

**Badger Tech Final Report**  
Interactive Timeline for Niagara on the Lake Museum



Sowmya Movva: Scrum Master

Rimpy Saha: Product Owner

Tyvon Factor-Gaymon: Product Owner

George Gramatikov: Tester

Peter Fung: Tester

April 30, 2023

## Table of Contents

<b>Section 1 – Introduction.....</b>	<b>4</b>
1.1 Objectives of this Document .....	4
1.2 Document Conventions .....	4
1.3 Intended Audience .....	4
<b>Section 2 – Scrum Process .....</b>	<b>5</b>
Section 2.1 Recurring Weekly Sprint Planning and Review Meetings.....	5
Section 2.2 Trello .....	6
<b>Section 3 – Requirements .....</b>	<b>8</b>
Section 3.1 – Initial requirements.....	8
Section 3.2 – Changing Requirements .....	8
Section 3.3 – Future Requirements .....	9
<b>Section 4 – Product Description.....</b>	<b>9</b>
4.1 Product Prospective .....	10
4.2 Product Features.....	10
Section 4.3 – Final Product.....	11
Section 4.4 – Final Product.....	11
4.4 Features created but not Implemented. ....	12
<b>Section 5: System Architecture.....</b>	<b>14</b>
Section 5.1 – Architecture Introduction.....	14
Section 5.2 - Architecture Overview .....	14
<b>Section 6 – Database .....</b>	<b>16</b>
Section 6.1 - Database Introduction .....	16
Section 6.2 - Database Design .....	16
Section 6.3 – Data Access .....	16
Section 6.4 – Database Maintenance .....	17
<b>Section 7 – Product Limitations and Challenges .....</b>	<b>17</b>
Section 7.1 – Software Reuse .....	17
Section 7.2 – Development Team Issues.....	17
Section 7.3 – Hosting.....	18
Section 7.4 – Loading Times .....	18
<b>Section 8 - Research and Data for a Historical Timeline.....</b>	<b>19</b>



<b>Section 8.1 – Research Methodology .....</b>	<b>19</b>
<b>Section 8.2 - Purpose of Research .....</b>	<b>19</b>
<b>Section 9 – User Interface .....</b>	<b>19</b>
<b>    Section 9.1 - Graphics Developed Using Figma .....</b>	<b>20</b>
<b>    Section 9.2 – Consistency Across Pages .....</b>	<b>21</b>
<b>    Section 9.3 – Timeline User Interface .....</b>	<b>22</b>
<b>    Section 9.4 – Museum Logo Redesign.....</b>	<b>23</b>
<b>Section 10 - Testing.....</b>	<b>24</b>
<b>    Section 10.1 – Program Testing .....</b>	<b>24</b>
<b>Section 11 - User Manual For the Website .....</b>	<b>27</b>
<b>Section 12 - GitHub Logs .....</b>	<b>28</b>

## Section 1 – Introduction

This document will observe requirements of various stakeholders for the interactive timeline project that is being developed for the Niagara on The Lake Museum. Museums has been struggling to attract a younger audience and engage viewers in digital spaces thus the use of interactivity will serve as a great tool to appeal to larger demographics. This document will discuss the details of the interactive timeline application developed as well as various aspects of the software development process.

### 1.1 Objectives of this Document

The main objective of this document is to observe the technical details of the development process and how the product was developed. We will be noting how the product evolved throughout the sprint cycles and modifications that were made to the scrum process as a result. Furthermore, the document notes rationale behind decisions and development stages of the projects as the team progresses through sprints. This document also includes code used in the creation of certain features and provides insight into the architecture of the application.

### 1.2 Document Conventions

The document is structured according to sections which are further broken down into subsections. The main sections of the report are introduction, insight into the scrum and development process, the final product, system architecture, database management system, research, and user interface design. Contributions and GitHub logs for individual team member contribution will be placed at the end of the document. Additional resources and links to supplementary material such as code and research will be provided throughout the document.

### 1.3 Intended Audience

This document is intended for stakeholders involved with the development of the Niagara on The Lake Museum Interactive Timeline Project. Relevant stakeholders include the museum staff and administration, software developers and partner organizations. Additionally, students enrolled in COSC 4P02 and the instructors are also the intended audience of this document.

## Section 2 – Scrum Process

### Section 2.1 Recurring Weekly Sprint Planning and Review Meetings

The team selected one-week sprints with two weekly sprint meetings with sprint planning meetings taking place on Wednesday and sprint review meetings taking place on Saturday.

The sprint cycles are made available to everyone in the GitHub ReadMe, Project Proposal and on Trello. Despite the short time frame in between meetings, this allowed for maximized communication and transparency regarding progress as well as allowed for the team to address concerns in a timely manner so that other team members are aware if tasks are not being completed within the sprint.

Recurring meeting invites were sent out via teams and with the following descriptions. Additionally, an agenda of action items is also shared every meeting either on a word document, Trello or through the details sections of the meeting invite. Additional communications take place on teams throughout the sprint where additional calls may be required when collaborating on certain tasks. As a result, team members are asked to check teams daily for updates.

The screenshot shows a Microsoft Teams meeting invite for a "Sprint Planning Meeting". The invite details are as follows:

- Date: April 5, 2023, 6:00 p.m. - 6:30 p.m.
- Meeting description: Recurring Sprint Planning Meeting every Wednesday at 6:00pm.
  - Please let team members know in advance if you cannot attend before 12pm so that we can try and reschedule.
  - Meeting agenda will be shared on the sprint board for a given sprint directly on Trello.
  - Please check Trello prior to attending meetings so that we can see what the current sprint looks like and the user stories that have been added.
  - Checking Trello beforehand will be required so that we can facilitate sprint planning.
- Join options:
  - Join on your computer, mobile app or room device
  - Click here to join the meeting
  - Meeting ID: 281 238 549 030
  - Passcode: QYJr33
  - Download Teams | Join on the web
- Join with a video conferencing device:
  - 172330346@teams.bjn.vc
  - Video Conference ID: 112 973 108 7
  - Alternate VTC instructions

On the right side, there is a "Tracking" panel showing the status of invitees:

- Sowmya Movva (Organizer): Accepted
- Rimpy Saha: Accepted
- Tyvon Factor-Gaymon: Accepted
- Peter Fung: No response
- George Gramatikov: No response

**Sprint Review Meeting** Chat Files **Details** Scheduling Assistant Attendance Meeting Whiteboard Breakout rooms Q&A +

Cancel meeting | Copy link Show as: Busy Category: None Time zone: (UTC-05:00) Eastern Time (US & Canada) ...

April 22, 2023 2:30 p.m. - 3:00 p.m. Edit series | Show meeting info

Quick sprint review (15mins) additional time allocated if necessary to address additional concerns

**Microsoft Teams meeting**

Join on your computer, mobile app or room device  
[Click here to join the meeting](#)

Meeting ID: 285 543 664 106  
Passcode: jgxTKB  
[Download Teams](#) | [Join on the web](#)

Join with a video conferencing device  
172330346@teams.bjn.vc  
Video Conference ID: 114 229 187 0  
[Alternate VTC instructions](#)

