Team QuickEats
Samantha Ngo, Myles Austin, Troy Daniello, Rumman Al-Karim
CSCI 49900 - 03 | MonWed 3:10-5PM
2021 May 19
Professor Grezes

QuickEats: Street Food Near Me
Final Presentation Transcript

Prep: Since it takes a while to start up, Myles will have the demo already running in the background. This will be done roughly 5-10 minutes before the presentation starts. She will be running the application locally as we have not deployed it yet. Myles will also be the one changing the slides to prevent technical difficulties and/or delays caused by Zoom transitions. He will also have the code open onto the main branch of our repo for any code related questions and references.

On Slide 1 (Title Slide) - QuickEats:
Rumman: Hello, everyone, nice to see you all again! My name is Rumman Al-Karim.
Samantha: My name is Samantha Ngo.
Myles: My name is Myles Austin.
Troy: And I'm Troy Daniello.
Rumman: Welcome to our final demo of QuickEatz! Before we go and show you what we have created let us refresh what our goal was.

Move to Slide 2 - Product Definition / Solution. On Slide 2:
Rumman: To review, let me remind you of our product definition and goals. Many New Yorkers including students have very busy lives and often don't have a lot of time during breaks to find good and affordable food near them. Many good and cheap food trucks also exist in various areas of the city but can often be difficult to find and therefore tough to attract New Yorkers. What our product is attempting to do is to solve this issue by using a feature where customers and food truck vendors can access a cohesive service and interact together. In this service customers can find a food vendor that they like conveniently and where vendors can easily display their business's information to attract customers.

This can be accomplished by letting food trucks display their location near a user's proximity. This enables users to be able to find cheap, fast and delicious food operated by local vendors and empowers both students and professionals by saving them time and money. This will also bring more awareness to these local vendors.

Move to Slide 3 - Demo. On Slide 3:

Myles: Now it's time for the demo!

Myles moves from the presentation to the localhost QuickEatz site. He starts on the public Homepage, with the login and create account button.

Myles: I’m going to demonstrate the vendor side of QuickEatz. Let’s start by creating an account!

Fills in details, announcing them as they go.

Myles: Now that we’ve created an account, let’s log in.

Myles: Here we have our dashboard. We can update the location of our food cart! The vendor defaults to a specific spot, but we can quickly change this.

Myles clicks the map to create a marker.

Myles, while clicking “Update” button: When we hit update, we can see our position change to the spot we clicked!

Myles, going to Profile through the top right: We can check our profile and tweak information, like opening our vendor for the day, or adding a menu item! Let’s do both. First let’s add a simple item. A hotdog with ketchup for 2.00.
Myles clicks “Edit Profile”, then adds the menu item details as described, and clicks the add button.

Myles: And then open up for the day!
Myles scrolls down and clicks the button to change the vendor’s open status.

Myles: Now, let’s head back to our profile.
Myles clicks “Back to Profile”

Myles: We can see our updated information! While we’re at it, let’s check out some other vendors and how they’re doing, too. An easy way to do this is to head on over to the Trending page!
Myles clicks the ‘Trending’ button

Myles: We can see the top rated vendors, as well as click on them to see more information.
Myles clicks on one of the vendors.

Myles: If we want to, we can use more precise methods of finding vendors, but we’ll save that for the customer demonstration.
Myles returns to the dashboard.

Myles: And before we move on to the customer side of things, let me show you what a vendor that has reviews and a rating would look like
Myles logs out, and logs back in as a different vendor, fruit8@lemons.com.

Myles: We can see the dashboard and profile as before, except this time we have handy information!
Myles shows off the dashboard and profile pages for a bit, and logs out.

Myles: Now, I'll hand it over to Samantha to demonstrate the customer account.

Myles stops sharing screen. Samantha starts sharing screen and is already open to the localhost QuickEatz app public Homepage.

Samantha: Okay, let's create a new customer account.

Clicks on "Create Account" button and navigates to the Customer Create Account Form.

Samantha: So, I'm going to register Spongebob Squarepants, who's checking the Krusty Krab's NYC competition, just in case they decide to expand there.

Fills in Spongebob's info, talking through every step. Also demonstrates email, username, and password checks as well as submit button disable.

Samantha: My username will be bestfrycook and my password will be krabbypatty. Now, we can create an account.

