FINAL REPORT

COSC 4P02

Team Members
Dhrumil Shah (6714380) - Product Owner & Tester
Dalton Comer (6413827) - Developer & Scrum Master
Royce Lando (6603625) - Developer
Aman Yadav (6858054) - Developer
Shahrear Chowdhury (6605273) - Developer Gurashish Anand (6855449) - Developer
Ben Grandy (6090484) - Developer

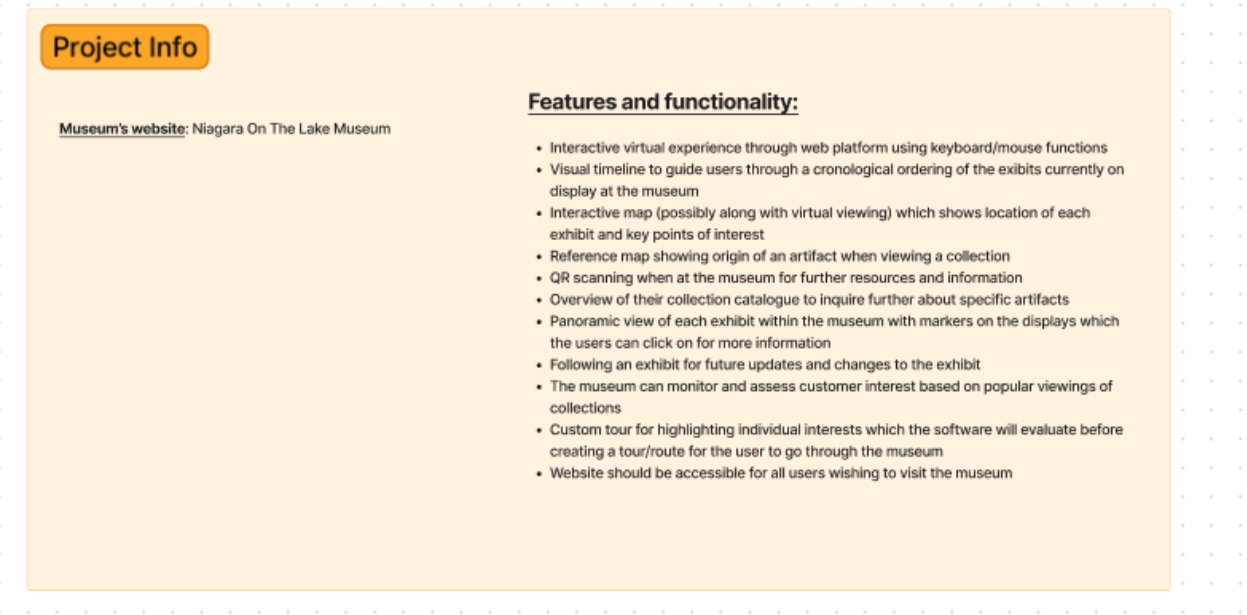
Table of Contents

Project Overview

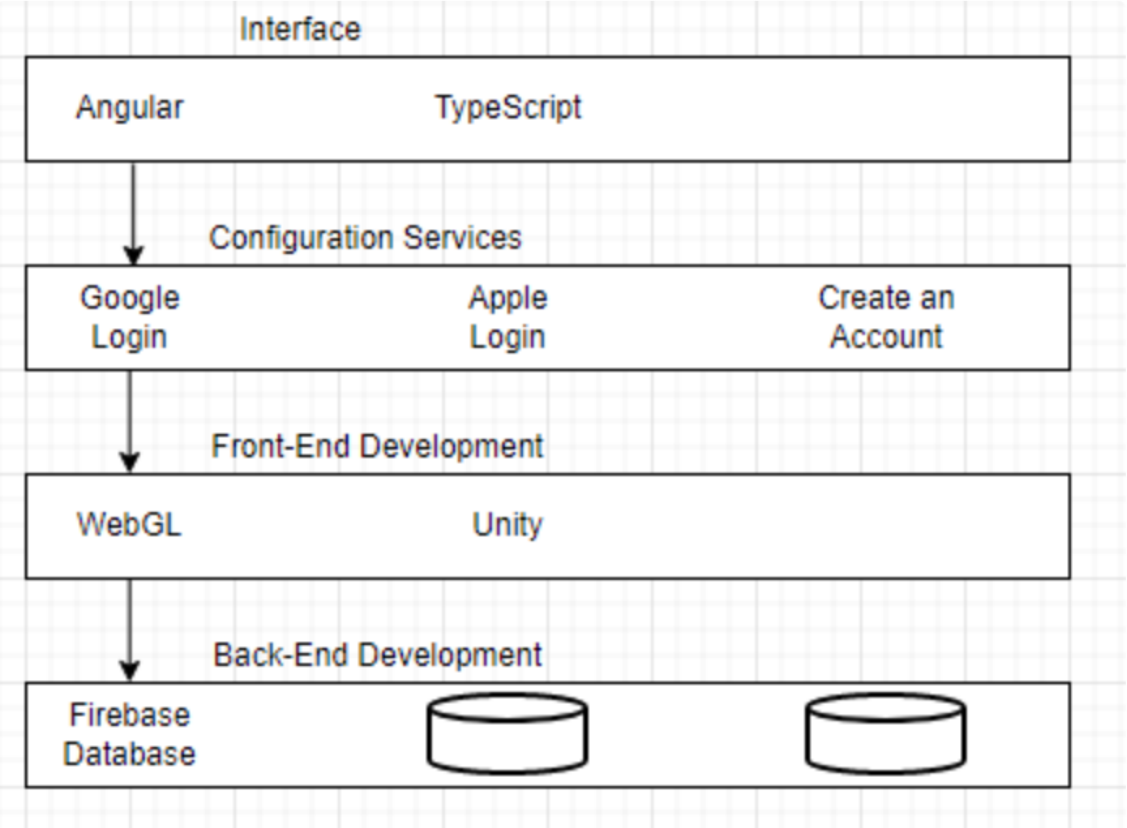
Goals

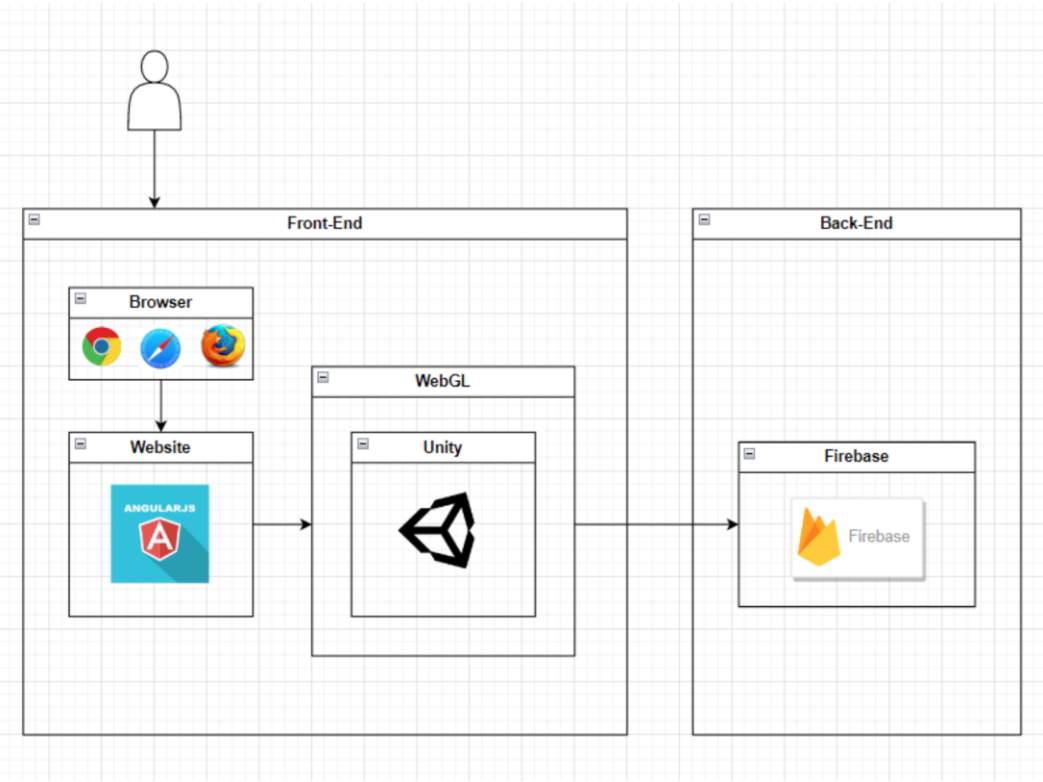
Upon completion of the project, our vision was to have visitors be transported to a fully immersive world where they can explore exhibits, interact with artifacts, and learn a lot more about the history and culture of Niagara-on-the-lake Museum. The team believed that we have truly put together a viable product and see great market potential. Here is a small walkthrough of the process through which the goals were identified. The project was kicked off by deciding to go with an agile framework, specifically known as Scrum, where the design thinking approach was used by first empathizing with the target user. After speaking with the folks at the museum, the team learned that the museum wanted to increase exposure to younger audiences. So, during the ideation phase, a wide range of ideas were generated for the virtual museum experience. The team explored different types of exhibits, interactive elements, and many different types of technologies that could potentially be used to achieve this. Finally, after visiting the museum, the team began narrowing down on which goals to focus on.