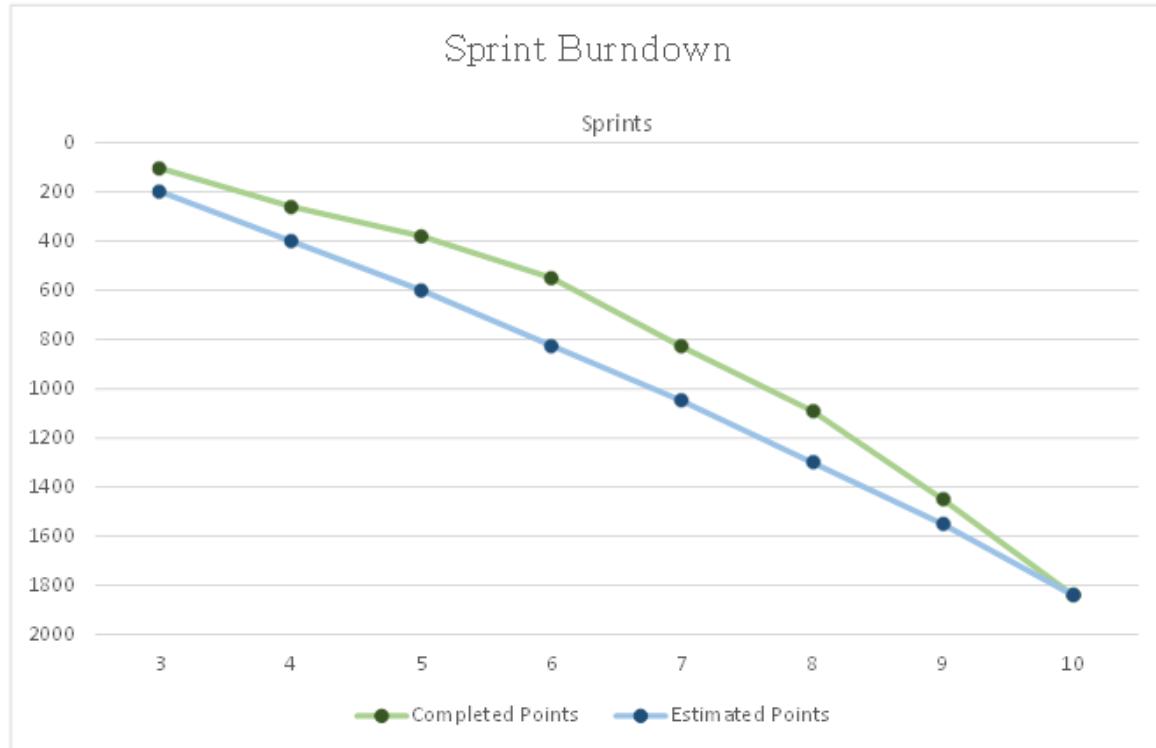
**Tracking**

Profile	Name	Status
	Sowmya Movva	Organizer
	Rimpy Saha	Accepted
	Tyvon Factor-Gaymon	Accepted
	George Gramatikov	Unknown
	Peter Fung	Unknown

## Section 2.2 Trello

Trello had at least 2-3 sprints being made in advance containing rough user stories for a given sprint. Once the sprint officially begins, the team reviews user stories on Trello and modifies them according to the revised product backlog on Trello. Once the team has discussed their tasks and roles for the sprint, they are responsible for updating the tasks that they are working on and have completed on Trello so that other group members are aware of their contributions.

There were initial issues some team members struggling to follow the scrum methodology and complete user stories on time. Additionally, following the product backlog to get a working product up for the first few sprints also posed as a challenge for some. Thus, Trello has served as a great tool for tracking contributions and what tasks a given team member is working on during a sprint. Trello contributions for all ten sprints have been documented and stored on a csv file and uploaded in the documentation section of GitHub titled ‘Trello Contributions for All Sprints.’



The Sprint Burndown figure shows our burndown chart for the last seven sprints. The full burndown chart document is uploaded on the main branch of GitHub in the documentation folder titled ‘Sprint Burndown’.

## Section 3 – Requirements

Requirements for the interactive timeline have changed over the course of the scrum process due to demonstrations and user testing which revealed discrepancies in our product, and we aimed to fill the gaps that we found.

### Section 3.1 – Initial requirements

Initial requirements were feature heavy and the team created many user stories which resulted in a feature heavy product. Many sub-systems were re-evaluated since earlier sprints such as booking, payment and user interactions systems to reduce unrealistic development goals and refocus on maximizing interactivity with timeline.

### Section 3.2 – Changing Requirements

#### Organization and Clean Website Structure

were determined to be a top priority to allow for maintainability and evolution for future changing requirements. This impacted the requirements and lead to the modification of some existing modifications. Though after initial backlog revision in progress 2, no further requirements were removed, a few requirements were identified and added to the sprints. Previously, we kept some features assigned as low priority which were moved over to the last sprints to be developed such as editor features and search functionality.

#### User Interface

Interface requirements were modified to accommodate responsive web design whereas the initial sprints focused on web-based applications. Although, the timeline requires additional modifications for optimal viewing on smaller screen sizes, the responsive web design framework allows for future implementation of an optimized mobile accessible timeline.

#### Database

Database requirements were changed to allow for the implementation of additional features due to changing requirements that allow for increased interactivity such as the more information sections, speech to text and filters. These features required additional queries and modification of tables which will be discussed in section 6 of this document.

#### Speech to Text

This feature was thought to provide increased interactivity as the viewer can click on the button for any event and rather than read, has the ability to hear the text displayed. This feature provides useful for any viewer with accessibility concerns but also allows all users to interact with the data in an audio-visual capacity.

### More Information

This feature was identified during user testing where the timeline application received positive feedback, but viewers found that the page in which the timeline was located appeared empty. Thus, to make use of the extensive data that has been collected and mitigate this issue, a section beneath the timeline was created to display further information for an event supplementary to the summary that viewers see on the event graphic.

### Branding

A well-organized website can help to establish a strong brand identity and voice. This can help to build credibility with visitors and ultimately drive conversation by engaging a wider audience. Thus, a complete website redesign with a consistent theme and logo were created.

## Section 3.3 – Future Requirements

Mobile accessibility will need to be optimized as this application is mainly intended for web users. While the timeline works on a mobile device, the user interface needs to be redesigned and refined. We understand that in order to capture a wider audience, we will have to cater to the smart phone demographic as a large majority of internet users prefer to browse the web on mobile devices. Additionally, vertical scrolling toggle will also need to be created to allow for ease of interaction. Features that were created but the team did not have the resources to implement during this software release are outlined in section 4.4. These features also realized many requirements that do not reflect on our current release but demonstrate that the developers did identify more user requirements.

## Section 4 – Product Description

A timeline is a graphical tool that displays a chronological sequence of historical events by year. These events are typically represented older to newer events from left to right. The traditional representation of the timeline involves a horizontal line with years increasing as the user reads from left to right. An interactive timeline web application allows the users to directly interact through a series of operations such as click scroll, click to expand and filter to limit events. They can scroll from left to right to see events through the years. The user is able to view on average four events at a time due to the large volumes of data present and to not overwhelm the user especially given that the application uses horizontal scrolling. In addition, an interactive web application will allow for multiple users to access a timeline at any time of the day without having to physically visit a museum.

## 4.1 Product Prospective

The interactive timeline is currently hosted on InfinityFree in order to test and develop without interfering with the existing live museum site. The product also features a complete website redesign to implement both a historical and modern theme that compliments the addition of an interactive timeline. The database also features updates to allow for various filtering options, there have been additional modifications to existing database was difficult to search and filter. The standalone web application is meant to be integrated with the current website. The database has been updated to allow for additional features such as edit content for staff, the more information section underneath the timeline which corresponds to a given event and the speech to text feature.

## 4.2 Product Features

Feature	Description
Horizontal scrolling timeline	Can only view on average 4 images at a time to not overwhelm the user.
View Event	Click on an event to see a photo of an event and information regarding the event. Allows user to control what event they see at a given time.
Event comparison	can click numerous events which remain clicked until user clicks again to close. Allows user to control the amount of data they are viewing to not overwhelm the user.
More information section	certain events may have more information than what is presented on the card and appears directly below the timeline
Text to speech	Viewers have the option to have the information read to them
Filter by category	events are classified into six main categories; user can click on the category icon and see events in that category
Filter by range	viewers can select minimum and maximum years (to and from) to see events within a specific time frame
Dark mode	users can select dark mode if they do not like the default theme
Login Feature	Users can sign up and login
Editor Features	Certain users that are staff will receive special permissions to be able to edit the timeline without having to modify the backend.

