Requirements

NOTE WE ARE A TEAM OF 6: HRUDITH, RAHUL S, RAHUL G, NICK, JHANGHIR, SUDEEP
OUR TEAM NAME IS: Gang_X1

Your product should meet the following requirements:

- **Develop a web-based application with JavaScript, HTML, and CSS:** The application's primary mode of interaction should be through a browser interface that is written in JavaScript in conjunction with HTML and CSS. It must run, at minimum, in Chrome. It should not require any extensions to run in the browser.
    - **Note #1:** React.js is the supported technology for developing your frontend user interface, but you may use alternative technologies as long as it is in JavaScript.
    - **Note #2:** TypeScript is also allowed since it is inherently JavaScript with Types and is compiled into JavaScript.

- **Host online with PostgreSQL using AWS:** The application must connect to the AWS server and PostgreSQL database you used in Project 2. You can (and probably should) modify your database's schema and seed data.
    - **Note:** Node.js and Express.js are the supported technologies for connecting your frontend to your backend, but you may use alternative technologies such as PHP, Flask, Django, Ruby on Rails, or Spring Boot at your discretion.

- **Serve different types of users or views:** The application should adapt to the different needs of each user type. ***All users are physically in the store, no mobile app is needed.***
    - **Teams of 3+:**
        - **managers:** desktop conventional interface
            - *note: required features are carried over from Project 2*
        - **cashiers:** point-of-sales system interface
            - *note: required features are carried over from Project 2*
        - **customers:** self-service kiosk interface
    - **Teams of 5+:**
        - **menu boards:** non-interactive large multi-display interface
    - **Teams of 6:**
        - **kitchen:** kitchen display interface

- **Make use of external APIs:** The application must utilize the following required external APIs to expand functionality.
    - **Teams of 3+:**
        - **User Authentication:** A web service that performs user authentication. E.g., Google OAuth2
        - Links to an external site.
        - .
        - **Machine Translation:** A web service that performs translation. E.g., Google Translate
        - Links to an external site.
        - .
        - **Weather Service:** A web service that provides weather data. E.g., OpenWeather
        - Links to an external site.
        - .
    - **Teams of 6:**
        - **Team's Choice:** A web service that is distinct from the required ones that is chosen by the team and approved by the instructor.
    - **Note:** no other external APIs or widgets are allowed for Project 3 unless expressly approved by the instructor.

- **Address accessibility by complying with WCAG 2.1**
- **Links to an external site.**
- **and the needs of the given persona in your design:** You can refer to accessibility resources such as **Introduction to Web Accessibility**
- **Links to an external site.**
- .

    - It is industry standard for the entire system to be accessibility compliant.
    - For this project, your ***customer interface*** at the minimum must be compliant for all your required personas.
    - Refer to **Project 3 Accessibility Personas** for further requirements details.

- **Deploy and (remain deployed) by the end of Sprint 1:** The application must stay deployed and be updated regularly at all times thereafter. At any point during Project 3, failure to have your MVP deployed and updated may result in penalizations to your subsequent GitHub releases.

- **Utilize GitHub to version control your repository:** The application code must utilize version control using GitHub in your project workflow.

- **Go beyond the above requirements:** The application must support non-trivial thematic functionality that extends beyond a straightforward restaurant POS system. Designing and developing this thematic functionality will require additional thought, features, and implementation. For each team with *n* members, that team must support (*n* - 3) different thematic functionalities for the ***customer interface***.
    - **Note:** consult your teaching assistant for each of your proposed beyond the above functionality to determine if they are legitimately beyond the above requirements.

Presentation and Demo

Your final product will be presented *at the end of the semester* (see **Finals Location and Times**).

- The final time associated with the lecture will be when you present your working product to the instructor, TAs, and other classmates.
    - The goals of this presentation are to show the evolution of your product from the initial design to the currently deployed service and to convince those funding you to continue.
- The final time you associated with your lab will be a technical demonstration with questions on back-end development to your TA.

    - The demo's goals are to verify that your application meets the assignment's requirements and assess the degree to which those requirements are met or exceeded.

Software Development Life Cycle

You will use an *Agile* SDLC for this project.

- Due to team size and, even more significantly, time constraints, we will not be able to follow a “real” agile approach, but we will try to capture some of the main themes from an Agile development process using Scrum.
- Time management will be crucial to your success. Requirement changes from your initial design will come from the user studies that you provide.

Approach

We will use an iterative approach to this assignment.

- Your team will be asked to implement the program over three “sprints,” each one week in length.
- Features and functionality (i.e., "value") should be added during each Sprint.
- At the end of each Sprint, your team should have working software.
- Between each Sprint is a 1-week reflection and planning period.
- In addition, you are to make backlogs, create burn-down charts, and hold Scrum meetings.

Management

We will use project managers in this assignment.

- Your team will be asked to hold SCRUM meetings with a chosen SCRUM manager at least three (3) times per sprint.
- These meetings are approximately fifteen (15) minutes each for your team to share what is working and not working with each other.