COSC 4P02 Software Engineering

Final Report

ARchaeology Web App

For

Professor Naser Ezzati-Jivan

By

Abe Shakeel (Software Developer)

Robert Worron (Software Developer)

Nickolas Kenyeres (Software Developer/Tech Lead)

David Boere (Software Developer)

Mohit Gummaraj Kishore (Product Owner/QA Engineer)

Table of Contents

Product Information ***3***

Technology Stack **3**

Front End 3

Backend 7

Agile Process ***8***

Agile Ceremonies **8**

Weekly Standup 8

Sprint planning 8

Backlog Refinement 8

Sprint Retrospective 9

Sprint Information **9**

User Stories **10**

Meeting Minutes **11**

January 24, 2023 11

February 2, 2023 12

February 16, 2023 13

March 2, 2023 14

March 16, 2023 15

March 23, 2023 16

March 30, 2023 17

April 6, 2023 18

GitHub Logs ***19***

Helpful Links ***20***

Business Information **20**

Product Access **20**

Technology Information **20**

Product Management **20**

Appendix ***21***

Product Information

Technology Stack

ARchaeology is a web application built with a Jamstack architecture, which separates the business logic and data layer from the web experience layer. See jamstack.org for more information on this architectural approach.

In the case of ARchaeology, the web app was implemented in the Remix framework using the React UI library and the Tailwind CSS framework.

ARchaeology was hosted on CloudFlare Pages, using edge functions to server-render HTML pages. The Edge functions execute in service workers in the V8 engine and implement many of the same APIs as web browsers. After rendering, the Remix Framework handles rehydrating the React app, making it fully interactive. Cloudflare Pages was also responsible for seeing that static asset, such as scripts, stylesheets, and images were hosted on its content delivery network (CDN). This was the separated web experience part of Jamstack.

As per the Jamstack architecture, the business logic and data were hosted on Supabase, which is an application development service. It has a number of features; however, ARchaeology only made use of its relational PostgreSQL database for data storage and authorization, its authentication service for handling sign in and sign out, and its simple storage service for handing file uploads and downloads.

Front End

Remix

Remix is an open-source frontend meta framework (i.e. it builds upon React) that supports creating hybrid multi-page applications (MPAs) and single-page applications (SPAs). It does this by allowing you to develop your application in React, where on the initial server render it builds up the component tree and renders it out as a string of HTML to be delivered to the client, where after the JavaScript downloads on the client, the React application is re-rendered, this time completely interactive within the browser. At this point, the application may choose to perform partial page apps and client-side routing like a single-page application, or it may make full page refreshes like a traditional multi-page application. Part of the reason for choosing this technology was that it was inexpensive to host on cloud computing platform under generous free tiers. Another reason for the choosing this technology was that the server-side rendering had an advantage in search engine optimization ensuring that the content could be parsed by crawlers even if JavaScript was disabled. The client-side rendering on the other hand provided a more modern look and feel that today's web users have been accustomed to, such as being able to navigate between pages without full page loads.

Tailwind

Tailwind is an open-source utility-based CSS framework where you style your website, web application, and/or components, by composing small utility classes in your HTML. For example, the class name "bg-gray-300" sets the background color of the element to gray.300 from the colour palette.

Tailwind is highly configurable, which would allow us to theme the web application based on the NOTL Museum brand; however, it provides strong defaults and a rigid system that helps to maintain a unified look and feel from component to component.

Storybook

Storybook is an open-source interactive preview environment for UI components. It is UI framework agnostic and can be used with React, Vue, Svelte, etc.

Storybook was used to support our component-first development approach. We developed components using stories in Storybook to iterate on the components and share our work with each other.

