



Cyclops Team 0 Final Report

By

Eric Jiang

Larry Gu

Megan Nguyen

Sumeet Bajwa

Table of Contents

Table of Contents	2
Project Description	4
Introduction	4
User Groups	4
App User	4
Admin	4
DFD Diagrams	5
DFD Level 0	5
DFD Level 1	6
Functional Requirements	7
App User	7
Admin	7
Software Build	7
Software Implementation	8
Test Report	8
Unimplemented Requirements	9
Partially Implemented Requirements	9
Bugs	10
Overview:	10
Split Pane Routing Bug Fix #218	10
Async Data Access Bug Fix #263	10
Bug fix: read progress bar when not logged in #285	10
Search Bar Bug Fix #288	10
Iphone Mode Layout Bug Fix #289	10
Bug Fix - Guard #292	10
Bug fix - Article read progress bar display #294	10
Edit Tool Method Test Page Bug Fix #322	11
Deploy the Newest Version and Fix Bugs #360	11
Main page Bug Fix #373	11
Eco Tracker ion-range Color Update Bug Fix #378	11
Article Edit Feature Bug Fix #391	11
Article bug fix #392	11
Step-by-Step Guide	12
Where to find Code	12
Installation details	12

Software Deployment Instructions	14
Update Web version	14
Overview:	14
Step 1: Setup the environment.	14
Step 2: Generate PWA	14
Step 3: Deploy on the firebase	14
Update Android version	14
Overview:	14
Step 1: Installing Android Studio	14
Step 2: Run “ionic build --prod” in the cmd.	14
Step 3: Generate android codes	15
Step 4: Package .APK file	15
Note:	16
Update IOS version	16
Note:	16
Maintenance	17
Fees	17
Set Administor	18
Incomplete Testing	19
Unimplemented Features	19
Links	20
GitHub Repository Link	20

Project Description

Introduction

Cyclops is an information-based website that discusses and informs users on the many different issues surrounding the environmental crisis, and the impact that crisis is having on us. Using this website, we have created the CyclopediApp.

The CyclopediApp is an educational application. This application is used to inform users of the environmental issues going on through articles, as well as help users that want to track their own environmental impact on the world through the implemented Eco Tracker features.

User Groups

App User

"App User" refers to all those who are concerned about the environment and want to actually measure how well they are at protecting the environment. Concretely, App Users can read articles, and use a special calculator called eco-tracker in CyclopediApp.

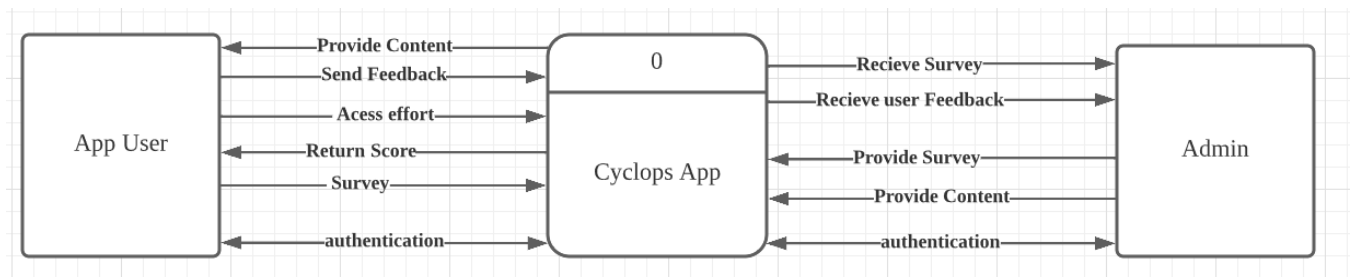
Admin

The Admins manages all the articles inside the App. They also answer the feedback from all the App Users. Admins have special logins and would be able to do everything within the App, this includes being able to edit, add/remove articles. No programming skills are required to be an Admin.

