Structure:

Bridgepoint is a web application that allows users to create an account or log in to find help based on their needs. The website focuses on helping the needy find essential resources like food assistance, housing support, healthcare, employment, etc.

Frontend: HTML, CSS, JavaScript

Backend: Firebase/Firestone databases and authentication (queries)

Database: Dynamic through different software tools of Firebase/firestone

Techniques:

1. *Programming languages:* HTML, CSS, JavaScript, and Firebase Queries
2. Firebase: Used to handle real-time data retrieval from the database, authentication of data, and storing data for access within the database

Algorithmic Thinking inside Code:

1. JavaScript validation: Validates fields like name, password, email, messages, etc.
2. Reusability: Reusable JavaScript functions to minimize redundancy and improve maintainability.
3. Sign-in/Sign-up algorithm: Using databases to step-by-step store information and retrieve it. Additionally, data gets used throughout the webpage after creation. When data is needed, firebase queries are algorithmically called and stored in locations that require it.

Programming Complexity:

1. **Database integration with Firestore***:* Storing user details through programming commands (setDoc(docRef, userData)). Creation/storing: (createUserWithEmailAndPassword, setDoc) and use .then() and .catch(). Used Firestore's document structure and ensured compatibility between Firestore, VsCode, and the terminal.
2. **Database authentication:** Firebase authentication to create users and sign in across multiple pages and data entries: (createUserWithEmailAndPassword) (signInWithEmailAndPassword). Added error handling for authentication to run smoothly (checks if email exists, duplicates, invalid cridentials, etc). Ensures that each user is distinct, and security is maximized.
3. **UI/CSS animations:** Dynamic toggling, opacity fade-out, icons, flexboxes, enhanced containers, padding, margins, hover effects, border radius, CSS grids, etc.

Algorithmic Thinking/Techniques used across certain webpages

User Registration:

- JavaScript is used to implement form validation, ensuring all fields are filled out.
- Used JavaScript and algorithms to give the user real-time feedback regarding password requirements, account management, log-in, and sign-up.

Sign-in:

- Users received dynamic feedback based on their ability to create an account. If information is missing, incomplete, or invalid, error messages appear.

Contact Us:

- Used JavaScript to improve the user experience by delivering real-time feedback on form validation (empty fields or incorrect email types)
- CSS styling was used to improve the form's overall appearance and usefulness, with CSS Grid for layout organization and Flexbox for form element alignment and readability.

Employment Opportunities:

- The survey utilizes JavaScript and Objects to write conditions that work as a filter to produce employment opportunities that match what the user input.
- A percentage match is displayed with other useful key features such as wages, location, job requirements, job type, etc.

Find Help:

- Information regarding the organization is shown based on the selected organization. Organizations are a combination of mental health, drug abuse, and homelessness support. Users can select the type of organization, and the produced result is shown after the selection
- The information displayed includes services, how it can help them, the organization's website, address, and location that utilizes the Google API key to produce the image. Embedded within Google Maps.

