

Museum Assistance Program

Progress Report 1

GROUP MEMBERS:

STEVEN SABHARWAL | 6537484
REET CHAHIL | 6465314
EVAN GALEA | 6477285
YANNICK DE CALUWE | 6418396
GEORGE BARAKAT | 6429161
HENRY TU | 6464143
ARSHDEEP SINGH | 6837827
FAHAD UMAIR | 6797658

GROUP MEMBERS:	1
Roles:.....	2
Sprint 1 (January 31 st to February 14 th):	3
Overview:	3
Database Work By: Yannick De Caluwe	5
Design Work (Figma).....	8
React Work (Front End of the website)	15
Sprint 2 (February 14 th to February 28 th):.....	16
Overview:	16
Database Work.....	17
React Work (Front End of the website)	19
Future Sprints:.....	20
Sprint 3 (March 7 th -April 21 st)	20
Sprint 4 (April 21 st -April 4 th)	20
Sprint 5 (April 4 th -April 21 st)	20
Meetings Log:.....	21
Git Hub Contributions:.....	22
Summary of Key Tasks	23

Roles:

- Steven Sabharwal: Scrum Master, Developer
- George Barakat: Product Owner, Developer
- Yannick De Caluwe: Developer
- Evan Galea: Developer
- Reet Chahil: Developer
- Henry Tu: Developer
- Fahad Umair: Developer
- Arshdeep Singh: Developer

Sprint 1 (January 31st to February 14th):

Overview:

Overall Goals:

- Design the webpage.
- Design the Database
- Implement the Database Queries
- Setups react for the webpage.
- Create the webpage with react.
- Admin login

Achieved:

- Design the webpage.
- Design the Database
- Implement the Database Queries
- Setup react for the webpage

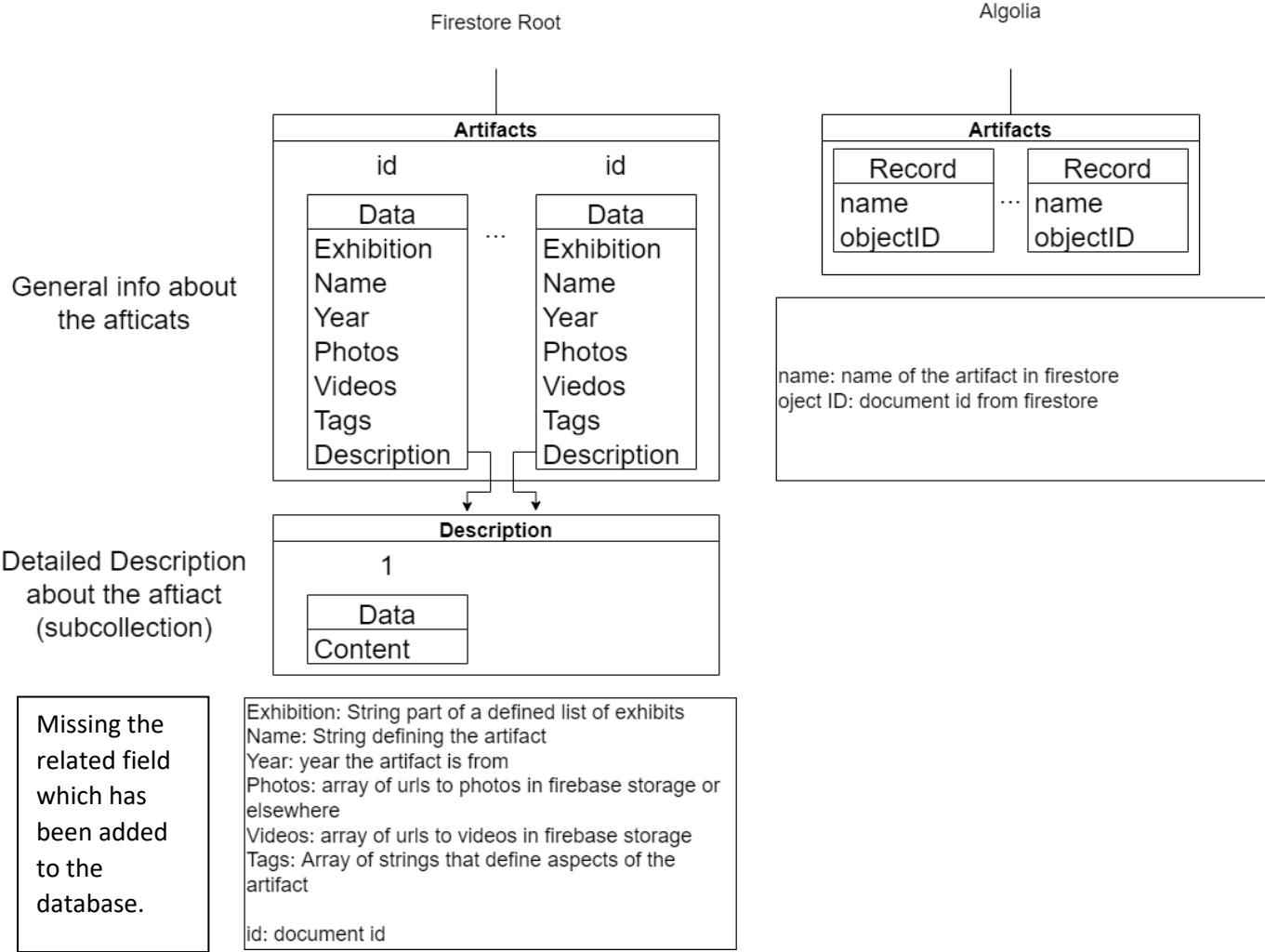
Sprint Plan:

▼ IT4P02 Sprint 1 31 Jan – 14 Feb (12 issues)
█ IT4P02-1 The timeline should be able to retrieve data about artifacts to fill the timeline from a data base.
█ IT4P02-2 Design the database for a timeline about artifacts and exhibits
█ IT4P02-3 Implement the database based on the design
█ IT4P02-4 Test that the created database works as expected
█ IT4P02-7 Identify other frameworks and APIs needed
█ IT4P02-8 Finalize design plans for artifact timeline page (TIME LINE FORMAT) AS A USER, I ...
█ IT4P02-5 Integrate the database into the timeline.
█ IT4P02-6 Host the website so it is accessible on the web
█ IT4P02-9 Landing Page and Exhibition Select (TIME LINE FORMAT) AS A USER, I ...
█ IT4P02-20 As an Administrator, I want to be able to click a login button to sign in. (LOGIN) AS AN ADMINISTRATOR, I ...
█ IT4P02-10 As a User, I want to artifacts to be in chronological order (TIME LINE FORMAT) AS A USER, I ...
█ IT4P02-11 As a User, I want to be able to see a basic information of an artifact when scrolling through the timeline (TIME LINE FORMAT) AS A USER, I ...

Sprint plan at the end:

▼ IT4P02 Sprint 1 31 Jan – 14 Feb (12 issues)
█ IT4P02-1 The timeline should be able to retrieve data about artifacts to fill the timeline from a data base.
█ IT4P02-2 Design the database for a timeline about artifacts and exhibits
█ IT4P02-3 Implement the database based on the design
█ IT4P02-4 Test that the created database works as expected
█ IT4P02-7 Identify other frameworks and APIs needed
█ IT4P02-8 Finalize design plans for artifact timeline page (TIME LINE FORMAT) AS A USER, I ...
█ IT4P02-5 Integrate the database into the timeline.
█ IT4P02-6 Host the website so it is accessible on the web
█ IT4P02-9 Landing Page and Exhibition Select (TIME LINE FORMAT) AS A USER, I ...
█ IT4P02-20 As an Administrator, I want to be able to click a login button to sign in. (LOGIN) AS AN ADMINISTRATOR, I ...
█ IT4P02-10 As a User, I want to artifacts to be in chronological order (TIME LINE FORMAT) AS A USER, I ...
█ IT4P02-11 As a User, I want to be able to see a basic information of an artifact when scrolling through the timeline (TIME LINE FORMAT) AS A USER, I ...

Database Work By: Yannick De Caluwe



Artifacts > 08C1DHn7Wp6...

More in Google Cloud

cosc4p02-project-a5335	Artifacts	08C1DHn7Wp6kD0v87HFo
+ Start collection	+ Add document	+ Start collection
Artifacts	08C1DHn7Wp6kD0v87HFo	Description
admin	0sRSLqi8xKsm0qV5VAzT	+ Start collection
artifacts	13GWFFBjZNaNbtq7zJGA	+ Add field
	28VampUIqbof2QBM8uW1	! https://dummyimage.com/100x204.png/00a0d0/fff
	3HxsnzTd2RhYB9eCzzP0	- Related
	5UDgEeDFYuU0PFmGKwod	- Tags
	6m3rVyLKhN00CEGuVpt4	0 "Letter"
	7BGUu6Gfn6dnAuzJMinn	1 "Furniture"
	80z8BYXBxz107ChBVuo	- Videos
	8TvE9JXDUG0V0tAVv5cT	0 "https://firebasestorage.googleapis.com/v0/b/cosc4p02-project-a5335.appspot.com/o niagara_on_the_lake.mp4%20(540p).mp4?alt=media&token=6c5aae5e-2767-488e-bbcc-b22649cc48d3"
	8hSjxRK45HuNUAZrbhx	Year: 1918
	Cjom25md6NW4Rgc6QttP	
	D01Q0yo8cJUcsP5V11kB	
	F59R0YScLSGtGyLwqEwt	

Adding data into the data base is simple with the method created we can simply give a JSON file with records that will be added to the data base in the format defined above.

