

Group 4

Milestone Report 2

SportsHub

Amateur Sport Events Platform

17.12.2021

Ali Alperen Durak
Ece Dilara Aslan
Fatih Akgöz
Mehmet Hilmi Dünder
Muhammed İrfan Bozkurt
Tolga Kerimoğlu
Yağmur Selek
Yiğit Sarıoğlu

1. Executive Summary

What has been done?

For this milestone, we had the map functionality in the center of our focus. The goal was to provide an easy-to-use map interface for searching and creating events. The backend also focused on formatting data according to the W3C activity stream standards.

The main thing we want to improve upon in the software development process is better versioning of our deployed apps. We have been working on facilities to quickly rollback to a previous version, currently just using docker versioning.

Going forward

For the final milestone, we would like to get the equipment market up and running, implement badges, comments, notifications and follows.

Pre-release tag:

<https://github.com/bounswe/2021SpringGroup4/releases/tag/v0.3>

2. Summary of Work

TEAM MEMBER	CONTRIBUTION
Tolga Kerimoğlu	<ol style="list-style-type: none">1. Got together with the backend team and walked them through a roadmap of what to accomplish before this demo and our task delegation.2. Implemented an activity handler module that regulates application-wide data transfer and transforms it to abide the W3C activity stream standards. The streams are also implemented in a way that it makes implementing any 'activity history' functionality very easy:3. Implemented comment functionality for events with relevant endpoint documentation.4. Implemented badges. Decided upon 8 badges by browsing Wikidata. Documented the relevant endpoints and usage.5. Wrote the executive summary and the API documentation for the Milestone Report.6. Reviewed pull requests.7. Created a thorough POSTMAN collection in addition to the documentation we already have so

	<p>that frontend/android teams can work smoother.</p> <p>8. Did a pair-programming session before the demo night with the frontend team to get ready for the presentation.</p>
İrfan Bozkurt	
Mehmet Hilmi Dünder	<ol style="list-style-type: none"> 1. Participated in weekly group meetings to plan the project. 2. Studied about w3c activity streams and examined the other implemented APIs in the backend to develop suitable code for other parts. 3. Investigated requirements about equipment API in the project requirements and understood how this API will be implemented 4. Implemented on equipment API to make proper w3c activity streams and created pull requests. 5. Implemented unit tests for the new version of the equipment API 6. Investigated previous pull requests about equipment API and some of them closed without merge. 7. Participated in the demo session.

FRONTEND

TEAM MEMBER	CONTRIBUTION
Yiğit Sarioğlu	<ol style="list-style-type: none"> 1. Leads the frontend team 2. Implemented the “user profile page”, “edit profile page”, “Google map functionality”, “Image/File Upload functionality”. 3. Implemented File Upload/Image Upload by Axios in React. Files/Images are stored in cloud service.I created an account on cloudinary, uploaded files here. Then I took the

	<p>links and sent them to the backend service.</p> <ol style="list-style-type: none"> Opened google developer console account and got google API key from here. Worked on setting google map in “create event page”. Implemented the user authorization part on the frontend side, integrated with backend. I edited the objects displayed on the homepage to be different according to guest and user. Opened a page on the wiki (github) for the React JS- Documented the React and frontend side Used the documentation prepared by the backend team to access to the API endpoints Used POSTMAN to test the API endpoints Did a pair-programming session before the demo night
Yağmur Selek	<ol style="list-style-type: none"> Implemented the autofill functionality using google map places api and add this functionality to event creation page location selection and checked the events api endpoint of backend to send the correct inputs. Integrated the google maps with the autofill functionality. Contributed to the integration of google map with the event creation page. Updated the necessary packages to install on React JS documentation on wiki. To test API events endpoint; I used postman. Participated the pair-programming session before the demo night
Ali Alperen Durak	<ol style="list-style-type: none"> Created the main branch called frontendmain for frontend codes and constructed the development pipeline to follow the better practice from prior milestone. Dockerized the frontend codes to standardize the working environment. Implemented the Google Map interface to the Events page to show the events on the map as markers. Additionally, implement the info window functionality to show the events' details and related actions. The interface is implemented in an opening modal to improve user interface.

	<ol style="list-style-type: none"> 4. Implemented the autocomplete functionality to the Events and Create Event page to improve user experience on creating and searching events. 5. Revised the ReadMe file of frontendmain branch and added the instructions about running the dockerized app locally. 6. Performed pair programming meetings with the team to finalized functionalities before the presentation. 7. Deployed the frontend to AWS EC2 instance. 8. Performed a research for unit testing. 9. Participate in team meetings to plan the development roadmap.
--	--

ANDROID

TEAM MEMBER	CONTRIBUTION
Fatih Akgöz	<ol style="list-style-type: none"> 1. Organized and joined android team meetings. 2. Added simple authentication 3. Implemented event detail User Interface 4. Refactored models reflecting the changes in backend. 5. Implemented create event User Interface 6. Implemented map function for creating events. 7. Presented the android milestone customer demo. 8. Reviewed the pull requests Ece had created. 9. Filled the Web User Interfaces and Progress according to Requirements part of the milestone report.
Ece Dilara Aslan	<ol style="list-style-type: none"> 1. Attended Android meetings with Fatih. 2. Studied callback, LiveData, RecyclerView.Adapter in Kotlin. 3. Fixed backend connections of login, register and event list functionalities. 4. Fixed and designed event list UI. 5. Implemented apply and undo apply event functionality. 6. Created backend connection of create event functionality. 7. Redesigned event list and event detail UIs. 8. Reviewed Fatih's pull requests. 9. Attached links to Android User Interfaces section of the milestone report.

