Final Report

COSC 4P02
Software Engineering II
Naser Ezzati-Jivan
Winter 2023

April 30th, 2023
Brock University

Alec Ames
Team Leader
Product Owner
aa19vl@brocku.ca
6843577

Matthew Benson
Developer
mb19hz@brocku.ca
6729388

Ibrahim Hashmi
Developer
ih17px@brocku.ca
6352926

Francis Monwe
Developer
fm19pe@brocku.ca
6724355

Tommy Pham
Developer
tp19if@brocku.ca
6733646

Abhijeet Prajapati
Developer
ap15qm@brocku.ca
5987722

Justin Stickel
Scrum Master
js19ge@brocku.ca
6718704

Haaris Yahya
Developer
hy20ao@brocku.ca
7054984

Table of Contents

1. Overview

This document serves as a technical guide, offering a walkthrough of our interactive timeline and presenting a comprehensive report on our development progress during this term. We adhered to a disciplined and structured incremental-iterative approach for our agile process, allowing us to product and enhance existing deliverables by incorporating significant additional features. Moreover, this report also includes manuals covering how to access and navigate through the application, along with explanations of each implemented feature and its functionality in our product.

2. User and Technical Manuals

User manual

Accessing the web application

The web application is available at museumtimeline.vercel.app and on the Google Play store as an app package. The application can also be run locally with the instructions found in the installation manual.

Navigating the timeline

The timeline can be navigated using multiple methods.

Timeline

- Click on timeline items in the left column to view it.
- Click the plus button or double-click to zoom in the timeline.
- Click the minus button to zoom out the timeline bar, and refresh to reset both zoom and position.
- Once zoomed in, click and drag or scroll on the timeline to move up or down.

Search Bar

- Click on the search bar or press the forward slash key and type in a keyword or year to search for a timeline item.
- Click on the timeline item in the search results to navigate to it.
- Click on the close button to clear the search bar field.

Navigation Buttons

- Mobile — Swipe left or right to navigate through timeline items.
- Mouse — Click on the up/down floating buttons at the top and bottom of the screen to navigate up or down through timeline items.
- Keyboard — Press up/left or down/right to navigate up or down through timeline items.

Fullscreen

- Click on the image to view it in fullscreen mode. Click on the image again to exit full screen mode.
- On mobile, pinch pinch to zoom in and out of the image.

Accessibility

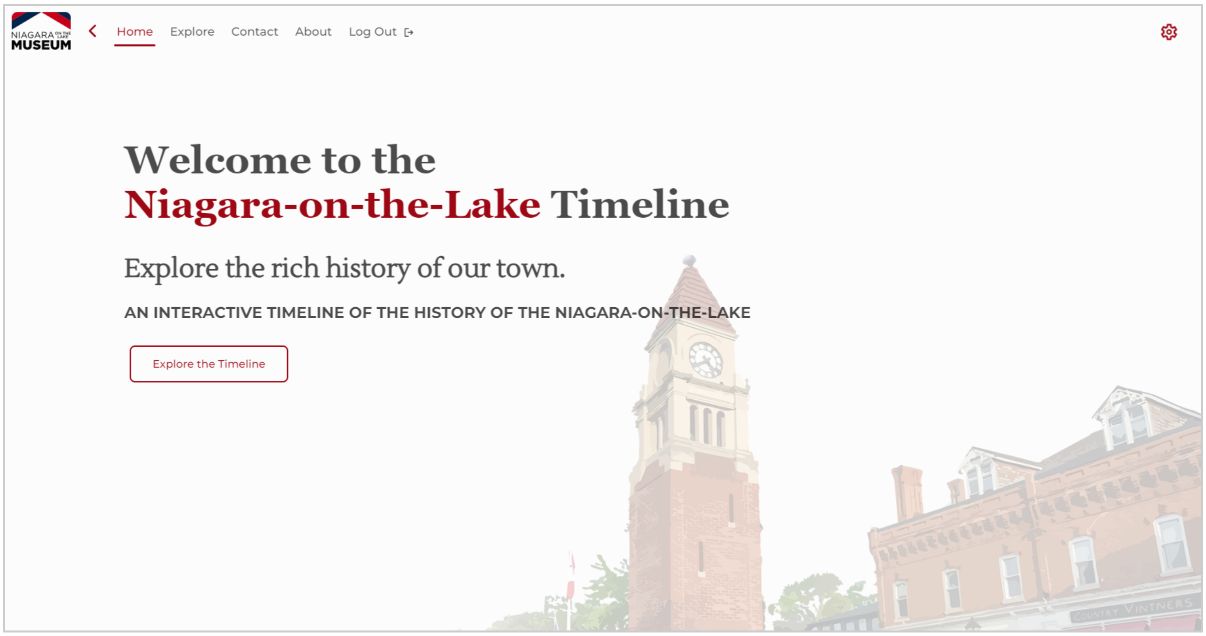
- Press the speaker button to hear a description of the current timeline item. Press the stop button to stop the audio playback.

Features and functionalities

Landing Page

The landing page was designed to be simplistic. This design choice promotes ease of use for new visitors. Upon first glance, visitors can immediately understand the purpose of the webpage. All buttons are labeled appropriately to help with navigation.

Landing page:



Welcome to the Niagara-on-the-Lake Timeline

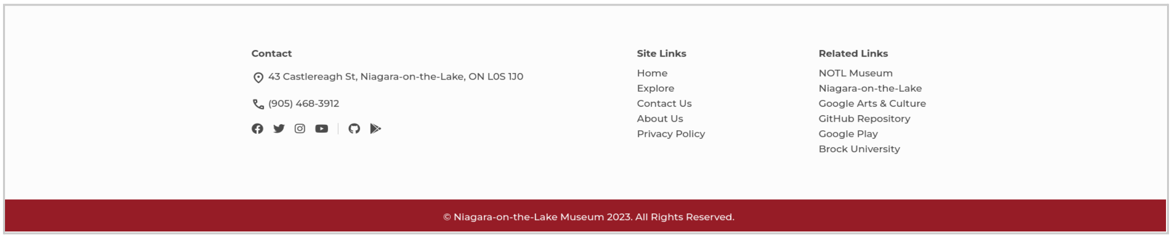
Explore the rich history of our town.

AN INTERACTIVE TIMELINE OF THE HISTORY OF THE NIAGARA-ON-THE-LAKE

Footer

A footer component can be found at the bottom of each page excluding the timeline. This footer includes links and additional information to help visitors find other related websites, access support features, or navigate to other pages of the website, as well as accessing the privacy policy.

Footer element:


Contact

43 Castlereagh St, Niagara-on-the-Lake, ON L0S 1J0

(905) 468-3912

Site Links

Home
Explore
Contact Us
About Us
Privacy Policy

Related Links

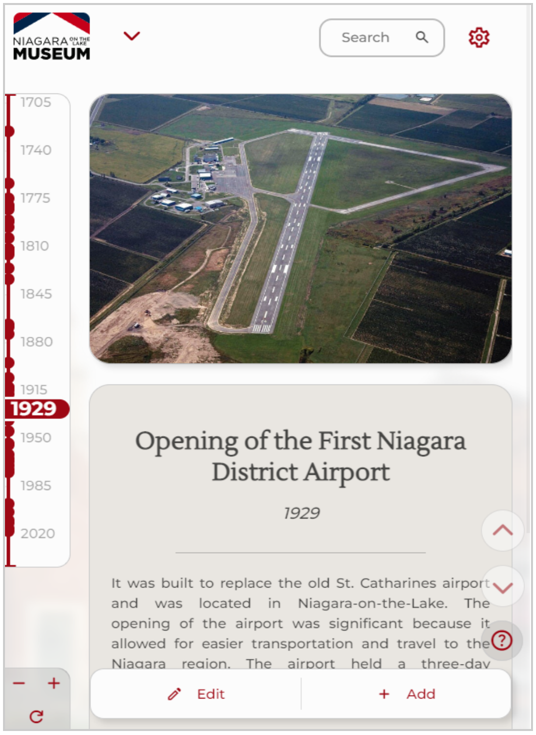
NOTL Museum
Niagara-on-the-Lake
Google Arts & Culture
GitHub Repository
Google Play
Brock University

© Niagara-on-the-Lake Museum 2023. All Rights Reserved.

Timeline Navigation

The timeline navigation component is the main feature of our web application. It arranges events chronologically across the timeline and allows for easy navigation between them. Users can view all events and gain context of the current item. The timeline features a time scale that snaps to the nearest 20, 10, 5, or single years depending on the zoom level, and switching between events plays a transition animation to enhance user experience.

View of the timeline navigation component on mobile:



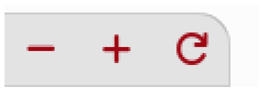
Opening of the First Niagara District Airport

1929

It was built to replace the old St. Catharines airport and was located in Niagara-on-the-Lake. The opening of the airport was significant because it allowed for easier transportation and travel to the Niagara region. The airport held a three-day

The navigation zoom and offset can be modified with the navigation controls below. To adjust the field of view on the timeline, the user can use the minus button to decrease, the plus button to increase it, and the reset button to revert both the zoom and offset to defaults.

Timeline navigation zoom controls:

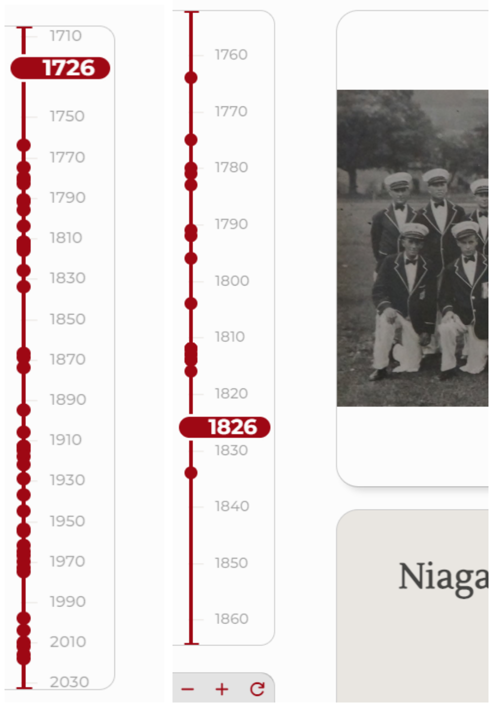


Cursor Component

The current selected item is indicated with a cursor element displaying the year of the event. The year, and thus the position, transitions smoothly between values for a better user interface design.