Above and below are a couple screenshots of the web fire store console of one document after 50 documents had been added (the data is mock data created using <https://www.mockaroo.com>)

The screenshot shows the Google Cloud Firestore interface. On the left, there's a sidebar with navigation links like 'Artifacts' and '08C1DHn7Wp6...'. The main area displays a collection named 'Artifacts' with 50 documents. One specific document, '08C1DHn7Wp6kD0v87HFo', is selected and expanded. This document contains fields such as 'Description' (with a value of 'Indigenous History'), 'Name' ('Bitwolf'), 'Photos' (two URLs), 'Related' (empty), and 'Tags' ('Letter' and 'Furniture'). Below this, another document 'sGPQ3YqlBjRFVqLYWsxf' is shown under the 'Description' field. At the bottom, there's an 'Index' section with 50 records and an average record size of 15.06B. The interface includes various buttons for managing data and a search bar at the bottom.

Testing some queries to make sure it works as expected:

```
console.log("Full fillter Test 1:")
await queryTagsExhibitsYearRange(["Indigenous History"], ["Furniture"], 1750, 1850)
console.log("Full fillter Test 2:")
await queryTagsExhibitsYearRange(["War of 1812"], ["Furniture"], 1750, 1850)
console.log("Full fillter Test 3:")
await queryTagsExhibitsYearRange(["Indigenous History"], ["Book"], 1750, 1850)
console.log("Full fillter Test 4:")
await queryTagsExhibitsYearRange(["Reminiscence"], ["Furniture"], 1700, 1850)
```

Tags Test 2:

- ▶ {Videos: Array(1), Exhibition: 'Indigenous History', Photos: Array(2), Name: 'Ronstring', Year: 1751, ...}
- ▶ {Videos: Array(1), Year: 1758, Tags: Array(2), Exhibition: 'War of 1812', Name: 'Fintone', ...}
- ▶ {Tags: Array(2), Year: 1777, Name: 'Kanlam', Photos: Array(2), Exhibition: 'War of 1812', ...}
- ▶ {Name: 'Flexidy', Year: 1786, Exhibition: 'Indigenous History', Photos: Array(2), Tags: Array(2), ...}
- ▶ {Photos: Array(2), Exhibition: 'Indigenous History', Tags: Array(2), Year: 1828, Name: 'Sonsing', ...}
- ▶ {Year: 1844, Tags: Array(2), Photos: Array(2), Exhibition: 'Indigenous History', Name: 'Stringtough', ...}
- ▶ {Exhibition: 'Indigenous History', Videos: Array(1), Name: 'Bitwolf', Photos: Array(2), Year: 1918, ...}
- ▶ {Name: 'Sub-Ex', Exhibition: 'War of 1812', Photos: Array(2), Tags: Array(2), Videos: Array(1), ...}
- ▶ {Photos: Array(2), Exhibition: 'Indigenous History', Year: 1962, Name: 'Fintone', Videos: Array(1), ...}
- ▶ {Name: 'Subin', Tags: Array(2), Exhibition: 'War of 1812', Videos: Array(1), Photos: Array(2), ...}

Full fillter Test 1:

- ▶ {Videos: Array(1), Tags: Array(2), Year: 1751, Exhibition: 'Indigenous History', Name: 'Ronstring', ...}
- ▶ {Name: 'Flexidy', Videos: Array(1), Photos: Array(2), Year: 1786, Tags: Array(2), ...}
- ▶ {Name: 'Sonsing', Photos: Array(2), Exhibition: 'Indigenous History', Videos: Array(1), Year: 1828, ...}
- ▶ {Year: 1844, Name: 'Stringtough', Photos: Array(2), Videos: Array(1), Tags: Array(2), ...}

Full fillter Test 2:

- ▶ {Year: 1758, Exhibition: 'War of 1812', Photos: Array(2), Name: 'Fintone', Videos: Array(1), ...}
- ▶ {Year: 1777, Videos: Array(1), Exhibition: 'War of 1812', Tags: Array(2), Photos: Array(2), ...}

Full fillter Test 3:

- ▶ {Videos: Array(1), Year: 1777, Name: 'Vagram', Exhibition: 'Indigenous History', Tags: Array(2), ...}
- ▶ {Tags: Array(2), Year: 1786, Videos: Array(1), Name: 'Flexidy', Photos: Array(2), ...}
- ▶ {Videos: Array(1), Photos: Array(2), Exhibition: 'Indigenous History', Tags: Array(2), Name: 'Sonsing', ...}

