



cafFIEND

Project Presentation



Features

1. Feature 1: photos & Google Maps API
2. Feature 2: recommended page
3. Feature 3: profile & update profile page

Login & Signup: Frontend

- The Login and Signup page utilizes dynamic elements that allow the user to choose between signup and login.
- This was done through implementing the functionality between both the signup and login page into one page.
- Additionally, adding prompts that allow the user to know when they have an account already, or if they are not meeting the requirements for signing up.
- The login page is linked to the homepage, so when the user signs up or logs in, the next page becomes the homepage.

Login & Signup: Backend

- Signup checks for existing users with the new email before creating a new user.
- Login makes sure the password matches with the email.
- When the user signs into the website, the website generates a token that corresponds to their email.
 - Anytime the website needs to access the user data, it dereferences the token to get their information.
 - The token is stored in a new sessions database.

Homepage

- Frontend
 - Created a grid to display multiple tiles ~ each one being a coffee shop with an image and details: ratings, location, features etc..
 - Created useful interface to query the database, with a search bar at the top where you can search for coffee shops by name.
 - Created a filter bar which used a slider for location, dropdown for average rating, price, and the number of reviews and checkboxes for bathrooms and wifi.
 - Both the filter bar and search bar were methods for user to conveniently query
 - Using the filter bar or the search bar the user could dynamically change which elements appeared on the grid with cafes not meeting user requirements disappearing temporarily while filters were applied meeting requirements for the displaying dynamic data.

Homepage Backend

- Retrieves based on queries made by user (searching names of coffee shops or applying filters)
 - E.g., each time user types character in search bar we retrieve coffee shops via a regex search into the MongoDB database
 - Have similar retrieval for other filters for noise, cost, location, studyability, wifi, etc
 - Dynamically displays coffee shops when users interact with search/filters
- We store a shop schema to store features about coffee shops and retrieve based on the schema
 - For location we were only storing latitude and longitude so I needed to use Haversine formula to convert it to miles so users can easily filter by distance from UCLA

Template: Photos & Maps

Feature 1: photos & Google Maps API

- Used the coffee shop image url to display the photo for that coffee shop
- Used latitude and longitude from each coffee shop to display the location
- Displays dynamic data to the user
 - Only one template page, but shows info for each individual coffee shop



Template Backend

- Ratings: allowed user to give a rating from 1-5 about the coffee shop on the coffee shop page
- Reviews: also gave them option to submit a review about the shop
- After rating and review is submitted (if rating is outside range then they get an error message), the database is updated with the new review
- Because the new rating was submitted for that shop, when the page is refreshed then the past reviews section is updated and displays the ratings and reviews of that shop on that page

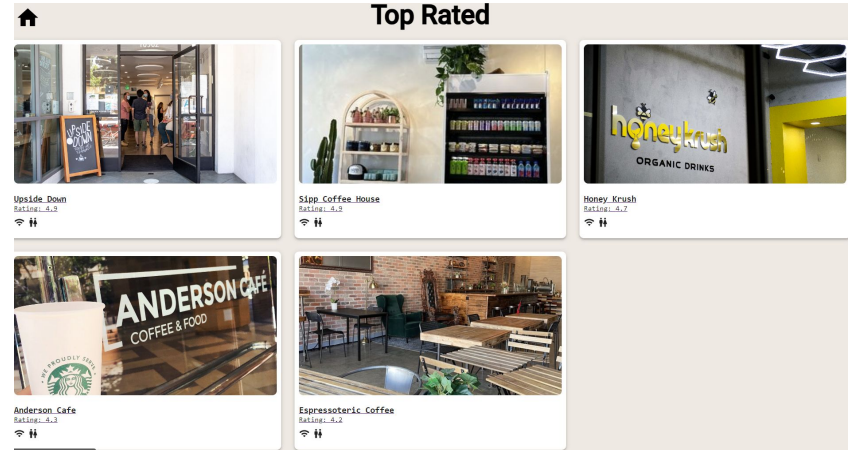
Template Frontend

- For the frontend, I used MUI styling create the containers and boxes for the features
- I also imported icons for each feature so they matched the other pages and made the page look better
- I created the frontend of the ratings and reviews as well so there's an up and down button that someone can change the ratings for
- I also made the past reviews display in a sliding box and so that the rating for each review is displayed in stars instead of a number

Top Rated

Feature 2: recommended page

- Did a retrieval from the database to get the top 5 rated coffee shops based on average rating
- Displayed the coffee shops in grid format like the homepage
- Some future possible additions:
Adding more user customizability to the recommendation



Profile

Feature 3: profile & update profile page

- Displays the user's information.
- Also there is logout functionality that deletes the token from the website's local storage and deletes it from the website.
- Created a slider to have a history list to not take over the page.
- The update profile page allows the user to:
 - Update name and email.
 - Update password: user must correctly enter their current password to update their new password.
 - This would update the database.