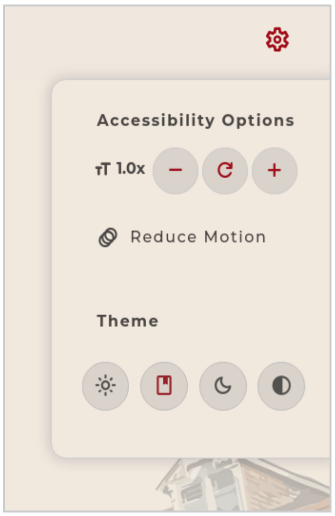
Left: fully zoomed out view of the timeline;

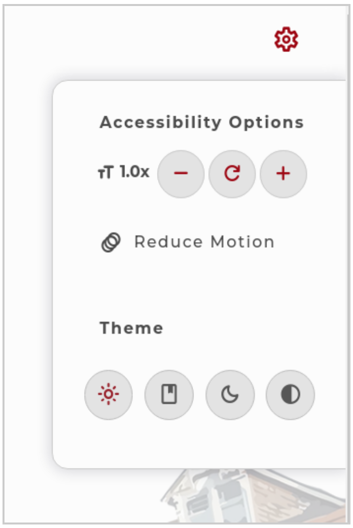
Right: zoomed in view of the timeline, offset to position year 1826 focused:

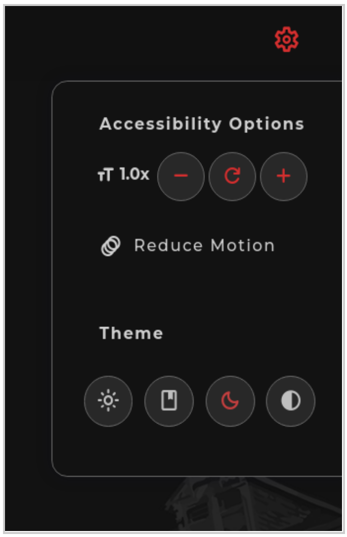


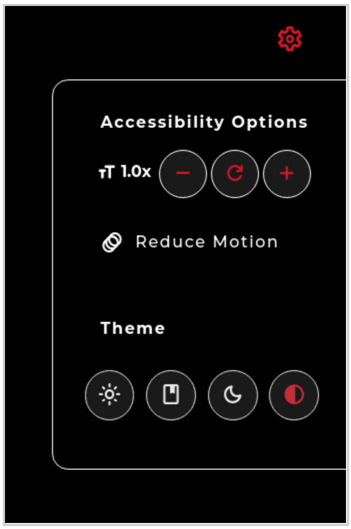
Accessibility Menu

An accessibility menu was designed to enhance the usability of the web application by offering various options. The menu consists of a font-size controller that allows zooming from 0.5x to 1.7x, reduced motion for timeline item transitions, and a theme selector with three color schemes. A text-to-speech feature that utilizes the built-in web voice synthesis engine can read item contents aloud to viewers.





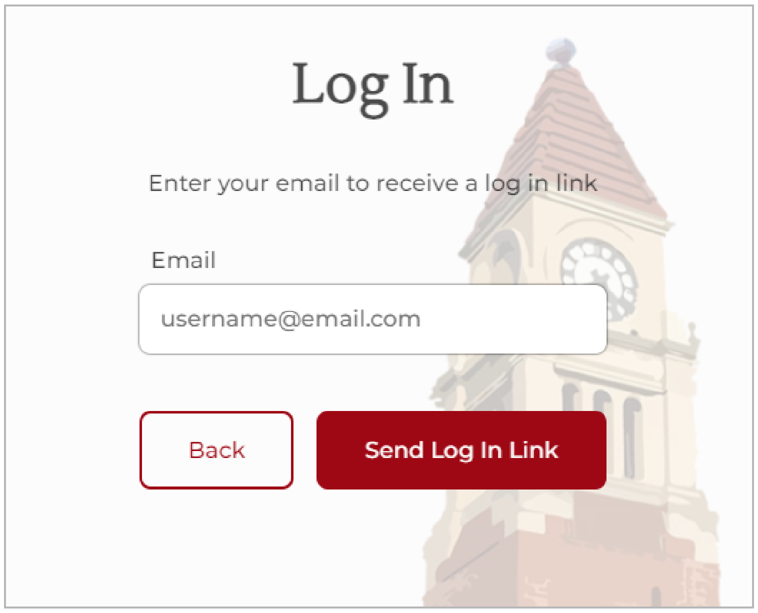




Staff Authentication

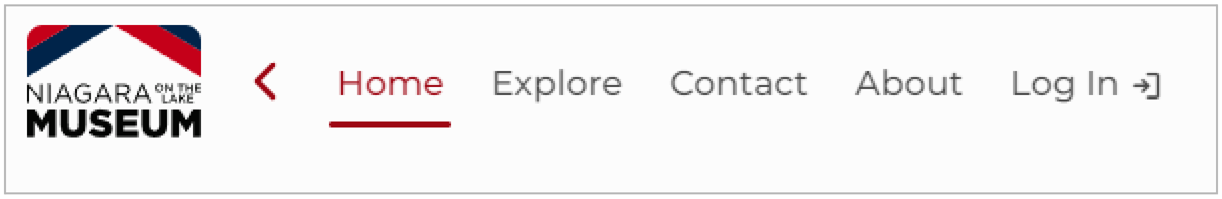
We implemented a user authentication feature for our client upon request. The process is user-friendly and simplistic in design to prevent confusion. It involves a single text field for authenticated emails and a submit button. Users with an authenticated email address will receive a sign in link after entering and submitting their email. The link will redirect the user to a modified version of the web application with exclusive administrative privileges.

Login page:

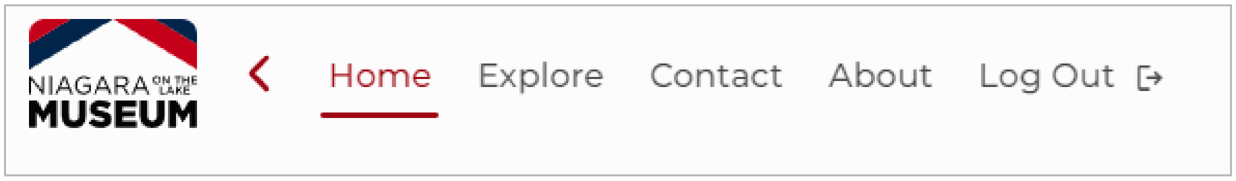


Additionally, the user's authenticated email is displayed in the navbar while logged in, providing an efficient and convenient experience for staff members.

Appearance of navigation menu when user is not signed in:

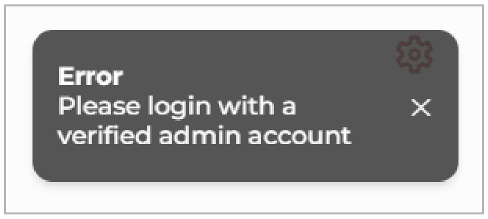


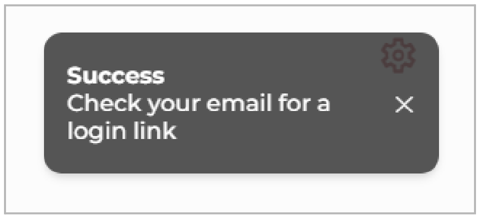
Appearance of navigation menu when user is authenticated, hovering over "Log Out" button shows authenticated email:

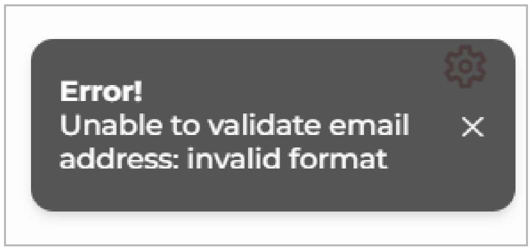


To meet the client's requirements, we ensured that the user authentication feature is straightforward and user-friendly, providing staff with an efficient login process. Our emphasis on user-friendliness guaranteed that the process met the client's specifications. To enhance security, we implemented the Magic Link authentication method, allowing verified users to access their privileges without passwords. This method is powered by Supabase, our backend solution, and attempted logins with unregistered email addresses trigger an error message stating that Magic Link sign-ups are disabled.

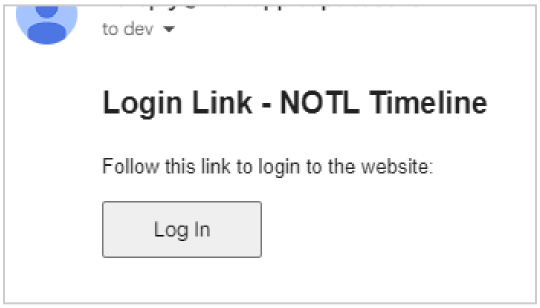
Top-left: toast notification when an unregistered user attempts to sign in; Top-right: toast notification when a registered user attempts to sign in; Bottom: toast notification when invalid email is entered:


Error
Please login with a verified admin account


Success
Check your email for a login link


Error!
Unable to validate email address: invalid format

Resulting email sent to the user's inbox:


to dev

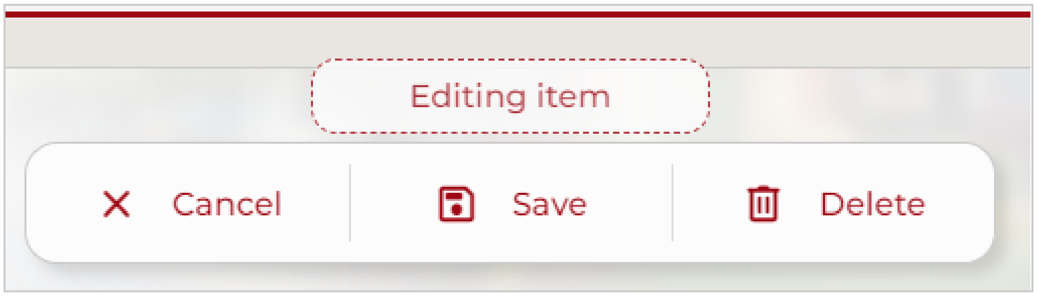
Login Link - NOTL Timeline

Follow this link to login to the website:

Event Editor

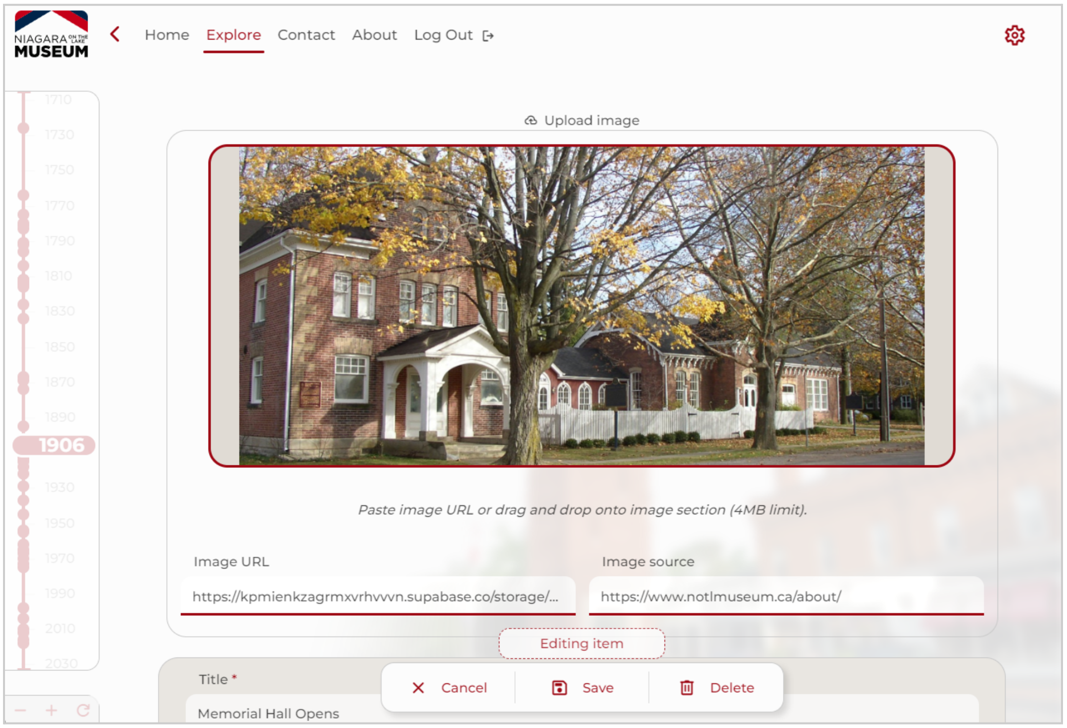
To enhance user experience, authenticated users were provided with additional menu options located on the bottom of the screen, enabling them to update, add, and remove timeline items. Every successful action, whether it involves saving, adding, or deleting an item, triggers a toast notification that provides similar functionality to the authentication process. When a user attempts to delete an item, a confirmation dialog box appears to ensure the user's intent to avoid mistakes.

Editor menu shown to authenticated users:



To make it easier to edit or add events, we redesigned the layout to show input boxes and text areas for each property of the timeline entry. This allows users to provide URLs for images, or upload images to the Supabase storage bucket by dragging and dropping or selecting a file. Additionally, the editor menu includes a notice to indicate if the user is editing or adding to the timeline.

Image editor UI:



Paste image URL or drag and drop onto image section (4MB limit).

Image URL: https://kpmienkzagrmxvrhvvvn.supabase.co/storage/...

Image source: https://www.notlmuseum.ca/about/

Title *: Memorial Hall Opens

Window for adding a new timeline event (text properties):

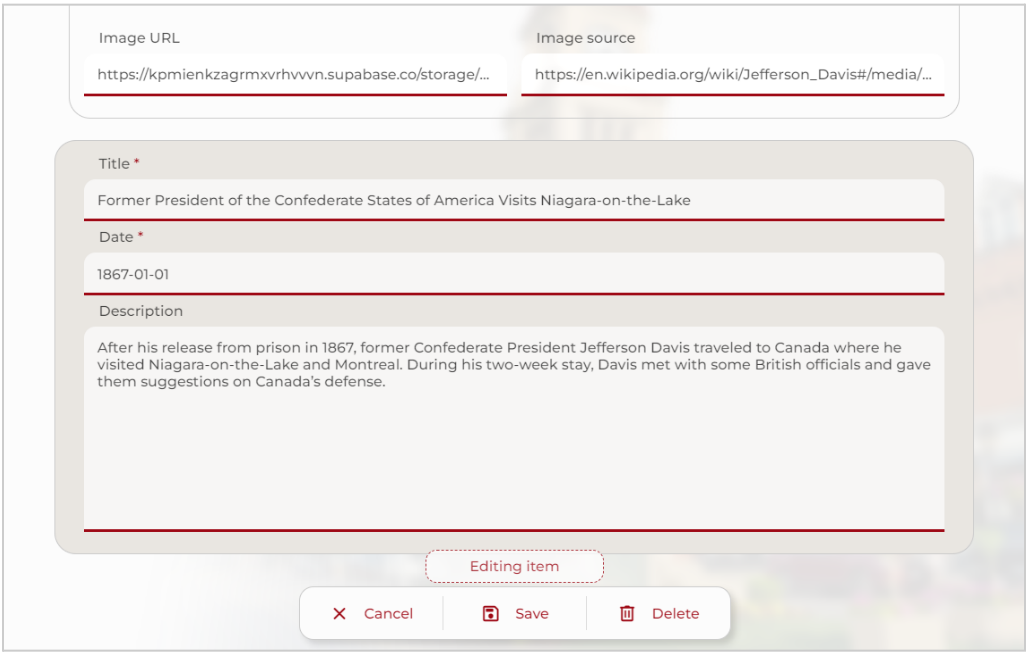


Image URL

https://kpmienkzagrmxvrhvvvn.supabase.co/storage/...

Image source

https://en.wikipedia.org/wiki/Jefferson_Davis#/media/...

Title *

Former President of the Confederate States of America Visits Niagara-on-the-Lake

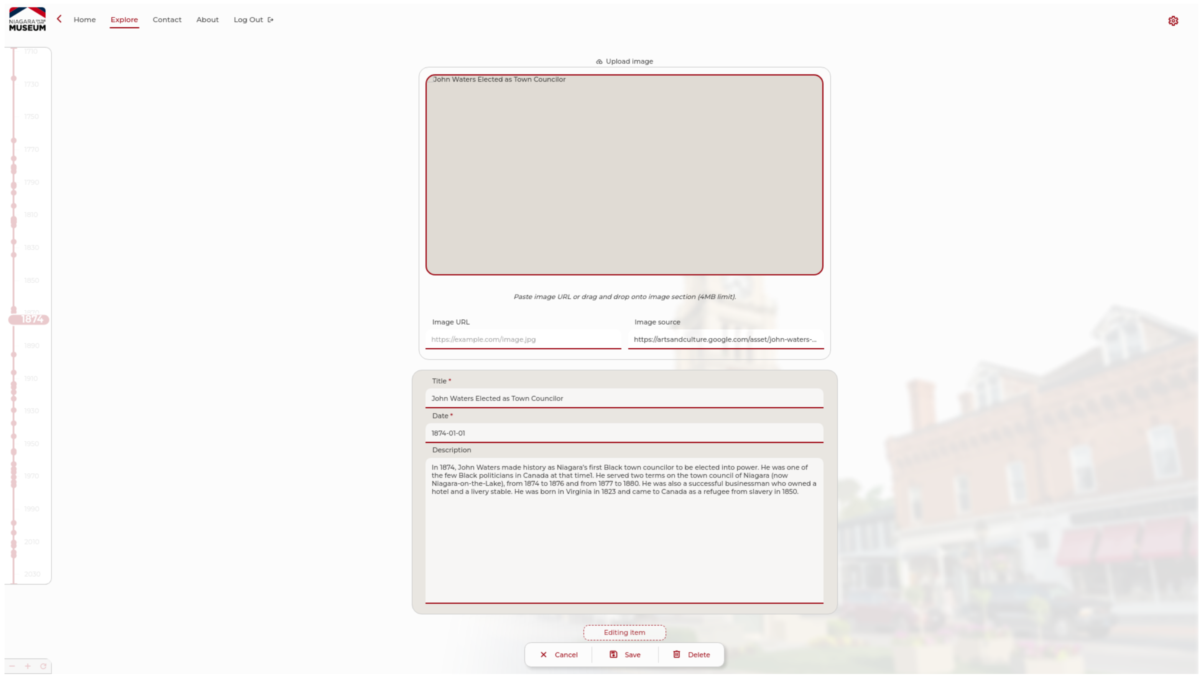
Date *

1867-01-01

Description

After his release from prison in 1867, former Confederate President Jefferson Davis traveled to Canada where he visited Niagara-on-the-Lake and Montreal. During his two-week stay, Davis met with some British officials and gave them suggestions on Canada's defense.

While in edit mode, the event editor populates the input fields with data from the selected entry, allowing the user to make edits as necessary. This feature prevents administrators from making mistakes when changing existing texts.



NIAGARA ON THE LAKE MUSEUM
Home
Explore
Contact
About
Log Out
1874
Upload image
John Waters Elected as Town Councilor

Paste image URL or drag and drop onto image section (4MB limit).

Image URL

https://example.com/image.jpg

Image source

https://artsandculture.google.com/asset/john-waters-...

Title *

John Waters Elected as Town Councilor

Date *

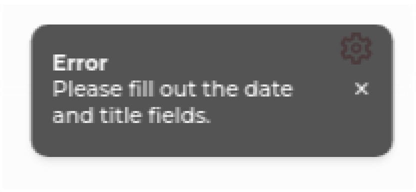
1874-01-01

Description

In 1874, John Waters made history as Niagara's first Black town councilor to be elected into power. He was one of the few Black politicians in Canada at that time1. He served two terms on the town council of Niagara (now Niagara-on-the-Lake), from 1874 to 1876 and from 1877 to 1880. He was also a successful businessman who owned a hotel and a livery stable. He was born in Virginia in 1823 and came to Canada as a refugee from slavery in 1850.