Features



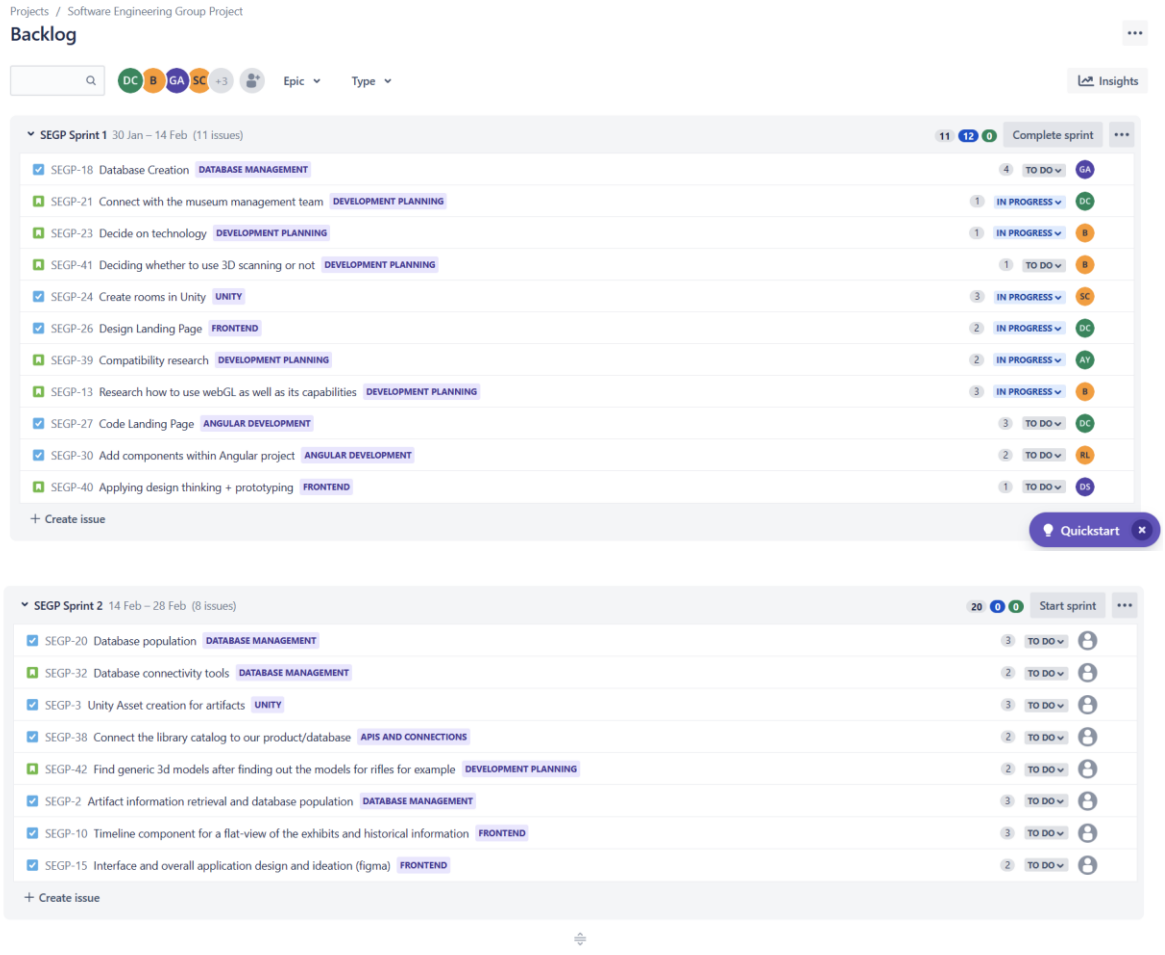
Technologies Used

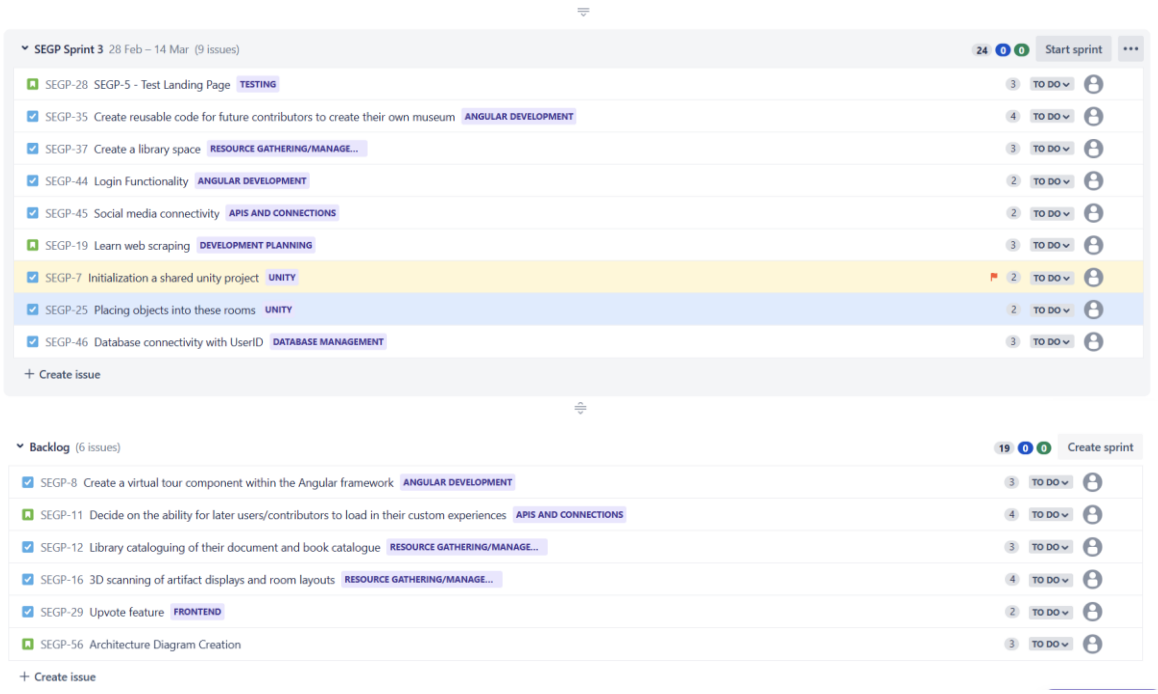




Initial Backlog

To start the planning process for a virtual reality museum application, it was crucial to establish an initial backlog that lists the specifications and features that will be required. The following things were considered for the initial backlog: user stories, technical requirements, timeline, and budget (if any). Listed below is the backlog the product owner came up with and further story pointed with the development team. Time and resource management can also be seen.




SEGP Sprint 3 28 Feb – 14 Mar (9 issues)
24 0 0
Start sprint
SEGP-28 SEGP-5 - Test Landing Page TESTING 3 TO DO
SEGP-35 Create reusable code for future contributors to create their own museum ANGULAR DEVELOPMENT 4 TO DO
SEGP-37 Create a library space RESOURCE GATHERING/MANAGE... 3 TO DO
SEGP-44 Login Functionality ANGULAR DEVELOPMENT 2 TO DO
SEGP-45 Social media connectivity APIS AND CONNECTIONS 2 TO DO
SEGP-19 Learn web scraping DEVELOPMENT PLANNING 3 TO DO
SEGP-7 Initialization a shared unity project UNITY 2 TO DO
SEGP-25 Placing objects into these rooms UNITY 2 TO DO
SEGP-46 Database connectivity with UserID DATABASE MANAGEMENT 3 TO DO
Create issue
Backlog (6 issues)
19 0 0
Create sprint
SEGP-8 Create a virtual tour component within the Angular framework ANGULAR DEVELOPMENT 3 TO DO
SEGP-11 Decide on the ability for later users/contributors to load in their custom experiences APIS AND CONNECTIONS 4 TO DO
SEGP-12 Library cataloguing of their document and book catalogue RESOURCE GATHERING/MANAGE... 3 TO DO
SEGP-16 3D scanning of artifact displays and room layouts RESOURCE GATHERING/MANAGE... 4 TO DO
SEGP-29 Upvote feature FRONTEND 2 TO DO
SEGP-56 Architecture Diagram Creation 3 TO DO
Create issue

Projects / Software Engineering Group Project

SEGP Sprint 1

DC B GA SC +3 Epic Type

TO DO 5 ISSUES

Database Creation
DATABASE MANAGEMENT
SEGP-18 4 GA

Deciding whether to use 3D scanning or not
DEVELOPMENT PLANNING
SEGP-41 1 B

Code Landing Page
ANGULAR DEVELOPMENT
SEGP-27 3 DC

Add components within Angular project
ANGULAR DEVELOPMENT
SEGP-30 2 RL

Applying design thinking + prototyping
FRONTEND
SEGP-40 1 DS

IN PROGRESS 6 ISSUES

Connect with the museum management team
DEVELOPMENT PLANNING
SEGP-21 1 DC

Decide on technology
DEVELOPMENT PLANNING
SEGP-23 1 B

Create rooms in Unity
UNITY
SEGP-24 3 SC

Design Landing Page
FRONTEND
SEGP-26 2 DC

Compatibility research
DEVELOPMENT PLANNING
SEGP-39 2 AY

Research how to use webGL as well as its capabilities
DEVELOPMENT PLANNING
SEGP-13 3 B

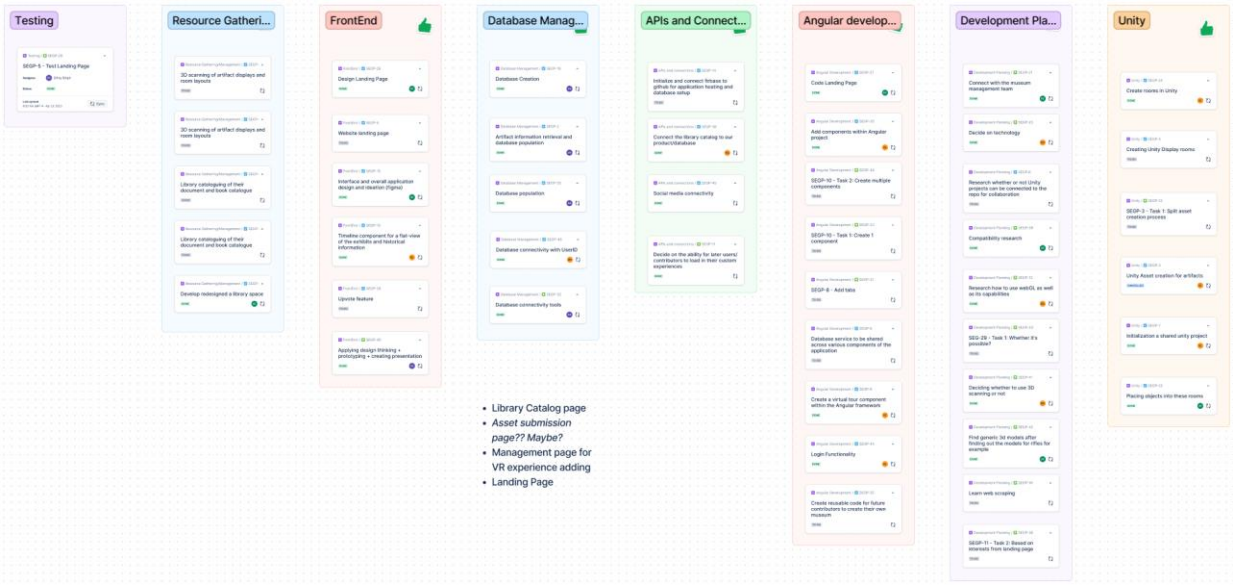
DONE ✓

Sprint Planning

After creating an initial backlog, the next step was to create a plan and design for the project, so that when the team were to execute the first iteration of the product, they would have an end goal in mind.

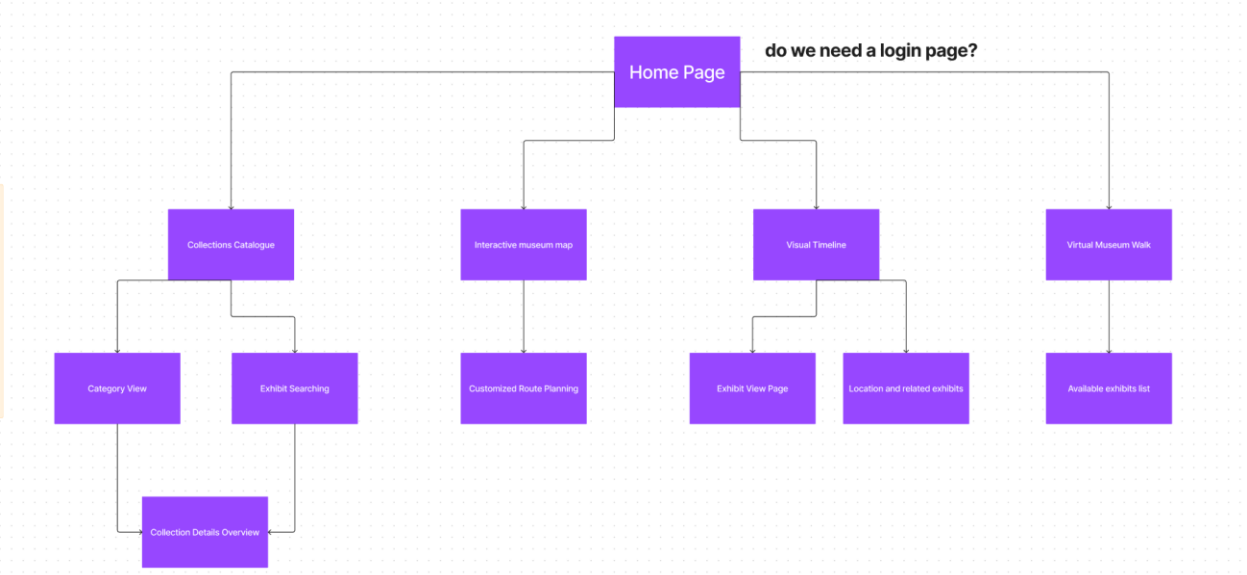
Figma Task Planning

The team used Figma to break up the larger tasks as a kanban board.