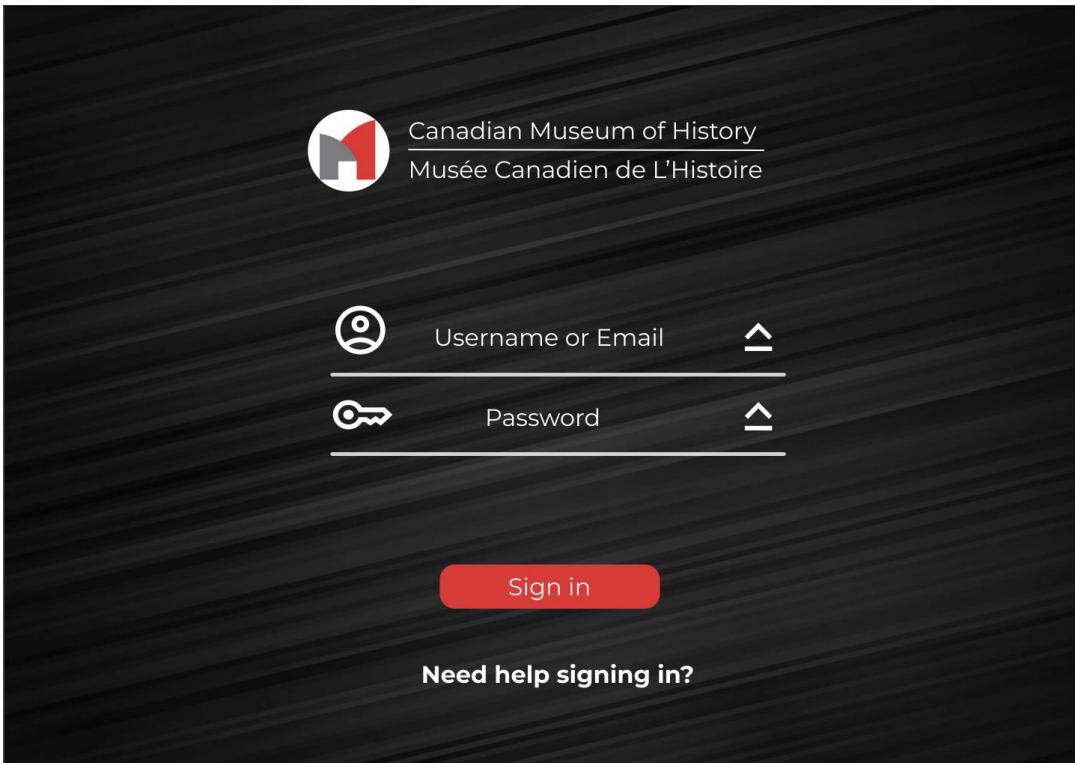
Full fillter Test 4:

- ▶ {Exhibition: 'Reminiscence', Videos: Array(1), Tags: Array(2), Year: 1701, Photos: Array(2), ...}
- ▶ {Videos: Array(1), Year: 1716, Photos: Array(2), Tags: Array(2), Exhibition: 'Reminiscence', ...}
- ▶ {Year: 1723, Name: 'Hativity', Tags: Array(2), Videos: Array(1), Exhibition: 'Reminiscence', ...}
- ▶ {Videos: Array(1), Photos: Array(2), Tags: Array(2), Name: 'Namfix', Year: 1740, ...}
- ▶ {Videos: Array(1), Exhibition: 'Reminiscence', Photos: Array(2), Tags: Array(2), Year: 1761, ...}

Design Work (Figma)

By: Reet, George, Henry, Arshdeep

Admin Pages:



 Canadian Museum of History
Musée Canadien de L'Histoire

Name of Artifact

Date of exhibit

Year BC / AD ▾

Exhibit ▾



Description

Add

 Canadian Museum of History
Musée Canadien de L'Histoire

Name of Artifact

Date of exhibit

Year BC / AD ▾

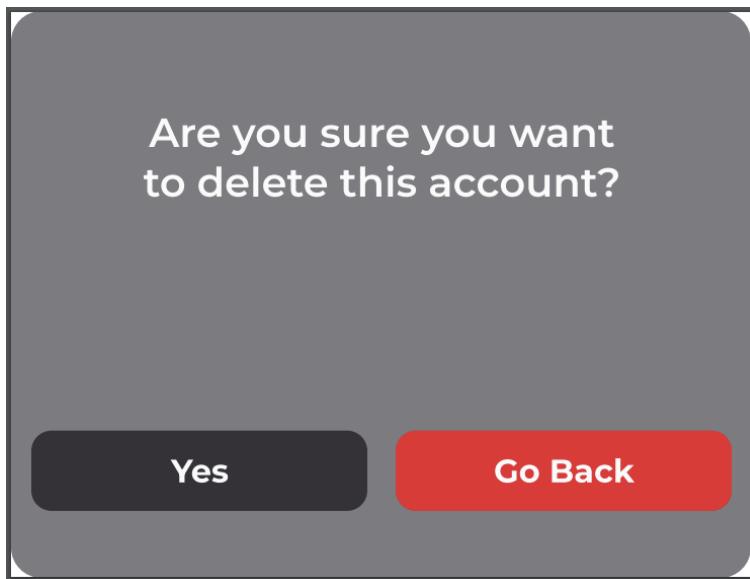
Exhibit ▾



Description

Update

The screenshot shows a mobile application interface for managing administrators. At the top left is the Canadian Museum of History logo. To its right, the text "Canadian Museum of History" and "Musée Canadien de L'Histoire" is displayed. On the far right are two buttons: "Sign out" and a gear icon. The main title "Manage Admins" is centered above a list of four email addresses, each with a red trash can icon to its right. Below the list are two buttons: "Return to Main Menu" and "Add Admin".