<u>Software tools and UI:</u>

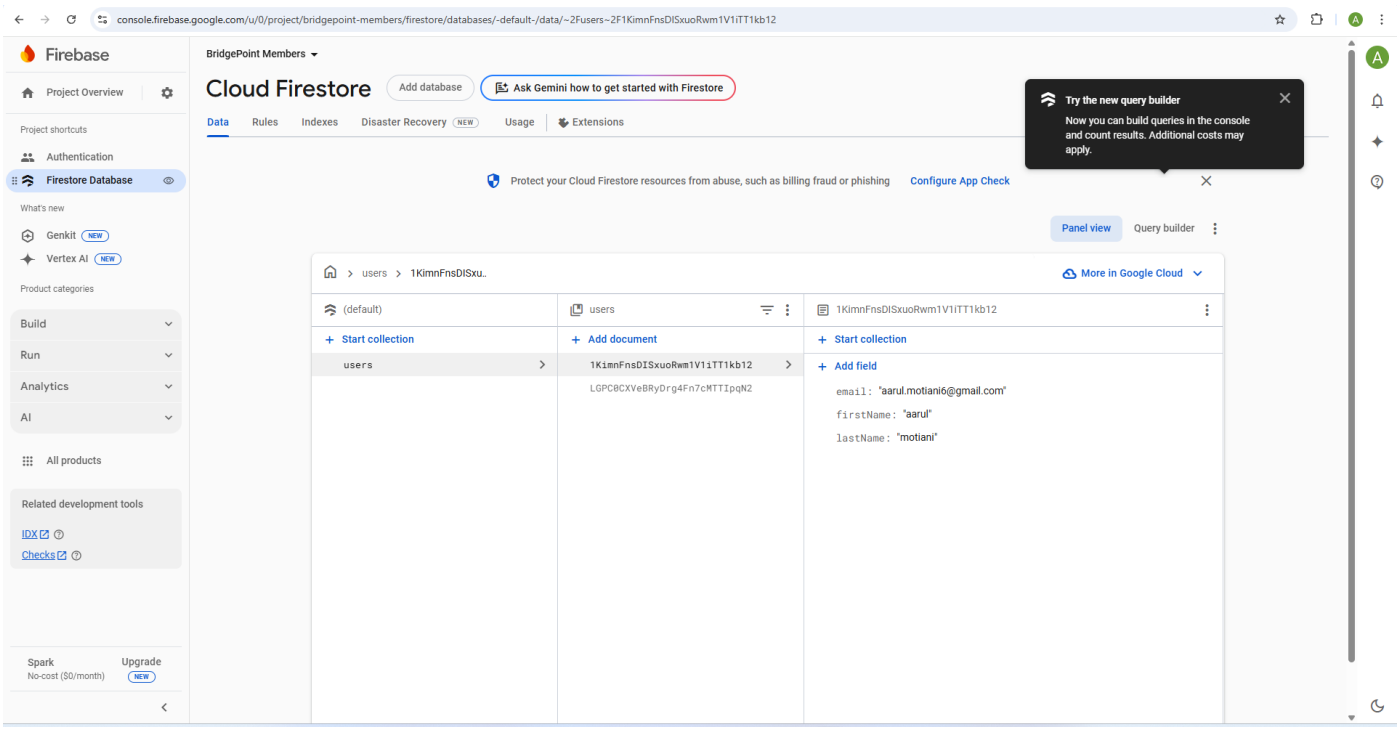
VsCode: Software used to develop websites with .css, .js, and .html files

Firebase and Firestone: Database storage with authentication that acts dynamically for users to update and access their accounts in real-time.