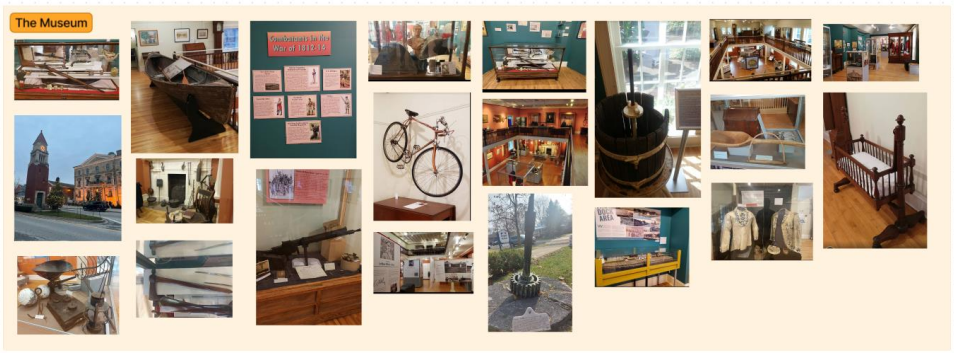


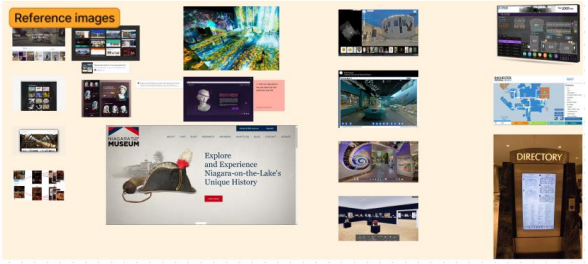
Initial Architecture

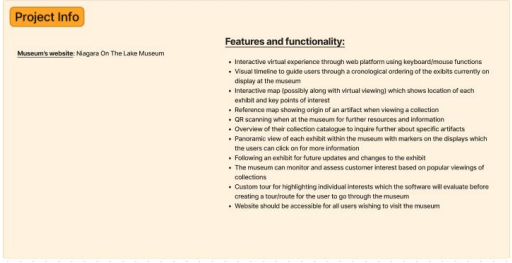
Then the architecture of the website was designed with the main features listed.



The team then decided to pay a visit to the museum so that accurate resource collection can be conducted.

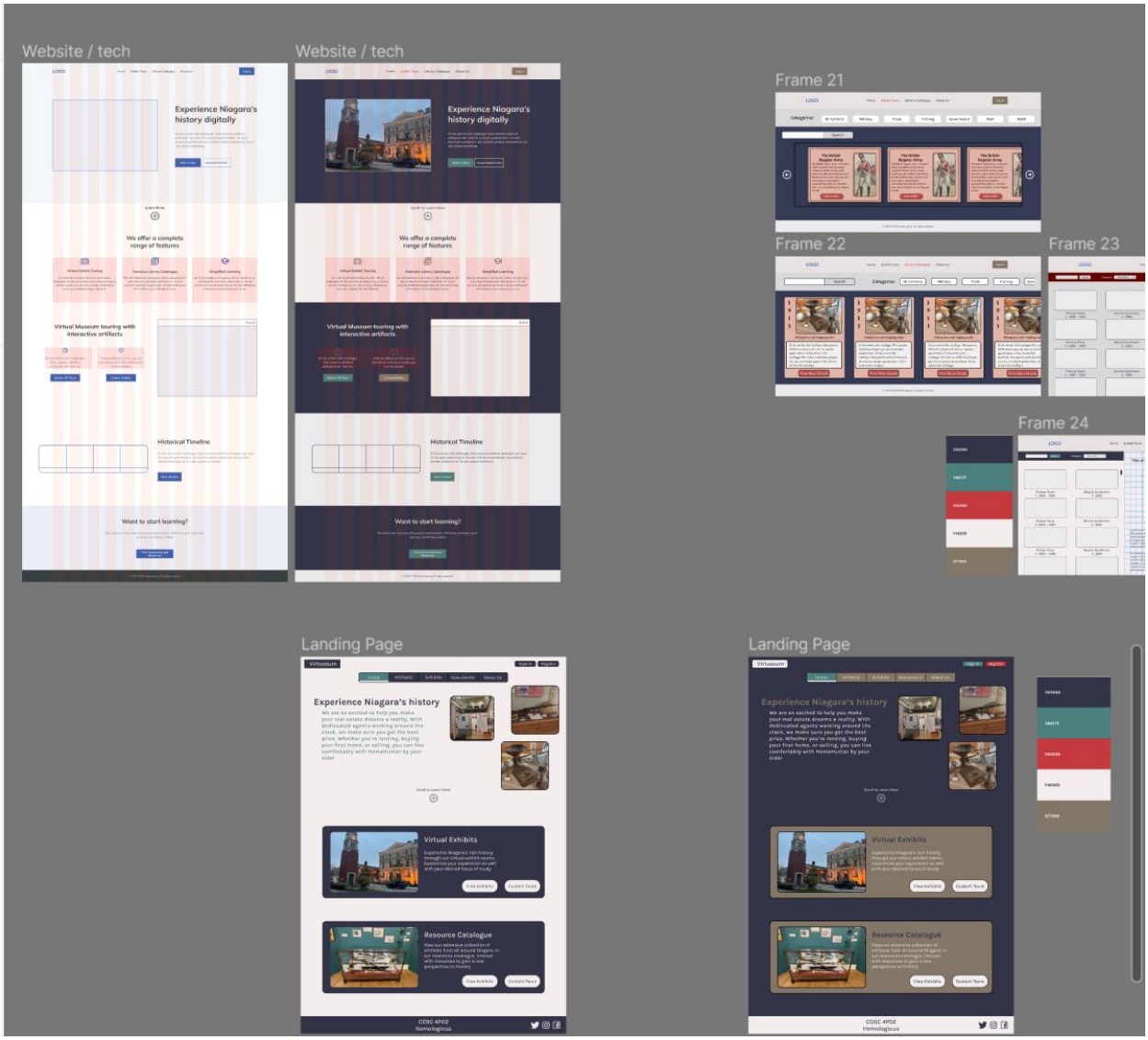


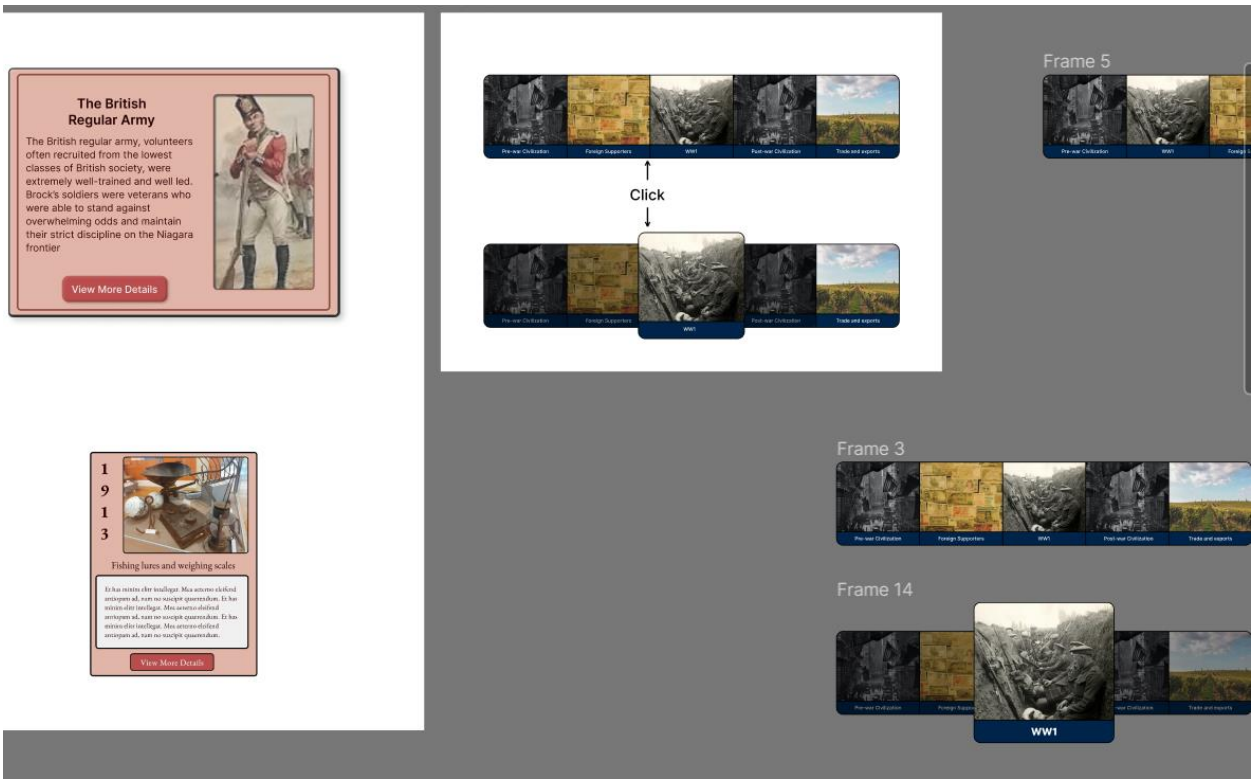




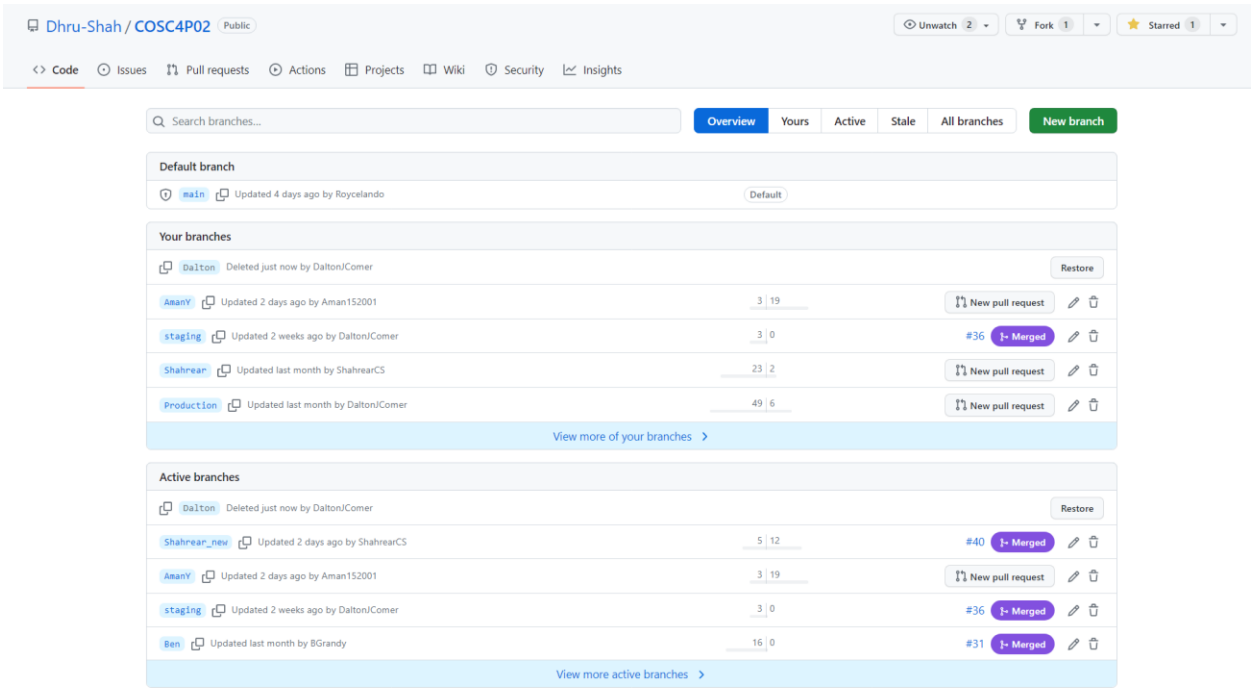
Finally, a list of specified features and functionalities were solidified based on the experience at the museum.

The team then jumped onto Figma, to create the front-end designs and visuals.




