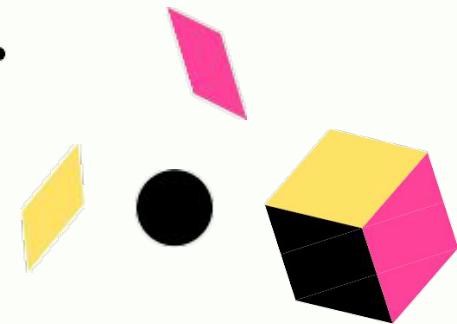


Group 4
Richard Bent
Luisa Cardona
Samantha Perez
Christopher Polynice
Cody Traywick

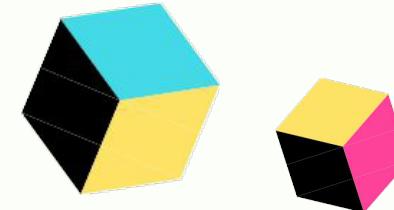


Sponsor
Margarita Azbel



Introductions

- Richard Bent – Backend Testing, Backend Developer
- Luisa Cardona – Project Manager, Frontend Designer
- Samantha Perez – Lead Frontend Designer
- Christopher Polynice – Lead Tester, Backend Developer
- Cody Traywick – QA, Lead Backend Developer



Who is Orlando Math Circle (OMC)?

Orlando Math Circle aims to demystify mathematics for students of all ages through playful, joyous problem solving

- **Mission** — Create a diverse and inclusive community of student mathematicians
- **Strategy** — Provide engaging and fun opportunities to do mathematics outside of school
- **Vision** — Equitable access and greater student participation in mathematics
- 501(C)(3) non-profit organization



Existing Solution

An event calendar system with an app-like experience on a mobile browser.

- Event management and check-in system
- Volunteer management
- Email notifications for events and newsletters
- Extended usability through admin panel
- Open source, documented, and extensible

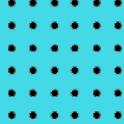
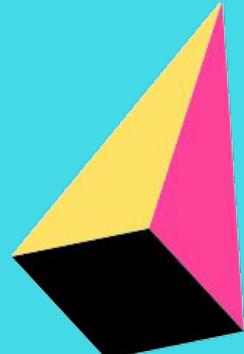
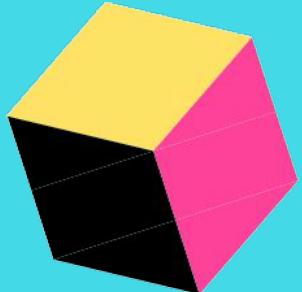
The screenshot displays a mobile application for the Orlando Math Circle. At the top, there's a navigation bar with three horizontal dots on the right. Below it is a header with the text "Orlando Math Circle". The main content area features a calendar for November 2020, with the 24th highlighted in blue. Below the calendar, a section titled "Events on November 24th" lists a single event: "JavaScript Course" on Tuesday, Nov 24th, 2020, at 11:00 AM, for Jane Doe, who is online. To the right of this event details is a small icon of a person at a computer. At the bottom of the screen are four navigation icons: Home (house), Events (calendar), Projects (puzzle), and Account (person). To the right of the main content area, there are two vertical panels. The first panel shows a registration list with "Jane Doe Grade 6 — Selected" checked. The second panel shows a payment section with a yellow "PayPal" button and navigation icons at the bottom.

Project Goals & Requirements

We want to improve the existing application for OMC to be self-sufficient while including volunteers and high school students in the process.

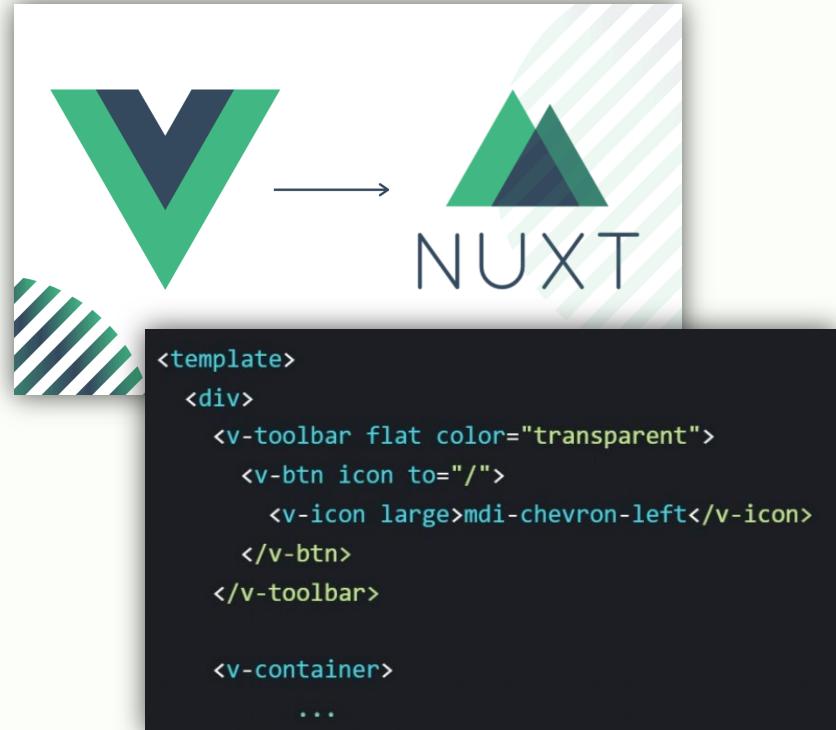
- New volunteer dashboard
- Membership and attendance management
- Allow swapping of shifts between volunteers
- Gamification through a point system
- Event time management

