



Project Report

COSC 4P02

JANUARY 2023

PROFESSOR

Naser Ezzati-Jivan

TEAM MEMBERS

Diego Vergara - 6583819

Michael Wisniewski - 6402176

Zachary McGovarin - 6817019

Haseeb Ahmad _ 6541957

Wayne Bessey - 6527303

Alexander Gibson- 6479364

Micah Rose-Mighty - 6498935

Samuel Onabolu - 6783369



NOTL Museum 360° Panoramic Tour

The NOTL Museum 360° Virtual Tour allows patrons to explore the history of the museum's architecture through an interactive web application.



Technical overview

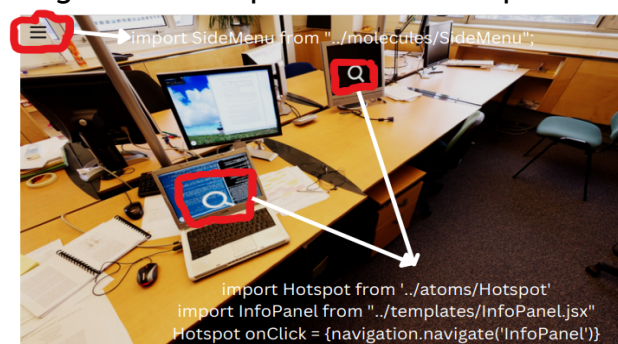
Throughout the scrum process our team decided to use React Native (Expo) as our technology. Firstly, Expo is a very versatile framework as it has the ability to be used on IOS, Android and a web browser. This kind of agility allows the application to be hosted on multiple platforms. Secondly, Expo uses special tags to translate pieces of code between these three platforms, for example the enclosing `<View>` tag can be rendered on any platform as well as styling components such as `<Text>` and `Stylesheets`. Thirdly, Expo allows easy imports of libraries which grants us the ability to create a webpack for our application and deploy on the web. By using this library our team was able to use HTML and CSS tags to make the application for the web browser, this was much quicker and efficient as more people on the team were familiar with HTML/CSS. The panoramic view is powered by React Three Fiber and Three.js, which requires individual support for mobile gestures and interactions. As it stands, the current version of the application is designed as a web application in mind first.

Frontend

The design for the frontend was initially prototyped using the figma web app, this is where our design principles and direction were defined by our scrum master. The designs were then realized within React using various dependencies that are commonly used along with react. Our team decided to use react mainly due to its flexibility. When handling a project of such scale, react's ability to bundle frontend code into custom components allowed us to streamline the development process. By allowing components to be worked on separately and combined as needed, we are able to accomplish various tasks simultaneously and allow us to reuse components as needed. This modularity was essential when attempting various problems and to use towards completing user cases. However, organizing a product of this scale requires communication, teamwork, and skill. With this in mind, we applied the Atomic design principle to categorize our components into Atoms, Molecules, Organisms, Templates

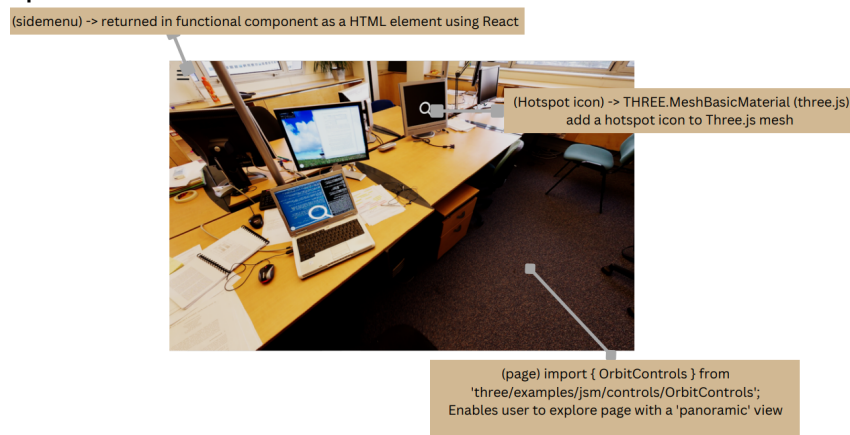
and Pages. Each member of the frontend would then use this system to organize their components for ease of use.