The British Regular Army
The British regular army, volunteers often recruited from the lowest classes of British society, were extremely well-trained and well led. Brock's soldiers were veterans who were able to stand against overwhelming odds and maintain their strict discipline on the Niagara frontier
View More Details
Click
Frame 5
Frame 3
Frame 14
WW1
1913
Fishing lures and weighing scales
View More Details

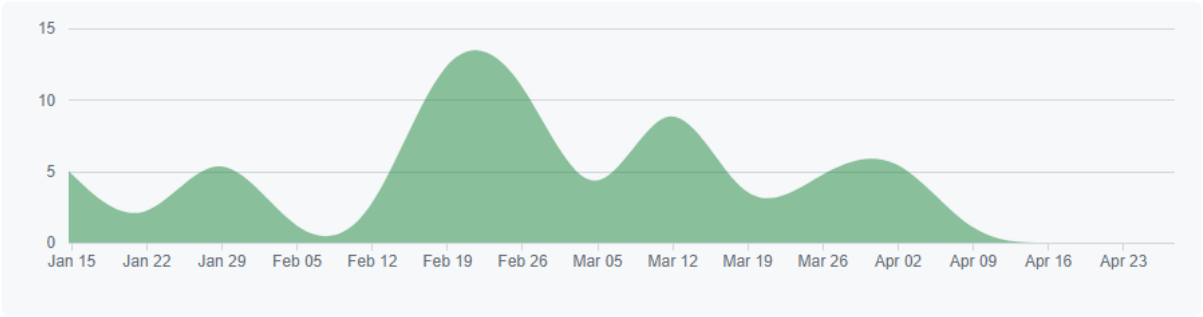
GitHub


Dhru-Shah / COSC4P02
Public
Unwatch 2
Fork 1
Starred 1
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Search branches...
Overview
Yours
Active
Stale
All branches
New branch
Default branch
main Updated 4 days ago by Roycelando Default
Your branches
Dalton Deleted just now by DaltonJComer Restore
AmanY Updated 2 days ago by Aman152001 3 19 New pull request
staging Updated 2 weeks ago by DaltonJComer 3 0 #36 Merged
Shahrear Updated last month by ShahrearCS 23 2 New pull request
Production Updated last month by DaltonJComer 49 6 New pull request
View more of your branches
Active branches
Dalton Deleted just now by DaltonJComer Restore
Shahrear_new Updated 2 days ago by ShahrearCS 5 12 #40 Merged
AmanY Updated 2 days ago by Aman152001 3 19 New pull request
staging Updated 2 weeks ago by DaltonJComer 3 0 #36 Merged
Ben Updated last month by BGrandy 16 0 #31 Merged
View more active branches