Canadian Museum of History
Musée Canadien de L'Histoire

Sign out 

Provide Email and Password for new admin

Email

Password

Re-enter Password

Submit



Canadian Museum of History
Musée Canadien de L'Histoire

Sign out

Profile Settings

Email

Password ⋮ 

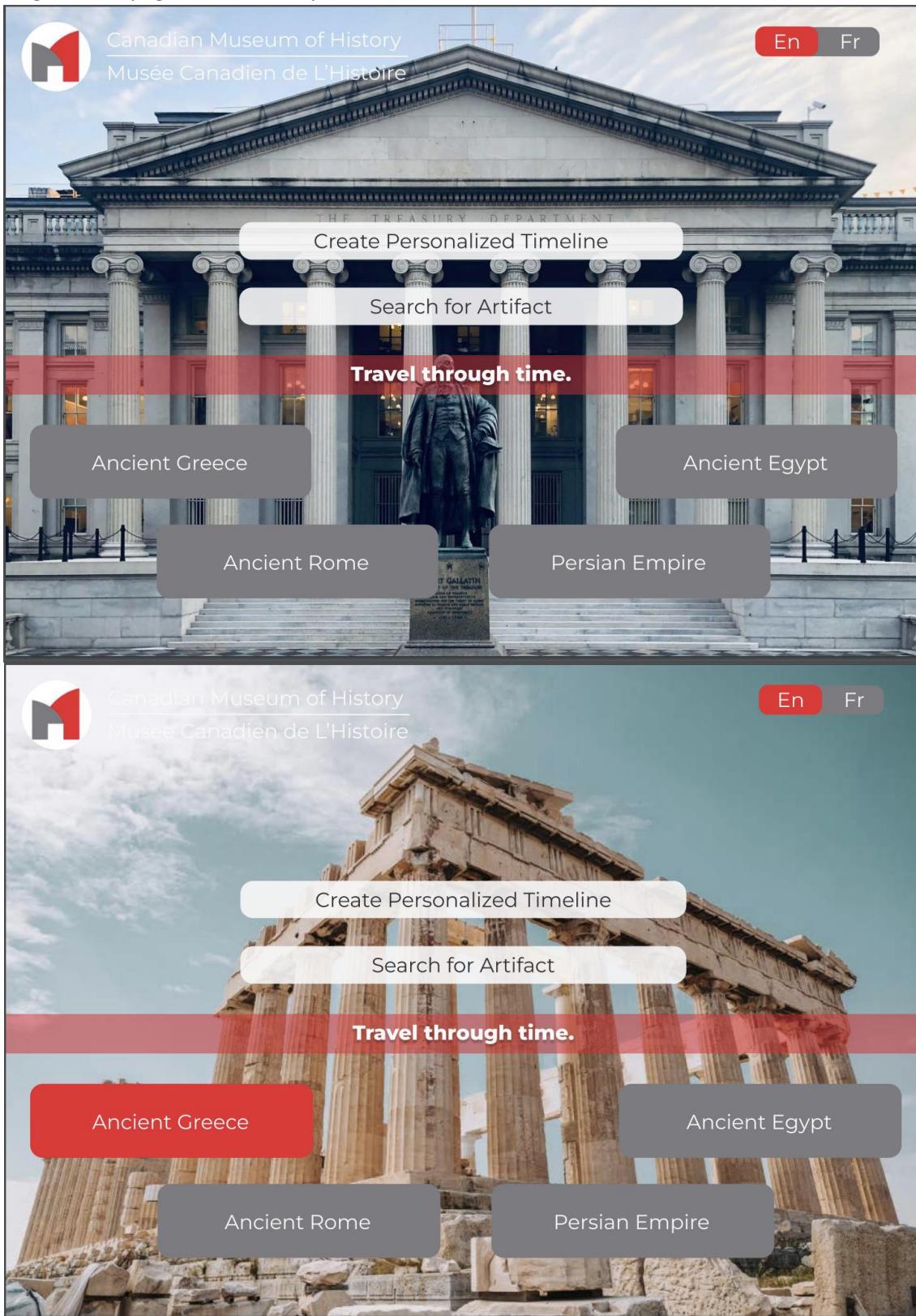
Re-enter Password ⋮ 

Save & Exit

Exit without Saving

Delete Account

User Pages (main pages to reduce report size):



Hovering over a button changes the background accordingly (done for all four, not shown here)

The image shows a mobile application interface for the Canadian Museum of History. At the top left is the museum's logo, a red and white stylized 'M'. To its right, the text "Canadian Museum of History" and "Musée Canadien de L'Histoire" is displayed in a black sans-serif font. Below this, a large red header bar contains the text "Discover history" in white. Underneath the header, there are two sections for selecting exhibits and tags. The first section, titled "Select one or more exhibits", contains four grey buttons with white text: "Ancient Greece", "Ancient Rome", "Persian Empire", and "Ancient Egypt". The second section, titled "Select one or more tags", contains four grey buttons with white text: "Paintings", "Technology", "Weapons", and "Tools". Below these sections is a horizontal timeline represented by a thick red line. On the left end of the line is a grey circle labeled "2000 BCE". On the right end is another grey circle labeled "Present". In the center of the timeline is a red button with the word "Search" in white. The main content area below the timeline displays a series of artifacts arranged in a grid. Each artifact is represented by a circular icon with a diagonal cross through it, indicating it is not selected. The artifacts are labeled "Artifact | 500 BCE" and are positioned along the timeline. A vertical red line runs down the center of the artifact grid. On the far left edge of the screen, the museum's logo is visible again. On the far right edge, there is a small circular icon with a gear symbol.