Beginning with atoms, these are the most simple functions that contain simple components that can be used in almost anycase and repeatedly. Molecules are composed of atoms, this folder created slightly larger functions that introduce behavior and graphics to the application. Organisms are even larger functions that incorporate various components from atoms and molecules, often times being the final stage in creating a complete component. Templates inherit more frequently used components that are likely not to change and would not incorporate many of the other previously established components, containing its own fully completed components. Pages represent actual finished web pages that combine some of the components from other stages of the projects' organization to produce our final product.



Here is an example of Atoms, Molecules and a (not shown) Template, from a test file used during development. These would then all be apart of a completed Page that is displayed as shown

The main dependencies we used are various classes from Three.js, various classes from React Three Fiber, React Navigation, Axios and the React library itself. The React library and both Three.js, and React Three Fiber dependencies mainly contributed to the creation of visual components and functionality that are utilized by our product. For example:



React Three Fiber and react navigation are mainly used for functionality, where react navigation allows for simple routing and navigation throughout scenes of our tour, and three fiber is used for building our scenes with re-usable and self contained components that can react to state, and allows for more easy use of three.js. Axios was

used to connect to the backend of our project and utilize api calls that can be created or deleted as necessary, and is then read within the code dynamically. The dependencies chosen for the frontend all work together to allow for a simple, readable project build.

Testing

Automated testing is crucial throughout the scrum process to which our team adopted. By testing we discover situations in which behavior of the software is incorrect, undesirable or does not conform to its specification. Utilizing validation testing we demonstrate to the team and system customer that the software meets its requirements. Using the imported library Jest our team was able to complete testing of the react components created in the project.

The automated tests reside in the `__tests__` folder which are further sub divided into `components`, `navigation` and `screens`. It is crucial for Jest that the folder is called `__tests__` as well as the class we are testing starts with the name of the class followed by `.test.js`, otherwise when running the automated tests with Jest, it will be unable to find the files to test. With this in mind, these sub folders allow easy classification of what files are being tested and how to write the tests. For example, the `.../components/button.test.js` aims to test if the button gets rendered correctly, and if the button when clicked responds to its intended behavior. In another case, `.../screens/tour.test.js` is designed to render the Tour component, this is done by using the `render()` and `expect()` commands to check if Tour is in the application and attempt a test.

Once the tests have been written, it can be run within the `.../panotourapp/` folder where `npm test` is the command used to run the testing. It's imperative that when testing that you install `react-test-renderer` to the exact version of react you are running. In the case of the application react version 16.0.1 was used locally and therefore, `react-test-renderer` must be installed to 16.0.1, this can be accomplished by running `npm install react-test-renderer@16.0.1`.

Backend

For the backend, our goal was to create an API to store and feed information to the front end. To do this, we opted to use the Spring Boot framework. Though none of our group members had previous experience with Spring Boot, its popularity and ease of configuration made it appealing. Spring Boot offers many benefits, including:

simplified dependency management, automated configurations, built-in web security, and more. For the actual storage of data, we used a MySQL database hosted on a VPS. However, our application should still run properly if used with any SQL-based database format (ex. Postgre), or when hosted locally. In essence, our backend setup works as follows: the front end connects to the back end through API calls (in the form of GET / POST requests), the back end processes these API calls through Spring Boot, and Spring Boot makes queries to pull / update information in the MySQL database.

In regards to our Spring Boot setup, we followed the typical three category implementation: entities, controllers, and repositories (also known as DAO or Persistence). This means that for each database table of our application, there is an entity, repository, and controller class in our Spring Boot implementation. For example, we have a `User` entity, `UserDAO` (repository), and `UserController`. The `User` class handles an instantiation of a user entity, and contains all the attributes one might expect of a user (ex. username, password, etc.). The `UserController` class handles the initial HTTP requests and specifies their paths. Such as specifying the “api/user/add” URL path being used to add a new user. The controller then processes these HTTP requests, calling functions in the `UserDAO` class to add / remove / modify users. More specifically, the `UserDAO` class will make direct queries to the database and function calls to the `User` class in order to complete the desired operations. This includes adding a new user, deleting a user, or getting an existing user’s information.

Our MySQL database utilizes seven tables for storing information: artifact, hotspot, location, panoview, section, user, and waypoint. A more detailed look at these tables is given in the database schema. Additionally, our application makes use of the hibernate framework, providing persistence for all queries and ensuring that failed transactions are properly rolled back.

In addition to our entity, controller, and repository classes, we implemented several classes for handling the permissions and security of our application (namely `ApplicationSecurityConfig`, `ApplicationUserPermission`, `ApplicationUserRole`, and `PasswordConfig`).