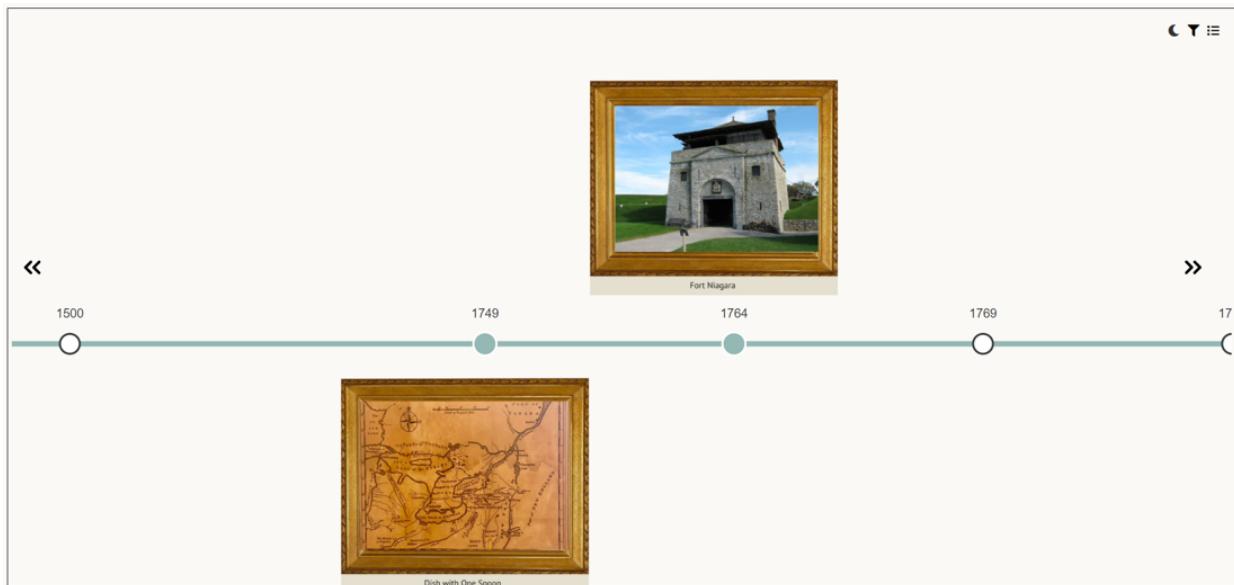


	<ol style="list-style-type: none"> <li>1. View Button: appears on edit page to redirect staff to timeline to view changes after they have been made</li> <li>2. Edit Button: appears on timeline page when staff views the timeline to make edits</li> </ol>
Forgot password	Users can verify their email and receive a password reset option in their email
Password reset	Users can click on the link in the email which prompts them to change their password

### Section 4.3 – Final Product

The group chose to create a dynamic timeline, but a static timeline may have been sufficient as the domain of the application does not require changing or adding ongoing events for the time being. However, dynamic allows for increased interaction though it was not implemented thoroughly during the duration of the project as users are not directly changing events, information, or commenting. Although dynamic is more complex and takes longer to render on the website, it allows for staff to edit the contents without having to access the database or the backend. In addition, providing the user with filtering capabilities.

### Section 4.4 – Final Product



This figure depicts the timeline viewbox containing the horizontally scrolling timeline with clickable events. The filter features can be seen on the top right-hand corner and the comparison feature can also be seen in this figure. Further product demonstrations can be viewed in the UI/UX section of the report.

#### [4.4 Features created but not Implemented.](#)

##### Transition feature for cards

Transition would have required the graphical elements to be updated but requires significant time and was implemented too close to the release date. Updating all graphical elements requires modifying in an external software, Figma, which then requires exporting as JPEG files and modification of the resolution. Creating graphical elements took over two weeks and changing the background of the software would require that we remove the dark mode feature thus this feature was not implemented in the final product.

##### Hover scroll

This feature was too reactive and moved very fast when slightly moving the cursor. The combined issues resulted in an accessibility issue and was deemed unusable as at times, the scroll was so fast that users could not stop to select an event. The fix would have taken time to modify and requires complex modifications to the code. Ultimately, it was replaced with click scroll due to usability concerns and the time required to fix it.

##### Calendar feature

The calendar was difficult to justify as to why it was on the timeline and design inconsistencies were fixed too close to release date. Given there being too many bugs to fix on the application some features could not be implemented. In the future, the calendar feature will be useful if the museum were to implement a booking feature as it would allow guests to book the time and date of visit as well as choose the number of guests that will be visiting.

##### Comprehensive Search

Users would be able to type anything into the search and are taken to a results page with options or if no options are available then the page generates a 'no results' message. Unfortunately, this required complex code and given time constraints, the search feature was limited to auto populated timeline search options.

### Category Signifiers

All categories in the drop-down menu would have an associated icon which would also appear on the back of every graphical element next to the more information and text-to-speech icons to provide the user with a visual cue of what information they are viewing. Due to time constraints and complexity of placing a row of icons on the back of an event card with changing positions, this feature was dismissed as low priority over debugging.

### Mobile Vertical Timeline

To create a mobile accessible timeline, we needed to modify the timeline to be vertically scrolling. Due to time constraints and programming complexity, we decided to move this task to a low priority task to be taken up if all other tasks were finished.

### Toggleable Timelines (Horizontal/Vertical)

The reason for removing the toggleable timelines feature was the same reason the vertical timeline for mobile devices were removed. Vertically scrolling.

### Embedded Videos

An option to embed videos directly into a given event on the timeline was also developed however due to a lack of resources we were not able to upload relevant videos from the museum on related topics. Additionally, early museum research conducted involved having a staff member ask us to contact them at a later date but due to the large volume of data requests by Brock Students, the museum was no longer able to provide us with additional data thus we were limited to text and images.

### Complete redesign of website

Given the constraints we experienced in the first few sprints; a complete redesign of the website became a low priority task. The team focused on additions to the website rather than designing a new website. Initial website prototypes developed by Peter Fung were not implemented as this was outside of the domain of the project and would require additional development time given the scale of the existing museum website.

### Filter Reset Option

Ideally when applying filters to the timeline, a user should have the ability to click a button to reset applied filters and see the full version of the timeline without having to reload the page. However, the original range can be reset manually so this is mainly an issue with the category filter.

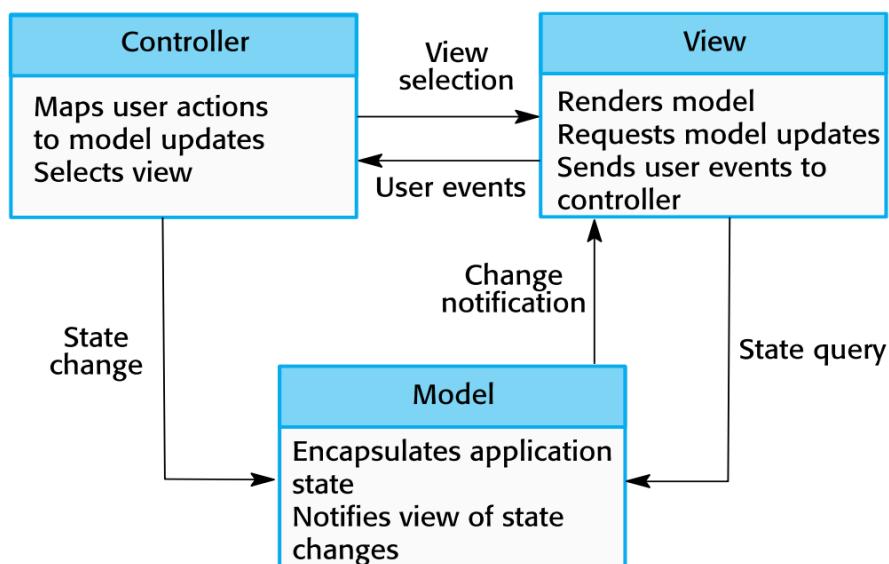
## Section 5: System Architecture

### Section 5.1 – Architecture Introduction

The interactive timeline application involves large amounts of organized data that is directly displayed on the timeline that user can interact with. User interaction and data handling is not only limited to displaying the timeline to viewers but also displaying timelines with real time modified features. This was found to be the best architecture for code reusability in the long run due to the separation of code offered by this architecture.

### Section 5.2 - Architecture Overview

The interactive timeline utilises The Model View Controller architecture. The system is separated into three interactive components, the controller, view and the model. The purpose of this architectural framework is to separate the data from what the user sees and interacts with. The model is responsible for managing the data and any operations performed on the data. The view component manages how the data is presented to the user. The controller manages interactivity and sends information regarding interactivity to the model and view components.



Although this architecture is generally regarded as complicated for smaller scale applications and adds further complexity to the code, it was deemed as the most suitable architecture as it allows for a higher degree of flexibility for client customization and software evolution. The strong foundation established with this system architecture will ensure that the application can be developed further in coming years. We ensured that software maintainability was a priority from a software development perspective.

### Section 5.3 – Implementation

The folder structure containing the system architecture can be seen in the figures below and are also available on GitHub.