Frontend



Frontend Overview

- Vue.js through the Nuxt.js framework
- Server-Side Rendered (SSR)
- Single-Page Application (SPA)
- Vuetify component library
- Separation of concern in single-file components
- UI/UX design by the previous Senior Design team





<

Registration

Welcome to Orlando Math Circle
Already registered? [Log in](#)

First Name	Last Name	
Birthday Month ▾	Day	Year
Gender	?	
Email		
Password		
Confirm Password		

Are you an industry professional?

<

Sign in

Email	
Password	?

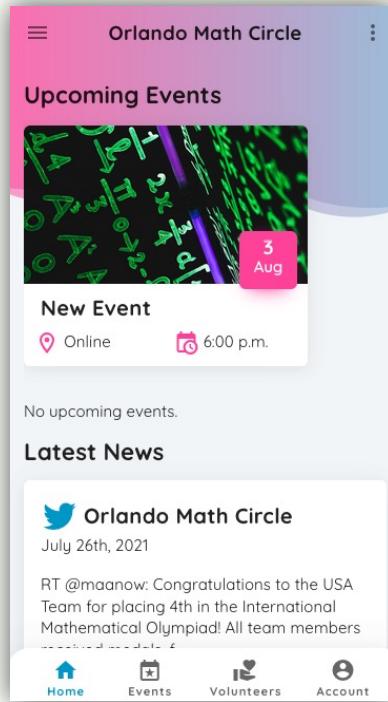
Remember me? [Forgot password?](#)