Fortunately for us, Spring Boot handles most security related concerns automatically, though we were still required to configure a lot of the features. Firstly, for our password encryption method chose to go with BCrypt, utilizing an iteration count of 10. BCrypt is designed to be easily scalable, meaning that as computing power increases, so too can the strength of the BCrypt encryption to match it. Of course the tradeoff here is that a higher iteration count would result in decreased efficiency. We chose to go with an iteration count of 10 (i.e. the strength of the encryption), which is

considered to be sufficient for a normal use application. Additionally, BCrypt automatically handles salting passwords (to prevent rainbow table attacks). Overall, we designed our application to be quite robust in terms of security.

Our focus on security continued for our implementation of permissions for the API calls. We set up three different roles that a user could be assigned to: User, Employee, and Owner. A User can read information pertaining to the museum (artifacts, sections, etc.), an Employee can read and modify museum data, and an Owner can read and modify both museum and user data. Each API call is assigned a permission required for execution, and therefore they are only able to be executed by certain roles. The authorized API users are stored in the “user” table of the application database, and a user must provide their username and password when making an API call in order to be successfully authorized. At the moment there are also several default users hard coded into the application, though the client may wish to later alter or remove these.

The specific software versions used for our backend are Spring Boot 2.7.9 and MySQL 8.0.32. Additionally, we made use of PHPMysqlAdmin for ease of use in managing and viewing the database, however this is not required for utilizing the application.

Testing

Throughout the development of our application we continually verified that our API calls and other methods were working correctly manually. Towards the end of development, we attempted to implement unit tests for our classes. This proved extremely difficult unfortunately, as the specific Spring Boot implementation we used contained a version of JUnit (5.7.0) that did not work properly for our application (see: [this discussion](#)). We were able to overcome this issue by manually excluding the default Spring Boot Junit dependencies and using newer versions, unfortunately however, the issues continued.

When attempting to set up an in-memory database through the use of the H2 dependency, further conflicts were discovered. Though we were eventually able to get the H2 in-memory database running properly, our use of EntityManagers and EntityManagerFactories throughout the application made using it extremely difficult. This is because they will always load their database connection through hibernate and the `persistence.xml` specified in the runtime application (not the testing one). Attempting to switch to the use of Persistence Contexts (`@PersistenceContext`) failed as well, leading to an authentication error causing all API calls to return a 401 error. Ultimately, if given more time, I believe the best way to solve this problem would have

to be updating to a newer version of Spring Boot and making use of the new query system and avoiding the use of EntityManagers all together.



Set-up Process

Certain files need to be changed in order to set-up the application.

```
private List<ApplicationUser> getApplicationUsers() {
    List<ApplicationUser> applicationUsers = Lists.newArrayList(
        new ApplicationUser(
            USER.getGrantedAuthorities(),
            passwordEncoder.encode( rawPassword: "user123@" ),
            username: "user123@",
            isAccountNonExpired: true,
            isAccountNonLocked: true,
            isCredentialsNonExpired: true,
            isEnabled: true),
        new ApplicationUser(
            EMPLOYEE.getGrantedAuthorities(),
            passwordEncoder.encode( rawPassword: "emp1812NOTL" ),
            username: "employee",
            isAccountNonExpired: true,
            isAccountNonLocked: true,
            isCredentialsNonExpired: true,
            isEnabled: true),
        new ApplicationUser(
            OWNER.getGrantedAuthorities(),
            passwordEncoder.encode( rawPassword: "ownerNOTL2023" ),
            username: "owner",
            isAccountNonExpired: true,
            isAccountNonLocked: true,
            isCredentialsNonExpired: true,
            isEnabled: true)
    );
}
```

User Profiles

The backend initializes a set of user profiles, make sure to change the passwords.

(/ApplicationUserDaoService.java)

The backend uses the USER account as a default. All api calls that make changes to the database are protected by permissions.

```
import axios from 'axios';

const username = 'user123@'
const password = 'user123@'

const token = Buffer.from(`${username}:${password}`, 'utf8').toString('base64')
const service = axios.create({
  baseURL: 'http://localhost:8080/api',
  headers: {
    'Authorization': `Basic ${token}`,
    'Accept': 'application/json',
    'Content-Type': 'application/json',
  },
});

export default service;
```

If you change the user profile information, make sure the frontend is updated as well. (/service.js). Make sure to also update the URL to the correct URL of the backend application.

Database

Run the sql script provided to set-up the SCHEMA and Architecture Tour data.