/htdocs/public_html/*.*	
Name	Size
..	
views	
sendemail	
models	
includes	
controllers	
config	
assets	
index.php	5 KB

/htdocs/public_html/sendemail/	
Name	Size
..	
src	
language	
VERSION	1 KB
SECURITY.md	8 KB
README.md	17 KB
LICENSE	26 KB
get_oauth_token.php	7 KB
composer.json	8 KB
COMMITMENT	3 KB

/htdocs/public_html/assets/	
Name	Size
..	
js	
images	
CSS	

/htdocs/public_html/models/	
Name	Size
..	
UserModel.php	9 KB
ContentModel.php	2 KB

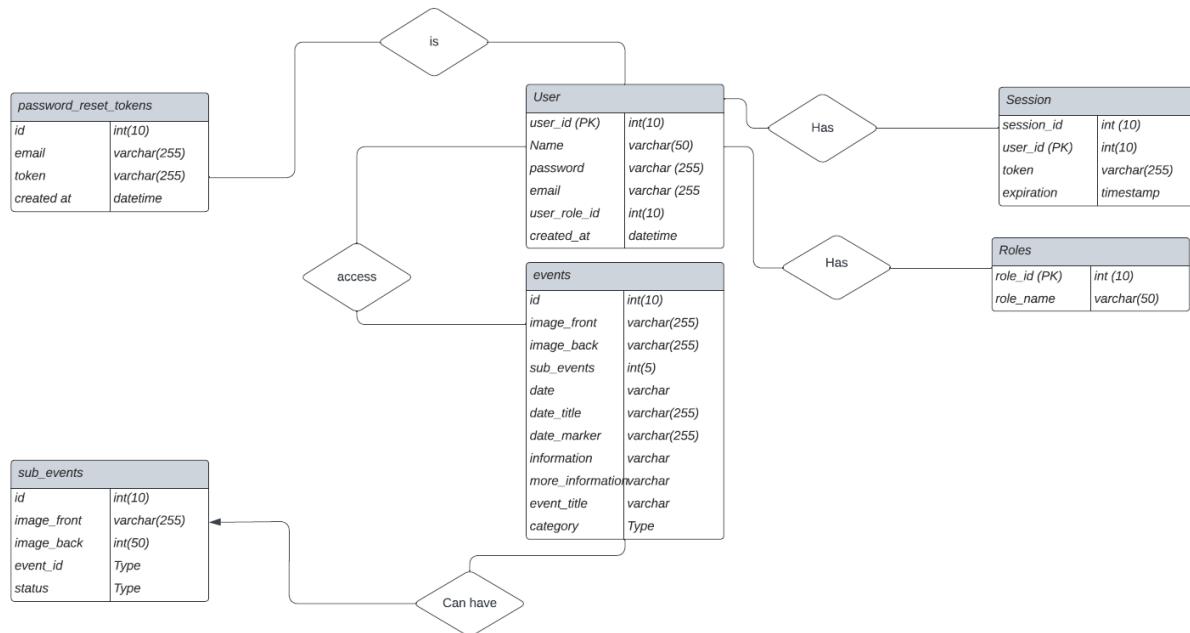
/htdocs/public_html/config/	
Name	Size
..	
config.php	2 KB

## Section 6 – Database

### Section 6.1 - Database Introduction

The database was developed using PHPMyAdmin and the entity relationship significantly changed as initially the database was set up for individual content pages with page types. This was modified to better frame our product so not instead of page types and contents, we have an events and subevent table that are related based on the event id. This allows for easy tracking of all information regarding any event or sub-event. The other change made was to user. With role hierarchy being much simpler now to only registered users, admins and guests I was able to get rid of the permissions table and instead include a password\_reset\_token table to keep track of unique random tokens users are sent for email verification.

### Section 6.2 - Database Design



### Section 6.3 – Data Access

With the change database structure, the software was not accessing rows of subevents for an event and instead it corresponds to combined string that it later splits into relevant information in turn simplifying the SQL as it did not require any nested queries or joined to match all conditions. Instead, the software is able to use readily available php and js functions to access relevant information.

The software is now able to provide user and role functionalities like login, signup, logout, profile and reset or forget your password. This modified every query as it determines the level of access for each user based on their information. The user information is now being accessed in addition. This resulted in a whole model controller structure for the user pages implying that new queries were implemented for all the functionalities.

## Section 6.4 – Database Maintenance

The Database was backed up every sprint and before every information import. It was constantly modified as the product requirements got updated throughout the scrum process to be able to handle the amount, type and variation in content being stored.

# Section 7 – Product Limitations and Challenges

## Section 7.1 – Software Reuse

Software reuse was not considered for the development of majority of the application thus created additional hurdles for developers as most of the source code was developed from scratch. This increased complexity and limited the amount of time that the team spent focusing on usability and product features.

## Section 7.2 – Development Team Issues

Some members of the team struggled with the scrum process which resulted in developmental delays and ultimately resulted in increased complexity in terms of the code that was developed in conjunction with software reuse not being the primary development method. The time constraints also created software refinement challenges prior to the final release date with minor refinements still being required. The challenges experienced early on modified the product backlog and placed an emphasis on the importance of revising the backlog every sprint. Some developmental tasks took longer than others and as such needed to be reflected in the sprint as larger tasks are harder to complete within one sprint. Larger tasks such as creating the timeline or data that are ongoing throughout the software development cycle.

In addition, the team was focused on a feature heavy timeline rather than user experience. There are many useful features such as editor features for staff, filtering and comparing events for users but ultimately the final product needed work on the user experience side in terms of having a title and additional signifiers. However, with the tutorial overlay appearing for new

user and existing signifiers, the timeline was still a quality product despite requiring refinements.

### Section 7.3 – Hosting

InfinityFree was used to host the website and as with any free hosting sites, there are limitations. There were downtimes which affected the deployment of the software despite all functionalities working the previous day. This can pose as an issue if the software were to be hosted on InfinityFree after being approved for client use. Additionally, there are also daily hit limits which can restrict website traffic by limiting the number of visitors per day. An SSL certificate was not provided by the platform and had to be included by the development team nor were backups included which posed as a threat when experiencing downtimes. The platform was great during initial stages of the development process with minimal features and a basic timeline however we noticed that into later stages of the development changes, certain program fragments were lost if another team member had the same file open simultaneously resulted in a large number of avoidable bugs.

### Section 7.4 – Loading Times

Page loading times also posed as an additional challenge given that we did not have access to funding or resources to mitigate this issue prior to demonstration. This also required optimal hosting solutions that are not free with minimal redirects and additional funding and resources to implement the storage of timeline graphics in cloud storage to further reduce loading times. Furthermore, the loading would have been further reduced for our css, JavaScript and php was minified which required further resources and time.

## Section 8 - Research and Data for a Historical Timeline

### Section 8.1 – Research Methodology

The research methodology included iteratively collecting data from the museum by organizing exhibits into time period such as indigenous history, 1700s, 1800s, and 1900s. Additionally, all data gathered was compared to the online museum database to obtain images of artifacts for the creation of graphics that are displayed on the timeline.

### Section 8.2 - Purpose of Research

The main purpose of providing accurate and complete data is for improved understanding, contextualization, and a learning tool.

Organizing a large amount of data into a chronological timeline can help people to better understand the relationships between events and how they led to historical developments. It can provide a clear and concise overview of important events and their impact and help people to see patterns and connections that they might not have noticed otherwise.

