AuMarche

Milestone 1 Report

Sarah Jang, Sarojini Torchon, Snigdha Thatikonda, Danny Cedrone

💡 Our Project Vision

AuMarche allows people from various ethnicities to look up different ingredients from their culture, providing them with in-person and online stores that carry those items. Additionally, it provides the english name of these ingredients and allows users to input different locations where they have personally seen the items as well. This crowdsources user feedback for more accurate locations where ingredients are found.

Key Functionalities

- Customizable user profile
- Searchable ingredient dictionary that translates a non-English ingredient name to English along with store locations to buy it
  - Speech-to-Text and language dropdown menu for better accessibility
- Saving feature that allows user to save favorite ingredients or stores
- A feedback form that collects user suggestions on where they've found specific ingredients, helping to guide other users to more accurate and reliable locations

Links

📌 **Tagged Repository:** https://github.com/Sarojini-T/CS426-auMarche/releases/tag/milestone-1-submission

📌 **Milestone & Issues:** https://github.com/Sarojini-T/CS426-auMarche/milestone/1

🛠️ The Builders

Name	**Primary Role**
Sarojini	Create Homepage
Sarah	Create Profile Page
Snigdha	Create Suggestion Page
Danny	Create Results Page

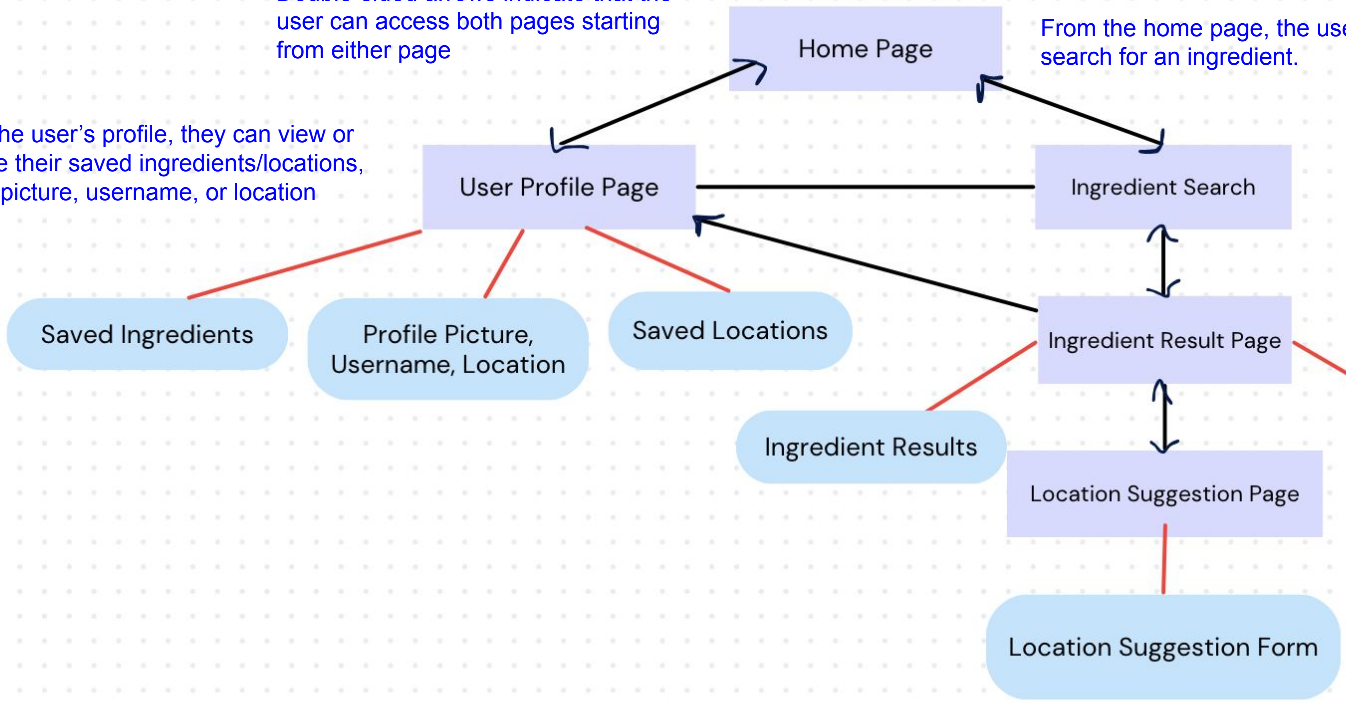
🖼️ Software Architecture Overview

Double-sided arrows indicate that the user can access both pages starting from either page

From the home page, the user can access their profile or search for an ingredient.

From the user's profile, they can view or remove their saved ingredients/locations, profile picture, username, or location

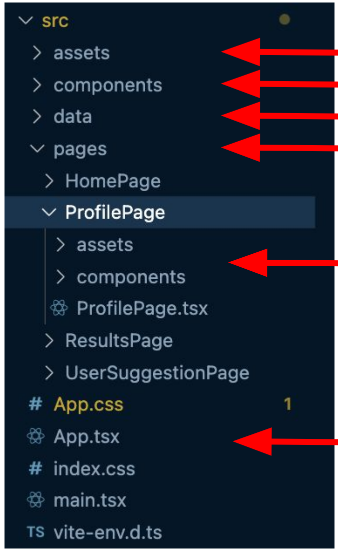
From the ingredient search, they can view the results page of that ingredient, which includes information about the ingredient and locations where they can find it.



From the results page, the user can navigate to the location suggestion page, where they can suggest where they've seen that ingredient before in a form.

Software Architecture Overview Pt.2

Project Folder Structure



```
src
  assets
  components
  data
  pages
    HomePage
    ProfilePage
      assets
      components
      ProfilePage.tsx
    ResultsPage
    UserSuggestionPage
  App.css        1
  App.tsx
  index.css
  main.tsx
  vite-env.d.ts
```

Folder for images, icons, etc.