DFD Diagrams

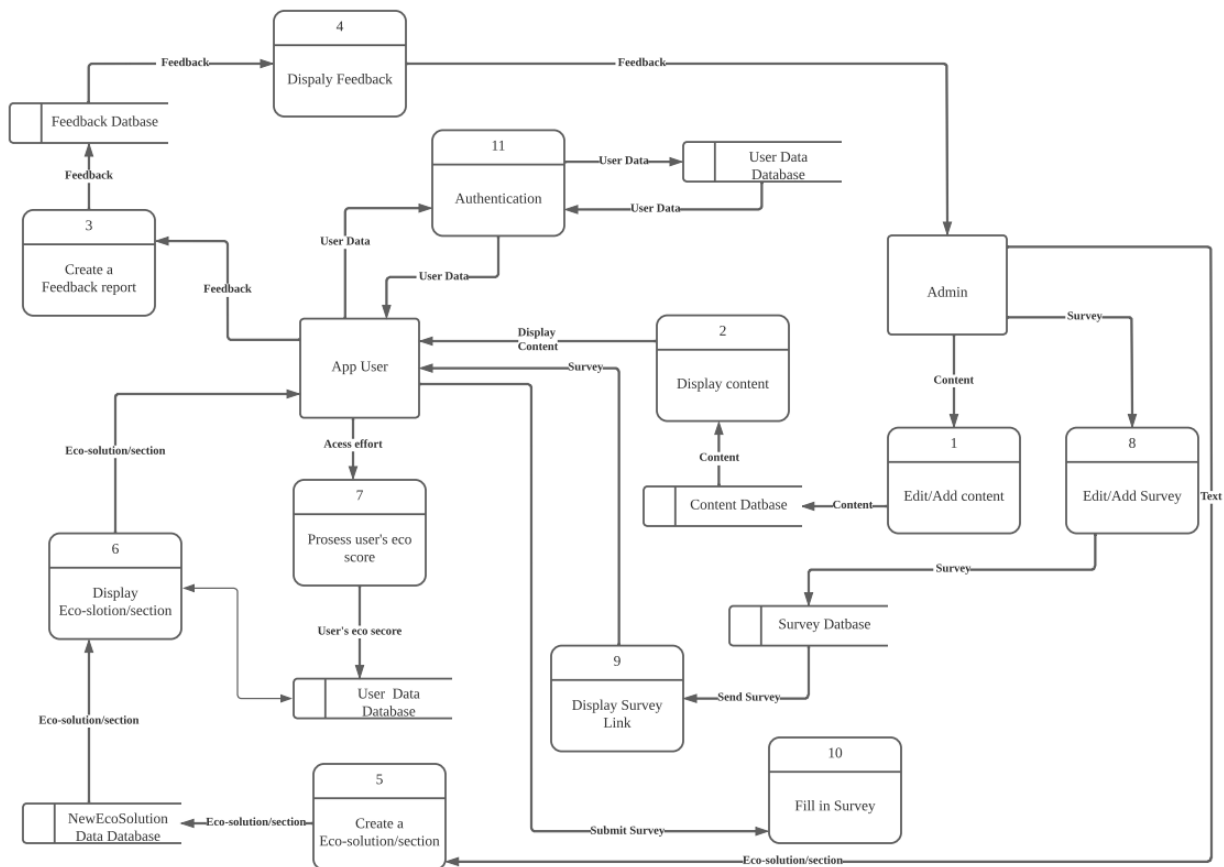
DFD Level 0

- There are two user group in the App:
 1. The user has rights to edit/add contents and edit/add surveys. We defined them as “Admin”.
 2. The user does not have rights to edit/add content and edit/add surveys. They are reviewers, they may give some feedback to the Author of content, complete surveys, and create scales. We defined them as “App Users”.
- A user can be both “Admin” and “App User”. Which means the “Admin” also can give some feedback to the Author of content.
- The content is provided by Admin and it will be displayed in the App. App users are able to review the content. Besides that, they are able to submit their feedback to the admin, and the admin will receive their feedback. Also, the App user is able to input their effort of the solution and receive scores. Moreover, Admin will provide some surveys as the link to users, the result of the survey will be display in the third party website.



DFD Level 1

- The content admin edit/add content by process 1, then that content will be stored into the database. The process 2 will get content from the database and display it to the user.
- When users review those contents, they are able to give admin some feedback by process 3, that feedback also will be stored into the database, and then process 4 will get that feedback and display it to the content admin.
- Admin can provide Eco-solution with the section by process 5, thoes data will be stored into the database and displayed by process 6. Then users can access their effort on the specific Eco-solution by process 7.
- Admin can provide/edit Survey link and description by process 8, thoes data will be stored into the database and displayed by process 9. Users can click on the external link to complete the survey by process 10.
- Process 11 authenticates the user account, and get user's account data from database.