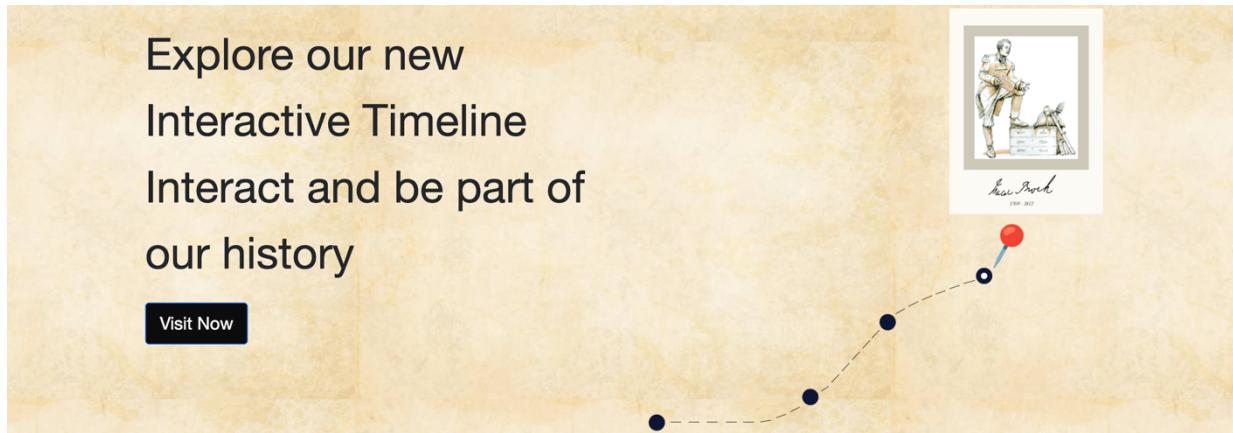
By organizing data into a historical timeline, it can be easier to contextualize events within their historical context. This can help people to better understand why certain events occurred, and how they were related to other events happening at the time. It can also help to provide a sense of continuity across time, and how events from different time periods are related to each other.

It allows for easier analysis and interpretation of historical events. By seeing events in chronological order, viewers can identify patterns and changes over time. It also provides insights into how events and developments have shaped the world presently. powerful tool for presentations and educational purposes. By presenting information in a clear and organized manner, people can better understand and retain information about historical events.

Overall, organizing a large volume of data into a historical interactive timeline can be a valuable tool for understanding and presenting historical events and developments. It can provide additional context and insights for the user that may not be possible for other forms of data already present in exhibits.

## Section 9 – User Interface

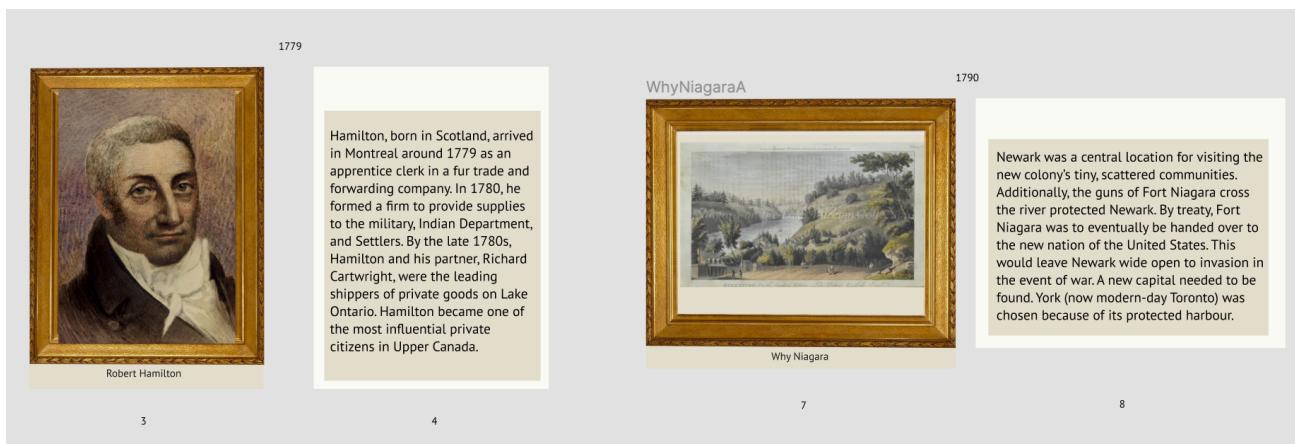
## Section 9.1 - Graphics Developed Using Figma



All graphics developed using figma follow a consistent clean and historic theme that represents the perfect mix of modern and historical to grab the user's attention. The landing page featured graphics designed using figma which served as a teaser version of the timeline to grab the attention of a website visitor and encourage to use the application. Using appealing visuals is an important part of attracting users and generating more interest in the museum.

### Timeline Graphics

The updated graphics for that depict the events on the timeline feature more visually appealing elements such as a gold frame which also provide the user with a historical feel thus contributing to the theme of a historical timeline.



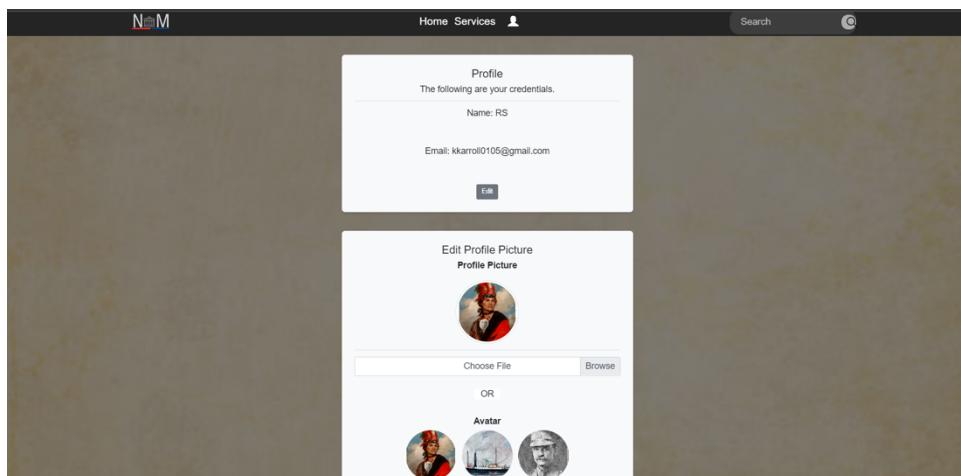
### Sub-Timeline Graphics

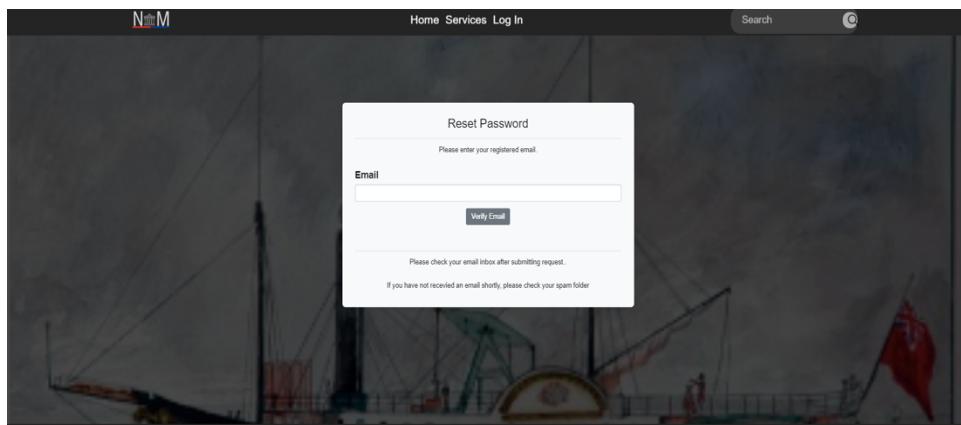
Sub-timeline graphics featured a similar vintage photograph design as seen in the progress 2 demonstration. The simplistic design was a preferred choice given the purpose of the sub-timeline and to indicate to the user that they are in a sub-component of the application.



## Section 9.2 – Consistency Across Pages

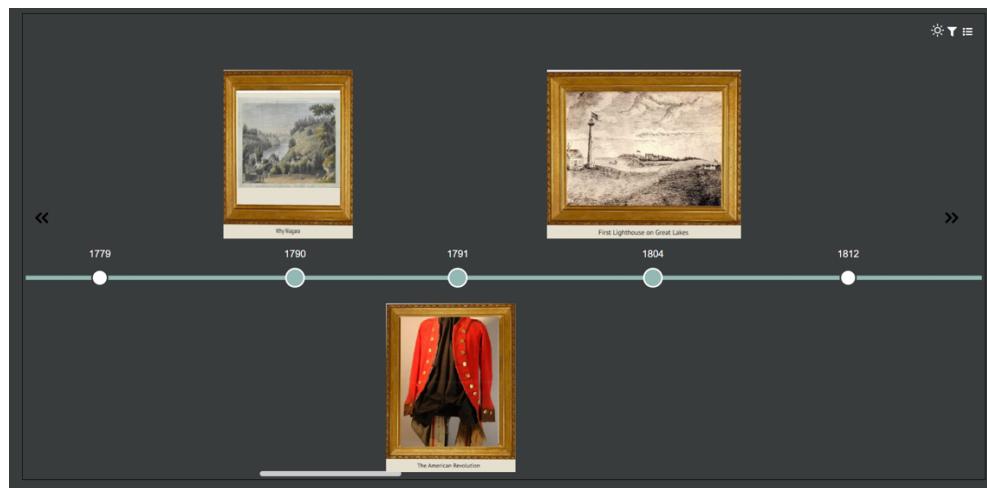
The webpages pages have a consistent theme across all pages which feature an older paper background and a historical Niagara ship painting which have a gradient filter applied. All pages have been designed using bootstrap to maintain further consistency and feature similar forms and buttons implemented throughout the page. Additionally, minimal variation across website themes and colour scheme also helped with maintaining the similar design elements.