Jan 15, 2023 – Apr 28, 2023

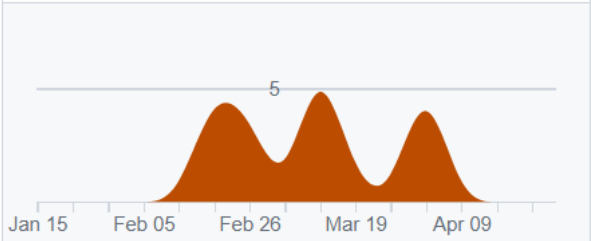
Contributions: **Commits**

Contributions to main, excluding merge commits and bot accounts



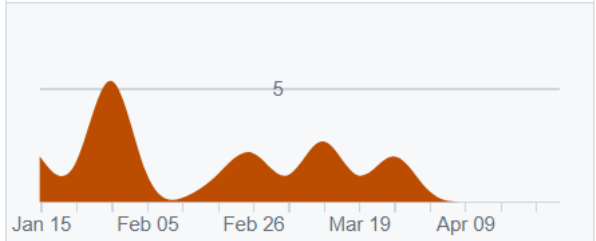
ShahrearCS #1

23 commits 1,535 ++ 921 --



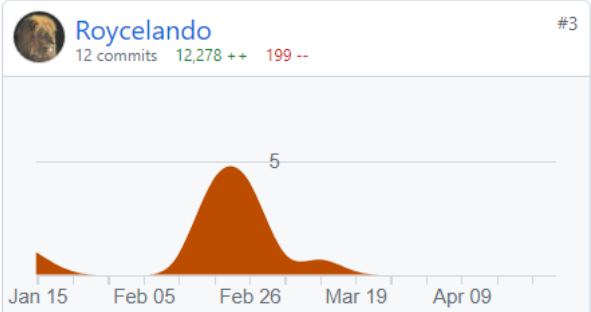
DaltonJComer #2

21 commits 16,231 ++ 2,906 --



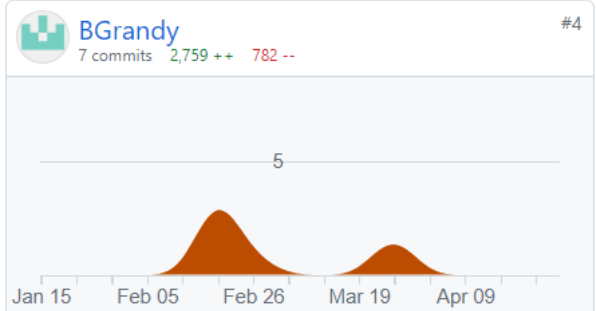
Roycelando #3

12 commits 12,278 ++ 199 --



BGrandy #4

7 commits 2,759 ++ 782 --



Dhru-Shah #5

3 commits 78 ++ 36 --

Aman152001 #6

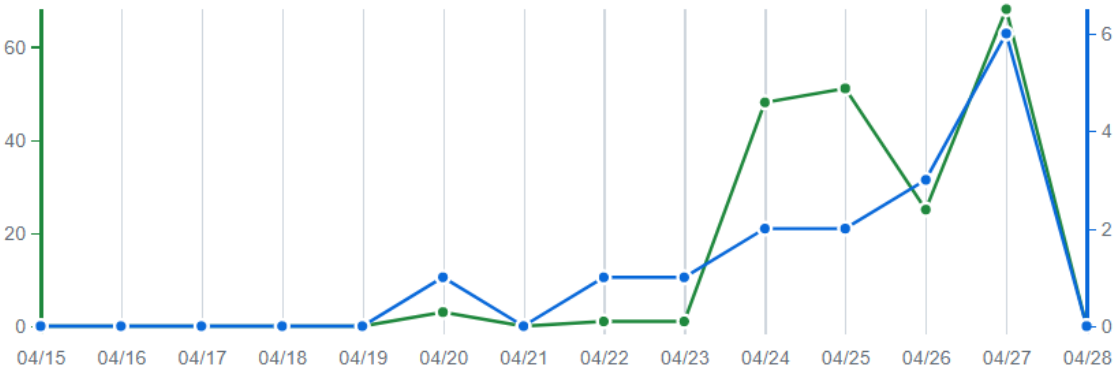
1 commit 1 ++ 1 --

Git clones

11 Clones

6 Unique cloners

Visitors



197 Views

11 Unique visitors

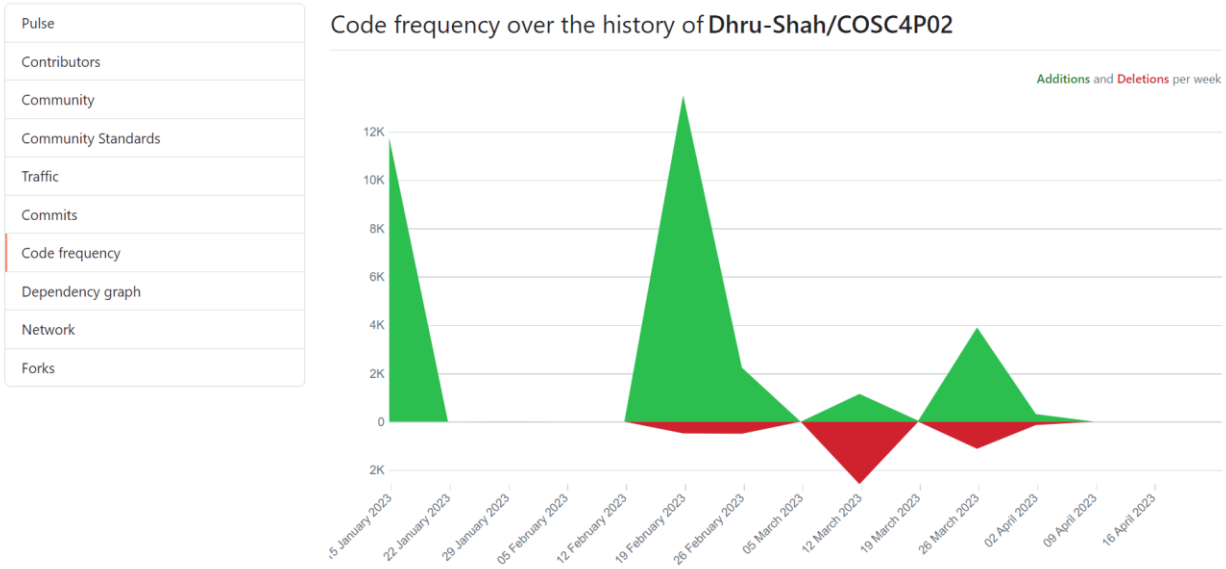
Referring sites

Site	Views	Unique visitors
github.com	88	6
statics.teams.cdn.office.net	3	2

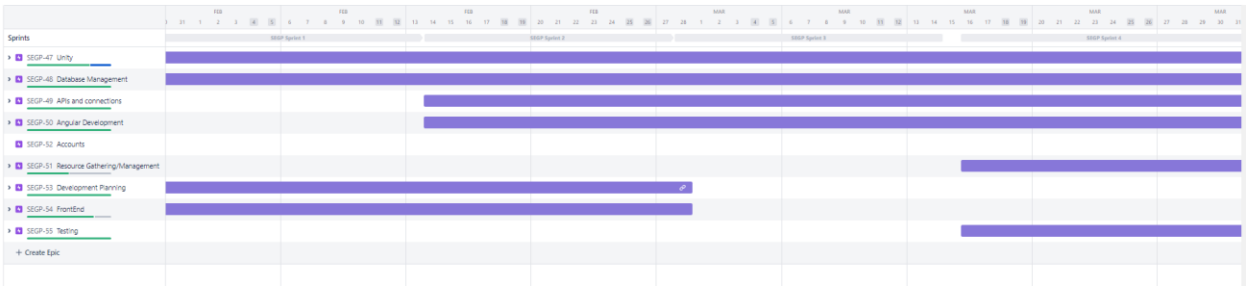
Popular content

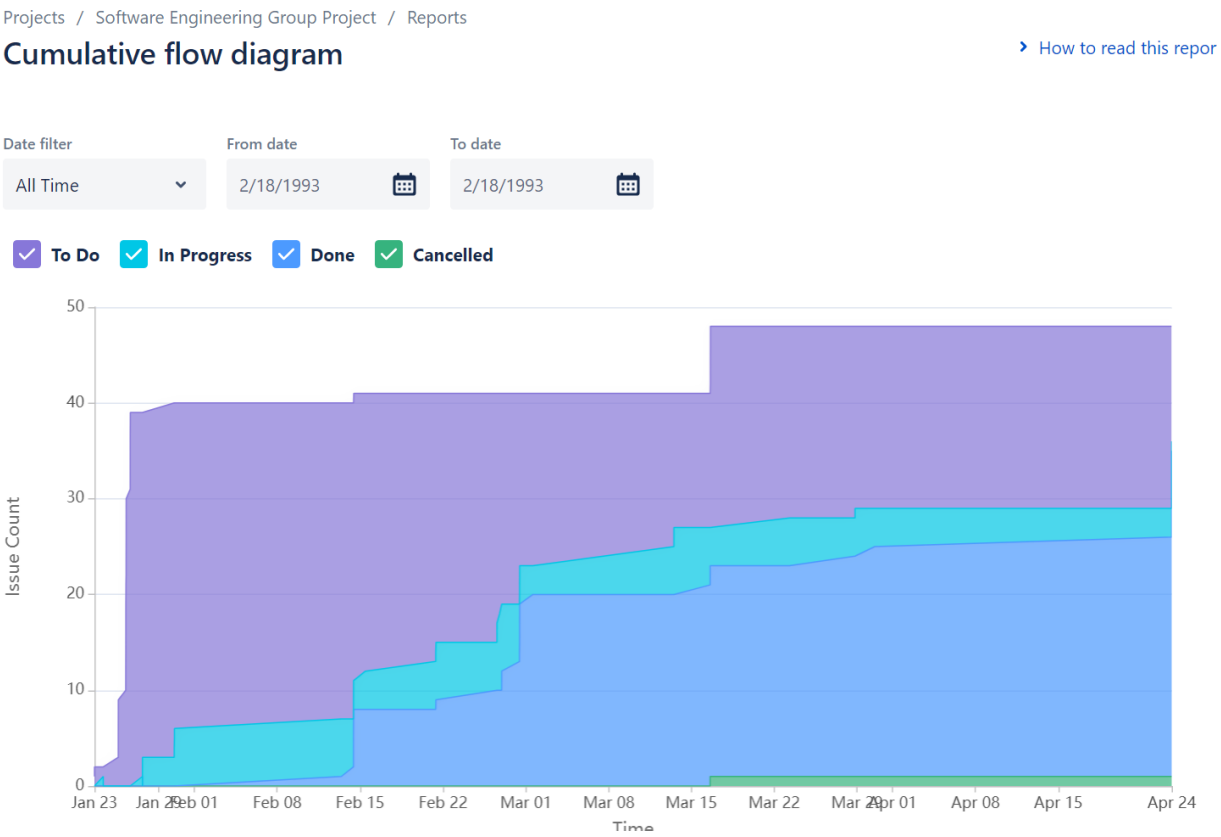
Content	Views	Unique visitors
GitHub - Dhru-Shah/COSC4P02: Te...	39	11
Pull requests	13	3
Contributors to Dhru-Shah/COSC4...	12	4
GitHub - Dhru-Shah/COSC4P02 at ...	9	4
Dhru-Shah/COSC4P02 at AmanY	7	3
Commits	7	2
COSC4P02/documents at main	5	4
Commits	5	2
Staging by Aman152001 · Pull Req...	5	2
Code frequency	4	2


Pulse
Contributors
Community
Community Standards
Traffic
Commits
Code frequency
Dependency graph
Network
Forks
15
10
5
0
04/30 05/21 06/11 07/02 07/23 08/13 09/03 09/24 10/15 11/05 11/27 12/18 01/08 01/29 02/19 03/12 04/02 04/23
8
6
4
2
0
0 0 1 0 2 2 2
Sunday Monday Tuesday Wednesday Thursday Friday Saturday


Pulse
Contributors
Community
Community Standards
Traffic
Commits
Code frequency
Dependency graph
Network
Forks
Code frequency over the history of Dhru-Shah/COSC4P02
Additions and Deletions per week
12K
10K
8K
6K
4K
2K
0
2K
15 January 2023
22 January 2023
29 January 2023
05 February 2023
12 February 2023
19 February 2023
26 February 2023
05 March 2023
12 March 2023
19 March 2023
26 March 2023
02 April 2023
09 April 2023
16 April 2023

JIRA


Sprints
SEGP-47 Unity
SEGP-48 Database Management
SEGP-49 APIs and connections
SEGP-50 Angular Development
SEGP-52 Accounts
SEGP-51 Resource Gathering/Management
SEGP-53 Development Planning
SEGP-54 FrontEnd
SEGP-55 Testing
+ Create Epic



Workload Management

The following procedures were useful for managing workloads within the Agile software engineering team building a virtual reality museum:

1. Decision to use Scrum or Kanban: Software development frequently makes use of the Agile techniques Scrum and Kanban. While Kanban is based on visualising the workflow and streamlining the process, Scrum emphasises time-bound sprints. Team Homologicus chose Scrum.
2. User stories: These tasks are brief, quantifiable objectives that define a particular product feature or capability. Prior to the start of each sprint or cycle, these stories were established based on the product backlog.

3. Task prioritisation: The tasks were sorted according to significance and urgency. Depending on the expertise and skill set, the tasks were assigned to respective team members.
4. Task division: The work was then further divided into smaller, more manageable chunks that were to be finished in a sprint or cycle. The team was able to work more efficiently this way and monitor the progress as a result.
5. Weekly stand-up meetings: These touchpoints were held to make sure that everyone is on the same page and any problems are quickly resolved. These discussions were focused on progress, difficulties, and subsequent measures to solve said problems.
6. Utilization of collaboration tools: To organise projects and track progress, the team used collaboration platforms specifically Jira and Confluence. Work was delegated, progress was monitored, and team members were able to interact with each other and the work at hand using these tools.
7. Retrospectives: These bi-weekly meetings were conducted on a frequent basis in order to assess team progress and pinpoint areas for development. This assisted the group in continuously enhancing and streamlining the operations.

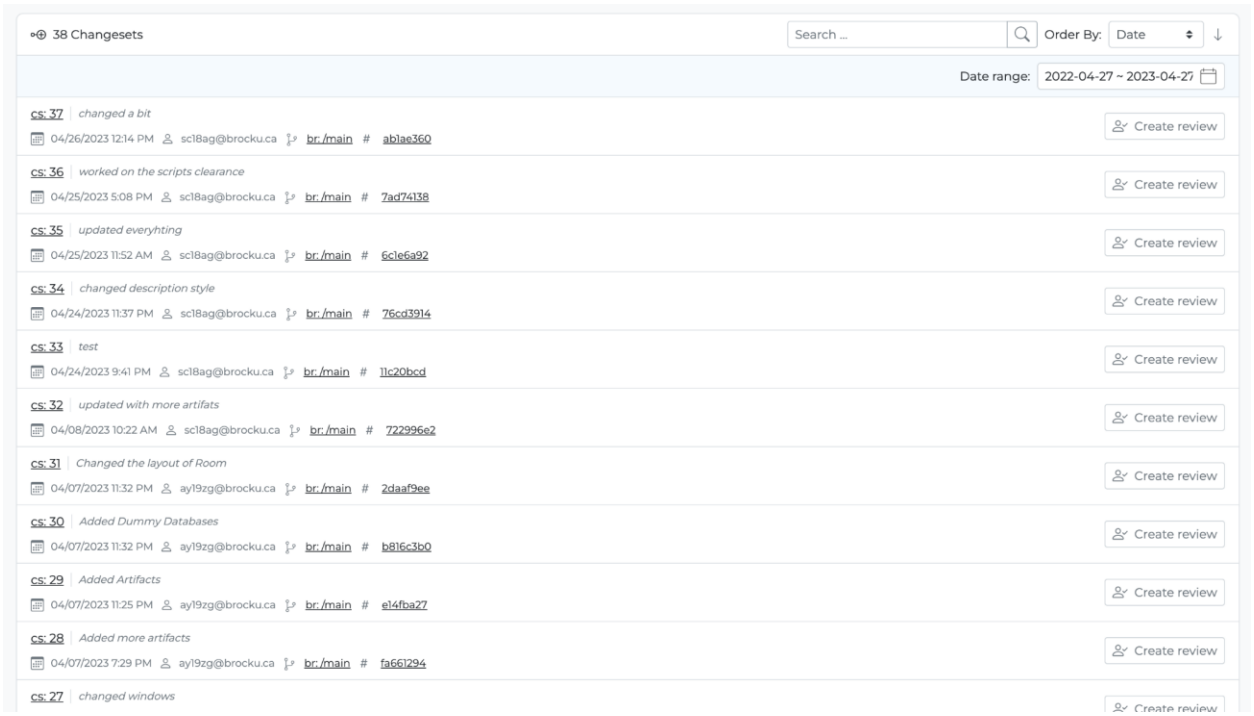
In conclusion, Team Homologicus was able to successfully manage the workload and build a virtual reality museum using Agile methodology by adhering to these practices.

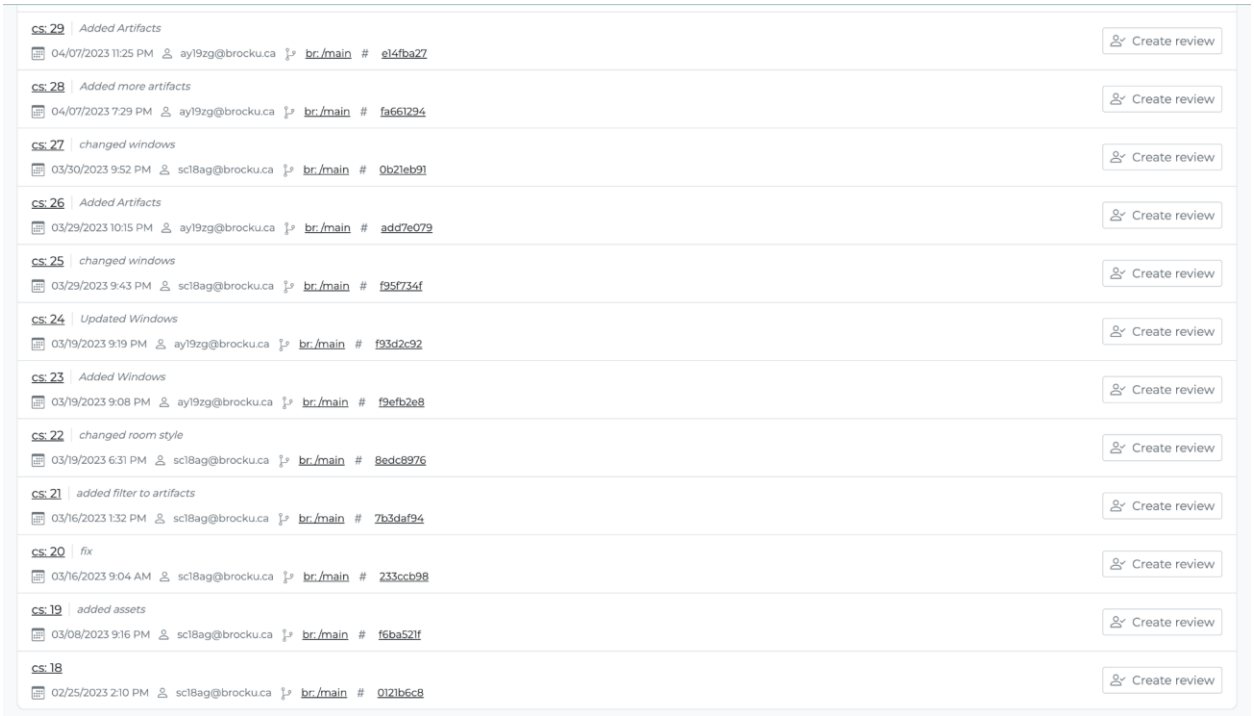
User Manuals

Interactive Exhibit

Shahrear and Aman collaborated on creating a Unity Room (Virtual Exhibit Tour) using PlasticSCM software. The project involved creating a 3D room and adding artifacts to it, followed by making the room browser compatible and ensuring that the artifacts were deploying correctly on the web browser.

Their Changeset in PlasticSCM:





The first step in the development stage was to ensure that the Unity Room was being hosted on the web browser. To achieve this, WebGL was used, which allowed the Unity Room to be viewed on both mobile and desktop devices. Once this was done, artifacts were added to the room and it was ensured that they were deploying correctly on the web browser.

