1. Create 1 document per team that describes how at least three features within your finished product will be tested.
    a. **Register**
        i. Check for existing users
            1. Scan DB for users with the same username
        ii. Check for invalid password(Special Characters; Length;
        iii. Provide user feedback of invalid credentials
    b. **Login**
        i. Check for invalid user/password
            1. Show error message
    c. **Insert new event into DB**
        - Use add event
        - Use delete event
        - Use modify event

2. The test plans should include specific test cases (user acceptance test cases) that describe the data and the user activity that will be executed in order to verify proper functionality of the feature.
    a. **Register**
        i. First enter various testing usernames and passwords that are easy to identify.
        ii. Once the test users are entered, we will try to register all of the existing users into the database
            1. This case should return an error message that describes the error
                a. "The username entered already exists"
                    i. This will occur if the user has already been registered
                b. "The password entered does not contain special characters"
                    i. This will occur if the password is trying to be registered does not fit the format required
                c. "The username format entered is invalid"
                    i. The username is invalid due to a large length
    b. **Login**
        i. First try a known existing username and password
            1. This should successfully show the home page for the athlete
                a. Should redirect with no errors and should render the correct user page

        ii. Try an unregistered username and password
            1. This should popup the error message
                a. "Invalid username. If the username has not registered, please register."
            2. This page should suggest the register button which will render the register page when clicked

c. **Event testing**
    i. Default events should be added to the DB
        1. Events should appear on schedule organized by date then time
        2. No error should occur upon logging into this page directly
    ii. Add new event
        1. Add event button is clicked on and a popup prompts the user to add event details
            a. If the title is blank
                i. "Please enter a valid title"
            b. If the time field is left blank
                i. "Please enter a valid time"
            c. If the description is blank
                i. This is not required but recommended
                ii. The text below should popup
                iii. "Warning: description is blank. Continue if not description is desired"
    iii. Delete event
        1. This should never fail
            a. Have a catch all statement that prints to console
                i. "Delete failed"
    iv. Modify event
        1. Try to modify an event that has the same title and date
            a. "Please have a unique title and date for this item"
        2. Try to submit a blank field for all the fields
            a. If the title is blank
                i. "Please enter a valid title"
            b. If the time field is left blank
                i. "Please enter a valid time"
            c. If the description is blank
                i. This is not required but recommended
                ii. The text below should popup

            iii. “Warning: description is blank. Continue if not description is desired”

3. The test plans should include a description of the test data that will be used to test the feature.
    a. **Register**
        i. Test Username: oigoi_ss {VARCHAR(30)}
        ii. Test Password: abcdefgh123! {VARCHAR(80)}→Length>8; Numbers: 0-9; Symbols !-&
    b. **Login**
        i. Test user: oigoi_ss
        ii. Test password: abcdefgh123
        iii. Test User: oigoi_ss randomTest
        iv. Test Password: 123456789
    c. **Insert New Event into DB**
        i. Adding event
            1. Event Name: Group Workout
            2. Event Time: 5:00 PM
            3. Event Name: Coach training
            4. Event Time: 5:15 PM
        ii. Deleting Event
            1. Deleting Event that does exist
                a. Delete Event: Group Workout
            2. Deleting Event that doesn’t exist
                a. Delete Event: Yoga
                    i. Error message
        iii. Modify Event
            1. Modify Event: Group Workout
                a. Change Name: Workout
                b. Change Time: 6:00 PM

4. The test plans should include a description of the test environment that will be used to test the feature.
    - Postman for Backend testing
    - Browser for front end testing
    - Postgres for Database testing

5. The test plans should include a description of the test results that will be used to test the feature.

- Add event
    - Success: Event is added into calendar database and page
        - “Login was successful”
    - Failure: Event is NOT inserted into calendar database and/or page
        - “Login failed”

- Register
    - Success: User is added to the database and allowed to sign in
        - Data payload : { email: “your_email” , “password”: }
    - Failure: User was NOT added to the user database
        - “Login failed”
- Login:
    - Success: Allows access to user info, view programs they are part of, view events
        - Data payload : { email: “your_email” , “password”: }
    - Failure: User is NOT able to view or access programs that they joined
        - “Login failed”

6. The test plan should include information about the user acceptance testers. Please note that the deployed application can only be tested on the CU Boulder campus. So make sure to select your testers accordingly.
    a. Run on local systems at CSEL for general user testing
        i. Target People in CSEL and Family/Friends/Co-Workers