Connect Database

Fill in the server url, user and password.

```
<?xml version="1.0" encoding="utf-8" ?>
<persistence version="2.0"
  xmlns="http://java.sun.com/xml/ns/persistence"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
    http://java.sun.com/xml/ns/persistence/persistence_2_0.xsd">
  <persistence-unit name="PanoTour" transaction-type="RESOURCE_LOCAL">
    <provider>org.hibernate.jpa.HibernatePersistenceProvider</provider>
    <properties>
      <property name="hibernate.dialect" value="org.hibernate.dialect.MySQL5Dialect"/>
      <property name="javax.persistence.jdbc.driver" value="com.mysql.cj.jdbc.Driver"/>
      <property name="javax.persistence.jdbc.url" value="jdbc:mysql://0.0.0.0:0000/panotour"/>
      <property name="javax.persistence.jdbc.user" value="user"/>
      <property name="javax.persistence.jdbc.password" value="password"/>
    </properties>
  </persistence-unit>
</persistence>
```

Compile web-build

Through terminal, in the projects directory (PanoTourApp), run the following commands to compile a web-build.

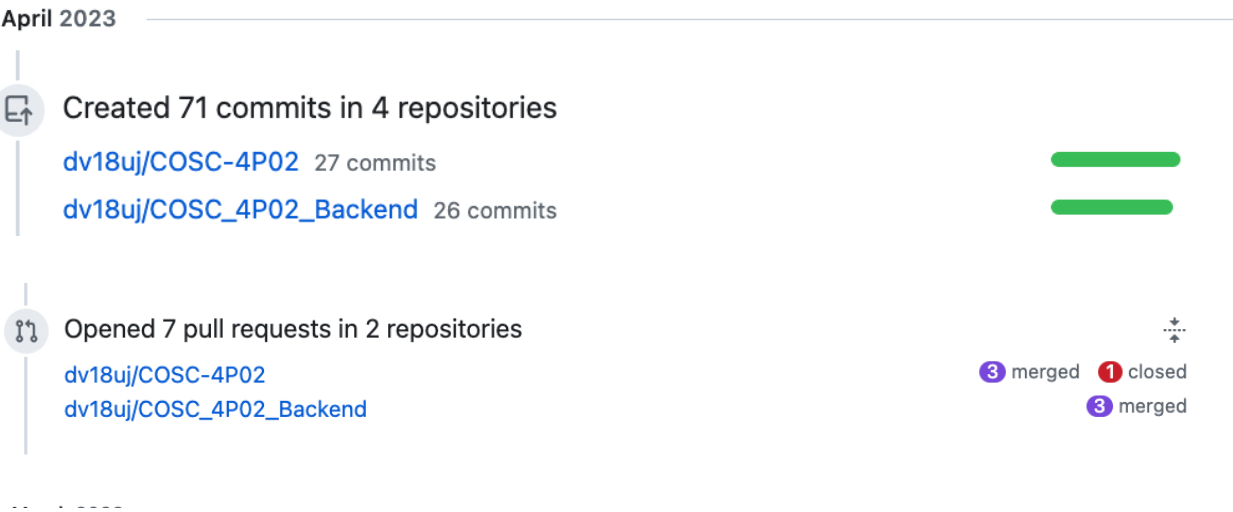
- npm install (this installs all dependencies)
- npx expo export:web
- npx serve web-build

Host this web-build and the backend application on the same server and the application is ready to access.



Contributions

Diego Vergara - 6583819



Michael Wisniewski - 6402176

main

Commits on Apr 30, 2023

Typo akaJengo committed 46 minutes ago	92f8e92	<>
Update README.md akaJengo committed 48 minutes ago	af852bd	<>
Cleanup akaJengo committed 1 hour ago	e89e4a1	<>
Merge pull request #14 from dv18uj/TestingUpdate akaJengo committed 1 hour ago	Verified 6583c65	<>

Commits on Apr 25, 2023

Test Update akaJengo committed 5 days ago	0aa6d80	<>
Testing akaJengo committed 5 days ago	adf7d9d	<>



Commits on Apr 18, 2023

testing akaJengo committed 2 weeks ago	70265a6	<>
---	---------	----





Commits on Apr 4, 2023

Tour akaJengo committed last month	9c7fc13	<>
Testing akaJengo committed last month	656b306	<>
Config update akaJengo committed last month	f2a18be	<>
File creation akaJengo committed last month	93c273c	<>
Update README.md akaJengo committed last month	91b499e	<>