Folder for globally shared components

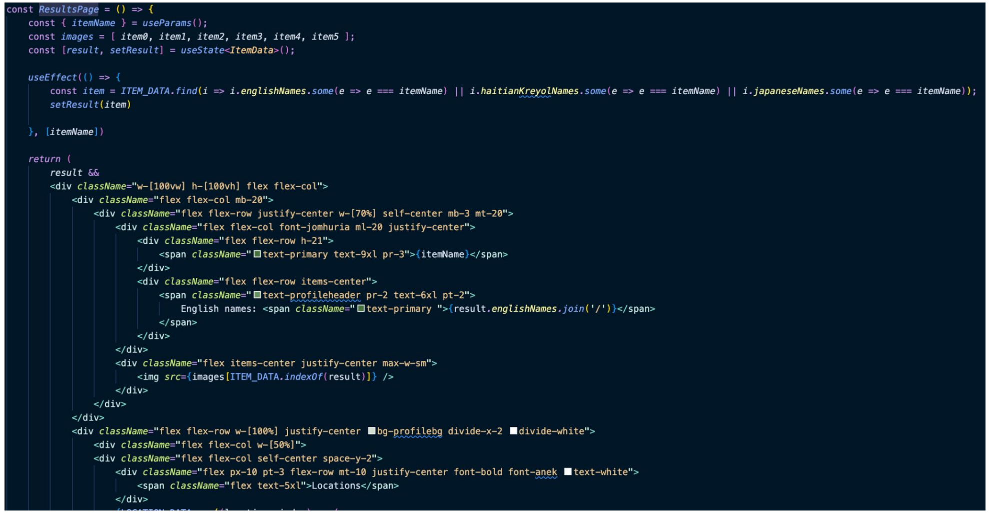
Folder for mock data

Folder for all pages

Assets and components folder for each page to store components local to a page

Entrypoint into project

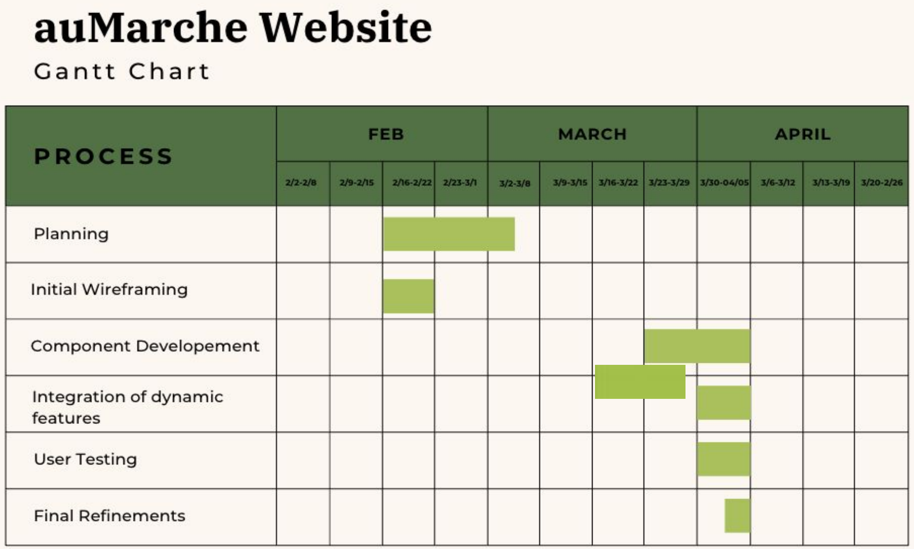
Sample Component Structure



```
const ResultsPage = () => {
    const { itemName } = useParams();
    const images = [ item0, item1, item2, item3, item4, item5 ];
    const [result, setResult] = useState<ItemData>();

    useEffect(() => {
        const item = ITEM_DATA.find(i => i.englishNames.some(e => e === itemName) || i.haitianKreyolNames.some(e => e === itemName) || i.japaneseNames.some(e => e === itemName));
        setResult(item)

    }, [itemName])

    return (
        result &&
        <div className="w-[100vw] h-[100vh] flex flex-col">
            <div className="flex flex-col mb-20">
                <div className="flex flex-row justify-center w-[70%] self-center mb-3 mt-20">
                    <div className="flex flex-col font-jomhuria ml-20 justify-center">
                        <div className="flex flex-row h-21">
                            <span className="text-primary text-9xl pr-3">{itemName}</span>
                        </div>
                        <div className="flex flex-row items-center">
                            <span className="text-profileheader pr-2 text-6xl pt-2">
                                English names: <span className="text-primary ">{result.englishNames.join('/')}</span>
                            </span>
                        </div>
                    </div>
                    <div className="flex items-center justify-center max-w-sm">
                        <img src={images[ITEM_DATA.indexOf(result)]} />
                    </div>
                </div>
            </div>
            <div className="flex flex-row w-[100%] justify-center bg-profilebg divide-x-2 divide-white">
                <div className="flex flex-col w-[50%]">
                <div className="flex flex-col self-center space-y-2">
                    <div className="flex px-10 pt-3 flex-row mt-10 justify-center font-bold font-anek text-white">
                        <span className="flex text-5xl">Locations</span>
                    </div>
```

This is the results page component. Since the results page depends on the searched item name, Danny used useParams() from React Router. Images from the ResultsPage local assets folder were stored and used. useState was used to dynamically show the results of any given ingredient, since an ingredient is not static. useEffect was used to set the ingredient stored in useState after the page has been rendered. ResultsPage() returns a page showing the ingredient information and locations where it's sold only if result is non-null.

Historical Timeline



Include dates and list of issue numbers completed

Key Issues Completed:

04/01 - Issue # 7 (homepage) closed by PR # 53
03/30 - Issue #8 (suggestions page) closed by PR #45
03/30 - Issue # 1 (profile page) closed by PR # 43
03/30 - Issue # 6 (results page) closed by PR # 44

All completed issues

🎨 Design and Styling Guidelines

Colors

🟩	🟩	🟩	🟩	🟩
Primary #48733F	Secondary #B6D7B0 (hover color)	#709C67 (not hovering color)	#B5C7B1 (saved pages sub-box)	#D7E6D3 (suggestion box color)

Text Colors

⬜	⬛	🟩
#ffffff	#000000	Primary #48733F

Breakpoint Sizes

Default breakpoints that come with tailwind

sm:	640px
md:	768px
lg:	1024px
xl:	1280px
2xl:	1536px

Typography

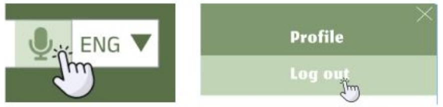
H1 Heading
Jomhuria 56 px

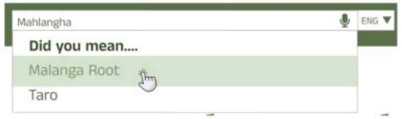
Body Text
Anek Gurmukhi **Bold** 24 px
Anek Gurmukh **Bold** 24 px

H2 Heading
Jomhuria Bold 48 px

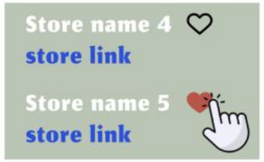
Component Behavior

Hover Color Change





On-Click Color Change



🎨 Design and Styling Guidelines

The Style Guideline can be found in the styleguideline.md file!

Component behavior includes:

- Color changes on hover for menus and buttons
- Dynamic removal of liked/disliked items from lists
- Smooth transition animations for feedback forms

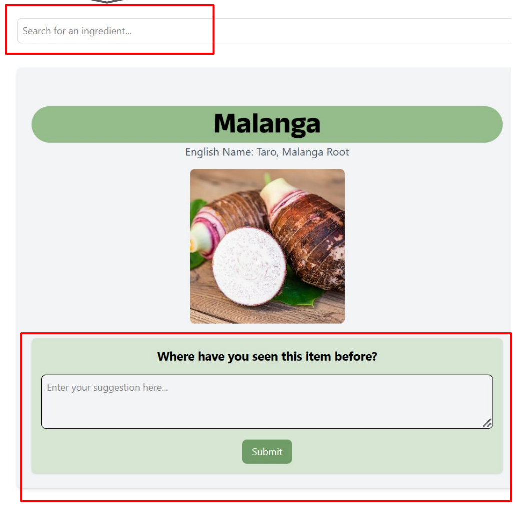
Accessibility standards are based on Apple's Human Interface Guidelines, so we emphasize:

- Thicker fonts for readability
- High-contrast buttons
- The use of color AND visuals for indicators
- Left/right-aligned text
- Feature icons (e.g., microphone for speech-to-text, arrows for dropdowns)
- Proportional font sizes
- Distinct button colors
- Proper user action feedback.

Component Documentation

Full component documentation can be found in [componentDocumentation.md]()

Search bar to select an item to make a suggestion for



Search for an ingredient...

Malanga

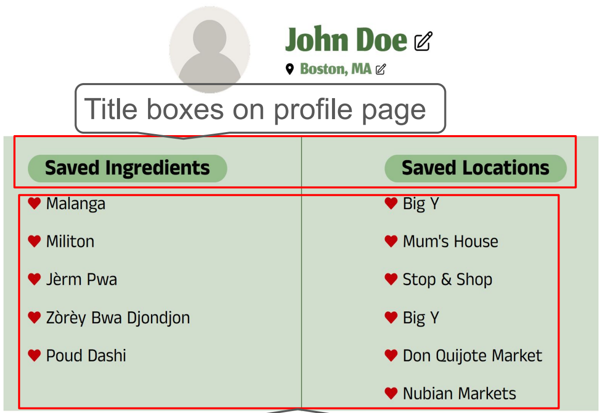
English Name: Taro, Malanga Root

Where have you seen this item before?

Enter your suggestion here...