Functional Requirements

App User

- Ability to Sign up using email and password
- Access Profile page once signed up and logged in
- Ability to Log in and Log out of application
- Click on available surveys on home page
- Read articles
- Search for articles using search bar
- Track how many articles a user has read
- Send Feedback using Feedback Form
- Input solutions and scores using ECO Tracker
- Update previously inputted solutions on ECO Tracker tab
- Ability to filter previously entered solutions on ECO Tracker tab

Admin

- Sign in using admin login information
- Edit surveys
- Add new surveys
- View any Feedback that has been submitted
- Add new Article Cards
- Edit contents of any Articles on the application
- Change cover photos on article cards

Software Build

For the CyclopediApp, it was decided that using Ionic would be the best building tool for the creation. The reasoning behind this decision was that Ionic provided us with more freedom because of its adaptive styling. With adaptive styling, we could use the same code base for multiple platforms as well as it allowed us to create a similar user experience across all platforms the application would be on. Ionic also contained theming and navigation that could be used in the mobile application.

Software Implementation

Test Report

During the sessions that were run during peer testing #2, there were a couple of minor issues discovered. Since then these minor issues have been addressed. The minor issues that were discovered included those that had to do with visibility and styling of the buttons and button placement in general. User's seemed to have difficulties figuring out which buttons meant what in the Articles tab when working with and editing the articles, this was addressed by changing the names on the buttons to ones the users are able to understand. There also was not a back button present when a user entered the profile page, since the peer testing sessions a back button has been implemented. Lastly on the application, user's struggled to find the sign up button since it was displayed in the login page. A separate sign up button has since been implemented.

Overall for the application testing has been administered on all the features. This includes testing on the user profile, this includes a user being able to create a new profile, email validation on a newly created profile, sign in and log out of an existing one, and being able to reset the password on your account.

All features on the articles tab have been tested as well. This includes features such as being able to scroll, search and select articles. If you are logged in with an account you can send feedback through a form, and track how many articles you have read overall. Lastly if an administrative account is logged in they have the ability to edit articles, add new articles, and delete articles.

The section of the application that has not been tested yet is the Eco Tracker tab. It initially was 2 forms to take a user input, but since peer testing #2 the layout of the whole tab has changed. More can be read on the non-testing done for the Eco Tracker tab below.



Unimplemented Requirements

The popover feature was not successfully implemented. We explored many ways on how this feature could be implemented, and after a few weeks we decided to focus on higher priority features and proposed a commercial plugin to the client that would work as an alternative solution which is the [CKPro Popover](#).

Partially Implemented Requirements

We tried to allow admins to upload images and use those images for article displays. Currently, admins cannot upload any new image to the database, but they can choose from a given set of images.

Bugs

Overview:

Bugs found in software testing have **all been resolved**. Here's a list of the records containing major ones. Details can be found on the Github Repository Issue section:

<https://github.com/IRIDIXVdt/2021W-UBCO-COSC-499-Cyclops/issues>

Split Pane Routing Bug Fix #218

Resolved a bug that under desktop view, the split pane (menu) redirects users to correct pages with incorrect URLs.

Async Data Access Bug Fix #263

Resolved a bug that occurred when Auth.service utilizes subscribe(), which is called multiple times unexpectedly.

Bug fix: read progress bar when not logged in #285

Resolved a bug with progress bar when users are not logged in.

Search Bar Bug Fix #288

Resolved a bug that caused search bar results not redirecting to corresponding pages under the desktop view.

Iphone Mode Layout Bug Fix #289

Resolved a bug that when using iPhone Mode at the edit page, the buttons on the top right corner are not displayed properly.

Bug Fix - Guard #292

Resolved a bug that caused the guard can not prevent non-admin users go to the admin page

Bug fix - Article read progress bar display #294

Resolved a bug so that we can prevent text cut off and implement responsive styling

Edit Tool Method Test Page Bug Fix #322

Resolved compilation bugs in the Ionic Page Component 'EditingToolTestPage' which caused software deployment failure.

Deploy the Newest Version and Fix Bugs #360

Fix various bugs of eco-solution page when generate the PWA website version

Main page Bug Fix #373

Resolved compilation bugs in the Main page which caused getting user eco score error If user not attend any solution yet

Eco Tracker ion-range Color Update Bug Fix #378

Resolved a bug that caused the ionic range component in the Eco Tracker section to update color incorrectly.

