Andrew Shay

1. This capstone will utilize an entirely JavaScript based tech stack with a backend developed through Node using Express.js and the frontend developed as a React.js app. This will fetch tasks for users to do from The Bored API: https://www.boredapi.com/ and store them on a user profile.
2. The focus of this capstone is primarily going to be on the frontend development. This has been an area of coding that I need to improve, and the backend will be relatively simple. The app will look excellent on both desktop and mobile interfaces, and a good amount of time will be spent designing badges, notifications, and tasks for users to do. The backend will only need to store minimal user data, number of tasks they have completed, and the tasks they are hoping to complete.
3. This will be a website, not a mobile app, but it will be designed with mobile use in mind. Basically the website should look great no matter where a user decides to access it, similarly to wordle.
4. The goal of this website is simple: alleviate boredom. Many people found themselves working from home when Covid hit, and the reality is that a good portion of those people never went back to the office. This means fewer social interactions on a daily basis, less mental stimulation, and perhaps even a reduced drive to partake in new activities. Additionally, this pandemic has started a new wave in mental health issues. This means that individuals may find it more difficult to get out of bed, out of the house, or even try new things at home. This website will help make home more exciting!
5. This site will most likely have two kinds of users. The first is the before-mentioned group of people who have found themselves at home most of the time with not much to do and are looking for new activities. The second group is party people! The random task generator would be a great option for a game like truth or dare, or some kind of drinking game.
6. The data collected is quite simple. Tasks from The Bored Api. User information which includes name, email, username, age. Then the last part is user interactions which will include tasks to do and number of completed tasks. There will be an API helper to make requests from The Bored Api and also store information in the Express backend.
7. Project Outline:
    a. Database Schema:
        i. Users:
            1. Id
            2. First/Last name
            3. Username
            4. Email
            5. Age
            6. Password
            7. Number of completed tasks
            8. badges
            9. Avatar
        ii. Tasks todo:

        1. Id
        2. User id
        3. Task description
    b. Potential issues with API could include the fact that it may go down on any given day. This is an issue on any site/app built on an external API, so would need to accommodate a loading screen/site down screen. This API does not appear to have a rate limit or need a key, so shouldn't have any issues with too many requests.
    c. The only information that would need to be secured is a user's password. This will be done using bcrypt.
    d. As far as functionality is concerned this site will have all the basics like signing up, logging in, and logging out. Additionally users will have the ability to look for things to do based on a few categories like price or number of participants. Then they will be able to save them to their profile which will have a list of tasks to do, so obviously functionality will include checking off a task. This will update the both databases including number of tasks completed, potential badges earned, and removal of the task from Tasks table.
    e. The user flow is as follows:
        i. Home page with random task generator
        ii. Page to search for task on criteria
        iii. User page that displays badges/tasks todo
        iv. Leaderboard page which displays users in order of number of completed tasks
    f. This is more than just a CRUD app due to users having accounts which can login and out with web tokens, updating display information based off of user achievements, and a well designed and user friendly frontend with components.