

DEAKIN UNIVERSITY

CAPSTONE TEAM PROJECT (B)

ONTRACK SUBMISSION

Company Handover and Showcase

Submitted By:

Elaheh KARIMI ZARANDI
ekarimizarandi

Tutor:

Kevin LEE

Group Members:

dkyrou	Diomedes	KYROU	ǚ ǚ ǚ
mengqianh	Qianhui	MENG	ǚ ǚ ǚ
gomesan	Anno	GOMES	ǚ ǚ ǚ
stitusmenacherr	Sam	TITUS MENACHERRY	ǚ ǚ ǚ
cdang	Cao Binh	DANG	ǚ ǚ ǚ
majoriti	Tim	MAJOR	ǚ ǚ ǚ
mtelley	Mark	TELLEY	ǚ ǚ ǚ
pastifo	Paul	ASTIFO	ǚ ǚ ǚ
africker	Ashley James	FRICKER	ǚ ǚ ǚ
truongt	The Tung	TRUONG	ǚ ǚ ǚ
s222181313	Pengyu	XIAO	ǚ ǚ ǚ
s222034126	Jikuan	LIU	ǚ ǚ ǚ
cjpark	Caitlin	PARKER	ǚ ǚ ǚ
ekarimizarandi	Elaheh	KARIMI ZARANDI	ǚ ǚ ǚ
hthrikawalabada	Harshana Thilanga	THRIKAWALA BADALGE	ǚ ǚ ǚ
mauryaa	Avinash	MAURYA	ǚ ǚ ǚ
leeyuns	Yunseong	LEE	ǚ ǚ ǚ
fengj	Jiahao	FENG	ǚ ǚ ǚ

June 1, 2023



Redback Operations Showcase Videos

Please find the showcase video links from each team below:

Company Overall showcase video : <https://www.youtube.com/watch?v=IdmxvHMHXcU>

IOT:

Functional threshold power exercise: <https://youtu.be/qDBMPSXhAb8>

Telegram IP Bot : [Video Instruction](#)

Fan speed depends on bike speed: <https://youtu.be/ibHfBDCcr-Q>

Strength and Endurance Workout: [Smart Bike project](#)

Heart-Rate Monitor Publish Big Fixed: <https://youtu.be/DfH4HdGneVw>



VR:

[showcase.mp4](#)

AI/Data:

https://video.deakin.edu.au/media/t/1_m91vtxy5

Mobile:

<https://www.youtube.com/watch?v=dOGvDk3e3F0>

Cyber:

<https://screenrec.com/share/iu0ApzNR6T>

Web-Design:

<https://www.youtube.com/watch?v=78Gabb9L2ho>

Devops:

https://video.deakin.edu.au/media/t/1_s3i2phd9

Redback Operations Showcase Videos

Please find the showcase video links from each team below:

Company Overall showcase video : <https://www.youtube.com/watch?v=IdmxvHMHXcU>

IOT:

Functional threshold power exercise: <https://youtu.be/qDBMPSXhAb8>

Telegram IP Bot : [Video Instruction](#)

Fan speed depends on bike speed: <https://youtu.be/ibHfBDCcr-Q>

Strength and Endurance Workout: [Smart Bike project](#)

Heart-Rate Monitor Publish Big Fixed: <https://youtu.be/DfH4HdGneVw>



VR:

[showcase.mp4](#)

AI/Data:

https://video.deakin.edu.au/media/t/1_m91vtxy5

Mobile:

<https://www.youtube.com/watch?v=dOGvDk3e3F0>

Cyber:

<https://screenrec.com/share/iu0ApzNR6T>

Web-Design:

<https://www.youtube.com/watch?v=78Gabb9L2ho>

Devops:

https://video.deakin.edu.au/media/t/1_s3i2phd9

Table of Contents

Project Smart Bike	9
1. Project Information	10
1.1.	Company Acting Director 10
1.2.	Project Team 10
2. Project Overview.....	<i>Error! Bookmark not defined.</i>
3. User Manual	<i>Error! Bookmark not defined.</i>
3.1.	-----Safety----- <i>Error! Bookmark not defined.</i>
3.2.	-----Maintenance----- <i>Error! Bookmark not defined.</i>
3.3.	-----Hardware Operation----- <i>Error! Bookmark not defined.</i>
3.4.	----- (If using Wahoo Mobile Application to update firmware etc) ----- <i>Error! Bookmark not defined.</i>
3.5.	-----Raspberry Pi ----- <i>Error! Bookmark not defined.</i>
3.6.	-----Connecting to Raspberry Pi Remotely----- <i>Error! Bookmark not defined.</i>
3.7.	-----IP Address----- <i>Error! Bookmark not defined.</i>
3.8.	-----Wahoo/Mongo/MQTT credentials----- <i>Error! Bookmark not defined.</i>
3.9.	-----Connecting to Wahoo Devices and Starting workout scripts----- <i>Error! Bookmark not defined.</i>
3.9.1. ----Heartrate monitor note-----	<i>Error! Bookmark not defined.</i>
3.10. ----Running the FTP Workout-----	<i>Error! Bookmark not defined.</i>
3.11. ----Running the Threshold/Strength/Endurance Workout-----	<i>....Error! Bookmark not defined.</i>
3.12. ----GCP hosted website for reviewing bike data-----	<i>..... Error! Bookmark not defined.</i>
3.13. ----Mobile and VR application (Not handled by T1 2023, included for archive purposes) -----	<i>Error! Bookmark not defined.</i>
4. Completed Deliverables.....	<i>Error! Bookmark not defined.</i>
4.1.	<i>Deliverable: Implement FTP system</i> <i>Error! Bookmark not defined.</i>