3. Progress According to Requirements

SRS req.num.	Requirements	Backend	Frontend	Android
	<u>-Functional Requirements-</u>			
	-User Requirements			
	Sign in/up			
3.1.1.1.1.	User shall include his/her name, username, password, e-mail during sign-up.	Completed	Completed	Completed
	Profile			
3.1.1.2.1.	User shall have interests with skill level, location, age and public badges shown on profile.	In progress	In progress	In progress
3.1.1.2.2.	User shall be able to see the private activity history.	In progress	Not Started	Not Started
3.1.1.2.3.	User shall be able to follow other users.	Not Started	Not Started	Not Started
3.1.1.2.4.	User shall be able to get notifications of the events that the users they follow create.	In progress	Not Started	Not Started
	Settings			

3.1.1.3.1.	User shall be able to delete his/her account.	Completed		
3.1.1.3.2.	User shall be able to customize notifications	Not Started	Not Started	Not Started
	Creating Events			
3.1.1.4.1.	Any user shall be able to create events.	Completed	Completed	Completed
3.1.1.4.2.	Event creator shall provide location, date, duration, and type of an event when creating an event.	Completed	Completed	Completed
3.1.1.4.3.	Type of an event shall include its name, rules, number of people needed, required equipment, skill level.	In Progress	In Progress	In Progress
3.1.1.4.4.	If the desired type cannot be found, user shall be able to use characteristics to define an event.	In Progress	In Progress	In Progress
3.1.1.4.5.	Event creator shall be able to check the appliers' badges or related information for whether to accept them or not.	In Progress	Not Started	Not Started
3.1.1.4.6.	Event creator shall select the attendees out of appliers.	Completed	Not Started	Not Started
3.1.1.4.7.	Event creator shall be able to create and give out badges to attendees.	Completed	Not Started	Not Started

	Searching and Attending an Event			
3.1.1.5.1.	User shall provide at least 3 kinds of sports and skill level for searching.	In Progress	Not Started	Not Started
3.1.1.5.2.	User shall be able to search by location, date, type, people, skill level.	In Progress	In Progress	In Progress
3.1.1.5.3.	User shall be able to apply to participate in or spectate an event.	In Progress	Not Started	Not Started
3.1.1.5.4.	User shall be informed on whether he/she was accepted to the event.	In Progress	Not Started	Not Started
3.1.1.5.5.	After the event, user shall be able to give badges defined by event creators to other people.	In Progress	Not Started	Not Started
3.1.1.5.6.	Event creator shall be able to contact system maintainers for desired badge inclusion.	Not Started	Not Started	Not Started
3.1.1.5.7.	User shall be able to comment on the event page.	Completed	In Progress	In Progress
	Buying Products and Other Services			
3.1.1.6.1.	User shall be able to search for equipment sales by sport and location.	Not Started	Not Started	Not Started
3.1.1.6.2.	User shall be able to check information of equipment sellers	Not Started	Not Started	Not Started

	and read descriptions written by them.			
3.1.1.6.3.	User shall be able to create equipment sale posts that include its description.	In Progress	In Progress	In Progress
	-System Requirements-			
	Sign in/up			
3.1.2.1.1.	The system shall provide users with options to create a new account or sign-in to an existing.	Completed	Completed	Completed
3.1.2.1.2.	The system shall allow users to enter information about the kinds of sports they are interested in and their skill level while they are signing up.	In Progress	In Progress	In progress
3.1.2.1.3.	The system should allow users to recover their password by sending a password-recovery e-mail.	Not Started	Not Started	Not Started
3.1.2.1.4.	The system shall verify the entered information before creating an account for registering users. Users who provide invalid information should not be allowed to create an account.	In Progress	In Progress	In Progress

	Search			
3.1.2.2.1.	The system shall allow users to search for events, according to event name, location, sport, date.	In Progress	In Progress	In Progress
3.1.2.2.2.	The system shall allow users to search for other users and view their location, schedule and interests if they have provided such information.	Not Started	Not Started	Not Started
3.1.2.2.3.	The system shall allow users to search for other users by name, view their profile and follow other users.	In Progress	Not Started	Not Started
3.1.2.2.4.	The system shall allow users to search for equipment for any sport they are interested in	In Progress	Not Started	Not Started
	Events			
3.1.2.3.1.	The system shall provide information about the events such as event name, location, date, sport and the number of players.	Completed	Completed	Completed
3.1.2.3.2.	The system shall maintain and provide analysis information about the events.	Not Started	Not Started	Not Started
3.1.2.3.3.	The system shall allow users to	In Progress	Not Started	Not Started

	request joining an event.			
3.1.2.3.4.	The system shall allow users to create events by specifying event name, location, date, sport and the number of players.	Completed	Completed	Completed
3.1.2.3.5.	The system shall provide default sport types for the event creators to select from.	Not Started	Not Started	Not Started
3.1.2.3.6.	If no matching sports type exist, the system shall allow event creators to specify the event characteristics.	In progress	In progress	In progress
3.1.2.3.7.	The system shall send a pop-up message after a user creates an event.	Completed	Completed	Completed
3.1.2.3.8.	The system shall allow users to cancel events they have created.	Not Started	Not Started	Not Started
	Recommendations			
3.1.2.4.1.	The system shall recommend users events based on their location and sports they are interested in.	Not Started	Not Started	Not Started
3.1.2.4.2.	The system shall recommend event creators other users based on their location, interests and schedules.	Not Started	Not Started	Not Started

3.1.2.4.3.	The system shall recommend sport-related content(gadgets, clothing, tools, merchandise) to users for the sports they are interested in.	Not Started	Not Started	Not Started
	Notifications			
3.1.2.5.1.	The system shall send notifications to event creators when a user sends a request to one of their events.	Not Started	Not Started	Not Started
3.1.2.5.2.	The system shall send notifications to users who are accepted or rejected for that event.	Not Started	Not Started	Not Started
3.1.2.5.3.	The system shall send notifications to users when one of the users they are following creates an event.	Not Started	Not Started	Not Started
	Badges			
3.1.2.6.1.	The system shall allow users to give badges to others users who they have joined an event together with.	Not Started	Not Started	Not Started
3.1.2.6.2.	The system shall allow event creators to give badges to users who have participated in their events.	Completed	Not Started	Not Started
3.1.2.6.3.	The system shall provide a default selection of	Completed	Not Started	Not Started

	customizable badges.			
3.1.2.6.4.	The system should allow users to make requests for new badge types they would like to see available.	Not Started	Not Started	Not Started
	Interactions			
3.1.2.7.1.	The system shall allow users to block other users.	Not Started	Not Started	Not Started
3.1.2.7.2.	The system shall not allow users to view other users profiles who have blocked them.	Not Started	Not Started	Not Started
3.1.2.7.3.	The system should allow users to leave comments on the events they have participated in.	Completed	Not Started	Not Started
	<u>-Nonfunctional Requirements-</u>			
	Availability			
3.2.1.1.	The system shall run as a web application, available through modern web browsers & an Android application.	Completed	Completed	Completed
3.2.1.2.	Cross-platform languages shall not be used implementing the system.	In Progress	In Progress	In Progress
3.2.1.3.	The system language shall be English.	In Progress	In Progress	In Progress