Clicks "Create Account."

Samantha: Now let's log in.

Navigates to the login page, types in info for the new account:

- ***spongebob@gmail.com***
- ***krabbypatty***

Samantha: And we can now use the app! You can click on a location and find nearby vendors and their information.

Demonstrates nearby vendor search.

Samantha: We can also see trending vendors and all the vendors in the database and sort them.

Demonstrates trending and all vendors page.

Samantha: If we want to see more information about a vendor, we can go to their vendor page and check it out. There are many ways to do this.

Navigates to a random vendor page via trending, nearby vendors list, and pin marker.

Samantha: If we've visited a vendor, we can leave a review on their page and see it on the vendor's page and our dashboard.

Makes up a random experience and review/rating. Then, navigate back to dashboard and vendor page.

Samantha: We can also check out our profile and update any information.

Navigates to profile page and demonstrates edit profile.

Samantha: Finally, we can log out and that's about it, Spongebob now has a place to do all his spying.

End of Demo. Samantha switches back to presentation.

Move to Slide 4 - Project Overview. On Slide 4:

Troy: So moving onto our project development, starting with the differing technologies that we used, we decided to learn and use the MERN stack which combines MongoDB, Node.js and React.js and Express js of which the last we didn't use. MongoDB is quite different from SQL based database systems which we will go into more detail soon, but it took us some time to learn the differences and how to utilize Mongo collections which are the equivalent of tables for our purposes. We also all needed either to refresh or learn React.js and learn JSX.

However, we soon decided that it was better to use Next.js instead of regular React.js which is based off of it but also enables server side rendering as well as generating static dynamic websites.

Also, as we started creating the front end design of the application, we decided it would be better to use tailwind css which would better organize our css into one file and allow for the configuration and reusing of css components in desired ways which we will also go into more detail about later.

Once the account system was better established, we needed a way to encrypt login credentials so we utilized Bcrypt to ensure that passwords were properly encrypted for security purposes for each account.

Finally, we utilized and integrated a Google Maps API as our goal for the project was to have searchable vendors which the locations thereof could be easily displayed to potential customers who wanted to know more about specific vendors via our app.

Moving onto some of the challenges we faced, one was figuring out the extent and scope of tasks and the time that should be spent on each, another issue was the dealing of the account system and sessions management.

The solutions we came up with for each were doing things early on and dividing tasks amongst ourselves. We also used local storage and tokens to get around the sessions management issue as we were not able to get the former working.

Move to Slide 5 - QuickEatz Questions, Comments, Suggestions . On Slide 5:

Rumman: That concludes our QuickEatz presentation. Are there any questions, comments, or suggestions about our QuickEatz app before we move onto the technical topics?

Pauses for Q&A, up to 3-4 minutes total.

Start of Technical Topic Presentation 1 - MongoDB.

Move to Slide 6 - MongoDB (Title Slide). On Slide 6:

Myles: Now, we'll move on to our first technical topic: MongoDB! Once again, my name is Myles Austin.

Rumman: And I am Rumman Karim and today we will explain the basics about MongoDB.

Move to Slide 7 - What is MongoDB? On Slide 7:

Rumman: MongoDb is a document database that provides you with the scalability and flexibility you need. It allows users to store data in flexible, JSON like documents. To be specific, it uses BSON, which is a binary format of JSON with additional data types. It, like other databases, also features indexing, real time aggregation, and Ad-Hoc queries, or queries that are considered short lived because the contents of the query and database are unknown ahead of time.

Move to Slide 8 - What makes MongoDB unique? On Slide 8:

Myles: What makes MongoDb so unique? MongoDb utilizes a flexible Data Model, being a NoSQL database, and has an expressive Query Syntax. NoSQL databases store content differently from the more traditional relational databases (often abbreviated RDB or RDBMS). In essence, they store information in documents that may or may not be directly related to one another, and which in turn are placed in collections. More specific and niche differences will be explained later, but the gist is that NoSQL databases can be expanded relatively easily, without needing to worry about maintaining relationships. A user can opt to mimic RDBMS structures if they choose, but this is not required. "Expressive Query Syntax" refers to the number of expressions/operators that aggregate information that can be used inside the queries and the database itself.