Submit

Suggestion box to enter locations where ingredient can be found



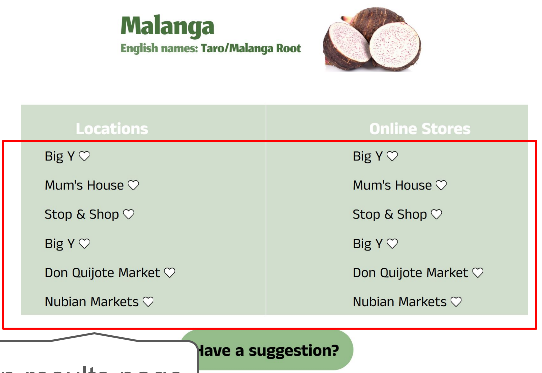
John Doe

Boston, MA

Title boxes on profile page

Saved Ingredients	Saved Locations
♥ Malanga	♥ Big Y
♥ Militon	♥ Mum's House
♥ Jèrm Pwa	♥ Stop & Shop
♥ Zòrèy Bwa Djondjon	♥ Big Y
♥ Poud Dashi	♥ Don Quijote Market
	♥ Nubian Markets

List items on profile page



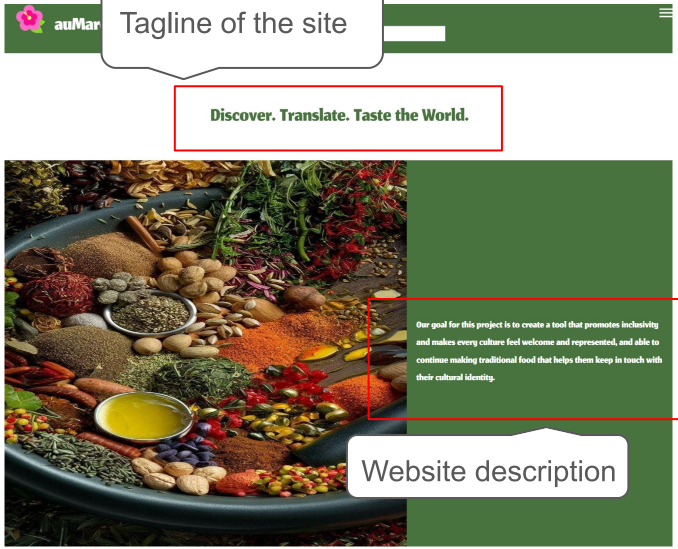
Malanga

English names: Taro/Malanga Root

Locations	Online Stores
Big Y ♡	Big Y ♡
Mum's House ♡	Mum's House ♡
Stop & Shop ♡	Stop & Shop ♡
Big Y ♡	Big Y ♡
Don Quijote Market ♡	Don Quijote Market ♡
Nubian Markets ♡	Nubian Markets ♡

Have a suggestion?

List items on results page



auMar

Tagline of the site

Discover. Translate. Taste the World.

Our goal for this project is to create a tool that promotes inclusivity and makes every culture feel welcome and represented, and able to continue making traditional food that helps them keep in touch with their cultural identity.

Website description

Performance Considerations

1. **Optimizations Made**:
   - **Conditional Rendering**: Some components (ex: `ListItem in the Profile Page` and `ListLocation in the Results Page`) are conditionally rendered based on the presence of valid data , which avoids unnecessary rendering of empty or irrelevant elements.
   - **Efficient Data Mapping**: The `ITEM_DATA` and `LOCATION_DATA` arrays are mapped directly to generate UI components, reducing the need for intermediate processing.
   - **State Management**: Pages use React's `useState` and `useEffect` hooks to fetch and display data dynamically based on the URL parameter (ex: `itemName`), which ensures that only the relevant data is processed and displayed.
   - **Image Handling**: Images are preloaded into an array (`images`) and accessed by index, minimizing redundant lookups and improving rendering performance.
2. **Profiling Results**:
   - **React Developer Tools**: Profiling with React DevTools showed that the component tree is efficiently structured, with minimal unnecessary re-renders.
   - **Load Time Improvements**: By optimizing conditional rendering and data mapping, the load time for pages isreduced, ensuring a responsive user experience.
   - **Memory Usage**: The use of preloaded images and direct array indexing reduced memory overhead compared to dynamically fetching images.
3. **Future Considerations**:
   - Implementing **lazy loading** for images and components to further optimize initial load times.
   - Adding **memoization** (e.g., `React.memo` or `useMemo`) for components prevent unnecessary re-renders when props remain unchanged.
   - Exploring **pagination** or **virtual scrolling** for large datasets in `ITEM_DATA` or `LOCATION_DATA` to improve scalability.

Snigdha Thatikonda, thatikos, 33842753

Assigned Work Summary

Assigned Issues:

- Set Up Mock Item Data
- Make the Suggestions Page
  - Display Mock Item Data
  - Customized Styling
- Components
  - Search Bar
  - User-Suggestion Box

Pull Requests: https://github.com/Sarojini-T/CS426-auMarche/pull/33, https://github.com/Sarojini-T/CS426-auMarche/pull/45, https://github.com/Sarojini-T/CS426-auMarche/pull/33,

Commits: https://github.com/Sarojini-T/CS426-auMarche/commits/main/?author=thatikos

I was not assigned any issues this milestone that weren't closed. I closed the Suggestion page issue and associated subissues.I closed the Suggestion Page issue and associated subissues.

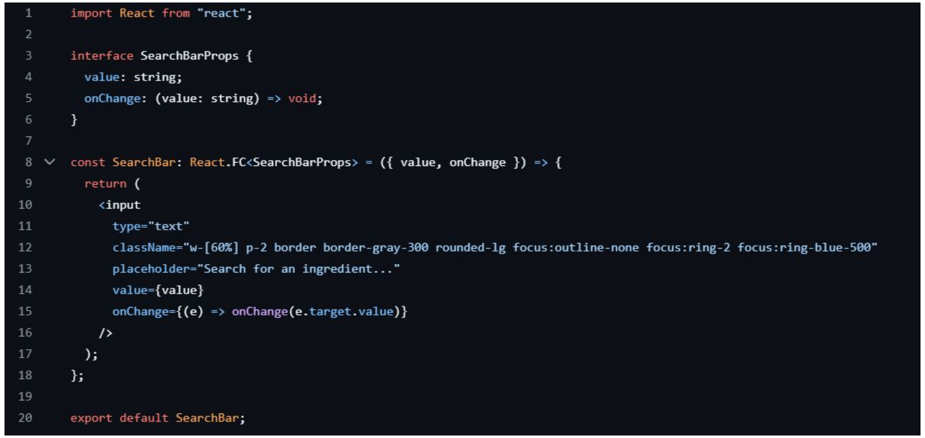
Assigned Work Summary

The main issues I was assigned were completing the suggestion page and helping set up the item data we would be using for display purposes. I added an image field to the item data and worked with Danny to figure out what we would need, then uploaded images for each item. I also made the suggestion page to render with each item. There was only one major PR that addressed all of my changes, Danny made an earlier one to make item data as well. I had another PR to add a back button.

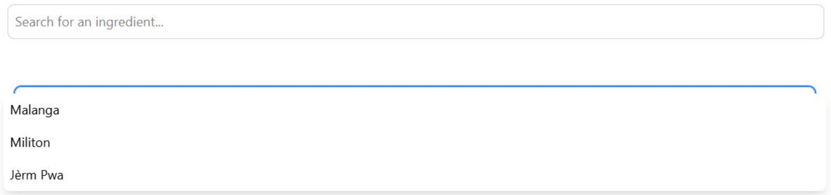
My commits included: adding the images and image field to the data, adding the route to the suggestion page, making the search bar, and then making the actual suggestion page with the suggestion box.