	Standards			
3.2.2.1.	The system shall follow W3C Activity Stream Standard 2.0 to implement subscriptions & notifications	Completed	In Progress	In Progress
3.2.2.2.	The semantic taggings should be supported with Wikidata.	Completed	In Progress	In Progress
	Privacy			
3.2.3.1.	Users shall accept a privacy policy during a sign-up, setting permissions on personal data.	Not Started	Not Started	Not Started
3.2.3.2.	Users should be able to block each other, further specifying who can see their data.	Not Started	Not Started	Not Started
3.2.3.3.	Users shall not be asked to grant any information besides than a name & an e-mail address.	In Progress	In Progress	In Progress
3.2.3.4.	Data should be extracted out of what users share and allow to be visible, to analyze statistically and optimize user experience.	Not Started	Not Started	Not Started
3.2.3.5.	User data shall be stored and processed according to General Data Protection Regulation(GDPR) and KVKK.	Not Started	Not Started	Not Started

	Security			
3.2.4.1.	Users shall not be able to track transaction history of other users.	Completed	In Progress	In Progress
3.2.4.2.	System shall not be accessible through old versions of browsers.	Not Started	Not Started	Not Started
3.2.4.3.	E-mail and password info for users shall be encrypted before storage.	Completed	Completed	Completed
3.2.4.4.	No user shall be granted admin permissions.	Completed	Completed	Completed

4. API Endpoints

Link: <http://3.67.188.187:8000/api/docs/>

Additional Documentation:

<https://github.com/bounswe/2021SpringGroup4/blob/backend-main/code/README.md>

You can find some of the available endpoint in the POSTMAN collection:

<https://www.getpostman.com/collections/fcba76c7f8264873047b>

Most API endpoints run with JWT authentication. You need to register and then login and append the access token you receive as a Bearer token for requests to other endpoints.

5. Implementation of core functionality

5.1. Mapping Functionality

1. Describe the status of this functionality

We have significantly completed the Map functionality. Now, both in web application and android, the user can select a place from the map and send the selected location (coordinates and also location as text) to the backend service.

2. Describe in detail the work that has been completed regarding this functionality

On frontend :

- Google Map interface implemented via using Google Map Api Key and "react-google-maps".
- Events are located on as a markers according to the their latitude and longitude fields.
- Info Window functionality is added to the markers to present events' details and possible actions.
- Search and autocomplete functionality is implemented to the map to navigate on the map efficiently.

When users creates an event :

- users could select an place by clicking on the map
- a marker placed on the clicked place
- Latitude and Longitude values of the adress is kept as a variable and sent to the backend api events endpoint
- One of the address (from the ten) values returned from google map service is selected and sent to the backend api events endpoint (as location parameter)

When users searches for an event (on frontend):

- Events created by users listed in map according to the latitude and longitude fields.
- Markers are placed on the events locations
- With the help of this interface, users could track the events on map and show the details (infowindow) and apply.

3. Describe the work that remains for this functionality

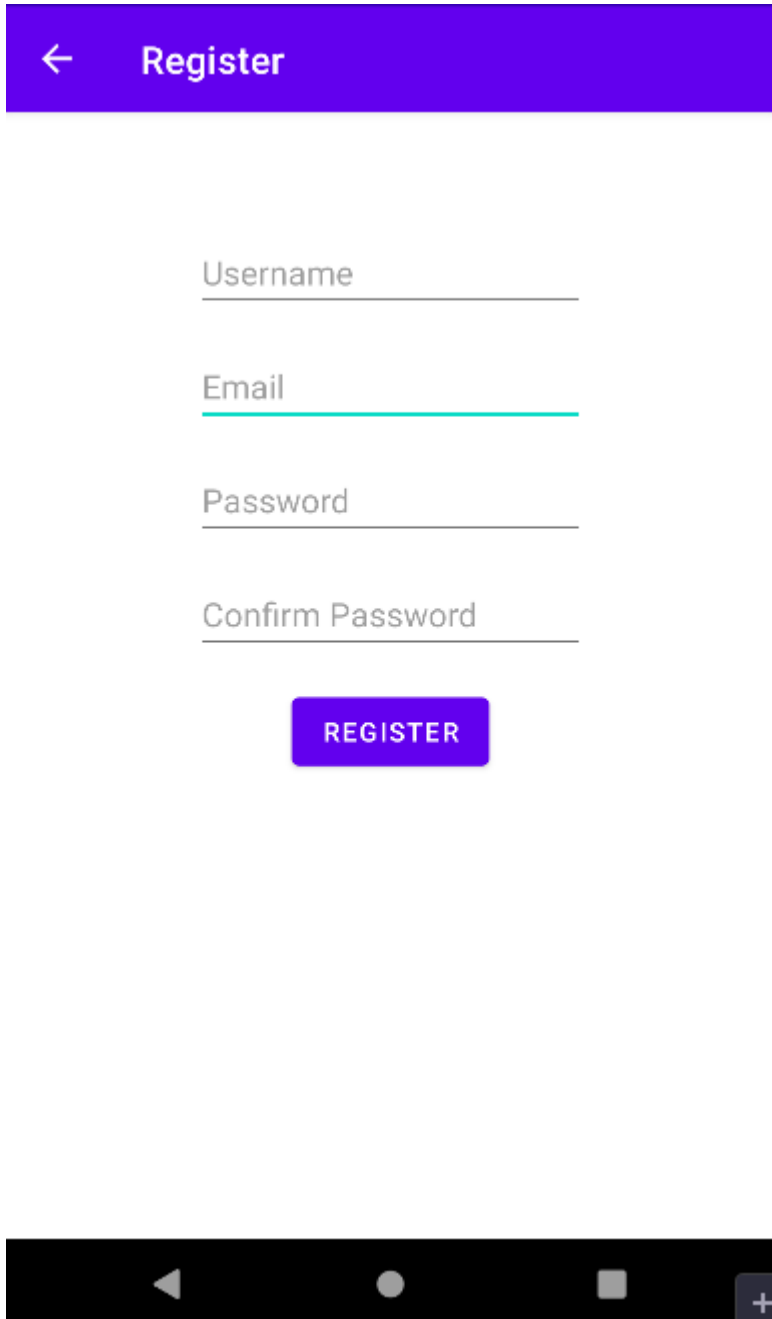
Search event with mapping functionality will be implemented on Android. Although the map functionality has been completed in general on frontend, the feature of obtaining the address of the user by determining the location (google geolocation api) from the browser can be added to the application. Also Unit tests need to be written for map functions.

6. Android User Interfaces

6.1. Register page

Link:

https://github.com/bounswe/2021SpringGroup4/blob/main/app/src/main/res/layout/fragment_register.xml

A screenshot of a mobile application's 'Register' screen. The screen has a purple header bar with a white back arrow and the word 'Register'. Below the header, there are four text input fields labeled 'Username', 'Email', 'Password', and 'Confirm Password'. The 'Email' field has a red underline. Below the input fields is a purple button with the text 'REGISTER' in white. At the bottom of the screen is a black navigation bar with three icons: a back arrow, a circle, and a square, and a small purple button with a white plus sign on the right.

