
Eventful

Application Development for
Events Sharing App

Yukie Matsumoto

Contents

- Project Overview
 - Our Team
 - Agile Meetings
 - Project Management Tools
 - Design
 - Database/ API Set Up
 - GitHub
 - Development
 - Java Script Best Practice
 - Copyright
-

Project Overview

SHARING APP

The project is to develop a web application for event organisers(user 1) to post events and communicate with attendees(user 2).

- All users need to log-in to use this app
 - Event organisers can add, edit and delete the events
 - Event organisers can communicate with attendees thorough the comment function
 - Attendees can use the comment function to ask questions to organisers and communicate with other attendees
 - Filter by event category, location and date
 - Search by category, event name and keywords
-

Our Team

Our Initial four team members were:

- Ryan Bakker
- Sunny Huang
- Shiori Ikigame
- Yukie Matsumoto

On week two, one of our team members had a different idea for design and functions, therefore we decided to work on the project separately. Our team became three members.

- Sunny Huang
 - Shiori Ikigame
 - Yukie Matsumoto
-

Agile Meetings

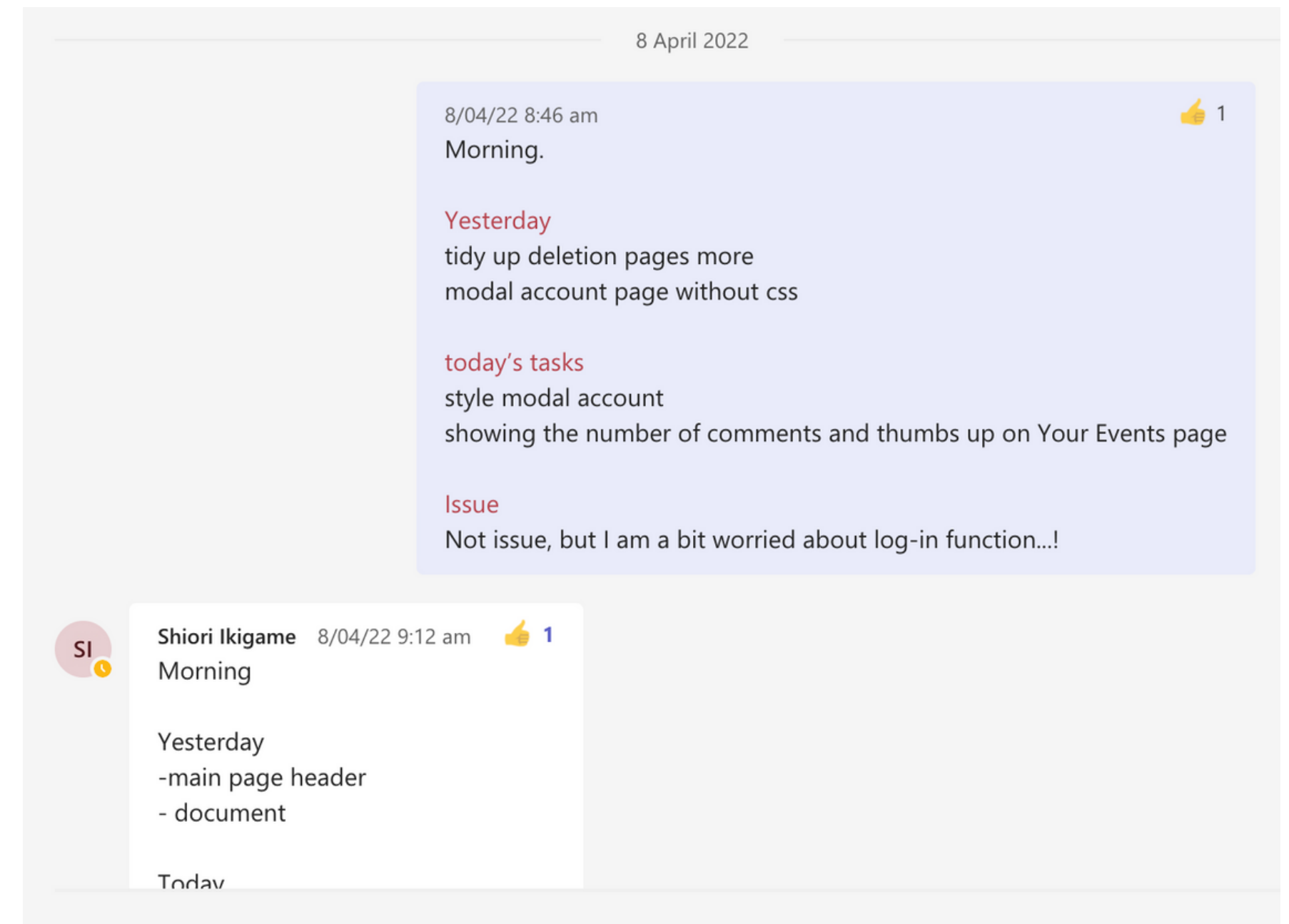
We had a daily scrum meeting every morning to share the information below:

- Yesterday's tasks
- Today's tasks
- Issues

Meeting Tool

- Microsoft Teams

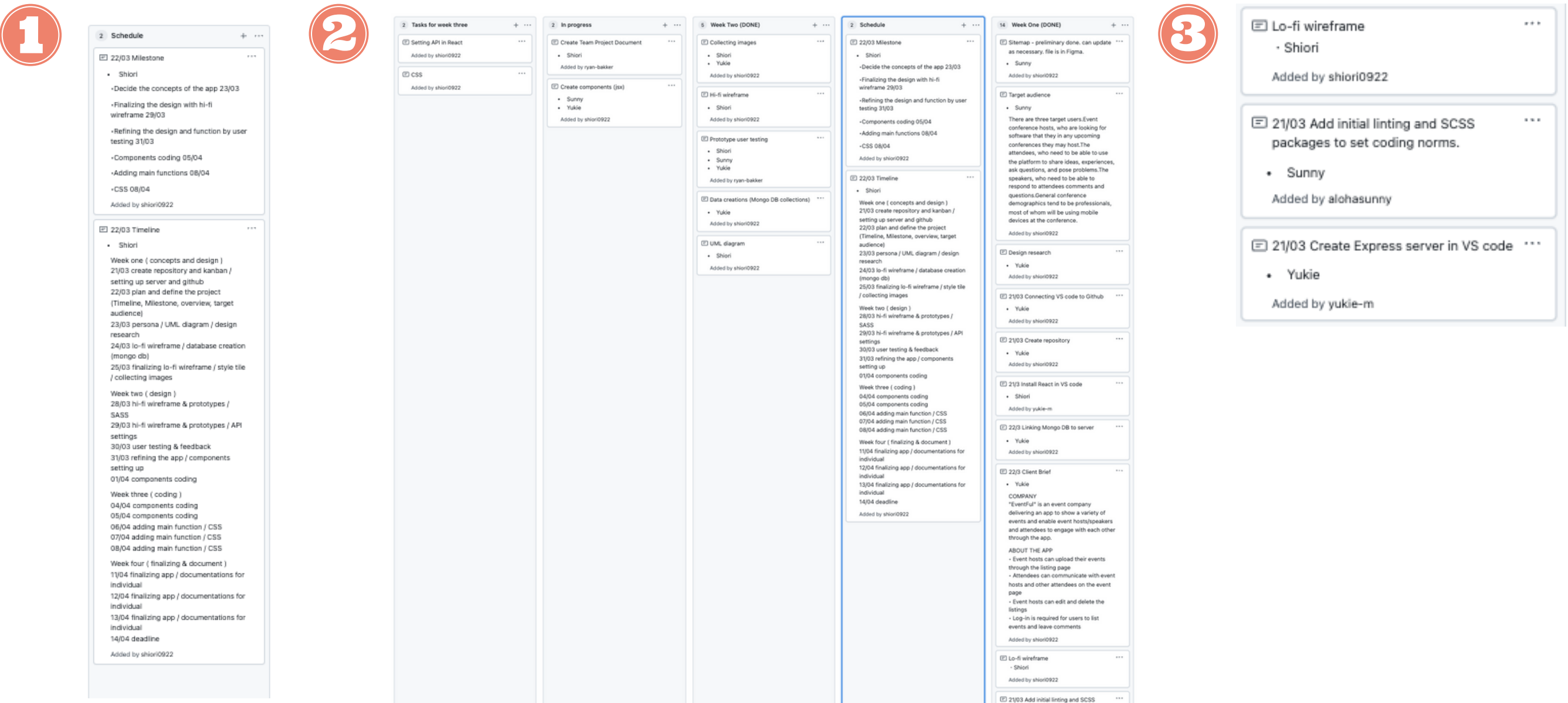
Each team member committed to the meetings and used leadership skills to ensure the project went smoothly.