- 4.2. Deliverable: KICKR won't start/ stops publishing bug
Error! Bookmark not defined.
- 4.3. Deliverable: Integrate KICKR Speed MQTT output topic for use in changing FAN hardware speed **Error! Bookmark not defined.**
- 4.4. .. Deliverable: Limit/Throttle the sending of zero value data from Pi (to avoid GCP fees)
Error! Bookmark not defined.
- 4.5. Deliverable: Heart-Rate Publishing Bug
Error! Bookmark not defined.
- 4.6. Deliverable: Strength Workout
Error! Bookmark not defined.
- 4.7. Deliverable: Endurance Workout
Error! Bookmark not defined.
- 4.8. 4.8 Deliverable Threshold Workout
Error! Bookmark not defined.
5. **Roadmap** **Error! Bookmark not defined.**
- 5.1. Improve authenticity/modularity of Strength/Endurance/Ramped/Threshold workouts on pi..... **Error! Bookmark not defined.**
- 5.1.1. Description: **Error! Bookmark not defined.**
 5.1.2. Requirements: **Error! Bookmark not defined.**
- 5.2. GUI (Graphical User Interface) application that displays bikes current metric values/updates resistance/incline in real-time (AKA 'free roam-mode' for user): **Error! Bookmark not defined.**
- 5.2.1. Description: **Error! Bookmark not defined.**
 5.2.2. Requirements: **Error! Bookmark not defined.**
- 5.3. Create custom workouts via GUI (feature extension for GUI application):
Error! Bookmark not defined.
- 5.3.1. Description: **Error! Bookmark not defined.**
 5.3.2. Requirements: **Error! Bookmark not defined.**
- 5.4. Integrating Mobile App and IoT processes to start workouts via MQTT (Integration with mobile team):..... **Error! Bookmark not defined.**
- 5.4.1. Description: **Error! Bookmark not defined.**
 5.4.2. System Requirements:..... **Error! Bookmark not defined.**
 5.4.3. Sub System Requirements:..... **Error! Bookmark not defined.**
 5.4.4. Overall:..... **Error! Bookmark not defined.**
 5.4.5. Idea for Mobile App to:..... **Error! Bookmark not defined.**
- 5.5. Throttle KICKR script from sending zero value data (Save GCP spending further)
Error! Bookmark not defined.
- 5.5.1. Description: **Error! Bookmark not defined.**
 5.5.2. Requirements / Recommendations..... **Error! Bookmark not defined.**
- 5.6. Create Source Control/integration documentation/video for using a Branch style Workflow compared to a Fork/PR workflow for Team..... **Error! Bookmark not defined.**

5.6.1. Description	<i>Error! Bookmark not defined.</i>
5.6.2. Requirements:.....	<i>Error! Bookmark not defined.</i>
6. Open Issues	<i>Error! Bookmark not defined.</i>
6. Lessons Learned	<i>Error! Bookmark not defined.</i>
7. Product Development Life Cycle.....	<i>Error! Bookmark not defined.</i>
7.1.	New Tasks <i>Error! Bookmark not defined.</i>
7.2.	Definition of Done <i>Error! Bookmark not defined.</i>
7.3.	Task Review <i>Error! Bookmark not defined.</i>
7.4.	Testing <i>Error! Bookmark not defined.</i>
7.5.	Branching Strategy <i>Error! Bookmark not defined.</i>
7.5.1. Current system:.....	<i>Error! Bookmark not defined.</i>
7.5.2. Future Recommendations:.....	<i>Error! Bookmark not defined.</i>
8. Product Architecture.....	<i>Error! Bookmark not defined.</i>
8.1.	UML Diagram <i>Error! Bookmark not defined.</i>
8.2.	Tech Stack <i>Error! Bookmark not defined.</i>
9. Source Code	<i>Error! Bookmark not defined.</i>
10. 11. Login Credentials	<i>Error! Bookmark not defined.</i>
11. Other Relevant Information.....	<i>Error! Bookmark not defined.</i>
12. Appendices.....	<i>Error! Bookmark not defined.</i>
Project ----Deployment of Redback website on GCP.....	89
13. Project Information	90
13.1. Company Acting Director.....	90
13.2. Project Team	90
14. Project Overview.....	90
15. User Manual	90
16. Completed Deliverables	91

17. Roadmap	99
18. Open Issues	99
19. Lessons Learned	100
20. Product Development Life Cycle.....	100
20.1. New Tasks	100
20.2. Definition of Done.....	100
20.3. Task Review	101
20.4. Testing	101
20.5. Branching Strategy	102
21. Product Architecture.....	103
21.1. UML Diagram.....	103
21.2. Tech Stack.....	103
22. Source Code	104
23. Login Credentials.....	104
24. Appendices.....	104
Mobile App.....	105
25. Project Information	106
25.1. Company Acting Director.....	106
25.2. Project Team	106
26. Project Overview.....	106
27. User Manual	107
28. Completed Deliverables	107
29. Roadmap	109
30. Open Issues	112
31. Lessons Learned	112
32. Product Development Life Cycle.....	113
32.1. New Tasks	113
32.2. Definition of Done.....	113
32.3. Task Review	113

32.4. Testing	113
32.5. Branching Strategy	113
33. Product Architecture.....	114
33.1. UML Diagram.....	114
33.2. Tech Stack.....	114
34. Source Code	115
35. Login Credentials.....	115
36. Other Relevant Information.....	115
37. Appendices.....	115
Project 2-factor authentication	116
38. Project Information	117
38.1. Company Acting Director.....	117
38.2. Project Team	117
39. Project Overview.....	117
39.1. What is the project about?	117
39.2. What problem is the project solve?	117
39.3. What are the aims of the project?.....	117
39.4. What are the deliverables?.....	118
40. User Manual	118
41. Completed Deliverables.....	119
42. Roadmap	120
43. Open Issues	120
44. Lessons Learned	120
45. Product Architecture.....	120
45.1. UML Diagram.....	120
45.2. Tech Stack	121
46. Source Code	121
47. Appendices.....	121
Project: Implementation of SIEM	122

48. Project Information	123
48.1. Company Acting Director.....	123
48.2. Project Team	123
49. Project Overview.....	124
50. User Manual	124
50.1. Pre-requisites	124
50.2. Tasks Description – Wazuh Docker Deployment	124
50.3. Post-installation	125
50.4. Tasks Description – OpenCTI Docker Deployment.....	125
50.5. Post-installation.....	126
50.6. Tasks Description – ElasticSearch Docker Deployment	126
50.7. Post-installation.....	127
50.8. Tasks Description – Wazuh Agent Installation	127
51. Completed Deliverables.....	127
52. Roadmap	128
53. Open Issues	128
54. Lessons Learned	128
55. Product Development Life Cycle.....	128
55.1. New Tasks	128
55.2. Definition of Done.....	128
55.3. Task Review	129
55.4. Testing	129
56. Product Architecture.....	129
56.1. Tech Stack.....	129
57. Login Credentials.....	129

Company Redback Operations

Project Smart Bike

Trimester 1, 2023

1. Project Information

1.1. Company Acting Director

Name *Imali Dias*

Title *Dr*

Email imali.dias@deakin.edu.au

Name *Kevin Lee*

Title *Dr*

Email kevin.lee@deakin.edu.au

1.2. Project Team

Name	Level	Role
ELLA KARIMI ZARANDI	Senior	Team Leader
WILLIAM DJOJODIREDO	Junior	Team Member
JARROD YONG	Junior	Team Member
ETHAN BENJAMIN	Junior	Team Member
THOMAS MORGAN	Junior	Team Member
MOMIN KHALID BUTT	Junior	Team Member
HARIS SAJJAD	Junior	Team Member
AHMAD RIAZ	Junior	Team Member

Project Name *Smart Bike*

Company *Redback Operations*

2. Project Overview

The Smart Bike project developed an IoT enabled indoor exercise bike which takes the sensor data and delivers it to the cloud, where it can be used by the mobile application for exercise, or to deliver an immersive VR experience in Project Sun Cycle.