To ensure completeness of the timeline entry, both the ‘title’ and ‘date’ fields are required, and the application will display an error toast to prompt the user if either field is left empty.

Toast prompt for date and title fields.



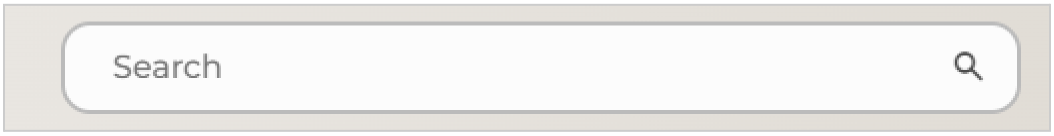
Error
Please fill out the date and title fields.

The ‘date’ field auto-formats the entered digits into the ISO standard date that entries are saved as in our database. The image frame provides multiple options for adding an image, such as drag-and-drop, pasting an image address, and picking a file from the disk. It also includes a live preview of the selected image. Changes made by the user are not saved to the server until they click the 'save' button or the 'delete' button (optional). When in ‘add’ mode, users are able to perform actions similar to those in the event editor, except now text fields contain placeholders to guide user input, and menu options are available only for saving and canceling changes. To improve usability, the chosen form-item border was enlarged to indicate the user’s selection. Moreover, the form styling was tested to ensure effective functionality across the four themes that are available for selection in the accessibility menu.

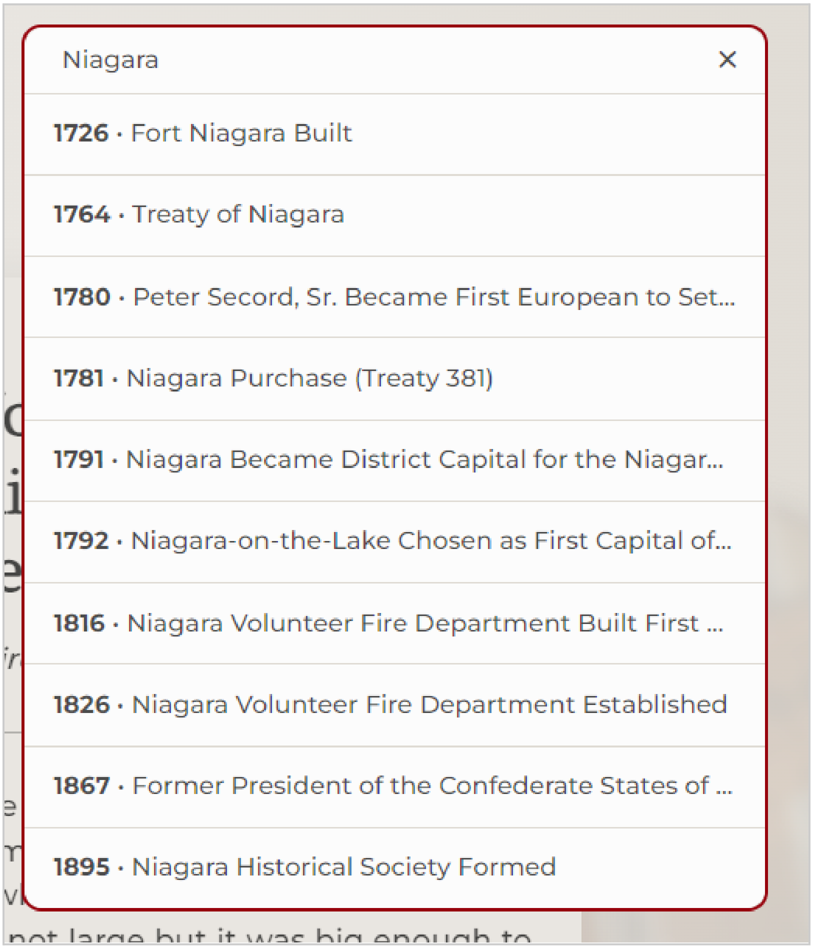
Search Bar

The search bar component filters a list (array) of dates and titles based on the user's search keywords. It can search for partial words, full sentences, and years. The component uses JavaScript’s standard library filter() function to perform the search, returning a new array containing all elements that satisfy a user defined testing function. When the user clicks on an item in the search results, the component navigates to that item in the list.

Search bar with no search query entered:



List of results in the drop down portion of the search bar:



Niagara

1726 · Fort Niagara Built

1764 · Treaty of Niagara

1780 · Peter Secord, Sr. Became First European to Set...

1781 · Niagara Purchase (Treaty 381)

1791 · Niagara Became District Capital for the Niagar...

1792 · Niagara-on-the-Lake Chosen as First Capital of...

1816 · Niagara Volunteer Fire Department Built First ...

1826 · Niagara Volunteer Fire Department Established

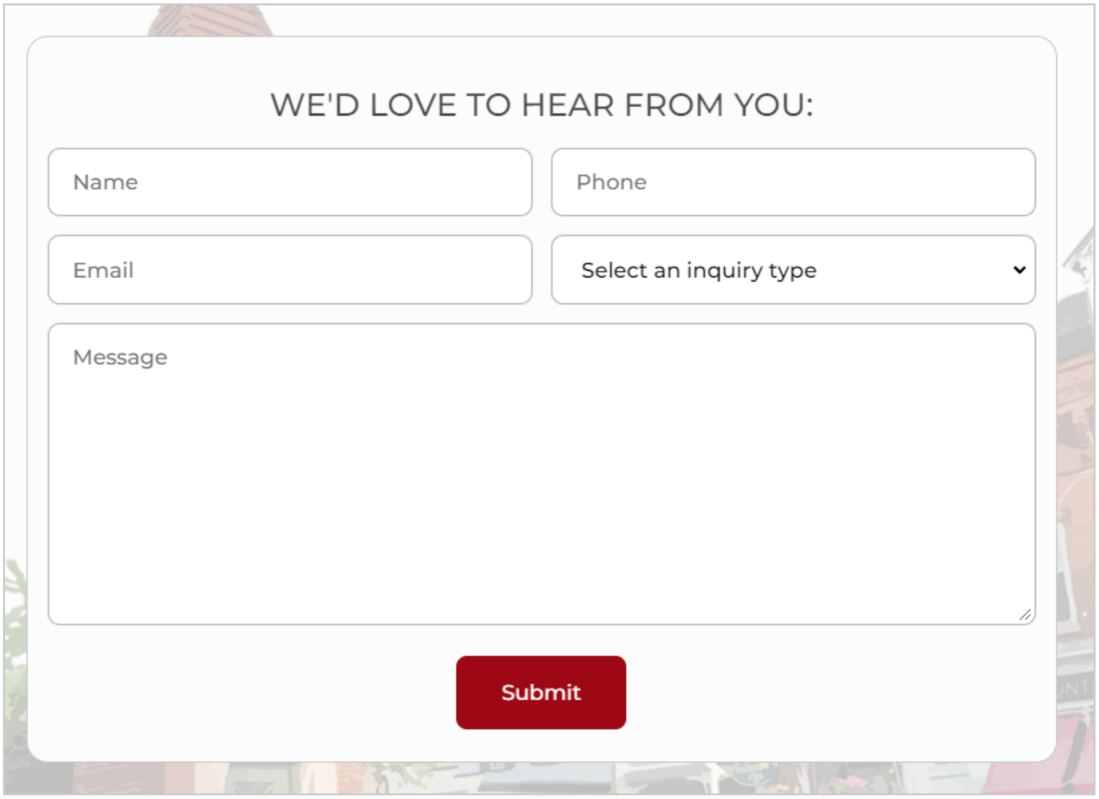
1867 · Former President of the Confederate States of ...

1895 · Niagara Historical Society Formed

Contact Page

The contact page allowed for feedback and inquiries to be sent to our developer team or the museum staff. The contact form includes standard contact information to allow for responding to inquiries. This contact page also includes an embedded Google Maps frame, as well as the address, contact and social media information for the museum.

Inquiry form on contact page:



WE'D LOVE TO HEAR FROM YOU:

About Page

The about page showcases information about the town of Niagara-on-the-Lake, the NOTL Museum, and a short description about our timeline project. There is also a section dedicated to showing our development team.

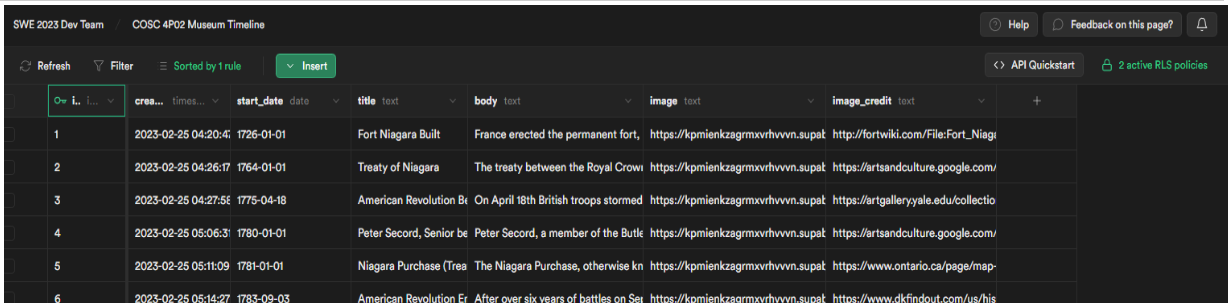
Technical manual

Application architecture

Database

To store timeline event data, images, and to handle user authentication, we used the Supabase back-end service. Our timeline table contains the data required to dynamically populate the timeline event components through Supabase API calls. The following table represents the structure of a single entry in the database:

Field	Type	Description
`id`	`Int`	Auto-incrementing integer for each timeline event added
`created_at`	`Timestamp+TZ`	Timestamp of creation date and time
`start_date`	`ISO Date`	Date for the timeline item in the format YYYY-MM-DD
`title`	`Text`	Event title
`body`	`Text`	Event description content
`image`	`Text`	Image or YouTube URL
`image_credit`	`Text`	Source of image URL



i.. i...	crea... times...	start_date date	title text	body text	image text	image_credit text
1	2023-02-25 04:20:4	1726-01-01	Fort Niagara Built	France erected the permanent fort,	https://kpmienkzagrmxvrhvvvn.supat	http://fortwiki.com/File:Fort_Niag
2	2023-02-25 04:26:17	1764-01-01	Treaty of Niagara	The treaty between the Royal Crow	https://kpmienkzagrmxvrhvvvn.supat	https://artsandculture.google.com/
3	2023-02-25 04:27:5	1775-04-18	American Revolution B	On April 18th British troops stormed	https://kpmienkzagrmxvrhvvvn.supat	https://artgallery.yale.edu/collectio
4	2023-02-25 05:06:3	1780-01-01	Peter Secord, Senior be	Peter Secord, a member of the Butle	https://kpmienkzagrmxvrhvvvn.supat	https://artsandculture.google.com/
5	2023-02-25 05:11:09	1781-01-01	Niagara Purchase (Trea	The Niagara Purchase, otherwise kn	https://kpmienkzagrmxvrhvvvn.supat	https://www.ontario.ca/page/map-
6	2023-02-25 05:14:27	1783-09-03	American Revolution E	After over six years of battles on Se	https://kpmienkzagrmxvrhvvvn.supat	https://www.dkfindout.com/us/his

Additionally, we kept the images used for the timeline in a secure storage container and provided links to them for user access.

Below is the PostgreSQL table definitions for our application:

Timeline Data:

```
create table
  public.timeline (
    id bigint generated by default as identity not null,
    created_at timestamp with time zone null default now(),
    start_date date not null,
    title text null,
    body text null,
    image text null,
    image_credit text null,
    constraint timeline_pkey primary key (id),
    constraint timeline_id_key unique (id)
  ) tablespace pg_default;
```

Contact Us Inquiries:

```
create table
  public.inquiries (
    id bigint generated by default as identity not null,
    submitter_name text null,
    email text null,
    phone numeric null,
    inquiry_type text null,
    message text null,
    created_at timestamp with time zone null default (now() at time zone
'edt'::text),
    constraint inquiries_pkey primary key (id)
    ) tablespace pg_default;
```

Technology Stack

For the development of the application, we used the following technologies:

- **SvelteKit** - Front-end framework
  - Svelte/Kit is a relatively new front-end framework that uses enhanced HTML, CSS, and Javascript syntax and compiles it in the build process, making it extremely performant and intuitive to use.
- **Supabase** - Back-end
  - Supabase is a free and open-source back-end service that provides a Postgres database, a storage bucket, and authentication service for development projects.
- **Playwright** - Testing
  - We used Microsoft's Playwright end-to-end browser testing toolkit to test our product on a variety of browser engines, and devices.
- **Lighthouse** - Application performance analysis
  - We leveraged the DevTools Lighthouse feature to capture performance metrics, as well as accessibility metrics. Lighthouse was also used to optimize for PWA and TWA.

Installation manual

The web application is available hosted on Vercel and available on Google Play as mentioned in the user manual, however, below are the instructions on getting the application running in a local development environment:

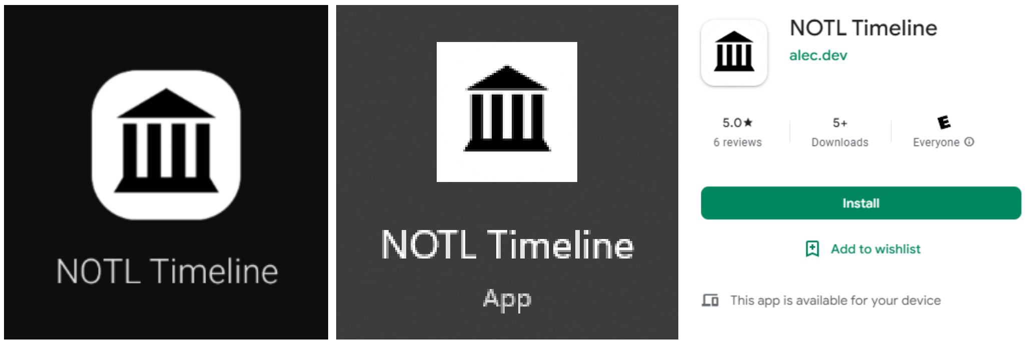
1. Install Node.js and npm if you don't already have these installed
2. Clone the repository:
   ```
   git clone https://github.com/SWE-2023/COSC-4P02-Project.git
   ```
3. Change directory to the timeline directory within the repo
   ```
   cd COSC-4P02-Project\timeline
   ```
4. Install the project dependencies:
   ```
   npm install
   ```
5. Run the project in the local dev server:
   ```
   npx vite
   ```
6. Press 'O' on the terminal or CTRL+Click the local URL provided by Vite.

This will open the locally-served web application in your browser.

PWA and Play Store Listing

An additional feature implemented was both Progressive Web Application (PWA), and Trusted Web Activity (TWA) support, by registering a service-worker and adding the required metadata and cache policies to the application. With PWA support, the timeline can be installed as an application on all platforms, offering convenience and an enhanced, immersive experience. By packaging it as a TWA, we were able to deploy the application to the Google Play store as a web activity that automatically updates whenever changes are made to the site.

Left: the application installed as an application on Android;
Middle: the application installed as an application on Windows 11;
Right: app listing on the Google Play store.



3. Sprints

Once the product backlog was created, we initiated our first sprint planning session the following week, with our sprint duration set at two weeks. These sprint planning sessions took place at the onset of each sprint, during which we determined our tasks and established our overall objective for the two-week period. Often, developers were recommended items for assignment by the Scrum Master and Product Owner based on their areas of expertise along with the areas of the project that they have previously been developing. To ensure fairness, as we entered each new sprint, developers with a higher number of story points were typically assigned fewer tasks, while those with a lower number were assigned more. This approach aimed to balance the workload among the team members and equalize the amount of work completed by each individual. Along with developmental work, other unrecorded contributions were also taken into account, such as: debugging, testing, researching, etc.

Meetings

Each sprint meeting was organized and hosted by the Scrum Master, with all developers in attendance to decide and collaborate on unassigned sprint items. Our team used Jira as a tool for organizing and planning our sprints. Tasks were assigned directly to developers and were accompanied with a detailed description and an effort value, which helped us estimate the completion time for each sprint item. At the end of each sprint, we used Miro to host a Sprint Retrospect by utilizing its collaborative whiteboard to collectively visualize and communicate our progress, as well as discuss any challenges we faced during the sprint.

Past Sprints

This section lists the main backlog items worked on in each sprint.

Sprint 1

- Creating the landing page
- Creating a mock-up for the timeline in Figma
- Creating font-size
- Developing functionality to store cookies for preferences for user sessions

Sprint 2

- Creating and finalizing our timeline engine
- Initializing database with Supabase
- Gathering images for selected database events
- Research and add to descriptions to timeline events
- Clean data and prepare for database
- Add menu to store accessibility options
- Implement dark mode as a theme
- Implement high contrast mode as a theme

- Include option to reduce motion for accessibility

Sprint 3

- Connect data from Supabase database with timeline application
- Create text-to-speech controls to read timeline text
- Timeline item search functionality
- Fix responsiveness of the timeline screen across different screen sizes
- Fix image overlap with logo when scrolling up on the timeline screen

Sprint 4

- Create edit buttons on timeline page
- Finalize login page and create authentication flag to use in C4SWE-24
- Make themes persist on reload
- Set up secure user authentication with Supabase
- Finalize transition on timeline event change
- Fill 'About' page on website with relevant information about the town, the museum and our project

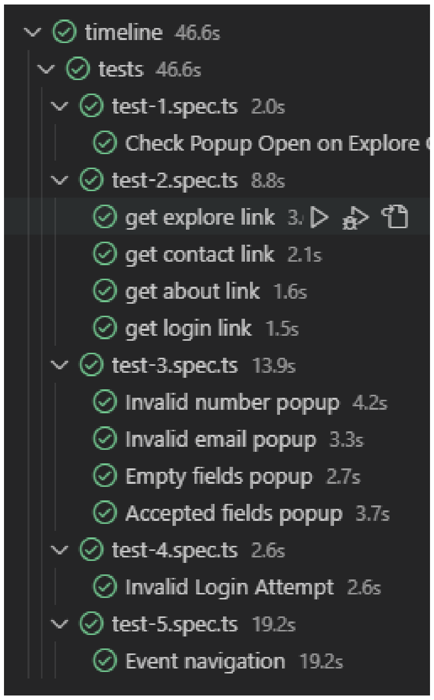
Sprint 5

- Create zoom in/out functionality on TimelineBar component
- Implement editing of timeline event data through Supabase
- Implement functionality for event edit buttons
- Create timeline cursor component
- Add functionality to handle years with several events

4. Testing

Throughout development, our team would ensure features would have all functionality tested before being merged to the main branch. If any issues were discovered, they would be promptly investigated and corrected upon identification. If these issues were significant and required more time, they would then be logged as a bug in Jira to later be assigned as a work item in a future sprint. Once Sprint 5 was concluded, the timeline's functionality was rigorously tested through self and peer testing. As a team we opted for the end-to-end testing strategy for our application, where we modeled real-world interactions with all aspects of the website. To facilitate testing, a separate dedicated 'testing' branch was created, where Microsoft's Playwright was employed. This framework enabled us to specify the anticipated behavior resulting from a sequence of actions and verify whether those actions indeed produced the desired behavior on various browser rendering engines such as Webkit and Mozilla. In the event that a test failed, we proceeded to debug and implement the necessary changes to guarantee proper functionality, while also addressing any edge cases. These tests were run whenever any additional features or changes were added to the main branch, allowing us to verify the expected behavior of new code. At the time of this report, all tests, encompassing functionality across all pages, have successfully passed, indicating that all features are working as expected:

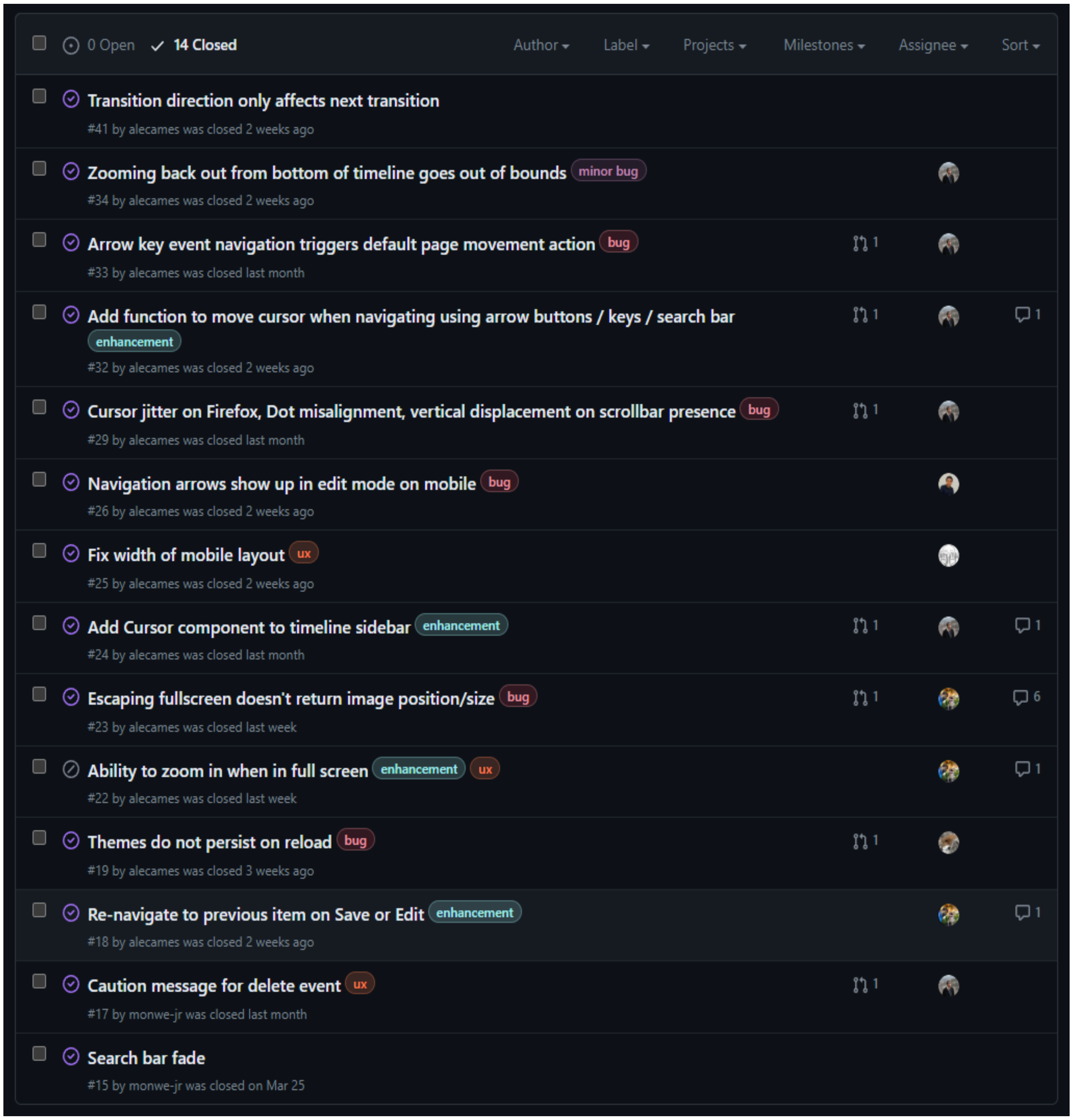
Test report generate by Playwright VS Code extension:



```
timeline 46.6s
  tests 46.6s
    test-1.spec.ts 2.0s
      Check Popup Open on Explore
    test-2.spec.ts 8.8s
      get explore link 3.
      get contact link 2.1s
      get about link 1.6s
      get login link 1.5s
    test-3.spec.ts 13.9s
      Invalid number popup 4.2s
      Invalid email popup 3.3s
      Empty fields popup 2.7s
      Accepted fields popup 3.7s
    test-4.spec.ts 2.6s
      Invalid Login Attempt 2.6s
    test-5.spec.ts 19.2s
      Event navigation 19.2s
```

5. GitHub Contributions

Below are logs of the GitHub Issues and Pull Request activity:



0 Open ✓ 14 Closed — Author ▾ Label ▾ Projects ▾ Milestones ▾ Assignee ▾ Sort ▾

Transition direction only affects next transition
#41 by alecames was closed 2 weeks ago

Zooming back out from bottom of timeline goes out of bounds `minor bug`
#34 by alecames was closed 2 weeks ago

Arrow key event navigation triggers default page movement action `bug` — 1
#33 by alecames was closed last month

Add function to move cursor when navigating using arrow buttons / keys / search bar `enhancement` — 1 — 1
#32 by alecames was closed 2 weeks ago

Cursor jitter on Firefox, Dot misalignment, vertical displacement on scrollbar presence `bug` — 1
#29 by alecames was closed last month

Navigation arrows show up in edit mode on mobile `bug`
#26 by alecames was closed 2 weeks ago

Fix width of mobile layout `ux`
#25 by alecames was closed 2 weeks ago

Add Cursor component to timeline sidebar `enhancement` — 1 — 1
#24 by alecames was closed last month

Escaping fullscreen doesn't return image position/size `bug` — 1 — 6
#23 by alecames was closed last week

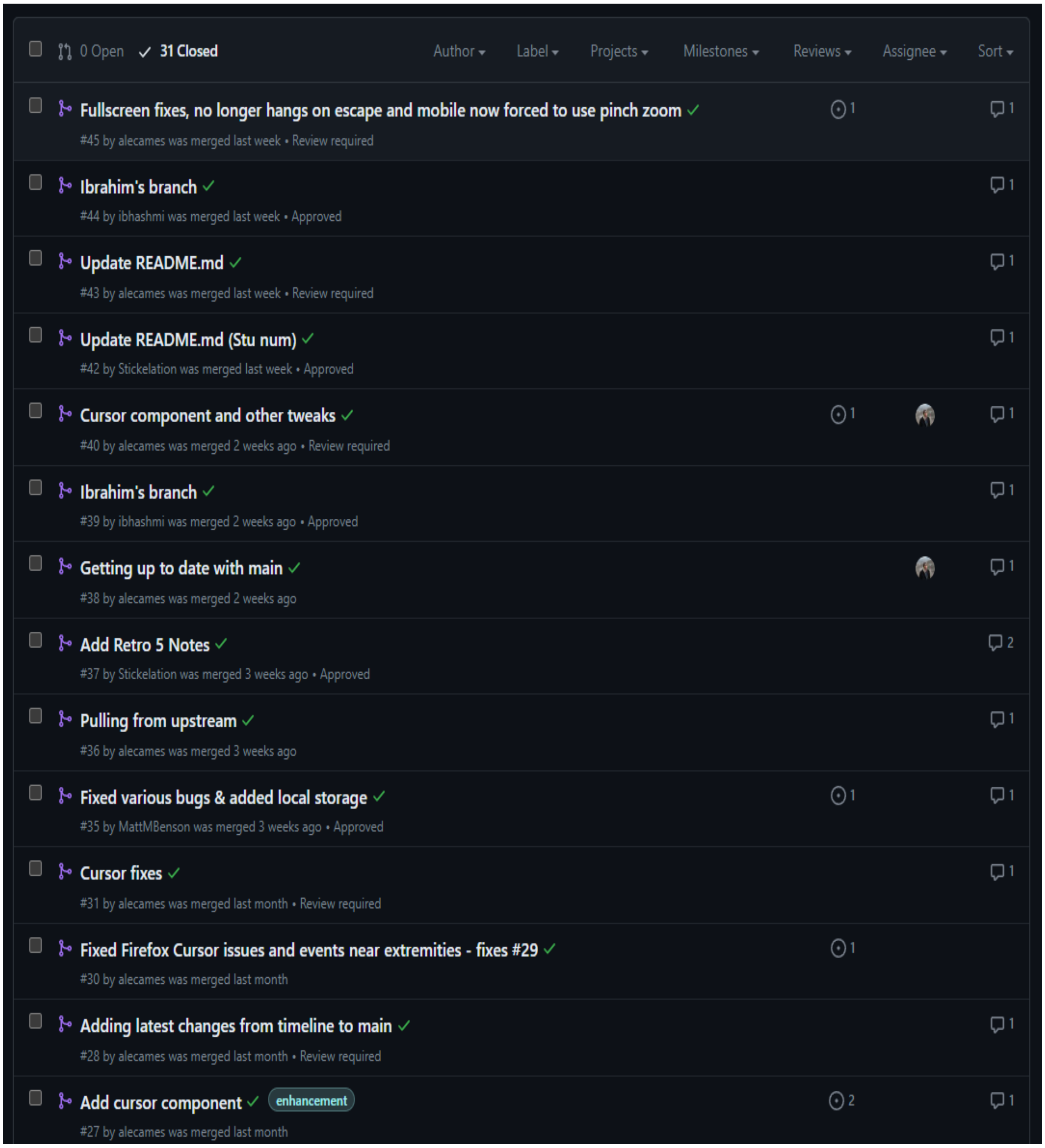
Ability to zoom in when in full screen `enhancement` `ux` — 1
#22 by alecames was closed last week

Themes do not persist on reload `bug` — 1
#19 by alecames was closed 3 weeks ago

Re-navigate to previous item on Save or Edit `enhancement` — 1
#18 by alecames was closed 2 weeks ago

Caution message for delete event `ux` — 1
#17 by monwe-jr was closed last month

Search bar fade
#15 by monwe-jr was closed on Mar 25



0 Open ✓ 31 Closed Author Label Projects Milestones Reviews Assignee Sort

Fullscreen fixes, no longer hangs on escape and mobile now forced to use pinch zoom ✓ 1 1
#45 by alecames was merged last week • Review required