Project Management Tools

GitHub Kanban was used for the project management.
[Click Here](#) to check our project Kanban.

- 1: We created a milestone and a To Do List
- 2: We divided the To Do List into three weeks
- 3: We assigned tasks to each team member



Sprint One

TASK

Week one (concepts and design)

21/03 create repository and kanban /
setting up server and github

22/03 plan and define the project
(Timeline, Milestone, overview, target
audience)

23/03 persona / UML diagram / design
research

24/03 lo-fi wireframe / database creation
(mongo db)

25/03 finalizing lo-fi wireframe / style tile
/ collecting images

DONE

- Sitemap (Sunny)
 - Target Audience (Sunny)
 - Design Research (Yukie)
 - Connecting VS code to Github (Yukie)
 - Create Repository (Yukie)
 - Install React in VS code (Shiori)
 - Linking Mongo DB to server (Yukie)
 - Client Brief (Yukie)
 - Lo-Fi Wireframe (Shiori)
 - Add initial linting and SCSS packages (Sunny)
 - Create Express server in VS code (Yukie)
 - Creating logo (Yukie)
 - Persona (Shiori)
 - Style Tile (Shiori)
-

Sprint One

Overview

Design

Shiori leaded our design.
I also assisted on some tasks.

Documentation

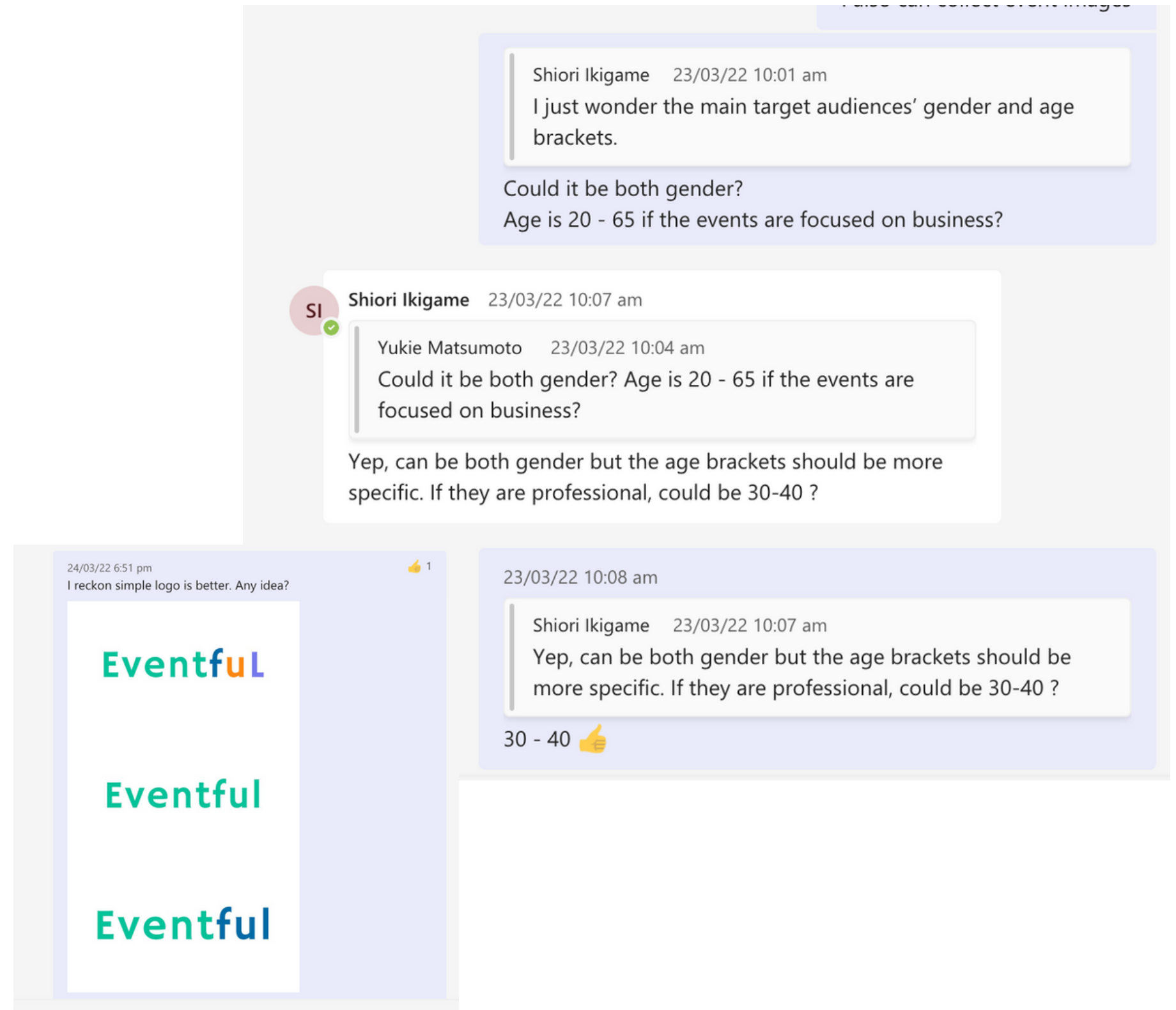
We assigned tasks to each person.

GitHub/MongoDB

I set up GitHub and MongoDB.

React

Shiori downloaded react app.



Sprint Two

TASK

Week two (design)

28/03 hi-fi wireframe & prototypes /
SASS

29/03 hi-fi wireframe & prototypes / API
settings

30/03 user testing & feedback

31/03 refining the app / components
setting up

01/04 components coding

DONE

- Collecting Images (Shiori & Yukie)
- Hi-Fi Wireframe (Shiori)
- Prototype User Testing (ALL)
- Data Creations (Yukie)
- UML Diagram (Shiori)

Sprint Two

Overview

- Unfortunately, Ryan did not show up to daily scrum meetings as he had a different interpretation of the project scenario
- We discussed Hi-Fi wireframe design before user testing
- 4 user testings were done

From Melanie

-Vey easy to log in and I liked the idea of having the social media as an alternative log in option
-easy to locate things with the search button and the categories
-I didnt know how to add a new event, dont have a button on the front page and I had to search - I found on the profile page.
Maybe it would be nice if we have a easy button to do so if that is the objective of the app


it was easy to know where to comment (but the prototype dont allow me to)

I understand that some functionalities wont work because is a prototype thats why I stated there

I like how clean the app is and how easy is to find things and functions, just the create a new post was hard to find, as I was looking for something on the main page 😊

From Luke

-Log in page looks really good and super easy to navigate
-Create event page looks good too
-Spelling mistake in my account pop up, should be username.
This list is too long so need to scroll down on my phone to see the bottom items
-Your events list looks good too, but when I delete an event, the deleted confirmation pops up but doesn't go away by itself
-Main page and list of events is good. Can click on event title to go to event page, would be good to click on the event image too
-Possibly useful to filter events by location?
-Maybe clicking on eventful logo could take you to an about page?
Or have an about button
-Overall looks really professional and easy to navigate / understand

SI Shiori Ikigame 29/03/22 5:57 pm  1
I have updated the hi-fi according to kevin's demo from today.

 Hi-fi wireframe after change.pdf ...

SI Shiori Ikigame 29/03/22 6:09 pm
<https://xd.adobe.com/view/cb2d0629-d3c5-4031-85ca-c0aa9fb203d6-9e7e/>

Can you check and give me feedback if you have as we will need to do user testing tomorrow.

SH Sunny Huang 29/03/22 8:29 pm
⊗ This is getting really close. Here are some of the things I think we should change:

- Post --> Create (this is more appropriate for events)
- Your Posts --> Your Events
- Instead of likes, perhaps you can mark your intention to attend? Similarly, perhaps the home page shows how many people plan to attend?

