagram

Test Results

The first test created an account using the Register POST API and it returned “Success”. The next test also tested the Register POST API but it had non matching passwords. This test should return "Passwords do not match." The third test tested the Login API with wrong credentials and it returned “User does not exist” and lastly, the final test sent in the credentials created by the first test and it should return “Success.”

Test Cases

The first use case was a friend of one our team members, who is our age and goes to CU Boulder. When the user first opened the application, they tried signing in because they most likely thought that there was placeholder data. They then registered and logged in. At first they were confused, but they tried adding a listing because they weren’t aware with the fact that the user has to update their profile before posting a listing. This behavior is consistent with the use cases.

As silly as it sounds, the second use case was a random person off of the CU Boulder 2025 story on Snapchat. We used this case because a lot of people on the story constantly post their listings either looking for a roommate or subleasing for a given amount of time. This use case is perfect for our application. When the user opened the application, they registered first and then logged in, which is not consistent with most cases because people believe that they can login first try. When they logged in, they went to their profile because they noticed that their were no listings. This is also not consistent with most cases because the profile is not visible until you click the drop down.

The third use case was another student in our CSCI 3308 recitation class. When the user opened the application, they registered and then logged in. This is not consistent with other cases because this student understands the process of creating an account first. They then looked at the map and saw listings from other users with the pins showing the locations. They then applied to a listing and went to their applications and feed. This user went through all of the features and tested everything to make sure it worked. This is not consistent with other use cases because this student understands where there could be flaws in an application.

The last use case was an adult who is not as acquainted with technology but is familiar with the housing market. This allowed us to see how applicable our website is in the real world. The user tried to login but failed, then realized they had to make an account. After logging in successfully, they were unaware of what to do until they clicked different tabs and eventually figured out that they had to update their profile. They then added a listing and viewed others to see how realistic the website is. At the end, the user logged out and was able to log back in as desired.

After these use cases, we updated the mapbox api and listings so that it displays listings based on the user’s local location rather than having to go to the profile and add an address. This makes it easier for users to see listings that are closer to them, but listings will be more accurate once they fill out the interests in the user profile.

Deployment

Link to Deployment Environment:
http://recitation-15-team-3.eastus.cloudapp.azure.com:3000/login