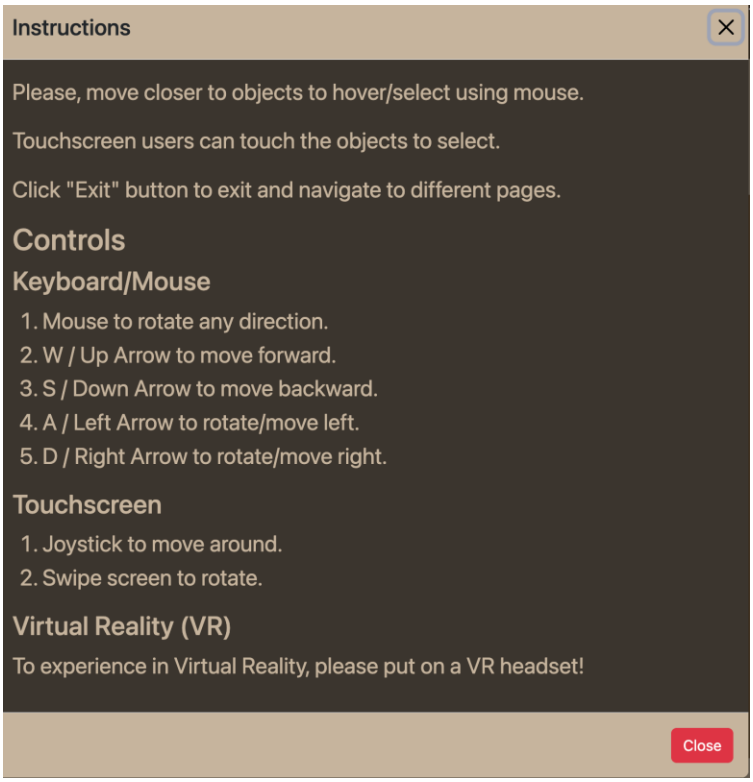
Functionality was added to the Unity Room by incorporating a C# script. This script allowed the user to hover or click on an artifact, changing its color to green or yellow depending on whether the artifact was being hovered over or clicked. Furthermore, when an artifact was clicked, the name of the object was displayed in the console of the Unity Application.

To enhance the user experience, dummy database entries were added. When an artifact was clicked on, an information window was displayed that included the date and a description of the artifact.

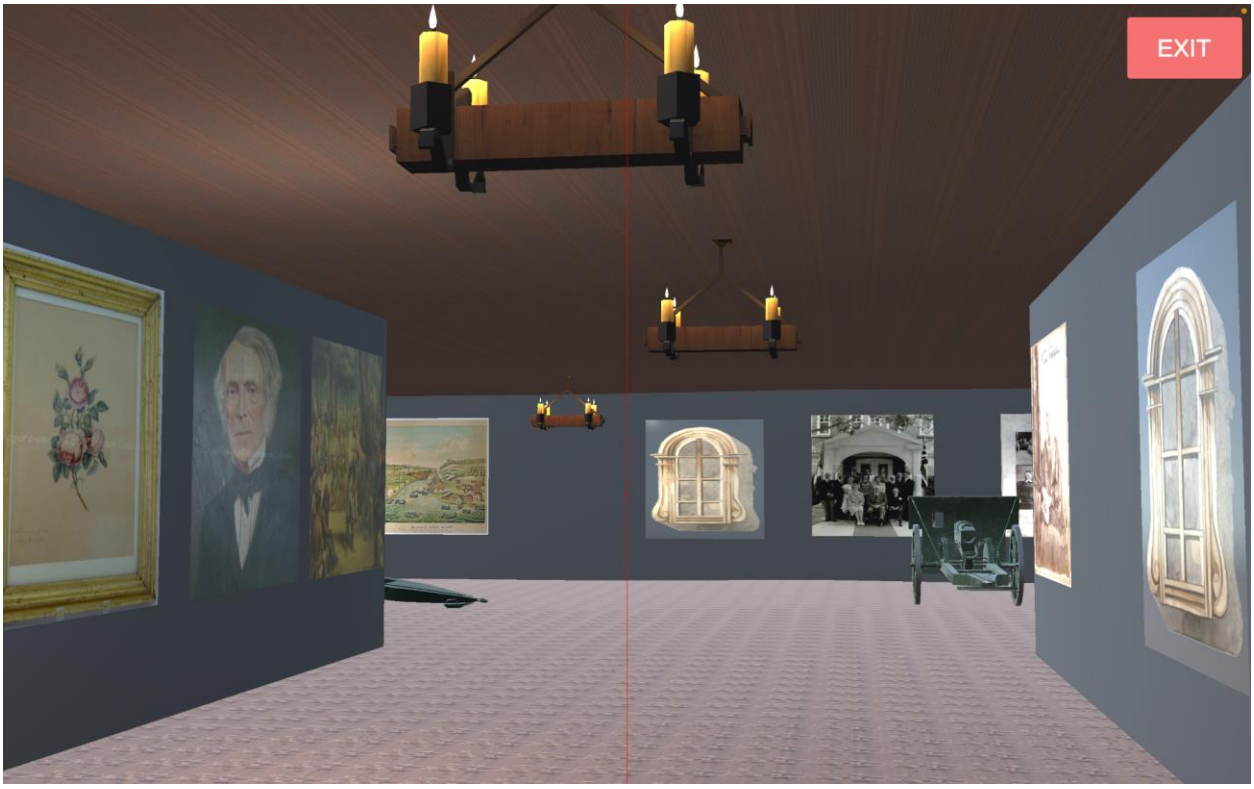
Connectivity between Unity and Angular, as well as Unity and Firebase, was also established during the project. This allowed for seamless integration between the Unity Room and other components of the project.

In the final stages of the project, the layout and lighting of the Unity Room were improved, ensuring that it matched the design of the website and database was connected to firebase database.

This is what the final look of the virtual exhibit tour looks like. The instruction popup window before entering to the museum:

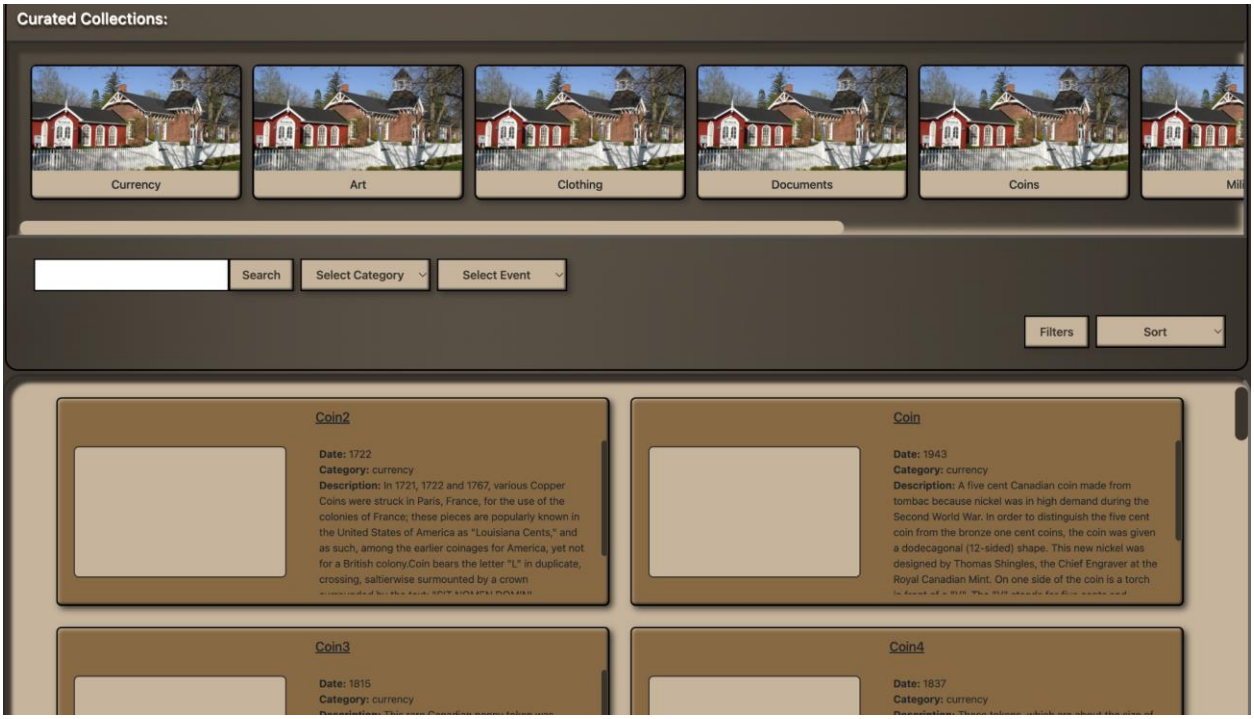


Final exhibit room:



Library Catalogue

After the firebase was initialized and populated, the library catalogue was able to be populated by pulling the information into an Angular model object. This allows for easy traversal as well as scalability in terms of adding new information.

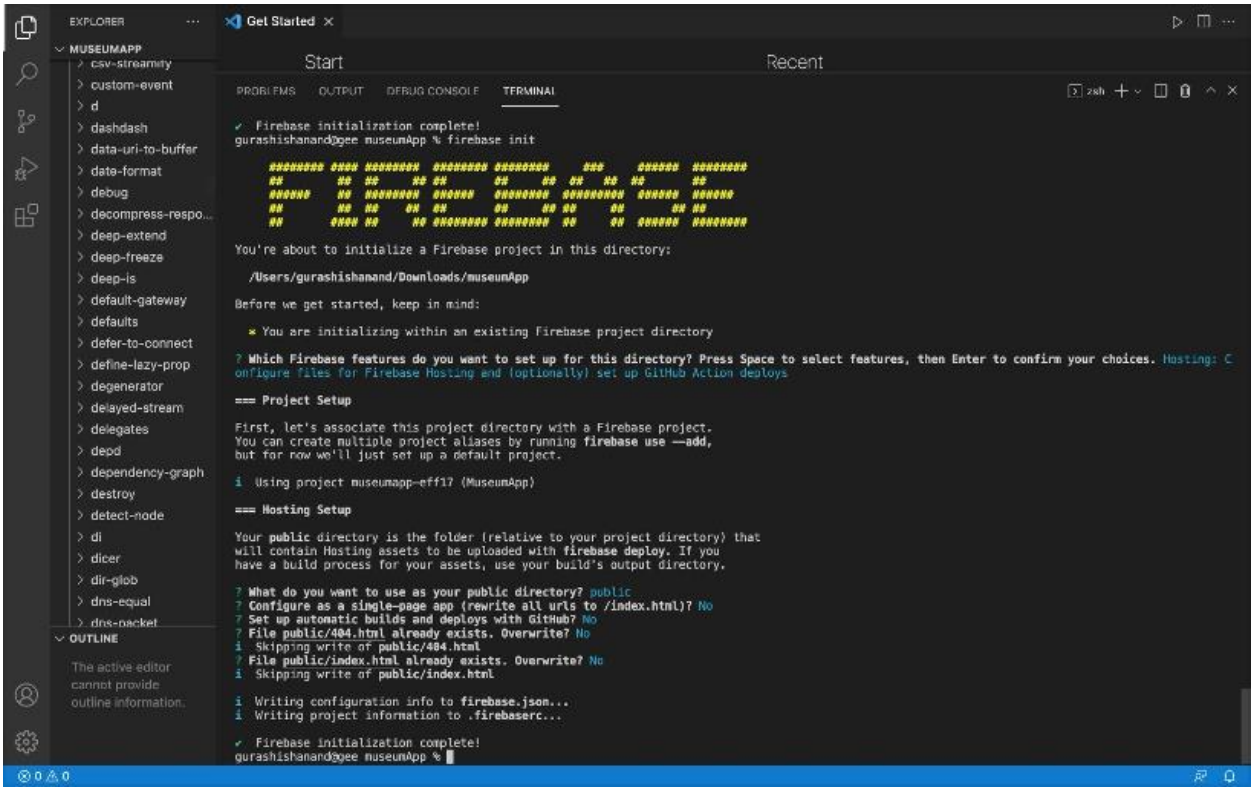


Deployment (Firebase deploy/hosting)

