Project Features List:

1. *Login page*

This page allow users to both register a new account and login to their personal account. User can user their email as the account number to register a new account, and create passwords. For login, they need to enter their account number and password to access their home page. For all the users they can create only one account for one email address. This page allow user to access their account and protected user's personal data. This login page will be the first page to access when customers use Caravan.

2. *Profile customization*

The Profile allows users to add descriptions about themselves and it is a public page for other users to view. In the profile page, users can edit and display their name, gender, age, type of climbing they do, experience Level, equipment they have, available schedule, and a short description about their personality and hobbies. This page is helping users who want to find a workout partner. This Profile page will help customers to attract potential workout partners, giving users the opportunity to gain change to chat with other users. The Profile page will be the first information user can view about other users.

3. *Find Buddy*

A Data-based tool present users with their top percentage matched users. This tool can calculate all users' information they entered in profile page. The tool will recommend potential workout partners who have the same availability and closed experience level. This tool is helpful for users to find a workout partner that satisfy their needs in the quickest way. Customers can use this tool based on their needs.

4. *Locations*

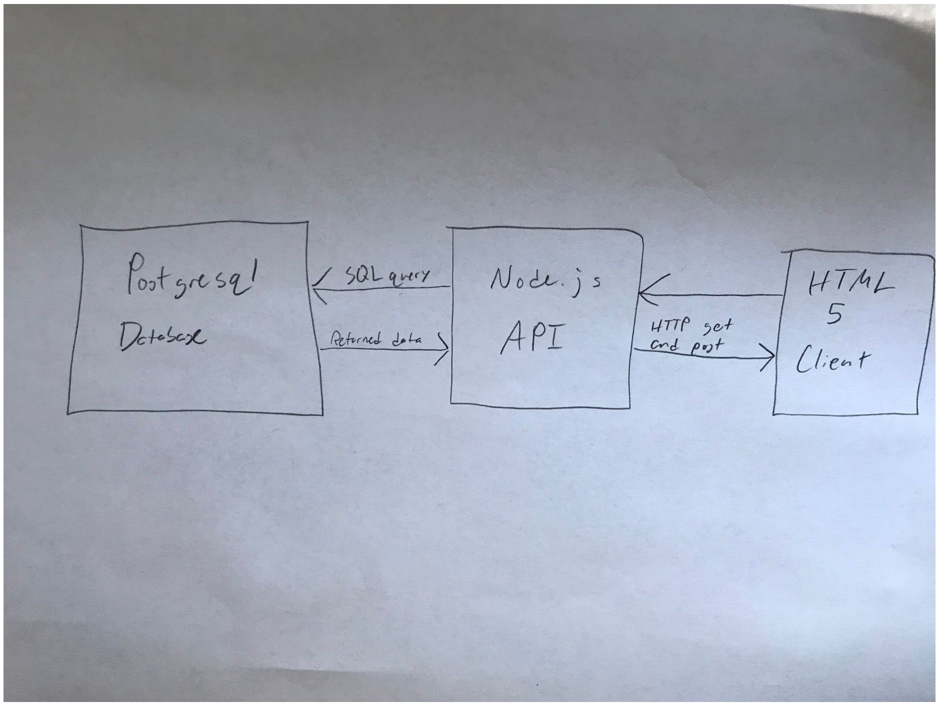
Users can report their trips to locations or desire to go to a location and then other users can find buddies by going through these locations, or contact users who have reported trips there for advice/information. When searching for a buddy, users can choose to filter their search using locations. To do so, in their profile, users can manually report their trips to locations and locations they wish to visit. Users can then search for buddies that want to go to the same place, or have gone to places they want to go to and ask for advice and information. Through the search, the option will be presented to the user to search for "Buddies Who Want to Go" and "Buddies Who Have Gone". They can then enter the location they are thinking about and press "search". Searching through the database, any buddies matching the criteria will be listed in order, based off of other compatibilities.