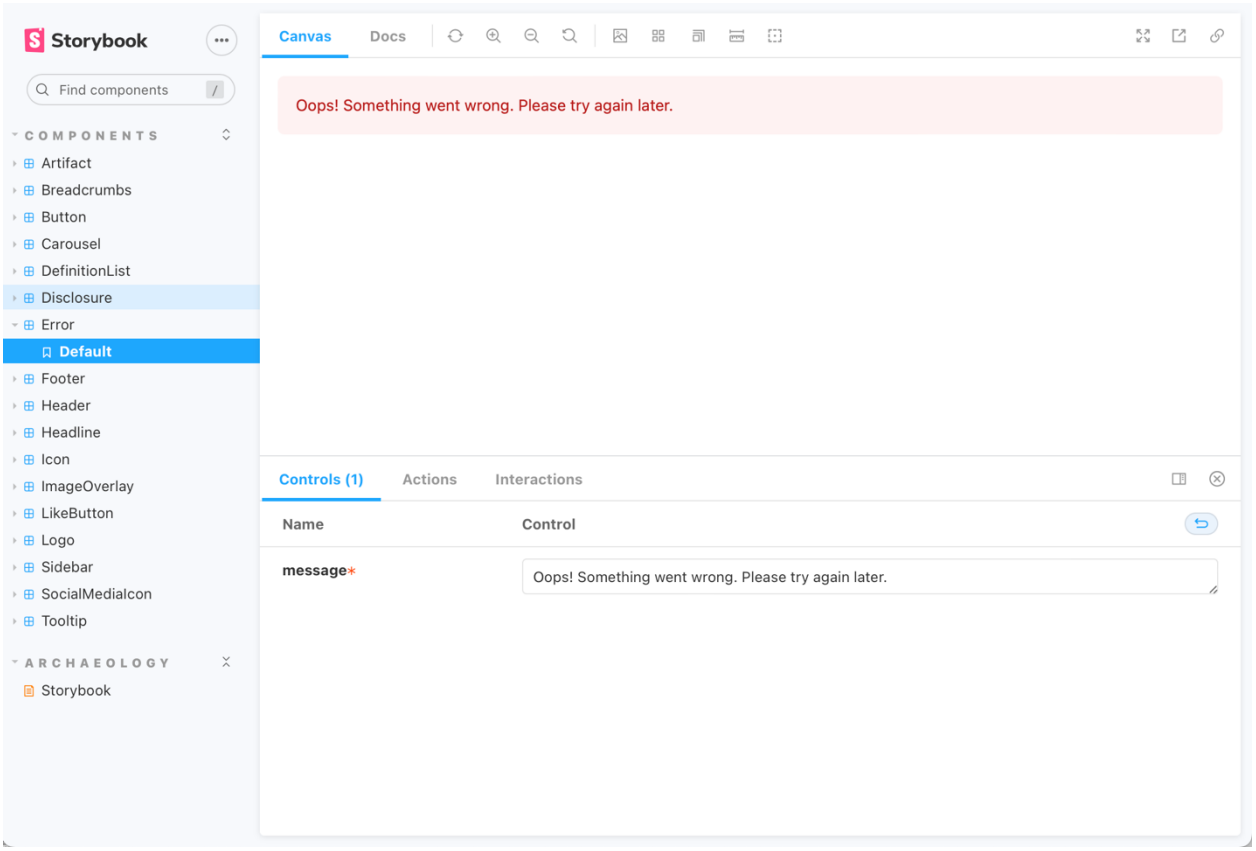


Figure 1 - An example of the Error component rendered in Storybook

Netlify

Netlify is a cloud computing development platform that offers a content delivery network (CDN) for hosting static content, edge functions for server-side rendering, serverless functions for backend logic, and more.

Netlify was utilized for hosting the ARchaeology Storybook and for generating deploy previews of the Storybook on every pull request. This allowed the team to review components as part of the PR review process without having to download, install and run the project locally.

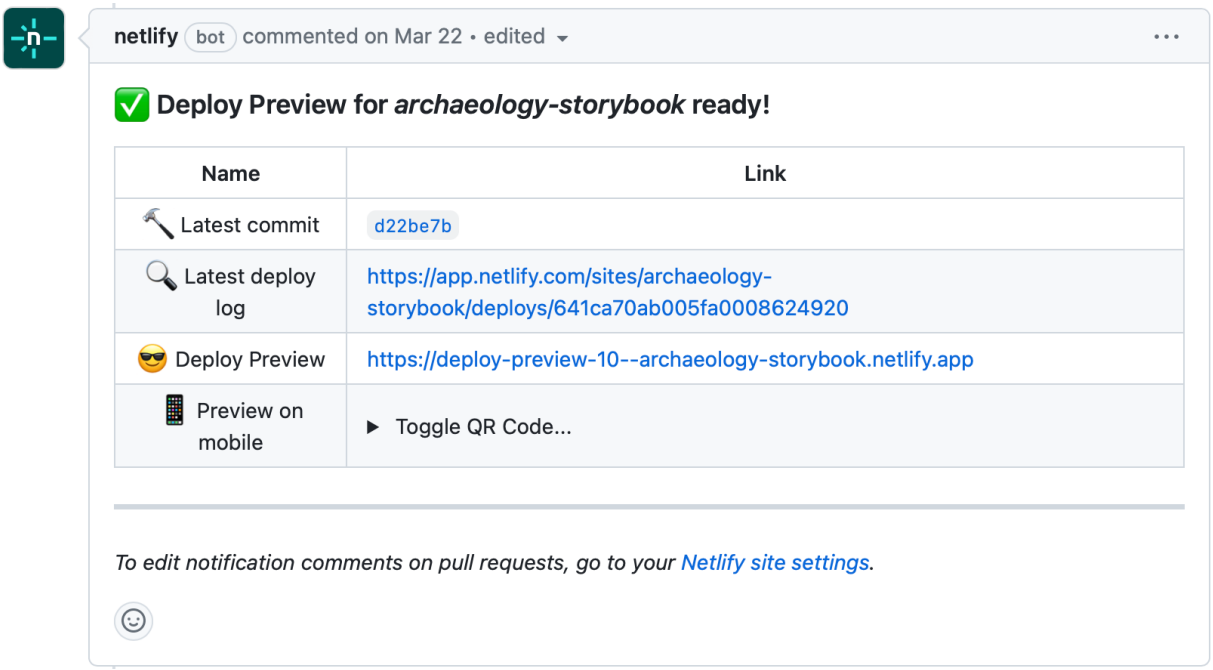


Figure 2 - Example of Netlify Bot posting an automatically created Deploy Preview of Storybook on an open PR in GitHub.

A main Storybook instance was also published on Netlify at https://archaeology-storybook.netlify.app/, which was configured to be automatically rebuilt whenever the main branch was updated. The main branch was typically updated by merging another branch into via a pull request.

It should be noted that Cloudflare offered a similar deploy preview feature; however we were unable to utilize it for the Storybook since Cloudflare only allows for one build command per repository and we already used it for running primary and deploy preview instances of the main ARchaeology web app.

Cloudflare

Cloudflare is a cloud computing development platform, like Netlify, but broader in its service operating. In addition to offering a CDN, edge functions, and serverless functions, Cloudflare also offers cloud cybersecurity services, distributed denial of service (DDoS) attack mitigation, domain registration, and more.

Cloudflare was utilized for hosting the ARchaeology web app. It was also configured to generate deploy previews of the web app on every pull request. This allowed the team to review changes to the web app as part of the pull request (PR) review process without having to download, install and run the project locally.