LOG IN

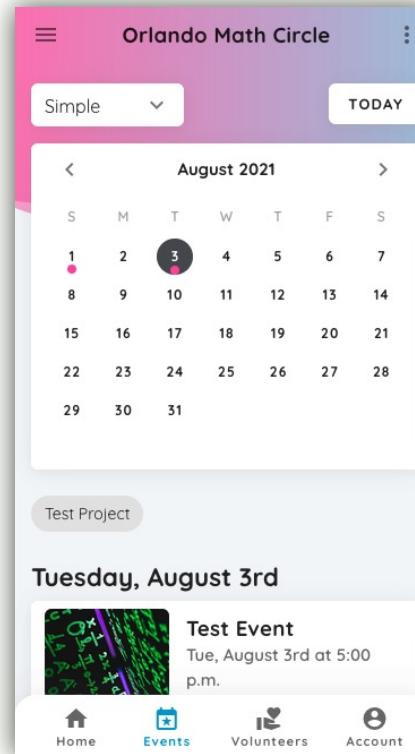
or

SIGN UP

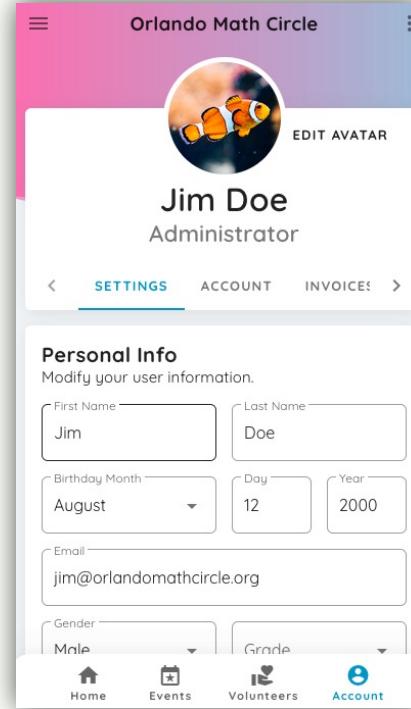
Landing Pages



Home Page



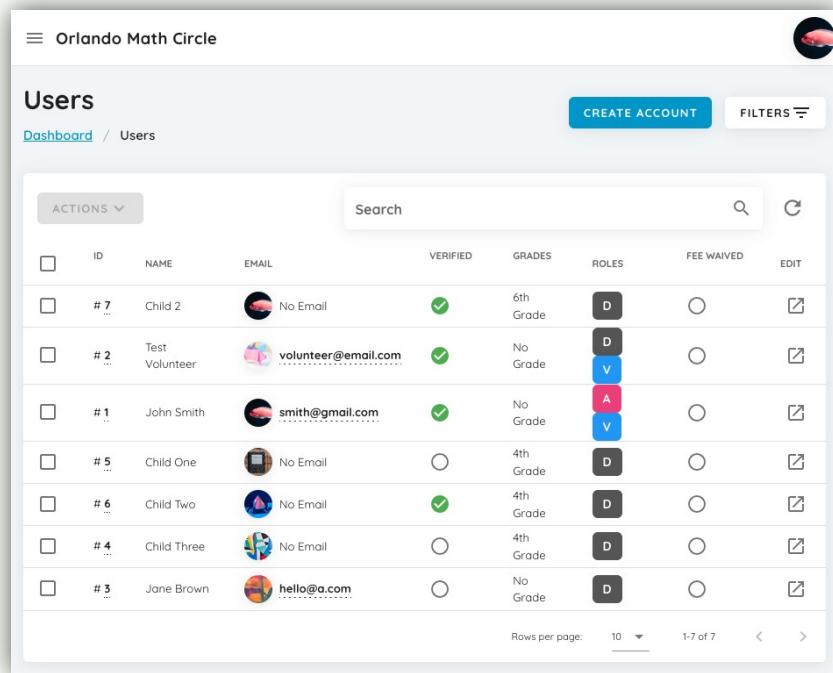
Events Page



Account Page

Admin Panel

- Available only to users with the administrator role
- Activity Records page has been added to access the shift swap request log
- New attendance page for admins to edit event attendance



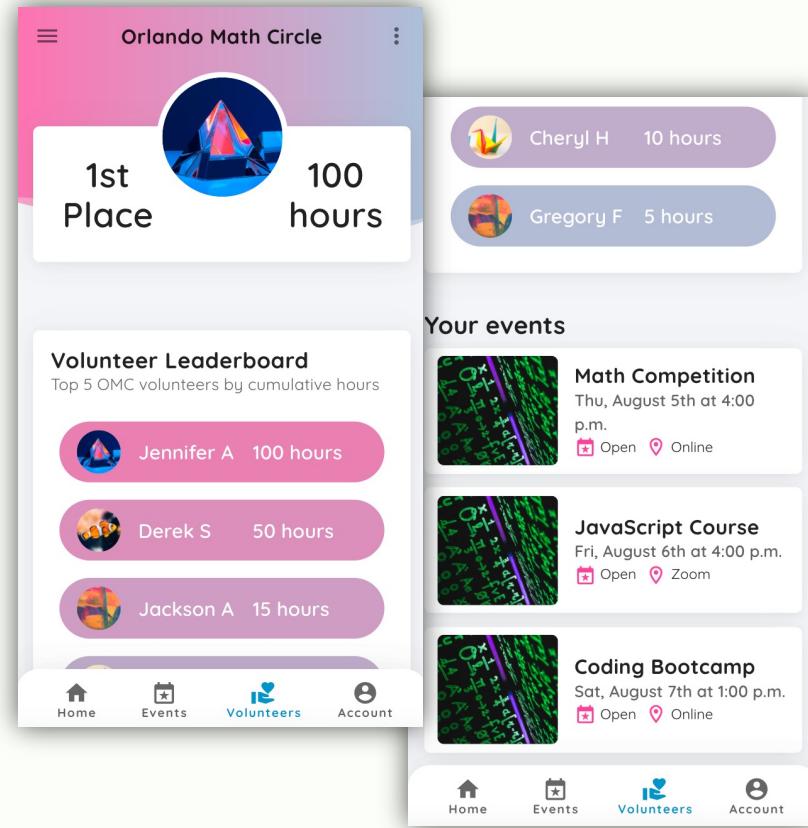
The screenshot shows the 'Users' section of the Orlando Math Circle Admin Panel. The interface includes a header with the organization name, a profile picture, and buttons for 'CREATE ACCOUNT' and 'FILTERS'. Below the header is a breadcrumb navigation showing 'Dashboard / Users'. A search bar and filter icons are also present. The main content is a table with columns: ACTIONS, ID, NAME, EMAIL, VERIFIED, GRADES, ROLES, FEE WAIVED, and EDIT. The table lists eight users:

ACTIONS	ID	NAME	EMAIL	VERIFIED	GRADES	ROLES	FEE WAIVED	EDIT
<input type="checkbox"/>	# 7	Child 2	No Email	✓	6th Grade	D	<input type="radio"/>	
<input type="checkbox"/>	# 2	Test Volunteer	volunteer@email.com	✓	No Grade	D V	<input type="radio"/>	
<input type="checkbox"/>	# 1	John Smith	smith@gmail.com	✓	No Grade	A V	<input type="radio"/>	
<input type="checkbox"/>	# 5	Child One	No Email	<input type="radio"/>	4th Grade	D	<input type="radio"/>	
<input type="checkbox"/>	# 6	Child Two	No Email	✓	4th Grade	D	<input type="radio"/>	
<input type="checkbox"/>	# 4	Child Three	No Email	<input type="radio"/>	4th Grade	D	<input type="radio"/>	
<input type="checkbox"/>	# 3	Jane Brown	hello@a.com	<input type="radio"/>	No Grade	D	<input type="radio"/>	