5. *Direct message*

Send messages to new buddies to arrange excursions, ask about gear, and so on When a list of buddies has been populated, users have the option to click on potential buddies' profiles and view it. On the profile, there will be a button labeled "Send a Message!". The user can then press the button, be directed to a messaging page, and can send messages to the potential buddy. When a message is sent, the user being messaged will receive the message from the other user, and be given access to their profile to view. The receiving user can then choose to respond or ignore the message. Users will also have the option to block users through a menu in the messaging page, recommend users to their friends, and enable notifications for specific users.

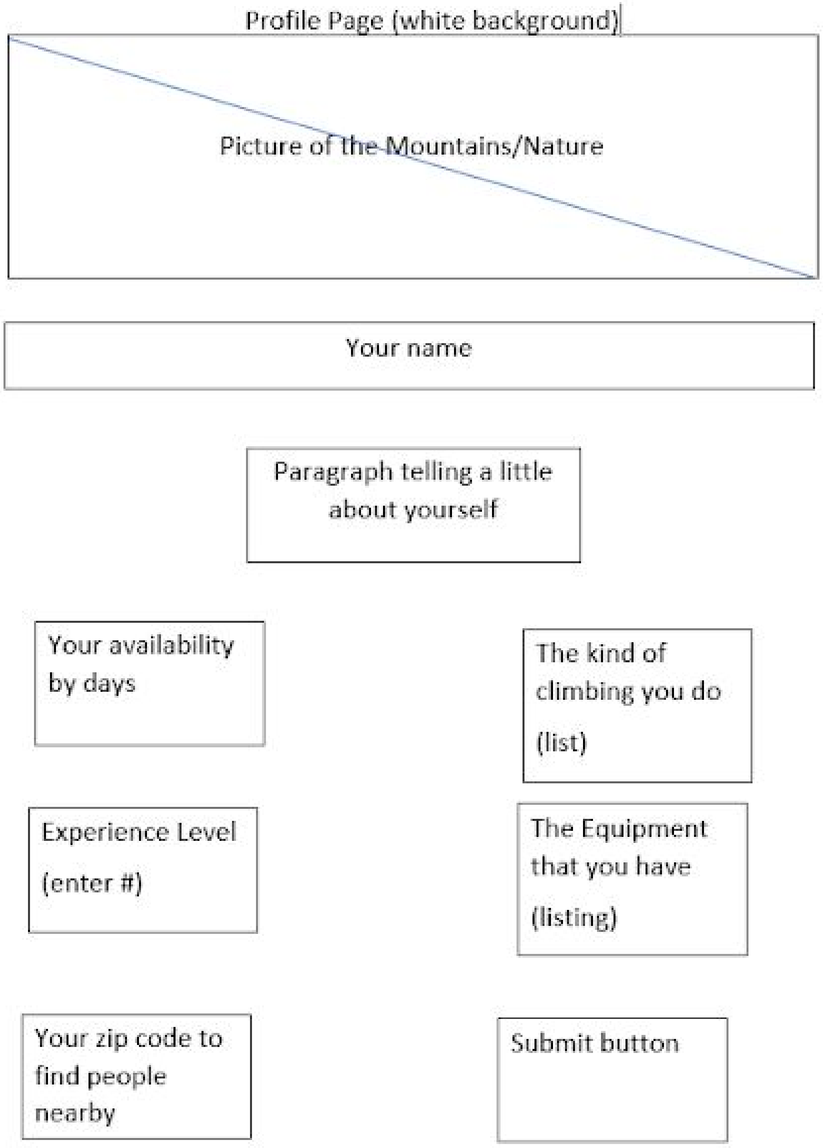
6. *Excursion list*

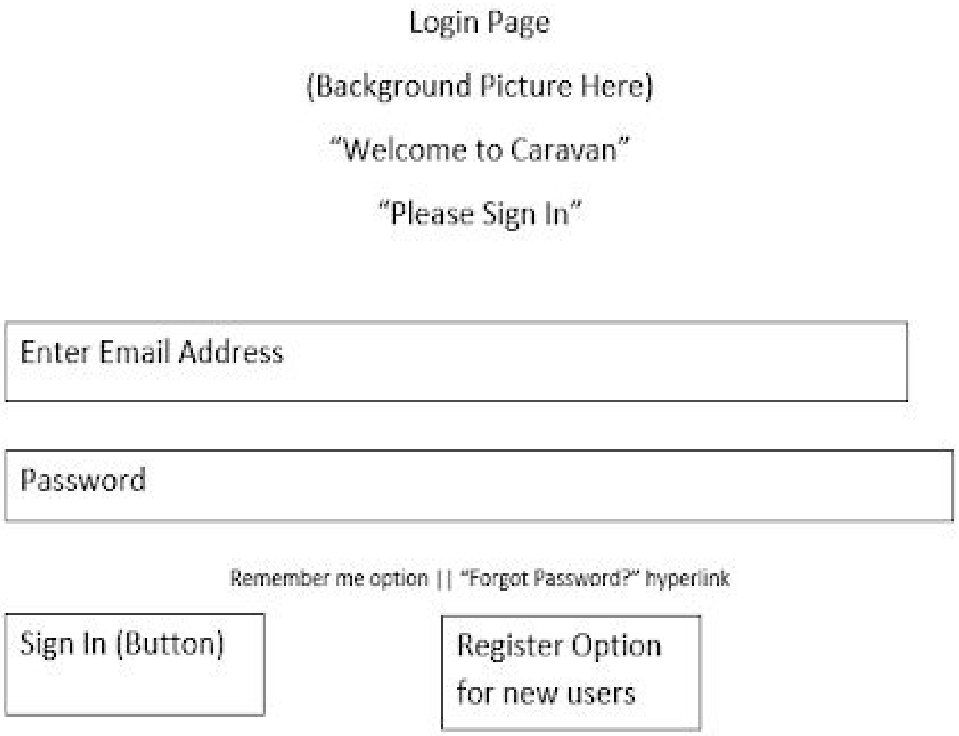
Users can add their excursions to a public list if they wish to make an open invitation, and then other users can browse through the list to join excursions that fit their schedule From the home page, users can access multiple features, one of them being the public excursion list, where users can upload an open invitation to all Caravan users within a certain mile radius. Users can change the distance settings for this feature. All public invitations within their radius will populate on their excursion list. The user can then browse through the list of excursions and chooses one that interests them by clicking on the listing. The event profile will be opened and users can view the details of the event and message the inviting user for more information.



Our database will be Postgresql, which will provide data to our Node.js API when an SQL query is submitted. From there, Node.js will use HTTP post and get to communicate with the client-side HTML 5 webpage.

Front-end Design


Profile Page (white background)
Picture of the Mountains/Nature
Your name
Paragraph telling a little about yourself
Your availability by days
The kind of climbing you do (list)
Experience Level (enter #)
The Equipment that you have (listing)
Your zip code to find people nearby
Submit button



Login Page

(Background Picture Here)

“Welcome to Caravan”

“Please Sign In”

Enter Email Address

Password

Remember me option || “Forgot Password?” hyperlink

Sign In (Button)