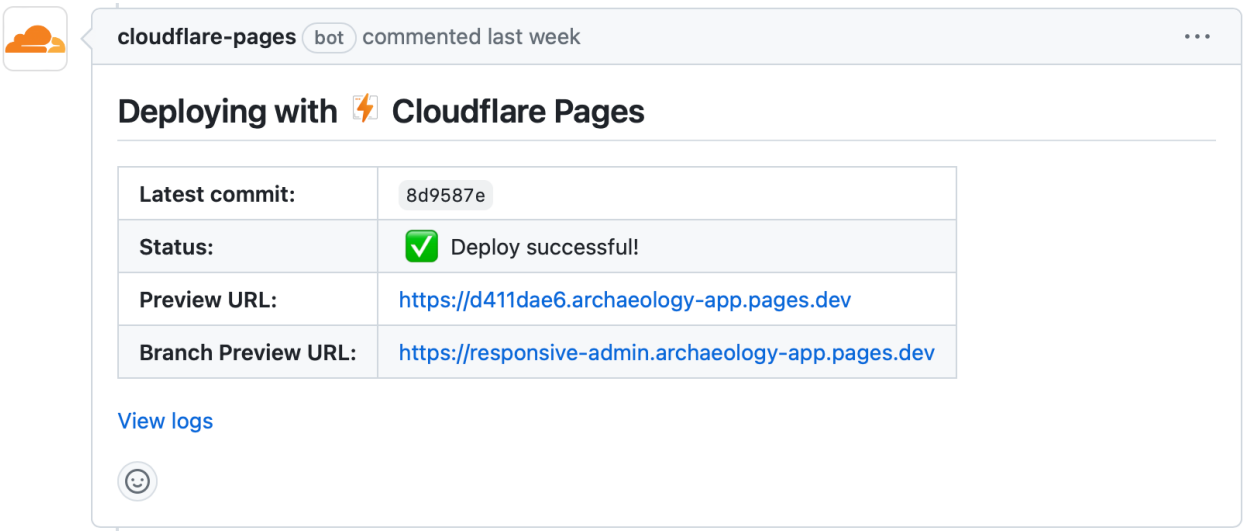


Figure 3 - Example of Cloudflare Pages Bot posting an automatically created Deploy Preview of the web app on an open PR in GitHub

The main web app instance was published on Cloudflare at https://archaeology-app.pages.dev/. It was configured to automatically rebuild whenever code was updated on the main branch, which was typically via merging a branch from a PR.

Midjourney

Midjourney is a generative artificial intelligence that takes text prompts as input and generates artwork as output. Since we did not have a graphic designer on our team, we used this AI to fill that role. We used it to create the logo below, which we modified slightly in Affinity Designer and used as the logo for the admin panel and as the favicon, touch icons, and splash screens.

Figure 4 - The unaltered version of the logo generated by Midjourney with the prompt: "A cute logo for an archaeologist wearing virtual reality glasses VR goggles. Rendering. No letters. No text. Ultra detailed."

Backend

Supabase

Supabase is an application development platform. It is a hosted platform; however, it is also open-source and can be downloaded for self-hosting. It includes a PostgreSQL database for data storage and authorization, an authentication system, instant REST, GraphQL and RPC APIs for the database, edge functions, real-time subscriptions and storage.

We used Supabase for the PostgreSQL data storage, authentication system, and storage. We used its JavaScript client, primarily interacting with it via the JavaScript client which used REST to communicate with the Supabase server. However, we also used the JavaScript client with RPC to trigger a stored procedure in the database which was used to “like” artifacts.

Agile Process

Agile Ceremonies

As an agile team, we focused on our process of building agile practices. We regularly conducted standups, sprint planning session, scoring our cards during backlog refinement and finally ended each sprint with a retrospective. Our goal was to run 7 sprints, but we had to adapt to the timing and condense the development into three sprints. The main reason for our delay was knowledge transfer and identifying the features of the product.

Weekly Standup

During our weekly standup, we provided everyone to talk about three main topics:

- What did they work on that week?
- What are they currently working on this week?
- Do they have any blockers?

If any team member had blockers, our tech lead ensured to provide them with any advice or links to unblock them. The weekly standup was also used to conduct peer reviews and merge any outstanding pull requests. Throughout the course of product development, we closed 23 pull requests.

Sprint planning

The sprint planning session was held every month. The goal of the sprint planning session was to understand how many points were carried over from the previous sprint and based on the team's velocity, we identified new sprint goal. We conducted three sprint planning sessions where we would discuss the business requirements, development tasks required, and the QA tasks required for each task.

The team was highly engaged in understanding not just the development, but also the business asks. This helped us build a better product that Niagara on the Lake can use.

Backlog Refinement

We used the various scrum principles while building ARCheology. During our backlog refinement sessions, we used two main tools Scrum Online Poker and our backlog board. While scoring the cards, we identified the effort in terms of time, learning and complexity of the task. We used the finocchi series to score our cards. Here is a brief outline of what the team considered before scoring the tasks

SCORE	TIME	LEARNING	COMPLEXITY
1	>24 hours	n/a	Easy
2	1 to 2 days	n/a	Easy
3	2 to 3 days	New topic, experienced	Medium
5	1 week	New topic, some experience	Medium
8	1 sprint	New topic, no experience	High

Sprint Retrospective

During the building of Archeology, we had the opportunity to have two sprint retrospectives. The topics of discussion during our sprint retrospective were:

- What's working on the team?
- What to improve?
- What's not working?
- Kudos & Thanks you

We used a Miro board to run our sprint retrospectives and the team provided many important feedback on how the sprints.

Sprint Information

Sprint Number	Sprint Dates	Velocity	Points Accomplished	Points Carried Over	List of Tasks Planned
0	Jan 24 – Jan 31	9	9	0	• Create design for menu items on the app • Create universal design standard for Archaeology • Create design for email capture screen • Create design for QR code scanner screen • Create design for splash screen
1	Feb 1 – Feb 13	3	3	0	• Create architecture for our application • Create user flow diagrams using flow charts
2	Feb 14 – Feb 27	7	5	2	• Create ARchaeology Logo • Setup Environment for Supabase Backend • Setup Environment for React Web Application
3	Feb 28 – Mar 13	14	14	0	• Deploy an AR Object to test if ARCore and ARView works as expected • Create a production pipeline using GitHub • Create a component that has a headline, subheading and a button • Create an icon wrapper component with size & colour as props
4	Mar 14 – Mar 27	13	13	0	• Create a component to show information about the artifact

					• Build a React component with Image and text overlay • Create a react header component with the current NOTL Header
5	Mar 28 – Apr 10	14	14	0	• Create a page with a list of all the available AR objects in the museum • Integrate the like button on Artifact Info component to Supabase • Create a placeholder page for an artifact
6	Apr 11 – Apr 24	5	5	0	• Create a button that links to NOTL donation in the footer
7	Apr 25 – May 1	-	-	-	•