Sprint Three

TASK

Week three (coding)

04/04 components coding

05/04 components coding

06/04 adding main function / CSS

07/04 adding main function / CSS

08/04 adding main function / CSS

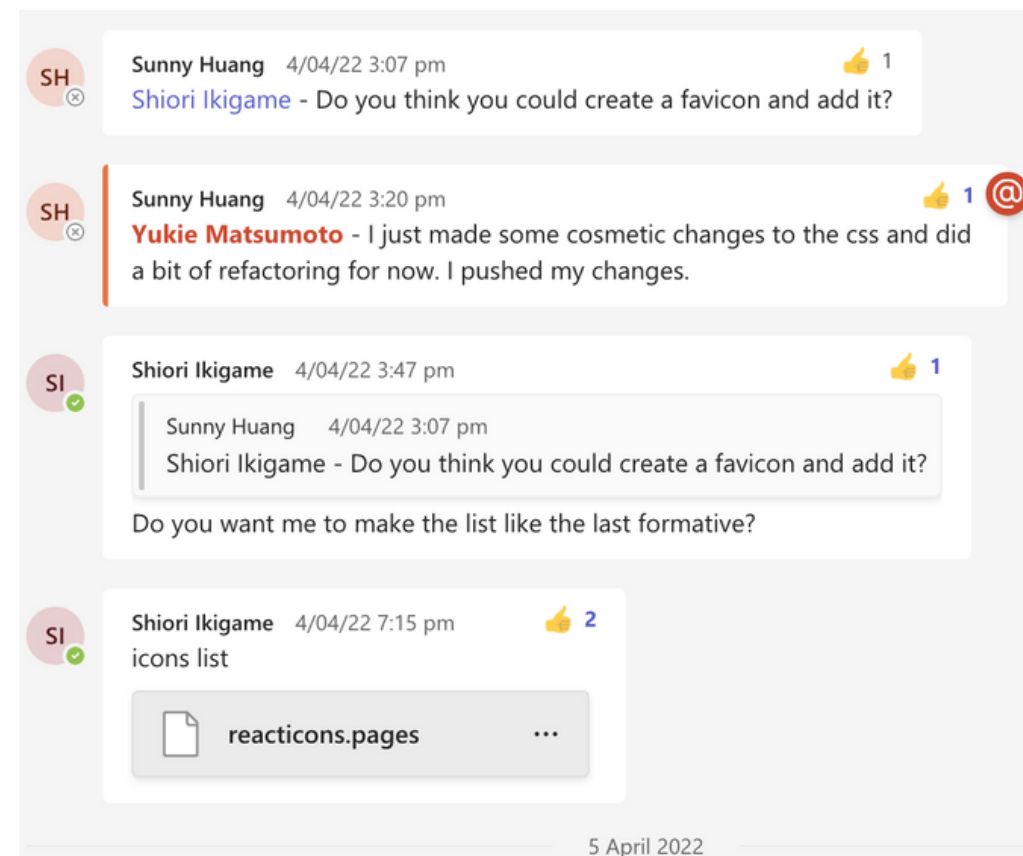
In Progress

- Components (Sunny & Yukie)
 - Basic CSS (All)
 - Advanced CSS (Shiori)
-

Sprint Three

Overview

- Basic coding was completed on week 3
- All functions worked well
- Event List
- Event Detail
- Create Event
- Delete Event



4/04/22 11:14 am Edited
I created a component and also styled as well.
Need to collect icons for the left top arrow button.
Shiori Ikigame Do you have any ideas for icons? Or is icons fine?

A screenshot of the 'Create Event' form. The form has fields for Title, Date (dd/mm/yyyy), Category (Business), Location (Auckland), Link URL, Upload image (Business), and Event details. A 'Create' button is at the bottom.