Snigdha Thatikonda, thatikos, 33842753

Code & UI Explanation



```
import React from "react";

interface SearchBarProps {
  value: string;
  onChange: (value: string) => void;
}

const SearchBar: React.FC<SearchBarProps> = ({ value, onChange }) => {
  return (
    <input
      type="text"
      className="w-[60%] p-2 border border-gray-300 rounded-lg focus:outline-none focus:ring-2 focus:ring-blue-500"
      placeholder="Search for an ingredient..."
      value={value}
      onChange={(e) => onChange(e.target.value)}
    />
  );
};

export default SearchBar;
```



Search for an ingredient...

Malanga
Militon
Jèrm Pwa

A key piece of code I contributed to is making a search bar for the site, which is instrumental to our UI as this is how the user navigates the site to find information about ingredients. It's currently being used to help render my suggestion page (when an item is selected, the necessary information populates a template), but it will also be used to lead people to the results page for the item. It fits into our site's design/style by using an approved font and shade of gray as well as having rounded corners (like other divs). The main challenge was just planning out how the search bar would be used across the site, which we solved by having group discussions about how the site could be navigated and organized, then choosing an option we all agreed on.

Component Hierarchy & Interaction

Within the application, my page allows us to pull information from the users to keep our site growing. Not only does it display information about the item, it allows users to suggest where other people can find an ingredient in their area. This is where the community building aspect of our site is fostered, as people connect and help one another learn about other cuisines and thrive.

The suggestions page is meant to automatically add an entry to the results page, so this is how we will be expanding the list. It uses the search bar (a global component), and will display the header and footer. You are able to directly navigate from the results page to this page, using a button in the top left.

Home Page -> Search Bar -> Results Page -> Suggestion Page

Challenges & Insights

Working within a collaborative team environment is wonderful. We troubleshoot issues together and problem solve as a team, so everyone is very involved with the project. My main takeaway is to prioritize communication they way we have been, as it has allowed us to tackle problems as they arise. The only real obstacle we are encountering is the volume of work that we are expected to complete in short amounts of time.

Completing in depth slideshow recaps, essay responses in sprint reports, and working on the website is next to impossible since we don't have a ton of structure or coherent deadlines. When talking with course staff, they either don't know how to answer our questions or have opinions that blatantly contradict each other. It's adding unnecessary difficulty.

Future Improvements & Next Steps

We want to collapse the current structure of the suggestion page so it's only accessible from the results page. For instance, a user looks up an ingredient, goes to the rendered results page, hits a button, and that should take them to the rendered suggestion page. Right now it renders off a drop down, so we want the rendering to be button dependant instead. This would also mean it won't be navigable from the home page.

For future improvements, I want to code with Danny in a meeting or in person so we can talk through changes and ideas together (since our pages are so intertwined). I also want to make the mental note of documenting the components as they happen rather than at the end of the milestone, and prioritize styling changes and responsiveness more.

Sarah Jang, ssjang25, 33244230

Assigned Work Summary

- **#34: Make Vite template:** Set up the project with Vite and create the general project folder structure
- **#32: Make UI/UX Design & Style Guideline Document:** Use Apple's Human Interface Guidelines and reference project Figma to create a comprehensive styles guideline including color schemes, typography, fonts, breakpoints. Add our custom colors and fonts to App.css stylesheet for easier use with Tailwind.
- **#1: Implement User Profile Page**
  - #2: Create editable profile picture
  - #3: Create editable user location
  - #4: Create editable saved ingredients list
  - #5: Create editable saved locations list

📌**Link to closed issues:**
https://github.com/Sarojini-T/CS426-auMarche/issues?q=is%3Aissue%20state%3Aclosed%20assignee%3Assjang25

📌**Link to commits:** https://github.com/Sarojini-T/CS426-auMarche/commits/main/?author=ssjang25

Sarah Jang, ssjang25, 33244230

Code & UI Explanation

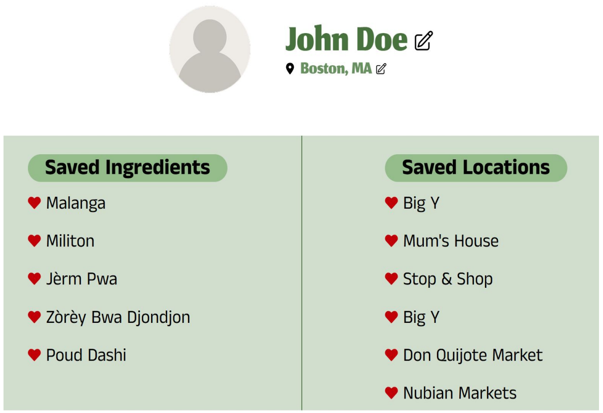
My main contribution was the user profile page.

The user profile page houses and displays the user's data,

- Indicates whether user is logged in or not
- Styles follow project's predefined color palette to boost accessibility
- Proper icons are used to indicate the function of a button (e.g, change username or location)
- Font size is large enough for better readability but not too much to distract away from the design

Challenges

- While using a flexbox layout, struggled getting icons/text to be centered inline.
  - Used padding to resolve
- Picking the right way to import icons
  - pngs -> React icons



John Doe

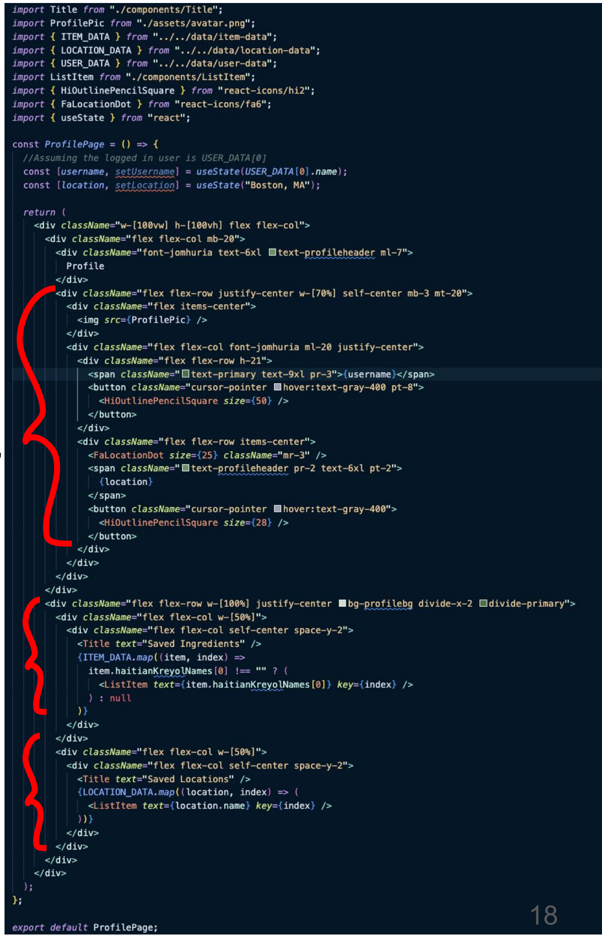
Boston, MA

Saved Ingredients	Saved Locations
Malanga	Big Y
Militon	Mum's House
Jèrm Pwa	Stop & Shop
Zòrèy Bwa Djondjon	Big Y
Poud Dashi	Don Quijote Market
	Nubian Markets