User Stories

For our product we identified two main actors

- NOTL Museum Visitor
- NOTL Museum Admin

Actor	User Story
NOTL Museum Visitor	As a user, I want to see NOTL Museum branding on all pages so that I can navigate to other pages when I desire.
	As a user, I want to be able to read about all the artifacts at Niagara on the Lake Museum so that I can get more information about the artifact.
	As a user, I should be able to see a list of artifacts available at the NOTL Museum so that I can get more information about the artifacts.
	As a user, I want to see NOTL Museum branding on all pages so that I can navigate to other pages when I desire.
	As a user I want to see a list of all NOTL artifacts so that I can interact with objects that I feel are interesting.
	As a user, I want to see the number of likes and views of an artifact so that I can see the most popular artifact in Niagara on the Lake Museum.
	As a user I want to see all the information about an artifact and view the object in AR so that I can learn more about the NOTL museum artifacts.

	As a user, I want to be able to donate to Niagara on the Lake Museum to immerse myself in the museum's offerings.
NOTL Museum Admin	As a user, I was to be able to upload information about an artifact so that NOTL Visitors can view information online.
	As a user, I was to be able to upload an AR Object of an artifact so that NOTL Visitors can view the artifact in AR.
	As a user, I want to be able to manage admin users so that NOTL Admins can add more information about the various artifact.
	As a user, I want to be able to generate QR codes for artifacts so that NOTL Visitors can visit the website to view AR Object.

Meeting Minutes

January 24, 2023

Attendees

- Robert Worron
- Nickolas Kenyeres
- David Boere
- Mohit Gummaraj Kishore

Agenda

- Product features decision
- Tech stack decsions

Minutes:

Landing page –

- Notification feed NOTL museum (something like an RSS feed?)
- Camera button
- Button to go to list of artifacts.
- Email input page
- Camera page
- After scanning, popup asking for AR or text based.
- If AR, camera stays open.
- Text-based, display blank page with text and pictures.
- Page with list of all the artifacts/exhibits
- Clicking on the item will expand it downwards, showing two buttons for AR or text-based education.

Stretch goals –

- Build continuity into app.
- Allow NOTL to easily add/remove artifacts to keep the app up to date.

February 2, 2023

Attendees

- Robert Worron
- Nickolas Kenyeres
- David Boere
- Mohit Gummaraj Kishore

Agenda

-

Minutes:

- **No meeting**

February 16, 2023

Attendees

- Abe Shakeel
- Robert Worron
- Nickolas Kenyeres
- David Boere
- Mohit Gummaraj Kishore

Agenda

- Introduction to Gurkirat
- Showcase of AR Object Created
- Backlog Refinement
- Sprint Planning
- Sprint kickoff

Minutes:

- Introduction to Gurkirat:
    - Gurkirat to help us with QR Scanner
- Showing AR object
- Showcase of AR Object Created
- Showed a video of the AR object created.
- Backlog refinement
    - We scored 5 cards and assigned them to all team members.
- Sprint Planning
    - Started sprint 2 with 5 tasks.
- Velocity is 12 points.

March 2, 2023

Attendees

- Robert Worron
- Nickolas Kenyeres
- David Boere
- Mohit Gummaraj Kishore

Agenda

- Sprint Retrospective
- Sprint Review
- Backlog Refinement
- Sprint planning & kickoff

Minutes:

- Sprint Review
  - Points Assigned : 15
  - Points completed: 3
- Sprint Planning:
  - Started sprint 3 with 17
- Sprint Retrospective
  - URL: https://miro.com/app/board/uXjVPh_nW_g=/?moveToWidget=3458764547761376333&cot=14

March 16, 2023

Attendees

- Robert Worron
- Nickolas Kenyeres
- David Boere
- Mohit Gummaraj Kishore

Agenda

-

Minutes:

- **No meeting**

March 23, 2023

Attendees

- Robert Worron
- Nickolas Kenyeres
- David Boere
- Mohit Gummaraj Kishore

Agenda

- Stand up
- Code review

Meeting Minutes

- Standup
    - Team is doing well with the progress of the application.
    - We have finished building the following components:
        - Create a component to show information about the artifact.
        - Build a React component with Image and text overlay.
        - Create a react component with the current NOTL footer.
- Code Review
    - Pull Request 9
    - Pull Request 10

March 30, 2023

Attendees

- Robert Worron
- Nickolas Kenyeres
- David Boere
- Abe Shakeel

Agenda

- Stand up
- Code Review

Meeting Minutes

- Abe said he completed the footer, and sent it to you? (not github, specifically to you)
- Nickolas made his own footer and merged it to main this week. he also said that if Abe's footer shows up before the final presentation, we can just switch out Nickolas' footer for Abe's.
- Rob is working on the text overlay on images; asked Nickolas for a reference point for the testing of that.
- David is working on the header
- Nickolas didn't get started on the like button yet, as the database hasn't been set up
- Rob is going to get to this over the weekend / next week