← Register

Username

Email

Password

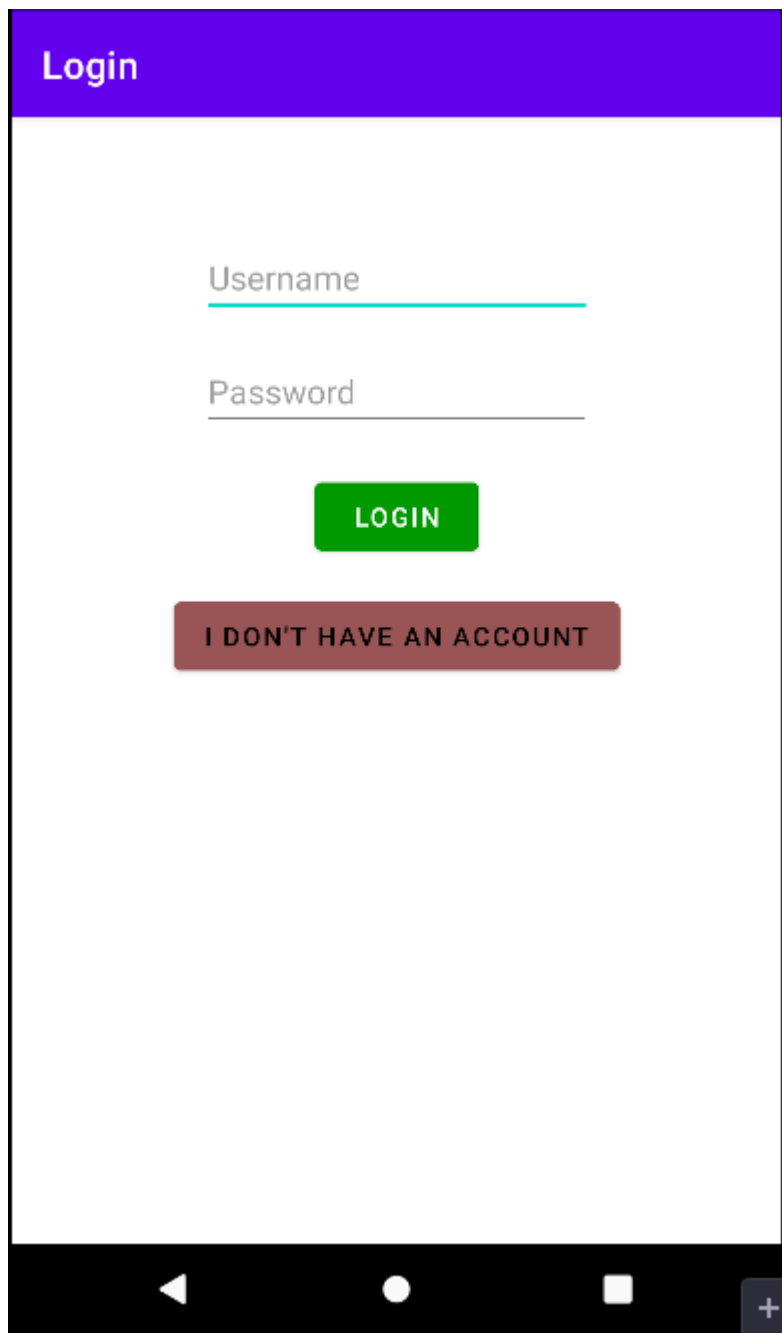
Confirm Password

REGISTER

6.2. Login Page

Link:

https://github.com/bounswe/2021SpringGroup4/blob/android-main/app/src/main/res/layout/fragment_login.xml



6.3. List Events Page

Link:

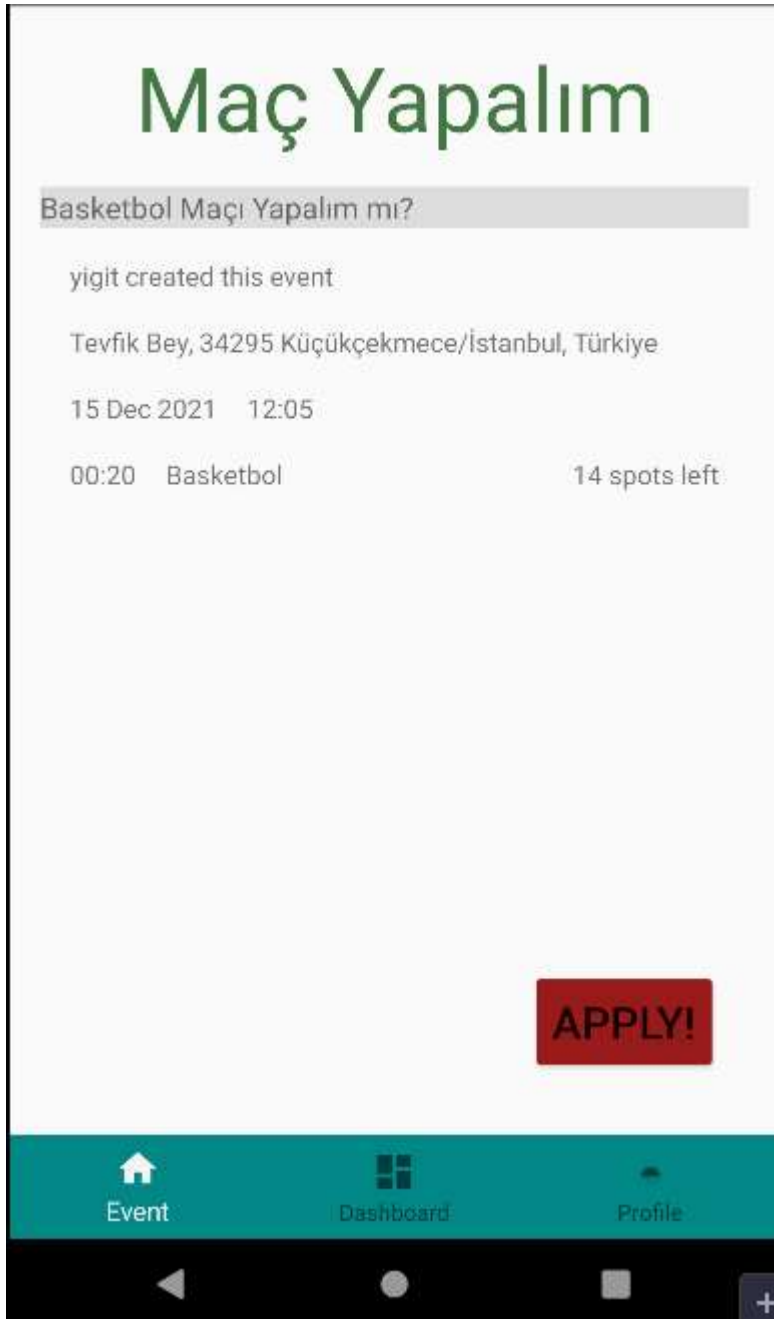
https://github.com/bounswe/2021SpringGroup4/blob/android-main/app/src/main/res/layout/fragment_event.xml



