

## Progress Report Document

George Gramatikov

Peter Fung

Rimpy Saha

Sowmya Movva

Tyvon Factor-Gaymon

March 5, 2023

## Table of Contents

1. Introduction .....	3
1.1 Objectives of this Document .....	3
1.2 Document Convention .....	3
1.3 Intended Audience .....	3
1.4 Readings .....	3
1.5 Project Scope .....	4
1.6 References .....	4
2. Description .....	4
2.1 Product Prospective .....	5
2.2 Product Functions .....	5
2.3 User Classes and Characteristics .....	6
2.4 Operating Environment .....	6
2.5 Hosting .....	6
2.6 Product Constraints .....	7
3. Product Design .....	7
3.1 Design Implementation .....	7
3.2 Proposed Timeline Designs .....	7
3.3 Basic Timeline Features .....	12
3.4 Database .....	13
3.5 Webpage Design .....	14
3.5.1 Class Diagram .....	14
3.5.2 Webpages .....	16
4. Sprints and Progress .....	17
4.1 Sprints .....	17
5. Project Management Tools .....	19
5.1 Taiga .....	19
5.2 Trello .....	20
5.3 Flowmapp .....	20
5.4 Teams .....	21
5.5 Outlook .....	21
5.6 GitHub .....	21
6. Progress and GitHub Log .....	21
6.1 GitHub Logs .....	21
6.2 Individual Team Member Task Breakdown .....	22



# **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. Information and research material relevant for this project have been gathered by visiting the museum and taking detailed notes of various exhibits and artifacts. Additionally, staff have provided a link to a website that houses information as well as 3d photos of all artifacts present at the museum.

## **1.2 Objectives of this Document**

The main objective of this document is to observe the requirements of the timeline from the perspective of various stakeholder. We will be noting initial stages of the requirements and developing them into functional and non-functional requirements. Furthermore, the document notes rationale behind decisions and development stages of the projects as the team progresses through sprints.

This document also aims to provide insight into high level design and architecture of the application at this stage and will not contain code. It will look at details of the application at higher levels of abstraction rather than technical details.

## **1.1 Document Conventions**

The document is structured according to sections which are further broken down into subsections. There are five main sections which will focus on the high-level details of the timeline project. The first section will provide an introduction that will provide basic details regarding the project, requirements, and document structure. This document may also reference other sections outside of a given section which will be indicated using the section number. Any references of external material will be referenced using ieee format.

## **1.2 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 taking COSC 4P02 and instructors are also the intended audience of this document.

## **1.3 Readings**

Team members had to conduct research on various topics to ensure that the client had a successful interactive timeline that can be implemented for their existing website. Once a thorough understanding of an interactive timeline and its features were established, team

members had to conduct research on Niagara on The Lakes History. Initial content research for the information being displayed on the timeline was done by having a team member (Sowmya Movva) physically collect data at the Museum. After organizing relevant information into coherent categories, this information was used to make sections for the timeline. Team members were then able to use this to find further information on these sections and events online. (Immigration, slavery, war of 1812, transportation, immigration).

## 1.5 Project Scope

The scope of the project aims to provide insight as to what the result of the interactive timeline will look like. The purpose of this deliverable is to provide the user with an interactive and semi-immersive experience throughout history where the user can click and drag a virtual timeline. After clicking and dragging a timeline, the user should be able to see historical events pertaining to a given era that they select in the application. Specific requirements and features outlined during the requirements engineering process will be described in later sections of the document. Requirements have been evaluated on a priority basis and were assigned points which were used to assign sprints. Features with lower points were determined to be the most important and thus developers will be implementing these features during initial sprints. Given that an agile methodology is being utilised, sprint planning meetings will review requirements and sprints designated for a given sprint by the scrum master and voice any concerns the team may have. Features not deemed as essential to the product will be reviewed at later sprints and all team members will conduct a feasibility analysis at that time in order to decide whether a feature will be implemented. This ensures that the most important features are implemented first.

## 1.6 References

The references used to create the interactive timeline are the existing Museum website, Wikipedia for various historical events included in the timeline, and artifacts database provided by the museum staff. All references can be found on the Main GitHub page under *Resources*.

# Section 2 – Description

This section will provide a high-level overview of the website/product, its purpose, and the many features that it employs. Data relevant to the basic features, types of users, database, and design will be briefly examined here. Only a general review of the future final product and its components will be addressed in this section.

A timeline is a visual tool that represents a chronological sequence of events. These events are displayed around a line indicating a direction in time. The most commonly used timelines tend to use a line that is drawn horizontally or vertically with year and data about an event on either side. It serves as a catalyst to facilitate one's understanding of the ordered events. An interactive timeline will allow the user to manually manipulate historical events by dragging their mouse or

finger when using a touchscreen to display a given event of a particular time period. The interactivity will allow for the user to view greater volumes of data such as exhibits and artifacts without overwhelming the user and also allows for control as to what events a user would like to view. 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.

## 2.1 Product Prospective

The product that is being developed for the museum is an interactive timeline which will be hosted on the existing Niagara on The Lake Museum website. The product will serve as an extension of the current website and will be a web application that allows users to have a virtual experience without having to physically visit the museum thus allows for greater reach. The software will initially be hosted locally so that developers and staff can test it without interfering with the live website. Given that this product will serve as an extension of the existing website, it requires networking as it will be hosted online. Additionally, all users that visit the website should have access to it whether it is on a pc or mobile. Thus, the product must be visible for all members of the public that have internet access and wish to view the museums timeline.

## 2.2 Product Functions

The end product must emulate an immersive user experience and will need to implement all necessary features to do so. The final product will include the following features and constraints:

- The software must be available on any personal device connected to the internet and must support a wide range of browsers and operating systems similar to the existing website
- The software must be always available online and there must be no service interruption during maintenance thus there must be no dependencies with other components of the website.
- There should be user controls that allow users to navigate forward and backwards through time and therefore these controls should utilize intuitive interactivity.
- There must be a main menu that provides the user with options that can enhance their interactive timeline experience and facilitate their navigation through the website.
- The software must not interfere with other applications or utilise a large amount of system resources since it is hosted online.
- The timeline should indicate what the user has last viewed.
- The sub timeline should not only be relevant but also visually indicate that the user is now viewing a sub timeline under a particular group of filters also chosen by the user.
- The sub timeline must show information relevant to the selected date within the sub timeline.
- The timeline must allow the user to explore between specific time periods using a basic zoom function and horizontal scroller.



