Ontario
Parks

Park Mindfulness

Members: Angad, Michael, Michele, Parth, Samantha, Tajwaar, Tony
Partner: John Leadston

Note: Interactive activity during the demo, have your phone ready to follow along

*All icons are taken from https://icons8.com

What is Park Mindfulness?

- In a time where mental health is a very high priority to many, practices such as mindfulness can go a long way in helping people. Forest therapy or forest bathing, as it is known in Japan, is one such practice. The intention of this project is to extend healthcare to our parks by making mindfulness something that can help people deal with things like stress, depression, and PTSD in a positive setting.

- For park visitors, our target users, they can scan a QR code at a bench which will take them to a guided meditation exercise for that specific bench.

- For park managers they can create benches, edit the details of existing benches and delete benches.

Functionality Demo

https://6-john-t-one.vercel.app/

Process and Deployment

- Group collaboration took place on GitHub - https://github.com/csc301-2023-winter/6-John-T

- For our workflow, we divided ourselves into 3 teams, in a 3-2-2 split:
  - Backend (3): responsible for the backend API and database
  - User Frontend (2): responsible for the user side web app experience
  - Manager Frontend (2): responsible for the manager side web app experience

- Each team works on their own feature branch, and upon successful user testing, merges to main branch after approval on their Pull Request from at least 1 other member.

- The deployments of all 3 sub-projects are automated. Once the commits are pushed to the main branch, the apps gets updated within 30 minutes.

- Interesting deployment related issues we had are addressed in the class blog [1]

[1] Class blog site: https://learning-software-engineering.github.io/Topics/Development_Process/Frontend_Automated_Deployment_Vercel.html

Process and Deployment

	Frontend (User)	Frontend (Manager)	Backend
Hosting Platform	Vercel	Vercel	Railway
Automated Deployment (CI/CD)?	✔	✔	✔
Automated Testing Support?	✔	✔	✔
Security Measures	- HTTPS - SSL Certificate	- HTTPS - SSL Certificate	- HTTPS - SSL Certificate
Cost	Free	Free	Free with limits

Railway vs. AWS vs. Vercel:
https://github.com/learning-software-engineering/learning-software-engineering.github.io/blob/main/Topics/Development_Process/Django_Deployment_AWS_Railway_Vercel.md

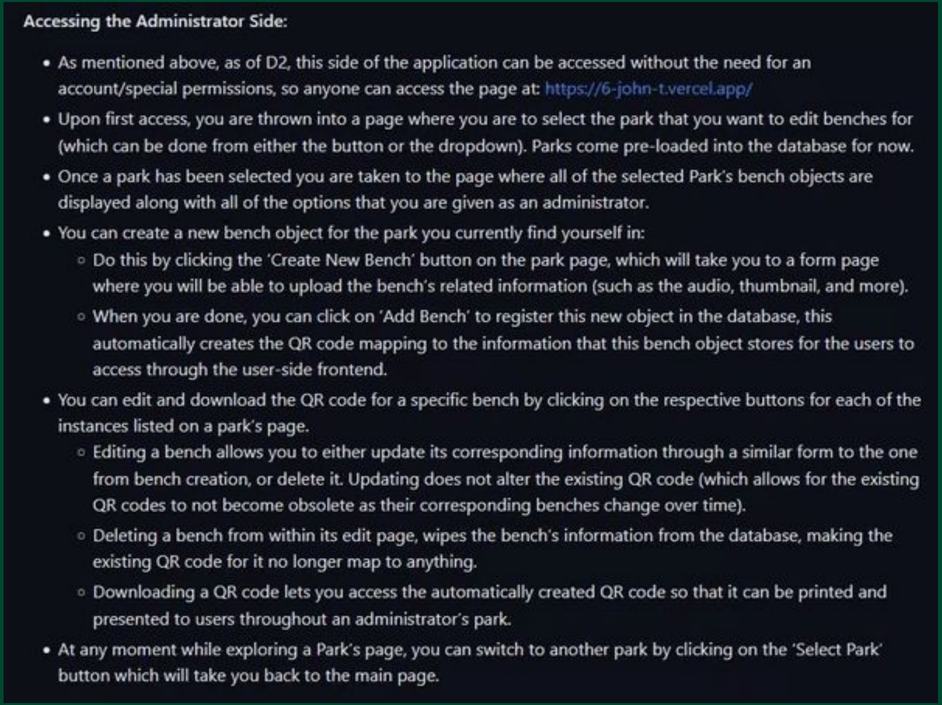
Accessing the Application

- Documented instructions for accessing the application can be found on GitHub: https://github.com/csc301-2023-winter/6-John-T#instructions

- Deployment will remain through Vercel and Railway with the flexibility and possibility for change