The project solves the problem of exercising, where cold weather or rain can diminish motivation to exercise outdoors, and obstacles such as traffic or pedestrians can make execution of an exercise plan difficult. The integration with VR makes indoor cycling a viable alternative and far more interesting than staring at four walls, alongside the future integration with the Mobile app for users to also enjoy a cycling experience without a headset if needed.

Two bikes were developed at Burwood Campus which obtain speed, cadence, power readings from a commercial trainer, and heart rate readings from a chest strap. The resistance and incline of the bike can be manipulated manually or through the operation of exercises contained in the raspberry pi interface.

Bike data is received on a raspberry pi via Bluetooth BLE connection, and transmitted to the cloud via MQTT to a broker which distributes the data to relevant processes and applications. These include the Flutter mobile application which, in future, can start and stop a workout, increase or decrease its intensity and display the sensor readings, the Unity VR game which can use the readings to alter the speed of the avatar, and a service to store them in a database for viewing by the web hosted content management system.

There are also internal workouts implemented on the pi, such as FTP measurement, strength, endurance and threshold power workouts. These are started via scripts, with hopes to use the mobile app in future to start these scripts via MQTT.

Work this trimester has been based around fixing known hardware/software issues related to MQTT, hardware connectivity and implementing exercises internally on the pi for use on the bike.

3. User Manual

3.1. -----Safety-----

Note that due to OH&S regulations it is not currently permitted to ride the bikes. The bikes need to be set up according to the Wahoo user manual to ensure safe and correct operation. This involves changing the pads on the legs of the trainer to the large size to support users larger than 81 kg, abducting them as wide as possible and ensuring they are clicked into place to prevent falling.

3.2. -----Maintenance-----

If the bike does not have an 11-speed cassette, it must be changed using the chain whip tool to fit on the trainer. The quick release system can be used to remove the back wheel of the bike, and then attach the rear of the bike to the trainer. An appropriately sized spacer should be used. The height adjustment of the trainer should be set appropriately to the size of the bike used.

The front wheel of the bike can similarly be removed, and the quick release system and spacer provided with the incline simulator used to attach the front of the bike.

3.3. -----Hardware Operation-----

The seat of the bike should be adjusted to an appropriate height for the user.

The cadence sensor can be installed on the crank using the provided clip.

The fan should be set up in front of the bike and all devices can be plugged in.

The chest strap should be removed from the case, the strap adjusted to a comfortable length and clipped around the chest, providing the electrodes contact with the skin.

The QR codes should be placed near the appropriate bike. These currently have the payloads {"bikeld":"000001"} and {"bikeld":"000002"}. If these are ever printed again, be careful to use ASCII double quotes (0x34), not curly quotes or other Unicode quotes which are not parsed properly by the application.

3.4. ----- (If using Wahoo Mobile Application to update firmware etc)-----

A mobile phone should be used to find and pair each device using the Wahoo fitness application. It may be necessary to move the pedals or use the control for the incline device for the sensors to enter pairing mode. This allows checking for correct operation and updating the firmware, although some parts of the software relied on sniffing out the proprietary Bluetooth protocol for the trainer and fan, so there is a potential that these may cease to function with future updates.

3.5. -----Raspberry Pi -----

The Raspberry Pi should be powered on. Upon startup, the pi requires a username and password (Given by leadership team)

The file /etc/wpa_supplicant/wpa_supplicant.conf should be edited to allow internet access depending on the environment where the bikes are used. In the RIOT Lab, the RIOT Lab SSID

is preferred to Eduroam, which requires a student's Deakin username and password to be stored unencrypted in the file on the SD card.

3.6. ----Connecting to Raspberry Pi Remotely----

Team members can connect to the raspberry pi remotely by using SSH in a terminal of their choice. To do this, users must install cisco AnyConnect VPN via the Deakin cloud site. Starting this VPN application, selecting Unencrypted and entering the student's login details will enable the connection to the Deakin Wifi, this way we are accessing the pi via a remote connection.

Below is a step-by-step instruction:

Once connected, in a Terminal (mingw, bash,powershell,cmd etc)

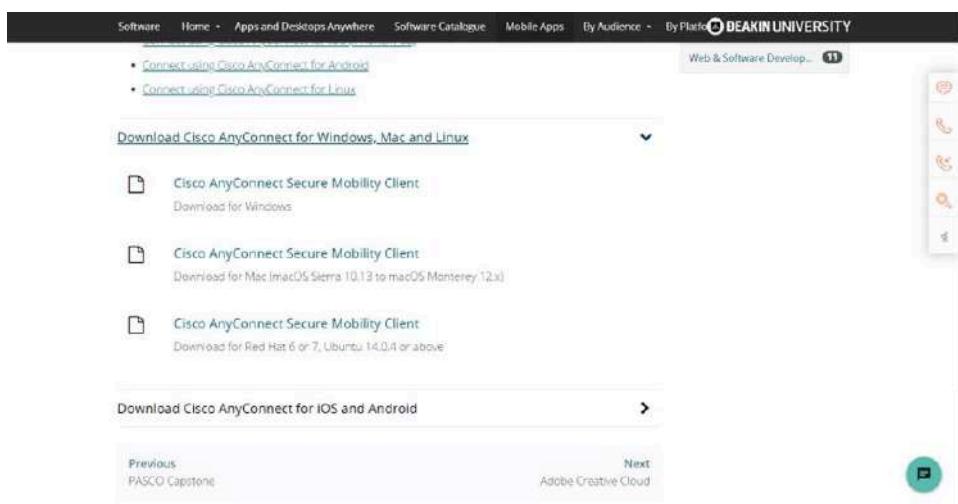
Enter: 'ssh pi@[ip_address_of_pi]

