For our project, we have decided to utilize and demonstrate the following technology stacks:

1. HTML (Frontend)
2. MongoDB (Database)
3. Node.js (Backend)

Our decision is not final, though we believe that the above stacks are the best suited for the particular project that we have chosen. In choosing our technology stacks, we considered the following factors:

1. Group member experience level
2. Architecture(s) for our project
3. Desirable system characteristics
4. Potential for future growth (hypothetical)

It was decided that, since most of the group members have experience with Java/Javascript and HTML, these languages would be the best choice for the front/backends of our project. Additionally, based on our application's clear distinction of the role and workload between the client/server sides, we decided that a thin client-server architecture would work best for our application, which must handle simple client requests. We also wanted to make sure that the design for our project prioritized important system characteristics like security, performance, and availability, which can all be reasonably incorporated into a client-server architecture.

HTML

We have decided to use HTML to implement the front end of our application because we find that it is the most simple and effective way to send and receive requests based on a web form. The HTML code we will write will allow a user to submit a form on a web page, which will then be used to trigger a request to our API (https://www.refugerestrooms.org/api/docs). Using HTML, we are able to reference our API and get back a response, which can then be displayed on the web page. We should note that the web page is hosted locally, meaning that the API is called from on the client's local machine.

MongoDB

We chose MongoDB because it is a nonrelational database, making it optimal for performance purposes. Though we are still unsure as to the amount of data that we will need to store, MongoDB is a useful tool in case our project would ever need to be scale, whereas using a relational database could compromise performance of our application. Additionally, MongoDB's key-value and document-oriented structure lends itself well to our project, which requires that more work be done on the server end (thin-client architecture), making it all the more necessary to have an effective database tool which can quickly fetch data and return it to the frontend.

Node.js (Backend)

We chose to use Node.js because it can be easily weaved into the event-driven process of our project. Since our application does not require any heavy computations, requires limited activity from the client and is more data-intensive on the server side, Node.js simply makes more sense. Even with the hypothetical scenario in mind in which our project would need to be scalable, Node.js is able to handle concurrent requests, meaning that many clients could be serviced without compromising performance.

Bathroom Finder

Country:
Ex: United States of Ameri

State:
Ex: Massachusetts

City:
Ex: Boston

Bathroom:
Type here...

Or use your current location

Use Current Location

Search Bathrooms

Bathroom Finder

Country:
Ex: United States of Ameri

State:
Ex: Massachusetts

City:
Ex: Boston

Bathroom:
Type here...

Or use your current location

Use Current Location

Search Bathrooms

Name: Mandarin
Address: 175 Mansfield Ave, Norton, MA, US
Name: 687 Washington St, Wrentham, MA 02093, USA
Address: 687 Washington St, Wrentham, Massachusetts, US
Name: Master's Method Tattoo and Body Piercing
Address: 116 North Main Street, Mansfield , Massachusetts , US
Name: Battleground Games & Hobbies
Address: 175 Mansfield Ave, Norton, MA, US
Name: Dunkin Donuts
Address: 45 Chauncy St, Mansfield, MA, US
Name: Fresh Catch Seafood & Deli
Address: 30 Chauncy St, Mansfield, MA, US
Name: MAC & Walt's
Address: 363 Old Colony Rd, Norton, Massachusetts, US
Name: Subway Sandwiches & Salads
Address: 130 Mansfield Ave., NORTON. MA. US
Name: MAC & Walt's
Address: 363 Old Colony Rd, Norton, Massachusetts, US
Name: Dunkin donuts
Address: 285 County street, Attleboro, Massachusetts , US

Our prototype works by taking the user's current position and using that to call the api to list all the bathrooms nearby. The API we use only takes in coordinates for location, so we haven't implemented a way to query bathrooms by country/state/city yet.