### Section 9.3 – Timeline User Interface

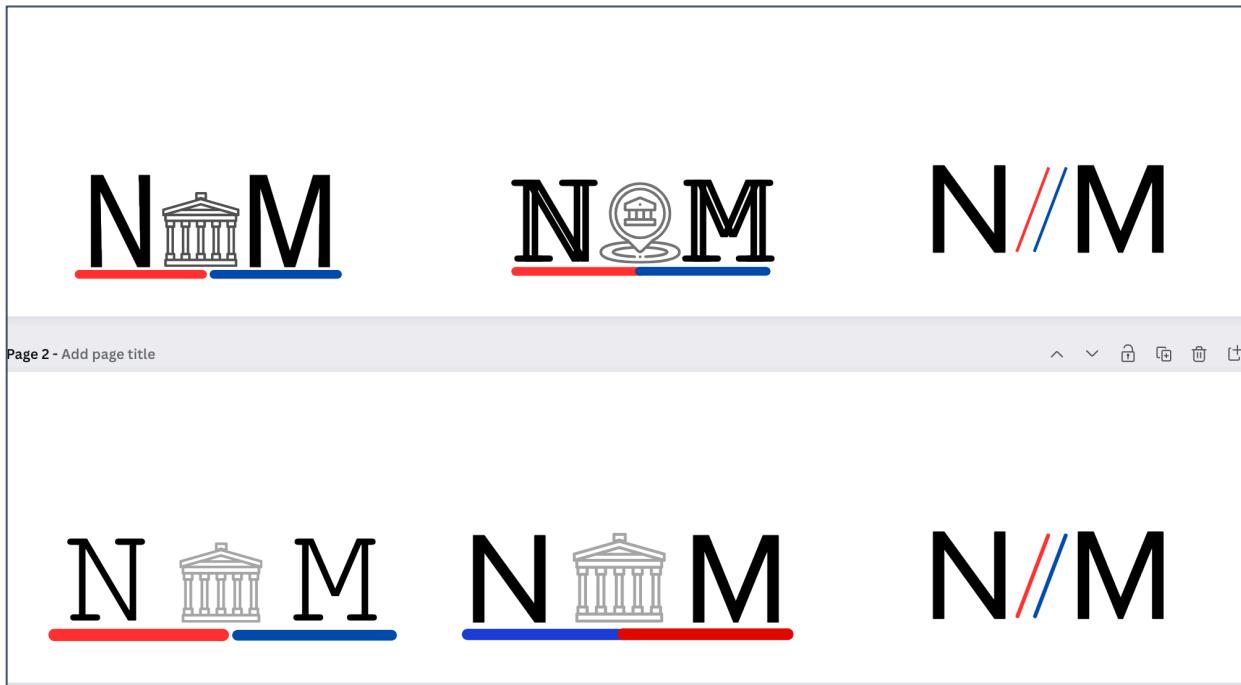
The timeline features many consistent and modern design elements that are visually pleasing. The top right-hand side of the timeline view box contains filter options and one of the most important features for a younger demographic is the dark mode feature. Users can toggle between dark and light mode using font awesome icon signifiers which indicate to the user that the timeline has dark mode functionality. In addition, the filter and category icons used are modern and commonly used in many web application thus indicating to the user that there are filter options present on the timeline.



Moreover, the timeline events feature modern buttons that resemble radio buttons which are a widely used clickable button that many users are familiar with which indicates to the user that they are clickable when they first enter the timeline page. Upon clicking, the buttons fill with the same light pastel background as the timeline replacing the white background and they explain to show the viewer that something has been clicked and they have to click again for the event to go away.

## Section 9.4 – Museum Logo Redesign

To ensure that the client receives a complete product, the website was redesigned, and new logos were created. Providing a website redesign ensured that the vision for the timeline remained consistent throughout the page. The original integrity of the logo remained unchanged with the original colour scheme being implemented into the new design. The team was presented with a few choices but ultimately chose the logo featured on the top left as it contains both modern and historic elements.



## Section 10 - Testing

During development, the development team conducted various types of testing which consisted of mostly manually testing continuously as new features are being developed and integrated as well as visual inspection. In addition, other testing methods were employed through various stages of development as described below.

### Section 10.1 – Program Testing

Program Testing is intended to show that a program does what it is intended to do and to discover program defects before it is put into use. We tested this manually using artificial data for filters and any input fields such as log in and sign up.

Results were checked for anomalies or errors. If tests received a passing result, they were further tested during various phases of integration for component and interaction testing. Majority of the testing was part of the verification and validation process. Given that Peter Fung and George Gramitkov were tasked with automated testing and continuous bug detection, developers did not always note down all bugs as they were integration-based errors.

#### Main Program Testing Discoveries

1. Responsive web design was not functioning as intended nor displaying correctly

Solution: testing every component of every webpage for all screen sizes and browsers

2. Timeline was not displaying in the correct position and would appear either near the bottom or near the top of the view box

Solution: position fixing for normal device sizes as developer kept testing on abnormally large monitor

3. Timeline events displaying in incorrect positions

Solution: appending it to the timeline relatively and not to the background which was an error made by the programmer that was refining the timeline. Developer kept manually fixing the position would display erroneously in subsequent fixes until main timeline developer caught the appending issues

4. Timeline filters displaying incorrect information and not showing all results

Solution: fixing database information and the arrays for the filters as they were being called incorrectly by the developer responsible for timeline refining

## **Input Output Model**

An input output model of program testing was used to ensure that users were interacting with the application as developers had intended. It was specifically used for form submissions and user authentication such as the login, signup, password reset and edit profile forms. Additionally, this testing method was also to test the database interactions.

### **1. User authentication**

#### **a. Login Form**

- User inputs their credentials: email and password
- User is able to login and view the web application with the correct permissions.

#### **b. Sign Up**

- User is able to input their details: username, email, password and confirm password with minimum eight characters, two upper case letters, one special character and at least one number
- User is not allowed to input a password without the required security checks
- User must re-enter their password to confirm their password
- And once user submits the sign-up form and an account is created with inputted credentials

#### **c. Forgot Password**

- User inputs emails that they have signed up
- A password reset form is sent to their email containing a reset link
- 

#### **d. Password Rest**

- User can be redirected to the reset page once clicking the link they receive in their email
- User can change their password for their account
- User can log in with new password

### **2. Edit Profile**

- a. User can edit content in the edit profile form
- b. By updating the name and username fields, the user should successfully see their credentials being updated to the new inputs

### **3. Edit Content**

- a. Editor / Administrators are able to add, remove, update events and their related subevents.
- b. Editors are allowed to modify events as both text and form input fields.
- c. Editor is not allowed to submit form without the required input sections like images.
- d. Editor can specify if subevents for an event should not show up on the page.



- e. Editors can specify if certain subevents don't show up on the timeline.
- 4. Search Content
  - a. User is able to search from all the events in the database and is redirected to the exact card on the sub-timeline

### **Sanity Testing for User Credentials**

1. **Password Validation:** This was implemented on the server side and tested with modified elements and console code to break the field, but the controller returns the correct invalid-password error while highlighting the field. On the signup page similarly only valid password that matched the conform password section is accepted on the server side.
2. **User email validation:** For forgot your password page, only registered users can get the verification link and on the signup page only valid email is accepted.

The documents for testing conducted by testers can be found [here](#).