 Canadian Museum of History
Musée Canadien de L'Histoire

Sign out 



Artifact Added
Ancient Greece Exhibit
Mask of Agamemnon
1250 BCE

[Return to Main Menu](#) [Add Another Artifact](#)

 Canadian Museum of History
Musée Canadien de L'Histoire

Sign out 



Artifact Added
Ancient Greece Exhibit
Mask of Agamemnon
1250 BCE

[Return to Main Menu](#) [Update Another Artifact](#)



Canadian Museum of History
Musée Canadien de L'Histoire

Mask of Agamemnon

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse consequat ut dui sit amet semper. Curabitur pharetra lacus in tellus tincidunt posuere. Pellentesque finibus purus vitae nunc efficitur vulputate. Suspendisse commodo magna at ornare porttitor. Sed dapibus sed dui ut tempus. Nunc elementum sagittis odio et suscipit. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus fermentum tempus facilisis. Ut in convallis purus, ac vestibulum quam. Fusce dolor lacus, finibus eu tellus non, porta vulputate felis. Donec pretium purus risus, et porta risus tempus eget. Vivamus a bibendum metus, vitae commodo tellus. Etiam volutpat justo diam, in rhoncus enim vulputate a.

[Return](#)
[Related Artifacts](#)
[Other Media](#)



Also, there is a mobile design version of everything that is formatted for smaller screens.

React Work (Front End of the website)

By: Evan, Steven

So far, our work has been in learning react and integrating it with existing technologies.

Tested a variety of approaches, settled on a combination of react scripts, npm packages, and other solutions. We had initially decided on a webpack configuration that bundled assets together into single packages, however this proved too complex for our level of skill when working with multiple pages.

The primary tool that we adapted for a multipage website is react router. This provided us with a secure and efficient way to present the user with a consolidated multi-page setup, and fits well with existing technologies.

We now have a solid base to build our website front-end and we are in the process of implementing the various services for our site.

Sprint 2 (February 14th to February 28th):

Overview:

Overall Goals

- Implement the database management functions.
- Create an easy way to use the database (simple functions for retrieving artifact data and changing artifact data)
- Create the front page of the website like in the design.
- Create the filer/search option page.
- Create a timeline page where artifacts will be seen in chronological order.
- Admin login

Achieved

- Implement the database management functions.
- Create an easy way to use the database (simple functions for retrieving artifact data and changing artifact data)
- Create the front page of the website like in the design.
- Create the filer/search option page.

Sprint Plan:

▼ IT4P02 Sprint 2 14 Feb – 28 Feb (7 issues)

<input checked="" type="checkbox"/> IT4P02-1 The timeline should be able to retrieve data about artifacts to fill the timeline from a data base.
<input checked="" type="checkbox"/> IT4P02-5 Integrate the database into the timeline.
<input checked="" type="checkbox"/> IT4P02-6 Host the website so it is accessible on the web
<input checked="" type="checkbox"/> IT4P02-20 As an Administrator, I want to be able to click a login button to sign in. (LOGIN) AS AN ADMINISTRATOR, I ...
<input checked="" type="checkbox"/> IT4P02-9 Landing Page and Exhibition Select (TIME LINE FORMAT) AS A USER, I ...
<input checked="" type="checkbox"/> IT4P02-10 As a User, I want to artifacts to be in chronological order (TIME LINE FORMAT) AS A USER, I ...
<input checked="" type="checkbox"/> IT4P02-11 As a User, I want to be able to see a basic information of an artifact when scrolling through the timeline (TIME LINE FORMAT) AS A USER, I ...

Sprint plan at the end:

▼ IT4P02 Sprint 2 14 Feb – 28 Feb (8 issues) 6

<input checked="" type="checkbox"/> IT4P02-1 The timeline should be able to retrieve data about artifacts to fill the timeline from a data base.
<input checked="" type="checkbox"/> IT4P02-46 Create Functions To modify artifacts in the Database
<input checked="" type="checkbox"/> IT4P02-5 Allow for easy integration of the database into the timeline.
<input checked="" type="checkbox"/> IT4P02-6 Host the website so it is accessible on the web
<input checked="" type="checkbox"/> IT4P02-20 As an Administrator, I want to be able to click a login button to sign in. (LOGIN) AS AN ADMINISTRATOR, I ... 4
<input checked="" type="checkbox"/> IT4P02-9 Landing Page and Exhibition Select (TIME LINE FORMAT) AS A USER, I ... 10
<input checked="" type="checkbox"/> IT4P02-10 As a User, I want to artifacts to be in chronological order (TIME LINE FORMAT) AS A USER, I ... 5
<input checked="" type="checkbox"/> IT4P02-11 As a User, I want to be able to see a basic information of an artifact when scrolling through th... (TIME LINE FORMAT) AS A USER, I ...