MongoDB possesses a relatively small learning curve and well organized documentation, such that anyone interested enough can use it. In short, MongoDB comes with various default features that make getting started or creating new items simpler. For example, each document in a collection is automatically given one pre-indexed field (called '_id') that is guaranteed to be unique among all other documents in that same collection. This way, the user can use most of the more basic database techniques without needing to concern themselves over setting a primary key or keys. Again, if their situation calls for it, they can add more indexes or remove/ignore this field, which adds to MongoDB's flexibility. Moreover, specific JSON/BSON types can be used to make non-trivial operations that would otherwise require the user to make aggregations or calculations on their own. Geospatial queries are particularly noteworthy (and well documented), in that they allow for the distance between two points, as specified within documents, to be calculated, with a number of additional parameters (e.g. 2d vs spherical, all points in a radius, etc). In addition, there are services that can aid the user in other ways.

Move to Slide 9 - Additional Services. On Slide 9:

Rumman: Mongo DB offers two helpful applications for the average user's needs. One of them is Mongo Atlas, which offers cloud-based database hosting. There are several plans involving the pricing of clusters available, with clusters serving as the deployment(s) of databases. The free tier and smaller tiers are great for small projects and proof-of-concept applications, and the larger, more expensive options are there for users ready to make their creations more official. The other helpful application provided by Mongo is Mongo Compass. This is a handy GUI that allows you to connect to your database and modify validation schema, collections, and other data. For users more comfortable with terminals, Compass comes equipped with a shell for command line queries. A validation schema provides users with more definitive control over what is inserted and updated in your collections. This feature uses a JSON/BSON script to warn or even outright deny a user from adding or updating anything that goes against it. Combined with expressive query syntax, this user has extreme control over what enters a collection. For instance, a user can deny documents that lack certain fields. The opposite is true: a user can accept documents that meet a criteria, but have additional fields alongside it.

Move to Slide 10 - Differences between RDBMS and MongoDB. On Slide 10:

Myles: As far as the differences between a RDBMS (more specifically, Relational Database Management System) and MongoDB go, the decision regarding which to use depends largely on user preference, the task at hand, and the resources available. As mentioned before, RDBs follow the relational data model, where MongoDB uses a non-relational or NoSQL model. If a user can learn beforehand what their data will look like, they can decide which to use accordingly. Regarding scalability, RDBs tend to be vertically scalable while MongoDB is horizontally scalable. Vertical Scaling refers to improving a database by adding more CPU/RAM to the machine(s) hosting it. RDBs tend to keep nodes/data together and process it in one place. Horizontal Scaling refers to improving the database by adding more machines, as NoSQL databases tend to partition nodes into pieces, and split logic in order to divide and conquer. An instance of this can be seen in Mongo Atlas, as it offers cluster sharding. This results in the NoSQL databases being more efficient overall regarding large amounts of frequently added or modified data, as more traffic can be handled. However, because NoSQL databases like MongoDB often value availability and speed to consistency, issues regarding duplicate data or inconsistent documents become relevant considerations. On the other hand, RDBs are the more consistent models of the two with regards to data, and in turn, are generally slower than NoSQL when it comes to reading and writing, although this can vary depending on the exact database and context.

Move to Slide 11 - Differences between MongoDB and MySQL . On Slide 11:

Rumman: Here we have a brief summary of the differences between MongoDB and MySQL. SQL has been an inexpensive option for businesses around the world that need a relational database. However as the variety and the volume of data has increased, non relational databases like MongoDB have arisen to meet the new needs for data. (Read the slide)

Move to Slide 12 - Why MongoDB is the best option. On Slide 12:

Myles: Why was MongoDb the best option for us? MongoDB is the best option for you if you want to:

Represent data with variability over time, or variability in its structure.

Scale to high levels of read and write traffic.

Scale your data repository to a massive size.

Evolve the type of deployment as the business changes.

Store, manage, and search data with text, geospatial, or time series dimensions. This was in line with a lot of our goals for QuickEatz, so it was an effective choice!

Move to Slide 13 - MongoDB Questions, Comments, Suggestions. On Slide 13:

Myles: That concludes our MongoDB presentation. Are there any questions, comments, or suggestions before we move onto the next technical topics?

Start of Technical Topic Presentation 2 - Tailwind CSS.

Move to Slide 14 - Tailwind CSS (Title Slide). On Slide 14:

Samantha: Now, for our second technical topic: TailwindCSS! My name is Samantha Ngo and I'll be presenting with Troy Daniello.