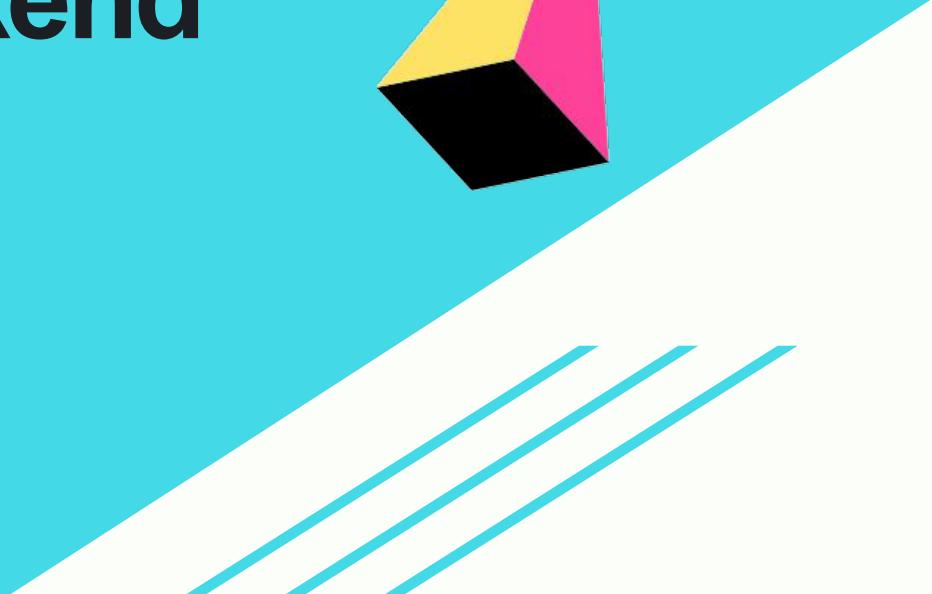
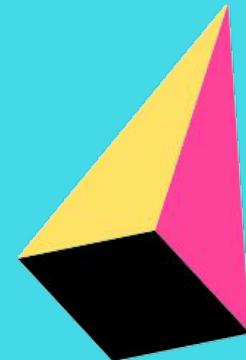
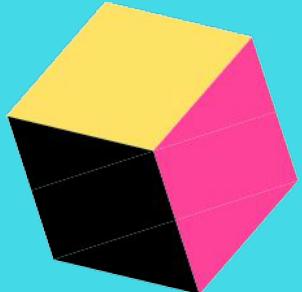
At the bottom, there are pagination controls for 'Rows per page' (set to 10), a page number '1-7 of 7', and navigation arrows.

Volunteer Dashboard

- Reengineered the existing Projects page into a more functional space for OMC
- Proposed by the Senior Design team
- Increase user engagement in the mobile app through friendly competition
- Leaderboard standing and accrued hours
- View personal upcoming and past events



Backend



Backend Overview

- PostgreSQL database with MikroORM
- Node.js and Nest.js
- Modular, class-based, and opinionated
- TypeScript and custom third-party interfaces



Backend Testing

- Insomnia
- Create API requests
- Test Backend features

The screenshot shows the Insomnia REST Client interface. At the top, it says "Development" and "Cookies". On the right, there's a "POST" button, a URL field containing "/account/register", and a "Send" button.

In the main area, there's a sidebar with a "membership" section containing "POST capture", "GET findAll", "GET FindOne", "DEL delete", and "POST create". Below this is an "Account" section with a "POST Register" item highlighted, followed by "POST Create", "GET Me", "GET FindOne (:id)", "GET FindOneByUser (:id)", and "DEL Delete (:id)".

The main panel shows a JSON payload being sent:

```
1 {  
2   "email": "richard@bent.com",  
3   "first": "richard",  
4   "last": "bent",  
5   "gender": "male",  
6   "password": "richard",  
7   "dob": "2000-01-01T00:00:00.000Z"  
8 }
```