Landing Page and Exhibition Select

As a User, I want to artifacts to be in chronological order

In Progress

Priority: Highest

Description:

- Utilize the database to display artifact names on a page

Child issues: Order by ... + 75% Done

<input type="checkbox"/> IT4P02-37	Make the Landing Page	RC	DONE
<input type="checkbox"/> IT4P02-40	Style the landing page with CSS	S	DONE
<input type="checkbox"/> IT4P02-43	Make Custome Timeline Page/Popup	RC	DONE
<input type="checkbox"/> IT4P02-38	Make a page that shows all artifacts from a exhibition selection	B	TO DO

In Progress

Priority: Highest

Description:

This is basically covered in our Database work, as we will be querying artifacts in their chronological order

Child issues: Order by ... + 50% Done

<input type="checkbox"/> IT4P02-44	add capability to get data in order of the date from the database	YC	DONE
<input type="checkbox"/> IT4P02-45	implement the query into the website	B	TO DO

Database Work

By: Yannick De Caluwe

Sample modification quires

```

console.log("addArtifact test:");
await addArtifact("War of 1812", "Left over couch", 1812, [
  "http://dummyimage.com/166x141.png/5fa2dd/ffffff",
  "https://firebasestorage.googleapis.com/v0/b/cosc4p02-project-a5335.appspot.com/o/sample_video.mp4%20(240p).mp4",
  ["Furniture"], "This couch was damaged and left over after the war used by the military");
let data=await queryName("Left over couch");
console.log(data);
let id=data[0][0];//array should contain 1 element which has id in the first element
data=await getDescription(id);
console.log(data);

console.log("updateArtifactDescription test:");
await updateArtifactDescription(id, "This is a new description for the couch");
data=await getDescription(id);
console.log(data.content);

console.log("updateArtifactExhibit test:");
await updateArtifactExhibit(id, "Reminiscence");
data=await getArtifact(id);
console.log(data.Exhibition);

console.log("updateArtifactName test 1:");
await updateArtifactName(id, "This is a Test");
data=await queryName("Left over couch");
console.log(data);
console.log("updateArtifactName test 2:");
data=await queryName("This is a Test");
console.log(data);

```

```
console.log("updateArtifactYear test:");
await updateArtifactYear(id, 1900);
data=await getArtifact(id);
console.log(data.Year);

//Photos
console.log("addArtifactPhotos test:");
await addArtifactPhotos(id, "new photo url");
data=await getArtifact(id);
console.log(data.Photos);

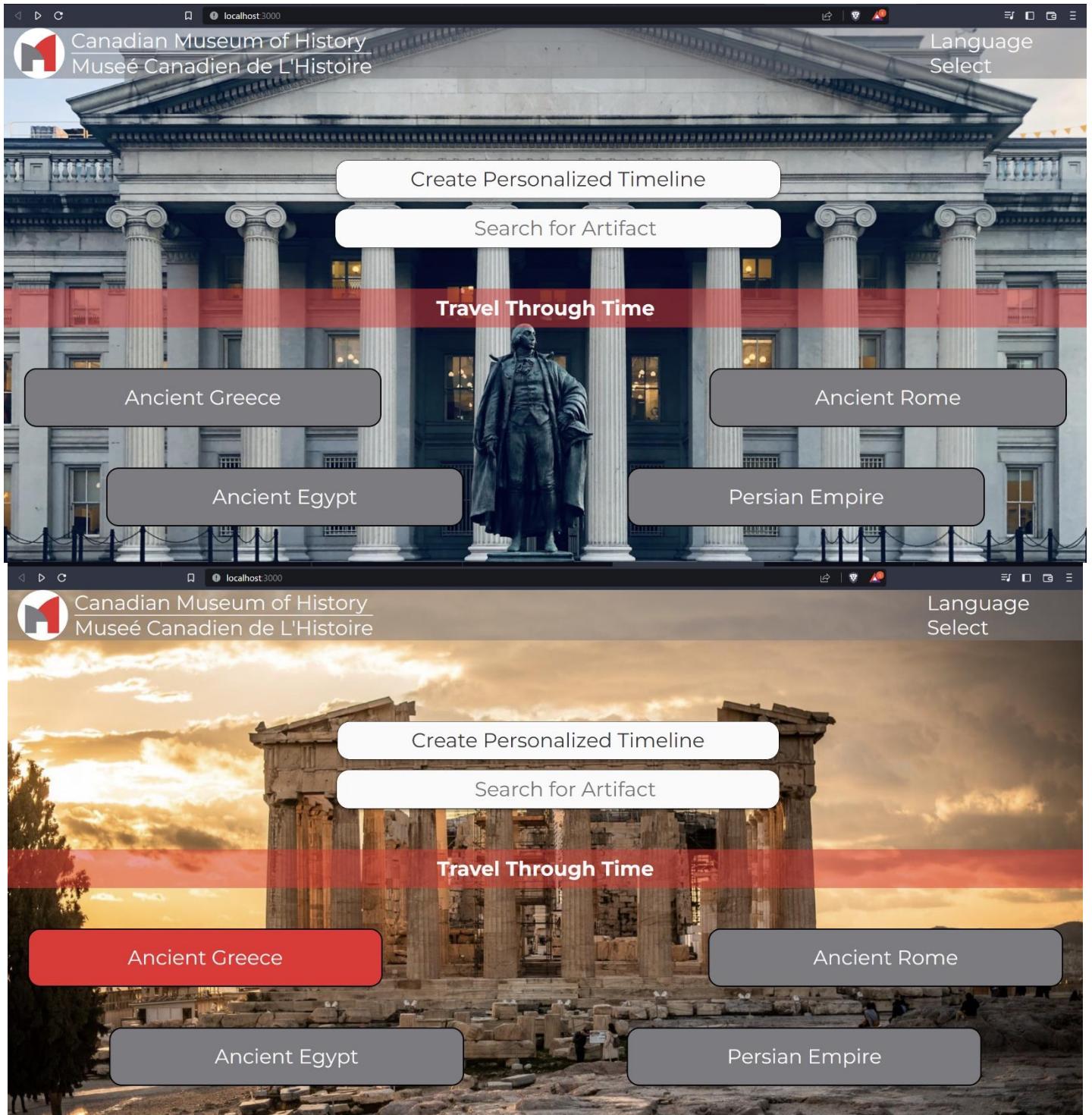
console.log("removeArtifactPhotos test:");
await removeArtifactPhotos(id, "new photo url");
data=await getArtifact(id);
console.log(data.Photos);

console.log("updateArtifactPhotos test:");
await updateArtifactPhotos(id, ["http://dummyimage.com/111x111.png/ddddd/000000", "http://dummyimage.com/246x190.png/ddddd/000000"]);
data=await getArtifact(id);
console.log(data.Photos);

addArtifact test:
▶ [Array(2)]
▶ {content: 'This couch was damaged and left over after the war used by the military'}
updateArtifactDescription test:
This is a new description for the couch
updateArtifactExhibit test:
Reminiscence
updateArtifactName test 1:
▶ []
updateArtifactName test 2:
▶ [Array(2)]
updateArtifactYear test:
1900
addArtifactPhotos test:
▶ (2) ['http://dummyimage.com/166x141.png/5fa2dd/ffffff', 'new photo url']
removeArtifactPhotos test:
▶ ['http://dummyimage.com/166x141.png/5fa2dd/ffffff']
updateArtifactPhotos test:
▶ (2) ['http://dummyimage.com/111x111.png/ddddd/000000', 'http://dummyimage.com/246x190.png/ddddd/000000']
```