6.4. Event Detail Page

Link:

https://github.com/bounswe/2021SpringGroup4/blob/main/app/src/main/res/layout/event_detail_fragment.xml



6.5. Create Event page

Link:

https://github.com/bounswe/2021SpringGroup4/blob/main/app/src/main/res/layout/event_create_fragment.xml

Create Event

Event title

Event description

Sport type

Skill level

Maximum number of players

SET LOCATION

2021

Sat, Dec 18

December 2021



Event



Dashboard



Profile



00 25

01 : 26

02 27

Duration

23 59

00 : 00

01 01

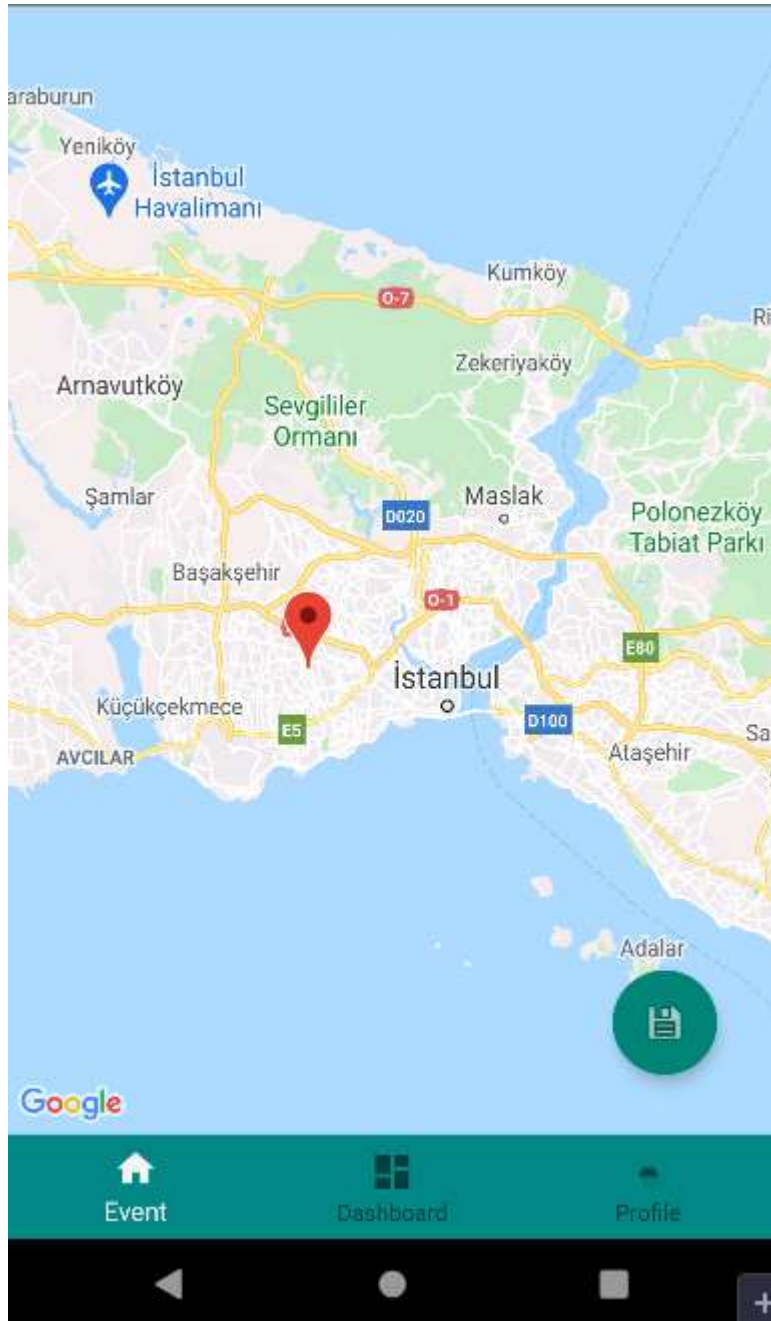
SUBMIT

Event Dashboard Profile

6.6. Pick Location Page

Link:

https://github.com/bounswe/2021SpringGroup4/blob/main/main/app/src/main/res/layout/fragment_create_event_map.xml



