AUSTIN EATS

Technical Report: Phase II

Group 10-3

Athul Nair
Christian Camp
Clint Camp
Mihika Birmiwal
Thomas Moore

Paper Boy
1203 E 11th St, Austin, TX 78702

Overview/Motivation

AustinEats was created with the intention of encouraging Austin residents to try more local restaurants, thus supporting our local businesses and enriching our city.

There are already so many good, well-established restaurants in the city, with more and more popping up everyday. By allowing users to sort through a list of restaurants based on location, pricing, cuisines, and ratings, AustinEats provides an easy user experience when finding their next dining experience.

After dining out, users can return to the website to find out more about their restaurant experience: AustinEats lists recipes for favorite menu items from restaurants, as well as descriptions of the cultural origins of these dishes. Through these features, AustinEats intends to be a wholly encompassing companion website for users experiencing the rich food scene of Austin, Texas.

User Stories: Received

Phase 1

1. Group menu items by culture and cuisine
    a. I am a user who is looking to try out specific menu items based on the cuisine that I'm feeling. I like that your website displays statistics for all Austin restaurants and their reviews. I would like to be able to group menu items by cuisine - for instance, show unique or highly rated menu items across all restaurants under a specific cuisine.
    b. RESPONSE: This feature is something we had planned as a feature for phase 2, where we will integrate all of our models with

each other (dynamically). This way, under the cuisines tab, it will have all the restaurants that serve that cuisine. As of right now, this is hardcoded in.

   c. UPDATED RESPONSE: We have this feature added! All of our models are now linked, so searching up a cuisine will return all the restaurants tagged with that cuisine as well. This is also no longer hard coded in, so if a new business gets added to the yelp database, then it will show up on the website.

2. Compare restaurants
   a. I am a user who is deciding between two different Austin restaurants to support. While all the data on your site is linked to one another, I want to view my top choices side by side. I would like a display option where I can compare two different restaurants and see their prices, menu items, cultures, and potential recipes.
   b. RESPONSE: I see how having the comparison feature can be useful, a lot of websites have this feature when displaying their products. We will be sure to implement this for phase 2.
   c. UPDATED RESPONSE: In our cards, you are able to see information about the restaurant and compare them side by side.
3. Most common cultures and recipes
   a. I am a user who is curious about studying the general trends of Austin restaurants. I enjoy looking up restaurants on your site and having all the information displayed for each restaurant. I would like to be able to view some aggregate statistics about what the most common cultures, cuisines, or recipes are across these restaurants, and perhaps connect those to their own regions.
   b. RESPONSE: We will allow the models to be filterable by popularity, perhaps filtering by number of reviews for the restaurant and the population for the culture. For the recipes, however, there is no way to track the popularity, so we unfortunately won't be able to implement this feature for that model.
4. Display multiple recipes
   a. I am a user who is looking to try out different recipes based on Austin restaurants. I like that your website has many filterable characteristics to show specific subsets of data regarding restaurants, recipes, and cuisines. I would like to try out a certain

difficulty level of recipes, and be able to view multiple recipes in the same difficulty category, but with different cuisines.

   b. RESPONSE: This is a feature we plan to implement in phase 3. The recipes will be filterable by difficulty. In order to display recipes of different cuisines and the same difficulty, filter only by difficulty level and not by cuisine.

5. Get directions to restaurant
   a. I am a user who is looking at the restaurant locations displayed on your website. While I can go to each restaurant's location on the map and find out how to get there, I want to see these locations more generally. I would like there to be a map showing markers where all the restaurants are located on the same map, along with my current location.
   b. RESPONSE: We will implement this in phase 3! We will have a map of all the restaurants on the model page. Thanks for all of your input!