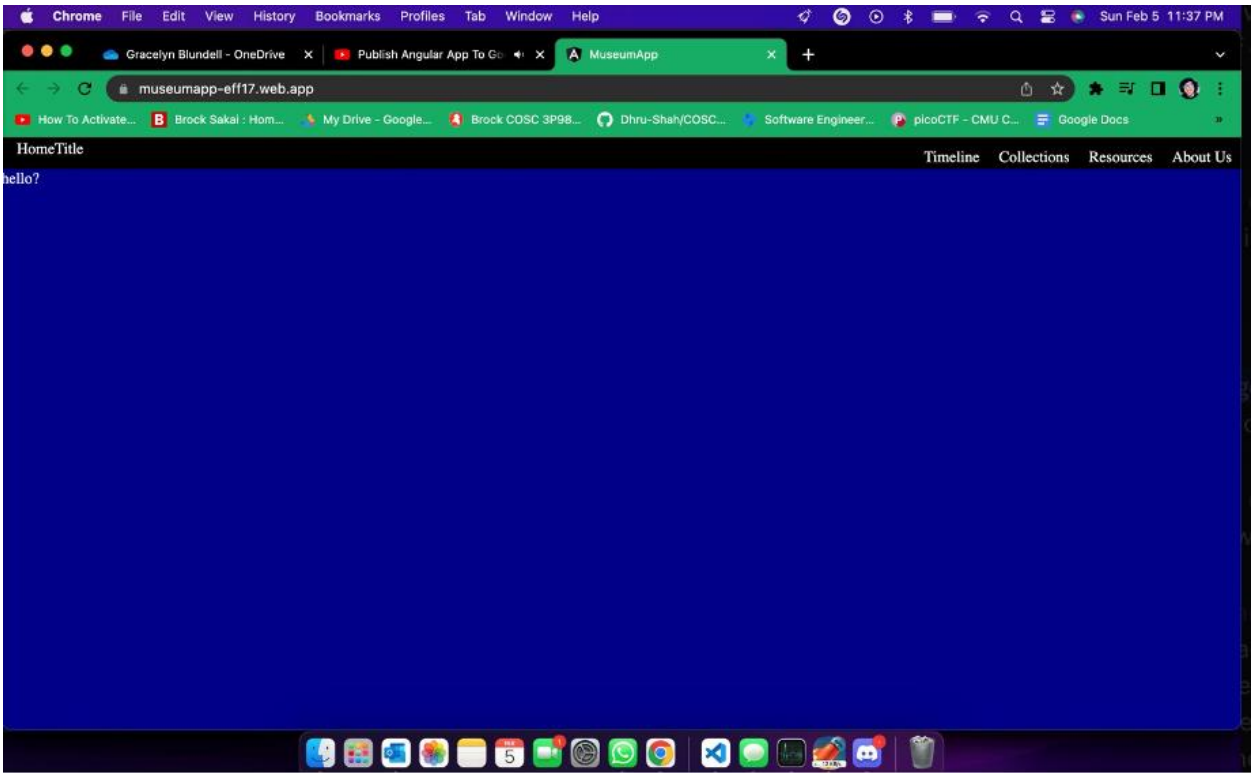
Deployment iterations (Versions) (Done by Gurashish Anand)

To get started with firebase, I started off by learning how to deploy and project and hosted a random angular project we had from our early work , and then working our way to deploying different iterations of our projects.

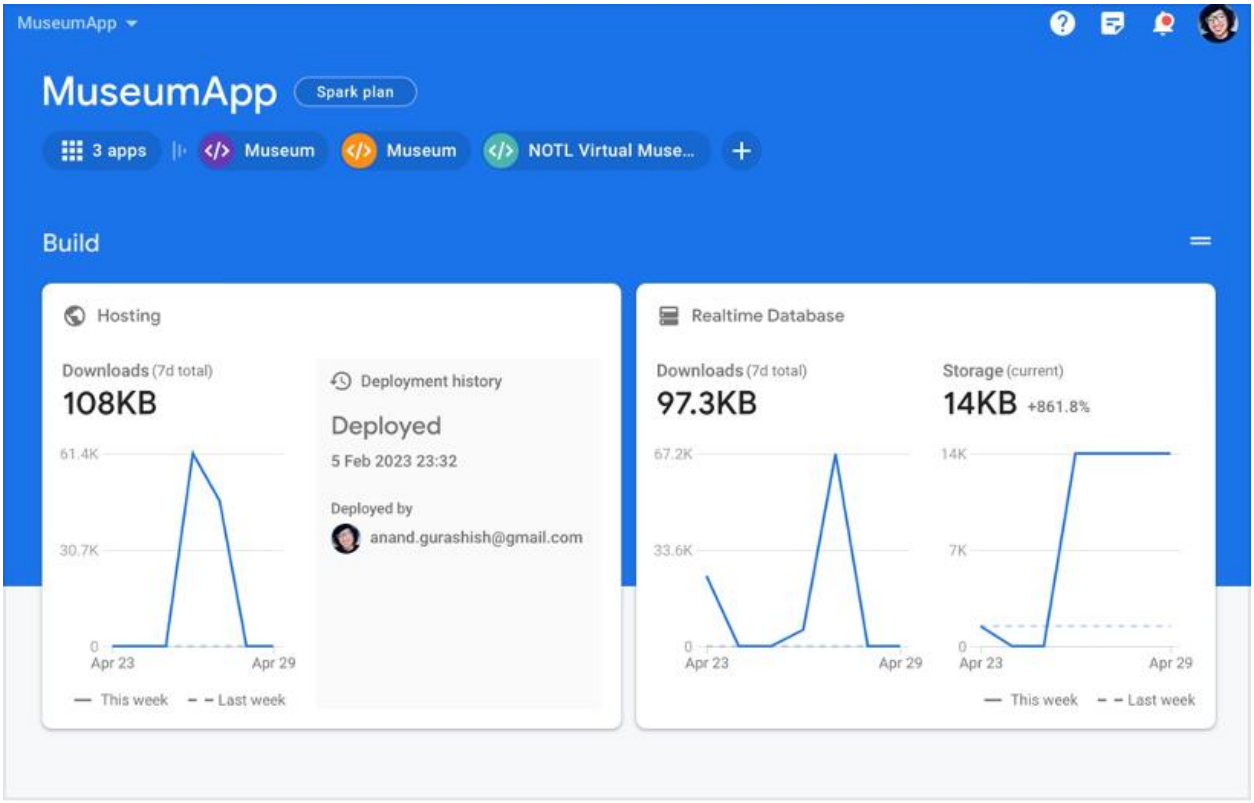
The following screenshot shows the first firebase initialization:-



First practice webpage deployed on Feb 5:-

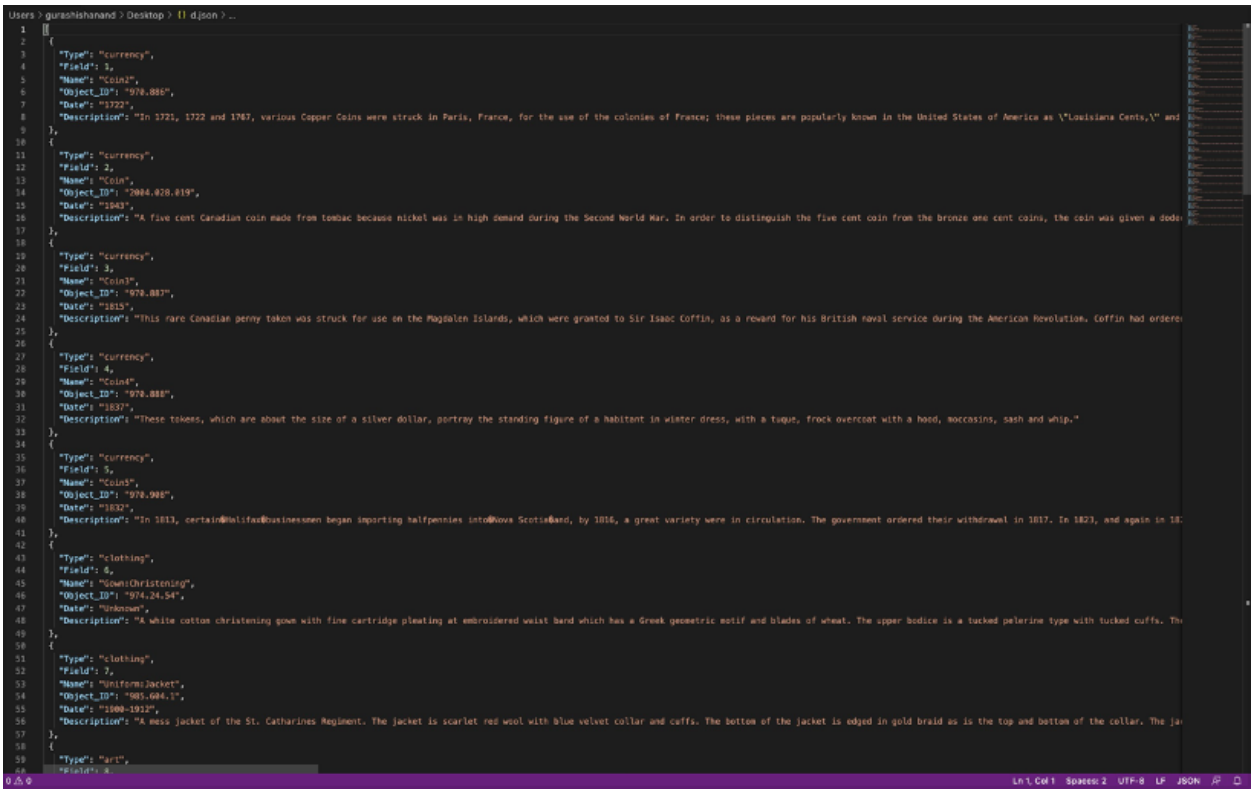


After this stage, we started to have problems with connecting the firebase directly to the main branch of our github repository because for some reason, firebase couldn't scan the repository properly for projects even after successful authentication. So, we had to take a static approach which requires us to manually deploy the web-application for every single updated iteration of our web app. That is a strong reason that we had different multiple hosted links for iterations.