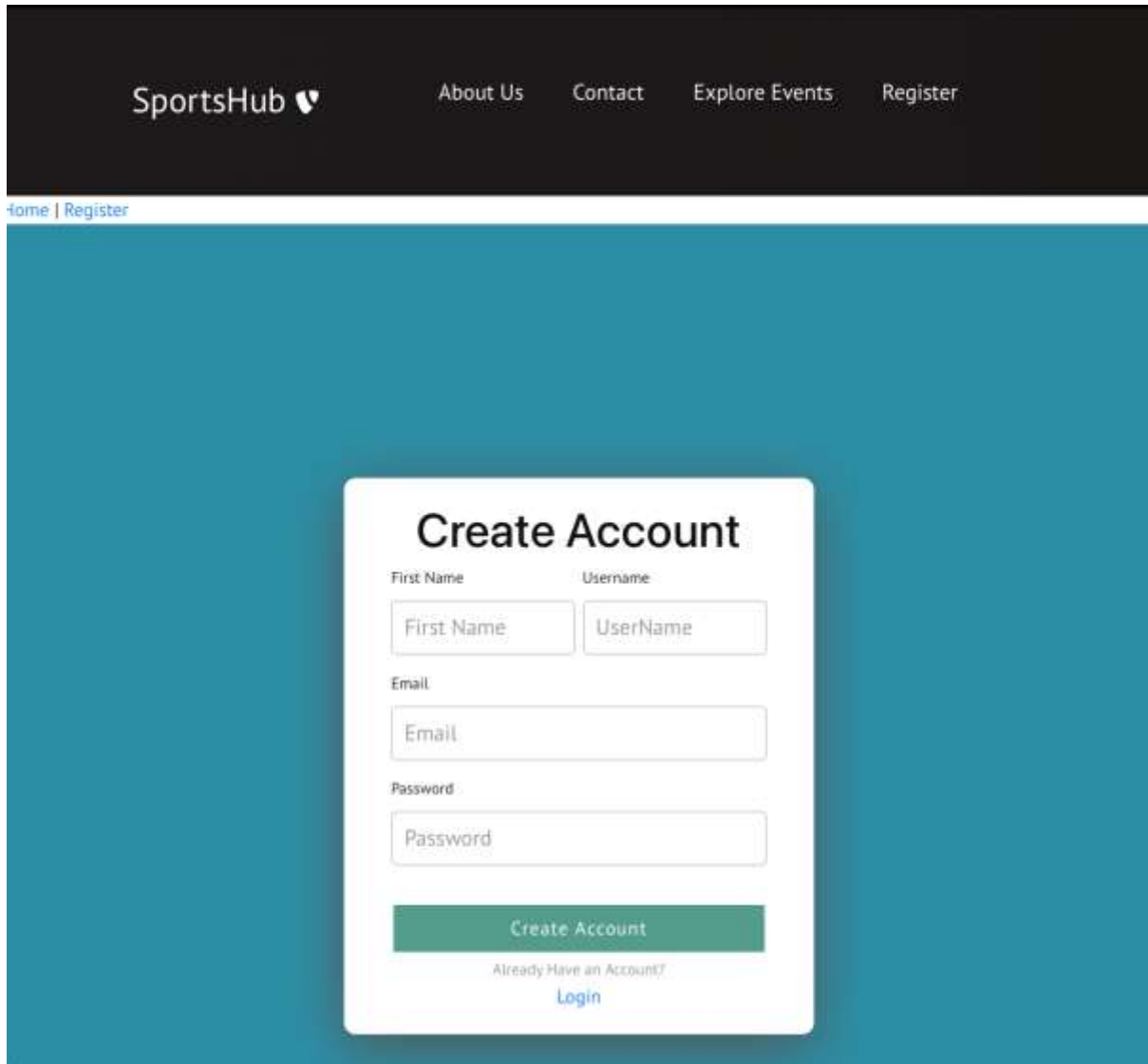
7.1.Web Interfaces

7.1.1- Register Page

Link to Code in project repository:

<https://github.com/bounswe/2021SpringGroup4/blob/master/code/frontend/src/pages/Register.js>

Screenshot:




The screenshot displays the 'Register' page of the SportsHub application. At the top, a dark navigation bar contains the 'SportsHub' logo and links for 'About Us', 'Contact', 'Explore Events', and 'Register'. Below this, a breadcrumb trail shows 'Home | Register'. The main content area has a teal background and features a white 'Create Account' form. The form includes input fields for 'First Name', 'Username', 'Email', and 'Password', each with a placeholder label. A green 'Create Account' button is positioned below the inputs. At the bottom of the form, there is a link that says 'Already Have an Account? Login'.


7.1.2- Home Page

Link to Code in project repository:

<https://github.com/bounswe/2021SpringGroup4/blob/master/code/frontend/src/pages/Home.js>

Screenshot:

SportsHub 

[About Us](#) [Contact](#) [Explore Events](#) [Register](#) 

[Home](#) | [Register](#)

UserName


Enter Username


Password

Login

Or Sign Up Using

[Sign Up](#)

SportsHub 



About Us

Contact

Explore Events

Register

[7.1.3- Login Page](#)

Link to Code in project repository:

<https://github.com/bounswe/2021SpringGroup4/blob/master/code/frontend/src/pages/Login.js>

Screenshot:

SportsHub

Home | Register

UserName

Enter Username

Password

Login

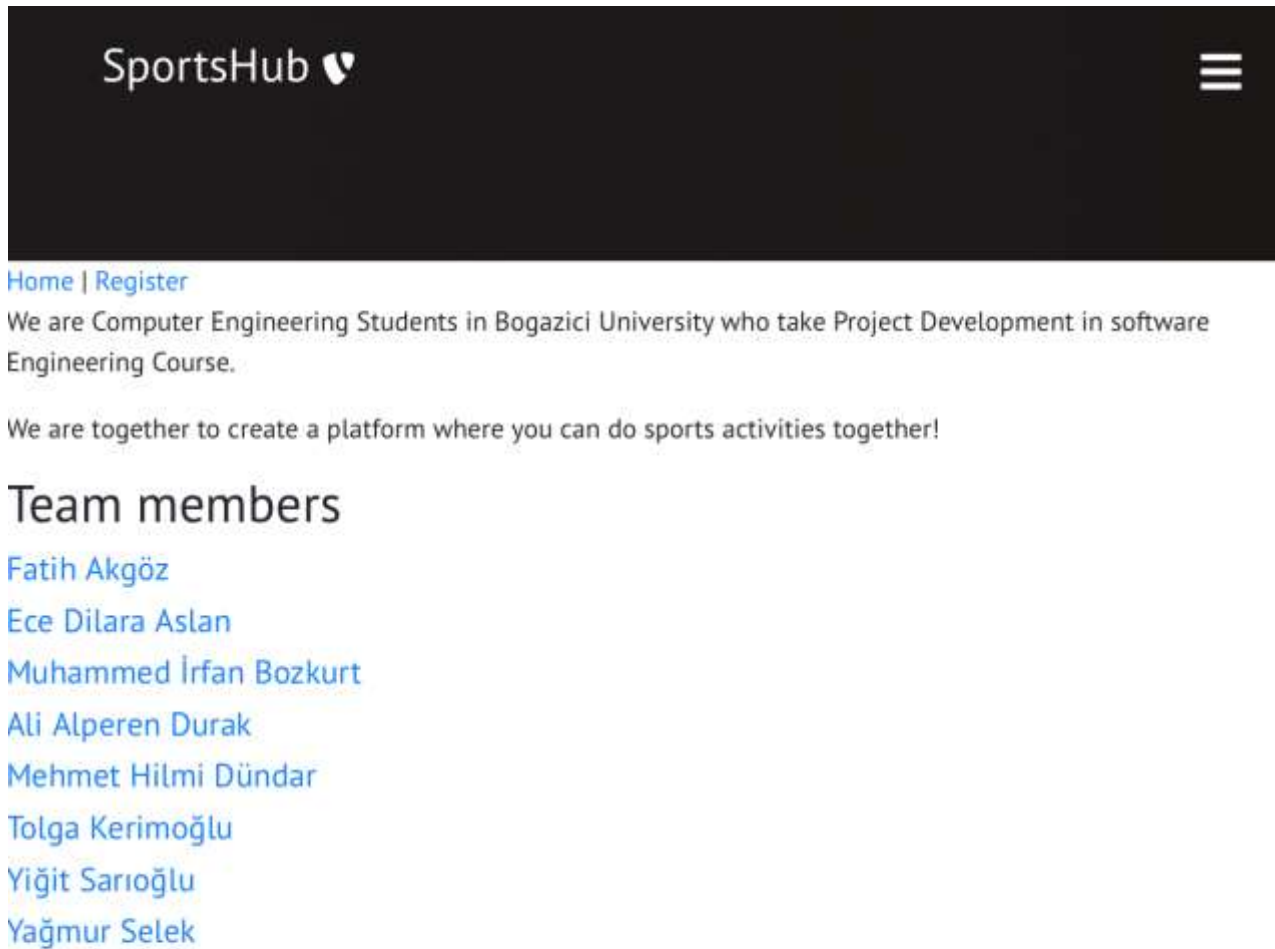
Or Sign Up Using
[Sign Up](#)