Article Edit Feature Bug Fix #391

Resolved a bug so that if admin clicks on a modal's backdrop, then that modal's corresponding action should be canceled instead of executed.

Article bug fix #392

Resolved a bug so that modal components corresponding to article add/remove features match the ones for the edit feature.

Step-by-Step Guide

Where to find Code

The code for the CyclopediApp we have created can be found with the github link that is provided below. Once on the github repository page, you will find the 4 main folders and the README.md file. The README.md is a collection of resources put in place to provide some insight into the development of the CyclopediApp. 2 of the 4 folders on the github repository show an accumulation of meeting notes and client notes while the project was being worked on. The code for the project can be found in the Cyclops folder. Once in the Cyclops folder, there are folders for firebase which is the database, the android folder which is the build for android version of the application, and the src folder. Once inside the src folder, there is another folder labeled app. The app folder is the main folder that contains the full build for the CyclopediApp we have created and worked on throughout the year. In this folder, there are separate folders for each individual feature of the application we have worked on, the main folders include:

- Authentication folder for anything regarding login, sign up, sign out.

- firebaseService folder is in regards to the database.

- Page-space-er is the folder for the Home tab.

- Page-space-la is the folder for the Articles tab.

- Page-space-me is for Articles, and the Articles Feedback Component.

- Page-space-su is for the Eco Tracker tab.

- sharedData is the folder that contains all the data used in the Application for articles, etc.

- Tabs is the folder for the tab bar on the mobile application.

- UserProfile folder contains everything in regards to the user's profile once it has been created.

Installation details

This guide is derived from ionic frameworks official documentations. You can find them here:

<https://ionicframework.com/docs>

Start by installing Node.js, the runtime environment for javascript.

<https://nodejs.org/en/>

With Node.js installed, you should be able to run the npm commands in your command prompt. If you type in

```
npm -v
```

you should see something like this:

```
F:\>npm -v  
8.1.0
```

This project utilizes the ionic+angular framework, so now let's install them as well. In your command prompt, type in

```
npm install -g @ionic/cli  
npm install @ionic/angular@latest --save
```

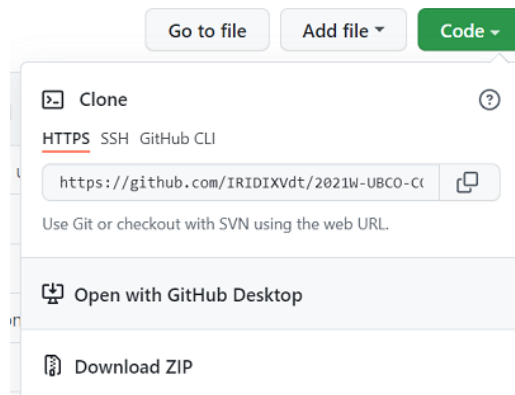
Now that we have both tools ready, we can now import the project from github.

<https://github.com/IRIDIXVdt/2021W-UBCO-COSC-499-Cyclops>

You may use any tool you like to clone this project, but as this is a github repository, Github Desktop is recommended.

<https://desktop.github.com/>

If you have the software ready, you may go to the repository and click on "Open with GitHub Desktop" to clone this project.



You should be able to find a folder called "2021W-UBCO-COSC-499-Cyclops" when you have cloned this project. This contains everything in the COSC 499 Capstone Course, the ionic+angular project you will be using is under the "Cyclops folder". Go to that directory with your command prompt, run

```
npm install
```

When the installation is finished, this project is ready for development.

If you want to test the software locally, then, under the software project's directory, run

```
ionic serve
```

You should see a website launched in your browser.

Software Deployment Instructions

Update Web version

Overview:

The project is hosted on the firebase. To host on the firebase, the ionic code has to be generated into PWA. Then deploy it on the firebase.

Step 1: Setup the environment.

Install some tools that can package the ionic code into PWA. We have already set up the environment, so no need to redo this process. The guidance documentation available at: <https://ionicframework.com/docs/angular/pwa>

Step 2: Generate PWA

Run in the cmd.

```
ionic build --prod
```

The program will automatically build a PWA website version under the “www” folder.

Step 3: Deploy on the firebase

Install firebase CLI first by run

```
npm install -g firebase-tools
```

in the cmd on your local machine. Then run “firebase deploy” in the cmd. The program will automatically deploy on the firebase. Then go to the firebase console, at the “Hosting” section, you can see the deployment history and the url link of the website.

