

HBDA Score Tracker Sprint 2

Zach Bender • Tyler Blair • Trey Thorne • Tessa Tryon

Source Code: https://github.com/tessa-hudson/Capstone_Fall2021

Project Site: https://tessa-hudson.github.io/Capstone_Fall2021



Project Focus

Who it's For

Hemophilia and Bleeding Disorders of Alabama (HBDA) for use with event attendance and summer camp scoreboard.

What it Does

Use a point tracking system for a friendly competition between groups of campers throughout the week¹

Track attendance of attendees for specific events (other than the summer camp)

Sprint Goals

- ⬡ Determine which service we will be using for hosting the app
- ⬡ Implement User Authentication
- ⬡ Implement User Roles
- ⬡ Allow a superuser to add new users

Project Backlog

ID	Story	Estimation (hours)	Priority	Sprint When Finished
1	Set up database	4	1	1
2	Set up python/flask	1	1	1
3	Set up React App	0.5	1	1
4	Users should be able to sign up and log in	12	2	2
5	Users should be able to view a scoreboard with points for each team or camper	4	1	1
6	Superusers/Admins should be able to add campers	4	1	1
7	Superusers/Admins should be able to add campers to groups/teams	5	1	2
8	Superusers/Admins should be able to add/subtract points	5	1	2
9	Superusers/Admins should be able to remove campers	3	3	
10	Superusers/Admins should be able to update campers	5	2	2
11	Superusers/Admins should be able to remove campers from a team	5	3	
12	Superusers/Admins should be able to remove users	4	4	
13	Superusers/Admins should be able to change user access	8	3	2
14	Superusers/admins should be able to track camper attendance	5	4	
15	User Interface should be easy to use on a mobile device	8	5	
16	Users should be able to update their password	6	5	2
17	Users should be able to make point update requests	6	4	
18	Superusers/Admins should be able to approve or deny point requests from users with less access	8	4	
19	Find a longterm host for web app	10	2	2
20	App should have simple navigation	4	4	
21	Add styling/theme to app	12	5	
22	Users can only sign up with approval from admin (link)	10	3	

Sprint Backlog

- ⬡ Users should be able to sign up and log in
- ⬡ Superusers/Admins should be able to add attendees to groups/teams
- ⬡ Superusers/Admins should be able to add/subtract points
- ⬡ Find a longterm host for web app
- ⬡ Superusers/Admins should be able to update campers
- ⬡ Superusers/Admins should be able to change user access

Major Achievements



Major Achievements - Auth0

- Each Group member created an account on the Auth0 site
- A project was created that supports our HBDA scorekeeper
- Through the use of the auth0 api, we were able to add a login and authentication to our scorekeeper site
- The content of the site now can only be accessed by verified users
- Users are able to login and signup through a user friendly interface

Major Achievements - Deployment

- Deployed backend (Python Flask API) using Azure App Services and GitHub Actions.
 - Build app successfully.
 - Deploy to Azure App Service.
 - Confirm that the deployed service can successfully connect to the database.
- Deployed frontend using Azure App Services and GitHub Actions.
 - Build app successfully.
 - Deploy to Azure App Service.
 - Check secure configuration, dependencies, and redirects successfully
- Developed YAML that is compatible with the React and Flask components used in development.

Major Achievements - Functionality

- Attendees can now be added to groups
- Group's points can be updated
- Events can be added and updated
- Added cascading deletion, so when an event is deleted, any groups connected to that event are also deleted and any attendees in those groups are no longer linked.
 - Also applies if only a group is deleted, attendee links are severed to the deleted group

Major Achievements - Documentation

- Full documentation of the API on project website - https://tessa-hudson.github.io/Capstone_Fall2021/api/
- Data models for each entity with types and descriptions
- List of available endpoints with functionality descriptions, example requests and example responses

Events

Events are the main focus of the app. Each event has a separate scoreboard and groups added to the event will be added to the scoreboard with a count representing the total points the group has earned.

Data Model

Name	Type	Description
event_id	UUID	Used to uniquely identify the event. Generated upon creation.
event_name	string	The name of the event; subsequently, the name of the scoreboard.
event_type	string	A short description of the event (ie camp, meeting, etc).
start_date	date	The day the event starts in ISO format: YYYY-MM-DD
end_date	date	The day the event ends in ISO format: YYYY-MM-DD

Endpoints

GET /events

Description: Lists all events. Outputs event_id, event_name, type, start_date and end_date of each.

Authentication:

Request Example:

```
GET /events
Accept: application/vnd.api+json
```

Response Example:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.api+json
{
  "events": [
    {
      "event_id": "F20ACF22-90B2-4A4B-9B53-SFC80DA47EDB",
      "end_date": "2021-10-30",
      "event_name": "New Event",
      "event_type": "Test",
      "start_date": "2021-10-29"
    },
    {
      "event_id": "F811FB42-C609-4A13-B0B2-AC6D74990E71",
      "end_date": "2022-06-30",
      "event_name": "Camp",
      "event_type": "Camp Clot Not 2022",
      "start_date": "2022-07-11"
    }
  ]
}
```

Challenges

Deployment

- ⬡ Backend - DB drivers caused incompatibility issues with a production environment.
- ⬡ Frontend - Building the project with Vite vs. Create-React-App
- ⬡ Pipelines - initially started by connecting repo to Azure Pipelines, but build issues forced a switch to GitHub Actions. Also briefly tried Heroku
- ⬡ Overall - little documentation on deploying (specifically) a Vite React application with a Flask middleware component.

Data Management

- ⬡ Deletion/updating of data (groups, points, etc.) touches many different areas (i.e. has a cascading effect)

Lessons Learned

- ⬡ Testing should be more thorough, especially regression testing
- ⬡ Workload should be distributed evenly through the sprint
- ⬡ Team needs to have a shared vision of the final product in mind at all times so the code integrates well
- ⬡ Project Backlog should be more dynamic to accurately represent the project as requirements/features change

Demo

1. Overview of new website functionality
2. Backend functions not yet on site