Phase 2

1. Estimate ingredient prices
   a. I am a user who wants to recreate dishes from my favorite restaurants within my budget. I like that I can filter by the number of ingredients that I need to buy. I would like to also be able to see the prices of the recipe ingredients to check whether I can afford them.
   b. RESPONSE: This is something we can implement in phase 3. We would probably have to run another query for the prices of generic ingredients and do some math calculations for proportion sizes. We can also make it a feature to sort by the total cost of the meal.
2. Filter by ingredient
   a. I am a user who wants to find a recipe to create with the ingredients I have in my house. I like that I can filter by what cuisine I want to cook. In addition, I would like to be able to filter recipes by ingredients that I already own.
   b. RESPONSE: This is something we will implement in phase 3. We will most likely have a dropdown of the most common

ingredients and you will be able to check some of them to see recipes that use those certain ingredients. Thanks for the suggestion.

3. Sort cultures by area demographics
    a. I am a user who just moved to a new city. I want to know which cultures are popular in my area so that I can know which cuisines will be good. I would like to be able to filter cultures by the most common ones in my area.
    b. RESPONSE: This is a feasible feature. A potential way to implement this feature is to modify our databases slightly to include the restaurants' distance from the current location and then keep track of the number of restaurants tagged with a particular culture in a certain radius. I'm not 100% sure we will be able to implement this, but we will do our best!
4. Indoor/Outdoor Seating
    a. I am a user who enjoys taking my dog to restaurants with me. I like that I can see logistical information such as hours of operation and distance from me. I want to be able to also filter by restaurants with outdoor seating so that I can see which restaurants will/will not allow my dog.
    b. RESPONSE: Unfortunately, this information is not publicly available most of the time for restaurants so this feature is not feasible for our project. However, we will try to post some of the reviews from yelp, which may have information about the location of the restaurant itself. We can also try to scrape a different API for that information. If we are able to find it, we will implement it.
5. Takeout/Catering
    a. I am a user who is hosting a dinner party. I want to find out which restaurants around me I can use to cater my party. I would like to be able to filter restaurants by those which allow takeout or catering.
    b. RESPONSE: This feature is feasible and we have already accounted for filtering by this in our models and schemas! This information can be pulled from yelp directly, so we will have updated information everytime we hit the yelp API. Be on the lookout for this feature in phase 3.

User Stories: Given

Phase 1

1. I'm interested in seeing if you guys could have historical w/l records for the teams, and not just current seasons. Or if possible, being able to sort by seasons and also cumulative seasons over the history of the team. It would be interesting to see the most dominant teams historically throughout the various leagues. This is probably a Phase 2 thing.
2. We want to be able to see who is the richest player in the league? It would also be cool to see the team which has the highest paid players; which team is the richest/can afford the most expensive roster? It would be interesting to see if there's any general trends here with location.
3. It would be nice to be able to sort players by who is performing the best in the most recent games - points, assists, rebounds, defensive stats, etc. https://www.nba.com/stats has this feature but I wish that there was some visualization for which player belongs to which team - it would make the site friendlier to newer basketball fans. It would potentially be nice to also have the player's faces displayed next to their name, but this can be done towards the end of the project.
4. It would be convenient to have a search bar that you could use with a filter to sort things out. I think that it would be convenient to narrow down my searches, or at least narrow down each of the different models and have the results dynamically shrink on the website based on filter. This would make the site better to use and more fun to navigate. Probably a phase 2/3 thing.
5. When accessing a webpage that doesn't exist, all that is returned is a blank white page. It would be nice to have a clear indication, so as not to confuse the user if any content is loading, that the webpage is invalid. Perhaps a return to home button would be neat too.

Phase 2

1. I'm interested in seeing the players with a certain attribute or range of an attribute. That could mean a filter to only see players within a certain salary range, players less than x rebounds per game, and so on. I would like at least 5 attributes to filter on as that would be useful for me to explore the data.
2. Hi, I am a user trying to learn more about basketball! I would like to be able to see related information about other instances from different models that relate to the current instance I am looking at. If I am looking at a player, I can see more information about the team he played on, but right now, it is only identified with a number, not the team name. Another example is I want to be able to see what coach this player plays under as well. Basically, I want more connectivity between models and their instances so I can jump from instance to instance throughout models.
3. Hi, I am an NBA enthusiast who wants to see highlight clips from a range of players. I want to see some embedded clips, through Youtube, or anything else, that could make the website more interesting to look at besides just the raw numbers of player stats. If there was another engaging piece of media a player could have, that could work too.
4. Hello, I am a recruiter who wants to find coaches to play for the team I represent. If possible, linking a Wikipedia article to read more about who they are and what they have done in the past could help me decide if I want to reach out to them. I'm not sure how feasible it is to embed information from the article into the instance page for a coach, but it would be ideal if I did not have to leave Lowball to get this information.
5. Hi, I see that you show a set number of instances per page on the player, team, and coach pages. Personally, sometimes I want to see more information on the screen than the given 10 or so rows to have a bigger picture of the instances so I can compare them easier than always flipping between pages. If you can give me the option to set the number of rows per page between 10, 20, 30, or more (whatever is appropriate), it would make my life easier.

