Friendless in Seattle

Brought to you by
Team Debugalize

Elevator Pitch

Pandemic life has meant a lot of changes for many, including moving to a new town or city.

Between lockdowns, working remotely, and masks, how do you organically make new friends in your new environment? Well, you don't. You're Friendless in Seattle.

Our app helps you connect locally or remotely with people that share your interests so that you can make a new friend… or seven!

Concept

- Description
  - *Friendless in Seattle helps you meet new friends locally or around the world that share your interests.*
- Motivation for development
  - *Being the actual target audience and having moved and struggled to make new local friends.*
- User story
  - *If I register and fill out my 'profile' and choose to meet virtual or local friends and choose an interest category, then when others are selecting the same criteria it'll populate my friends page.*
  - *If I select "message" on a friend in my friends page, then I am taken to a page that contains messages between myself and just that friend.*
  - *When I view the messages page, I am presented with messages that i have open.*

Process

- Technologies used
  - *Git, HTML, CSS, Bulma, JavaScript, JQuery, Handlebars, Node.js, Express.js, MySQL, Sequelize*
  - ***New technology used*** - *https://api.multiavatar.com/*
- Breakdown of tasks and roles
  - ***Concept/Design/Wireframing*** - *LC, CR*
  - ***HTML/CSS/Bulma/Handlebars*** - *LC, EW, LP, CR*
  - ***Sequelize*** - *SC, LP, CR*
  - ***Routes*** - *SC, EW*
  - ***Heroku*** - *SC*
  - ***Javascript*** - *LP, EW, SC*

Process

- Challenges
  - Handlebars (LC)
  - Starter code (LC, EW, CR, LP, SC)
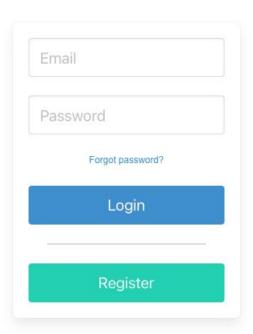 - Git/branch merging (LC, EW, CR)
  - Complications in Sequelize (LP)
- Successes
  - HTML/CSS/Bulma (LC)
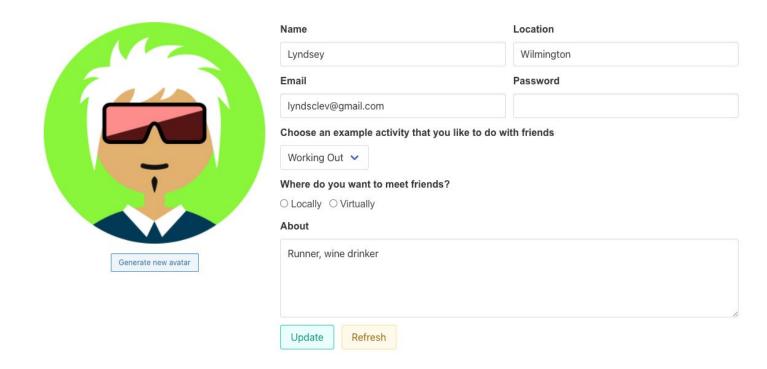 - Sequelize Relationship Mapping and Routing Data to Front End (SC)

Demo

Friendless in Seattle

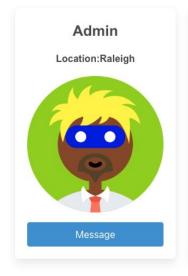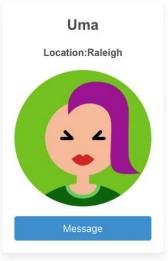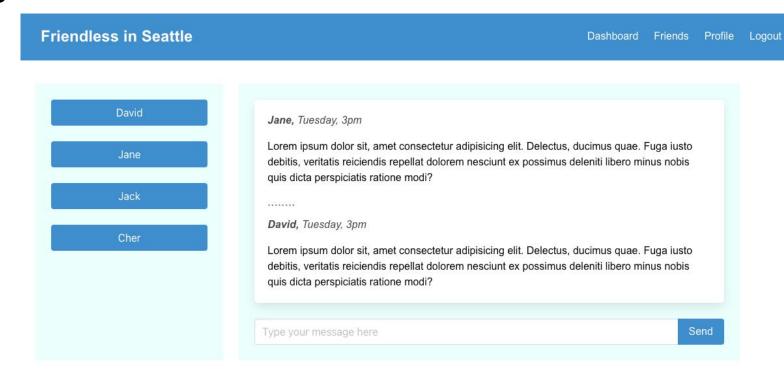Meet new friends locally or around the world that share your interests.

Email

Password

Forgot password?

Login

Register

Demo

Demo

Demo

Run Down of Message

Messages

Id	ID
userId	Users.Id
conversation_id	Conversation.Id
message	text

You cannot have a message without a conversationId

Users

Id	ID
username	Text
email	Email
password	password

Conversation

Id	ID
createdAt	date
senderId	userId
receiverId	userId

UserToConversation

Id	ID
userId	Users.Id
conversation_id	Conversation.Id

Title
http://lucidchart.com

Message

Title
http://lucidchart.com

New Message to:
PErson Name

New Conversation #

Message to send here

Send

The following tables will be populated:
1. Conversation Record - ConversationId
2. Message Record

1. Check if users have any conversation if not create a new one.

SELECT conversation_id where senderId = myUserID and receiverId = receiveruserId

if no results = No conversation with destination user
If resutl (only 1) then extract messages related to that conversation
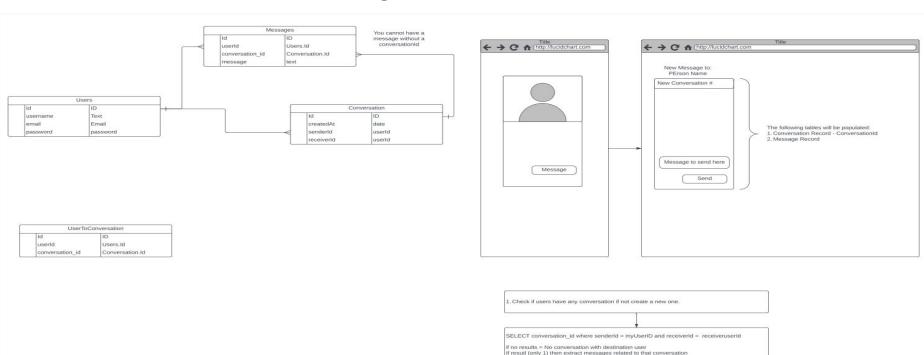
Get Messages for sender/reciver conversation:
Select * from messages where conversationId = converasionID

Directions for Future Development

- Ability to accept or deny friend matches
- Ability to filter/search community dashboard
- Wider array of interests to narrow down friend search

Links

- Repo - https://github.com/Clos01/Friendless-in-seattle
- Deployed app - https://friendless-in-seattle.herokuapp.com/