Create Event

Title

Date

dd/mm/yyyy

Category

Business

Location

Auckland

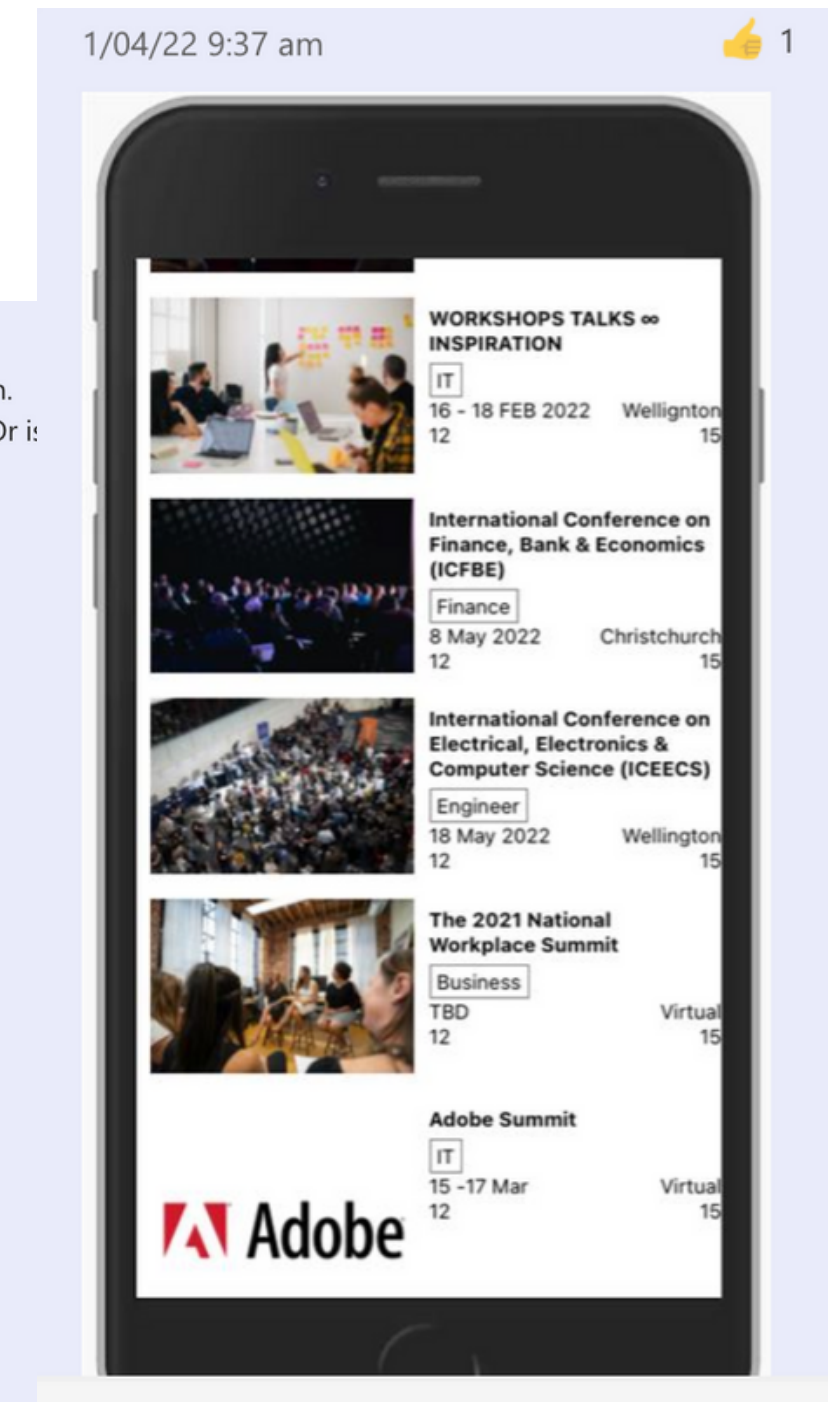
Link URL

Upload image

Business

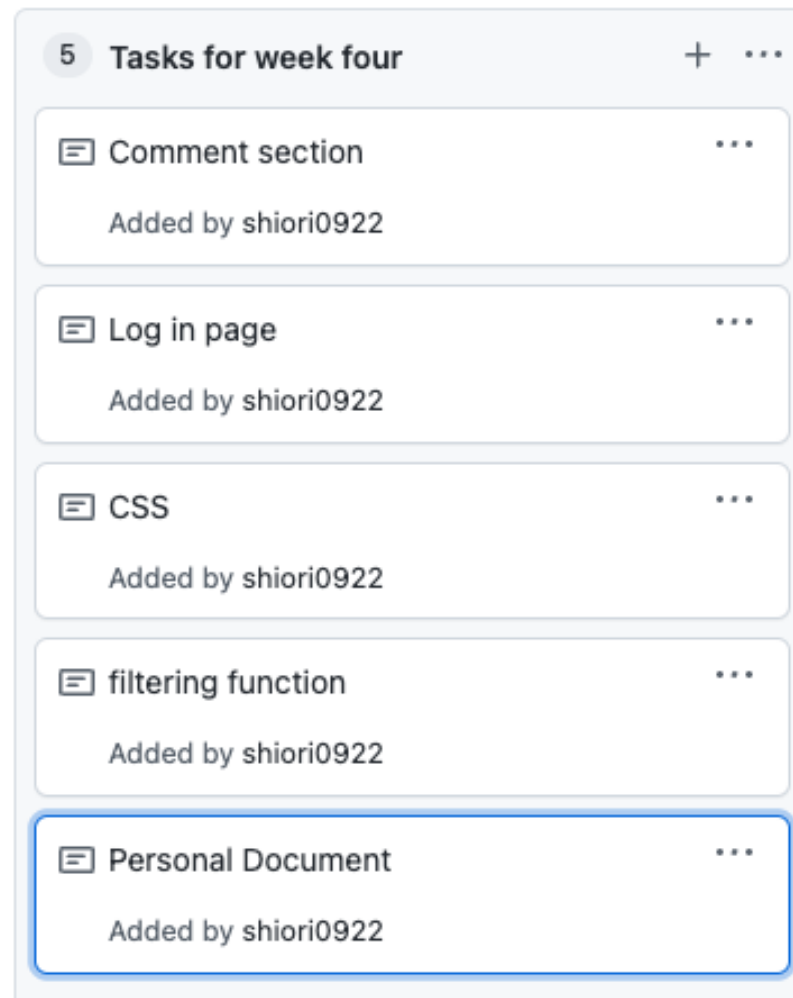
Event details

Create



Sprint Four

TASK



DONE

- Filter Function (Shiori & Yukie)
- Basic CSS (All)
- Advanced CSS (Shiori)
- Comment Function (Sunny)
- Personal Document (All)