Register Option for new users

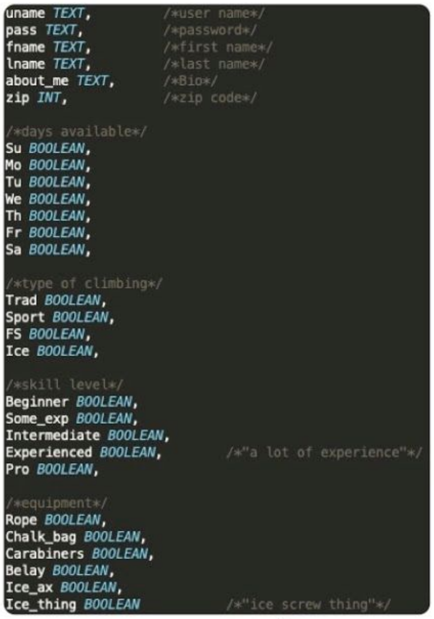
Web Service Design

For our application, Node.js will process the HTTP request. Using REST API, the clients will POST, GET, PUT, and DELETE data regarding the account's login and profile information from the database. When certain criteria are searched by the user, other users' profiles aligning with those criteria will be retrieved and displayed for the user. The Web Services will also handle login, account and profile updates, account deletes, and buddy requests.

Caravan - Database Design

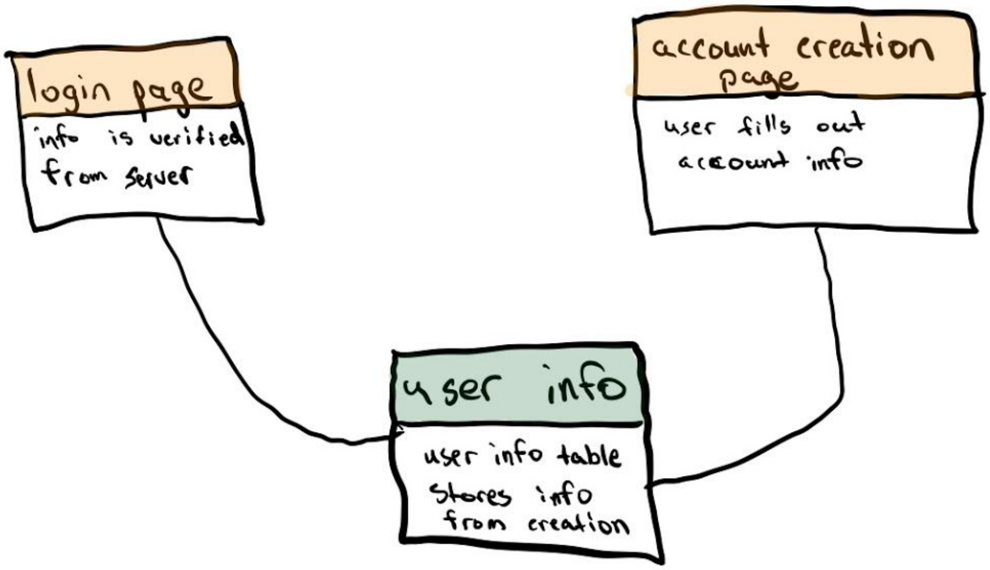
Our database currently consists of one table. We may break this table into smaller tables in the future.

Tablename = user_info



```
uname TEXT,           /*user name*/
pass TEXT,            /*password*/
fname TEXT,           /*first name*/
lname TEXT,           /*last name*/
about_me TEXT,        /*Bio*/
zip INT,              /*zip code*/

/*days available*/
Su BOOLEAN,
Mo BOOLEAN,
Tu BOOLEAN,
We BOOLEAN,
Th BOOLEAN,
Fr BOOLEAN,
Sa BOOLEAN,

/*type of climbing*/
Trad BOOLEAN,
Sport BOOLEAN,
FS BOOLEAN,
Ice BOOLEAN,

/*skill level*/
Beginner BOOLEAN,
Some_exp BOOLEAN,
Intermediate BOOLEAN,
Experienced BOOLEAN,          /*"a lot of experience"*/
Pro BOOLEAN,

/*equipment*/
Rope BOOLEAN,
Chalk_bag BOOLEAN,
Carabiners BOOLEAN,
Belay BOOLEAN,
Ice_ax BOOLEAN,
Ice_thing BOOLEAN             /*"ice screw thing"*/
```

we implemented this table in a database using the PostgreSQL 12 server. We then created an app server on Heroku and hosted our database there.


login page
info is verified from server
account creation page
user fills out account info
user info
user info table stores info from creation