This will prompt for the pi's password,

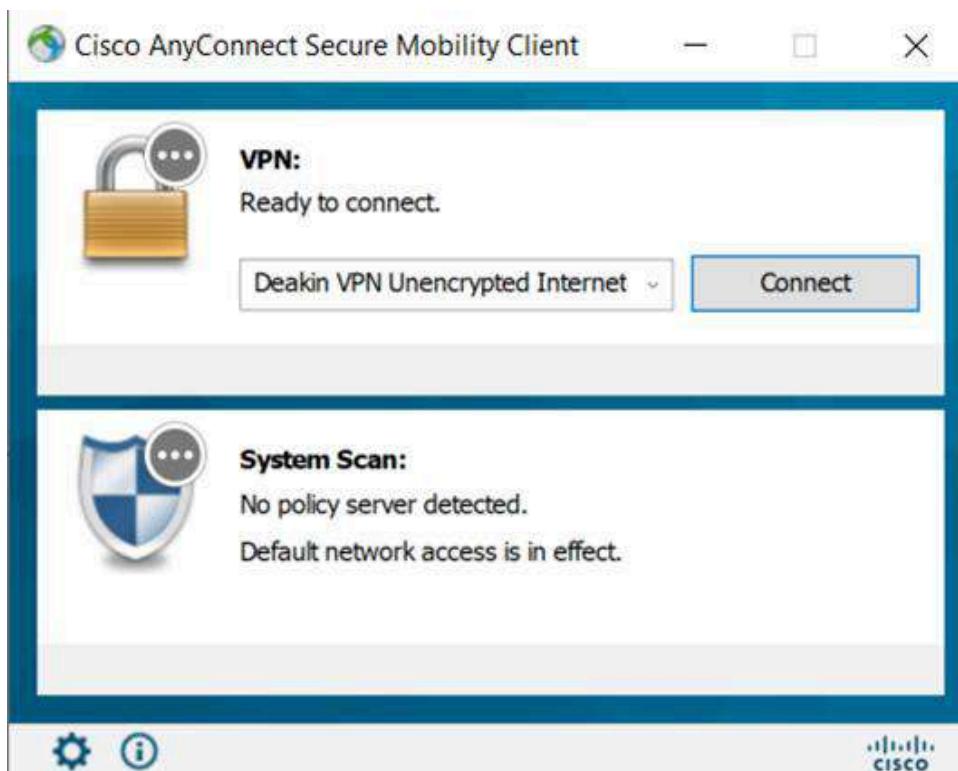
IP address information will be shared to students or accessed via going to the lab and logging into the pi, details are below.

Here is instruction manual to connect to Raspberry Pi remotely:

1. <https://software.deakin.edu.au/2019/04/16/cisco-anyconnect/>. First thing first, we need to download the Cisco AnyConnect (Deakin VPN) from the link above.



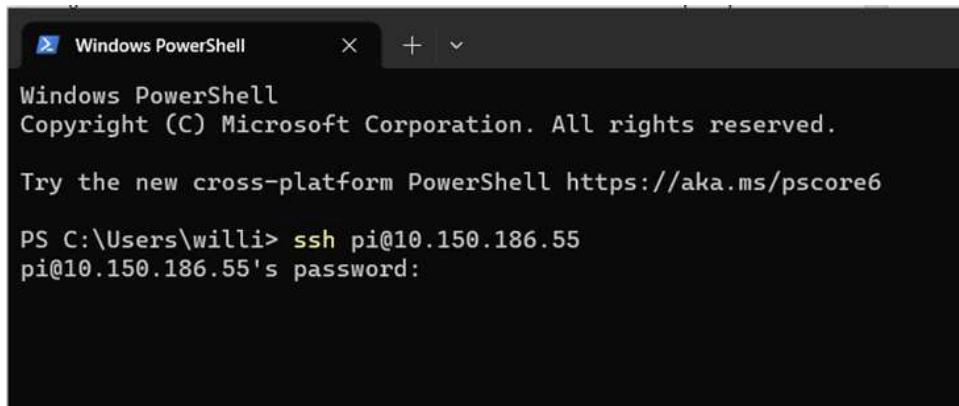
2. Make sure that you download link for your Operating System.



3. Open Cisco AnyConnect and press "connect". Make sure you have your Deakin's credentials ready because it will redirect you to Deakin login page



4. The interface should show like this after you successfully connect to Deakin VPN
5. Open terminal/command line in your computer

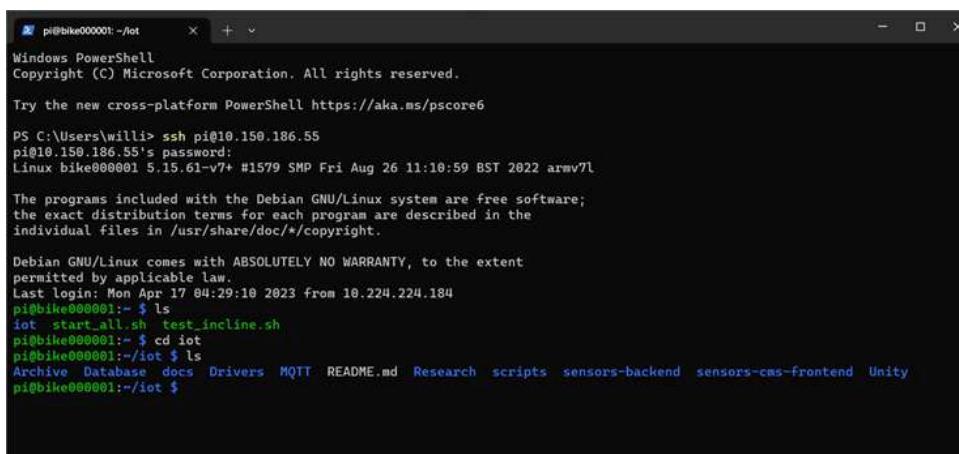


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\willi> ssh pi@10.150.186.55
pi@10.150.186.55's password:
```

6. Type “`ssh pi@{{ip_address_of_pi}}`” in your terminal and it will prompt you a password.