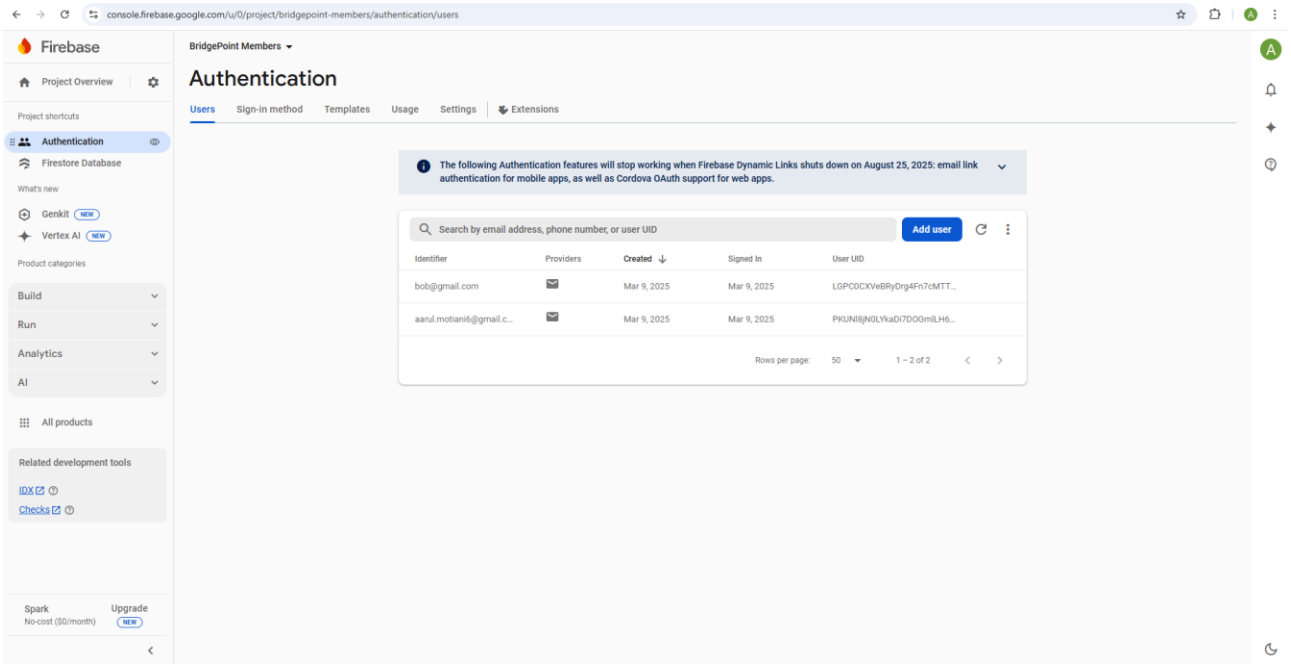
Firebase configuration in vsCode to connect databases:

```
const firebaseConfig = {
  apiKey: "AIzaSyBDyZwaPOV-A8FUKBA_rz8Ub0pd5la-m94",
  authDomain: "bridgepoint-members.firebaseapp.com",
  projectId: "bridgepoint-members",
  storageBucket: "bridgepoint-members.firebasestorage.app",
  messagingSenderId: "87779290497",
  appId: "1:87779290497:web:86c3b6da632fa2439fcc80",
  measurementId: "G-1QYRMHWHV6"
};
```

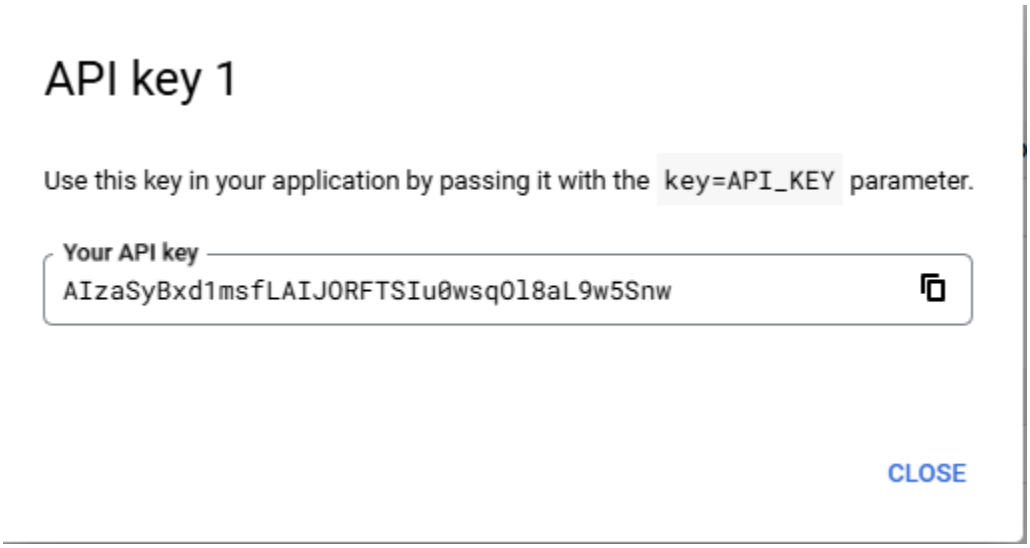
Firestone database collection proof:



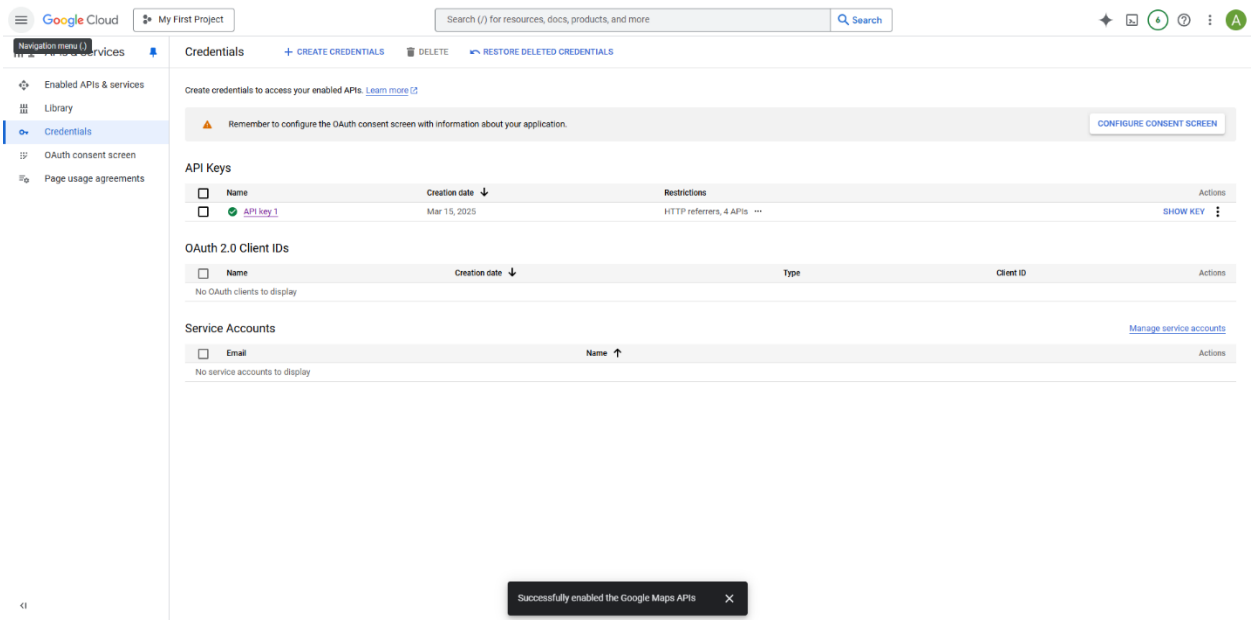
Authentication:



Created a Google API key to access Google Maps/display map locations for support organizations.



Google API key credentials:



Code Development:

Navigation bar functions as a scaffold across all pages. Updates dynamically when the user switches tabs within the navigation bar.

```
<nav id="navbar">
  <ul>
    <li class="home-li"><a class="active-link" aria-current="page" href="homepage.html">Home</a></li>
    <li><a href="about.html">About</a></li>
    <li><a href="contact.html">Contact</a></li>
    <li><a href="employment.html">Employment Opportunities</a></li>
    <li><a class="accent-link" href="FindHelp.html">Find Help</a></li>
  </ul>
</nav>
```