Commits on Feb 13, 2023

- 2x3 Paragraph abstraction
akaJengo committed on Feb 13  5aa7bef 










Commits on Feb 9, 2023

- Styles
akaJengo committed on Feb 9  eebb422 
- Basic testing Setup
akaJengo committed on Feb 9  3bc11b3 

Commits on Feb 2, 2023

- Update README.md
akaJengo committed on Feb 2  5565df3 
- Update .gitignore
akaJengo committed on Feb 2  6cc1789 
- Update Readme
akaJengo committed on Feb 2  2e4e94f 




Commits on Jan 31, 2023

- Update commands
akaJengo committed on Jan 31  95132b6 
- Merge pull request #2 from dv18uj/deployment   2c47cbe 
- Create README.txt
akaJengo committed on Jan 31  8e468ae 
- Update to Expo
akaJengo committed on Jan 31  c919a45 

[Newer](#) [Older](#)


StyleSheets

Commits on Mar 15, 2023

- Updoot 
akaJengo committed on Mar 15  88bc519 



Login

Commits on Apr 25, 2023

- Update Login.js 
akaJengo committed 5 days ago  4a9f44d 
- Login page test
akaJengo committed 5 days ago  b0e3556 

ChangeStateMenu

Commits on Mar 7, 2023

- ChangeState
akaJengo committed on Mar 7  6a29eb4 

Git log:

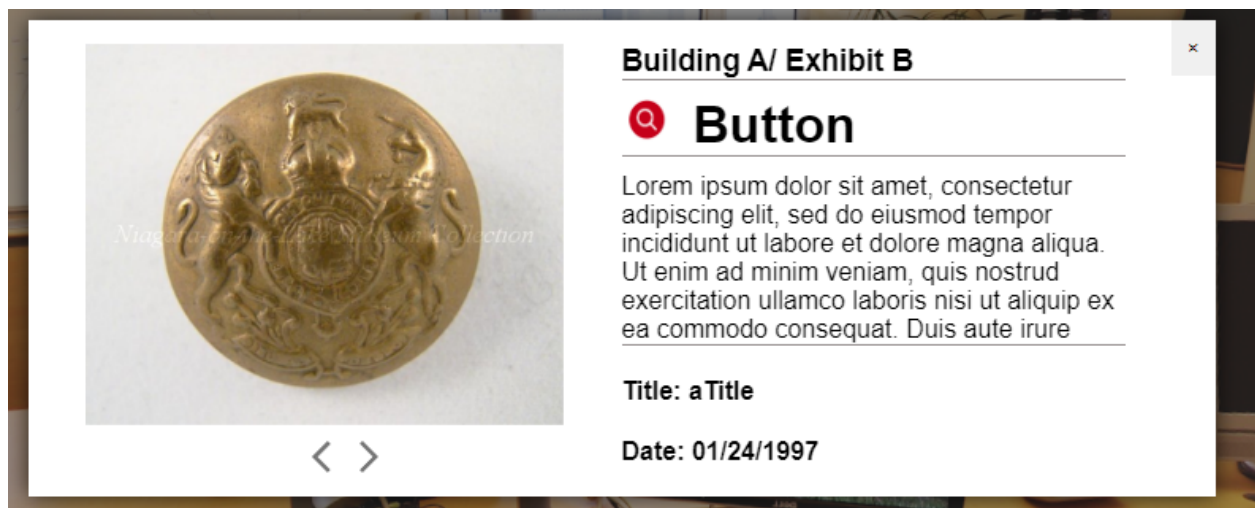
92f0692 - Mikes, 2 hours ago : Typo
af852bd - Mikes, 2 hours ago : Update README.md
ea9e4a1 - Mikes, 2 hours ago : Cleanup
6583c65 - Michael, 2 hours ago : Merge pull request #14 from dv18uj/TestingUpdate
fedc4fb - Mikes, 3 days ago : On Login: !!GitHub_Desktop<Login>
9a8d734 - Mikes, 3 days ago : index on Login: 4a9f44d Update Login.js
4a9f44d - Mikes, 5 days ago : Update Login.js
b0e3556 - Mikes, 5 days ago : Login page test
0aa6d80 - Mikes, 5 days ago : Test Update
adf7d9d - Mikes, 5 days ago : Testing
70265a6 - Mikes, 13 days ago : testing
9c7fc13 - Mikes, 4 weeks ago : Tour
656b306 - Mikes, 4 weeks ago : Testing
f2a18be - Mikes, 4 weeks ago : Config update
93c273c - Mikes, 4 weeks ago : File creation
91b496e - Mikes, 4 weeks ago : Update README.md
2aeb13c - Mikes, 5 weeks ago : Testing
88bc519 - Mikes, 7 weeks ago : Updoot
6a29eb4 - Mikes, 8 weeks ago : ChangeState
5aa7bef - Michael, 3 months ago : 2x3 Paragraph abstraction
eebb422 - Mikes, 3 months ago : Styles
3bc11b3 - Mikes, 3 months ago : Basic testing Setup
5565df3 - Mikes, 3 months ago : Update README.md
6cc1789 - Mikes, 3 months ago : Update .gitignore
2e4e94f - Mikes, 3 months ago : Update Readme
95132b6 - Mikes, 3 months ago : Update commands
2c47cbe - Michael, 3 months ago : Merge pull request #2 from dv18uj/deployment
8e468ae - Mikes, 3 months ago : Create README.txt
c919a45 - Mikes, 3 months ago : Update to Expo

