IST 411 Software Engineering: Front-end Design and Development

Project Information

Developer name: Ryan Lasauskas

Recognize

Project description

In your description, include a description of the API to be used, the user interaction, and how the information will be used and displayed for the user.

API: Open Brewery API

API Description: Open brewery is a database with lots of identifying information for thousands of breweries. Below shows all of the parameters that the database can use to narrow the search and return the desired breweries. The information returned is the ID #, name, brewery type, street (address), address 2, address 3, city, state, county/province, postal code, country, longitude, latitude, phone number, website URL, last database update time, creation in database time. A search can provide the user a lot of identifying information about breweries within the search parameters.

API String: https://api.openbrewerydb.org/breweries/

API Parameters:

- by_city - filter breweries by city, uses underscores for spaces
- by_dist - filter by distance from origin point, latitude then latitude
- by_name - filter breweries by name, underscores for spaces
- by_state - filter breweries by state, underscores for state, full name for state required
- by_postal - filter by postal code, can use 5 or 9 digit postal code to search
- by_type - filter by type, all lowercase letters in the word
    - Micro
    - Nano
    - Regional
    - Brewpub
    - Large
    - Planning
    - Bar
    - Contract
    - Proprietor
    - Closed
- page - which page of breweries to return
- per_page - number of breweries returned in a call, default 20 and max of 50
- sort - sort results by multiple fields, - ascending (default), + descending

Many of the parameters may require validation in order to input the information correctly such as entering a city. San Diego will be put into the API string as san_diego, so it needs to lowercase all the letters and use underscores instead of spaces. At the moment, I believe hyphens can remain the way they are.

Chosen API Parameters: The filters I am going to allow the user to use are City, State, Type, and Name. Additionally there is a direct search by database ID number that will be used, and the random brewery button will use this feature as well as a direct search by user if they have the

number.

User Interaction: The user can select a variety of filters such as state, type, city, and name. The city and name parameters will likely be done as text boxes while state and type will be done as dropdowns. Use of a form to submit the selected criteria and display results.

There will be a random brewery button that uses the ID search of a random number within the ID range. The random brewery button will display a single brewery's information.

Search by ID number may be used as well via text box. This uses the database ID number that the user would have to know beforehand or at least use a number inside of the ID range. This will display a single brewery's information.

Display: Display results as a "card" potentially using bootstrap for a nice template where each important field can be displayed with the information. Within the card, all returned information will be displayed in a list much like the JSON but cleaned up to look nice and be readable. Will not display the created_at information as the user likely does not care about when it was put in the database. Anything that returns "null" will display N/A or something with the same meaning as the information is unavailable.

Project Plan

Add rows as needed and renumber as appropriate when adding your milestones.
At a minimum, add a milestone for the completion of each user story in the Analyze section.

Milestone	Description	Complete by	Done
	Sprint 1 begins	Thur, 4/15	4/22
R1	Project assigned	Thur, 4/15	4/15
R2	API		
R2.1	Research APIs	Mon, 4/19	4/15
R2.2	Select API and confirm with instructor	Mon, 4/19	4/15
R3	Project Information document		
R3.1	Download, rename, modify and submit (initial)	Mon, 4/19	4/21
R3.2	Update and log project activity	Ongoing	
R4	Sprint 1 midpoint standup due by 9:00am via email	Mon, 4/19	4/18
	Sprint 2 begins	Thur, 4/22	4/29
R5	Sprint 2 midpoint standup due by 9:00am via email	Mon, 4/26	4/25
R5.1	Practice with additional tools (research if needed)	4/29	4/29