User profile header with username, profile picture, and location

List of saved ingredients

List of saved locations

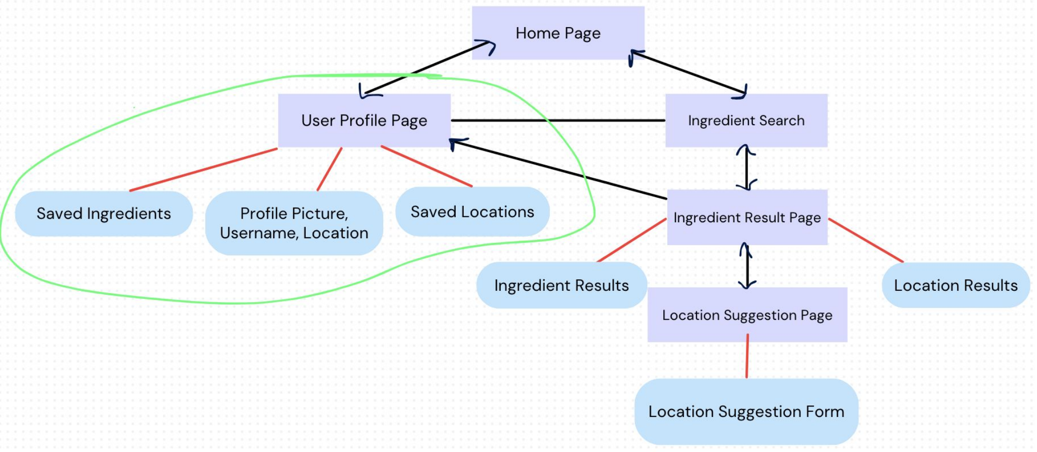


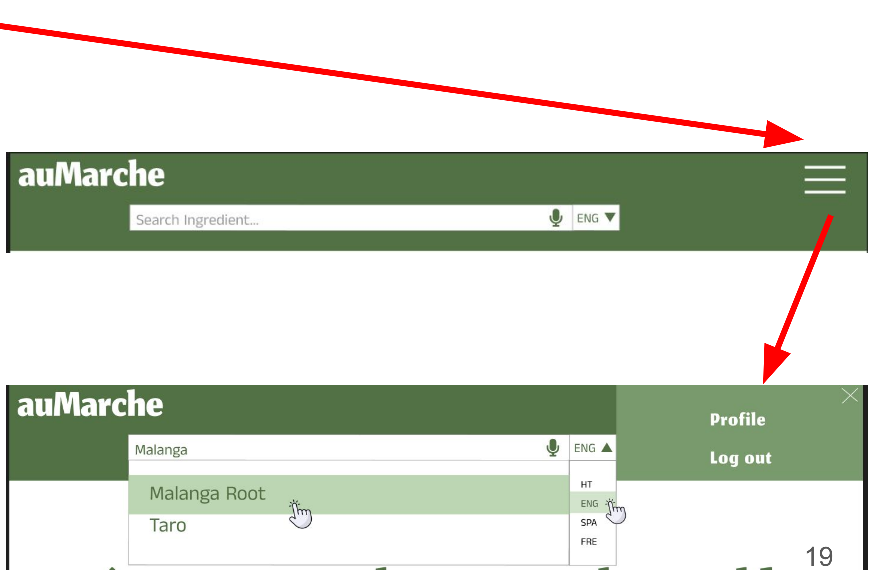
```jsx
import Title from "./components/Title";
import ProfilePic from "./assets/avatar.png";
import { ITEM_DATA } from "../../data/item-data";
import { LOCATION_DATA } from "../../data/location-data";
import { USER_DATA } from "../../data/user-data";
import ListItem from "./components/ListItem";
import { HiOutlinePencilSquare } from "react-icons/hi2";
import { FaLocationDot } from "react-icons/fa6";
import { useState } from "react";

const ProfilePage = () => {
  //Assuming the logged in user is USER_DATA[0]
  const [username, setUsername] = useState(USER_DATA[0].name);
  const [location, setLocation] = useState("Boston, MA");

  return (
    <div className="w-[100vw] h-[100vh] flex flex-col">
      <div className="flex flex-col mb-20">
        <div className="font-jomhuria text-6xl text-profileheader ml-7">
          Profile
        </div>
        <div className="flex flex-row justify-center w-[70%] self-center mb-3 mt-20">
          <div className="flex items-center">
            <img src={ProfilePic} />
          </div>
          <div className="flex flex-col font-jomhuria ml-20 justify-center">
            <div className="flex flex-row h-21">
              <span className="text-primary text-9xl pr-3">{username}</span>
              <button className="cursor-pointer hover:text-gray-400 pt-8">
                <HiOutlinePencilSquare size={50} />
              </button>
            </div>
            <div className="flex flex-row items-center">
              <FaLocationDot size={25} className="mr-3" />
              <span className="text-profileheader pr-2 text-6xl pt-2">
                {location}
              </span>
              <button className="cursor-pointer hover:text-gray-400">
                <HiOutlinePencilSquare size={28} />
              </button>
            </div>
          </div>
        </div>
      </div>
      <div className="flex flex-row w-[100%] justify-center bg-profilebg divide-x-2 divide-primary">
        <div className="flex flex-col w-[50%]">
          <div className="flex flex-col self-center space-y-2">
            <Title text="Saved Ingredients" />
            {ITEM_DATA.map((item, index) =>
              item.haitianKreyolNames[0] !== "" ? (
                <ListItem text={item.haitianKreyolNames[0]} key={index} />
              ) : null
            )}
          </div>
        </div>
        <div className="flex flex-col w-[50%]">
          <div className="flex flex-col self-center space-y-2">
            <Title text="Saved Locations" />
            {LOCATION_DATA.map((location, index) => (
              <ListItem text={location.name} key={index} />
            ))}
          </div>
        </div>
      </div>
    </div>
  );
};

export default ProfilePage;
```

Sarah Jang, ssjang25, 33244230

Component Hierarchy & Interaction

The user profile page is accessible from the home page, ingredient result page, and location suggestions page, as indicated by the arrows shown in the diagram on the left. This is because the profile is ALWAYS accessible from the navigation bar. This allows for a seamless user experience. Instead of navigating back to get to the profile page, the user always knows where to find it within the navigation bar.





Sarah Jang, ssjang25, 33244230

Challenges & Insights

The biggest obstacle we faced as a team was time management:

- As seniors, we all have busy schedules: not all 4 of us were able to simultaneously attend the planned meetings. This resulted in holes and some confusion in our communication
- However, we still updated our groupchat on meeting summaries
- Additionally, some items for this Milestone were forgotten and left until the last minute:
    - The performance checklist
    - A software architecture diagram.
- Next time, as a team, we should thoroughly review the Milestone specifications together.

My biggest takeaway is to be thorough about assignment specifications so that they aren't handled during the last minute, and be consistent with communication as increasingly busy schedules can add difficulty to good team collaboration.

✍️ Future Improvements & Next Steps

1. There is a discrepancy between the figma design and the code: A location field should be added for every user in the mock data

2. The team should consider using useReducer to make state changes more compact instead of using multiple useState statements

3. General CSS styling updates and improvements would be great for the overall project UI

Danny Cedrone, dannycedrone, 33176603

Assigned Work Summary

- #24 Set up mock user data
- #37 Set up mock location data
- #38 Create routing between pages
- #6 Implement Results Page
    - Create list of locations
    - Display name of item and all English names
    - Display image of item

Link to closed issues: https://github.com/Sarojini-T/CS426-auMarche/issues?q=is%3Aissue%20state%3Aclosed%20assignee%3Adannycedrone

Link to commits: https://github.com/Sarojini-T/CS426-auMarche/commits/main/?author=dannycedrone

Link to pull requests: https://github.com/Sarojini-T/CS426-auMarche/pulls?q=is%3Apr+author%3Adannycedrone+is%3Aclosed