Update Android version

Overview:

The process is similar to the deployment of the web version. You need to run some commands to generate the ionic codes into android codes, and then package the android into the .APK file.

Step 1: Installing Android Studio

Download Android Studio from the [Android website](#). More detailed installation instructions can be found in the [User Guide](#).

Step 2: Run “ionic build --prod” in the cmd.

The program will automatically build a PWA website version under the “www” folder.

Step 3: Generate android codes

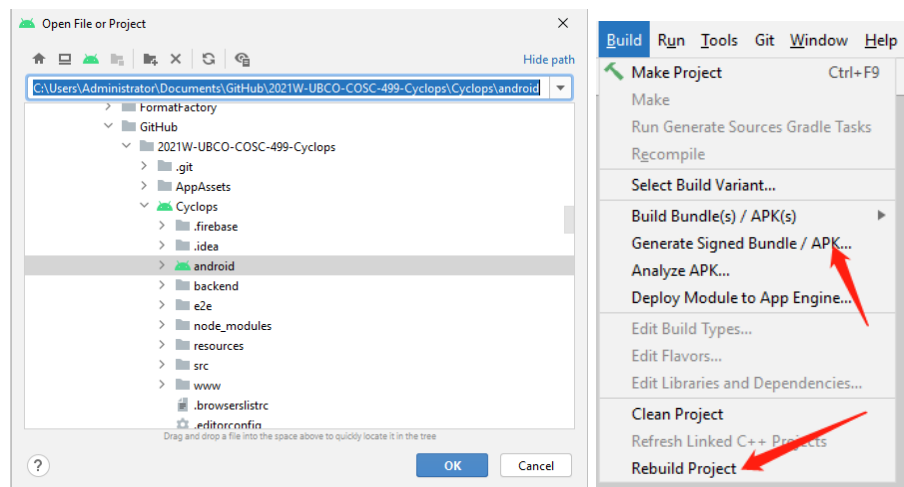
Run in the cmd.

```
ionic capacitor sync --prod
```

This command will generate android codes under the “android” folder. To the package ID, open the capacitor.config.json file(Under the Cyclops folder) and modify the appld property

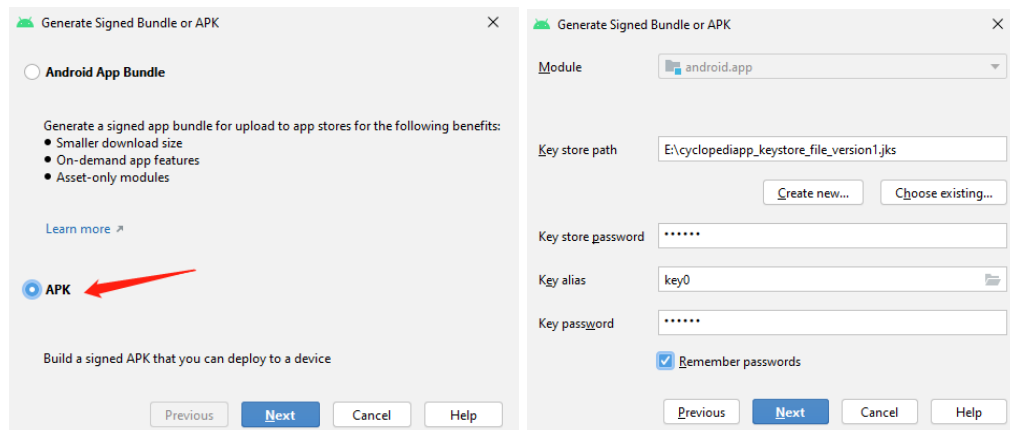
Step 4: Package .APK file

Open the “android” folder in android studio.

