1. **Login with Google**: When the user visits the webpage, they'll see a link to "Login with Google". Upon clicking this link, they are redirected to a Google login page. The user enters their Google account credentials and authorizes the website to access their basic profile information. The server then receives an OAuth token from Google, which it can use to fetch the user's name and email.
2. **Displaying User Info**: Once the user has logged in, the website fetches their user data from the server. The server sends back the user's name and email in JSON format, which is then displayed on the webpage.
3. **Entering a City and Preference**: The user is provided with an input field to enter a city name and a dropdown menu to select a type of place (e.g., bar, cafe, museum, etc.).
4. **Fetching and Displaying the Map**: After the page has loaded, Google Maps is initialized with a default location. Once the user enters a city and selects a preference, and clicks the "Search" button, the Google Maps display updates to center on the entered city, and markers are added to the map to show places of the selected type in that city.
5. **Displaying Places**: Additionally, these places are listed below the map. When the user clicks on a place in the list, an alert box pops up with detailed information about the place, including its name, address, rating, and website.
6. **Fetching and Displaying the Weather**: When the user clicks the "Search" button, the website also fetches a weather forecast for the entered city from the OpenWeatherMap API. The forecast for a selected date is displayed below the map. The weather forecast includes a recommendation based on whether it's raining, snowing, or clear, and the temperature range.

To summarize, this website allows a user to log in with Google, enter a city and a type of place, and see a weather forecast and a list of nearby places of that type in the city. The places are displayed both on a map and in a clickable list that shows more details about each place.