- An Ontario Parks Technical Team will be involved to ensure a seamless handoff

- We also plan to provide a short video demonstration of the finished product



Accessing the Administrator Side:

- As mentioned above, as of D2, this side of the application can be accessed without the need for an account/special permissions, so anyone can access the page at: https://6-john-t.vercel.app/
- Upon first access, you are thrown into a page where you are to select the park that you want to edit benches for (which can be done from either the button or the dropdown). Parks come pre-loaded into the database for now.
- Once a park has been selected you are taken to the page where all of the selected Park's bench objects are displayed along with all of the options that you are given as an administrator.
- You can create a new bench object for the park you currently find yourself in:
  - Do this by clicking the 'Create New Bench' button on the park page, which will take you to a form page where you will be able to upload the bench's related information (such as the audio, thumbnail, and more).
  - When you are done, you can click on 'Add Bench' to register this new object in the database, this automatically creates the QR code mapping to the information that this bench object stores for the users to access through the user-side frontend.
- You can edit and download the QR code for a specific bench by clicking on the respective buttons for each of the instances listed on a park's page.
  - Editing a bench allows you to either update its corresponding information through a similar form to the one from bench creation, or delete it. Updating does not alter the existing QR code (which allows for the existing QR codes to not become obsolete as their corresponding benches change over time).
  - Deleting a bench from within its edit page, wipes the bench's information from the database, making the existing QR code for it no longer map to anything.
  - Downloading a QR code lets you access the automatically created QR code so that it can be printed and presented to users throughout an administrator's park.
- At any moment while exploring a Park's page, you can switch to another park by clicking on the 'Select Park' button which will take you back to the main page.

https://github.com/csc301-2023-winter/6-John-T#instructions

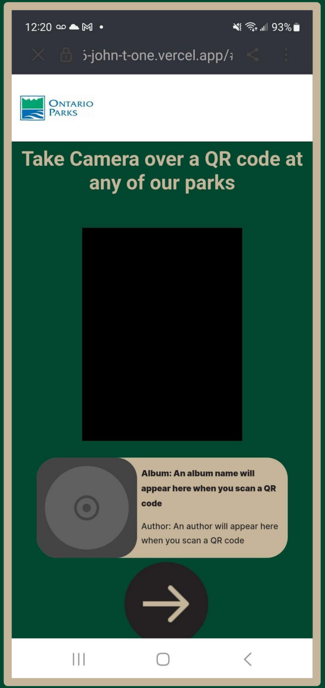
Tech Stack and Design

Frontend

React

- Simple, easy to learn flexible and has many online resources

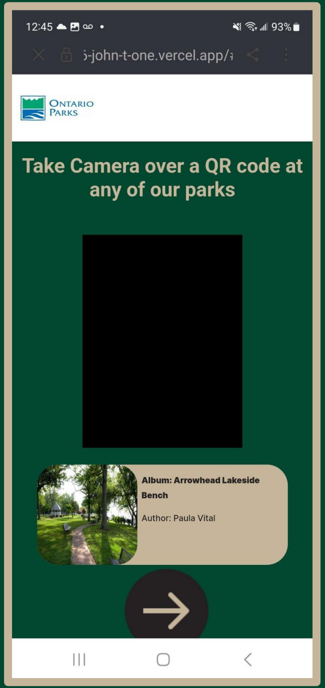
Design:



Take Camera over a QR code at any of our parks

Album: An album name will appear here when you scan a QR code

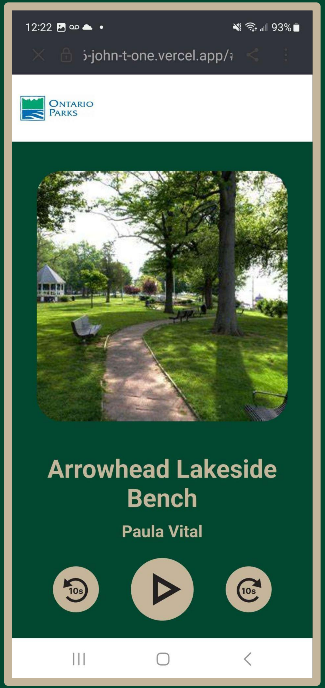
Author: An author will appear here when you scan a QR code