Sprint Four

Overview

We added more functions on week four:

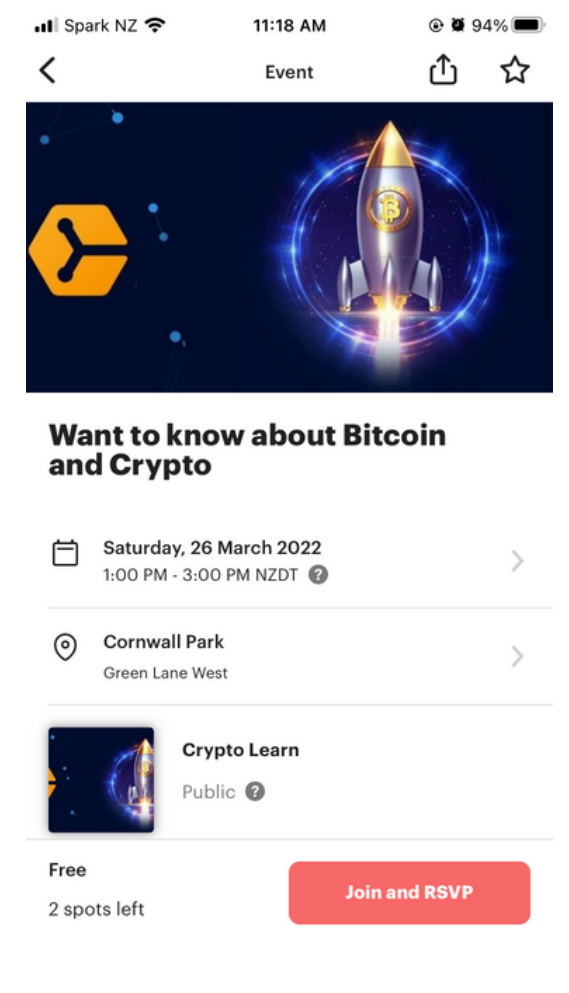
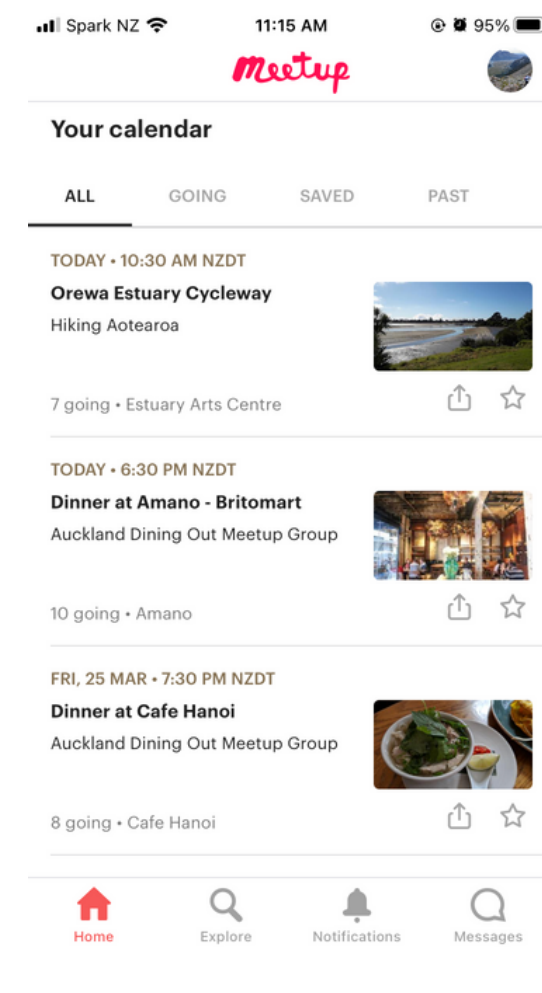
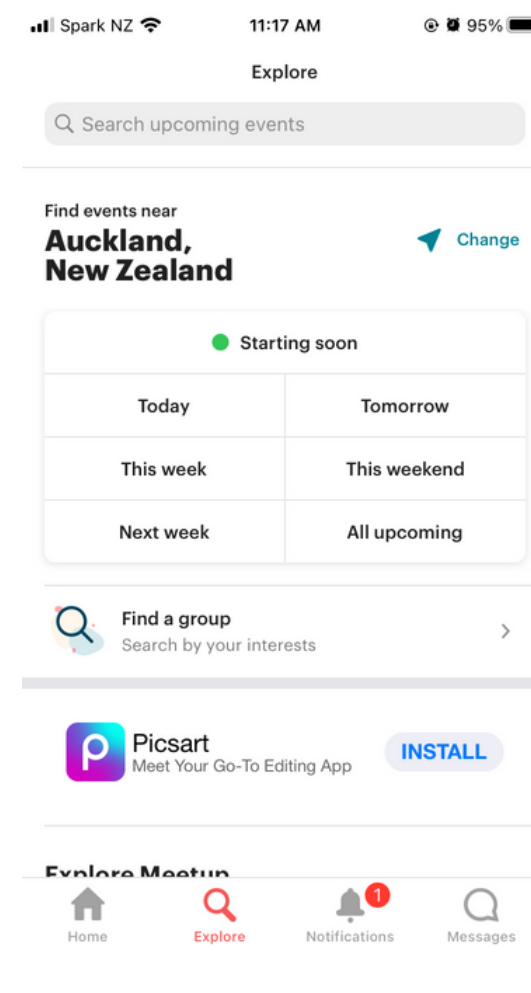
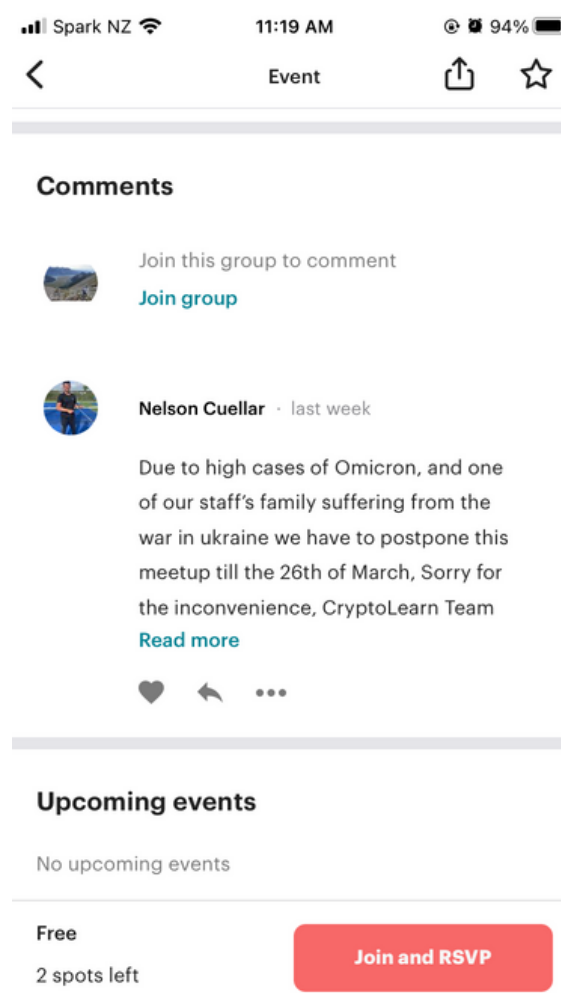
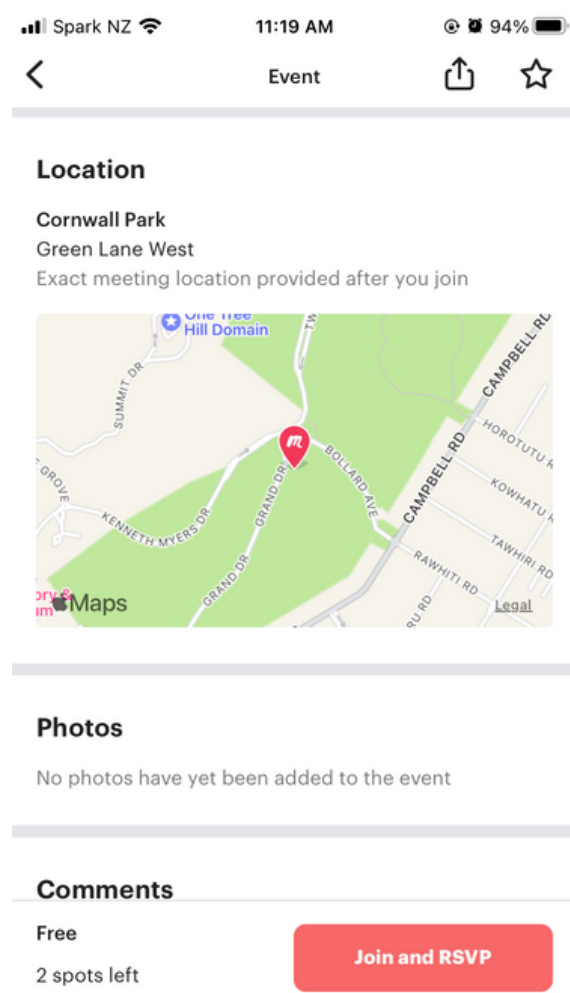
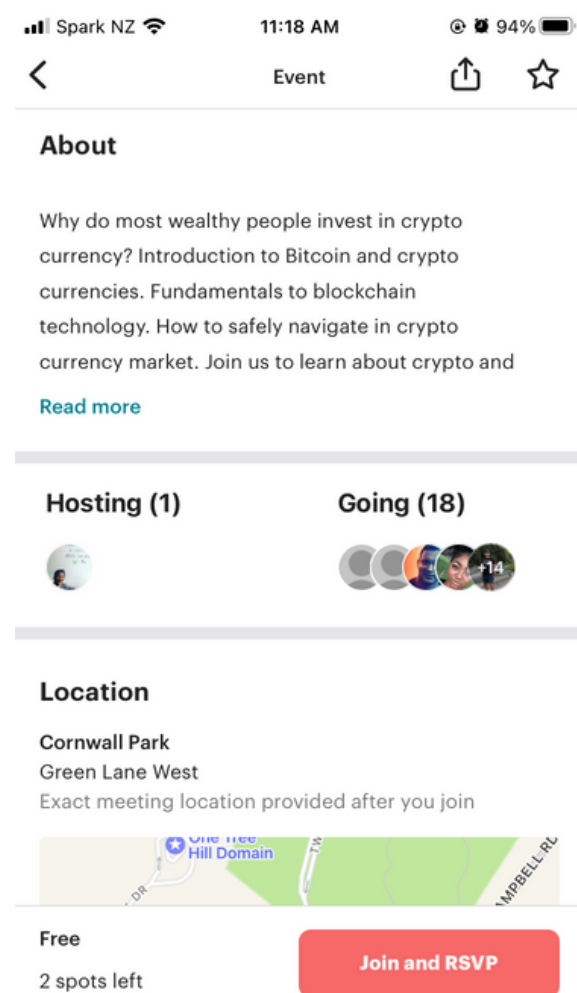
- Category filter
- Edit page
- Comment Function