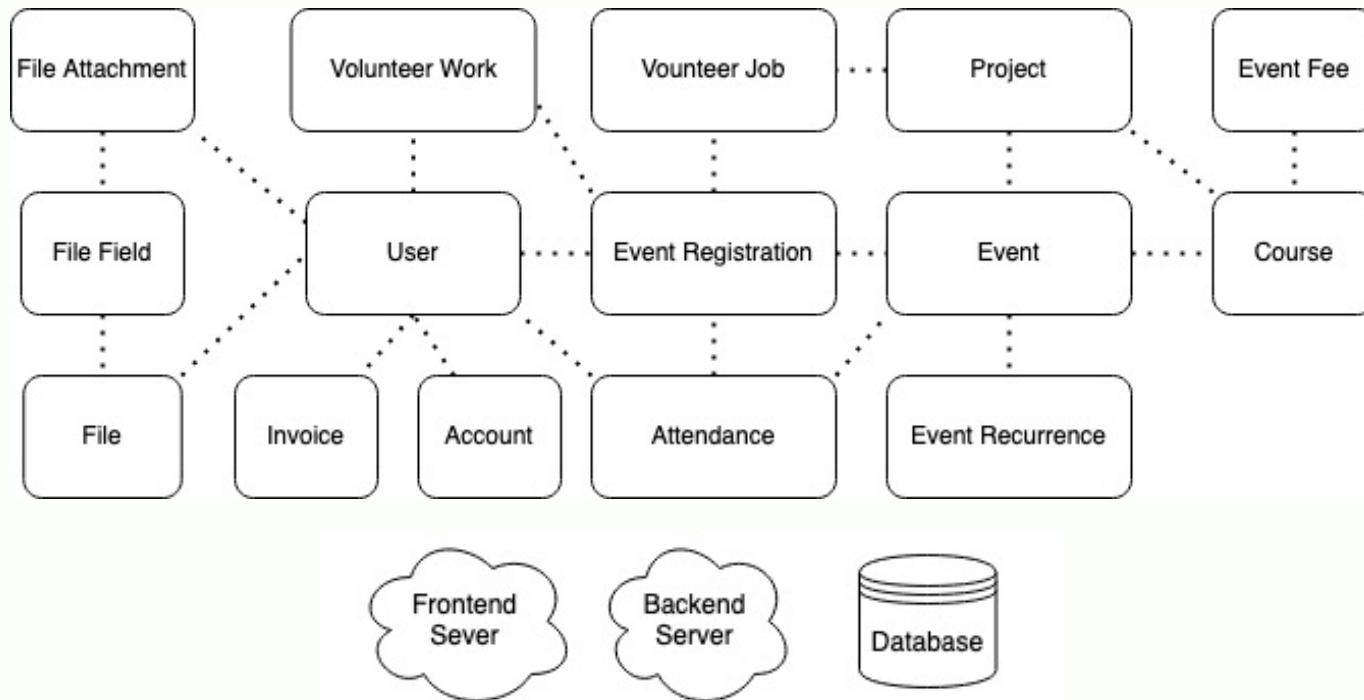
Below the payload, it says "Beautify JSON". The response status is "201 Created", the time is "365 ms", and the size is "649 B". The timestamp is "7 Days Ago".

The "Preview" tab is selected, showing the response body:

```
1 {  
2   "token":  
3     "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlaWQiOjM3LCJpYXQoIjE2M  
4     jc1MjIwMTN9.EgNzDGI0cNe4N4zSpDqHuZ9VxjA60kttwec6Ua15jAg",  
5   "complete": true,  
6   "user": {  
7     "id": 37,  
8     "roles": [  
9       "default"  
10    ],  
11  },  
12  "store": {  
13    "books": [  
14      {"author": "Richard Bent"}  
15    ]  
16  }  
17 }
```

The "Header" tab shows 7 items, and the "Cookie" and "Timeline" tabs are also present.

System Design



Attendance

- Both volunteers and students attend events
- Add hours from volunteer job to volunteer's account by creating a work entity from the job
- Transition from manual attendance checking and hour counts to automated calculating
- Admins can verify work/attendance through the admin panel

```
public async create({ userId, eventId, ...data }: MarkAttendanceDto) {  
  const attendance = this.attendanceRepository.create({  
    ...data,  
    user: userId,  
    event: eventId,  
  });  
  
  const event = await this.eventService.findOneOrFail(eventId);  
  
  const user = await this.userService.findOneOrFail(userId, [  
    'work',  
    'registrations',  
  ]);  
  
  // Check for user, registration, and permissions  
  
  if (event.isEnded) {  
    attendance.user = user;  
  
    if (!event.hasPermission(user)) {  
      throw new ForbiddenException();  
    }  
  }  
}
```