- The zoom feature must be compatible across a variety of different devices including laptops, touchscreens, monitors etc.
- The information rendered will be dynamic meaning the information will not be stored on page and rather in a database. This is to ensure easy edits of the displayed timeline.
- The website should have a feature, for authorized personnel, that facilitates online editing of content without prior technical knowledge.
- A navigation feature can be enabled to help new users get around the website.
- Allow users/clients to have an option to review and comment on anything they feel worked well and what did not work well after using the website.

Every team member contributed to user stories. The product owners then converted all user stories into backlog items and created an ordered product backlog. The scrum master is responsible for making each sprint and communicating the rationale behind the decisions with the team. The team then discusses the decision and asks for changes if necessary and proceed with selecting or continuing tasks.

## **2.3 User Classes and Characteristics**

The interactive timeline will be accessible to all visitors of the museum as well as all website visitors who are interested in learning more about Niagara's history. Thus, it will be available 24/7 and can be accessed from anywhere. Not all users will have knowledge on interactive timelines or how to operate them thus intuitive design will be utilised where users can click and drag with a mouse or use their keyboards to access information. Additional resources on how to operate an interactive timeline such as tutorials and a help page will also be always provided and easily accessible to users.

## **2.4 Operating Environment**

One of the main objectives of this product is to provide an alternative access to information, albeit limited, relevant to the artifacts presented at the museum. As mentioned in previous sections, the software will be available online and assumes that the user has access to a wireless internet connection. The software will only be available if the user is connected to wireless internet as with the museum website. The application is standalone and will not require any external applications for it to run nor any special hardware. In addition, the application will need to support modern browsers and operating systems to accommodate a wide range of personal devices that connect to the internet. Currently, team members are local hosting to minimize interference with the live site.

## **2.5 Hosting**

Upon conducting research, hosting was determined to be a later concern. Brief research has been conducted on services such as infinity free with extensive pre-development research being conducted on firebase for database, development, and hosting. We are currently focused on developing rather than hosting, so we are currently locally hosting the pages. It must also be noted that the firebase platform has its own database, but it was non-relational so it was decided that it would not be used.

## **2.6 Product Constraints**

The scope of the project is defined by the constraints on the applications domain which are not limited to the accessibility, application, components, device, interface, interaction, programming language, and system constraints.

### Language Constraints:

It is important to note that the museums exhibits are currently provided on English and French. As a result, the content contained within the website will be available in English and French. All accessibility features, including text-to-voice, and closed captions for videos, will be in English for the initial stages of development. After receiving client approval, bilingual services will be added. However, there are some browsers that support the translation of English to other languages and so it must be noted that the translations are not necessarily perfect.

### Device Constraints:

Our website is intended for desktops and laptops. It will not be directly translated to work for the use of mobile devices. Thus, not all features will work on the mobile device or adapted for touch screen use. Some features will be transferred over to mobile such as the zoom feature.

## **Section 3 – Product Design**

### **3.1 Design Implementation**

To realize the different intuitive features for the timeline, it is necessary to have a basic timeline that is easy to edit, access, understand and build upon. So, one of the first tasks after consolidating requirements was to find or create potential timelines that have the potential to hold all the features like sub-timelines, accessibility, search, filter and more. Because we are to build on these, the initial primary timelines, we were more focused on reuse. We researched and narrowed the following few templates.

## 3.2 Proposed Timeline Designs

### Timeline 1 (George, Rimpay, Tyvon)

The initial template that we chose was from a static timeline maker who would have been perfect for building on. Multiple timeline modification tools have been used to get an idea of the timeline design. Such tools include: Visme, timeline generator, and timeline maker. The one from Visme looked visually pleasing. However, on further discussion, it was decided not to use this template as the code was already split up into a couple of files with a substantial amount of code without having any dynamicity or interactivity. The code also used unconventional programming practices making the rationale behind choices challenging to understand and highly confusing to edit. An image of the timeline could be downloaded but it would be inconvenient to make changes to the timeline and access it via JavaScript and CSS. The template was eventually ruled too complex to work with and dropped as a potential idea or component feature.



Figure 3.2a: The Interactive timeline made using Visme

## Timeline 2 (Rimpy, Tyvon)

Along with providing users with horizontal timelines, we also wanted to give them the option of interacting with a vertical one. This vertical timeline would be more mobile-friendly. The timeline below is a template of the one we were considering on using. The idea is to detect screen dimensions and display the timeline accordingly. So as a sample solution, we found a basic vertical timeline with minimal code and features. Although it uses elements inefficiently, the one we found does an excellent job of segmenting the coding aspect for when we implement the mobile-friendly feature on our timeline. This would make it easier to add and modify content on the timeline. It also gave decent insight into the visual aspects of a timeline, which we are going to work on incorporating in the coming sprints.



*Figure: Timeline using a simple existing template*

## Timeline 3 (Rimpy)

One of the consistent issues we faced with basic timelines was the ruler or timeline indicator, which was usually a margin of a DIV element. The third timeline is a simple yet dynamic graphic code that places as many points on the line as needed. As can be seen in fig 3.3.2, as the user zooms in, the dots get finer. This displays more detailed information on more specific dates. This would be compatible with almost all of our features, including scroll and zoom. The graphics elements are now also communicating accurately with the information elements like the div block. The next step is to replace any timeline ruler that we are working on to get the features ready so as not to halt progress. The constant change in div elements would make the page lag,

and with an SVG element instead, we can simply move information instead of the whole timeline. We are also currently looking into easy tools for graphics for the timeline.