React Work (Front End of the website)

By: Reet, Steven, Evan



Select one or more exhibits

Ancient Greece
Ancient Rome
Ancient Egypt
Persian Empire

Select one or more tags

Paintings
Technology
Weapons
Tools

Filter your date range

1000BCE
238AD

Search

Future Sprints:

Sprint 3 (March 7th -April 21st)

- Create a timeline page where artifacts will be seen in chronological order.
- Create the artifact page to see details of a given artifact.
- Ability to use all search options.

Sprint 4 (April 21st -April 4th)

- Create the administrator pages and features.
- Create mobile pages.

Sprint 5 (April 4th -April 18th)

- Language changing feature.
- Add history feature for admins to look at past changes.
- Other Small additions to improve user experience.

Meetings Log:

Thursday January 12th at 7pm online:

- ❖ Everyone attended and worked on the project proposal.

Sunday January 15th at 8pm online:

- ❖ Everyone who could attend did (everyone showed up for at least a bit) and finalized parts of the proposal.

Tuesday January 24th at 2:30pm in person at Brock library

- ❖ Discussed the current product backlog and made some revisions.

Thursday February 2nd at 2:30 pm online:

- Everyone attended and were split into doing certain parts.
 - Yannick and Fahad were put on the database.
 - Evan, Steven, and Henry were put on the front end react.
 - Reet, George, and Arshdeep were put on the Figma design.

Thursday February 9th at 2:30 pm:

- Everyone attended and gave some updates.
 - Reet and George showed the parts of the Figma design to get any pointers.
 - Mostly needed to be phone friendly (making a separate version for it)
 - Yannick mentioned that he was still working on the database.
 - Was in person with Reet and George
 - Reet gave some pointers on how to improve the database.
 - Evan was getting React setup.

Thursday February 16th cancelled and moved to the 19th.

- Yannick, Steven, George, Reet, Henry, and Evan attended.
- Yannick mentioned that most of the database is done just cleaning up parts of it and testing.
- Everyone else was set to work on some part of the React front end.

Git Hub Contributions:

Github's summary does not represent everyone's contributions accurately since design work like the Figma is not included and work done together is only added by one person.



Summary of Key Tasks

- Figma designs
 - Reet, George, Henry, and Arshdeep
- Firebase setup
 - Yannick and Fahad
- Firestore data modelling and query construction
 - Yannick
- React JS front-end development
 - Steven, Evan, and Reet
- ❖ Product Manager
 - George
- ❖ Scrum Master
 - Steven