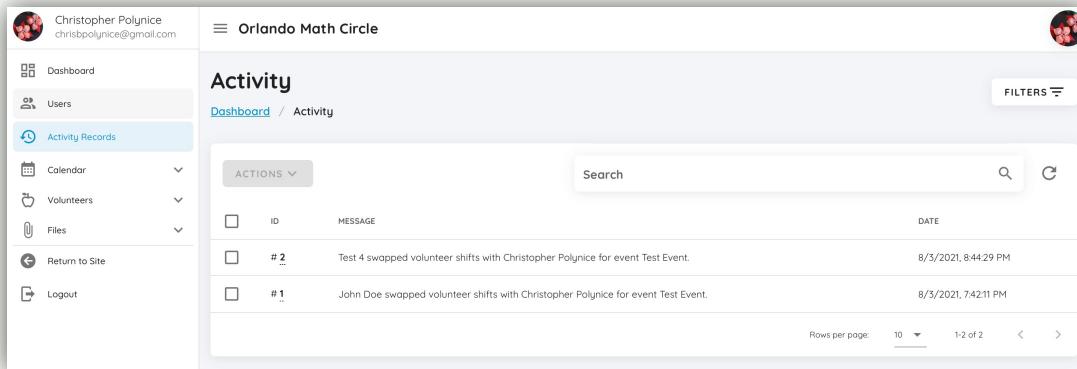
Volunteer Shift Swapping

- OMC v1.0 did not support transfer of event ownership
 - Volunteers needing shift coverage needed to handle process offline
- OMC v2.0 solves this
 - Process re-engineered to store user information as well as status of registration
 - Automatically handled in the application!

```
const registration = await this.registrationRepository.findOneOrFail(  
  { id, volunteering: true, isCoverable: true },  
  ['user', 'event.course'],  
);  
  
if (registration.event.isClosed) {  
  throw new BadRequestException('Event registrations are closed');  
} else if (registration.event.course?.isClosed) {  
  throw new BadRequestException('Course registrations are closed');  
}  
  
// Create audit log.  
await this.auditLogService.create({  
  userId: user.id,  
  changes: [  
    {  
      new_value: user.id,  
      old_value: registration.user.id,  
    }],  
  user:  
  type: AuditType.VOLUNTEER_SWAP,  
  target_id: registration.event.name,  
}, user);  
  
registration.assign({ user, isCoverable: false });
```

Volunteer Shift Swapping – Audit Log

- Volunteers handle swapping shifts without admin approval
- Admins still need to be informed of the swap
- Introduced a new Activity Records page with:
 - User ID
 - Details of the record
 - Date the transaction was initiated

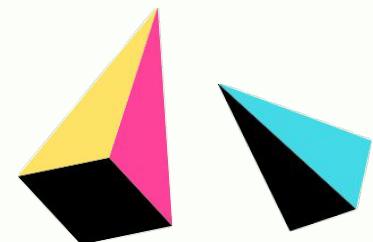


The screenshot shows a user interface for managing volunteer shifts. On the left, there's a sidebar with a profile picture of Christopher Polynice and his email (christbpolynice@gmail.com). The sidebar includes links for Dashboard, Users, Activity Records (which is currently selected and highlighted in blue), Calendar, Volunteers, Files, and Logout. The main content area has a header "Orlando Math Circle" and a sub-header "Activity". Below that, it says "Dashboard / Activity". There's a "FILTERS" button with a dropdown arrow. A search bar with a magnifying glass icon and a refresh/clear icon are also present. A "ROWS PER PAGE" dropdown set to 10, and a "1-2 of 2" page indicator. The main table lists two activity records:

ACTIONS	ID	MESSAGE	DATE
<input type="checkbox"/>	# 2 ..	Test 4 swapped volunteer shifts with Christopher Polynice for event Test Event.	8/3/2021, 8:44:29 PM
<input type="checkbox"/>	# 1 ..	John Doe swapped volunteer shifts with Christopher Polynice for event Test Event.	8/3/2021, 7:42:11 PM

Support of Additional Time to Events

- OMC v1.0 did not support editable events
 - after event creation
- Previously, editing event information required the event to be deleted and recreated
 - Problematic since event registration information is lost
- For OMC v2.0...
 - We went a step further and extended support to edit full event information
 - Naturally, this included event times, which can now be added down to the minute



Point System

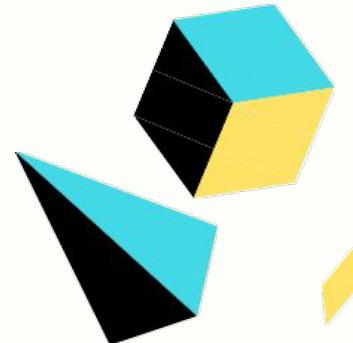
- Students will be rewarded points for volunteering and attending events
- Points will be given automatically for students who have recorded attendance for the events
- Users can view total accumulated points in the Account Page

```
@Property({ persist: false })
get points() {
  if (!this.attendances.isInitialized()) return null;

  return this.attendances
    .getItems()
    .filter((a) => a.event.isInitialized() && a.attended == true)
    .map((a) => (a?.event.points as number) || 0)
    .reduce((prev, cur) => prev + cur, 0);
}
```

Memberships

- \$50 – High School
 - Includes learning resources and workshops
 - Event registration
 - Must volunteer
- \$25 – Middle School and Below
 - Event registration
 - May upgrade to top membership for benefits
- Parent account is responsible for annual membership payments
- Users will be able to manage memberships in the Account Page