*Figure a: An interactive timeline using graphic elements.*

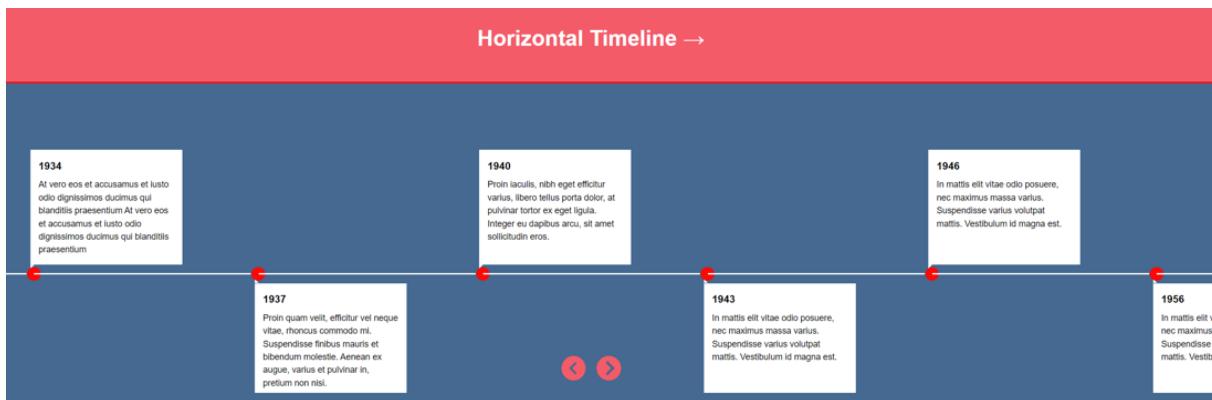


*Figure b: The graphic timeline with the div element.*

#### Timeline 4 (Rimpy, Tyvon)

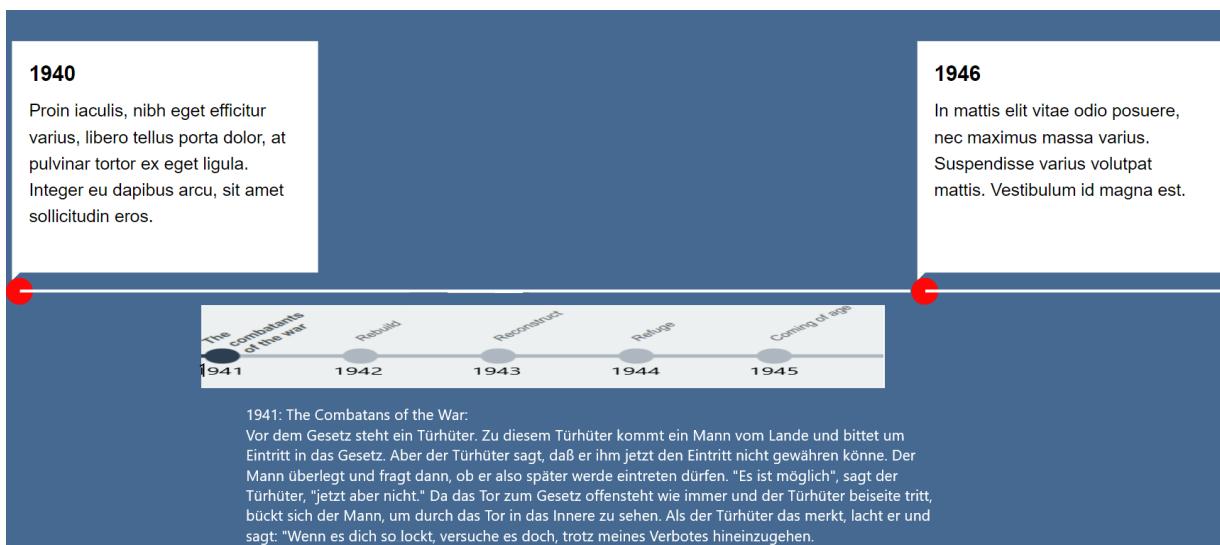
This is another template that we were able to find. It had a decent markup for the information being displayed, making it easy to make interactive and dynamic, so we chose this as the skeleton structure. Another reason it was selected amongst the others was that it was simple and adding content to it was easy. This timeline is compatible with the scroll and zoom features. The scrollbar is not visible on the timeline but using the “left” and “right” arrow keys on the keyboard or by clicking the left and right arrow buttons will allow a user to scroll. With more content, the user could also use the “up” and “down” arrows to view more information. The zoom feature zooms in on the whole timeline. In the future, we would like to have a zoom that actually zooms in to where the mouse is at.

This timeline also clarified for us some challenges in terms of the timeline ruler, further solidifying our decision of a graphic timeline ruler.



*Figure a: Extended timeline with zoom and scroll methods.*

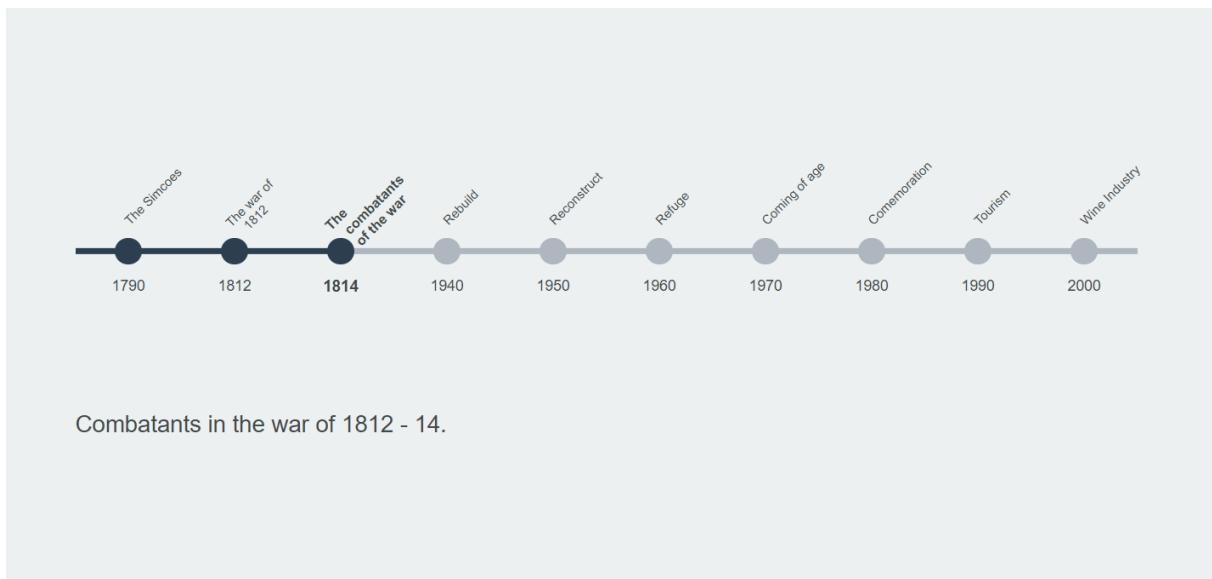
An example use case of the sub timeline can be seen below:



*Figure b: Extended timeline with potential sub-timeline.*

### Timeline 5 (Rimpy)

This timeline was considered for some of its visual features. Albeit small, the features add to user experience and, with the small markup, it's easy to incorporate into any code. The idea was to add this timeline as a sub timeline in between major events presented on the overall/main timeline. When a circle on the sub timeline is clicked, it would generate information relevant to a specific time period. This will be further looked into during the refining stages of our timeline.