April 6, 2023

Attendees

- Robert Worron
- Nickolas Kenyeres
- David Boere
- Mohit Gummaraj Kishore

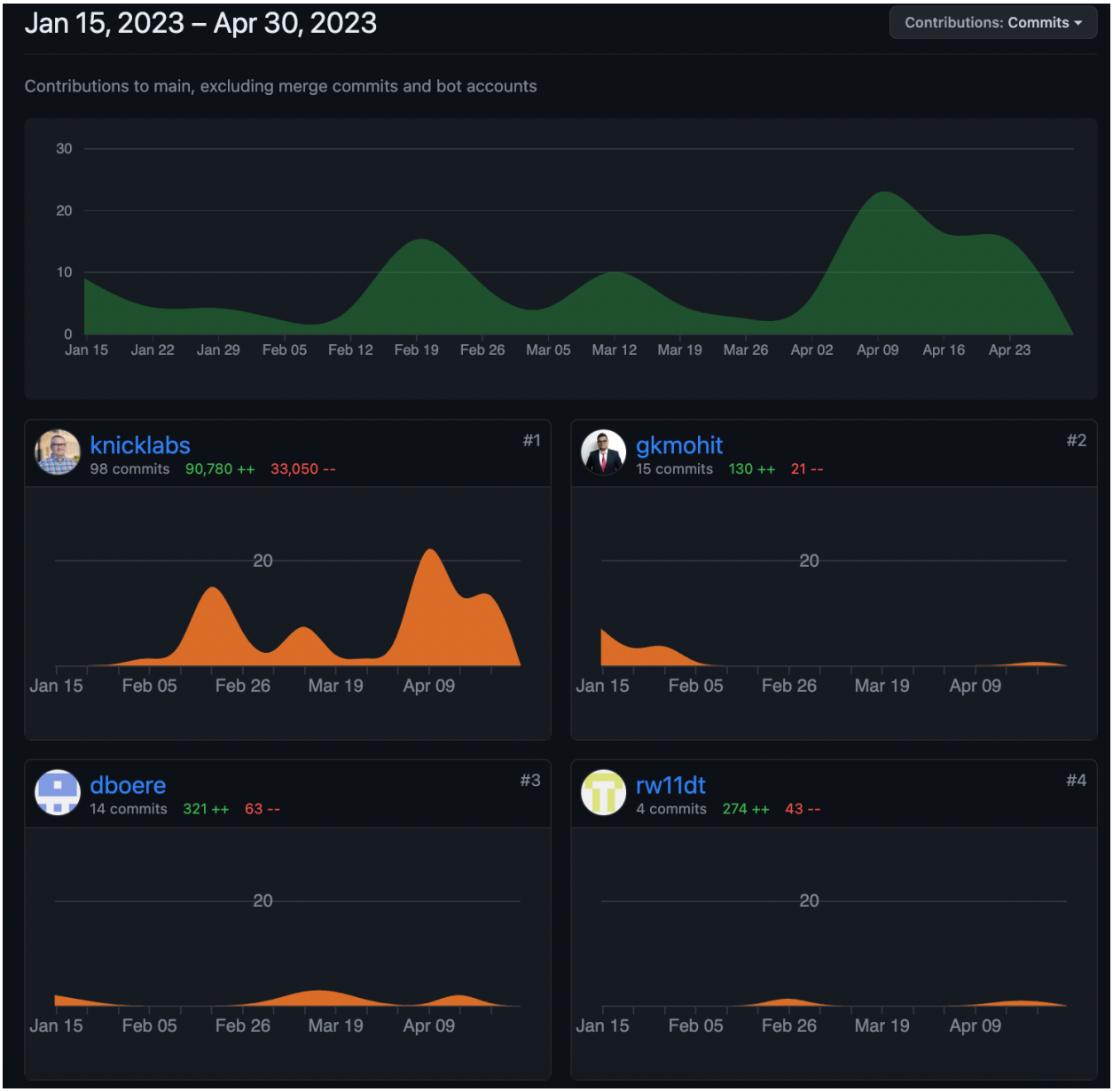
Agenda

- Standup
- Code Review
- Sprint Retrospective

Meeting Minutes

- Standup
  - Team went over the list of tasks done
  - Team agreed to join a War room on April 22 to finish up any unfinished tasks
- Code Review
  - Provided feedback on Pull request: 13
- Sprint Retrospective
  - URL: https://miro.com/app/board/uXjVPh_nW_g=/?moveToWidget=3458764549083690800&cot=14

GitHub Logs


Jan 15, 2023 – Apr 30, 2023
Contributions: Commits
Contributions to main, excluding merge commits and bot accounts
30
20
10
0
Jan 15 Jan 22 Jan 29 Feb 05 Feb 12 Feb 19 Feb 26 Mar 05 Mar 12 Mar 19 Mar 26 Apr 02 Apr 09 Apr 16 Apr 23
knicklabs
98 commits 90,780 ++ 33,050 --
#1
20
Jan 15 Feb 05 Feb 26 Mar 19 Apr 09
gkmohit
15 commits 130 ++ 21 --
#2
20
Jan 15 Feb 05 Feb 26 Mar 19 Apr 09
dboere
14 commits 321 ++ 63 --
#3
20
Jan 15 Feb 05 Feb 26 Mar 19 Apr 09
rw11dt
4 commits 274 ++ 43 --
#4
20
Jan 15 Feb 05 Feb 26 Mar 19 Apr 09

Helpful Links

Business Information

Niagara-on-the-Lake Museum - https://www.notlmuseum.ca/

Product Access

GitHub Project - https://github.com/COSC-4P02-Brock/COSC4P02-ARchaeology

Story book: https://archaeology-storybook.netlify.app

Production App: https://archaeology-app.pages.dev/

Technology Information

Jamstack - https://jamstack.org/

CloudFlare- https://CloudFlare.com/

Supabase - https://supabase.com/

Netlify - https://www.netlify.com/

Remix - https://remix.run/

Tailwind - https://tailwindcss.com/

Storybook - https://storybook.js.org/

Midjourney - https://www.midjourney.com/

Product Management

Scrum Online Poker - https://www.scrumpoker-online.org/

Miro Board - https://miro.com/app/board/uXjVPh_nW_g=/?share_link_id=780907268133

Product Backlog.- https://tasks.office.com/brocku.onmicrosoft.com/en-US/Home/Planner/#/plantaskboard?groupId=1781306b-e14a-4f04-8943-1b76f53d8cf0&planId=93-I37y8w0W_QMFTfxsMe2QABqhA