Take Camera over a QR code at any of our parks

Album: Arrowhead Lakeside Bench

Author: Paula Vital



Arrowhead Lakeside Bench

Paula Vital

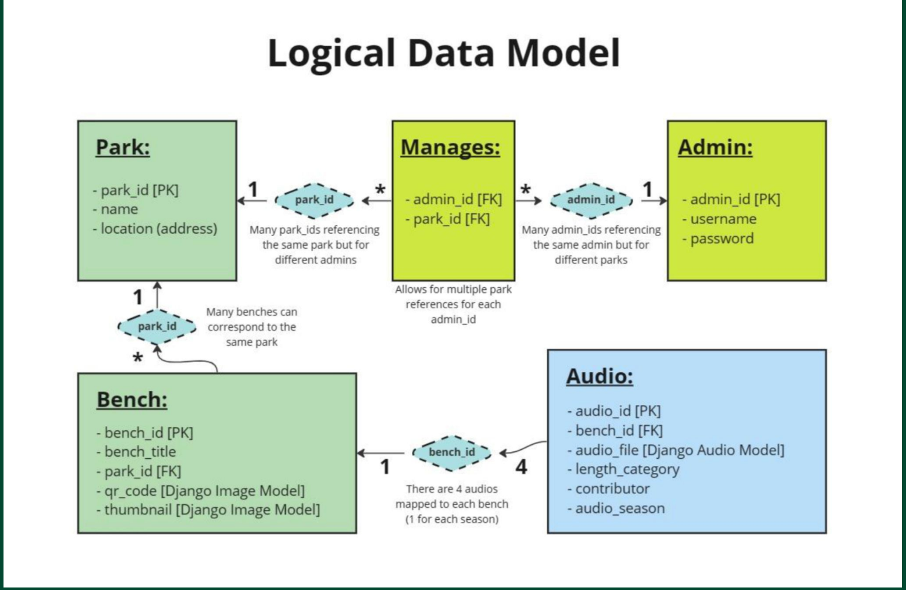
Tech Stack and Architecture

Backend

- Powerful and flexible web framework
- Previous experience

- Lightweight, file-based database system
- Built into Python's standard library

Logical Data Model (LDM):



Logical Data Model

Park:
- park_id [PK]
- name
- location (address)

Manages:
- admin_id [FK]
- park_id [FK]

Admin:
- admin_id [PK]
- username
- password

Bench:
- bench_id [PK]
- bench_title
- park_id [FK]
- qr_code [Django Image Model]
- thumbnail [Django Image Model]

Audio:
- audio_id [PK]
- bench_id [FK]
- audio_file [Django Audio Model]
- length_category
- contributor
- audio_season

Many park_ids referencing the same park but for different admins

Many admin_ids referencing the same admin but for different parks

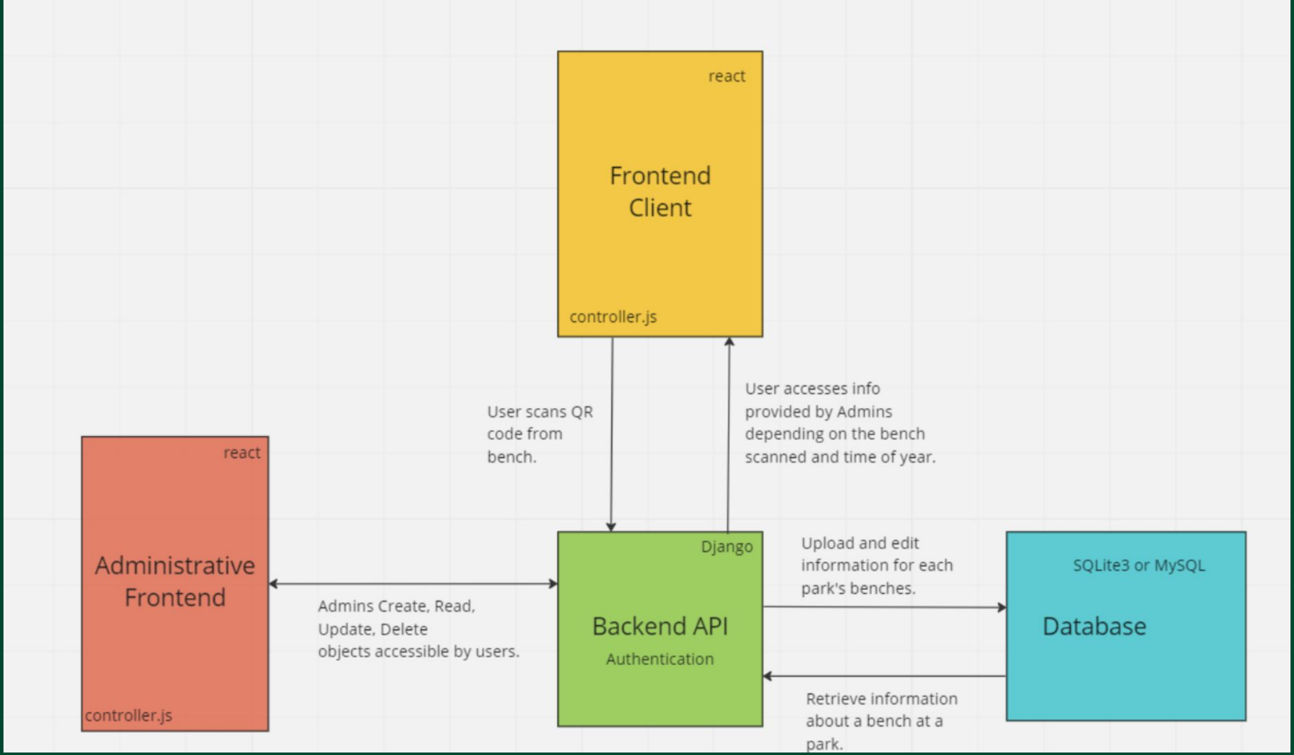
Allows for multiple park references for each admin_id