Zachary McGovarín - 6817019

ZachM, 24 seconds ago : fix final panoview permission
ZachM, 14 minutes ago : Merge branch 'main' of https://github.com/dv18uj/COSC_4P02_Backend
ZachM, 14 minutes ago : set final permissions
ZachM, 57 minutes ago : Merge branch 'main' of https://github.com/dv18uj/COSC_4P02_Backend
ZachM, 57 minutes ago : add database setup sql file
ZachM, 3 days ago : fixed api calls, this version runs, to be continued
ZachM, 4 days ago : bug fixing attempt still
ZachM, 4 days ago : continued
ZachM, 4 days ago : attempting to fix broken api calls
ZachM, 4 days ago : revert a couple unnecessary changes
ZachM, 4 days ago : test finally runs, errors seemingly resolved, user class to be further tested to ensure something else didn't break
ZachM, 4 days ago : improvement, the project will now build, test still not working yet
ZachM, 4 days ago : attempting to make unit tests work with mock db... some improvement
ZachM, 5 days ago : comment out tests for now
ZachM, 7 days ago : Merge branch 'main' of https://github.com/dv18uj/COSC_4P02_Backend
ZachM, 7 days ago : create first automated test, adduser
ZachM, 4 weeks ago : fix location post
ZachM, 4 weeks ago : fix section add
ZachM, 4 weeks ago : fix loading permissions by role for user
ZachM, 4 weeks ago : fix post requests
ZacharyM, 4 weeks ago : Merge pull request #2 from dv18uj/api

ZachM, 4 weeks ago : use permission based auth instead of roles
ZachM, 4 weeks ago : load api auth users from database
ZachM, 4 weeks ago : Merge branch 'api' of https://github.com/dv18uj/COSC_4P02_Backend into api
ZachM, 4 weeks ago : add employee and owner roles, restrict user management to owner
ZachM, 5 weeks ago : remove project duplicate
ZachM, 5 weeks ago : initial api authentication implementation
ZachM, 6 weeks ago : add initial auth classes
ZachM, 7 weeks ago : add getuser, fix typos
ZachM, 7 weeks ago : Merge branch 'api' of https://github.com/dv18uj/COSC_4P02_Backend into api
ZachM, 7 weeks ago : create user / userDAO / usercontroller
ZachM, 7 weeks ago : remove useless files
ZachM, 7 weeks ago : get artifact name by oid
ZachM, 8 weeks ago : add api call classes
ZachM, 8 weeks ago : create sql connection and perform a test insertion

Haseeb Ahmad - 654195



(created info panels, my github was not working for first two months, so a lot of my commits were pushed through the scrum master (Diego)).