Ibrahim's branch ✓ 1
#44 by ibhashmi was merged last week • Approved

Update README.md ✓ 1
#43 by alecames was merged last week • Review required

Update README.md (Stu num) ✓ 1
#42 by Stickelation was merged last week • Approved

Cursor component and other tweaks ✓ 1 1
#40 by alecames was merged 2 weeks ago • Review required

Ibrahim's branch ✓ 1
#39 by ibhashmi was merged 2 weeks ago • Approved

Getting up to date with main ✓ 1
#38 by alecames was merged 2 weeks ago

Add Retro 5 Notes ✓ 2
#37 by Stickelation was merged 3 weeks ago • Approved

Pulling from upstream ✓ 1
#36 by alecames was merged 3 weeks ago

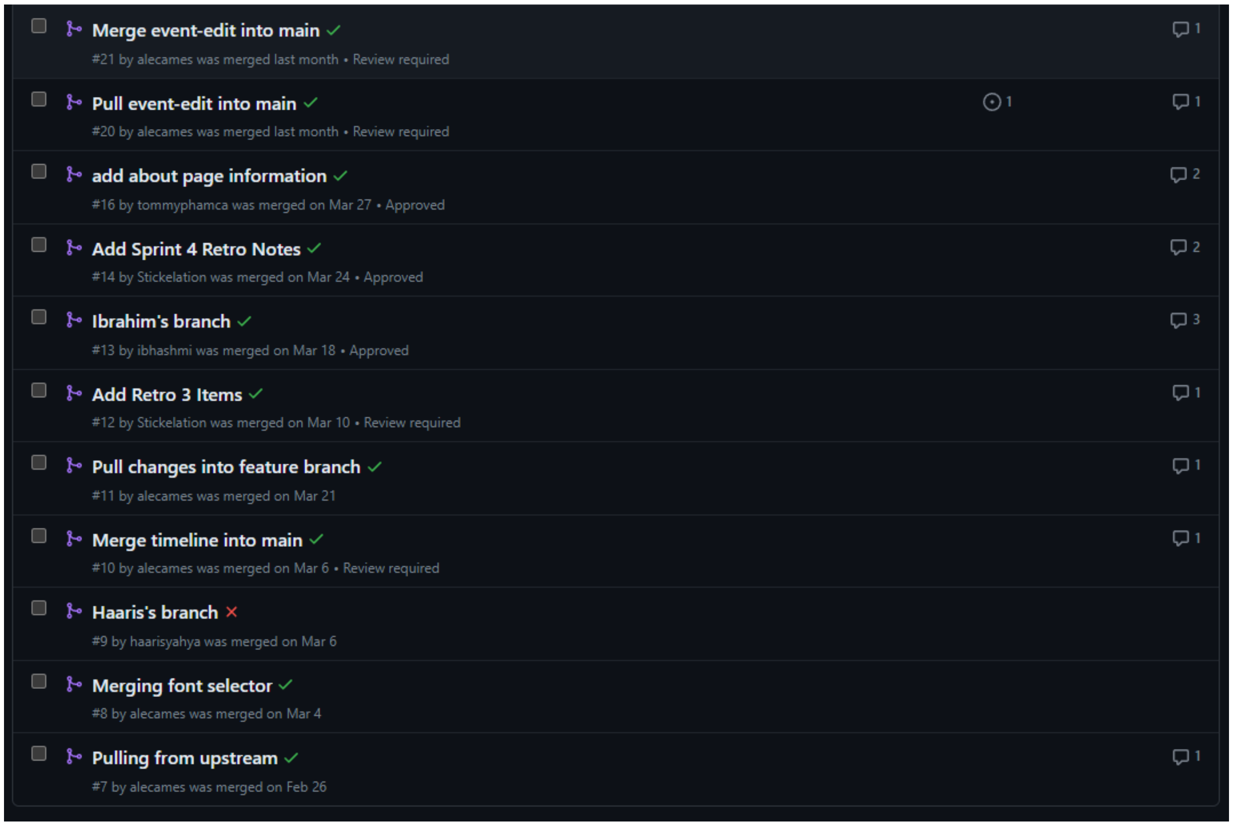
Fixed various bugs & added local storage ✓ 1 1
#35 by MattMBenson was merged 3 weeks ago • Approved

Cursor fixes ✓ 1
#31 by alecames was merged last month • Review required

Fixed Firefox Cursor issues and events near extremities - fixes #29 ✓ 1
#30 by alecames was merged last month

Adding latest changes from timeline to main ✓ 1
#28 by alecames was merged last month • Review required

Add cursor component ✓ enhancement 2 1
#27 by alecames was merged last month



Merge event-edit into main ✓ 1
#21 by alecames was merged last month • Review required

Pull event-edit into main ✓ 1 1
#20 by alecames was merged last month • Review required

add about page information ✓ 2
#16 by tommyphamca was merged on Mar 27 • Approved

Add Sprint 4 Retro Notes ✓ 2
#14 by Stickelation was merged on Mar 24 • Approved

Ibrahim's branch ✓ 3
#13 by ibhashmi was merged on Mar 18 • Approved

Add Retro 3 Items ✓ 1
#12 by Stickelation was merged on Mar 10 • Review required

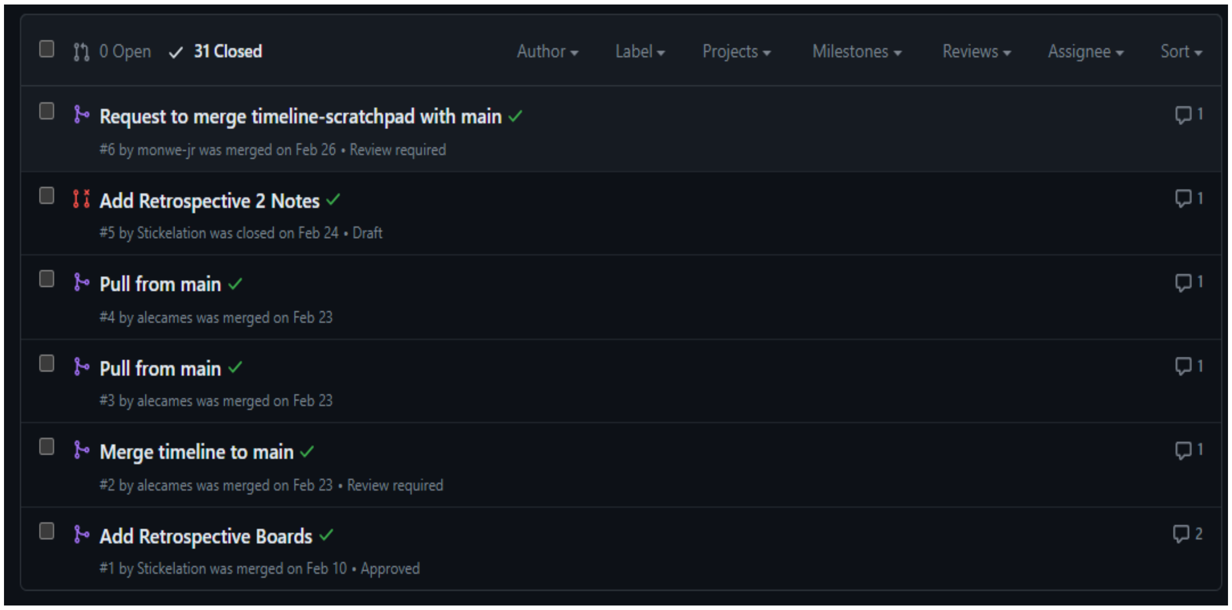
Pull changes into feature branch ✓ 1
#11 by alecames was merged on Mar 21

Merge timeline into main ✓ 1
#10 by alecames was merged on Mar 6 • Review required

Haaris's branch ✕
#9 by haarisyahya was merged on Mar 6

Merging font selector ✓
#8 by alecames was merged on Mar 4

Pulling from upstream ✓ 1
#7 by alecames was merged on Feb 26



0 Open ✓ 31 Closed Author Label Projects Milestones Reviews Assignee Sort

Request to merge timeline-scratchpad with main ✓ 1
#6 by monwe-jr was merged on Feb 26 • Review required

Add Retrospective 2 Notes ✓ 1
#5 by Stickelation was closed on Feb 24 • Draft

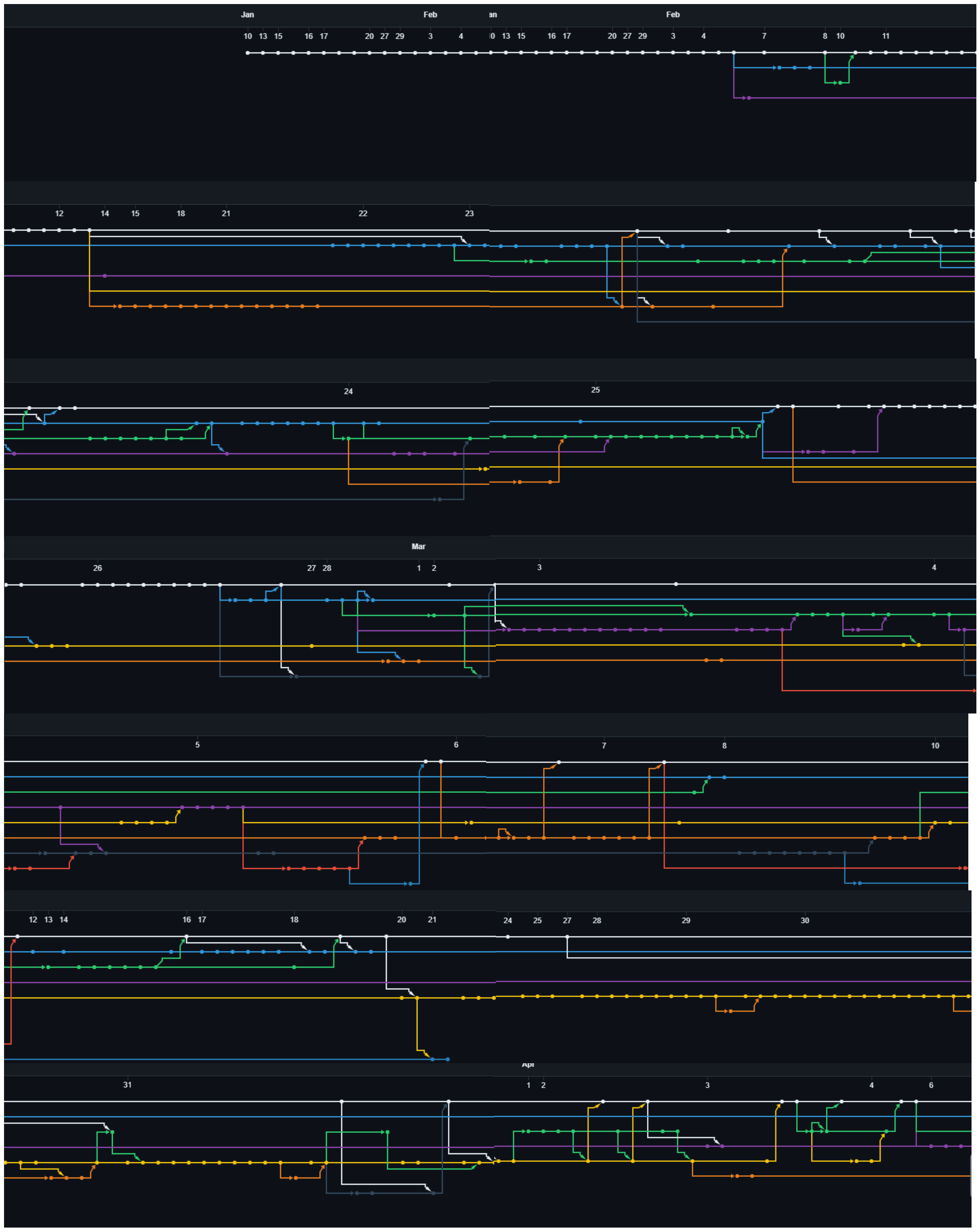
Pull from main ✓ 1
#4 by alecames was merged on Feb 23