Embedded API key and map development. Use of organization constants to provide real/existing information regarding organization services and aid with location and map that provides easy access to directions and location relative to user.

```
document.addEventListener("DOMContentLoaded", function () {
const apiKey = "AIzaSyBxd1msfLAIJORFTSIu0wsqOl8aL9w5Smw"; // GOOGLE API KEY EMBEDDED TO THE SCRIPT FOR DEMONSTRATION PURPOSES ONLY.
    /* This script populates a dropdown with organizations that provide help for homelessness and related issues.*/
    /*When an organization is selected, it updates the displayed information and the map.*/
    /* The organizations are hardcoded in an array for demonstration purposes.*/
    /* The map is displayed using an iframe with a Google Maps embed link.*/
    /* The script also includes a function to update the organization details based on the selected option in the dropdown.*/
    /* The dropdown is populated with the names of the organizations, and when an organization is selected, its details are displayed in the respective HTML elements.*/
    /* The map iframe source is also updated to show the location of the selected organization.*/
    /* The script is executed when the DOM content is fully loaded to ensure all elements are available for manipulation.*/
    /* The organizations array contains objects with details about each organization, including name, address, services offered, help provided, website link, and map URL.*/
    /* The organizationns are real and exist in the Seattle area, providing various services to help individuals experiencing homelessness and related issues.*/
    /* The map URLs are Google Maps embed links that display the location of each organization on a map.*/
    const organizations = [
        {
            name: "Seattle Homeless Outreach - Non Profit Organization",
            address: "12345 Lake City Way NE, #177 Seattle, WA 98125 USA",
            services: "Food distribution, temporary shelter, hygiene kits, clothing assistance, and job search support.",
            help: "Seattle Homeless Outreach provides immediate relief by distributing warm meals, clean clothing, and essential hygiene kits to those experiencing homelessness.",
            website: "https://seattlehomeless.org",
            mapUrl: "https://www.google.com/maps/embed?pb=!1m14!1m8!1m3!1d10761.405491471223!2d-122.313756!3d47.599857!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x54906abf87de4ad7%3A0x3f26dae79363b5
        },
        {
            name: "Mary's Place - Non Profit Organization",
            address: "100 23rd Ave S, Seattle, WA 98144 USA",
            services: "Emergency shelter, housing navigation, employment training, childcare services, and family support programs.",
            help: "Mary's Place offers safe, temporary housing for women, children, and families experiencing homelessness.",
            website: "https://www.marysplaceseattle.org",
            mapUrl: "https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d2696.139117677197!2d-122.353815323663 34!3d47.487203171179 08!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x549

        },
        {
            name: "King County Regional Homelessness Authority (KCRHA) - Non Profit Organization",
            address: "700 5th Ave, Suite 2300, Seattle, WA 98104 USA",
            services: "Coordinated entry for housing, outreach and engagement, policy advocacy, and mental health support.",
            help: "KCRHA works to unify and improve homelessness response efforts across King County by providing streamlined access to housing resources.",
            website: "https://www.kcrha.org",
            mapUrl: "https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d5380.484257784394!2d-122.3282071!3d47.60190140000001!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x54906aba49d
        },
        {
            name: "Department of Corrections Washington State - Non Profit Organization",
            address: "7345 Linderson Way SW, Tumwater, WA 98501 USA",
```

HTML <iframe> element that is used for embedding an external website or web content - a Google Maps view

```
<h2>Map Location: </h2>
<iframe id="mapFrame" width="600" height="450" style="border:0;" allowfullscreen="" loading="lazy" referrerpolicy="no-referrer-when-downgrad
```

JavaScript and Firebase handle authentication state changes: retrieving the user ID from local storage to fetch and display user data from Firestore. If the user is not logged in, the screen remains empty.

The logout button clears user data, redirects to the login page, and resets local storage. Firebase restores access to the homepage when login is completed.

```javascript
onAuthStateChanged(auth, (user) => {
    const loggedInUserId = localStorage.getItem('loggedInUserId');
    if(loggedInUserId) {
        console.log(user);
        const docRef = doc(db, "users", loggedInUserId);
        getDoc(docRef)
        .then((docSnap) => {
            if(docSnap.exists()) {
                const userData = docSnap.data();
                document.getElementById('loggedUserFName').innerText = userData.firstName;
                document.getElementById('loggedUserEmail').innerText = userData.email;
                document.getElementById('loggedUserLName').innerText = userData.lastName;
            }
            else {
                console.log("no document found matching id")
            }
        })
        .catch((error) => {
            console.log("Error getting document");
        })
    }
    else {
        console.log("User Id not Found in Local storage")
    }
})

// Logout Functionality
// This function handles the logout process when the user clicks the logout button
// It removes the user ID from local storage, signs the user out of Firebase, and redirects the user to the index.html page
const logoutButton = document.getElementById('logout');

logoutButton.addEventListener('click', () => {
    localStorage.removeItem('loggedInUserId');
    signOut(auth)
    .then(() => {
        window.location.href = 'index.html';
    })
    .catch((error) => {
        console.error('Error Signing out:', error);
    })
})
```

Word Count: 684