Danny Cedrone, dannycedrone, 33176603

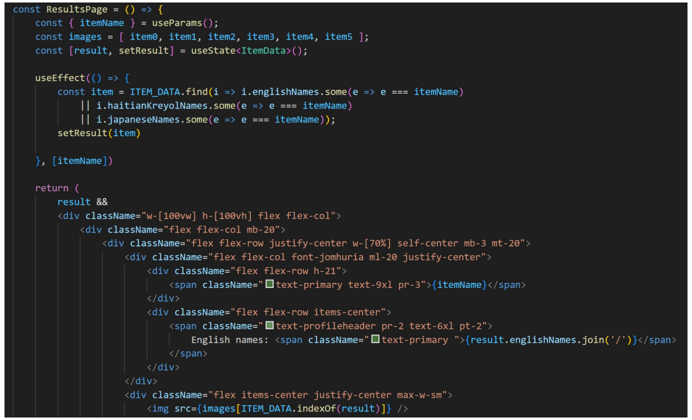
Code & UI Explanation

My main contribution for this milestone was the results page

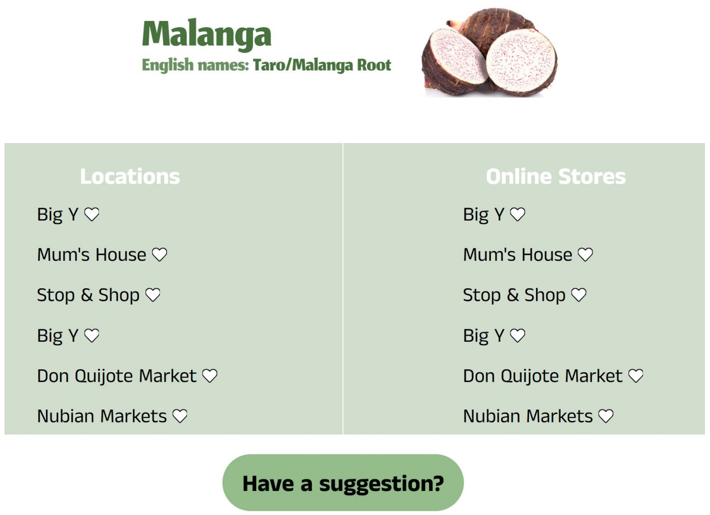
The code uses flexboxes for each section of the page, and uses useParams to get whichever item is being searched for.

The results page uses the themes set in index.css and which can be seen elsewhere in the app such as the background color, text color, and font style

The main challenge was deciding what format to use for each section of the page, especially since I don’t have much experience with frontend development, and especially styling/designing. After talking with Sarah about the profile page and referencing the wireframes from sprint 1, I decided to match the format of the profile page which uses flexboxes. This helps to make the site more cohesive



```
const ResultsPage = () => {
    const { itemName } = useParams();
    const images = [ item0, item1, item2, item3, item4, item5 ];
    const [result, setResult] = useState<ItemData>();

    useEffect(() => {
        const item = ITEM_DATA.find(i => i.englishNames.some(e => e === itemName)
            || i.haitianKreyolNames.some(e => e === itemName)
            || i.japaneseNames.some(e => e === itemName));
        setResult(item)

    }, [itemName])

    return (
        result &&
        <div className="w-[100vw] h-[100vh] flex flex-col">
            <div className="flex flex-col mb-20">
                <div className="flex flex-row justify-center w-[70%] self-center mb-3 mt-20">
                    <div className="flex flex-col font-jomhuria ml-20 justify-center">
                        <div className="flex flex-row h-21">
                            <span className="text-primary text-9xl pr-3">{itemName}</span>
                        </div>
                        <div className="flex flex-row items-center">
                            <span className="text-profileheader pr-2 text-6xl pt-2">
                                English names: <span className="text-primary ">{result.englishNames.join('/')}</span>
                            </span>
                        </div>
                    </div>
                    <div className="flex items-center justify-center max-w-sm">
                        <img src={images[ITEM_DATA.indexOf(result)]} />
```



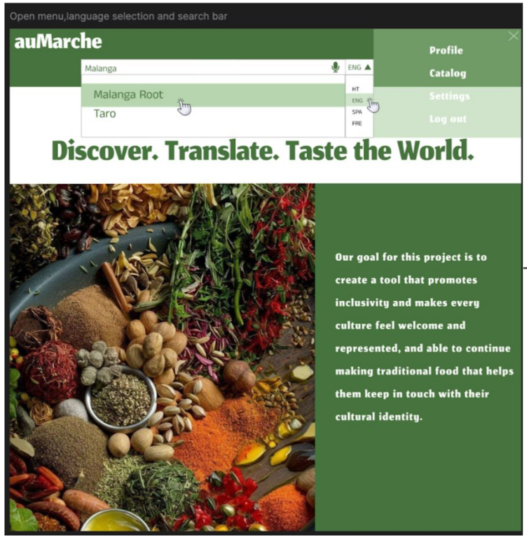
Malanga

English names: Taro/Malanga Root

Locations	Online Stores
Big Y ♡	Big Y ♡
Mum's House ♡	Mum's House ♡
Stop & Shop ♡	Stop & Shop ♡
Big Y ♡	Big Y ♡
Don Quijote Market ♡	Don Quijote Market ♡
Nubian Markets ♡	Nubian Markets ♡

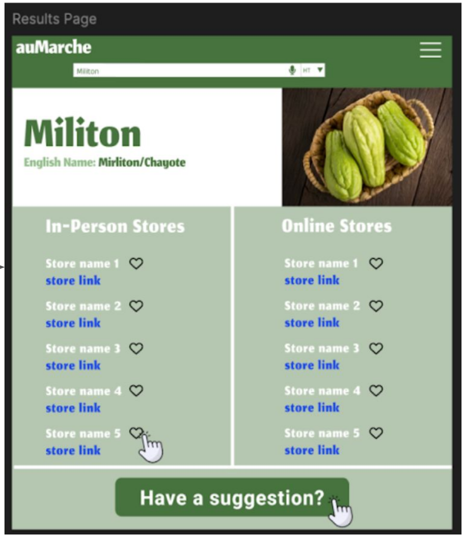
Danny Cedrone, dannycedrone, 33176603

Component Hierarchy & Interaction


Open menu,language selection and search bar
auMarche
Malanga
ENG
Malanga Root
Taro
HT
ENG
SPA
FRE
Profile
Catalog
Settings
Log out
Discover. Translate. Taste the World.

Our goal for this project is to create a tool that promotes inclusivity and makes every culture feel welcome and represented, and able to continue making traditional food that helps them keep in touch with their cultural identity.

Home page (search bar)



Results page

The results page appears after a user searches for and selects an item

One item is displayed at a time on the results page, and displayed the name of the ingredient in the current site language and the English names of the ingredient, as well as locations where the ingredient can be purchased

The results page is connected to the rest of the site via the search bar which is accessible on every page, and it is connected to the suggestions page through a button at the bottom which allows users to suggest locations where the ingredient can be found

Danny Cedrone, dannycedrone, 33176603

Challenges & Insights

The biggest challenge we faced for this milestone was timing. There were many logistical aspects that took more time than we anticipated, which left us less time to tackle the technical aspects. Being sick and unable to work for over a week was a heavy hit to my ability to contribute for this milestone. I'm lucky that my team was active and we collaborated well so we were able to bounce back relatively quickly.

My biggest takeaway from this milestone is to focus on the technical work first, and then the logistics/clerical work. Documentation is important, but you can't document something that doesn't exist. In the future, I'll try to prioritize the bigger picture which is the project actually functioning, and then work on the finer details, instead of the other way around.

