Places Nearby Web Project

Monday, October 04, 2021

Increment 2 Report

Team Members

Kalyani Nikure

Vineeth Reddy Sheri

Piyush Narhire

Yousef Almutairi

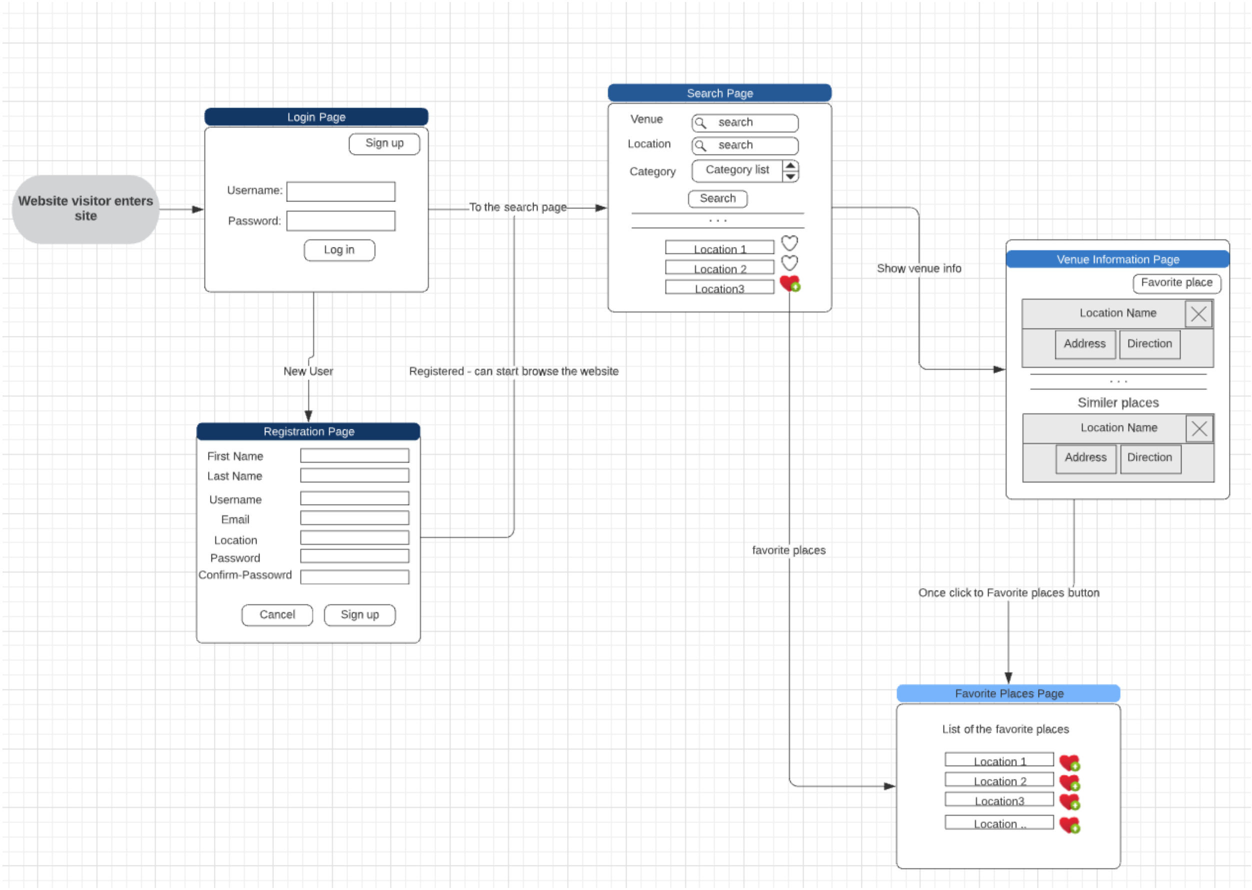
Background

Our intent is to develop a MEAN stack web application for finding places nearby and presenting them to its users. Our application aims to satisfy the needs of those who are looking for a simple solution for searching and saving the places they wish to visit. Any newcomer to the place would find this application useful for exploring nearby locations and managing in the profile these places for future reference.

Ideally, we'd like to save the login information of registered users as well as their favorite places in a NoSQL-like MongoDB database. This is an open-source database that stores data in JSON format.

Project Workflow

We attempted to create a simple and user-friendly workflow for those landing and signing up for our application for the first time and as well as registered users. Based on the various application flows used by the users, we document a total of 5-page layouts.



The figure above depicts the project workflow for our application.

Login Page

On this page, the user will be able to log in to the website using his/her username and password.

Registration Page

This page is for new users to create a new account.

Search Page

Users can search by venue name, location, and the category dropdown list to narrow the search results.

