Group #10

Band Together

Technical Report

By: Abhi, Adam, Faezah, Gavin, and Jason

Table of Contents

Overview

1

1.1 Project Purpose

This website is created with the purpose of enabling users to find concerts in any location from artists and bands they enjoy. It is hosted on the domain https://bandtogether.events.

1.2 Models

The information on Band Together is divided into three main models. Artists, Concerts, and Locations. The models are defined and related to each other in the following ways. Location pages define places where concerts and artists will be, additionally, the page gives a short description of the location and allows the user to visit the page for that location, as well as visit the artist and concert pages relevant to the location. Artist pages allow users to find artists they want and track where and when they will be performing. The Artist pages also display information and allow them to follow the links for the locations and concerts they will be performing at. Finally, Concert pages show the user what concerts are coming up and displays the artist, city, and venue. From the Concert pages, the user can view information on the concert and the artist.

1.3 Technical Specs

The front-end of the site is done mainly through ReactJS. Routes control the paths for the site and allow you to simply input the pathname to link to any of the pages. There are some dependencies imported into the React App, you can download them once you have the file and are in the band-together directory by running npm install with node. This should download everything you need to run the app locally using `npm run dev`. To import any new components or tools that you wish to use, create the import statement and use `npm install <input>`, this will update the requirements and allow anyone to get it once they run `npm install` again.

1.4 File Structure

From the website repository, the initial folder contains a `README` with some information if you want to get started faster, but here we will be going more in-depth. You can see there are three folders and three package files. The folders in no particular order are `dist`, `src`, and `node_modules`. For the most part, you can ignore the `node_modules` folder as it is only to hold build tools that are installed from `npm`. The `dist` folder's files can also be mostly ignored.

What is important about `dist` is that it holds the resources used on the website, such as the splash art or the front page image. The `src` folder is the main code folder, it holds two folders, `layout`, and `pages`. The `layout` folder holds the JS code responsible for the main site-wide navbar in `navbar.js`, the code for an image slideshow in `Slide.js`, and the code for special element boxes in the `Card.js` set of files. More in-depth guides on these files will be covered later on in this report.

1.5 Resources

All image resources for Band Together are stored in the `images` folder, which itself is contained in the `dist` folder. All images or other media that are not loaded dynamically from some other website can be found here.

Main Application

2

2.1 Amazon Web Services (AWS)

The website is currently being hosted by Amazon Web Services. The site is stored in an S3 bucket and distributed using CloudFront.

2.2 Postman

Postman is used to create a RESTful API that we can use to supply information to the frontend from our database.

2.3 GitLab

Gitlab is the source for version control within the app. The link to the GitLab is here https://gitlab.com/Adam-Bomb/band-together

2.4 App.js

The App.js file contains the Switch for our React Router. Inside the Switch are all of the different pages and their respective urls.

2.5 index.js

The index.js file renders the app. It uses a React Router so that we only have to host the index.html and bundle.js files in our S3 buckets.

List Pages

3

3.1 Overview

All ‘List’ pages are accessible from the `src/components/pages` folder. Each file is saved as `<model>ListPage.js` where `<model>` is the model the page is for. The ‘List’ pages are pages directly accessible from the site’s navbar. These pages are the places where users can browse all existing instances of each model. Clicking on an instance will take you to its related Instance page. All of these files are currently stored in the `src` directory.

3.2 Concerts

The list for concerts uses instances `ConcertCard.js` to display its information. The main info listed is the concert name (often just the name of the artist/band), the city the concert is taking place in, and the venue of the concert. There are additional links in the concert’s card that link the user to either the concert’s instance page, the artist’s instance page, or the location’s instance page. The time and date of the concert are also displayed on the card. For more information on ConcertCard.js, see *Section 5.3* of this document.

3.3 Artists

The list for artists uses instances `ArtistCard.js` to display its information. The main information listed is the name and a picture of the artist, the genre of music, year started, their record label, hometown and their most popular song. Additionally, there are links to upcoming concerts and their respective locations. For more information on `ArtistCard.js`, see *Section 5.3* of this document.

3.4 Locations

The list for locations uses `LocationCard.js` to display its information. The main information shown is an image of the location, the name of the city which also serves as a link to that city’s page. Additionally, there is information given on the population size of the location, major concert venues, the main airport of the location, and the per capita crime rate of the location. For more information on `LocationCard.js`, see *Section 5.3* of this document.

Instance Pages

4

4.1 Overview

The instance pages are where the bulk of the information is displayed. They are what the list pages point to when a user wants to see the individual item in the list. The instance pages are generated by inputting information to their `<model>DetailPage.js` where `<model>` is the model the instance page is for. The information is input in the corresponding list for that detail page and is generated based on what item the user has selected.

4.2 Concerts

Concert instance pages will display the following information in a table.

- Style of music
- Artist
- Venue
- Average ticket price
- Number of seats available (Capacity)
- Address
- Starting Time
- Number of tickets left
- Noise level
- Whether the venue is indoors or outdoors
- Whether alcohol is sold on the premises
- The weather forecast for the day of the concert

4.3 Artists

Artist instance pages will display the following information.

- Venues they will be playing at
- Cities they will be touring
- Year they started
- Music Genre
- Albums
- Top played songs
- Members if they are a band
- A short bio about the artist
- Related Artists
- Record label

4.4 Locations

Location instance pages will display the following information.

- City
- Country
- Different venues in the city
- Population
- Crime/safety rating
- Food Availability
- Nearby hotels
- Airports
- Public Transport Availability
- Weather forecast

Miscellaneous Files

5

5.1 Navbar.js

This file is located in the `src/components/layout` folder. The navbar allows the user to traverse to any of the main pages at any point. Clicking the icon in the left corner returns the user to the main page. The navbar is able to be seen on every single page of the website with no exceptions. The code itself is relatively straight-forward, as it is simply a rendered set of links passed into `<a>` tags, with an image from our `images` folder being displayed as the logo.

5.2 AboutPage.js

This file is located in the `src/components/pages` folder and contains information about the developers including an image, bio, number of commits to the GitLab and the number of issues they have solved. It dynamically pulls this information from the GitLab API. More information on using the GitLab API can be found here: https://docs.gitlab.com/ee/api/ and will not be covered in this document.

5.3 Cards

As mentioned earlier in this document, we have a set of `<page>Card.js` files in the `src/components/layout` folder where `<page>` is the page (like the 'About' page) utilizing the Card displays. These are simple, compact containers for displaying information and the only major difference between each file are the React props they are expecting.