[7.1.4-About Us Page](#)

Link to Code in project repository:

<https://github.com/bounswe/2021SpringGroup4/blob/master/code/frontend/src/pages/About.js>

Screenshot:

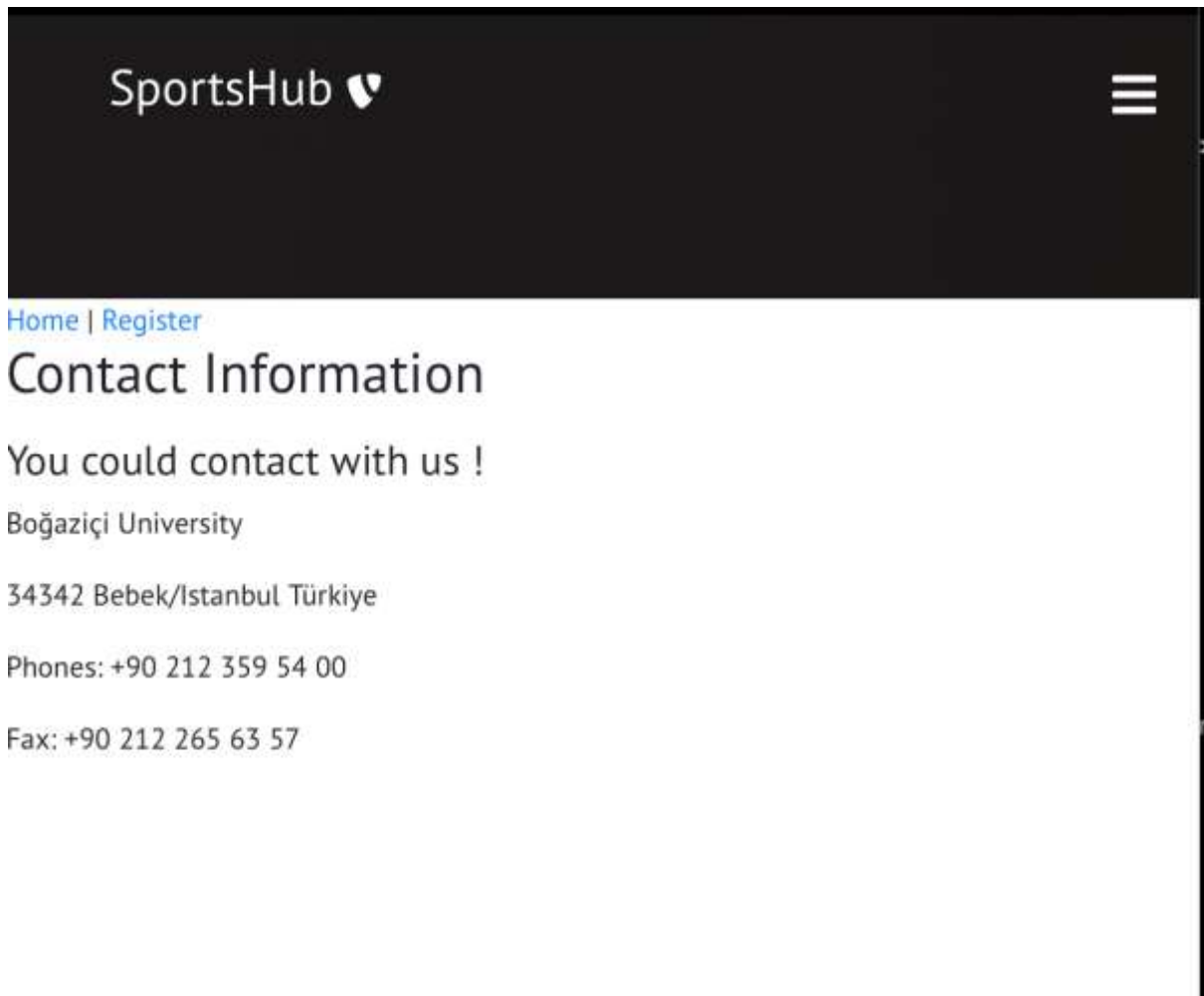


[7.1.5-Contact Page](#)

Link to Code in project repository:

<https://github.com/bounswe/2021SpringGroup4/blob/master/code/frontend/src/pages/Contact.js>

Screenshot:

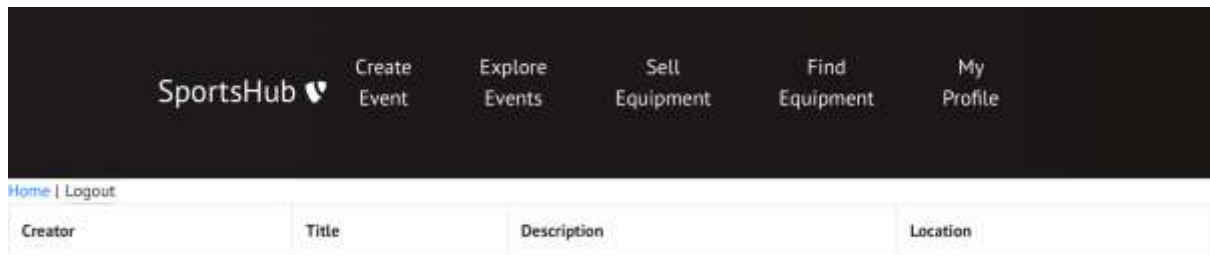


[7.1.6-Find Equipment Page](#)

Link to Code in project repository:

<https://github.com/bounswe/2021SpringGroup4/blob/master/code/frontend/src/pages/EquipmentList.js>

Screenshot:



[7.1.7- Sell Equipment Page](#)

Link to Code in project repository:

<https://github.com/bounswe/2021SpringGroup4/blob/master/code/frontend/src/pages/Equipment.js>

The screenshot shows the 'Create New Equipment' form in the SportsHub application. The form has a dark header with the SportsHub logo and navigation links. Below the header, the title 'Create New Equipment' is centered. The form itself consists of three input fields labeled 'Title', 'Description', and 'Location'. A 'Submit' button is located at the bottom left of the form.