Future improvements and next steps

1. Work with Snigdha on integrating the results and suggestions pages. Also, standardize the way images are stored and accessed between the results and suggestions pages

2. Continue to improve the responsiveness of the results page to ensure the page is adaptable to different screen sizes

Assigned Work Summary

Assigned Issues

#7- Homepage with reusable components

- #10- Navbar
- #13-Footer
- #22- Display mock ingredient Data
- #26- Change page to match selected language

#29- Check that core components are developed

#50- Add translated text data for each page

Closed Issues

Commits: Milestone 1 Commits

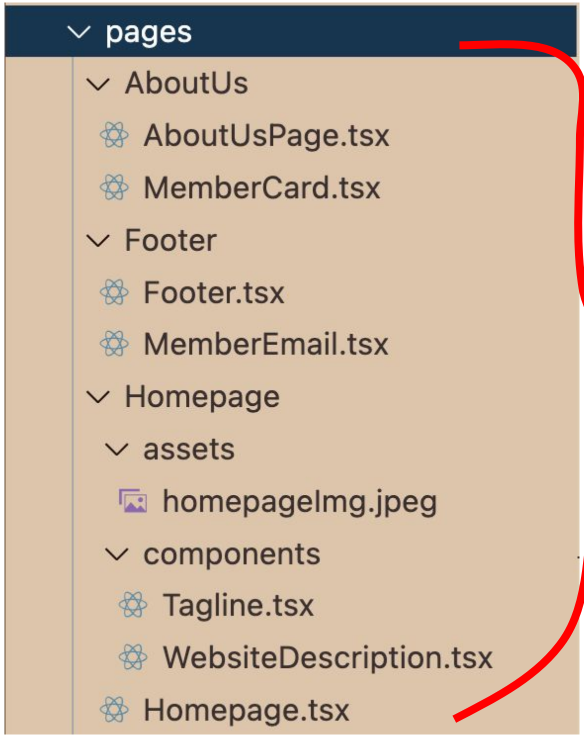
PRs: Closed PRs

I closed all issues that I was assigned for this milestone.

Sarojini Torchon , Sarojini-T, 33333898

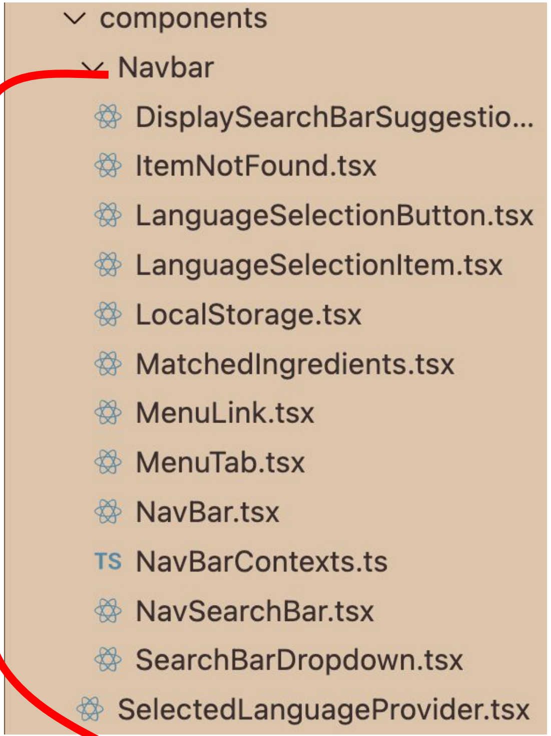
Screenshots & Demonstrations

Screenshots of UI components with annotations that you designed and implemented.



```
pages
  AboutUs
    AboutUsPage.tsx
    MemberCard.tsx
  Footer
    Footer.tsx
    MemberEmail.tsx
  Homepage
    assets
      homepageImg.jpeg
    components
      Tagline.tsx
      WebsiteDescription.tsx
    Homepage.tsx
```

Implemented the AboutUs, Footer, Homepage and NavBar Components



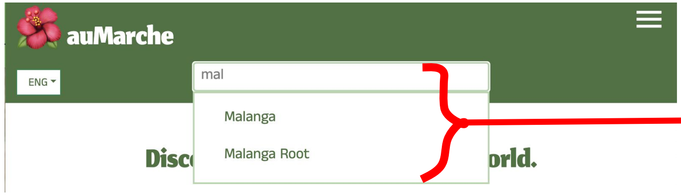
```
components
  Navbar
    DisplaySearchBarSuggestio...
    ItemNotFound.tsx
    LanguageSelectionButton.tsx
    LanguageSelectionItem.tsx
    LocalStorage.tsx
    MatchedIngredients.tsx
    MenuLink.tsx
    MenuTab.tsx
    NavBar.tsx
    NavBarContexts.ts
    NavSearchBar.tsx
    SearchBarDropdown.tsx
  SelectedLanguageProvider.tsx
```

Sarojini Torchon , Sarojini-T, 33333898

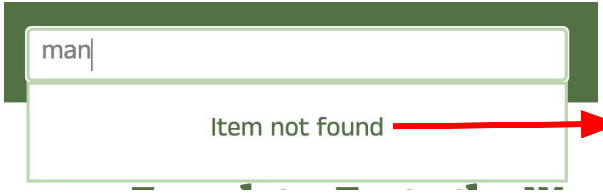
Screenshots & Demonstrations (continued)

Highlight the key functionality built by you, the team member

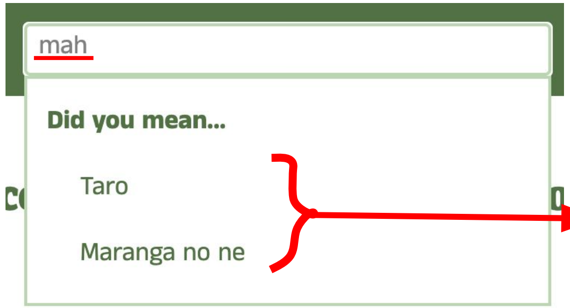
Key functionality: The key functionality that I built during this milestone was the global search bar, stored in the nav bar. It allows users to search for various food items, provides suggestions as they type and when they misspell an item's name. Once a user selects an item, they get redirected to the results page.



Autocomplete feature



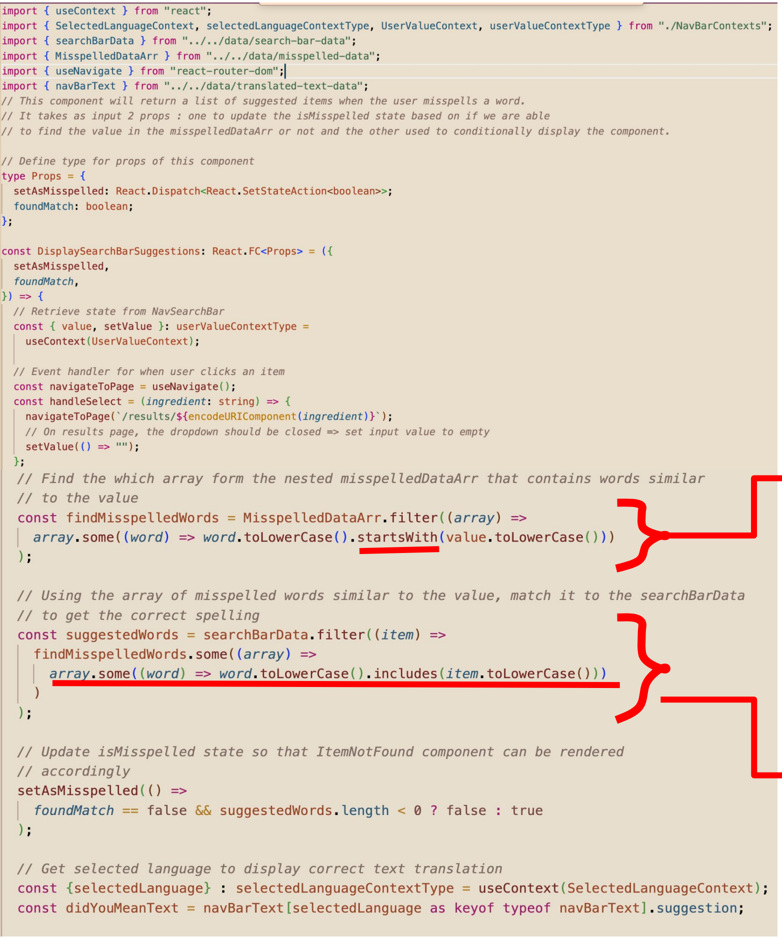
Displays message when item does not exists in our data