Wayne Bessey - 6527303

```
b53411e - WayneB17, 4 days ago : needs API
539d24b - WayneB17, 4 days ago : On unfinished_sidemenu: !!GitHub_Desktop<unfinished_sidemenu>
a1361eb - WayneB17, 4 days ago : index on unfinished_sidemenu: 58caa57 still not working
58caa57 - WayneB17, 5 days ago : still not working
7e3ae46 - WayneB17, 6 days ago : display not working (sideMenu in molecules)
35c8630 - WayneB17, 6 days ago : ignore these
6ecd64 - WayneB17, 6 days ago : not displaying properly
5451f55 - WayneB17, 3 weeks ago : unfinished, needs rotation updates to face camera + maybe orbitControls camera lock
4f8e818 - WayneB17, 4 weeks ago : added hover
10b7cc5 - WayneB17, 5 weeks ago : Update Tour.js
4e50ec6 - WayneB17, 5 weeks ago : added drei
57d0397 - WayneB17, 5 weeks ago : fixed
9b91beb - WayneB17, 5 weeks ago : idle icon finished, needs hover/onclick
a279628 - WayneB17, 5 weeks ago : removed stylesheet from sidemenu in molecules & menu in atoms
477fb4a - WayneB17, 5 weeks ago : needs work
d3d8b09 - WayneB17, 6 weeks ago : multiple hotspots
bfc4e69 - WayneB17, 6 weeks ago : a
30ab9c1 - WayneB17, 6 weeks ago : displays hotspot but not functional.
c97471a - WayneB17, 7 weeks ago : functional
f624da9 - WayneB17, 7 weeks ago : Unfinished but functional
60ce22d - WayneB17, 7 weeks ago : Delete NavMenu.jsx
ad1ba7e - WayneB17, 7 weeks ago : Update NavMenu.jsx
65b4c94 - WayneB17, 7 weeks ago : Update Menu.jsx
945bd62 - WayneB17, 7 weeks ago : Update Menu.jsx
10a6121 - WayneB17, 8 weeks ago : Revert "npm install expo-gl"
038baca - WayneB17, 8 weeks ago : Merge pull request #6 from dv18uj/SideMenu
c20bf67 - WayneB17, 9 weeks ago : Added Side Menu
```

main *only showing commits that ended up on main branch/final product

Commits on Apr 26, 2023

needs API
WayneB17 committed 4 days ago
Created sidemenu that could accept list items to display -> Diego later added API calls to dynamically render sidemenu navigation

Commits on Mar 28, 2023

idle icon finished, needs hover/onclick
WayneB17 committed on Mar 28
Icons now implemented properly

Commits on Mar 27, 2023

needs work
WayneB17 committed on Mar 27
Tried to implement actual display icons, and make the infopanel respond to onclick functions of the hotspot, but displayed incorrectly (later fixed by diego)

Commits on Mar 21, 2023

multiple hotspots ...
WayneB17 committed on Mar 21
a
WayneB17 committed on Mar 21
These two commits => hotspot is now able to display in multiple instances

Commits on Mar 20, 2023

4/30/23, 8:36 PM Commits - dv18uj/COSC-4P02

displays hotspot but not functional.
WayneB17 committed on Mar 20
Problem solving displaying hotspots onto three.js

Commits on Mar 13, 2023

functional
WayneB17 committed on Mar 13
(above and below commits) Basic functionality for info panel, displays a button -> onclick open panel -> opens a basic styled blank panel

Commits on Mar 9, 2023

Unfinished but functional
WayneB17 committed on Mar 9

Commits on Mar 7, 2023

Merge pull request #6 from dv18uj/SideMenu ...
WayneB17 committed on Mar 7

Commits on Feb 27, 2023

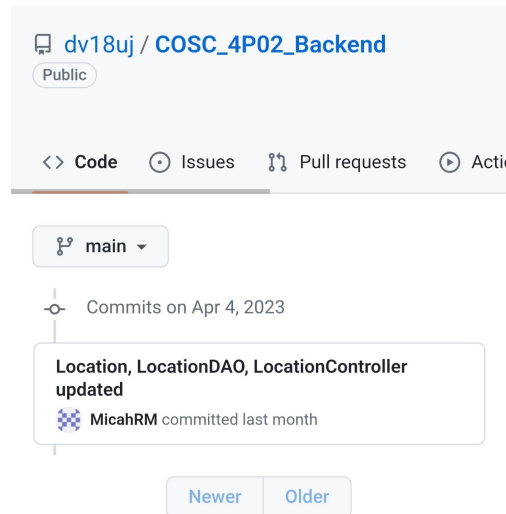
Added Side Menu
WayneB17 committed on Feb 27
created sidemenu (body), sidemenu button, displayed on mesh. menu was empty of content but was functional