Appendix

Supabase Relational Database Schema

```
create table
  public.artifact_ar_images (
    id bigint generated by default as identity not null,
    created_at timestamp with time zone null default now(),
    updated_at timestamp with time zone null,
    url text not null default ''::text,
    artifact_id bigint not null,
    constraint artifact_ar_images_pkey primary key (id),
    constraint artifact_ar_images_artifact_id_key unique (artifact_id),
    constraint artifact_ar_images_artifact_id_fkey foreign key (artifact_id)
references artifacts (id) on delete cascade
  ) tablespace pg_default;

create table
  public.artifact_images (
    id bigint generated by default as identity not null,
    created_at timestamp with time zone null default now(),
    updated_at timestamp with time zone null,
    caption text not null default ''::text,
    url text not null default ''::text,
    artifact_id bigint not null,
    constraint artifact_images_pkey primary key (id),
    constraint artifact_images_artifact_id_fkey foreign key (artifact_id)
references artifacts (id) on delete cascade
  ) tablespace pg_default;

create table
  public.artifacts (
    id bigint generated by default as identity not null,
    created_at timestamp with time zone not null default now(),
    updated_at timestamp with time zone null,
    name text not null default ''::text,
    description text not null default ''::text,
    dimensions text not null default ''::text,
    object_id text not null default ''::text,
    date text not null default ''::text,
    constraint artifacts_pkey primary key (id)
  ) tablespace pg_default;
```

```
create trigger handle_updated_at before
update on artifacts for each row
execute function moddatetime ('updated_at');

create table
  public.likes (
    id bigint generated by default as identity not null,
    artifact_id bigint not null,
    created_at timestamp with time zone not null default now(),
    count bigint not null default '0'::bigint,
    updated_at timestamp with time zone null,
    constraint likes_pkey primary key (id),
    constraint likes_artifact_id_key unique (artifact_id),
    constraint likes_artifact_id_fkey foreign key (artifact_id) references
artifacts (id) on delete cascade
  ) tablespace pg_default;

create trigger handle_updated_at before
update on likes for each row
execute function moddatetime ('updated_at');

BEGIN
  EXECUTE format('INSERT INTO likes (artifact_id, count)
  VALUES ($1, 1) ON CONFLICT (artifact_id) DO update
  SET count = likes.count + 1 WHERE likes.artifact_id = $1')
  USING artifact_id_to_check;
END
```

The *public.artifact_images* table had a *url* field that referenced the public *url* of an image stored in the artifacts storage bucket in the Supabase simple file storage.

The *public.artifact_ar_images* table had a *url* field that referenced the public *url* of an image stored in the *ar_artifacts* storage bucket in the Supabase simple file storage.

Below is the entity relationship diagram of the database.

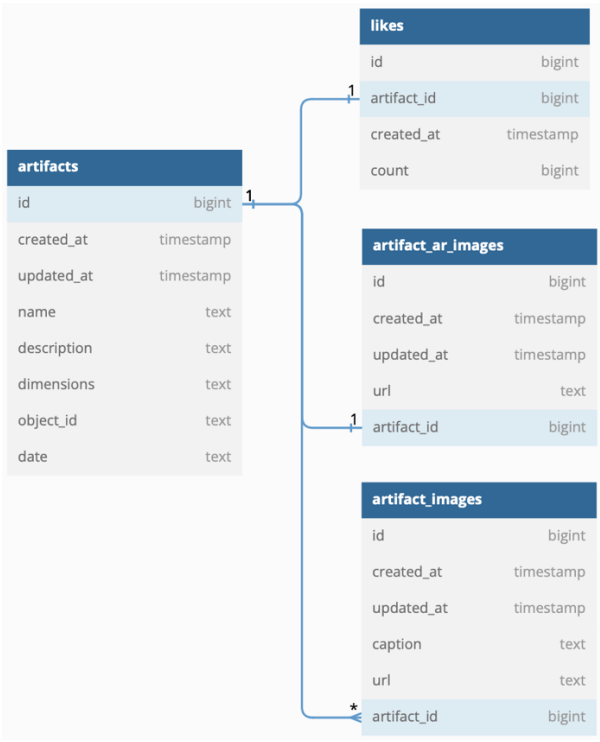


Figure 5 - Entity relationship diagram of database

We implemented row-level security on the database. The following figure shows the policies that were setup for authorization on each row of each table.