*Figure: Extended timeline highlight and keyboard accessibility functions.*

### Timeline 6 (Tyvon, George)



*Figure: Timeline with multiple features.*

This timeline was designed mainly from scratch. Therefore, we made it easy to add or modify any future content. The content under each year can easily be accessed and changed. It also has a horizontal scroller. Should there be conflict with the currently selected timeline, this one may be used instead. Each image when clicked will provide more information on the selected years content in a more zoomed in feature. This will blur the rest of the screen to focus primarily on the selected year.

### **3.3 Basic Timeline Features (Tyvon)**

Currently, the team is able to employ the use of simple timeline features such as scrolling and zooming. However, the team is investigating other unique functions that can be added to the timeline. A possible additional feature that is to be considered are timelines that can flow both

horizontally and vertically. Another feature the team found intriguing was the ability to zoom in on a dynamic timeline that adds more dates/elements as the user zooms in. A feature that the team has also been exploring is the overlay function once an image on the timeline has been clicked. This overlay should darken the background and bring up a larger picture of the clicked image with detailed information about its background.

### 3.4 Database

Another issue was the user access for CMS-like features; the first step towards that was a database or similar structure. We started with a basic er diagram targeting our primary ideas of users, content, and data.

The next step was to set up a database that aids in quickly reading and writing data both directly through the database interface and through database connection. With the team's standard familiarity with SQL and PostgreSQL, we decided to use one of these. We set up a temporary database in PhpMyAdmin, according to the ERD. It will undergo more changes as our content information and data are rearranged for storage.

A general-includes file was also built to establish a one-time connection and keep the password separated for security purposes. The system architecture so far seems to be a Repository structure, and one of our focuses is to remedy the one-point failure aspect of this structure.

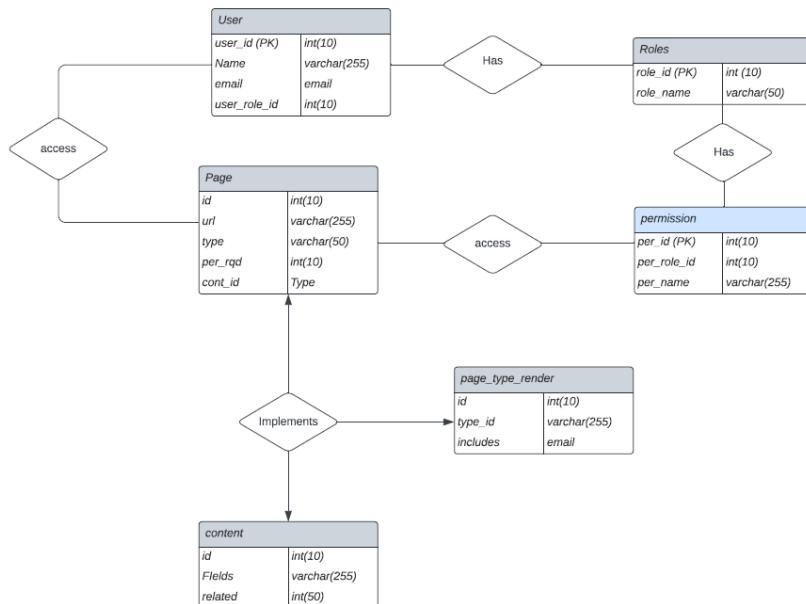


Figure 3.4 Entity Relationship Diagram

## 3.5 Webpage Design

### 3.5.1 Class Diagram

Each webpage that is being designed for the product will act as a separate class. Thus, a high-level overview of the proposed pages was created to provide a visual aid as to which pages will be developed for the initial sprints.

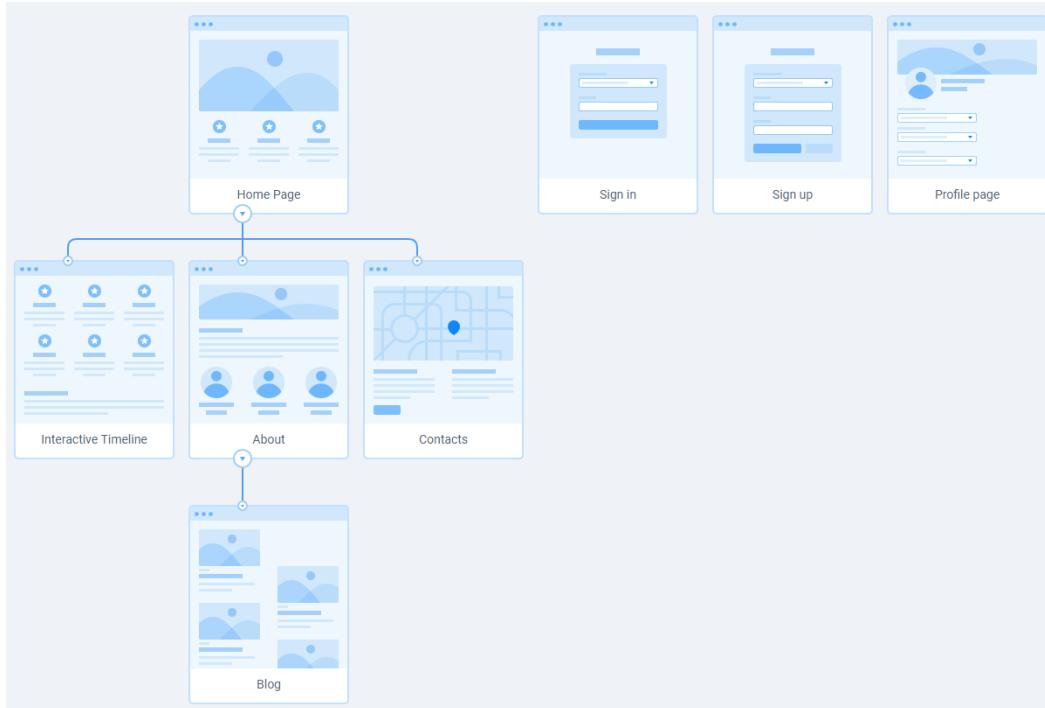


Figure a: Flowchart Diagram created using Flowmapp

Upon clicking a given page in the flow chart, a prototype of that page will be generated. Proposed elements for a given page will then be displayed. Figure 3.10 depicts the Home Page which shows a header, introduces the interactive timeline, a few important elements from the live site and the footer. It is important to note that this is not the final design of our product and only serves as a visual class diagram and prototype. Design is subject to change based on research conducted by the Front-end team which consists of Peter Fung and Sowmya Movva.