Alexander Gibson- 6479364

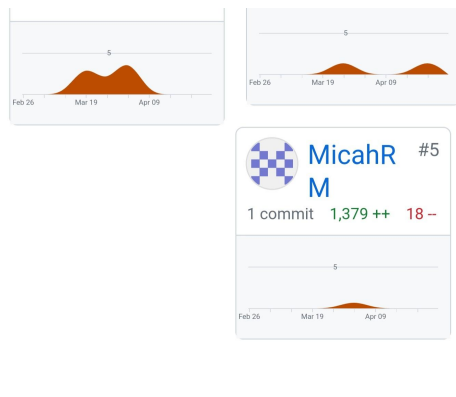
main		
Commits on Apr 4, 2023		
updated	605f31e	<>
AlexGibby2000 committed last month		
Update Waypoint, WaypointDAO, and WaypointController	9f09df9	<>
AlexGibby2000 committed last month		
updated Hotspot, HotspotController, and HotspotDAO	9571359	<>
AlexGibby2000 committed last month		
updated	11d989b	<>
AlexGibby2000 committed last month		
Deleted Backend folder	2dfd21b	<>
AlexGibby2000 committed last month		
Commits on Mar 28, 2023		
Updated +added HotspotController and WaypointController	4497c58	<>
AlexGibby2000 committed on Mar 28		
Commits on Mar 23, 2023		
updated	053616c	<>
AlexGibby2000 committed on Mar 23		
Commits on Mar 21, 2023		
moved to PanoTour	6d2a14b	<>
AlexGibby2000 committed on Mar 21		
Create Hotspot / HotspotDAO / Waypoint / WaypointDAO	68b588f	<>
AlexGibby2000 committed on Mar 21		
Create Hotspot / HotspotDAO / Waypoint / WaypointDAO	0b28c55	<>
AlexGibby2000 committed on Mar 21		

main		
Commits on Apr 4, 2023		
updated	605f31e	<>
AlexGibby2000 committed last month		
Update Waypoint, WaypointDAO, and WaypointController	9f09df9	<>
AlexGibby2000 committed last month		
updated Hotspot, HotspotController, and HotspotDAO	9571359	<>
AlexGibby2000 committed last month		
updated	11d989b	<>
AlexGibby2000 committed last month		
Deleted Backend folder	2dfd21b	<>
AlexGibby2000 committed last month		
Commits on Mar 28, 2023		
Updated +added HotspotController and WaypointController	4497c58	<>
AlexGibby2000 committed on Mar 28		
Commits on Mar 23, 2023		
updated	053616c	<>
AlexGibby2000 committed on Mar 23		
Commits on Mar 21, 2023		
moved to PanoTour	6d2a14b	<>
AlexGibby2000 committed on Mar 21		
Create Hotspot / HotspotDAO / Waypoint / WaypointDAO	68b588f	<>
AlexGibby2000 committed on Mar 21		
Create Hotspot / HotspotDAO / Waypoint / WaypointDAO	0b28c55	<>
AlexGibby2000 committed on Mar 21		

Micah Rose-Mighty - 6498935



Implemented



[Terms](#) [Privacy](#) [Security](#) [Status](#) [Docs](#)
[Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)

 © 2023 GitHub, Inc.

Implemented the Back-end functionality of the Location feature within the program by developing the various location-related classes in the system. During the early stages of the project, I was unfamiliar with the use of Github for collaborative programming and at first it was a learning curve to get my work committed to the project. Eventually,

our Scrum Master Diego was able to get me up to speed which allowed me to make my contributions. This explains the one fairly large commit that can be seen from my end.

Samuel Onabolu - 6783369

The screenshot displays a GitHub commit history for the user Samuel Onabolu (6783369). The interface is dark-themed and shows a list of commits grouped by date. Each commit entry includes a title, the user's profile picture and name, the commit date, a copy icon, the commit hash, and a code diff icon.

Commits on Apr 25, 2023

- New APIs** (topmansam committed 5 days ago) - Hash: 31cf3e2

Commits on Apr 24, 2023

- file creation** (topmansam committed last week) - Hash: 5d90aaa

Commits on Mar 30, 2023

- sections created** (topmansam committed on Mar 30) - Hash: d4cc790
- Merge branch 'api' of https://github.com/dv18uj/COSC_4P02_Backend int...** (topmansam committed on Mar 30) - Hash: ac169f0

Commits on Mar 28, 2023

- created sections table** (topmansam committed on Mar 28) - Hash: 3fe1e04

Navigation buttons: [Newer](#) [Older](#)

Commits on Apr 4, 2023

- created quitArtifact.test.js file** (topmansam committed last month) - Hash: ebd356f

Navigation buttons: [Newer](#) [Older](#)

World in the backend implementation the API calls. Created the sections and pano view SQL database tables and java files.

Github Repository

<https://github.com/dv18uj/COSC-4P02>

https://github.com/dv18uj/COSC_4P02_Backend