R5.1.1	Less **A6**	4/29	4/28
R5.1.2	Vue.js **A6**	4/29	4/29
R5.2	Practice with the API **A1**	4/29	4/25
R5.3	Find Bootstrap Styles that I want to use	4/29	4/28
	Sprint 3 begins	Thur, 4/29	5/6
R6	Sprint 3 midpoint standup due by 9:00am via email	Mon, 5/3	
R6.1	Begin developing the final page	5/2	
R6.1.1	CSS	5/2	
R6.1.2	HTML layout	5/2	
R6.2	Incorporate Vue.js and React Together	5/2	
R6.3	Finish final page and post bugs if needed **A8**	5/6	
R6.4	Fix bugs **A8**	5/6	
R6.5	QA review for my page	5/6	
	Sprint 4 begins	Thur, 5/6	
R7	Sprint 4 midpoint standup due by 9:00am via email		
R8	Final project		
R8.1	Application, including QA, complete Application files submitted in Canvas	Mon, 5/10	
R8.2	Project Information fully updated in Canvas	Mon, 5/10	
R8.3	Improvements if needed	Thur, 5/13, 6:00	
R8.4	Project Demonstration/Presentation **A10**	Thur, 5/13, 6:00	

Analyze

User Stories

Include the completion of each as a milestone in your project plan.
Complete each list item to specifically reflect what your application demonstrates or uses.

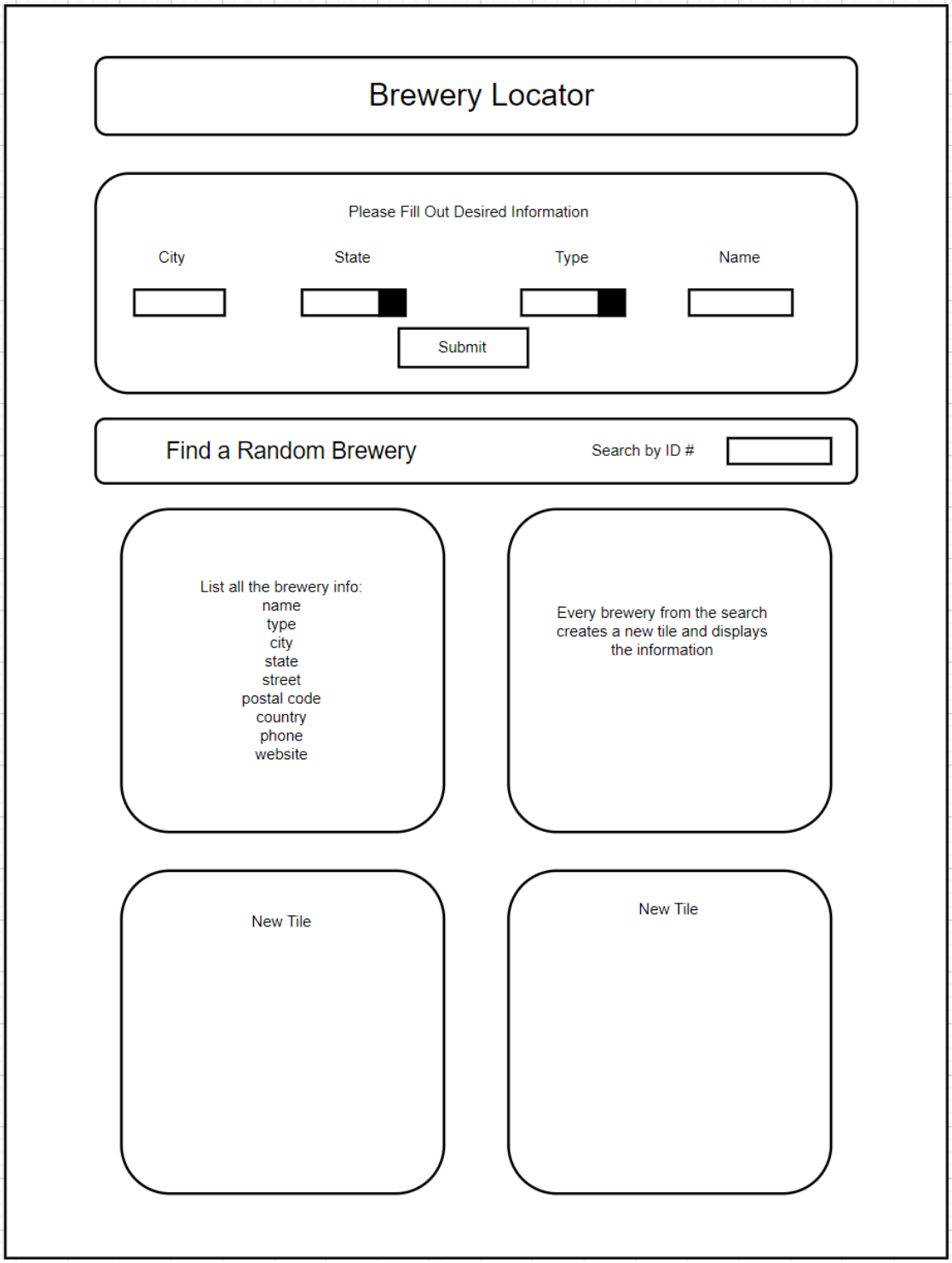
	User story
	Complete all of the following for **full credit**
A1	The application uses an API approved by the instructor to provide information to the user. Open Brewery DB
A2	The application interacts with the user allowing the user to select an input as described below to determine what information is provided. 1.
A3	The application demonstrates one best practice in front-end design. 1.
A4	The application demonstrates one best practice in responsive design. 1.
A5	The application demonstrates one best practice in accessible design. 1.
A6	**All** of the required front-end tools listed in the Design section are effectively and meaningfully used
A7	**One** additional front-end tool not listed in the Design section is effectively and meaningfully used 1.
A8	All known problems that need to be addressed are listed in the Support section.
A9	All target dates as listed on the Project Plan in the Requirements section are met.
A10	The final presentation demonstrates the completion of all relevant user stories and provides additional information on the new front-end tool used.
A11	Two Quality Assurance (QA) reviews are completed identifying the user stories that are completed and those yet to be completed: 1. Your QA review of your own project (incorporated into your presentation video) 2. Your QA review of a peer's project (peer review assigned in Canvas)
	Additionally, complete the following for **exceptional credit**
A12	The application interacts with the user allowing the user to select two or more input to determine what information is provided. 1. ID # 2. City