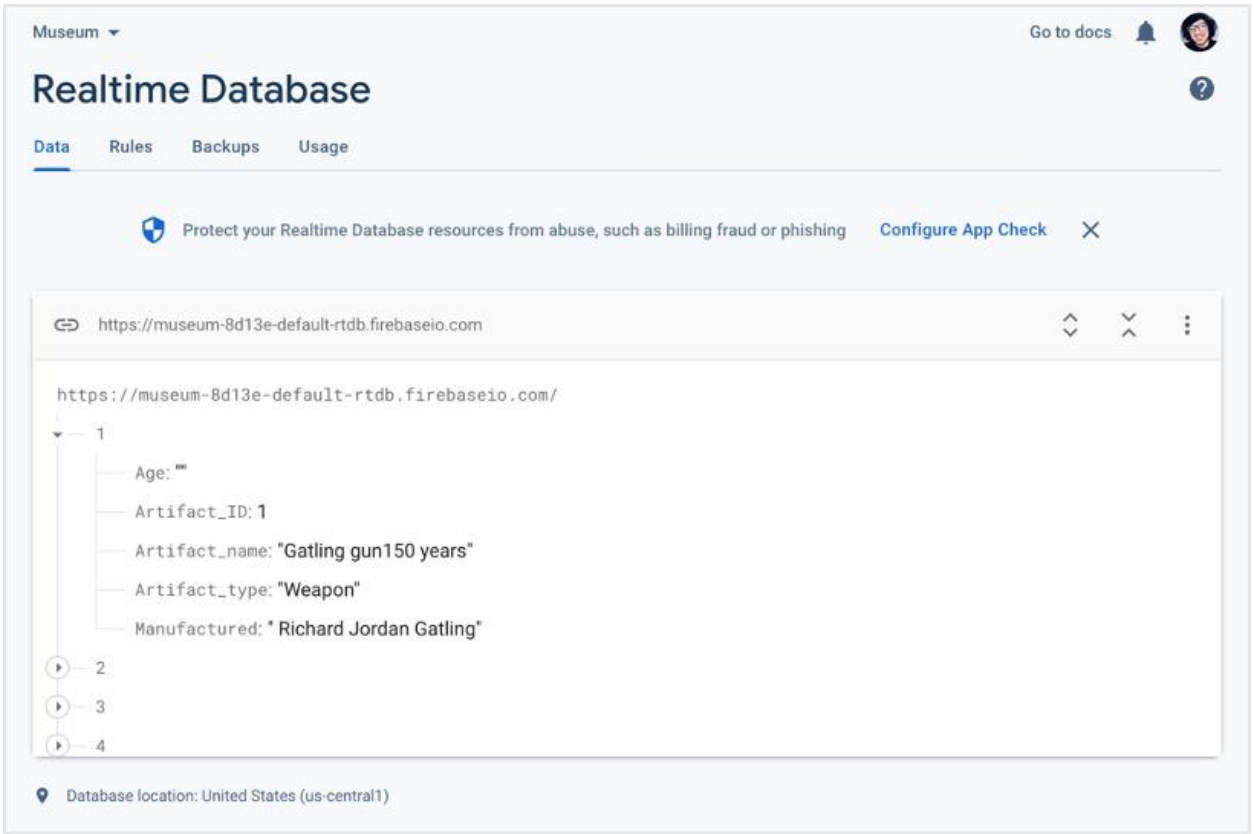


We took a similar approach with the realtime database, while we did not have access to the museum's database, just to get familiar with the technology and library catalog connectivity, I went ahead and added dummy entries by writing a Json file and uploading it to firebase.

Latest version of the json file:-



Hosted dummy entries at the time:-



After we were successfully able to connect the library catalog and unity to our firebase, we started using real data of the museum entries.

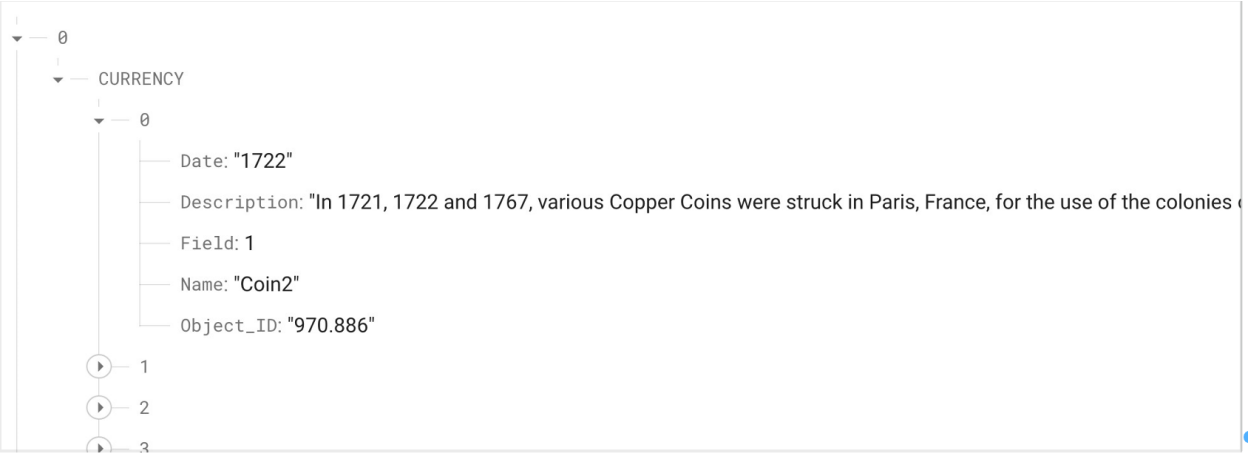
Then we started to get errors and our project and we also had to add the authentication feature, so I took part in pair programming sessions with Dalton and Ben to figure this out.

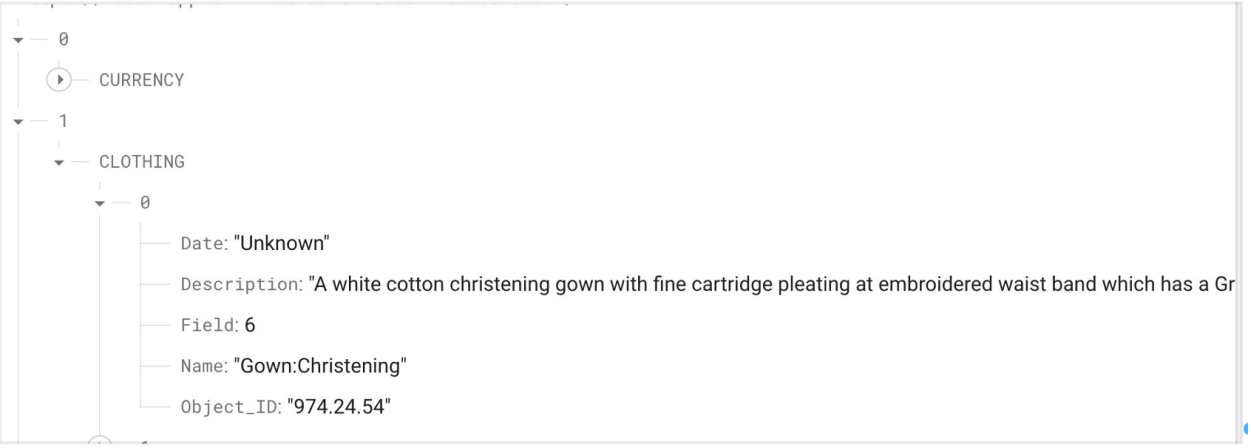
While reaching the later stages of our development, I took part with Aman in pair programming, in which we browsed the museum's database for artifacts that we would use. On one side, Aman added artifact models of our selections in unity, and on the other hand, I was making sure that all its information is stored in the realtime database.

Just when we thought our project was almost complete, we were missing exhibits , so I made changes to the realtime database to present the artifact entries as per their category but modified the json file.

Live updates

Current realtime database:-





Database Model

Decide whether or not to use 3D scanning

- Naser scanning artifacts using Brock 3D scanner
    - Most likely will be the best option. We can see what kind of quality and format the scans come in and try to implement.
- Monster Mash
    - Good alternative but it has some issues

- Has difficulty with complex objects such as a gun.

- Looks good head on

- But has issues filling in the sides. Can be fixed in blender

Research how to use WebGL and its capabilities

- Build the Unity project using WebGL
    - https://learn.unity.com/tutorial/how-to-publish-for-webgl?signup=true#
- Calling JS functions through Unity and vice-versa
    - https://docs.unity3d.com/Manual/webgl-interactingwithbrowserscripting.html

Test Documents

Security testing

Finding and testing potential flaws that an attacker could exploit is part of performing security testing for the museum virtual reality application for Niagara-on-the-Lake Museum. The following actions can be taken to conduct security testing for the virtual reality application in upcoming sprints.

1. Determine potential attack vectors: The first step is to determine the potential attack routes that might be utilised to take advantage of application vulnerabilities. Network assaults, attacks on online apps, and virus attacks are a few of the frequent attack vectors for virtual reality applications.
2. Conduct a Security Assessment: Perform a thorough security assessment of the application, which should include code review, vulnerability scanning, and penetration testing. This will make it easier to find any security flaws that attackers might use.
3. Test authentication and authorization: Verify the security of the application's authentication and authorization processes. This entails checking the login information, the access restrictions, and the implementation of user roles and permissions.
4. Verify the security of the data: Make sure that private information is adequately encrypted, kept in a safe place, and kept hidden from unauthorized users. This entails scanning for security holes in data transport and storage.
5. Test of vulnerabilities in third-party components: Verify that any third-party libraries or components used by the application are current and free of known vulnerabilities by conducting a vulnerability test on them.
6. Workers education: Provide security best practises and procedures training to museum workers. They will learn how to handle sensitive information, recognise and report security concerns, and

react to security incidents. These procedures can assist in ensuring the security and safety of virtual reality museums for both personnel and tourists.

Unit testing

Testing different parts or pieces of the programme to make sure they work properly is known as unit testing for virtual reality museum applications. The following instructions can be followed in upcoming sprints for unit testing.

1. Identification of units: The first stage is to determine which specific parts or units of the virtual reality application for museums need to be evaluated. These might include the application's user interface, 3D models, interactive elements, and other features.
2. Write test cases for each unit: Once the units have been determined, test cases can be written for each unit. The expected behaviour of the component and the results of a successful or unsuccessful test should be described in a test case. For a 3D model, checking that it loads properly and appears as expected in the virtual environment might serve as a test case.
3. Use a testing framework: A testing framework can assist with automation of the testing process and make it simpler to run and manage tests. Virtual reality apps can be tested using a variety of frameworks, including Unity Test Runner and VRaptor.
4. Run the tests: Tests will be run to see if each component of the museum virtual reality application is working properly after developing the test cases and creating a testing framework. Any problems or bugs that are discovered during testing can be recorded and fixed.
5. Iterate and enhance: After the preliminary round of testing is through, the developers can refine the test cases as well as the application itself. This will make it easier to guarantee the accuracy and functionality of the virtual reality application for museums. It's vital to remember that evaluating a museum virtual reality application involves more than just unit testing. To make

sure the application is functional and meets the demands of its users, other testing methods, such as user acceptability testing and integration testing, could also be required.

Stability testing

In order to ensure that the VR experience is stable and uniform for users across all devices and platforms, stability testing is carried out in virtual reality museums. The following steps will help the team undertake stability testing in future sprints.

1. Define the testing parameters: Select the VR museum's features that need to be tested, along with the platforms and devices that are in the scope.
2. Make test cases: Come up with a set of test cases that cover all the VR museum's desired features. This could entail evaluating the audio and video components, the VR environment's reliability, and user interactions with the museum's objects.
3. Selecting testing tools: Choose testing instruments that will enable the developers to carry out the test cases and track performance indicators. Frameworks for VR testing as well as tools for tracking performance and analytics may fall under this category.
4. Execute test cases: Run the test cases on various hardware and software platforms to assess the stability of the VR museum. This will assist the team in finding any performance problems, flaws, or glitches that require attention.
5. Recording and Analyzing results: Outcomes are recorded and examined to look for patterns and trends. The outcomes of each test case are recorded and examined. This will enable the team to decide which VR museum exhibits require additional testing and optimisation.
6. Test and enhance iteratively: The VR museum should be tested and improved continuously to increase its reliability and efficiency. This could entail rerunning tests, finding fresh test cases, and putting identified problems into practice. In general, stability testing in VR museums

necessitates a methodical approach and in-depth knowledge of the VR environment and testing instruments available. The team can make sure that your VR museum offers visitors a steady and interesting experience by adhering to these guidelines.

Security Measures

Google Authentication

A secure method of authenticating users and granting them access to the VR museum application is Google authentication, often known as Google Sign-In. To confirm the user's identity and secure their permission to access their data, it employs the OAuth 2.0 and OpenID Connect protocols. A broad explanation of Google authentication is provided below:

1. The user accesses the website and hits the "Sign in with Google" button.
2. With the client ID and secret, the VR application submits a request to Google's authentication server.
3. A distinct token is created by the authentication server and sent back to the application.
4. This token is used by the application to ask Google's API for information about the user's Google profile.
5. The user's data is returned to your application and the user is deemed authorised if the token is valid. Access tokens and refresh tokens are the two types of tokens used in this process. To access the user's data from the Google API, access tokens are used. Long-lasting refresh tokens are used to get a new access token when the current one expires. In general, Google authentication gives consumers a simple and safe way to log into the website, and it gives the user access to their data with their consent.