Emailing Service & Notifications

- Emailing service was updated due to monthly charges accrued
- MailerSend provides a free plan for organizations (12,000 emails/month)
- Event notifications are sent out based on the user's desired frequency
- Application checks every 15 minutes for new notifications
- Admins can send emails to users when needed

```
// @Timeout(5000)
@Cron('*/*15 * * * *')
@Use RequestContext()
public async sendEmailReminders() {
    const events = await this.getEvents();
    const map: Map<Event, { freq: ReminderFreq; users: User[] }> = new Map();

    for (const event of events) {
        // No registrations, no emails.
        if (!event.registrations.length) continue;

        const freq: ReminderFreq = this.eventToFrequency(event);

        // If an event has been notified for this type, abort.
        if (event.notified.includes(freq)) continue;

        for (const registration of event.registrations) {
            const user = registration.user;

            // User doesn't want, or can't receive emails.
            if (!user.reminders || !user.email || !user.emailVerified) continue;

            // User didn't want this type of notification.
            if (!user.reminders.includes(freq)) continue;
        }
    }
}
```

Volunteer Involvement

High Schooler Involvement

- OMC high school volunteers will be able to continue developing the app once SD2 concludes
- Summer curriculum plan divided into 2 phases
 - OMC documentation to introduce students to relevant technologies
 - Week long Internship at the end of July
- Beneficial to future students interested in web/app development

The diagram features a large grey circle on the right, partially overlaid by a light blue circle. A teal diagonal line extends from the top-left towards the bottom-right, intersecting the circles. To the left of the circles, there is a teal wavy icon. Below the circles, the text "HIGH SCHOOL CURRICULUM" is centered. The section is divided into two main parts: "1 - LEARNING" and "2 - APPLYING".

HIGH SCHOOL CURRICULUM

1 - LEARNING

Resources/tutorials/background info on:

- Web / Application Development
- GitHub
- TypeScript
- PostgreSQL
- Node.Js / NestJs
- Vue.Js

PLAN TO FOLLOW THROUGHOUT SUMMER

2 - APPLYING

Give the students specific exercises implemented directly within the app (mostly frontend).

PLAN TO FOLLOW DURING WORKSHOP IN JULY

CURRICULUM WOULD BE WITHIN GITHUB DOCUMENTATION

Documentation Curriculum

- Initial google survey to determine the student's level of familiarity with web/app development
- Brief the students on GitHub, Visual Studio Code, and Markdown
- Pages dedicated to introducing the volunteers to web development and web stacks, gradually introducing them to the advanced technologies

Web Development

The basics of traditional web development.

Background Information

As its name implies, web development is the process of building and maintaining websites. Web development is typically divided up into frontend and backend development. Frontend development makes up the visual appeal of the website, handling the site's design, layout, and interactivity. Backend development handles the behind the scenes of the site where data and the server is stored. This subject will be further explored in a later section.

Beginners to web development typically become familiar with HTML and CSS (frontend languages) before delving into the more complex languages involved in web & mobile applications as simple static websites do not require any work from the backend.

Table of Contents

- Background Information
- HTML
- CSS
- JavaScript
- Websites vs. Web Applications

Documentation

For ease of reference, below are links to all relevant documentation covered in the following sections.

- GitHub
- Visual Studio Code
- Markdown
- HTML
- CSS
- JavaScript
- TypeScript
- Vue.js
- Nuxt.js
- Express.js
- Node.js
- Next.js
- PostgreSQL

OMC Internship

- Conducted a series of workshops with high school students from OMC to introduce the technologies used in the application and web app development fundamentals
- Day 1: Frontend Overview + Exercises
- Day 2: Backend Overview + Exercises
- Day 3: WebApp Hands-On

Day 1
Lectures and exercises completed during day 1 of the workshop.

Lecture
View the presentation given during day 1 of the workshop [here](#).

CSS Flexbox Practice
Flexbox Froggy is an interactive game that helps familiarize players with CSS's Flexible Box Module, known as flexbox, a responsive layout model that allows elements in a container to automatically adjust dependent on screen size. The game challenges players through 24 levels, teaching them how to justify and align content along with ordering and wrapping items.

Table of Contents

- Lecture
- CSS Flexbox Practice
- Color Guess Game
- HTML Structure
- CSS Styling
- Implementing Functionality with JavaScript
- Simple To-Do List
- HTML Structure
- CSS Styling
- Implementing Functionality with JavaScript
- CHALLENGE - Save List



Day 2
Lectures and exercises completed during day 2 of the workshop.

Lecture
View the presentation given during day 2 of the workshop [here](#).

Exercises
JavaScript Practice
The attached [JSFiddle](#) includes the code for the example done during lecture.

Table of Contents

- Lecture
- Exercises
- JavaScript Practice
- Use Case Diagram

Use Case Diagram
Students participated in helping create a use case diagram for a practical real-world example, that being a banking management system. The completed diagram is shown below.



Day 3
Lectures and exercises completed during day 3 of the workshop.