Tours   Pricing   About Us   Help Center   Book A Tour

**VIRTUAL EXHIBIT**

**Interactive Timeline**

[Visit Now >](#)

**Interactive**  
User interface design includes selecting and arranging interface elements  
[Action link >](#)

**Virtual Exhibits**  
User interface design includes selecting and arranging interface elements  
[Action link >](#)

**Archives**  
User interface design includes selecting and arranging interface elements  
[Action link >](#)

**Niagara On The Lake Museum**

Located in historic old town, we hope to preserve Niagara's rich history and educate the thousands of tourists that visit Niagara each year.

The digital age is transforming the way we interact with history and we are seeking to adapt as well.

**Indigenous History**   **Immigration**   **War of 1812**   **Tourism and Transportation**

<b>03</b> april Workshop 1	<b>12</b> april Presentation with Local Institution	<b>20</b> april Explore Tourism
----------------------------------	---	---------------------------------------

**Niagara On The Lake**  
[hello@notlmuseum.com](mailto:hello@notlmuseum.com)

**Explore**  
Explore and Experience Niagara-on-the-Lake's Unique History

**History**  
Building History  
Strengthening Community  
Capital Campaign

Figure b: Prototype of Homepage created using Flowmapp

### 3.5.2 Webpages

All primary webpages shown in a class diagram have been created and hosted locally. Figure a is a screenshot of the current homepage which is provided below as a sample of the progress. The remaining pages have been uploaded to the main branch of GitHub and can be accessed from the repository.

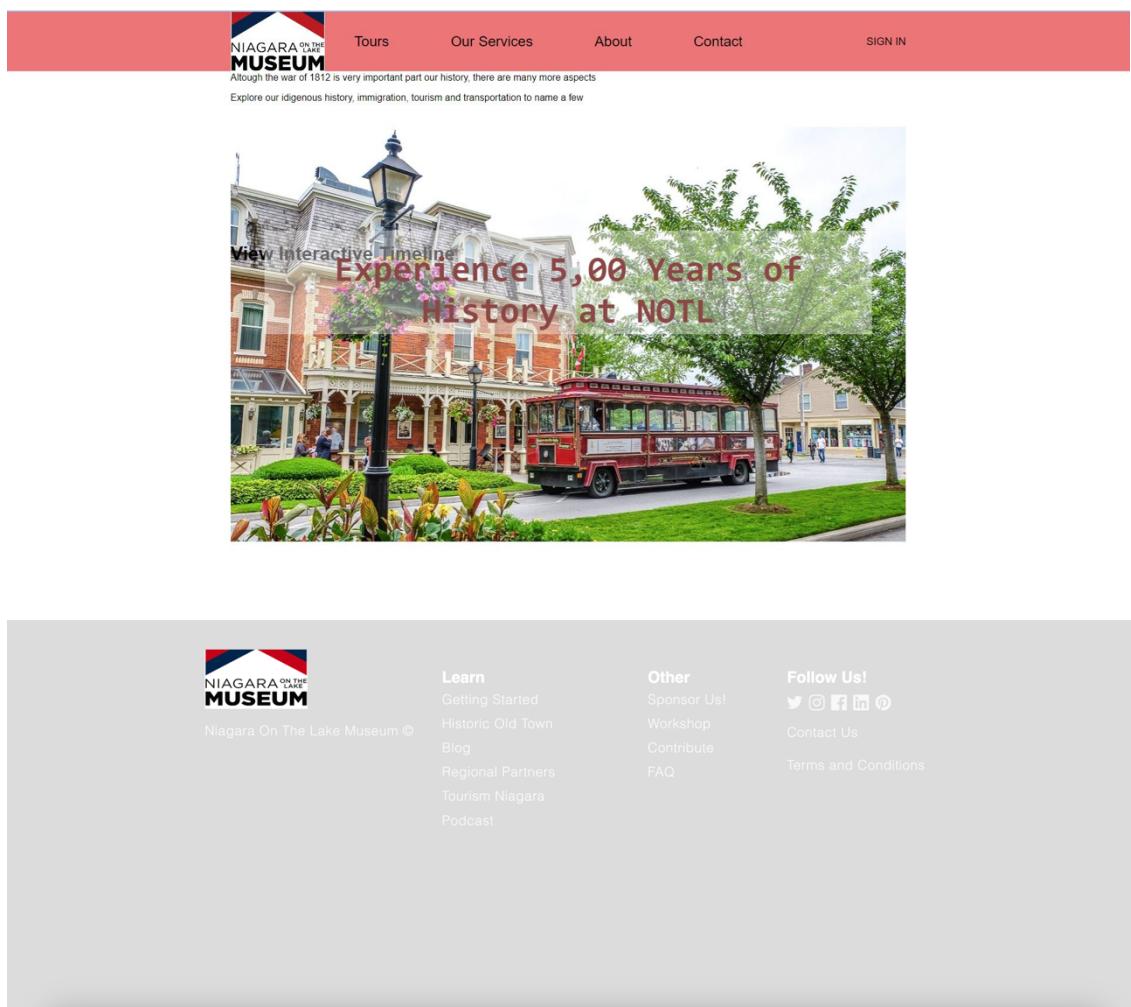


Figure a: Locally Hosted Sample Webpage created using HTML and CSS

The design was inspired by other modern museum webpages. However, CSS is still being modified as this is a very rough prototype. With additional modifications to CSS, Script and Content, the design will adhere to modern design principles in terms of visuals as well as accessibility.