Sunny was making a log-in & sign-up function which was most complicated function and completed on the last day. The css was changed from the initial design that we created so that we decided not to merge her branch to main as we ran out of time to fix it. Functionally, it works perfectly, so we are willing to continue working on it after this group project is submitted.

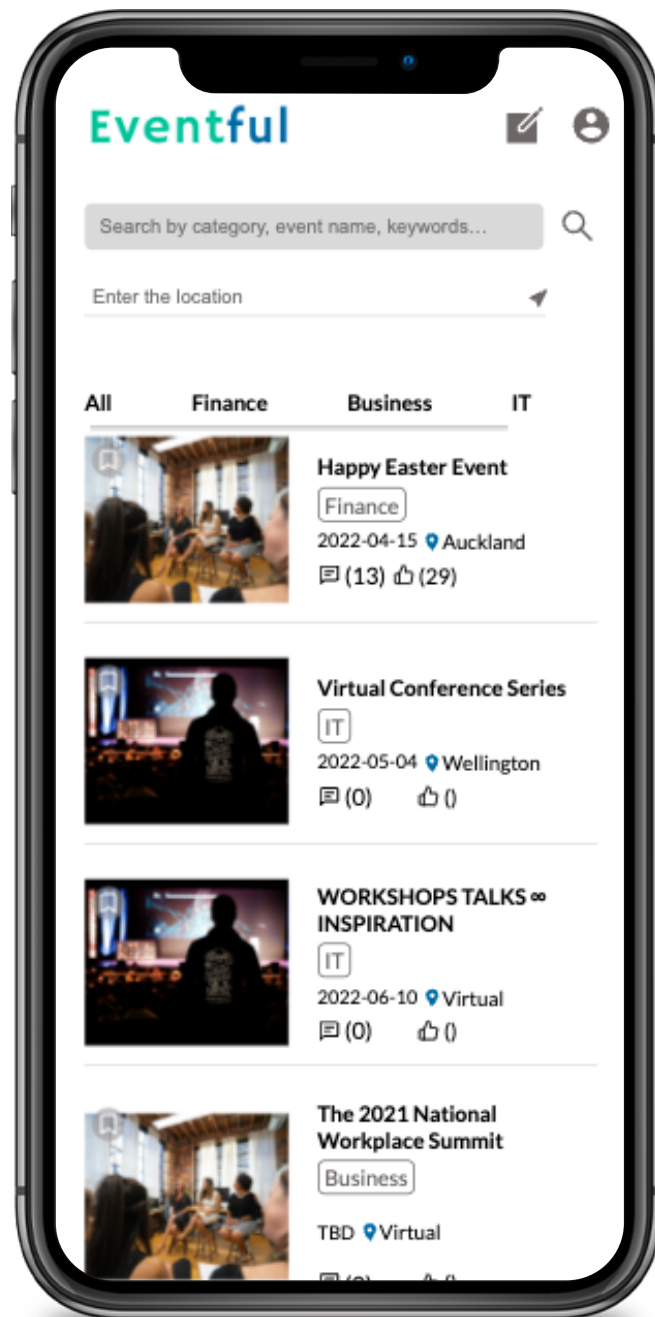
Design Research

Design Ideas from meetup App

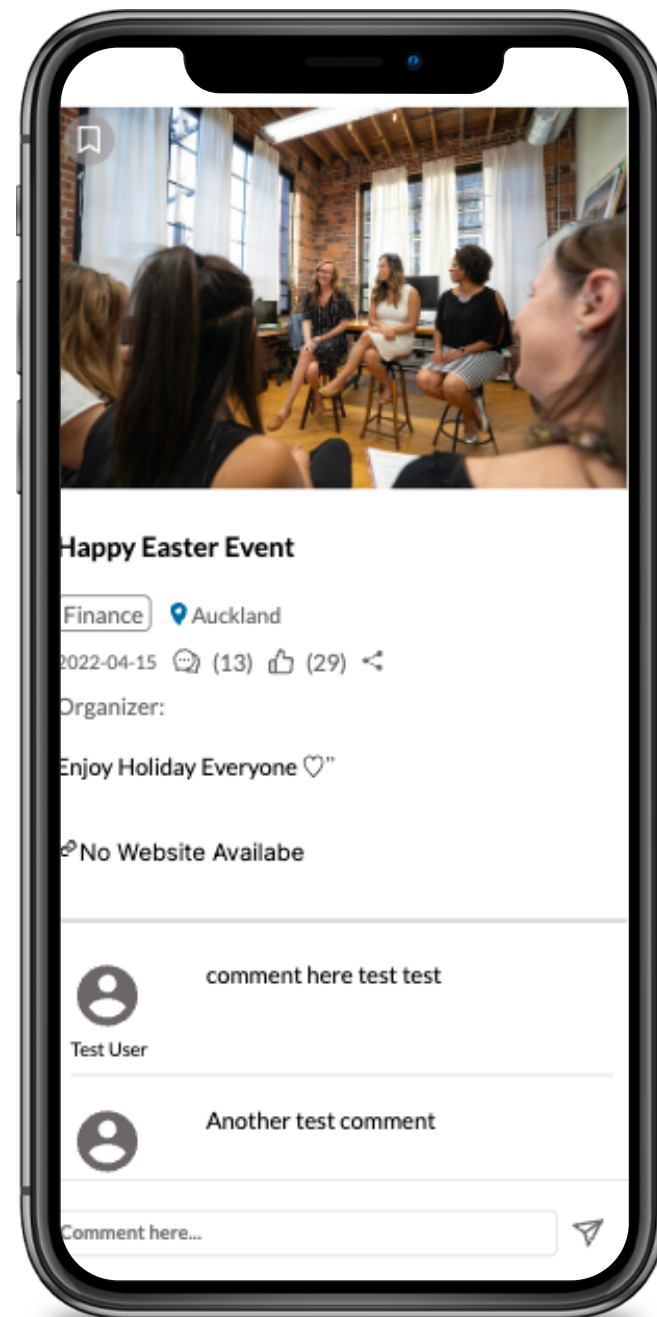
- Simple design is critical
- Use limited colours
- Entire page design should be consistent
- Concise information and text
- Page navigation should be comparable with industry standards



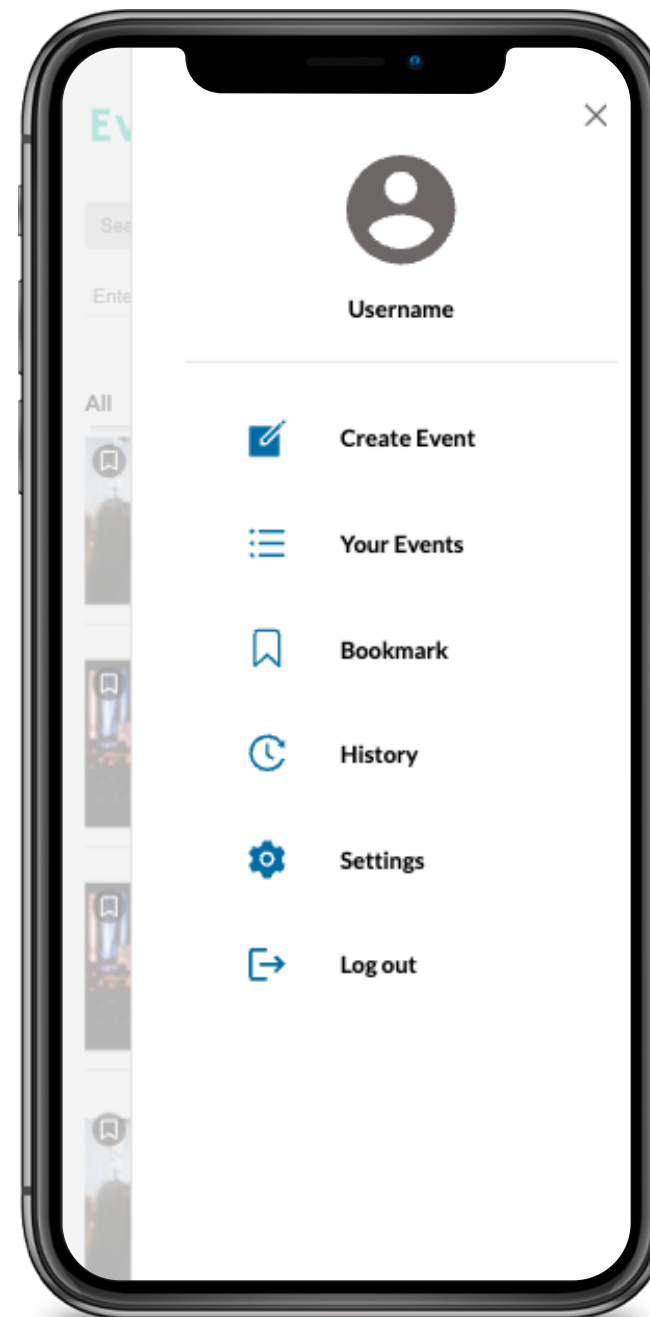
Design



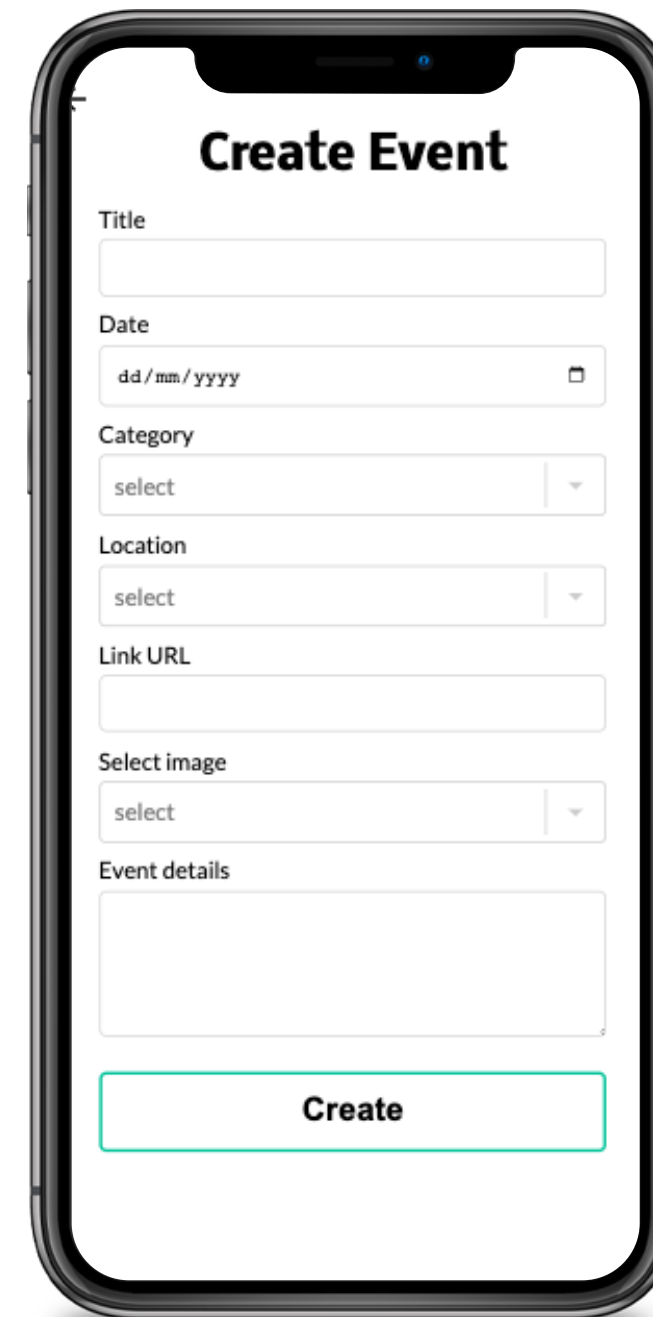
- Event List
- Filter by Category
- Create Event
- Account Menu