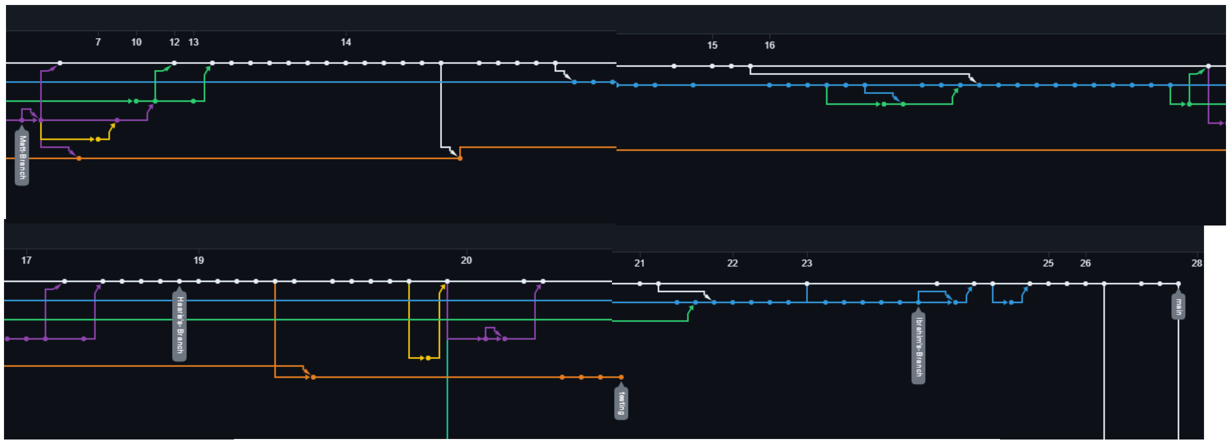
Pull from main ✓ 1
#3 by alecames was merged on Feb 23

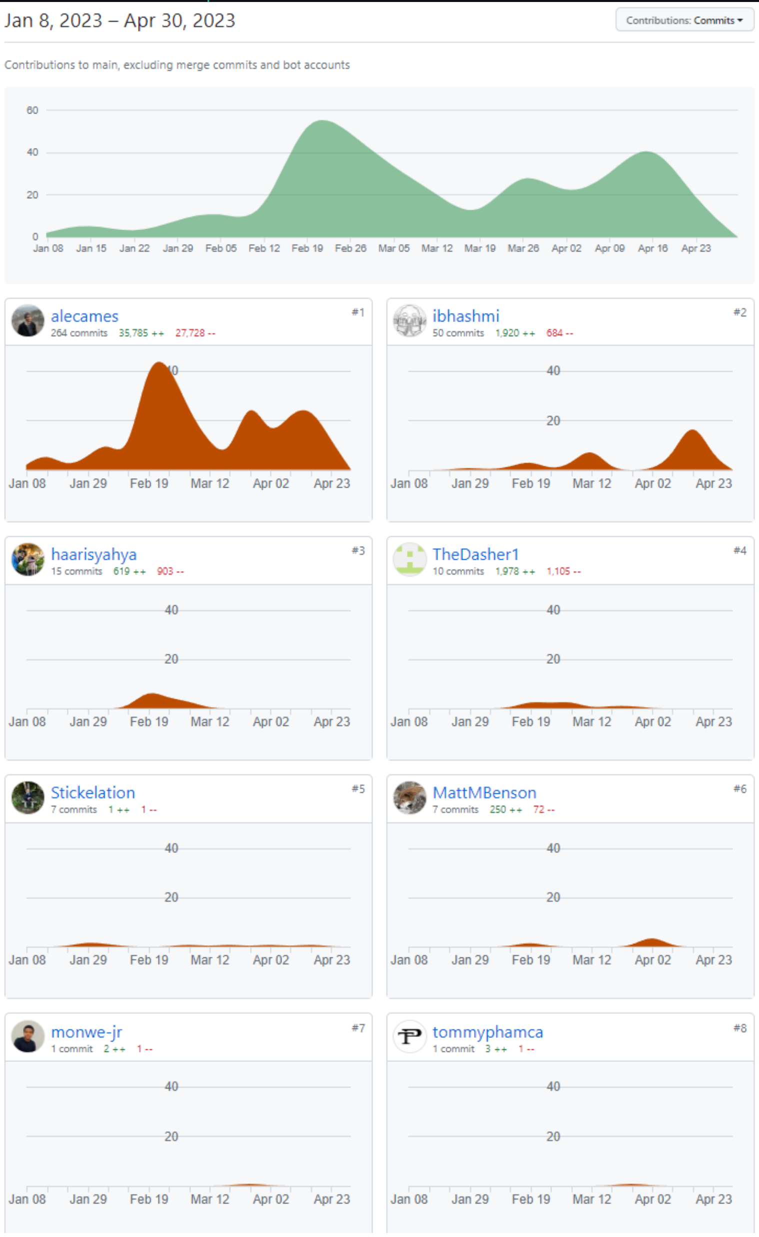
Merge timeline to main ✓ 1
#2 by alecames was merged on Feb 23 • Review required

Add Retrospective Boards ✓ 2
#1 by Stickelation was merged on Feb 10 • Approved

The following is the network graph of the overall group contributions throughout the project's development:






Jan 8, 2023 – Apr 30, 2023

Contributions to main, excluding merge commits and bot accounts

alecames #1
264 commits 35,785 ++ 27,728 --

ibhashmi #2
50 commits 1,920 ++ 684 --

haarisyahya #3
15 commits 619 ++ 903 --

TheDasher1 #4
10 commits 1,978 ++ 1,105 --

Stickelation #5
7 commits 1 ++ 1 --

MattMBenson #6
7 commits 250 ++ 72 --

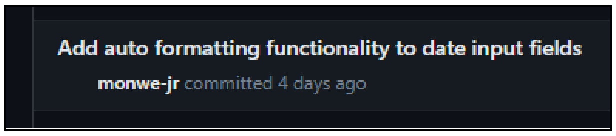
monwe-jr #7
1 commit 2 ++ 1 --

tommyphamca #8
1 commit 3 ++ 1 --

Github Usernames

Member	**Github Username**
Alec Ames	`alecames`
Matthew Benson	`MattMBenson`
Ibrahim Hashmi	`ibhashmi`
Francis Monwe	`monwe-jr`
Tommy Pham	`tommyphamca`
Abhijeet Prajapati	`TheDasher1`
Justin Stickel	`Stickelation`
Haaris Yahya	`haarisyahya`

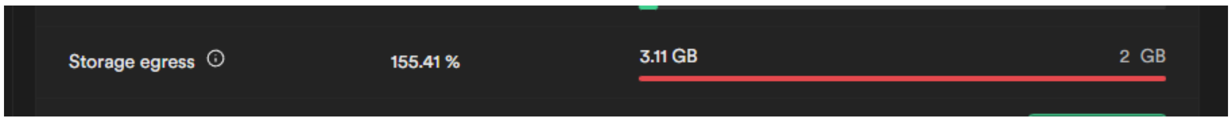
Note:
The GIT configuration for Francis' user was under a different email, resulting in all commits being authored under his username 'monwe-jr' but not linked with GitHub. He was able to fix this issue at the end of development, however, it did not retroactively link the prior commits to his account. This has the effect of GitHub reporting only the commits after the fix. Francis has contributed throughout the development process on his self titled branch shown as unlinked commits. For reference, the following screen capture is what the unlinked contributions look like.


Add auto formatting functionality to date input fields
monwe-jr committed 4 days ago

Additionally, this only shows commits to main, not to other active or experimental branches (such as testing and deprecated branches), nor does it capture contributions in the back-end. As mentioned prior, all members assumed fair responsibility in the completion of the project.

6. Known Issues

As we are on the free-tier of Supabase, we have limited storage egress. Near the end of April, we went over the usage limits. This did not cause any issues at the time as the images in the storage bucket were still loading on the application, however, this is an issue we will address in the case that there is a cut off point after higher egress or fees.



Storage egress 155.41 % 3.11 GB 2 GB

7. Links

- GitHub repository: https://github.com/SWE-2023/COSC-4P02-Project
- Vercel deployment: https://museumtimeline.vercel.app/timeline
- Google Play listing: https://play.google.com/store/apps/details?id=app.notltimeline.twa