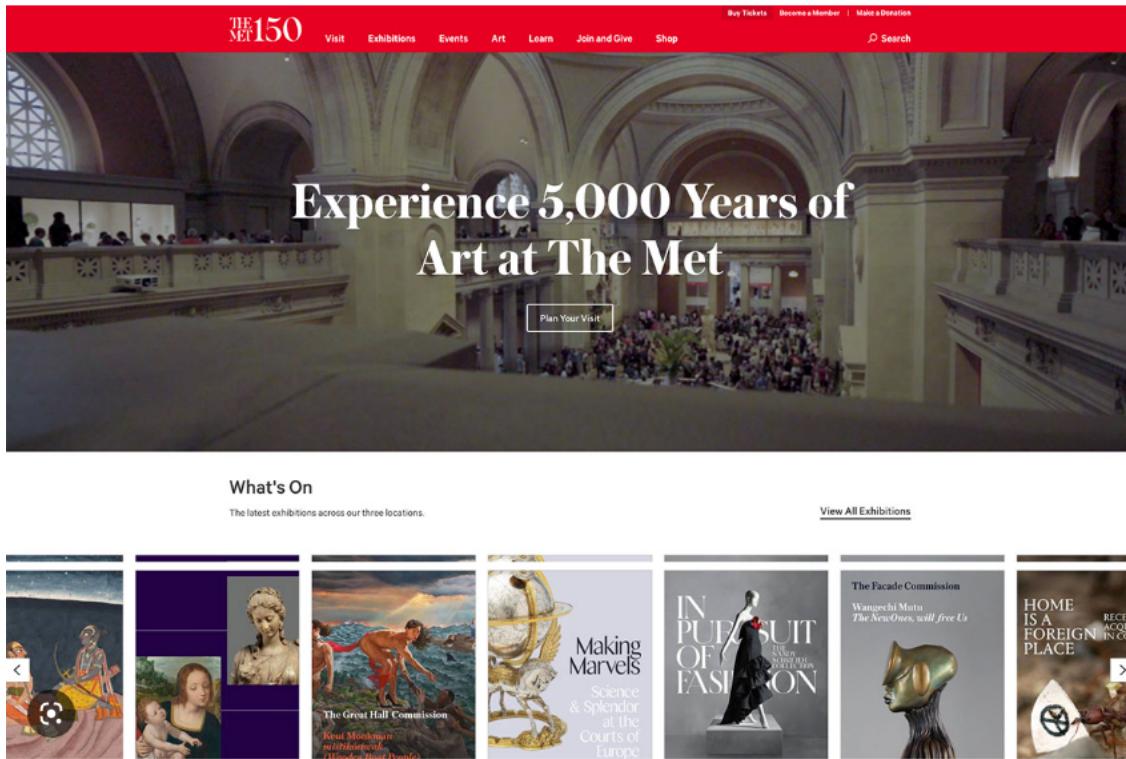


Figure b: MET Website used for research

Figure 3.12 depicts one of the sites obtained during research and was used as inspiration. Given that The Met website shows modern design in the context of museums as with many other large museums, we were able to gather that there is a large emphasis on content with minimal visual noise. In addition, there was simple and minimalistic design elements used. For example, the header is very simple and contains a logo as well as few main pages and a search bar. The remainder of the homepage features one main attraction with some content below. Minimalistic and modern design was then used as a source of inspiration for our webpage.

## Section 4 – Sprints and Progress

### 4.1 Sprints

Meeting materials consisting of meeting notes which also describe weekly progress, sprint planning and sprint review are available in the main branch of GitHub in the *Meeting Materials* folder. First two sprints mainly involved product owners and scrum master.

#### Sprint 1

The team explored various interactive tools that can be developed to attract more visitors and expand the reach of the Niagara on The Lake Museum. Team members conducted research individually and met to discuss the pros and cons of each potential project. Based on findings, the team concluded that an interactive timeline will be developed. In addition, roles were assigned, and tasks were established with three members assigned to back-end and two members

assigned to front-end. At this stage, a project proposal was submitted to the client which can be found in the Documentation folder of the Main GitHub Repository Branch.

### Review

During this sprint review, product owners presented potential projects so that the team can discuss which project was best suited for client requirements and resource limitations. In addition, the team receives a demonstration of the project management and administrative tools that will be used for development. The most notable tools were Taiga, Trello, Microsoft Teams, and GitHub. One of the tools highlighted was Taiga. It was used as the primary tool to manage all the user stories and features. Trello was also shown, and the team appreciated its ability to manage the implementation tasks and sub tasks. The first sprint heavily comprised of requirement engineering research and presenting tools for backlog management: Taiga and Trello.

### Retrospective

Project presentations allowed for the team to decide on moving forward with the Interactive Timeline Project. The scrum master created and submitted a project proposal to the client. Product owners began to discuss client requirements and create a product backlog.

### Sprint 2

Product owners and scrum masters facilitated the requirements engineering process. Product owners created backlog items and organized them into Product backlogs. All team members participated in the requirements engineering process by creating at least five to ten user stories and discussing them during the requirements meeting. Apart from organizing all the requirements, this sprint comprised of multiple meetings to decide on the basic framework of our interactive timeline. We also researched ideas about timeline frameworks, typical content and organizing data into a database.

### Review

Due to the iterative nature of the methodology being used, administrative tools were subject to review with earlier tools being replaced by more efficient ones after members had the chance to review. The team also discussed client requirements and user stories.

### Retrospective

At the end of this sprint, the team talked about the organization of the requirements and the division of the implementation tasks based on the product and sprint backlogs. We had discussed the challenges that we were facing and how to mitigate them. We noticed that we were spending far too much time in our scrum meetings and at times there would be misunderstanding on who performed what tasks. We addressed the first problem by following meeting agenda provided in the meeting invitation so that team members are aware of action items prior to the meeting. To combat miscommunications and misunderstandings, we decided to use Trello as the base tool for all our tasks and each member is to update every task they are assigned and additional tasks that they are working on.

### Sprint 3

It was in this sprint that the team began coding components to be included on different webpages. For this reason, we focused our efforts on establishing the database connection, header, footer, homepage, basic timeline, login, and sign-up pages. The backend team focused their efforts on the database and timeline research. Whereas the front-end team focused on the remainder of the tasks as well as research on modern UI/UX principles.

### Review

The team was able to showcase the different webpages and website components that they were working on. The team presented multiple different timelines that were compiled from online sources and decide on which is best to use as a template based on which features were best for client and user requirements.

### Retrospective

During our discussion, we wanted to refocus our efforts on making a unique interactive timeline that stands out from others. For this reason, we decided to prioritize implementation tasks relevant to the timeline. There were also two notable horizontal timeline scroller templates that were found that would greatly add to the interactive timeline experience. It was too complex to decompose and understand so they are less of a priority. It did help to point out that as developers, the team should not expend too much time on one task, if it isn't necessary.