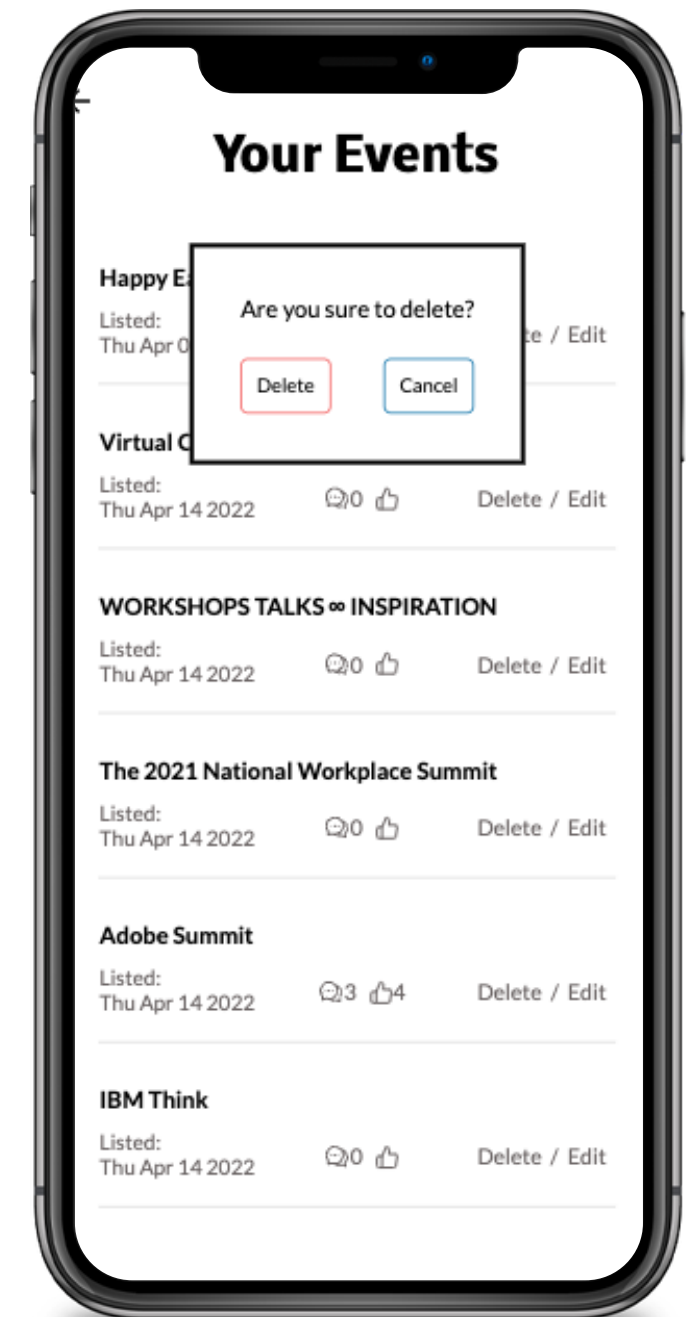
- Event Details
- Comments
- Like Function



- Modal Account
- Create Event
- Your Events >> delete/edit



- Create Event
- Design is consistent



- Your Events
- Ideally, it should be connected to each logged in account, but we ran out of time to do so.

MERN stack

We use combined stack of MongoDB, Express, React and Node.js to create **CRUD** web application.

MongoDB: A document oriented database for creating a strong web application.

Express: Express is a framework that helps in creating REST APIs.

React: A Javascript library that creates interactive UI.

Node.js: We used it for non-blocking and event-driven servers.

CRUD Web Application

APIs routes were created with Express.js, MongoDB and Node.js

CRUD	REST API	URL
Create	POST	/create-event
Read	GET	/view-events /view-event-by-id/:id /view-event-by-category/:category
Update	PUT	/update-event/:id
Delete	DELETE	/delete-event-by-id/:id

MongoDB & Define Mongoose Schema

We created a mongoDB schema for interacting with mongoDB database.

summativeThree

comments

events

users

summativeThree.events

STORAGE SIZE: 36KB TOTAL DOCUMENTS: 6 INDEXES TOTAL SIZE: 36KB

Find Indexes Schema Anti-Patterns Aggregation Search

FILTER { field: 'value' }

QUERY RESULTS 1-6 OF 6

_id: ObjectId("624d8242ac00f8aeb2aaa813")

comments: Array

title: "Happy Easter Event"

date: "2022-04-15"

category: "Finance"

location: "Auckland"

link: ""

thumb: "women.jpg"

description: "Enjoy Holiday Everyone 🥰"

createdAt: 2022-04-06T12:06:26.032+00:00

updatedAt: 2022-04-14T07:31:20.279+00:00

likes: 30

_id: ObjectId("6257bb56bcb14e5726f41a7a")

comments: Array

title: "Virtual Conference Series"

date: "2022-05-04"

category: "IT"

location: "Wellington"

link: "https://www.dwexperience.com/"

thumb: "it.jpg"

description: "Industry experts and leading practitioners in IT, HR, People Analytics..."

createdAt: 2022-04-14T06:12:38.598+00:00

updatedAt: 2022-04-14T07:30:42.942+00:00

__v: 0

likes: 1

EXPRESS-SUM...

models

JS comments.js

JS events.js

> node_modules

.gitignore

JS connection.js

{ } package-lock.json

{ } package.json

JS server.js

models > JS events.js > ...

1 const mongoose = require("mongoose");

2

3 var Schema = mongoose.Schema;

4

5 var EventsSchema = new Schema({

6 {

7 title: String,

8 description: String,

9 host: String,

10 price: Number,

11 date: String,

12 time: String,

13 thumb: String,

14 location: String,

15 category: String,

16 attending: Number,

17 link: String,

18 follower: Number,

19 following: Number,

GitHub

GitHub was used for collaborating on our team project.

Repositories (Public)

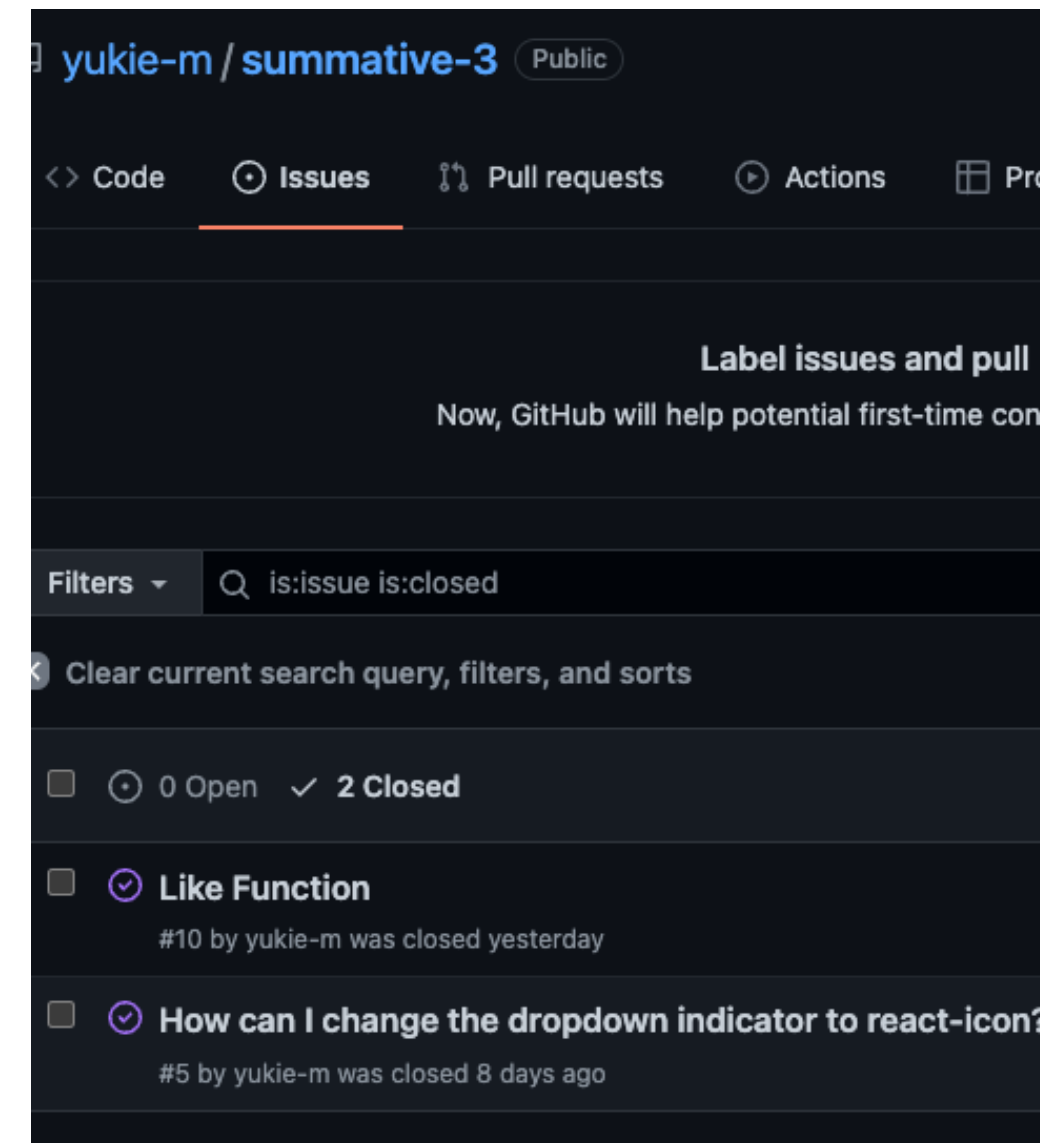
- yukie-m/summative-3 for a react app
- yukie-m/express-summative-3 for an express server