	3. State 4. Type 5. Name
A13	The application demonstrates two or more best practices in front-end design. 1. No global variables 2. Comment code thoroughly 3. Understandable variable names
A14	The application demonstrates two or more best practices in responsive design. 1. Mobile First 2. Scalable pictures and video 3. Media queries
A15	The application demonstrates two or more best practices in accessible design. 1. Screen Reader 2. Alternate Text
A16	**Two or more** additional front-end tools not listed in the Design section are effectively and meaningfully used. 1. Less 2. Vue.js

Design

Sketch of UI

Insert image.



Description of UX

Clearly explain in detail how the user will interact with the components in your UI sketch, and how your components will behave.

There will essentially be three different ways to search for a brewery.

The first way will be to fill in parameters to narrow the search. The city and name will be text boxes so these will be free to type whatever but will only work if typed correctly. The state and type parameters will be dropdown boxes as they only have so many to choose from and are less variable than the other options. A submit button will call the api and when the results come in they will be displayed lower on the page where each separate brewery will have its own card with the corresponding information.

The second way to make the api call will be to search by id number. It is a simple way to search but the id number to brewery relation is only in the api's dataset so the id number is useless anywhere else. This would be a nice and easy way to find a brewery that the id number is already known. Also displays the brewery like before

The third way to make the api call is to do a random search. The search will generate a random number between 8034 and 15895. It does the same api call as before only utilizing the id number. Also displays the brewery like before.

On all types of inputs there will be validation to make sure that the input is usable inside the API string.

