

# Event Champ

Jam 23

Milestone 1 3/30/25

# Project Overview

Our app Event Champ is an event hosting and finding app that is meant to be used by communities, clubs, rec groups or any organization to post about their upcoming events. And then users can go on scroll through the list of events in their area and look for events to go to or sign up for.

The primary goals of our app is to allow users to create events, browse events happening in their area, and to be able register for events.

# Team Members

Sami - Worked on setting up Github/Framework and the 'My Events' page where users could create events, edit created events, and view the events they have created and registered for.

Aryan - Worked on linking the create/edit components to the displayed events page.

Bao - Worked on the navbar, and buttons for the app, created a button that when clicked popups a login window, and can be switched to a create an account window.

Marco - Worked on account page included creating sign up component and associated container components. Continuing progress on login component and functionality.

# Software UI Architecture Overview

Our project uses as React for our base with React Router for routing users through the website's pages.

Additionally we use these technologies:

- Motion - Component animations
- TailwindCSS - Class styling
- Lucide - Component icons

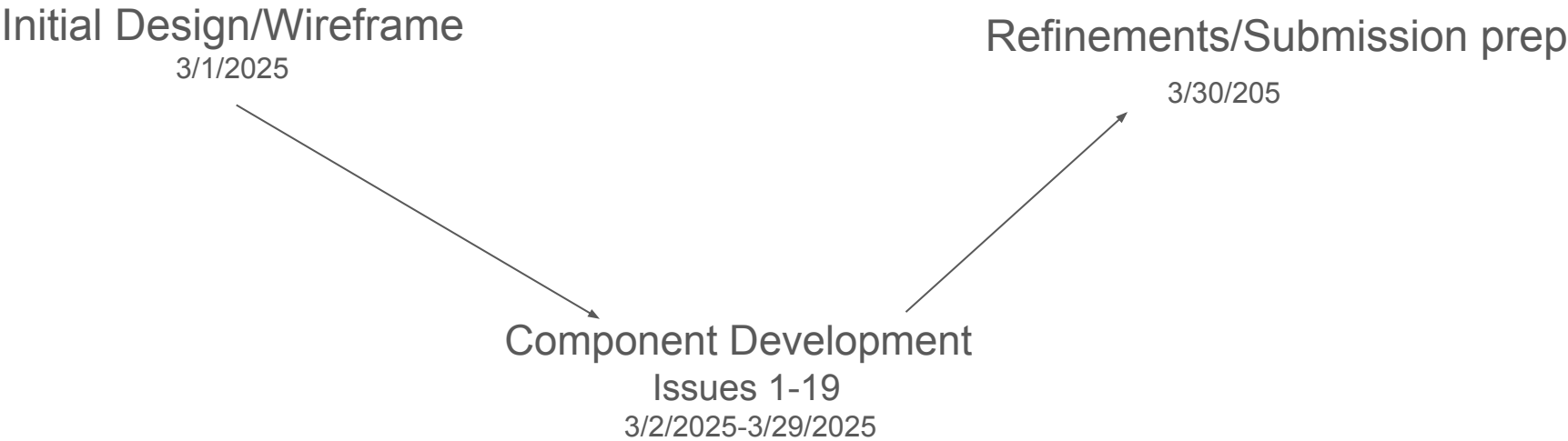


**Tailwind CSS**



**React Router**

# Historical Development Timeline



# Design and Styling Guidelines

- Our UI will sleek and modern with temporary colors, and text fonts
- Our UI will also implement a sticky navbar so users can easily navigate through the website whenever they want
- Buttons and components will have mini animations
- Our events will be in a grid layout for intuitive navigation and readability.
- We will have a counter above a bell icon to notify users of any upcoming events
- Users will be able to see which events they already subscribe, events they already subscribed to will be marked in green
- Styling doc link:  
[https://docs.google.com/document/d/15W-Sd1X\\_mrVnAwOCijoz8t9kTj0QeX2hplC4HzqrDC4/edit?usp=sharing](https://docs.google.com/document/d/15W-Sd1X_mrVnAwOCijoz8t9kTj0QeX2hplC4HzqrDC4/edit?usp=sharing)

# Performance Considerations

- Our React State Management has been optimized by lifting state only when necessary.
- Our project will use memoization to prevent unnecessary re-renders. This is done with `useMemo` and `useCallback`.
- Our project will also use React lazy loading to improve our project's initial load times.