## Section 11 - User Manual For the Website

### Timeline

This feature allows users to interact with a timeline by clicking on circle elements, which open up additional information. Users can filter the timeline by different categories and search for specific events. To use this feature, simply click on the circle elements to view additional information, use the filter options to narrow down the events, and use the search bar to search for specific events.

### User Profile

Users can edit their profile information, including their username, email address, and password. Users are assigned avatars and are able to change their credentials on the profile page if they wish to do so.

### Login/Signup

This feature allows users to create an account or log in to an existing account. The Login link takes us to this page, where users can sign up or log in. Enter your existing valid credentials to either signup or login.

### Forgot Password

On the login and signup page, users are able to go to the forgot your password page to get their emails verified. To do this, Users enter their registered email and follow the link they receive in their emails to reset the password. Without this step, the user is not logged in.

### Admin Content Editing

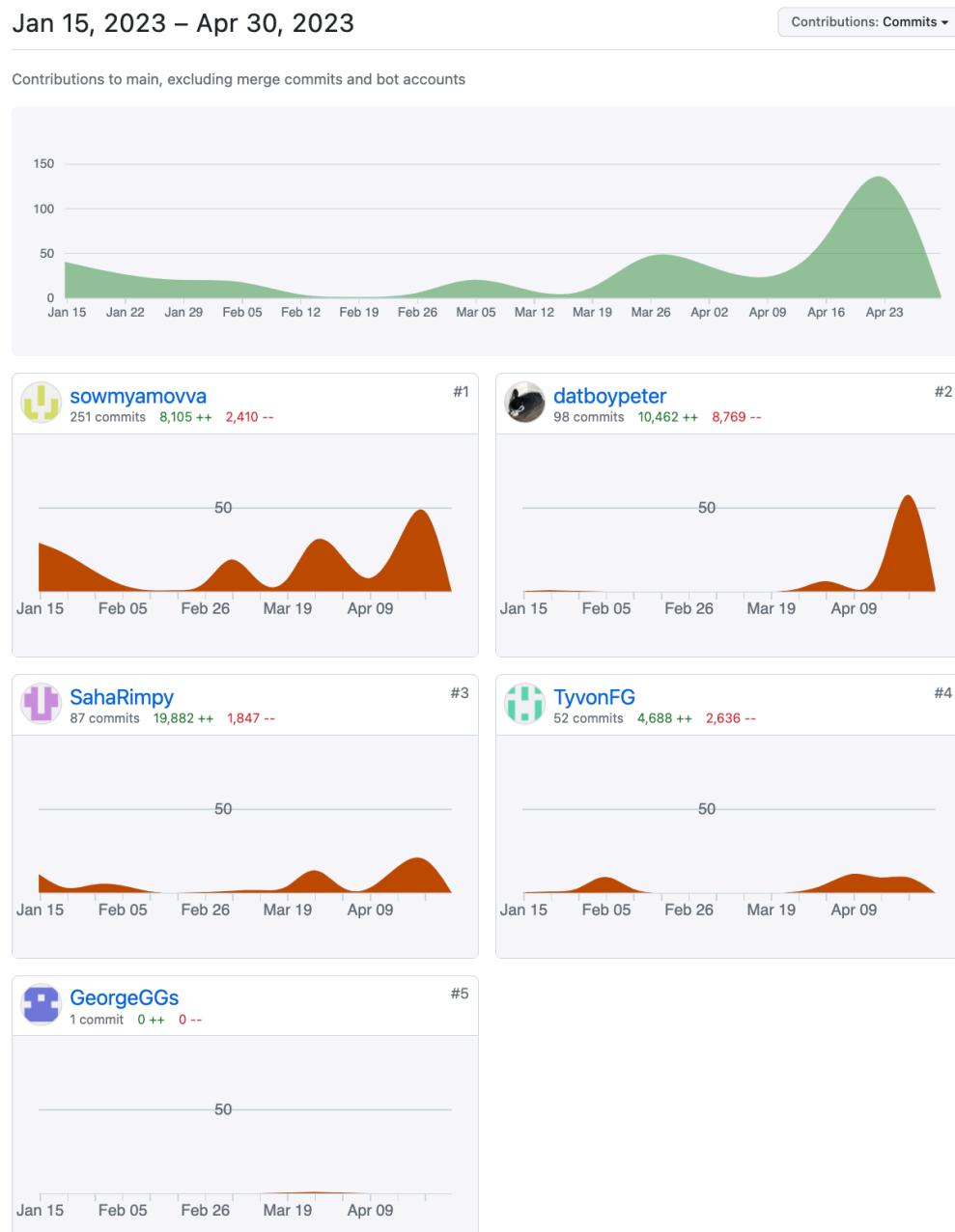
The events, subevents, relations, information and more information, along with the text being spoken, can all be edited, but users of roles editor and higher. The edit button shows up on the timeline page on clicking, and the editor is redirected to the edit-content page to modify any content and has a view button at the top to see the changes they have made.

### Search

The user is able to access information through the search by simply typing and selecting what the user is looking for. After this, the user is redirected to the exact card on the timeline.

## Section 12 - GitHub Logs

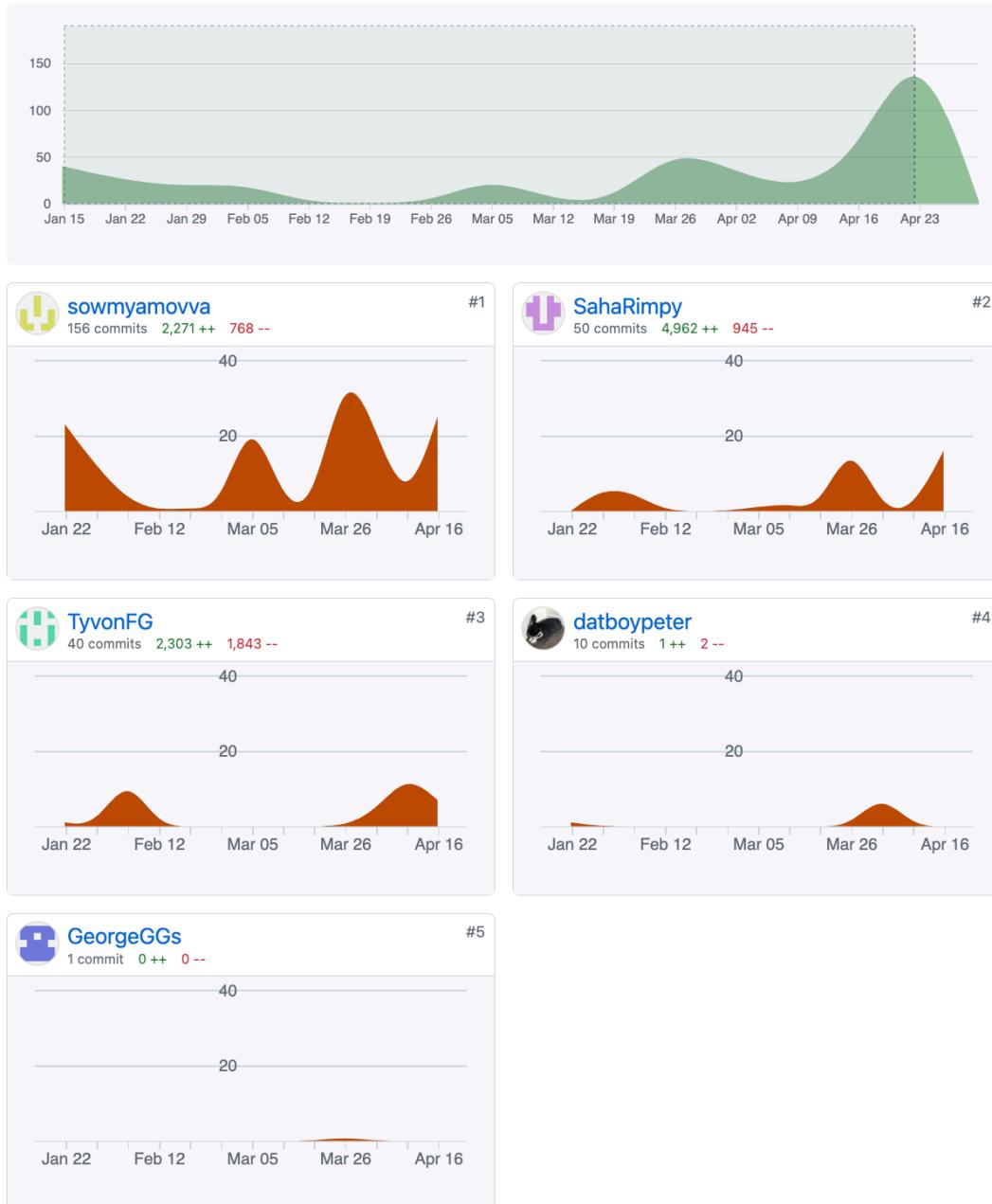
The GitHub logs contributions for the main branch can be viewed below. The first image features contributions throughout the duration of the project and the second image in the next page features progress ranging from Jan 15, 2023 until April 22<sup>nd</sup> which outline groups members that have consistently contributed and were active on GitHub and those that have committed later into the end of the project. Through testing and software bugs were not by testers with automated testing only posted two days prior to the release of the product. Most software bugs were caught by developers and testing was also largely conducted by developers. The automated testing and code provided by testers largely only tested buttons and links.