Suggestions when item is misspelled

Code & UI Explanation

Sarojini Torchon , Sarojini-T, 33333898



```
import { useContext } from "react";
import { SelectedLanguageContext, selectedLanguageContextType, UserValueContext, userValueContextType } from "./NavBarContexts";
import { searchBarData } from "../../data/search-bar-data";
import { MisspelledDataArr } from "../../data/misspelled-data";
import { useNavigate } from "react-router-dom";
import { navBarText } from "../../data/translated-text-data";
// This component will return a list of suggested items when the user misspells a word.
// It takes as input 2 props : one to update the isMisspelled state based on if we are able
// to find the value in the misspelledDataArr or not and the other used to conditionally display the component.

// Define type for props of this component
type Props = {
  setAsMisspelled: React.Dispatch<React.SetStateAction<boolean>>;
  foundMatch: boolean;
};

const DisplaySearchBarSuggestions: React.FC<Props> = ({
  setAsMisspelled,
  foundMatch,
}) => {
  // Retrieve state from NavSearchBar
  const { value, setValue }: userValueContextType =
    useContext(UserValueContext);

  // Event handler for when user clicks an item
  const navigateToPage = useNavigate();
  const handleSelect = (ingredient: string) => {
    navigateToPage(`/results/${encodeURIComponent(ingredient)}`);
    // On results page, the dropdown should be closed => set input value to empty
    setValue(() => "");
  };
  // Find the which array form the nested misspelledDataArr that contains words similar
  // to the value
  const findMisspelledWords = MisspelledDataArr.filter((array) =>
    array.some((word) => word.toLowerCase().startsWith(value.toLowerCase()))
  );

  // Using the array of misspelled words similar to the value, match it to the searchBarData
  // to get the correct spelling
  const suggestedWords = searchBarData.filter((item) =>
    findMisspelledWords.some((array) =>
      array.some((word) => word.toLowerCase().includes(item.toLowerCase()))
    )
  );

  // Update isMisspelled state so that ItemNotFound component can be rendered
  // accordingly
  setAsMisspelled(() =>
    foundMatch == false && suggestedWords.length < 0 ? false : true
  );

  // Get selected language to display correct text translation
  const {selectedLanguage} : selectedLanguageContextType = useContext(SelectedLanguageContext);
  const didYouMeanText = navBarText[selectedLanguage as keyof typeof navBarText].suggestion;
```

Key code contribution

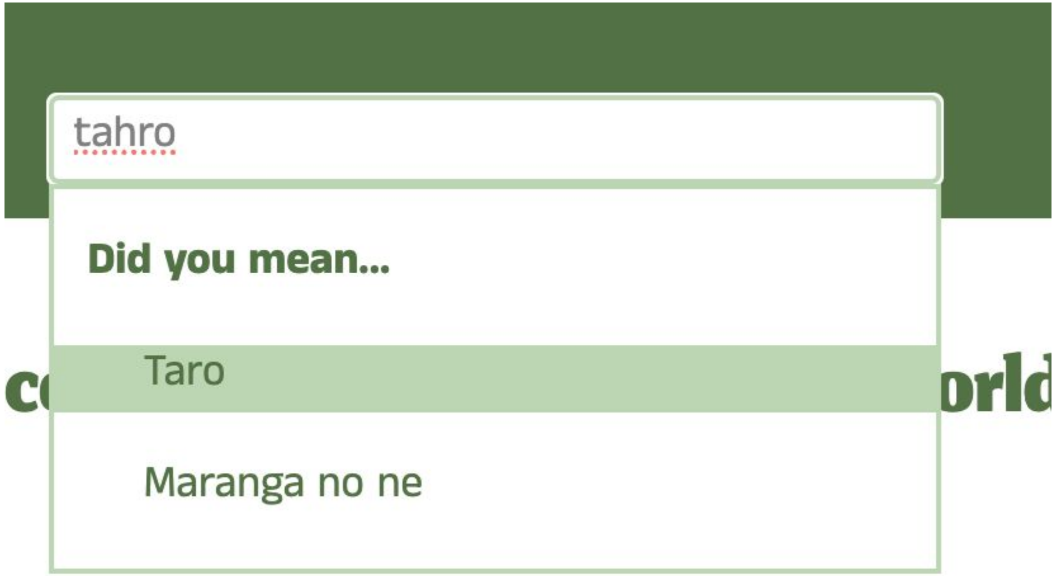
Suggestion feature in the search bar for when a user misspells an item. As the user types the word incorrectly, the search bar generates suggestions based on what it thinks the user is trying to find. Once the user clicks the suggestion, they get taken to the results page to find more information about the item.

Challenge 1: Providing suggestions as the user is typing
Sol: Filtered an array of potential misspelled words and compared each word to the value that the user is typing using the **.startsWith(**value**)** method => only return words that start with the same letters as the ones the user is currently inputting.

Challenge 2: Mapping the misspelled input to the correctly spelled item
Sol: Used the same array that was used to implement the autocomplete feature -> compared misspelled user input to items in said array (searchBarData) => return words that includes the misspelled input => Display words returned as suggestion to user

Code & UI Explanation (continued)

Illustrate and explain how the component design and style adheres to the application's style guidelines



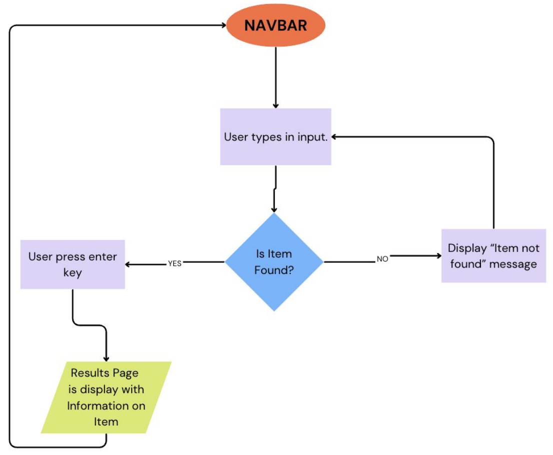
- Text color is Primary
- On hover - background of the item is set to Secondary color
- Text font is Anek Gurmukhi

Sarojini Torchon , Sarojini-T, 33333898

Component Hierarchy & Interaction

In the larger structure, my code allows user to look up items even if they're unsure of how it's spelled. The purpose of the application is to allow users to find ingredients that are common in their culture but that they have trouble finding in the US; It could be the case that the user might've only ever heard of the ingredient but never seen it written. Thus, this component makes the app more accessible for everyone. Once the suggested item is selected, the user then gets directed to the results page where more information about the item is displayed.

User Flow


NAVBAR
User types in input.
Is Item Found?
YES
NO
User press enter key
Display "Item not found" message
Results Page is display with Information on Item

User Flow description
Navbar (containing search bar) is accessible on every page thus from the navbar the user will access the search bar, input an item and if it is found, will be redirected to the results page.

Challenged and Insights

One of the main challenges I had was with the CSS, it took a lot of debugging and researching to try to correctly position the various components of the homepage on the screen, especially given the fact that I was aiming to make it look exactly like our wireframe. Consequently, I ended up spending a lot of time on that which could have been used to implement other more important features such as the global search bar or the language selection button.In the future, I plan to implement core features first then worry about the styling later.

Additionally, I had challenges making atomic commits in the beginning, which resulted in some being a bit bigger than expected. But as I continued to work on the milestone,I got the hang of it and learned when to make commits.

My key takeaway from working within a collaborative environment is that clear and frequent communication is crucial to a successful project. Consistently communicating, sharing ideas and setbacks with my team throughout this milestone was a key factor in how I was able to built good components.

Future Improvements & Next Steps

Aspect that could be optimized further/ Improved:

- Adding more css styling to the components of all pages such as animation on hover, on click, on select or shadows to make the site more visually appealing
- Adding comments to the code throughout all the components
- Making the suggestion feature for misspelled words in the search bar more accurate

Features for next sprint:

- Voice-to-text option for the search bar
- Adding more languages to the language selection button