Initial tests of these optimizations have proven successful and we will begin to implement it in our next sprints.



# Sami - Assigned Work Summary

✓	🕒	Registered events list	Component Development	#7 - samiOahmed opened last month · 📌 Milestone 1 – Fr...
✓	🕒	List of created events	Component Development	#6 - samiOahmed opened last month · 📌 Milestone 1 – Fr...
✓	🕒	Manage Event button	Component Development	#5 - samiOahmed opened last month · 📌 Milestone 1 – Fr...
✓	🕒	Create Events Page	Component Development	#4 - samiOahmed opened last month · 📌 Milestone 1 – Fr...
✓	🕒	Create Event Button	Component Development	#3 - samiOahmed opened last month · 📌 Milestone 1 – Fr...

## Updated createEventButton, editEventButton, eventPage, with eventContext #22

🔗 Merged samiOahmed merged 1 commit into `main` from `SamiBranch` 1 hour ago

💬 Conversation 0

↔ Commits 1

🔍 Checks 0

📄 Files changed 3

+193 -300



samiOahmed commented 1 hour ago

Owner

Reviewers

No reviews

Assignees

No description provided.

Link to PR: <https://github.com/samiOahmed/426Project/pull/22>

I was assigned tasks to create our events page, create event button, edit event button.

Specifically during this milestone I set up our app framework. I also then went ahead and created the event context which is to wrap our entire app and to be used by any event based component. I also then went ahead and made the createEventButton and editEventButton. I also made the 'My Events' page which houses those two buttons and the list of events that users have registered for or created.

Setup styles folder	83ba1d5	📄	📁	<>
Set up context folder and EventContext.tsx	ccc8ce7	📄	📁	<>
Set up components and pages folders	d1e93a1	📄	📁	<>
Set up Vite framework	ff8974c	📄	📁	<>

### Merge pull request #22 from samiOahmed/SamiBranch

samiOahmed authored 1 hour ago

Verified

e3a9a07

📄

📁

<>

### Updated createEventButton, editEventButton, eventPage, with eventContext

samiOahmed committed 1 hour ago

7c1d3c0

📄

📁

<>

### Cleaned up src

samiOahmed committed 11 hours ago

e37fcf7

📄

📁

<>

### Merge branch 'main' into Azuresnow-patch-1

samiOahmed authored 11 hours ago

Verified

7e215a9

📄

📁

<>

Commit history

# Sami - Code & UI Explanation

```
1 import {useContext, useState} from 'react'
2 import {EventContext} from '../context/eventContext'
3
4 export const CreateEventButton = () => {
5   const {createEvent} = useContext(EventContext)!
6   const [name, setName] = useState('')
7   const [host, setHost] = useState('')
8   const [date, setDate] = useState('')
9   const [location, setLocation] = useState('')
10  const [capacity, setCapacity] = useState=number | undefined>(undefined)
11  const [error, setError] = useState('')
12
13  const handleCreateEvent = () => {
14    if (!name || !host || !date || !location) {
15      setError('Fill all fields')
16    }
17    return
18  }
19  const newEvent = {
20    name,
21    host,
22    date,
23    location,
24    isReserved: false,
25    capacity,
26  }
27  createEvent(newEvent)
28  resetForm()
29
30  const resetForm = () => {
31    setName('')
32    setHost('')
33    setDate('')
34    setLocation('')
35    setCapacity(undefined)
36    setError('')
37  }
38 }
```

This is a piece of the code I worked on for creating our createEventButton. It integrates into our UI as seen below. It creates a bunch of fields for the user to enter in the relevant info such as event name, host name, date, location, and capacity. The main challenge with this came with how we could use the event created by this button across other parts of our app. To solve this we created a context for event, eventContext, which allows for our events to be used across all components and pages of the app.

test event	sami	04 / 18 / 2025 , -- : -- -- 📅	Burlington	1	⬇️ Create Event
------------	------	-------------------------------	------------	---	-----------------



# Sami - Challenges and Insights

Some obstacles I faced during this milestone was making sure that my components would incorporate events globally. This means that they would have to be able to work for my page as well as the other pages in our app. As a result I learned a lot about the importance of context. My key takeaway from working in a collaborative environment is how important it is to know ahead of time what parts of your work is reliant on just you and what is reliant on others in the group.