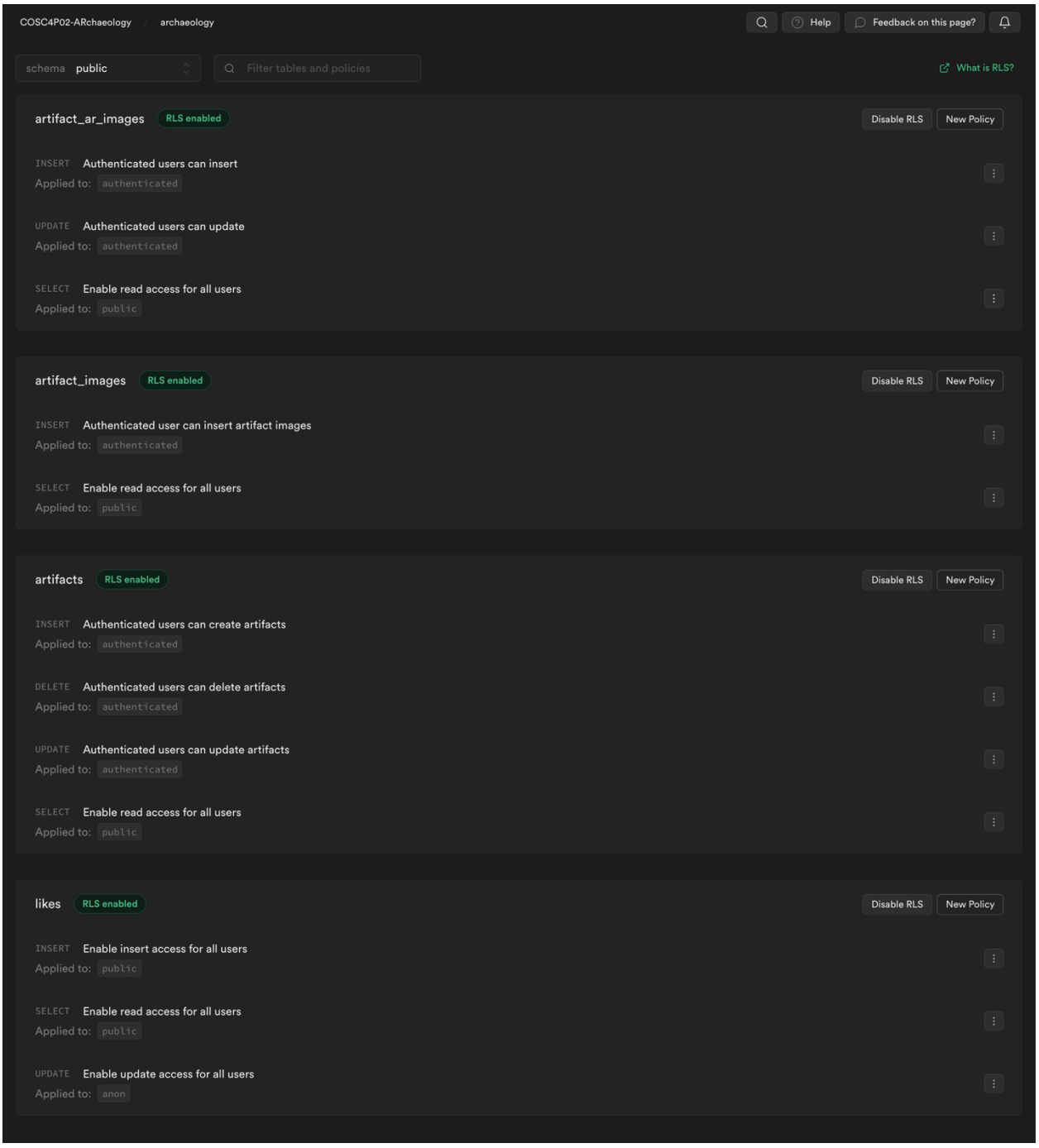


Figure 6 - Row-level security on database table rows

We also setup an authentication system. The users in the following screenshot are users of the web app, not Supabase. These users are the “authenticated” users referenced in the row-level authorization strategies shown above.

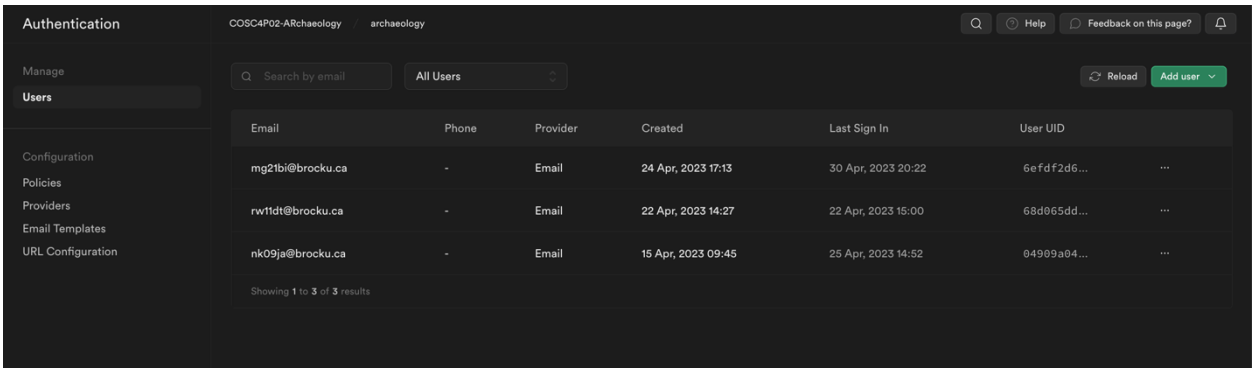


Figure 7 - Authenticated users of the web application

1. Summary

Meeting title	Sprint Start
Attended par	6
Start time	2/16/23, 2:27:59 PM
End time	2/16/23, 3:49:41 PM
Meeting dura	1h 21m 42s
Average atte	1h 9m 14s

2. Participants

Name	First Join	Last Leave	In-Meeting D	Email	Participant ID	Role
Mohit Gumn	2/16/23, 2:2	2/16/23, 3:4	1h 20m 40s	mg21bi@brc	mg21bi@brc	Organizer
Robert Worr	2/16/23, 2:2	2/16/23, 3:4	1h 18m 58s	rw11dt@bro	rw11dt@bro	Presenter
Nickolas Ken	2/16/23, 2:3	2/16/23, 3:4	1h 12m 52s	nk09ja@broc	nk09ja@broc	Presenter
Gurkirat Bair	2/16/23, 2:3	2/16/23, 3:4	1h 12m 52s	gb21wi@bro	gb21wi@bro	Presenter
David Boere	2/16/23, 2:3	2/16/23, 3:4	1h 7m 48s	db19zj@broc	db19zj@broc	Presenter
Abe Shakeel	2/16/23, 3:0	2/16/23, 3:4	42m 13s	as17tg@broc	as17tg@broc	Presenter

3. In-Meeting Activities

Name	Join Time	Leave Time	Duration	Email	Role
Mohit Gumn	2/16/23, 2:2	2/16/23, 3:4	1h 20m 40s	mg21bi@brc	Organizer
Robert Worr	2/16/23, 2:2	2/16/23, 3:4	1h 18m 58s	rw11dt@bro	Presenter
Nickolas Ken	2/16/23, 2:3	2/16/23, 3:4	1h 12m 52s	nk09ja@broc	Presenter
Gurkirat Bair	2/16/23, 2:3	2/16/23, 3:4	1h 12m 52s	gb21wi@bro	Presenter
David Boere	2/16/23, 2:3	2/16/23, 3:4	1h 7m 48s	db19zj@broc	Presenter
Abe Shakeel	2/16/23, 3:0	2/16/23, 3:4	42m 13s	as17tg@broc	Presenter

1. Summary

Meeting title	Sprint Start
Attended par	3
Start time	3/02/23, 2:28:12 PM
End time	3/02/23, 4:01:04 PM
Meeting dura	1h 32m 51s
Average atte	1h 29m 9s

