Friendly Beads

Alexandra LaMear

Problem Description

My project is called Friendly Beads, which is an online beaded friendship bracelet designer. Right now, the only way to design these beaded friendship bracelets is either with paper or a notes app on your phone. Friendship bracelets have become increasingly popular in the past year, with people trading them showing up at a variety of different events, from concerts to Formula 1 races. Their popularity in such a vast number of events shows just how diverse the user base for such an application could be. The number of different personas is a wide range from people newer to the hobby to people who are highly experienced at it. The targeted personas would be on the newer to moderately experienced end of the spectrum. Designs can get quite complicated as users become more experienced and the goal would be to share ones that could be made widely.

Friendly Beads would solve both these problems by being a concentrated location to create and share various beaded friendship bracelet designs. They would be able to input information about the types of beads they wish to use and other details about the bracelet, as well as tag the design with various categories that it relates to. Users would then post these designs as they are saved. After a user has made a bracelet, they would be able to mark the bracelet as made and upload an image of the final product. Users would also be able to search the various designs and save them for later use, being able to see the details of the designed bracelet. After making and saving designs, users would begin to receive targeted recommendations for designs of bracelets.

MVP

Features

My program will need various features to support the user's needs to solve the problems. The user will need to be able to create an account and log in so that they can save and access the designs at a later time. There will also be a function that allows them to create a bracelet design and upload a photo of the finished product, as well as edit the design at a later time. Another feature is the ability to search and view other people's bracelet designs based on tags and titles. The users will want to be able to save these designs they find for later in case they also want to create it or are inspired by the design. As users save these designs, I plan on creating a recommendation feature to show these designs to other users and similar ones to the user who originally saved it. This means that I need to create various controllers and services through the frontend and backend programs to process and render the data as needed.

Architecture

I plan to use Ionic for my front-end rendering and API. Ionic can integrate with a number of backend programs, of which I plan on using Angular to build them. Angular will be integrated with MongoDB to handle the bulk of the data that will be collected and used by the program. Cloudinary will handle the photos the users upload as it is made to handle photographs as data in a database which MongoDB does not. Once I begin looking to deploy the software, I plan to use Amazon Web Services S3 to handle the deployment and scaling of the application.

Data

My program will need various pieces of data that will be stored as JSON values since that is what MongoDB works with. The main pieces of data will be a user's username and password as the username will be used as an identifier to be able to store other pieces of data linked to their account including bracelets made, photos uploaded, etc. As briefly mentioned photos and bracelet data will be stored in the database as well. Bracelet data includes answers to the variables such as a name for the bracelet, a description of the bracelet, a unique ID generated on the backend, the color of the beads, letter beads used, tags, as well as other pieces of information I may add throughout the development of the process. Users will also be able to save other posts for later reference which means there will be a way to store these linked to their account. There are a few other pieces of data that will be obtained for business flow purposes, not UI interaction. This includes user OS which Ionic can capture, which would allow me to direct development towards the most common device being used to access the site and see if developing a native or hybrid app would be a successful venture. Users often used and saved tags will also be stored on the backend to help with creating a referral algorithm to show users similarly tagged designs in the future.