# Sami - Future Improvements and Next Steps

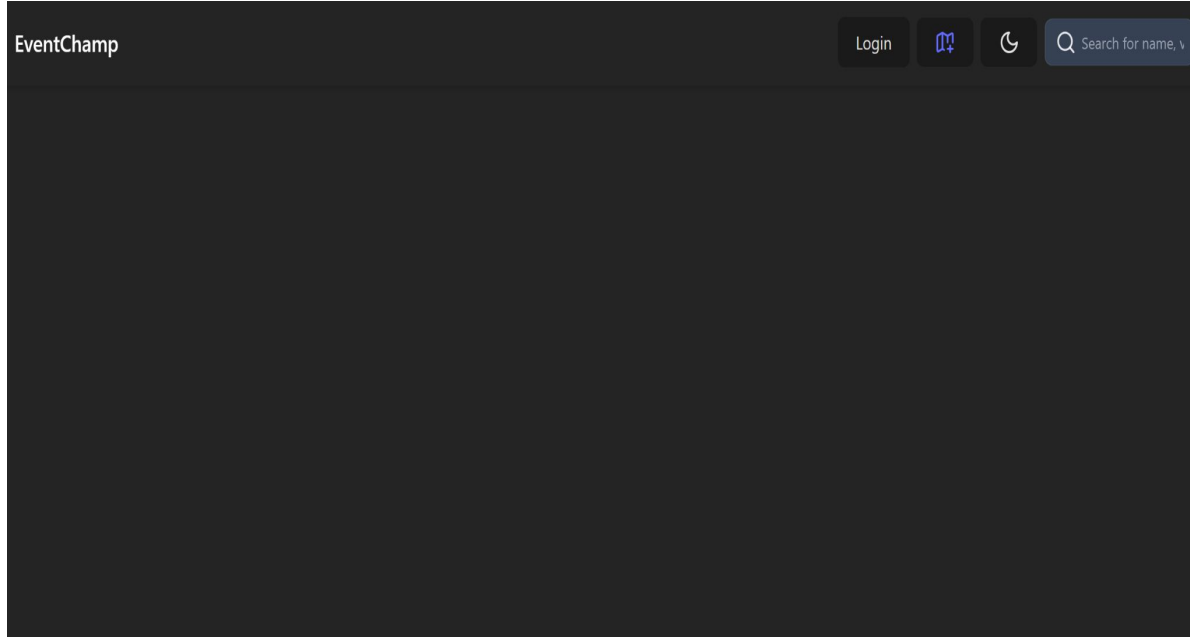
I plan on working on making the event components functional so that our website starts coming to life. This will be crucial step in making our team's project closer to a functioning website than it is now. I will also perform any component modifications as needed in case there are changes when conducting the next development cycle.

# Bao slide 1

- I created the basis for a UI layout with a login button when clicked a login popup will appear prompting users to either login or sign up
- A create events button, that links to the create an event page via router
- A dark mode button which will be functional soonTM



