Project Features List

FUNCTIONAL - USER SIDE

1. Table Status Information
    a. Tracks the statuses of tables for the management and consumer.
2. Restaurant Information
    a. Stores name, location, size, phone, url, specials, hours, holidays
        i. The application should store and display Restaurant Information for use by managers and employees for quick reference.
            1. Store - on database - edit function
            2. Display - web app
            3. Quick - instant access from other pages
3. Floor Layout
    a. Grid layout with click and draggable tables for management to setup the layout of their restaurant floor and tables.
4. Bill Tracking
    a. Track orders per table and calculate the final bill with tax.
5. Employee Tracking / Manager in charge
    a. Tracks which employees are in and the manager on a timesheet.

NON-FUNCTIONAL - DEV SIDE

1. Database Querying
    a. Ability to query database from UI
2. Onclick/Ondrag logic
    a. Javascript functions to handle a user's clicks correctly based on the component
3. Information Storing
    a. Database contains employee, customer, product, table information
4. Hashed Login Process
    a. Hash passwords with secure algorithm before checking against hashed PWs in a database. Some kind of encryption we haven't decided on yet.
5. AWS
    a. Hosting the website.
6. Grid/Matrix for UI
    a. Layout capable of working with the draggable functions for moving tables

Requirements

Restaurant Information Requirements

As a restaurant employee, I want restaurant information to be easily accessible to provide customers with quick responses.

What: Updateable restaurant information access page

Who: Employees of the restaurant answering queries from customers on site or over the phone, and managers updating information to maintain accuracy.

Why: As a restaurant employee, I want restaurant information to be easily accessible to provide customers with quick and accurate responses. To prevent inaccurate information about hours, holidays, or specials, the information page should be editable by managers.

When: This function is fairly basic, but involves login, backend data, and front end design.

Acceptance Criteria: The restaurant information page can be accepted when it can display and edit information including name, location, phone number, URL, hours, holidays, and specials. Edit ability should be limited to certain users. The page should look professional and consistent with the look of the rest of the application.

Table Status Information

What: Table Status Tracker, which tells the user what the current status of a table is.

Who: The user will use it to check the status of the table, such as the customer, the manager, or the wait staff.

Why: This would tell them if a table is empty, in use, needing service, or needs cleaning.

When: As soon as database is implemented, to see information visually. Low priority.

Acceptance Criteria: User is able to get table information on a table and its status.

Bill Tracking

What: Bill Tracking, which tells the user the current bill amount with tax.

Who: Guests of the restaurant, managers, and employees.

Why: Guests can see what they've ordered and the current bill with tax. This will mainly help resolve bill issues such as if the user ordered something and didn't receive it, they can see if it was accidentally placed on the bill and resolve the issue quickly. Managers and employees will be able to see all bills and items ordered.

When: Very low priority. Can be added after most other things are done, this is just a bonus for convenience's sake. Probably added after table status information.

Acceptance Criteria: Users are able to get all information regarding bill status.

Password Hashing

What: Hash user passwords entered in the login form with SHA-256 before sending the hashed password to the server. Once received by the server, it will check if the password is correct against the stored hashed password.

Who: Any employee who needs to login to the system securely.

Why: Hashing the user entered password before sending it will protect the plain text password from being leaked. If a hacker was able to access the hashed passwords in transit or in the database, they would have to use brute force to find the real password that matches the hash to login with it. This is important to keep the system secure for the client, so no unauthorized users can access/change any data in the system by obtaining a plain text password.

When: Low priority: Can be implemented towards the end of development after the login process is completed. System can be tested without hashing implemented but it will be important for security in the final version.

Acceptance Criteria: Only hashed passwords are sent over the Internet or stored in our database.

Floor Layout

What: This is on the main page of the application which includes interactable react components. The component that will be included in the floor layout will be a table that will have Onclick and Ondrag functions.

Who: Any employee that would need to update information on a table or move table to a different place in the layout.

Why: Being able to interact with a table is crucial to the applications success. The employees need to see a tables status and what their current bill is. If the tables move in the actual restaurant they need to be able to move in the app as well.

When: High priority: Since this is the centerpiece of the main page of our application and will be interacting with our API, this will be constantly adapting as we develop our API and other features. We want to get the Onclick and Ondrag functions working as soon as we can so we can implement more features to the floor layout.

Acceptance Criteria: Onclick/Ondrag working, Onclick actions (book,price) working properly, Ondrag not changing any of the tables properties on moving it permanently.

Information Storage

What: Information about employees, customers, products, and tables will be stored in relations in a database.

Who: Restaurant managers and other employees

Why: Managers will want to store employee information such as phone number, time availability. Managers and employees will want to store information about bills, table status, schedule, and customer reservations.

When: As soon as possible, so we will know how our information is stored and how it will be accessed

Acceptance Criteria: User is able to store and save all information pertinent to employees, customers, tables, and products.

Project Plan

https://trello.com/invite/b/Kok7lf5G/d3273289819adbd479535c48f682cfac/git-that-bread