Troy: Hi everyone, today we will explain to you the wonders of Tailwind CSS.

Move to Slide 15 - What is Tailwind CSS? On Slide 15:

Troy: So first let's go over what Tailwind CSS is.

Tailwind CSS is a CSS framework that can be incorporated into many other frameworks such as React. It offers customizable css utility classes which are provided for almost all css styling properties which are also made to be customizable with Javascript. It can be utilized for rapid and adaptable UI production and the framework is written in Post CSS which is a software development tool that uses Javascript plugins.

Move to Slide 16 - What does Tailwind CSS have to offer? On Slide 16:

Troy: So what are some things that Tailwind CSS has to offer?

Well, for one, it can be easily integrated with many other frameworks and is also easy to use which includes the setup process which consists of using a default configuration file or creating one yourself which the details of which will be explained more in depth later.

Tailwind CSS also offers rapid and intuitive user interface creation which can aid in speeding up development. The framework is also designed with consideration for customization from the ground up as colors, styling, and size for different components can be set up in the configuration file. It also implements a sort of single source of truth setup for components and templates, which is great for cases when reusable components are a necessity. Furthermore, another interesting feature is that unused css in an app during building for production is removed by tailwind which makes the app more lightweight and more efficient.

Also, every utility class created can be conditionally applied at different breakpoints which makes building complex responsive and adaptive user interfaces a lot less of a challenge.

Additionally, tailwind also offers the option to create custom directives and functions, for instance a developer could insert different styles into their created CSS.

Move to Slide 17 - Configuration File and Customization Features. On Slide 17:

Troy: Going back to the subject of a configuration file**,** in order to use Tailwind in a project, a developer can either use a default configuration or create a config file to define any customizations. This is meant to be created at the root of a project and is where a developer can define any customizations they want to make.

This consists of many customizable sections some which are but are not limited to: Themes, Breakpoints, Colors and Spacing. Themes themselves have customizable components, those being modification of colors, spacing and screens. Breakpoints can be defined in the config file and deal with formatting to device sizes. The colors section lets a developer have the ability to select an array of different colors that might be wanted in their code. Finally, tailwind's spacing amount can be also modified in the theme section.

Move to Slide 18 - How to Use It . On Slide 18:

Samantha: First, of course, you need to install TailwindCSS, which is really simple because it's an NPM package. Then, you can keep the default configuration or add your own. Then, all you have to do is add the class names listed in the documentation and you're good to go! Tailwind provides many prewritten classes for you and you call it by using their keywords. Here are some examples. You can easily make a nice gradient just by providing four keywords. The first one indicates you want a gradient. The second indicates the starting color, the third indicates the middle color to go through, and the fourth indicates the ending color. Then, Tailwind applies it to your component for you. There's no need to write complicated CSS. This second example creates a ring around a container. Normally, you'd have to write complicated CSS that will create the ring for you, but here, you just use their keyword "ring-offset-" and then you choose one of the widths: 0, 2, 4, whatever you want.

Move to Slide 19 - Custom Configs . On Slide 19:

Samantha: If Tailwind doesn't have what you want in the default config, you can easily add your own customizations. Here are some we added to our project. We added these colors and set the keywords so that we could use them the same way we would call a default color, with "bg" for background and then "-" and the color name. Similarly, we added two new fonts and called it wherever we wanted by just using "font-" and then the font keyword we set. This was our first time using Tailwind and it has made it so much easier to build and design the UI. You don't need extensive CSS knowledge to implement it and it's reusable across your app. The real value in Tailwind is that it makes it easy for you to build your UI by providing pre-written classes rather than writing your own and does so in a way that doesn't restrict your design like Bootstrap for React does. We really hope you'll try it out in your next project!

Move to Slide 20 - Tailwind CSS Questions, Comments, Suggestions. On Slide 20:

Samantha: And that's our Intro to Tailwind CSS presentation. Are there any questions, comments, or suggestions about TailwindCSS?

Move to Slide 20 - Tailwind CSS Questions, Comments, Suggestions. On Slide 20:

Rumman: If there are no questions, thank you so much everyone for listening and have a great summer!

Samantha, Myles, and Troy join in and say thanks in the background.

End of presentation.