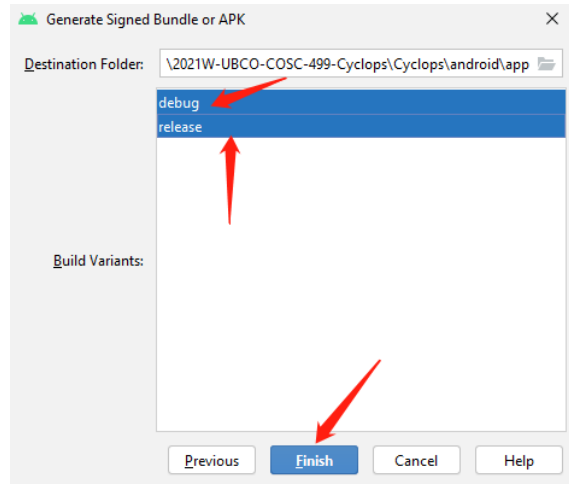


Then click on “Generate Signed Bundle”

If it gery, Click on “Rebuild Project”, and then “Generate Signed Bundle”



The key store file is under the root of the project. The password is 123456, the key alias is “key0”. You can create a new one if you want.



Set your destination folder and select debug and release, then click on Finish.

Note:

As this project contains a website version deployed in PWA, users can install the software directly from the website:

- Open the URL address of the website in the browser on Android.
- Open browser's settings.
- Scroll down and tap Add to Home screen.
- Confirm by tapping Add.

Update IOS version

We do not have an iOS version launched due to a technical limitation.

<https://ionicframework.com/docs/developing/ios>

Document for developments on iOS version can be found here.

Note:

Similar to the android version, users can add the software directly to the home screen:

- Open the URL address of the website in the browser on **safari**.
- Then tap the 'Share' button.
- Scroll down and tap 'Add to Home Screen'
- Enter the name for the app then tap add.
- The PWA will show up on your home screen like a native iOS app.
- *Experience of using this app can be limited on the iOS platform.

Maintenance

The link of backend console: <https://firebase.google.com>

Account: cyclopediappcyclops0@gmail.com

Password: qcg@nva8dtg0DWP_hbq

Fees

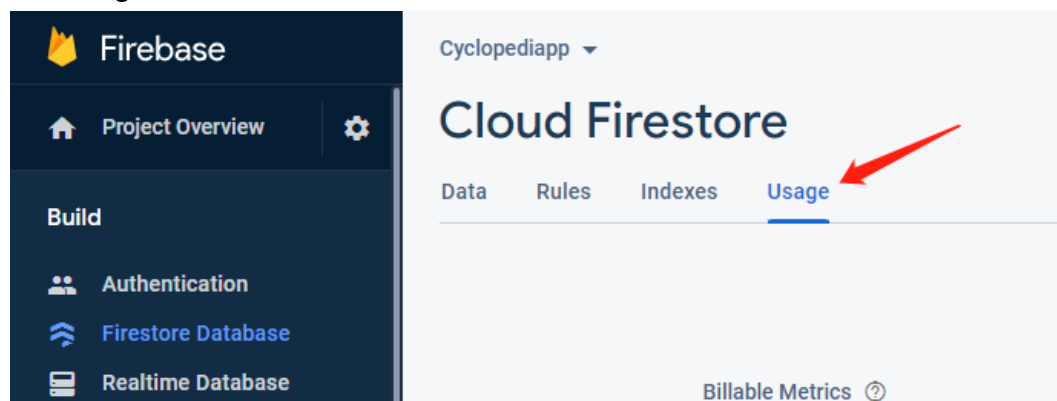
All information available at: <https://firebase.google.com/pricing>