RESTful API

Our RESTful API is documented with Postman. Paths are defined for retrieval of data by restaurants, recipes, cultures, as well as individual instances for each model. We have yet to create a schema to depict the expected returns for calls to our API. https://documenter.getpostman.com/view/23508831/2s83tJGW4m One of our biggest challenges came from figuring out our idea and what APIs we would scrape from. We overcame this with trial and error of a few different APIs.

Models

The 3 models in AustinEats are restaurants, cultures, and recipes. Each model has at least 5 attributes, which allows the models to be filterable, searchable, and sortable. These models are connected to each other so that users can find local Austin restaurants, learn something about the culture behind the cuisine the restaurants offer, and have access to recipes of their favorite menu items to try at home.

Models	Sorts/Filters*[1]	Rich Media*[3]
Restaurants	• Alphabetic • Star rating • Which meal (b, l, d)*[2] • Open now / closed • Location (proximity to current) • Takeout/delivery allowed • Culture of origin • Review count • Price $-$$$$	• Photos of the restaurant • Photos of the menu items • Link/pdf of menu • Location on map • Yelp reviews • Link to website

Recipes	● Alphabetic ● Cooking difficulty ● Time to cook ● Which meal (b, l, d) ● Culture of origin ● Nutrition ● Price $-$$$$ ● Spice levels	● Photos of the finished recipes ● Link to original recipe ● Video tutorials ● Instructions ● Twitter feed of hashtag for recipe
Cultures	● Alphabetic ● Continent ● Country ● Language	● Location on map ● Flag (if available) ● Photos of the region ● Culture description ● Videos/documentaries on the culture

*[1]- sorts and filters have not yet been implemented - a lot of these categories will be dependent on what our API has to offer
*[2]- (b, l, d) = Breakfast, Lunch, or Dinner
*[3]- not all rich media has been implemented for Phase I

Tools

We used React to build out our front end, which dynamically displays restaurant, recipe, and culture data pulled from various different APIs.

We used Bootstrap to power our CSS functionality and design our page layouts.

Postman was used to send GET API requests to scrape data for our models.

We used Flask to build the back-end for our application. We utilized Elastic Beanstalk to host our back-end application, and created our API endpoints with AWS API Gateway. We hosted our database on Amazon RDS in PostgreSQL. We used SQLAlchemy with Flask to set up our database.

Hosting

Our website's frontend is hosted on AWS Amplify on the domain https://austineats.me. This domain was obtained via Namecheap and CNAME records were used to transfer DNS ownership to AWS.

We have automatic deployment set up with both the main (production) and develop (development) branches.

Features (Phase 2)

We set up our back-end as a Flask application, utilizing Flask-SQLAlchemy to set up our database in PostgreSQL. We started by getting a database up and running on AWS RDS, and setting up our Flask app's database URI to point to our online database. We populated our database using scraped data we sourced from our APIs stored in JSONs. We defined the endpoints for our application using @app.route in Flask. Our endpoints would first query our database based on a certain model and ID. Then, we built a Schema object based on our models with Flask-Marshmallow, and populated the schema with our database query. We then dumped the schema out into a JSON which we served to the front-end dynamically to display our data.

On the front end, we had already set up our cards to work dynamically based on JSON data, so all we had to do was clean up our pages and set up pagination. For pagination, we allow users to select how many instances per model page they would like to display on their device. We used the useTable and usePagination components from react-table to implement our pagination. This allowed us to divide our model instances into different pages which the user could navigate between to find different instances.