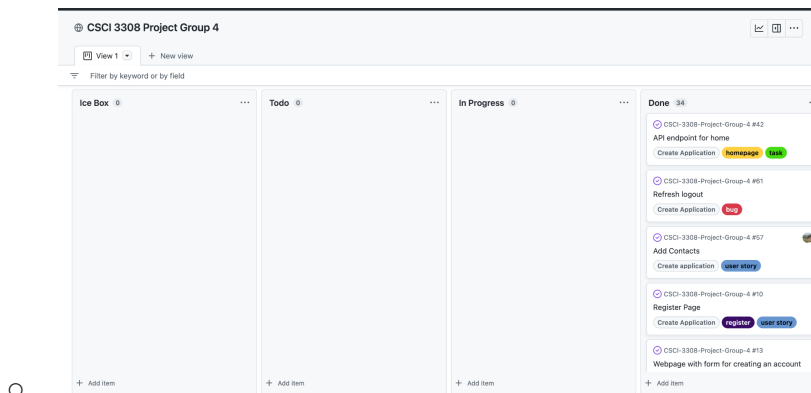


Title: Ping-pong

Who: Aiden Bub, Conner Parker, Cam Frederickson, Adrian Reghitto, LeeLee James

Project Description: Our project is a messaging application which has a variety of functionalities. When first visiting the application, users can register with an email, username, and a password. Once users have an account, they can log in with their username and password. Once logged in, there is a navigation bar on top that allows users to navigate to their home page, and also a place to log out. Upon log out, the application takes users back to the login screen. The home page lists the user's contacts, and allows them to add new contacts. It also lists the contacts with which the user has messaged. The user can choose to reply to a specific contact, bringing them to the message page. The message page shows a specific instance of a conversation with one of the user's contacts, with each message listed out with the text, who it is from, and what time it was sent. The user can type out a message and send it, and upon sending a message the page reloads with the new message included in the conversation.

- Project Tracker - GitHub project board:
 - <https://github.com/users/ConnerParker42/projects/3>



- Video: <https://youtu.be/yrThUB-T4YA>
- VCS: <https://github.com/ConnerParker42/CSCI-3308-Project-Group-4>

- Contributions:

- Aiden Bub:

For this project, I worked on a variety of things throughout. I contributed mostly on the front end by working on the messages page and creating the front end for individual person to person messaging and fixing errors that came up while doing so. I also worked on the home page and the login and logout pages on the front end side. I also contributed to the overall organization and management of the project. I made sure to keep communication clear and efficient between all group members and worked to maintain good channels of communication. Lastly, I worked with the backend to get the front end and back end working together smoothly.

- Conner Parker:

For the project I mainly worked on the backend with some frontend structure. Specifically, I worked on the API endpoints for the login, register, home, and messages page. These endpoints along with database structure was my main area of work. I also worked on the base message page but then handed that off to another contributor to polish. During our weekly meetings I tried to keep people up to date with the project board and make sure everyone came to the meetings to debrief everyone on their contributions.

- Cam Frederickson:

I primarily worked on the backend and database side of the application. I created the database structure, ER diagrams, example data, and helped write queries for the backend routes to interface with the database. I also implemented the logout and addContact routes, along with some backend error handling for user registration, login and adding contacts. In addition, I set up the structure for the project board, created the majority of the user stories and tasks, and maintained the board throughout the project.

- Adrian Reghitto:

Most of the work I did was on the frontend. I worked on creating the home page, as well as the menubar and the login, logout, and registration pages. I worked on showing contacts and displaying error messages for the /addContact endpoint. I also made the login/logout buttons work, which included both frontend and backend work. On the

backend, I worked on updating the database to match the registration form. I also fixed various errors on both the front and backend.

- LeeLee James:

I started my journey with the app name and concept, “Ping-Pong”, using my experience with branding to contextualize the action of messaging and transform that action into something auditory. I then used my frontend skills to draft the Use Case Diagram. I then implemented the HTML, CSS, Bootstrap, and JavaScript of the Home.ejs page, which housed the contacts and messages of the Ping-Pong app. I also contributed the dope graphics for our group presentation slide deck, and edited the grammar and stylings of the presentation slide deck’s text. Lastly, I learned and implemented the Mocha/Chai testing frameworks, and implemented the basic tests before being told that they would not be counted toward the grade.

- Test results: During the testing plan lab, it was suggested that we practice TDD (Test Driven Development), but the components, database, and basic functionality had already begun getting written, so there were complications with writing tests for continually updating components. Writing tests for Bootstrap and EJS with dynamically generated components was also extremely difficult. The backend database testing was possible by building a mock database to test, but that level of work was advised against since testing was not a requirement for mvp. However, in order to get experience with the Mocha/Chai test suite with this architecture, one basic test was written for each page to ensure that it was being read by the IDE, connected, and outputting html page elements. All of which were passing.

```
(base) leeleee@engr2-1-21-16-edu Project-Code % npm run test
> test
> mocha

home
  ✓ should return true

login
  ✓ should return true

message
  ✓ should return true

register
  ✓ should return true

Array
  #indexOf()
    ✓ should return -1 when the value is not present

5 passing (23ms)
```

- Deployment:

We initially had the application deployed on the CSCI server (as done in lab 12), though due to technical difficulties the current version is not deployed to it. The project can be deployed on localhost using docker. Instructions are as follows:

1. Navigate to the project-code folder
2. Run docker compose up
3. Navigate to localhost:3000

- Use Case Diagram