The fees contains two parts

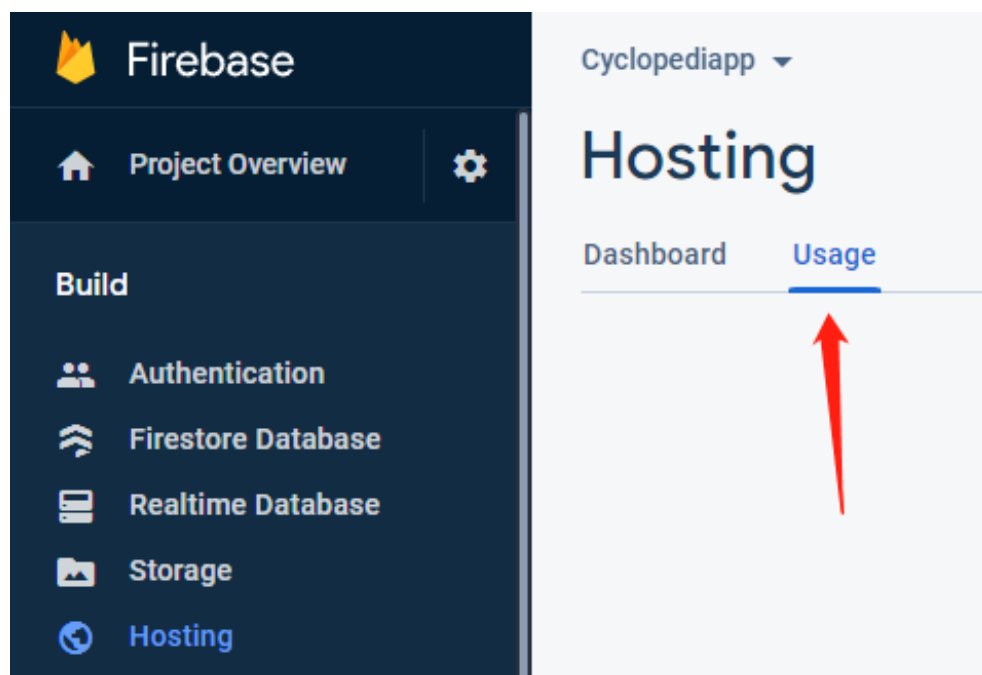
Database

Hosting

To review the usage of database:

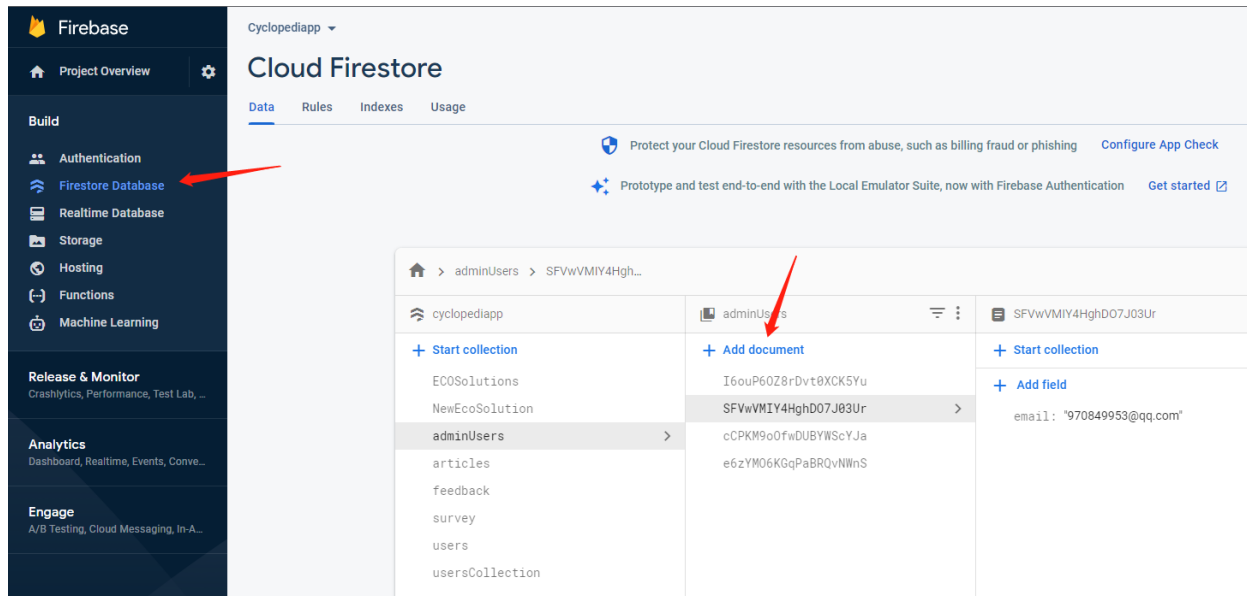


To review the usage of Hosting:

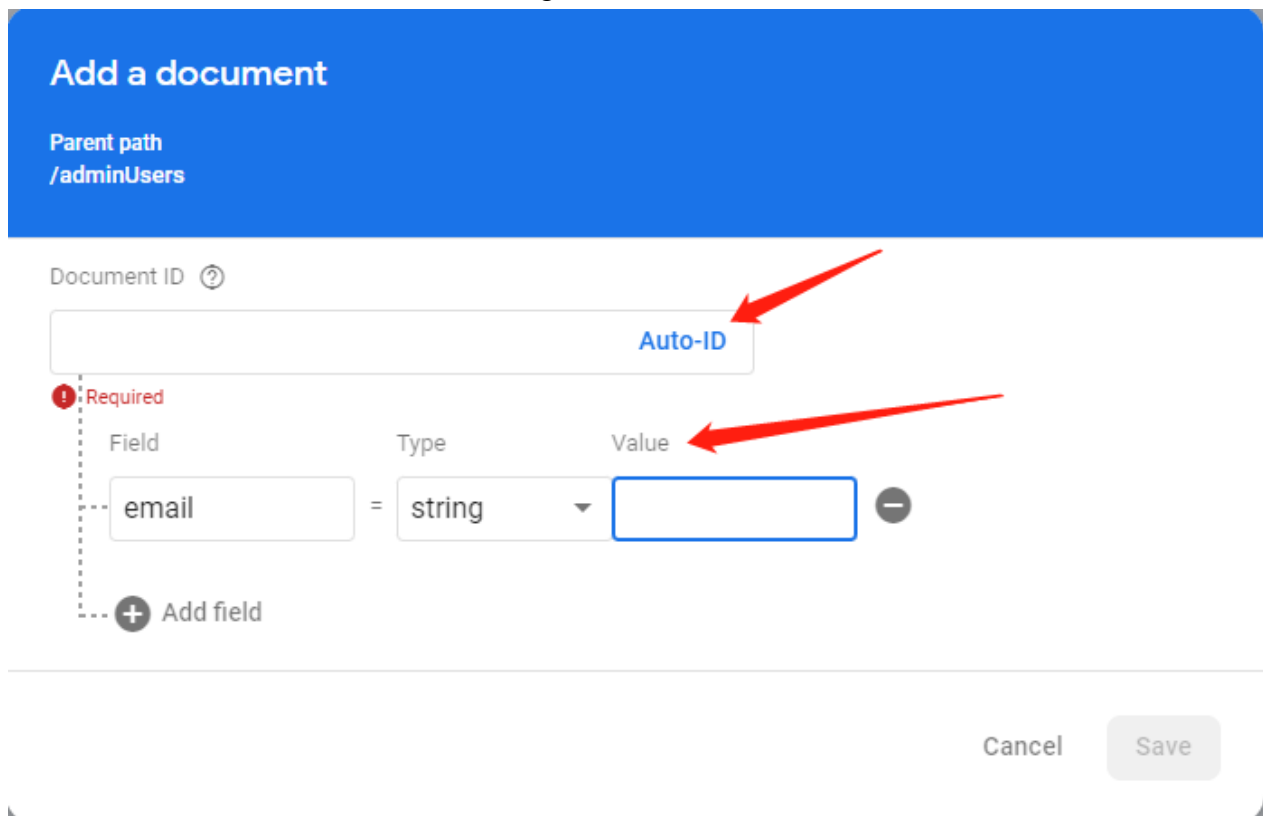


Set Administor

At the firebase console, click on the “Add document”



Click on the “Auto-Id” and enter the registered email of that account in the “Value”



And then click on save.

Incomplete Testing

Originally when peer testing #2 was conducted, on the Eco Tracker tab the user had the ability to input and complete a solution by opening up a pop out form. On this form the user could filter and submit a solution either by article section or by its star weighting. But since then the Eco Tracker Tab has been changed and there are features in place that have not been tested yet. The first is that instead of opening a form to input a solution, the user can now complete and update previous solutions straight from the Eco Tracker tab itself after they have logged into their account. Another brand new feature that has been implemented is the sorting feature for solutions. User's now have the ability to sort out the solutions. The solutions can be sorted by star weight, either ascending or descending. User's also have the ability to sort out solutions by a specific section or by the solutions name. Lastly a user can sort solutions based on if they have been completed, not completed, or a user can choose to display all the solutions. Completed solutions are sorted by attended, and incomplete solutions are sorted by not attended.

Unimplemented Features

The main feature that we were not able to incorporate into the CyclopediApp, was the popovers feature. The popovers feature was supposed to be implemented into the articles and the Eco Tracker tab. In the articles, the popovers were supposed to provide extra information when the user would hover over certain highlighted words shown in each of the articles present in the articles tab. In the Eco Tracker tab, popovers were going to be implemented on each of the solution cards. This popover on the solutions card would have shown the user extra information about the solution they'd had submitted and completed.

Links

GitHub Repository Link

<https://github.com/IRIDIXVdt/2021W-UBCO-COSC-499-Cyclops>