Jan 15, 2023 – Apr 22, 2023

Contributions: Commits ▾

Contributions to main, excluding merge commits and bot accounts



The commits from 28<sup>th</sup> and beyond were just to ensure all features and the MVC structure were moved from our website file manager to GitHub. Furthermore, contributions for individual members gathered by running a script can be found under the documentation section of GitHub titled ‘Contribution Script’. In addition, the link has also been included in this report.

## Section 12.2 – Individual Contributions

The following are rough lists outlining the teams contributions, please email or contact members if any further clarification is required.

### Rimpy Saha

1. Designing Database
2. Hosting website and setting up MVC folder structure that I maintained.
3. Making the database and initial set of queries and function that we might need later on.
4. Implementing secure HTTPS connection for website with http redirects to the HTTPS (Peter and George researched SSL certificate, tested it and raised the concern about forcing traffic to HTTPS)
5. Made other rules such as to remove the .php and .html extensions
6. Researching different timelines.
7. Building on one timeline using graphics and showing elements for events on clicking circles.
8. Refined the timeline with events placement, debugging etc. throughout the later sprints.
9. Added filters to the timeline and indications for user interactions like click styling, hover scrolling, signifiers etc.
10. Built, refined and debugged sub-timeline - also refined so it's a different graphical element and visually separable.
11. Built Login, Logout Signup functionality with user input sanitization, password hashing. Password security standards and sessions. - Used the queries and function I made earlier. - This is where I also implemented user roles.
12. Built user profile edit and reset password functionality
13. Further built upon reset password for forgot your password and setup mailing system for email verification system.
14. Made an edit content page only available for editors where they are able to add, remove update any event and / or its sub-events. The view/edit tab was also implemented at this stage.
15. Search functionality – on search it redirects to timeline and scrolls and opens the relevant card – also dynamic.
16. Database population and modification to adapt to new features along with minor features like making the text to speech customizable, along with speech ongoing detection to make a stop button available to the users with user friendly buttons on all cards.
17. I was responsible for all code integrations for data, database, frontend, features implementation while making sure it was all dynamic.
18. Making product backlog and refining it as we progressed to better match the desired product.
19. Debugging throughout the scrum procedure.



## Sowmya Contributions

1. Responsible for All Reports and Documentation which can be located under the documentation [folder](#)
2. Developed meeting agendas for all [meetings](#)
3. Facilitated every sprint planning and review [meetings](#)
4. Created all front-end for all pages using css, javascript and [html](#)
5. Visited museum on multiple occasions for data gathering and research after task was delayed by other group [members](#)
6. Compiled data from the online museum database and compared with existing [data](#)
7. Redesigned logo for museum
8. Created homepage graphics which acted as a timeline teaser using [figma](#)
9. Created timeline graphical elements using [figma](#)
10. Create sub-timeline graphical [elements](#)
11. Created all front-end webpages for progress [presentations](#)
12. Integrated front-end with back-end for progress [presentations](#)
13. Developed responsive webpages for all [pages](#)
14. Testing and debugging throughout application development and integration [process](#)
15. Scrum process oversight
16. Made sprint and product [backlogs](#)
17. Remade and continuously updated backlog.
18. Diagnosed team [issues](#)
19. Connected events and sub-events tables in the database with graphical elements to be displayed in the timeline.
20. Debugged events and sub-events database issues throughout integration [phase](#)
21. Upscaled resolution of graphics in figma
22. Created text to speech functionality for all browsers (Chrome, Edge, Safari, and Firefox)
23. Debugged header and search bar and recreated css after interfering with existing [css](#)
24. Picked up tasks left incomplete by other group members.

### **George Gramatikov**

1. Front-end main timeline initially with Tyvon.
2. Created a static timeline.
3. Aided with the initial sub-timeline research.
4. Researched Horizontal Scrolling Functionality and implementation with Tyvon.
5. Researched zoom feature/functionality and overlay for our timeline to implement with Tyvon.
6. Refined the timeline and design of it.
7. Testing code, creating documents, and tests-cases done with Peter:
8. Testing Document which includes test-cases and the tables for each testing method.
  - a. Test current bugs, future possible problems, and the testing methods themselves.
  - b. (testing\_document.pdf) on GitHub under the testing folder.
9. Tested the cases from the “full\_list\_of\_test\_cases.pdf” located in the GitHub.
10. Code implementation for unit testing using JUnit and Selenium. Located in the GitHub Testing Folder. Some code unfortunately was removed.
11. Cyber Security. SSL Certificate.

### **Peter Fung**

1. Developed prototype for all proposed web pages based on class diagram. These web pages include: homepage, user account page, our services, about us, login, sign up, donate, contact us, a second homepage to reflect when a user is logged in.
2. Researched and obtained an SSL certificate on the web-application for Cyber Security.
3. Developed new test cases for web pages and features after every sprint meeting.
4. Visited the museum in person and gathered data by photographing various artifacts along with its descriptions.
5. Developed code for unit test cases for the Chrome and Safari browser using JUnit and Selenium web driver, though unfortunately was forced to delete most of the code under team orders. The additions and deletions can be seen in the history of commits on GitHub. More information can be found in the email titled sent clarifying the additions to deletions ratio.
6. Developed code for unit test cases for incognito mode on Chrome using JUnit and Selenium web driver, though unfortunately was forced to delete most of the code under team orders. The additions and deletions can be seen in the history of commits on GitHub. More information can be found in the email sent clarifying the additions to deletions ratio.
7. Created and co-authored alongside George the testing plan document used for Progress report 2
8. Created test cases used for progress report 2
9. Documented bugs found on the timeline and the steps needed to reproduce the bugs.



1. Created and co-authored testing document used for the final progress report with George.
2. Created and authored the document/excel file containing the full list of test cases done on the website, includes testing for bugs found to see if they were fixed. The full list of test cases can be found on the GitHub repository under the 'testing' folder.
3. Extensively tested the website using all test cases found in the 'full\_list\_of\_test\_cases' document/excel sheet found on the GitHub repository.

### Tyvon Factor-Gaymon

#### **As Product Owner:**

- Continual identification and prioritization of product features for development
- Assessed and reviewed, product backlog multiple times.
- Assisted in setting up Taiga, a project management tool, to establish a structured process for documenting the product backlog and made a tutorial for using it.

#### **As Backend Developer:**

- Developed and refined a static timeline template that enabled the team to visualize the goals we wanted to achieve

**Developed many features:** Calendar, Stacked Images, Card Flipping, Card Transitions, Timeline Toggle Orientation, Embedded videos, **Accessibility: Speech-To-text**, Accessibility: Text-To-Speech, Rich Fields function, Various Zooming features

#### **Timeline Development:**

- Responsible for processing data into suitable format so that timeline and sub timeline images could be flipped
- Responsible for initial positioning of timeline elements (circles, data) on timeline and repositioning of elements when scrolling.
- Ensured that these positions best accurately represented distances between dates
- Refined previous functions such as the range filter and category filters
- Helped develop area for providing additional information
- Helped improve accessibility by creating initial version of text-to-speech function
- Assisted in Timeline integration (towards end)
- Debugging: a major component that helped to shape and refine timeline