[7.1.8- Profile Page](#)

Link to Code in project repository:

<https://github.com/bounswe/2021SpringGroup4/blob/master/code/frontend/src/pages/UserProfile.js>

<https://github.com/bounswe/2021SpringGroup4/blob/master/code/frontend/src/pages/EditUserProfile.js>

Screenshot:

SportsHub

Create Event

Explore Events


Sell Equipment

Find Equipment

My Profile

[Home](#) | [Logout](#)

User Profile

Profile photo:	
Username:	yagmur
E-mail:	yagmur1@gmail.com
First Name:	yagmur
Last Name:	selek
Age:	24
About:	Computer Engineering Student
Location:	Istanbul
Going:	
Applied:	

You could edit your profile


[Edit Profile](#)

7.1.9- Create Event Page

Link to Code in project repository:

<https://github.com/bounswe/2021SpringGroup4/blob/master/code/frontend/src/pages/EventCreator.js>

Screenshot:

SportsHub 


[Create Event](#)[Explore Events](#)[Sell Equipment](#)[Find Equipment](#)[My Profile](#)

[Home](#) | [Logout](#)

Create New Event

Title	<input type="text"/>
Description	<input type="text"/>
Date	<input type="text" value="11/17/2021"/>
Time	<input type="text" value="12:30 PM"/>
Duration	<input type="text" value="12:30 PM"/>
Sport Type	<input type="text"/>
Maximum Players	<input type="text"/>
Choose a skill level:	<input type="button" value="Beginner"/>

MapSatellite



Enter Location

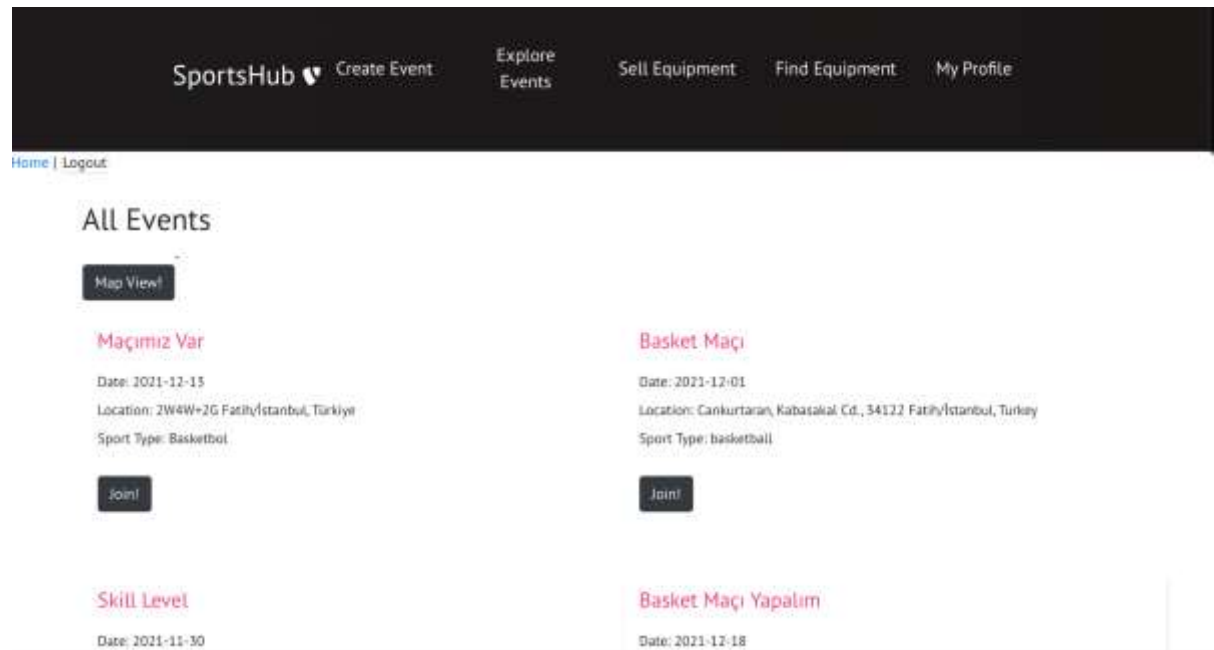
Create

7.1.10 - Explore Events Page

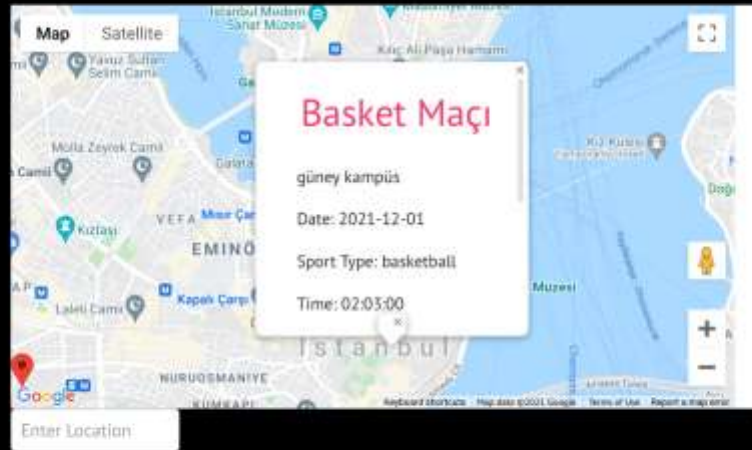
Link to Code in project repository:

<https://github.com/bounswe/2021SpringGroup4/blob/master/code/frontend/src/pages/SearchEvents.js>

Screenshot:



event detail



7. Standards

W3C activity stream standards: There is an activity handler module that creates formatted data representations anytime a defined event is triggered. This data is then used for communication between different backend apps. A more detailed view can be found in the related issue:

<https://github.com/bounswe/2021SpringGroup4/issues/176>

Wikidata: Our badges are extracted from Wikidata. You can find the matching in the related issue: <https://github.com/bounswe/2021SpringGroup4/issues/184>