Venue Information Page

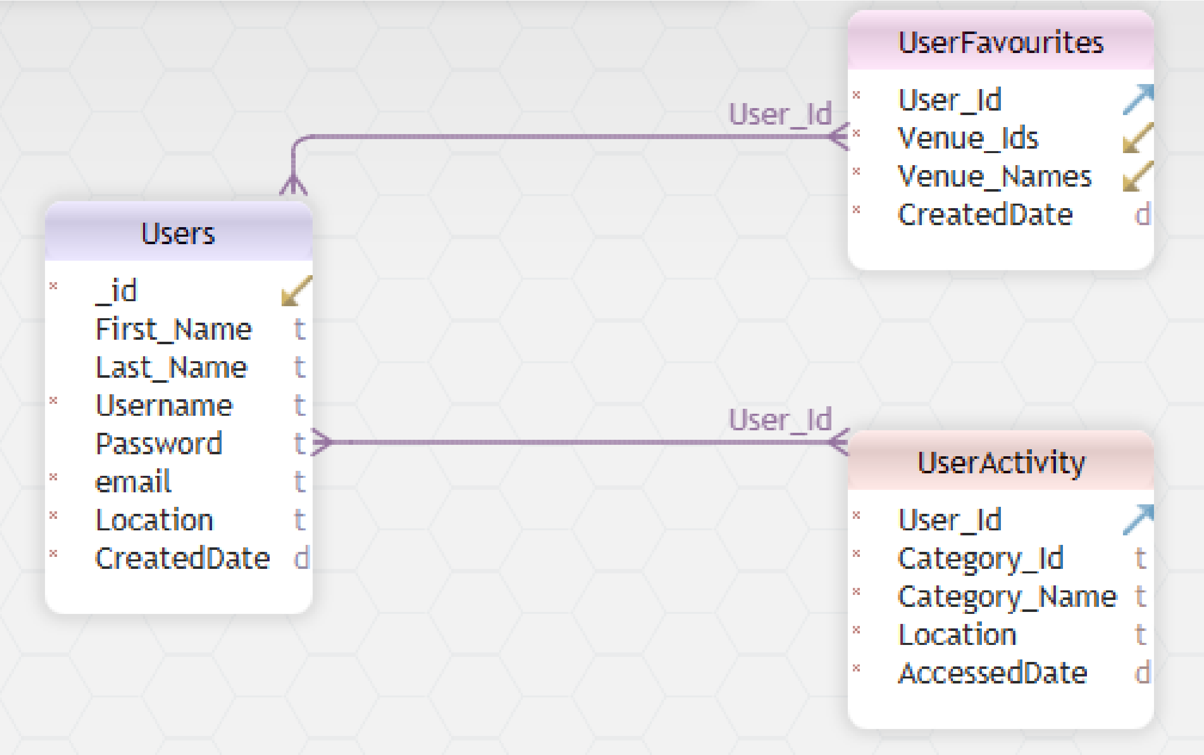
Users will get detailed information in regard to the selected venue on the previous page.

Favorite Places Page

This page will show all the favorite/saved places.

Database Design

The Figure Below shows the tentative database design for the MongoDB database for our application.



The important components are discussed below-

1. Users Document

The referred document holds the most important information data for our application. It will contain all the data User enters while doing the registration process. Personal information such as First Name, Last Name, Location, and email will be saved while creating a unique id for each user. This id would be referred to in other dependent tables to refer to a particular User along the way.

2. UserFavourites Document

The User's favorite places are stored in this document in relation to his/her registered login. Venue_Ids are unique to each venue name that is retrieved from the Foursquare API response. The multiple venues a user may store can be collected in the Venue_Names array with its corresponding Venue_Ids array. The array-style information makes it easy to map and retrieve data from JASON formatted documents with the unique User_Id of a particular user. These unique _ ids in the User document would be a foreign key for the UserFavourites document.

3. UserActivity Document

When a user performs a search activity on the web application we developed, we would like to record this information, which can be extended further to perform analytics tasks on the data collected for future use. Using this data, user profiles can also be used to suggest exciting places to users.

Data entered by the user on the Search Results Page while making a search is logged in the database mapped to the User_Id contained in the parent User document.

API's Used

FourSquare API

Foursquare API has an enormous database on the places/venues that are related to different categories. The API provides information like Name, Address, popularity, photos, tips, category, and also a summary regarding the venue/place. The API has a total of 9 categories at top-level, 281 categories at level two, and 122 categories at level three.

The **top-level categories** are:

1. Arts & Entertainment
2. College & University
3. Outdoors & Recreation
4. Professional & Other Places
5. Residence
6. Shop & Service
7. Travel & Transport
8. Food
9. Nightlife Spot

The top three categories that have more venues under them are Professional & Other Places, Shop & Services, and Food. The Foursquare API is free and easy to use. It provides different endpoints to deal with different scenarios.

- The **/categories endpoint** returns a list of all the categories. In the search filter option, this endpoint displays the categories that can be helpful to narrow down the search scope.
- The **/suggestcompletion endpoint** returns a list of locations that partially match the search query and the provided location. Similar to Google's search feature, this endpoint will provide all the possible places names to the user based on the partial name entered.
- The **/explore endpoint** is used to fetch the list of places based on a search query and location. It displays the venues on the Search Results page.
- We use the **/similar endpoint** to get information about places that are similar to the provided one or that fall into one of their categories. When a user visits the Place Details page, this endpoint will be used to display all similar places as well.

Yelp API

Yelp is another Application Program Interface that is used to communicate between software using data points to provide developers with their custom applications or websites. This comprises information such as a company's ratings, pricing, or category. Yelp API is also called Yelp Fusion API because the data is organized into categories (sometimes called endpoints). These endpoints enable getting a large amount of data from a single or two requests more easily. There are seven key endpoints in Fusion which are listed below and each of them provides a useful listing of data.

1. Business Search
2. Phone Search
3. Transaction Search
4. Business Details
5. Business Match
6. Reviews
7. AutoComplete

For example, the first is Business Search where it will search for businesses using keywords, categories, locations, pricing ranges, and other criteria.

So, in our Places Nearby Web project if we search, for example, Nelson-Atkins Museum the Yelp API will provide all of the details regarding that particular museum.

References

- https://docs.mongodb.com/manual/core/data-model-design/
- https://developer.foursquare.com/docs/places-api/
- https://lucid.app/
- https://www.reviewtrackers.com/blog/yelp-api/
- https://www.yelp.com/developers/documentation/v3/get_started
- https://spectralops.io/blog/yelp-api-guide/