Required front-end tools being used as part of the technology stack

Tool	Purpose
HTML, CSS, Javascript	Foundational front-end technologies
VS Code	Code editor/Integrated Development Environment (IDE)
Source/version control	git/Github
Bootstrap	Front-end CSS framework
React (also called ReactJS)	Javascript Library

Additional front-end tools being used as part of the technology stack

Create an entry for any tool you meaningfully use in completing your application

Tool	Purpose
Required:	Less
Exceptional (optional):	Vue.js

Implement

Project log

Update with an entry for each time you work on the project.

Date	Minutes	Project activity performed
4/15/21	180	Researched for API to use off the provided list of APIs and chose the open brewery database. Early research on additional development tools and chose Less which will be an additional styler, and Vue.js which will be an additional JavaScript Library and framework. May have to do two pages as Vue.js and React are similar frameworks, but will have to do more research to determine if I can use components from one to put into the other. Less will be used along with bootstrap for styling. ID # in the OpenBrewery DB range from 8034 to 15895. The User interface is rough sketched in which there will be an area the user interacts with and then an area where the returned information will be neatly displayed. Multiple search options will call the API and present information to the user. Will need to flesh out how the components of React or Vue work with the user once I know more about the interaction between Vue and React. User stories are filled in to attempt the exceptional level grade. Best practices were researched and filled in for anticipated use.
4/22/21	180	Found good resources for Less, Vue.js, and the integration of Vue and React. Vuera is an option as it is a library with the sole purpose of using Vue inside of React or the other way around. Will likely do Vue inside of React since I have more experience with React. Also found a tutorial over Vue to go through by itself so I can get a base understanding. Additionally found a tutorial using Vuera that I will go through. I believe Babel is also required for the use of Vuera, but I have not looked much at Babel yet. Found good information about Less and using it along with Bootstrap to develop a good looking page. I added information in the Project Description that gives a better understanding of what is going on for the project. I also added information to the Project Plan by adding milestones to hit and what user stories go along with the milestone.
4/25/21	60	Practiced with the OpenBrewery DB API. Made a test page so that I could figure out how I wanted to handle the different selection options and create the resulting API string. Testing all of this showed that I can use my original idea of dropdowns for the types and dropdowns and then text boxes for the remaining inputs. Can accurately create a working string that searches for what I specify.
4/28/21	420	Found a Bootstrap template that I want to use. Spent time figuring out how to use the Bootstrap template and customizing the styling with Less. Also replaced the favicon to one more appropriate to my page than the React logo. Cleaning up the public and src folders of React to get rid of things I don’t need. Lots of styling done to the page. Trying to make it nice to look at with the functionality. Most items are getting some sort of custom styling besides the Bootstrap styling. Added a footer for credit to Open Brewery DB API. Used the

		Bootstrap masthead and found a fitting picture to have in the background. Made my “cards” for the output from the API, easy to distinguish one brewery from another. Hyperlinked the website given from the API so the user can click the link if available.
4/29/21	210	Added the functionality to search by database ID. The number must be known to search this way but is only intended for someone who already knows the ID number. Worked on tidying the page. Making things look nicer and adding some user interaction components like input error. For the ID search there is a weird range that the database uses which I specify the range, but if someone inputs a number outside that range they will now get an error that says it is an invalid input and it will not call the API. Researched and done tutorials over Vue by itself. Still on the fence on if it will be easier to incorporate a Vue component into React, or if it will be easiest to do it over in Vue and I have already gotten most of the page done in solely React. Have done some tinkering with Vuera and how to use that. Vue is still new and I will need to continue to research and work on practicing with Vue to fully understand what is going on.

Support

Known problems

Update with an entry for each known problem that needs to be addressed.

Priority: As a result of the problem,
High - the application is not functional and/or the user experience is poor.
Medium - the application has an exception that negatively impacts the user experience.
Low - the user experience could be enhanced.

Date entered	Priority	Description of problem	Status

User instructions

Provide instructions to the user if necessary