## Section 5 – Project Management Tools

### 5.1 Taiga

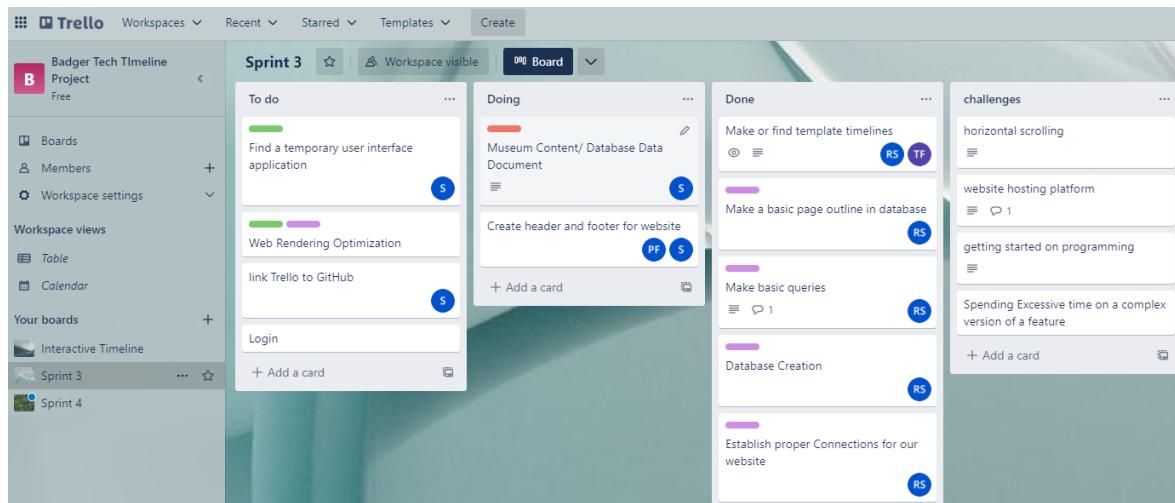
This is a cloud-based tool used to manage different agile methodologies. It is currently being used to manage the scrum process. The product backlog and the sprint backlog are kept and tracked here.

The screenshot shows the Taiga interface for managing a project titled "Interactive Timeline Project". On the left, a sidebar lists "Scrum", "Backlog", and "Issues" sections, with "Sprint 10: Testing" currently selected. The main area is divided into two main sections: "Scrum" and "Backlog".  
  
The "Scrum" section displays a progress bar at 0% completion, 1,000 defined points, 0 closed points, and 0 points/sprint. Below this, there's a "CUSTOMIZE YOUR BACKLOG GRAPH" section with a note about setting up points and sprints through Admin.  
  
The "Backlog" section shows 62 user stories. It includes filters for "subject or reference" and "Tags", and buttons for "+ USER STORY" and "+". A table lists three user stories:

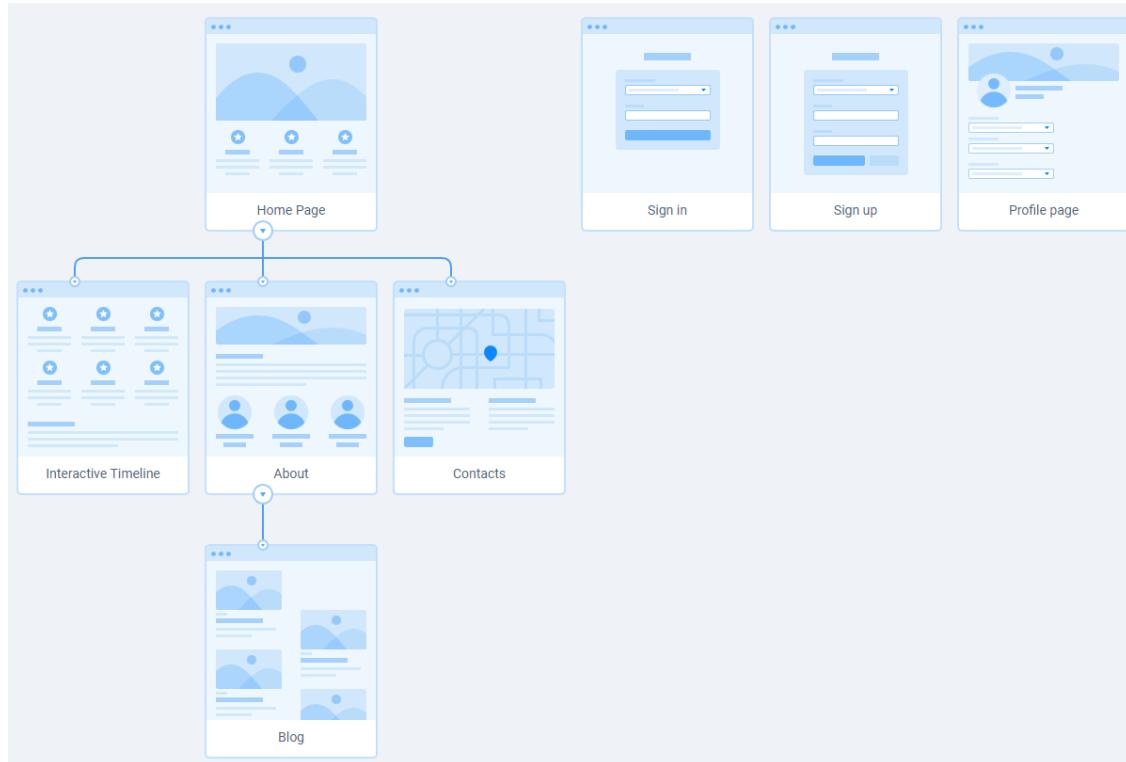
USER STORY	STATUS	POINTS
#27 As an editor or above, I want to be able to manage the content published to the website to ensure that the data is accurate. <small>main</small>	New	60
#73 As a user, I want to be able to have access to a menu of the most used features so that I can quickly navigate the website.	New	60
#69 As a User, I want to be able to have easy access to contact information and features on every page.	New	60

## 5.2 Trello

This is an organizational tool that allows the team to have fine control over a project. It is where the team breaks down user stories and features into implementation tasks. It also allowed the team to write about any challenges they were facing.



## 5.3 Flowmapp



An application that provides a visual overview of the proposed webpages that are set to be created for the web application. This tool essentially acts as an in-depth class diagram as well as a prototype of the pages.

## 5.4 Microsoft Teams

An application that serves has a hub for team collaboration, communication and weekly meetings. It is being used to hold meetings, converse with each other, provide updates, and link/share files.

