

Style Guide

1. Var Names: Camel Case
2. Max Line Length: 80
3. Spaces per indent : 2
4. Indent Lines: Spaces (Not Tabs)
5. Quotes: Double
6. Only add quotes around object properties where required
7. Trailing commas wherever possible
8. Spaces between brackets in object literals
9. Bracket Line (> in a multi-line HTML file) always alone on next line
10. Always include parenthesis around a sole arrow function parameter: (x) => x

Definition of Done:

- Sprint 1
 - {P1} As a user, I want to have an identity on the application.
 - Task: Create Figma of front end design
 - DOD: Have a viewable design blueprint
 - {P1} As a user, I want to be able to login in from the home page
 - Task: Creating sign in functionality
 - DOD: Implement OA functionality that we can verify with console logs to view information that we can eventually enter into a DB
 - Task: Create Navbar for homepage
 - DOD: Have a viewable Navbar with product name and logo with links ready to be routed to different pages. Incorporate Sign in functionality in the form of a login button on navbar.
 - Task: Create home page with integrated login from firebase
 - DOD: Integrate information obtained from login to a database holding user information
 - {P2} As a user, I want to have a page where I can see events
 - Task: Create a basic events page
 - DOD: Have a page viewable under the “Discover” link where events will be displayed.
- Sprint 2
 - {P1} As a user, I want to be able to create events viewable by others [10 points]
 - Task: Event creation with location, description, organizer, time, etc.
 - DOD: Under the create event link / button, there will be a form popup with all relevant user inputs. The time can be selected as well as a Date (not in the past). The name of the event, description, and capacity can also be inputted.
 - Task: Create boxes / bubbles on the discover page for the front end.
 - DOD: Be able to use data from event creation form to create a box on front end with fields for location, organizer, date, time, description, capacity, and a button to join.
 - {P1} As a user, I want to be able to view and join other events [10 points]
 - Task: Make sure events can be viewed by viewers (and organizers)
 - DOD: Create a function to pull events from the database and send them to the frontend for viewability
 - Task: Functionality to join event, show user name, and how many attendees
 - DOD: Create a join event button that will append the user to a list of attendees (a field in the event class of the database) that only works if the list of attendees is smaller than the capacity for the event
 - Task: Reroute and change pages based on login, connect back and frontend

- DOD: All links work, and take you to the correct pages (pages associated with link name) When logged in the user dashboard should show up as home.

Sprint 3

- {P1} As a user, I want to be able to RSVP for an event [10 points]
 - Task: Add join button to UI
 - DOD: Create a join event button that shows up on the event card in the discover UI.(this was an uncompleted task from sprint 2)
 - Task: Writing and connecting join button functionality
 - DOD: Create a function call for the join button that will append the user to a list of attendees (a field in the event class of the database) that only works if the list of attendees is smaller than the capacity for the event
- {P2} As a user, I want to be able to interact with joined and created events in my user dashboard [10 points]
 - Task: Be able to view joined and created events
 - DOD: When a card is expanded, pull all attendees from the database and make it viewable to the user.
 - Task: Be able to delete an event.
 - DOD: Add a delete button to the users homepage UI that will delete their created event from the database.
 - Task: Be able to leave an event
 - DOD: Create a leave button in the joined portion of the homepage and in the discover page (only for events joined) that will connect to a backend function to remove the user from the attendees field of the event in the database
- {P3} As a user, I want to be able to click on the box to expand and view who else is attending the event [10 points]
 - Task: Created expanded box view with relevant fields
 - DOD: Users can click the expand feature on the bottom of the box that entails all information on the smaller box with the added map and joined members.

○ Sprint 4

- {P1} As a user, I want to be able to view my events on a map [20 points]
 - Task: Integrate a map (possible google maps) to the website
 - DOD: When a location is chosen and a box is created, a map should pop up on the expanded box that shows the google maps location of the location.
- {P2} As a user, I don't want my feed to be flooded with events that have already passed
 - Task: Add functionality to delete events that are expired before they are presented to a user
 - DOD: Adjust backend functionality so that when pulling events for the discover page or homepage, the system will automatically delete events from the database if the current date is past the date of the event
- {P2} As a user, I want to be able to edit event details after I've posted it

- Task: Create capacity field in event database
 - DOD: Add capacity field to event creation pop up and connect that data entry to a corresponding field for the event in the database
- Task: Create button and navigation to editing feature
 - DOD: Click a button to edit the fields that are already in an event.
- Task: Create functionality with firebase to modify documents
 - DOD: We must then connect edit button to a backend function to update the entry in the database with the new data
- {P3} As a user, I want to be able to see the creator of the event and its members
 - Task: Create spots for members / creators / map and edit formatting
 - DOD: When a box is created or expanded, be able to see the organizer and all of the members.
 - Task: Add integration between database and frontend to show members and creator:
 - DOD: Add backend functionality to return creator and attendees to the front end for display