Many benches can correspond to the same park

There are 4 audios mapped to each bench (1 for each season)

Tech Stack and Architecture

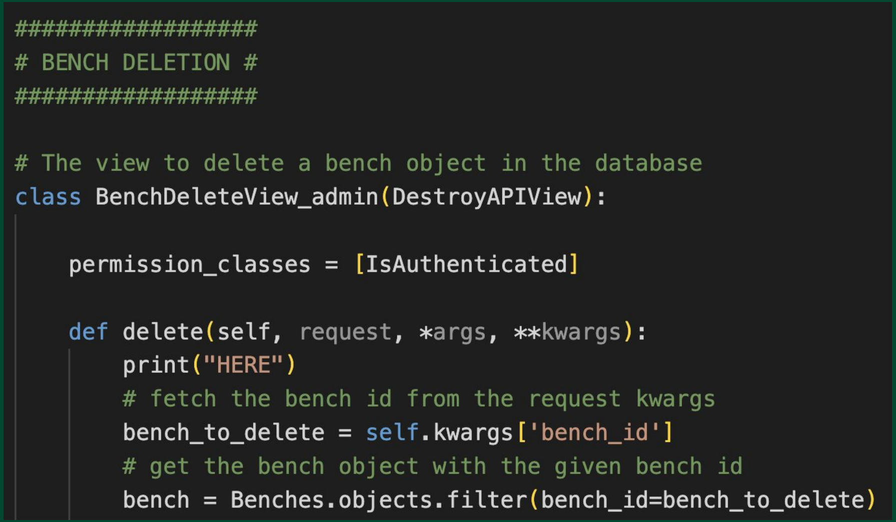
Backend to Frontend Connection


react
Frontend Client
controller.js
User scans QR code from bench.
User accesses info provided by Admins depending on the bench scanned and time of year.
react
Administrative Frontend
controller.js
Admins Create, Read, Update, Delete objects accessible by users.
Django
Backend API
Authentication
Upload and edit information for each park's benches.
SQLite3 or MySQL
Database
Retrieve information about a bench at a park.

Code Structure and Practices

- Clean coding practices
  - Follow correct naming conventions (snake_case)
  - Make necessary comments
  - Avoided unnecessarily long functions

- Proper documentation
  - Clean and understandable ReadMe



```
##################
# BENCH DELETION #
##################

# The view to delete a bench object in the database
class BenchDeleteView_admin(DestroyAPIView):

    permission_classes = [IsAuthenticated]

    def delete(self, request, *args, **kwargs):
        print("HERE")
        # fetch the bench id from the request kwargs
        bench_to_delete = self.kwargs['bench_id']
        # get the bench object with the given bench id
        bench = Benches.objects.filter(bench_id=bench_to_delete)
```

Reflections and Takeaways

Reflection:

- The development process
  - Finding errors you aren’t familiar with can be agonizing
  - Had issues with various errors that came mostly from team being unfamiliar with tech stack, causing unnecessary stress
  - Had good communication and a clear understanding on what was expected from each team member/sub-team

Takeaways:

- Use a tech stack you are familiar with whenever possible
- Give more people access to the deployment account
- Organizing work via Notion and other means helped improve workflow and making deadlines

Thank You

Any Questions?

Contributions

- Angad: manager frontend - bench creation page, park creation page, park edit page.
- Michael: manager frontend - authentication, user creation and update, updating previous views, backend deployment
- Michele: backend - bench entity CRUD, user authentication and creation functionality, user model
- Parth: user frontend - media page & frontend automated deployment
- Samantha: backend - park entity CRUD, bench creation and models
- Tajwaar: Backend - Backend automated deployment & testing; Park app creation; Park & Bench read views
- Tony: user frontend - QR scanner page, manager frontend - connecting frontend to backend, various misc.