2. Participants

Name	First Join	Last Leave	In-Meeting D	Email	Participant ID	Role
Mohit Gumn	3/02/23, 2:2	3/02/23, 4:0	1h 31m 14s	mg21bi@brc	mg21bi@brc	Organizer
Robert Worr	3/02/23, 2:3	3/02/23, 4:0	1h 29m 51s	rw11dt@bro	rw11dt@bro	Presenter
Nickolas Ken	3/02/23, 2:3	3/02/23, 4:0	1h 26m 21s	nk09ja@broc	nk09ja@broc	Presenter

3. In-Meeting Activities

Name	Join Time	Leave Time	Duration	Email	Role
Mohit Gumn	3/02/23, 2:2	3/02/23, 4:0	1h 31m 14s	mg21bi@brc	Organizer
Robert Worr	3/02/23, 2:3	3/02/23, 4:0	1h 29m 51s	rw11dt@bro	Presenter
Nickolas Ken	3/02/23, 2:3	3/02/23, 4:0	1h 26m 21s	nk09ja@broc	Presenter

1. Summary

Meeting title	Sprint Start
Attended par	5
Start time	3/09/23, 2:29:47 PM
End time	3/09/23, 2:51:24 PM
Meeting dura	21m 37s
Average atte	17m 28s

2. Participants

Name	First Join	Last Leave	In-Meeting D	Email	Participant ID	Role
Mohit Gumn	3/09/23, 2:2	3/09/23, 2:5	21m 28s	mg21bi@brc	mg21bi@brc	Organizer
Robert Worr	3/09/23, 2:3	3/09/23, 2:5	19m 45s	rw11dt@bro	rw11dt@bro	Presenter
David Boere	3/09/23, 2:3	3/09/23, 2:5	17m 29s	db19zj@broc	db19zj@broc	Presenter
Abe Shakeel	3/09/23, 2:3	3/09/23, 2:5	14m 24s	as17tg@broc	as17tg@broc	Presenter
Nickolas Ken	3/09/23, 2:3	3/09/23, 2:5	14m 15s	nk09ja@broc	nk09ja@broc	Presenter

3. In-Meeting Activities

Name	Join Time	Leave Time	Duration	Email	Role
Mohit Gumn	3/09/23, 2:2	3/09/23, 2:5	21m 28s	mg21bi@brc	Organizer
Robert Worr	3/09/23, 2:3	3/09/23, 2:5	19m 45s	rw11dt@bro	Presenter
David Boere	3/09/23, 2:3	3/09/23, 2:5	17m 29s	db19zj@broc	Presenter
Abe Shakeel	3/09/23, 2:3	3/09/23, 2:5	14m 24s	as17tg@broc	Presenter
Nickolas Ken	3/09/23, 2:3	3/09/23, 2:5	14m 15s	nk09ja@broc	Presenter

1. Summary

Meeting title	Sprint Start
Attended par	4
Start time	3/16/23, 2:55:46 PM
End time	3/16/23, 3:56:05 PM
Meeting dura	1h 18s
Average atte	54m 32s

2. Participants

Name	First Join	Last Leave	In-Meeting D	Email	Participant ID	Role
Mohit Gumn	3/16/23, 3:0	3/16/23, 3:5	52m 55s	mg21bi@brc	mg21bi@brc	Organizer
David Boere	3/16/23, 2:5	3/16/23, 3:5	1h 3s	db19zj@broc	db19zj@broc	Presenter
Nickolas Ken	3/16/23, 3:0	3/16/23, 3:5	52m 48s	nk09ja@broc	nk09ja@broc	Presenter
Robert Worr	3/16/23, 3:0	3/16/23, 3:5	52m 22s	rw11dt@bro	rw11dt@bro	Presenter

3. In-Meeting Activities

Name	Join Time	Leave Time	Duration	Email	Role
Mohit Gumn	3/16/23, 3:0	3/16/23, 3:5	52m 55s	mg21bi@brc	Organizer
David Boere	3/16/23, 2:5	3/16/23, 3:5	1h 3s	db19zj@broc	Presenter
Nickolas Ken	3/16/23, 3:0	3/16/23, 3:5	52m 48s	nk09ja@broc	Presenter
Robert Worr	3/16/23, 3:0	3/16/23, 3:5	52m 22s	rw11dt@bro	Presenter

1. Summary

Meeting title	COSC-4P02 War room
Attended participants	4
Start time	4/22/23, 11:16:07 AM
End time	4/22/23, 1:47:09 PM
Meeting duration	2h 31m 1s
Average attendance time	1h 45m 30s

2. Participants

Name	First Join	Last Leave	In-Meeting Duration
Mohit Gummaraj Kishore	4/22/23, 11:40:56 AM	4/22/23, 1:35:32 PM	1h 54m 36s
Nickolas Kenyeres	4/22/23, 11:16:13 AM	4/22/23, 1:35:10 PM	2h 18m 57s
Robert Worron	4/22/23, 11:30:00 AM	4/22/23, 1:47:09 PM	2h 17m 8s
Abe Shakeel	4/22/23, 11:45:56 AM	4/22/23, 12:17:18 PM	31m 21s

3. In-Meeting Activities

Name	Join Time	Leave Time	Duration
Mohit Gummaraj Kishore	4/22/23, 11:40:56 AM	4/22/23, 1:35:32 PM	1h 54m 36s
Nickolas Kenyeres	4/22/23, 11:16:13 AM	4/22/23, 1:35:10 PM	2h 18m 57s
Robert Worron	4/22/23, 11:30:00 AM	4/22/23, 1:47:09 PM	2h 17m 8s
Abe Shakeel	4/22/23, 11:45:56 AM	4/22/23, 12:17:18 PM	31m 21s

Email	Participant ID (UPN)	Role
mg21bi@brocku.ca	mg21bi@brocku.ca	Organizer
nk09ja@brocku.ca	nk09ja@brocku.ca	Presenter
rw11dt@brocku.ca	rw11dt@brocku.ca	Presenter
as17tg@brocku.ca	as17tg@brocku.ca	Presenter

Email	Role
mg21bi@brocku.ca	Organizer
nk09ja@brocku.ca	Presenter
rw11dt@brocku.ca	Presenter
as17tg@brocku.ca	Presenter