Lecture
View the presentation given during day 3 of the workshop [here](#).

OMC App Exercise: Page Creation

The following exercise can ONLY be completed if you have the application running locally on your machine. However, students are welcome to view the exercises whether or not they have a local environment running.

The goal of this exercise is to create a new page within the app that will appear in the bottom navigation bar. After the basic template for the page is set up, students can customize the page however they would like. Completion of the exercise will resemble the GIF below.



Table of Contents

- Lecture
- OMC App Exercise: Page Creation
- Navigation Bar Modification
- Defining Initial Component
- Adding Cards
- Displaying User Data
- Styling Elements
- Github Practice

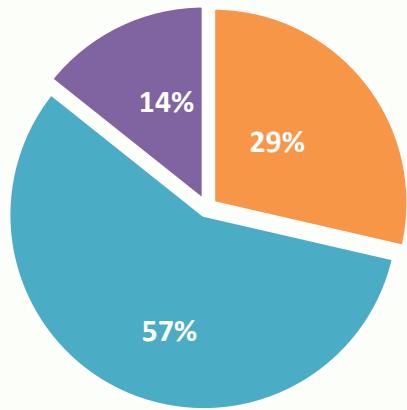
Beta Testing Team

- Weekly tasks focused on specific app functionalities
- Feedback surveys with targeted areas for review
- Additional Issue/bug reporting

The screenshot shows the 'Beta Testing Team - WebApp' interface. At the top, there's a purple header with the title and a 'Class code: aiehmwn' field. Below it are 'Meet link' and 'Generate Meet link' buttons. To the right is a decorative illustration of a laptop, a notepad with handwritten text, and a pen. A sidebar on the left contains sections for 'Upcoming' (no work due soon) and 'Saved announcements'. The main area displays a single announcement from 'Luisa C OMC-App' about a new assignment: 'First Testing Task Survey' posted 'Yesterday'. The sidebar also includes a 'Select theme' and 'Upload photo' option.

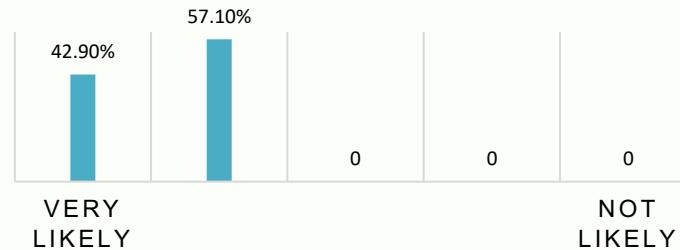
Issue ID	Title	Steps to Reproduce (Alt + Enter for new line)	Expected	Actual	Completed?
1	Admin Panel Dark Mode Text	1. Once logged in, access admin panel 2. Change view to Dark Mode 3. Navigate over the curve and you will notice the text is not inverted in the pop-up box	Texts needs to automatically invert when changing from light mode to dark mode and vice versa	Text is not visible in dark mode	Y

How easy was the application to use?

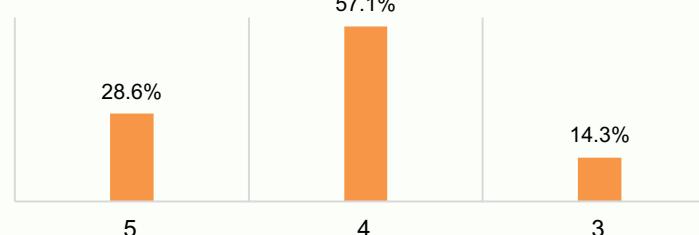


- It felt like I've been using it all my life
- Next time I use it, I'll know exactly what to do
- It will take me a few uses to master the app
- I need someone to show me how to use certain features
- I need someone to show me how to use the app

How likely are you to recommend this app to another OMC volunteer?



How would you rate your experience using the mobile app?



Project Budget

- Current \$3500 Yearly Credit with Microsoft Azure
- No monthly or set up fees associated with PayPal
- Standard transaction fee of 2.9% + \$0.30 for every processed payment through PayPal
- Free organization MailerSend for 12000 emails/month + \$1 per 1000 emails over the limit
- No designated budget for the development of the application
- No external costs to OMC



Microsoft Azure



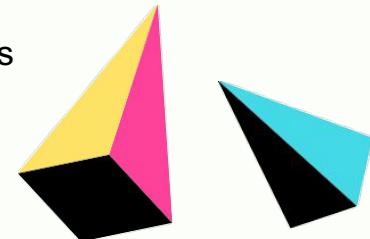
PayPal



mailersend

Project Continuation

- Volunteer dashboard expansion to include a discussion or comment section for events
- Gamification system to allow Admins to create math challenge questions to be answered by users to gain additional points
- Facebook feed and Google Class integration could be implemented to increase functionality and engagement on the Home Page
- Future maintenance and development to be handled by OMC volunteers



Questions?