# Bao slide 2



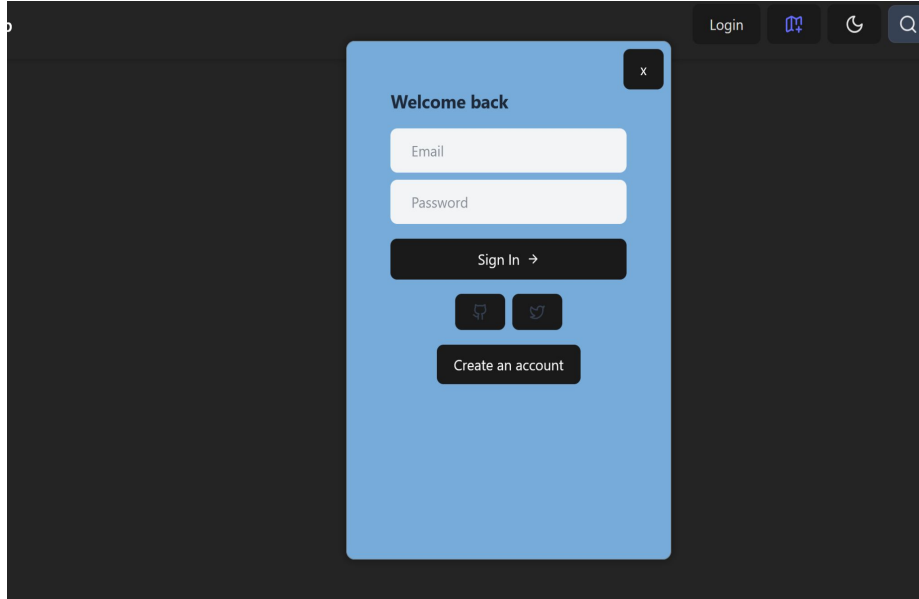
-This is the main page

**Login:** popups a login UI

**Map icon:** brings you to the create an event page

**Moon icon:** Dark and light mode (not working)

## Bao slide 2 continued





A screenshot of a login popup form on a dark background. The popup is light blue with a title bar that says "Welcome back" and a close button (X). The form contains two input fields for "Email" and "Password", a "Sign In →" button, two social media icons (Google and Twitter), and a "Create an account" button. The background shows a navigation bar with "Login", a menu icon, a refresh icon, and a search icon.

Welcome back

Email

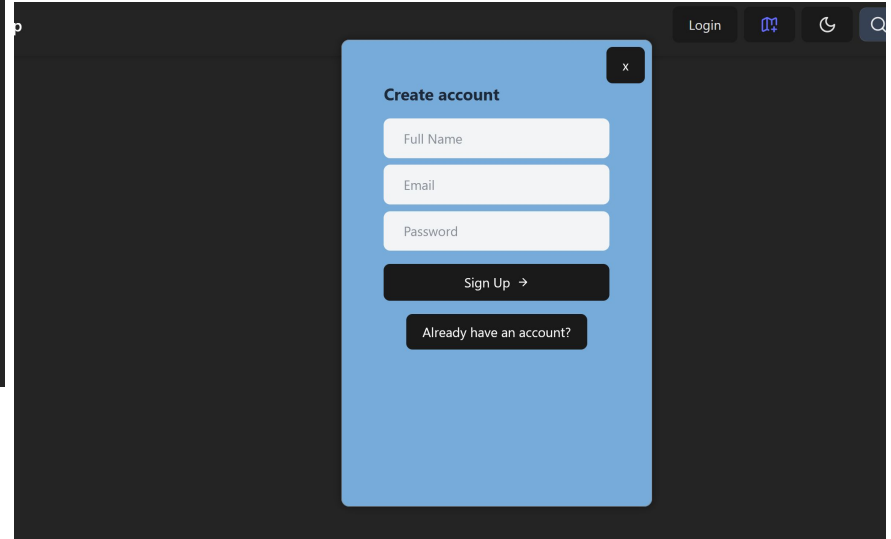
Password

Sign In →

Create an account

This is the login popup page, users can switch to signup when clicking create an account.



A screenshot of a "Create account" popup form on a dark background. The popup is light blue with a title bar that says "Create account" and a close button (X). The form contains three input fields for "Full Name", "Email", and "Password", a "Sign Up →" button, and a link that says "Already have an account?". The background shows a navigation bar with "Login", a menu icon, a refresh icon, and a search icon.

Create account

Full Name

Email

Password

Sign Up →

Already have an account?

# Bao Challenge/Future improvements

- It was a bit difficult to align the UI
- Users will be able to quickly navigate our website because all of the important buttons are easily accessible through the navbar
- Users won't be redirected to a login or signup page since they're part of the same page.
- Future plans: For the homepage to have a grid layouts of posted events
- Add a bell icon for notifications

# Aryan slide 1

I linked the create/edit event component to a display events page. Now when a new event is created, it will be displayed on a separate page. I created a new file in the src/pages folder. I styled this page so each event is displayed as a rectangular container arranged horizontally across the screen.

# Aryan slide 2: code explanation

The code is very straight forward right now. I linked the components used for the create event page and attached them to the display events page. Each event has a couple attributes attached to them. Such as name, data, location, how many people can go, and image. I attached these values to the display event container.