Issues

Technical issues were posted on GitHub Issues to get the code to work.

Branches

We usually worked on each person's own branch, then checked if there were any bugs before we merged.



Main Project Dependencies

A variety of project dependencies are managed by a package manager. Complex and extensive tasks are automated through a task runner without error.

```
> nodemon ./server.js
[nodemon] 2.0.15
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node ./server.js`
Phew!, listening on port 4000
Success! Connected to MongoDB
```

Create a react project	npx create-react-app <i>my-summative</i>			Node -v	Version of node
icons	npm install react-icons -save			npm-v	Version of Node Package Manager
axios	npm install axios --save				
React Router	npm install react-router-dom@6			ls	List files and Folders
sass	npm install sass --save			pwd	Present Working Directory
	npm init -y			cd	Change directory
Create Express Server	npm install express nodemon cors mongoose --save	Edit the package.json "scripts": { "test": "echo \"Error: no test specified\" && exit 1" }, Becomes "scripts": { "start": "nodemon ./server.js" },			
Modal	npm install react-modal --save				
Js-cookie	npm install js-cookie --save				
react-slide	npm i react-slide-routes				
eslint	npm install eslint-plugin-react --save-dev				

Framework & Libraries

React JS

React is a JavaScript library built and maintained by Facebook.

The key benefits to use React JS for us were below;

- Fast speed
- Easy to use
- Reusable components
- Ability to build rich UI

React-slide

Used for a page slide animation.

React-icon

All icons are chosen from react-icon.

React-select

Great UI of select tags.

React-Modal

Nice and clean UI and easy to use.

Framework & Libraries

Axios

Axios library was used to send asynchronous HTTP requests to REST endpoints.

We used Axios to perform CRUD operations.

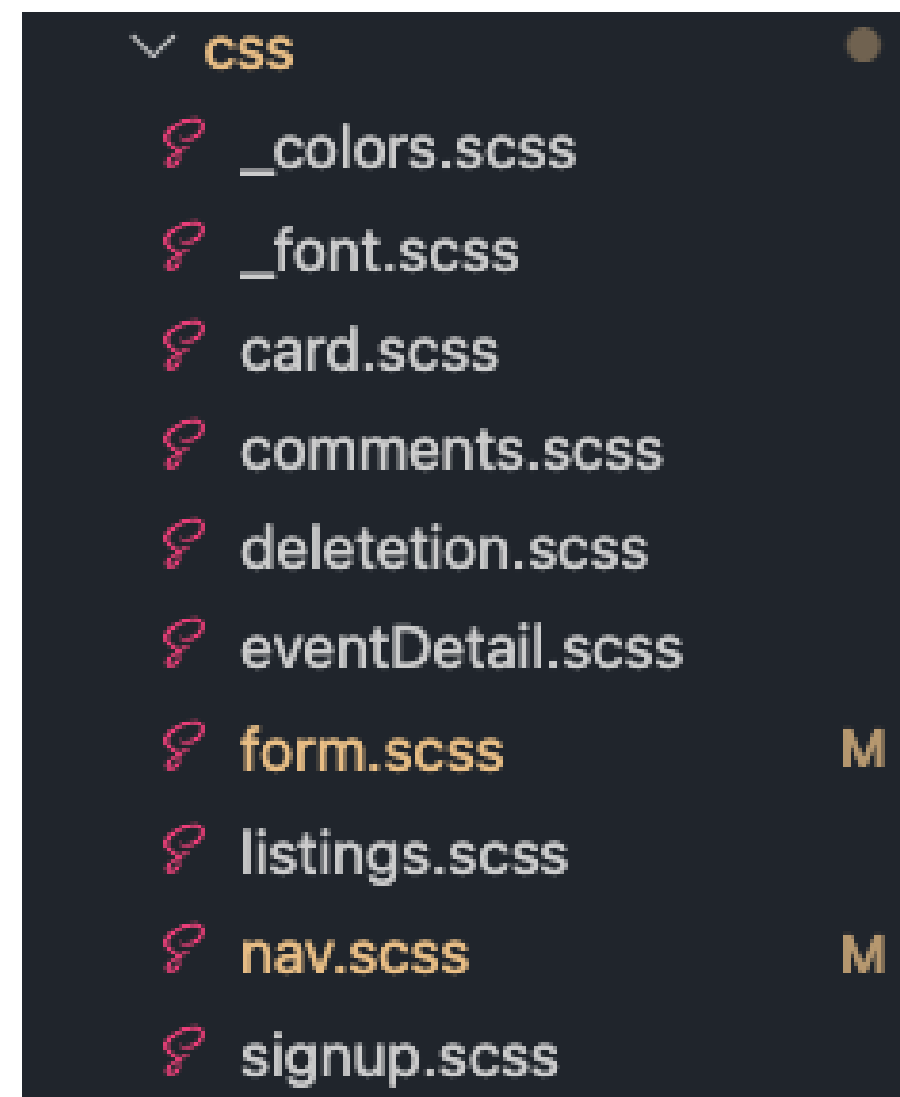
```
useEffect(() => {
  axios.get("http://localhost:4000/api/view-events").then((response) => {
    if (response.status === 200 && response.data.length > 0) {
      setListings(response.data);
    }
  });
}, []);
```

```
router.get("/view-events", function (req, res) {
  Events.find().then((response) => {
    res.json(response);
  });
});
```

Framework & Libraries

SASS

Sass enabled the app to have consistent design such as font-family/size and colours.



```
@use "colors";  
@use "._font";
```

```
$navy: #006ba6;  
$light-grey: #d9d9d9;  
$dark-grey: #6c6666;  
$green: #02c39a;  
$black: #000000;  
$red: rgb(243, 80, 80);
```


Java Script Best Practice

- Used consistent naming
- Named functions and variables logically
- Always declared variables with const and let

 EventSingle.jsx

```
const loadEventDetails = (event) => {
```

```
function EventSingle({ detailsObj }) {  
  const navigate = useNavigate();
```

Used Ternary for if statement

```
<div className="event-date">  
  {detailsObj.date ? detailsObj.date : <p>TBD</p>}
```

Java Script Best Practice

- Comment our code
- Always use semicolons

```
// Modal
.modal-account {
  width: 90%;
  display: flex;
  flex-direction: column;
  justify-content: center;
```

Used Javascript linting tools

- ESLint
- Prettier

```
"eslintConfig": {
  "extends": [
```

✓ Prettier

Copyright

Images: [Unsplash](#)

Icons: [React Icons](#)

Document + Logo Design: [Canva](#)