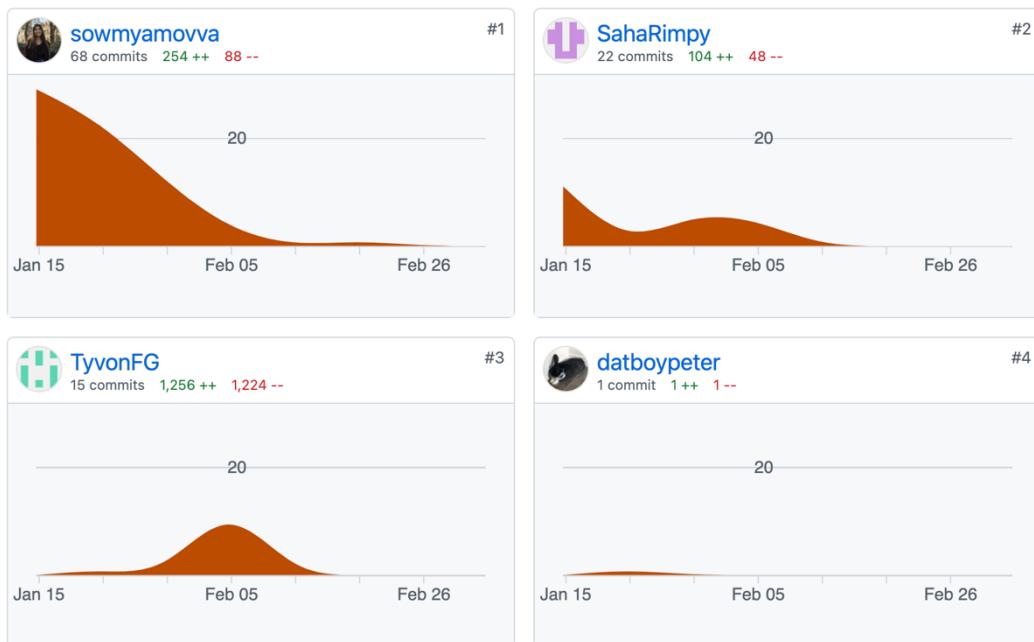
## 5.5 Microsoft Outlook

Outlook is utilised to share meeting agendas and meeting links. It also serves as a team calendar communicating meeting times and deadlines to team members. Once a meeting has been conducted, meeting materials are shared through email and subsequently posted to GitHub.

## 5.6 GitHub

A cloud-based git repository that allows the team to store all of the website folders and files. All relevant project source code files and documentation can be always accessed by the team. In addition, GitHub allows members to simultaneously work on code.

## Section 6 – Progress and GitHub Logs



## 6.1 GitHub Logs

The GitHub log alone is not an accurate representation of the amount of work an individual has done. The first member shown in the log with the greatest number of commits is the scrum master and owner of the repository which involves configuration of the repository as well as constantly uploading and updating documents and file structures thus resulting in more commits. Trello tasks will be provided in this report and can be used by the instructor to determine a given team members contributions. In addition, this report will outline tasks completed by a team member for each given sprint. The team has only started to make programming related commits from Sprint 3 onwards as prior Sprints mainly involved documentation and research.

## 6.2 Individual Team Member Task Breakdown

### Scrum Master - Sowmya Movva

- Created project selection presentation consisting of all pros and cons for each potential project in order to facilitate discussion
- All documentation and presentations
- Project proposal and progress report creation
- Facilitation of meetings and creation of meeting agendas
- Creation of sprints in Taiga and Trello based on product backlogs
- Conducted research and gathered data for all museum artifacts and exhibits
- Organized data based on exhibit and time period
- Consolidated data into a word document for use in later sprints to populate timeline database
- Front-end development tools
- Designed class diagram and flow chart for all webpages
- Created header, footer and CSS for all webpages
- Created homepage template using HTML and CSS based on prototype
- Created outlines for all webpages
- Setup GitHub Repository and folder structure
- Send weekly meeting invites on Teams with agenda
- Researched and Implemented project management tools and set up framework
- Researched and setup firebase for GitHub (redacted)
- Researched subsystems for integration in later sprints
- Researched modern design principles for front end

### Product Owner and Developer – Rimpy Saha

- Facilitated project selection discussion
- Requirements Engineering team members.
- Making Backlog Items.
- Making Product Backlog.
- Conducted research and gathered data for all museum artifacts and exhibits
- Designing ERD for the database.

- Creating and managing the database.
- Making a compiled list of functions for accessing data from database.
- Set up how to connect to database for anyone on the team directly via vs code.
- Defining use of page-types for content distinction.
- Setting up hosting on infinity-free and changing to the infinity-free database and researching for firebase.
- Making a simple timeline using visme (refer to timeline 1).
- Scraping features for vertical timeline (refer to timeline 2).
- Graphic elements for timeline and their apt interaction with other elements (refer to timeline 3)
- A timeline with click scrolls – mostly a template and will lose most it's elements (refer to timeline 4)
- A timeline which highlights the user exploration of the timeline making for a more intuitive user experience – the feature will be filtered for our timeline (refer to timeline 5).
- Working on svg vertical timeline and integrations and compiling of all approved features so far.

#### Product Owner – Tyvon Factor-Gaymon

- Facilitated project selection discussion
- Requirements Engineering team member.
- Created backlog items and product backlog.
- Facilitated client requirement discussion.
- Researched potential database and hosting platforms.
  - Set up firebase and firebase database (deemed too complicated)
- Looked into potential tools to aid backend development (nodeJS, Express.js)
- Investigated potential timelines and their features
  - Timeline 1: investigated visme timeline and exporting it as webpage files.
  - Timeline 2: Analyzed timeline for possible horizontal use.
  - Timeline 4: Added scroll and zoom functionality to this timeline and examined possible modifications.
  - Timeline 6: Created this timeline and implemented different functionalities.
  - Looked at other timelines but they were deemed far too complex.
- Looking into potential features that can be adapted to our timeline

#### Developer and Tester – George Gramatikov

- Participated in project selection discussion
- Requirements engineering team member
- Participated in discussion on client requirements
- Participated in discussion on website features
- Researched potential timelines and their features
  - Timeline 1: Modified potential Visme Timeline
  - Timeline 6: Helped to create this timeline
- Participated in selecting and refining official timeline design

- Reviewed Project Proposal
- Reviewed Progress Report

#### Developer and Tester – Peter Fung

- Participated in project selection discussion
- Requirements engineering team member
- Participated in discussion on client requirements
- Participated in discussion on website features
- Participated in outline creation on Flowmapp
- Utilised design of homepage and translated it to Contact page, services, and about us
- Modified CSS for webpages
- Researched modern website designs for our website
- Researched subsystems for integration in later sprints
- Reviewed Project Proposal
- Reviewed Progress Report
- Reviewed and provided feedback for Flowmapp