426Project / EventChamp / src / pages / DisplayEvents.tsx

aryan-pothanaboyina Added display events page and connected it to the create events page,...

```
Code Blame 44 lines (40 loc) · 1.59 KB Code 55% faster with GitHub Copilot

1 import { useContext } from 'react';
2 import { EventContext } from '../context/eventContext';
3
4 export default function DisplayEventsPage() {
5   const { allEvents, registeredEvents, registerForEvent, unregisterFromEvent } = useContext(EventContext)!;
6
7   return (
8     <div className="display-events-page">
9       <h1>Upcoming Events</h1>
10      <div className="events-row">
11        {allEvents.length === 0 ? (
12          <p>No events available</p>
13        ) : (
14          allEvents.map(event => (
15            <div key={event.id} className="event-container">
16              /* Event Image */
17              <img src={event.image || '/default-event.jpg'} alt={event.name} className="event-image" />
18
19              /* Event Details */
20              <div className="event-details">
21                <h3>{event.name || "Unnamed Event"}</h3>
22                <p><strong>Host:</strong> {event.host}</p>
23                <p><strong>Location:</strong> {event.location}</p>
24                <p><strong>Time:</strong> {event.date}</p>
25                <p><strong>Spots:</strong> {event.attendees}</p>
26              </div>
27
28              /* Register/Unregister Button */
29              {registeredEvents.some(e => e.id === event.id) ? (
30                <button onClick={() => unregisterFromEvent(event.id)} className="unregister-btn">
31                  Unregister
32                </button>
33              ) : (
```



## Aryan slide 3: challenge and insights

- I am new to React so I was having a challenging time connecting my Display Events component to the team's existing code base. I plan on studying up more on React to help alleviate my trouble in the future.
- It was difficult thinking about the way I should have created the Display Events page before it existed as I did not have an exact vision for the page before creation.



## Aryan slide 4: future improvements

Clicking on an event will display a pop up that reveals more information about each event. There will need to be an additional information component on the create events page as well.

# Marco - Assigned Work Summary

During this milestone, I focused on beginning the creation of the account functionality for our project. I began this task by functionally designing and then coding the signup component for the account functionality of our application. This also included creating the base “container” components that house the signup component and will house the future login and forgot password components. All of these created components will interface with the work that Bao did in his UI component visual design.

## Create signup.tsx #23

Merged

marco-dm1 merged 1 commit into `main` from `accountbranchdev`  now



Conversation 0



Commits 1



Checks 0



Files changed 1

## Accounts - Create “Sign Up” Component #9

Closed



marco-dm1 opened 27 days ago

Create sign up component that will be on the my account page. It will allow users to input send to the server so that the account can be created.

Create sub-issue



marco-dm1 added `Component Develo...` `enhancement` 27 days ago



marco-dm1 added this to the `Milestone 1 – Front-End UI` milestone 27 days ago



marco-dm1 self-assigned this 27 days ago



marco-dm1 now

Resolved in pull request [#23](#)



marco-dm1 closed this as `completed` now

# Marco - Code Explanation

As pictured here, I worked on the Signup component for our project. This included letting the user input a username, email, and password (w/ confirmation). This was done by utilizing input fields in an HTML form. The input field data is captured when the user clicks on the submit button.

```
export function Signup(){
  return(
    <div id="accountSignup">
      <a className="text-lg">Sign Up</a>
      <form onSubmit={onSignup} className="flex flex-col items-center space-y-3 pt-5">
        <input
          id="usernameInput"
          type="text"
          placeholder="Enter your username"
          value={username}
          onChange={(e) => setUsername(e.target.value)}
          className={`w-64 p-2 border bg-white placeholder-black text-black rounded shadow-sm`}
          required
        />
        <input
          id="emailInput"
          type="email"
          placeholder="Enter your email"
          value={email}
          onChange={(e) => setEmail(e.target.value)}
          className={`w-64 p-2 border bg-white placeholder-black text-black rounded shadow-sm`}
          required
        />
        <input
          id="passwordInput"
          type="password"
          placeholder="Enter your password"
          value={password}
          onChange={(e) => setPassword(e.target.value)}
          className={`w-64 p-2 border bg-white placeholder-black text-black rounded shadow-sm`}
          required
        />
        <input
          id="passwordConfirmInput"
          type="password"
          placeholder="Confirm your password"
        />
      </form>
    </div>
  )
}
```

# Marco - Challenges & Insights

The main challenge I faced for this milestone was the team coordination with our project's account functionality development. I had originally planned to create all 4 main components for the project's account system during this milestone however I needed to restart midway through coding my components as our team decided to have another team member handle the UI development for the account system. This decision was intended to standardize the UI design throughout the entire application so it was important however it caused a delay in the development I had been conducting.

# Marco - Future Improvements & Next Steps

- The account functionality of our website is intended to be one of the next things we will finish therefore I will be steady at work on the components.
- During the next few weeks, I plan on completing the work that had been delayed due to the previous decision made by the team. Additionally, it was a learning experience for our team and this delay will help us stay on track in the future.
- Lastly, I plan on getting the account system components to be truly functional in the next milestone.