```
pi@bike000001:~$st
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\willi> ssh pi@10.150.186.55
pi@10.150.186.55's password:
Linux bike000001 5.15.61-v7+ #1579 SMP Fri Aug 26 11:10:59 BST 2022 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Apr 17 04:29:10 2023 from 18.224.224.184
pi@bike000001:~$ ls
iot start_all.sh test_incline.sh
pi@bike000001:~$ cd iot
pi@bike000001:~/iot$ ls
Archive Database docs Drivers MQTT README.md Research scripts sensors-backend sensors-cms-frontend Unity
pi@bike000001:~/iot$
```

7. Enter “`capstone374`” as the password and you have now connected to Raspberry Pi through SSH.

3.7. -----IP Address-----

Research was done into assigning a static IP to the Pi but unfortunately, we did not receive one. The solution to this was the creation of a set of scripts that monitors the Pi’s IP address and uses a Telegram Bot to publish it when changed. The scripts have been tested on a dev Pi off campus but we are currently awaiting feedback from IP support as the Telegram Bot API seems to be blocked on campus.

For more information, a showcase video is provided.

3.8. -----Wahoo/Mongo/MQTT credentials-----

The Raspberry Pi contains a file `/home/pi/.env` which contains the credentials for MQTT, MongoDB, the identifiers for the hardware it searches for (Bluetooth alias and MAC addresses of the sensors) and the Device ID. These must be kept up to date if these details change, and need to be sourced using `source ~/.env` to load the configuration when running commands from the command line, although they are called automatically when running the scripts that include the dotenv library (for ex: `os.getenv("MQTT_HOSTNAME")`) found in some driver scripts retrieve the hostname variable string from the `.env` file stored on the pi.

3.9. ----Connecting to Wahoo Devices and Starting workout scripts----

It is recommended to open separate SSH connections with the Raspberry Pi to run each process in a separate window.

Change directory to the `iot/scripts` folder and run `./start_all.sh` script to execute all scripts that begin connecting bike hardware to the pi for value processing/hardware manipulation.

If individual hardware needs to be monitored or tested, you can run each hardware process individually by calling `start_heartrate.sh`, `start_kickr.sh`, and/or `start_fan.sh` to begin pi connection to the bike hardware.

Also as a note:

You must ensure that the incline controller that is attached on or below the handlebar is set to 'unlocked' position when running the incline/resistance script for remote control of the incline functionality.

You can also locate all the workout and individual wahoo sensors scripts in the `iot/Driver` directory.

If the Kickr script fails to connect, restart the pi and turn all wahoo sensors off and on again. All sensors need to be in range and switched on to connect to the pi.

3.9.1. ----Heartrate monitor note----

There are several heartrate monitor sensors in the drawers of the RioT Lab, these have MAC addresses written on the boxes.

The pi with bike ID 000001 has a specific mac address associated with a specific heartrate monitor. The details of what hardware is being used for a pi is in the hidden `.env` file. With regards to which one is which the EB09 one is connected to the bike with ID 000001.

3.10. ----Running the FTP Workout----

Setting up:

Make sure the kickr is online and connected to the pi via the 'start_kickr.sh' script and the hardware is running/sending data, as these scripts depend on values being received via the bike/device/power topic (Can use `publish_tester.py` script on a windows machine to manually publish power data)

Run

Located in the iot/Drivers/FTP directory, run 'python3 FTP_workout.py' in the Drivers/FTP/ directory. The default minute duration is 20 minutes, add a command line arg after the script to set minutes example : python3 FTP_workout.py 2 (sets duration to 2 minutes)

Output:

Once the exercise is complete, the terminal will output the FTP score to the screen, but will also save the FTP score in the .env file stored on the pi. This FTP score can be accessed for use in any other script via utilising the dotenv library, loading the .env file and instantiating a variable with os.getenv(FTP SCORE)

Troubleshooting

if you are not sure if any MQTT messages are being sent and received, run the subscriber.py this should listen for messages from the power topic.

Ensure the pi is receiving bike data via the ./start_kickr.sh script in another terminal window.

3.11. ----Running the Threshold/Strength/Endurance Workout----

These operate the same as the FTP workout, upon running each of these scripts, the user will be asked to enter an initial resistance/incline/threshold value. Upon enter, the workout begins, increasing the resistance/incline/Power until it reaches its max point. The user can enter values for the time which is up to 20 minutes and the resistance ranging from 0 to 100% and the resistance will increase in the strength workout program every 2 minutes by 5% for the Strength Workout Program. The same time limit of 20 minutes is set for the endurance workout program which can be input at the start of the program and incline values are asked ranging from -10 to 19 which will also increase by 1 after every minute with a step of 0.5. The user will be prompted in the start of each workout to enter these values.

3.12. ----GCP hosted website for reviewing bike data----

The CMS is currently hosted at <http://34.129.10.237> that is under the company owned GCP account.

The dashboard shows an overview of the data collected and displays the recent stored sensors data in real time (if there's any activity using the mobile cycling fitness app), the bikes page can be used to set up additional bikes, the devices page used to provide details for additional sensors, and device data used to search or edit the collected data.

3.13. -----Workouts -----

The mobile phone application requires a Flutter installation, instructions here are for using Android Studio. Details on how to achieve this are given in the app-frontend repository. The extra-data-fields branch of the backend should also be downloaded and started on a laptop.

The IP address of this laptop needs to be saved into the frontend code in lib/constants/dart.conf and the application built using flutter dev apk –release –no-sound-null-check and then deployed to the phone. This builds the Android release version, however a bug prevented this from successfully logging in to the backend. While the fix is being tested, it is instead recommended to use a Mac and compile for IOS.

Once the app is running, it is possible to create a new user, or log in with the existing one. Clicking on start workout allows the workout type to be selected, for ramped workout is the QR code for the bike being used should be scanned.

After the 5 second timer, the workout begins. For the ramped workout, the bike begins at 24% resistance and 0 degrees incline and increases every 30 seconds by 4% resistance and 1 degree incline, until it reaches the maximum of 100% resistance and 19 degrees incline. The increase and decrease intensity buttons are not available during the workout as the intensity is continually increased automatically. The user should cycle continuously until they reach exhaustion, at which point they must press finish workout to end the workout and view the summary screen.

The other workouts are the strength, endurance and threshold workouts which were implemented in this trimester and details are as below:

The strength workout program that lets users input their preferred workout length (up to 20 minutes) and choose a resistance level from 0 to 100%. The programme uses a special progression system that raises the resistance by 5% every two minutes to offer a difficult and advanced training environment. It consists of a variety of exercises that target various muscle groups, enhancing both cardiovascular and physical fitness. The programme can be used by people of various fitness levels, and users can change the resistance to meet their needs and objectives. People may simply track their progress and modify their workout settings for a tailored training session thanks to its user-friendly UI.

The Endurance workout programme that lets users input their preferred workout time (up to 20 minutes) and choose the starting incline from a range of -10 to 19. The programme uses a special progression mechanism that raises the incline by 1 with a 0.5 step size per minute. This progressive gradient increase puts users' stamina to the test and simulates various terrains for a challenging and intense workout.

The VR files are now hosted on the GitHub repository, previously they were stored on a Plastic SCM yet admin have increased the size of files for the GitHub allowing large file transfers.

The Windows build has been tested and works with keyboard controller. To enable receipt of MQTT commands from the bike, find the player avatar (called Skateboard Shader Cel) and enable the checkbox for UseBikeSpeed in the Inspector. This instructs Unity to connect to the MQTT broker (using the credentials in the MQTT GameObject) and subscribe to the bike's topics. When speed data is received, it is scaled and applied to the forward transform of the

avatar, although keyboard or VR controller is still required for steering sideways or backwards.

4. Completed Deliverables

4.1. Deliverable: Implement FTP system

Description: To measure functional threshold power (FTP) in order to provide more personal workouts for users. This will result in users being able to get the most out of the smart bike by training to their maximum sustainable workload.

- Measuring FTP (Functional Threshold Power) is essential for cyclists who want to improve their performance and train effectively.
- Monitor progress: Measuring FTP regularly allows you to track your progress and assess the effectiveness of your training program. Improvements in FTP over time indicate an increase in fitness level.
- Optimize pacing: FTP is also helpful for pacing during races or time trials. By knowing your FTP, you can set a sustainable pace that maximizes your performance without burning out too soon.

Key Progress: completed (to run, check User Manual)

FTP Measurement Workout finalised and implemented. FTP measures the Functional Threshold Power of a rider, meaning the longest sustained power output over time.

The Trello board updated with adding description in the ticket for others to understand the use case and reasoning for a user to want an FTP measurement.

FTP workout contains:

- 20 Minute (or custom set time) non-stop ride at riders maximum push capacity
- Measure power in watts per second
- Average of every measurement per second * 0.95 == FTP score
- FTP score stored in .env file in pi to use in further workouts that allow riders to aim in increasing their FTP.

<https://trello.com/c/dp4VC7T5/62-implementing-ftp-measurement-system>

<https://github.com/redbackoperations/iot/pull/65>

<https://github.com/redbackoperations/iot/pull/74>

FTP WORKOUT:

```

1  #!/usr/bin/env python3
2  #! bin/bash
3  #! bin/sh
4  import sys
5  import time
6  import os
7  from mqtt_client import MQTTClient
8  from FTP_class import FTP
9  from dotenv import load_dotenv, set_key
10
11 def perform_ftp_test(ftp_object):
12     ## Reads previously saved FTP value from the .env file
13     print("Current FTP: ", ftp_object.get_ftp())
14     print("Starting FTP test in 5 seconds...")
15     time.sleep(5)
16
17     start_time = time.time()
18     try:
19         #NOTE: adding a value in the command line will set the duration of the test in minutes
20         while time.time() - start_time < (ftp_object.duration*60):
21             # Counts up to the duration set by the user
22             time.sleep(1)
23             if(ftp_object.power_data != None):
24                 print("Current power: ", ftp_object.power_data[-1])
25                 print("Current time: ", time.time() - start_time)
26                 # adds the current power to the power data list in FTP object
27                 ftp_object.power_data.append(ftp_object.current_power)
28             else:
29                 print("No power data received")
30
31     except KeyboardInterrupt:
32         print("Test stopped")
33         print("Count of data points given: " + str(len(ftp_object.power_data)))
34         pass

```

```

36 def set_workout_duration(ftp_object) -> FTP:
37     # reads command line arg for setting duration of workout
38     if len(sys.argv) > 1:
39         ftp_object.duration = int(sys.argv[1])
40         print("Duration set to " + str(sys.argv[1]) + " minutes")
41     else:
42         # default duration is 20 minutes (no argument given)
43         ftp_object.duration = 20
44         print("Duration not specified, defaulting to 20 minutes")
45
46 def main():
47     try:
48         # Load environment variables from pi's .env file
49         # This is necessary to get the MQTT credentials
50         # The .env file is not included in the repository
51
52         #Instantiate FTP object and initialize duration to user set parameter
53         env_path = '/home/pi/.env'
54         load_dotenv(env_path)
55         ftp_object = FTP()
56         ftp_object._init_()
57         global mqtt_client
58         global devicedId
59         set_workout_duration(ftp_object)
60
61         # Initialize MQTT client and subscribe to power topic
62         mqtt_client = MQTTClient(os.getenv('MQTT_HOSTNAME'), os.getenv('MQTT_USERNAME'), os.getenv('MQTT_PASSWORD'))
63         devicedId = os.getenv('DEVICE_ID')
64         topic = f'bike/{devicedId}/power'
65         print(topic)
66         mqtt_client.setup_mqtt_client()
67         mqtt_client.subscribe(topic)
68         mqtt_client.get_client().on_message = ftp_object.read_remote_data
69         mqtt_client.get_client().loop_start()
70

```

```

72         # Start FTP test
73         print("Starting the FTP test...")
74         perform_ftp_test(ftp_object)
75         ftp_object.calculate_ftp()
76         result = ftp_object.get_ftp()
77         print(f"Your estimated FTP is: {result:.2f} watts")
78         print("Test complete, saving FTP to file...")
79         set_key(env_path, 'FTP_SCORE', str(result))
80
81     except KeyboardInterrupt:
82         pass
83     mqtt_client.get_client().loop_stop()
84
85 if __name__ == "__main__":
86     main()
87

```

FTP CLASS:

```

1  import time
2  import json
3  import os
4  from dotenv import load_dotenv
5
6  class FTP():
7      def __init__(self):
8          load_dotenv('/home/pl/.env')
9          self.duration = 0
10         self.power_data = []
11         self.ftp = os.environ.get("FTP_SCORE")
12         self.current_power = 0
13
14     def set_ftp(self, ftp):
15         self.ftp = ftp
16
17     def get_ftp(self):
18         return self.ftp
19
20     def get_duration(self):
21         return self.duration
22
23     def set_duration(self, duration):
24         self.duration = duration
25
26     def get_power_data(self) -> list:
27         return self.power_data
28
29     def set_power_data(self, input_data) -> list:
30         self.power_data = []
31         for x in input_data:
32             self.power_data.append(x)
33
34     def calculate_ftp(self):
35         avg_power = sum(self.power_data) / len(self.power_data)
36         self.set_ftp(avg_power * 0.95)
37
38     # This is a callback function to be used a message is received via MQTT in the FTP_workout.py script,
39     # Its use case is only for the FTP workout mode, and it is not to be used in any other context
40     def read_remote_data(self, client, userdata, msg):
41         payload = msg.payload.decode("utf-8")
42
43         try:
44             # Attempt to parse the payload as JSON in line with incline and resistance script output
45             dict_of_payload = json.loads(payload)
46             power_value = dict_of_payload["value"]
47             temp = self.power_data[-1]
48             if temp != power_value:
49                 print("Received " + msg.topic + " " + str(msg.qos) + " " + str(msg.payload))
50             self.current_power = power_value
51         except json.JSONDecodeError:
52             # treat it as a singular string value
53             power_value = payload

```

Potential improvements:

Setting a default resistance value in the KICKR at the beginning of the exercise via publishing a value of 25 to the resistance topic or including a function for the user to enter a resistance to use before the exercise begins, the current resistance may be too weak.

Project Members: Thomas Morgan

4.2. Deliverable: KICKR won't start/stops publishing bug

It was an issue such that the incline/resistance control scripts that handled the initialisation of the KICKR connection to the raspberry pi was not working, this had been an issue for quite a while and became urgent to complete as MQTT data being sent to the CMS website required by other teams was necessary for completing tasks,

Key Progress: Fixed on the 11/04/2023

Several members worked several days in the lab of troubleshooting, testing and finally implementing a fix that has been documented in the current Trello board ticket, as below

- Seemed that the kickr device mac addresses needed updating
- Ran the Drivers/lib/ble_devices_scan.py script to scan for any BLE devices near rpi.
- Got mac address that aligned with wahoo kickr alias (prefix alias dependant on the kickr model)
- In raspberry pi, cd to home/pi
- Ran 'ls -la' to see hidden files
- Ran 'nano .env' to open hidden environment file in command line nano text editor (good luck)
- Navigated to and update KICKR_MAC_ADDRESS value to value found from running ble_devices_scan.py.
- Saved and exit

Also as a note:

You must ensure that the incline controller that is attached on or below the handlebar is set to 'unlocked' position when running the incline/resistance script for remote control of the incline functionality.

Link To Trello: <https://trello.com/c/GsZhExZl/89-kickr-wont-start-stops-publishing-bug>

Now when running the script that starts the kickr/incline, it connects and successfully publishes speed/cadence/power/resistance/incline data to the CMS sit via MQTT.

Team Members:

Ella Zarandi: Identified the issues, sprite planning, assigned tasks and deadlines

Ella Zazrandi: Tested/checked out the CMS to be sure the values >0 is also transported

William Djojodiregio ,Jarrod Yong, Thomas Morgan: troubleshooting, testing and finally implemented a fix and documented to the Trello board

4.3. Deliverable: Integrate KICKR Speed MQTT output topic for use in changing FAN hardware speed

Description: As it stands, the Wahoo fan is being controlled from the Mobile app or the Unity VR app, thus when running any scripts from the Pi, the fan doesn't function as a stand-alone

component of the bike and is tightly coupled with unnecessary integrations of applications. We want to have the KICKR speed MQTT output to be fed into the FAN python script, such that the Wahoo fan speed increases/decreases as the KICKR speed increases.

Reasoning

It makes more sense that the fan output is assimilated with the ‘speed of the bike’ for realism, regardless of Workout, App or VR game implementation. The windspeed values from the App or VR can still manipulate the fan, if necessary, yet running as a standalone until based off the ‘speed of the cyclist’ makes the hardware not dependent on other applications.

Progress: Completed

When running the ./start_fan.sh script alongside the ./start_kickr.sh scripts the fan will now set its speed in relation to the speed of the kickr. William developed the ratio required to normalise the values needed to set the fan here:

<https://github.com/redbackoperations/iot/pull/73>

<https://trello.com/c/TfyGXVnW/98-integrate-kickr-speed-mqtt-output-topic-for-use-in-changing-fan-actual-speed>

The value received by the MQTT speed topic is typically between values of 0 to 20, where the fan is set to receive values anywhere between 0 and 100, a ratio has been made to convert the received speed value to a value that the fan will use to output the representative wind speed.

Updates message callback now implementing ratio for speed input:

```
# When a message is received from MQTT on the fan topic for this bike, it is received here
✓ def message(client, userdata, msg):
    payload = msg.payload.decode("utf-8") #msg received is speed of the bike in m/s
    print("Received " + msg.topic + " " + str(msg.qos) + " " + str(msg.payload))

    #Extract value from payload
    dict_of_payload = json.loads(payload)
    bike_speed = int(dict_of_payload["value"])
    print("Processed speed to set to device: ", bike_speed)
    if bike_speed != 0:
        if bike_speed < 0:
            print(f"Invalid speed in message: {msg}")
            return
        # Maximum bike speed is around 20 m/s, setting fan_speed according to bike_speed
        if bike_speed == 0:
            fan_speed = 0 # Minimum fan speed
        elif bike_speed > 0 and bike_speed <= 4:
            fan_speed = 20
        elif bike_speed > 4 and bike_speed <= 8:
            fan_speed = 40
        elif bike_speed > 8 and bike_speed <= 12:
            fan_speed = 60
        elif bike_speed > 12 and bike_speed <= 16:
            fan_speed = 80
        elif bike_speed > 16:
            fan_speed = 100 # Maximum fan speed
        else:
            print(f"Invalid speed in message: {msg}")
            return

        print(f"Setting speed to {fan_speed}")
        device.set_speed(fan_speed)
```

Team members:

Identified the issue and provided Solution: Ella Zarandi

Researched, solutions given and tested by Thomas Morgan, Implemented by William Djojodiredjo

4.4. Deliverable: Limit/Throttle the sending of zero value data from Pi (to avoid GCP fees)

Description: When the Drivers/Scripts that activate the hardware on the bike are executed, publishing of data begins (MQTT publishes sent to GCP). But due to the nature of the scripts, currently a lot of zero data is published particularly the fan.

We want to throttle the amount of zero's, perhaps limit to 10 before we skip the publishing until the value is changed.

Reasoning:

GCP can cost upwards of \$20 per day in network costs when the CMS service is enabled. Throttling this rate to use QOS 2 and only publish changed values may be a way to avoid this.

Key progress: completed for fan & heartrate (should also update KICKR script to include this functionality)

Fan output has been fixed. When both the `start_fan.sh` and `start_heartrate.sh` scripts are ran, a counter is implemented such that if zero (0) value data has been processed from the hardware more than 10 times before a value change, we do not publish the respective value to the MQTT topic/broker, thus saving GCP processing costs and also freeing up the CMS and other dependency applications from being flooded with redundant data.

<https://github.com/redbackoperations/iot/pull/79>

<https://github.com/redbackoperations/iot/pull/82>

<https://trello.com/c/EJQMHFsH/83-rate-limit-sending-data-from-pi-to-avoid-qcp-fees-3>

<https://trello.com/c/aInW4WBk/115-add-publishing-rate-limiting-to-heartrate-sensor>

Updated specific method that handles publishing of data in fan script:

```
def characteristic_value_updated(self, characteristic, value):
    if characteristic == self.enable_characteristic:
        # This remains for debugging, but never seems to be called
        # Instead Notifications enabled is received
        print(f"Updated enable: {value}")
    if characteristic == self.fan_characteristic:
        # The Fan has several payloads to report its speed, but when
        # Idle, it returns fd 01 xx 04, where xx is the speed (0 to 100)
        if len(value) == 4 and value[0] == 0xFD and value[1] == 0x01 and value[3] == 0x04:
            # Check for zero value and value of zero counter. Continue if value is not 0 or 0 limit not reached
            if not(value[2] == 0x00 and self.zeroCount >= self.zero_limit):
                # Check if value is 0 and if so inc the zero counter. If value is not 0 then reset 0 counter
                # Zero counter will always reset when non-zero data published so it is most efficient to reset the zero counter
                # every time a non-zero is published as opposed to checking if zero counter is already 0
                if value[2] == 0x00:
                    self.zeroCount += 1
                else:
                    self.zeroCount = 0
            reported_speed = value[2]
            topic = f"ulike/{deviceId}/fan"
            payload = self.mqtt_data_report_payload(reported_speed)
            mqtt_client.publish(topic, payload)
            print(f"Published speed: {reported_speed}")

def mqtt_data_report_payload(self, value):
    # TODO: add more json data payload whenever needed later
    return json.dumps({"value": value, "unitName": 'percentage', "timestamp": time.time(), "metadata": { "deviceName": platform.node() } })
```

Team Members:

Notified by and Suggestions from Ella Zarandi:

Implemented by Ethan Benjamin, additional research, and review by Thomas Morgan & Jarrod Yong

4.5. Deliverable: Heart-Rate Publishing Bug

Description: Heart rate seems to be published for a few seconds (and received by the subscribers), and after a few seconds it continues to report it is successfully published but nothing is received.

Reasoning: We want to eliminate any false positives of published data and enable accurate data transfer between sensor data and the MQTT broker, such that the VR team and Mobile team and accurately receive working sensor data for their products.

Key progress: Completed Heartrate now publishes all the time to the CMS.

Through a multitude of trouble shooting and figuring out what was wrong the problem was identified in the heartrate.py file. Initially it was found to be weird that it would publish for around 10 seconds and then stop so initial thoughts led to the MQTT broker failing after publishing for a while. However, by testing the subscriber and publishing outputs it was sending and held the connection. It was then found that it was missing a MQTT client loop so the code would not loop anymore readings after a few seconds. Upon adding this loop and removing some error lines the Heart-Rate Sensor now works and publishes to the CMS properly.

<https://trello.com/c/qPVXhr0/61-heart-rate-stops-publishing-bug>

<https://github.com/redbackoperations/iot/commit/fba463df6884c4445d52edf1a830960d798fe143>

https://github.com/redbackoperations/iot/blob/main/Drivers/heart_rate_sensor/heartrate.py

Part that had been updated with client loop

```
182
183     def main():
184         try:
185             adapter_name=os.getenv('HEART_RATE_ADAPTER_NAME')
186             alias_prefix=os.getenv('HEART_RATE_ALIAS_PREFIX')
187
188             global mqtt_client
189             global deviceId
190             mqtt_client = MQTTClient(os.getenv('MQTT_HOSTNAME'), \
191                 os.getenv('MQTT_USERNAME'), os.getenv('MQTT_PASSWORD'))
192             mqtt_client.setup_mqtt_client()
193             deviceId = os.getenv('DEVICE_ID')
194             mqtt_client.get_client().loop_start()
195
196             manager = AnyDeviceManager(adapter_name=adapter_name)
197             manager.prefix=alias_prefix
198             manager.start_discovery()
199             manager.run()
200         except KeyboardInterrupt:
201             pass
202         mqtt_client.get_client().loop_stop()
203
204
205     if __name__=="__main__":
206         main()
```

Team Members:

Identified by Adrian Grigo from Trimester 3 2022

Implemented by Jarrod Yong and William Djojodiredjo with additional help from Thomas Morgan

4.6. Deliverable: Strength Workout

Description: To increase the strength of cyclists the user will enter desired resistance percentage to start the workout and the duration they want to workout with a maximum limit of 20 minutes and after every 2 minutes the resistance will increase by 5% until it reaches 100% for the workout time. It is essential for users who want to improve their strength through cycling.

It is a fantastic way to improve leg strength as well. It engages more muscles than regular cycling. Your leg muscles are forced to work harder by increasing resistance, which gradually increases their strength.

Key Progress: completed (to run, check User Manual)

The Strength Workout has been completed and implemented. It works with the Resistance of the bike and push the strength of the cyclist.

The Trello board updated with adding description in the ticket for others to understand the use case and reasoning for a user to do a Strength Workout.

Strength Workout contains:

- 20 Minute (or custom set time) non-stop ride.
- Starts with user desired resistance.
- Increase Resistance after every 2 minute by 5%

<https://trello.com/c/ijMUERWJ>

https://github.com/redbackoperations/iot/tree/main/Drivers/Strength_WOrkout

Strength Workout:

```

11 MAX_WORKOUT_DURATION = 20 # Maximum duration of the workout in minutes
12
13 def perform_actions(resistance_level):
14     mqtt_client.publish(f"bike/000001/resistance/control", resistance_level)
15
16
17 def perform_strength_workout(strength_workout_object):
18     print("Starting strength workout in 5 seconds...")
19     time.sleep(5)
20
21     resistance_level = -1 # Initial value to enter the loop
22
23     while resistance_level < 0 or resistance_level > 100:
24         resistance_level = int(input("Enter the resistance percentage (0-100): "))
25
26         if resistance_level < 0 or resistance_level > 100:
27             print("Invalid resistance percentage. Please enter a value between 0 and 100.")
28
29
30     start_time = time.time()
31
32     try:
33         while True:
34             current_time = time.time() - start_time
35             if current_time >= (strength_workout_object.duration * 60) or current_time >= (MAX_WORKOUT_DURATION * 60):
36                 break
37
38             # Store the resistance level in the strength workout object
39             strength_workout_object.resistance_data.append(resistance_level)
40
41             # Perform the strength workout action based on the resistance level
42             perform_actions(resistance_level)
43
44             if current_time % 120 == 0 and resistance_level < 100:
45                 resistance_level += 5
46
47             time.sleep(1)
48
49     except KeyboardInterrupt:
50         print("Workout stopped")
51         print("Count of data points given: " + str(len(strength_workout_object.resistance_data)))
52
53 def set_workout_duration(strength_workout_object):
54     # Read the command line argument for setting the duration of the workout
55     if len(sys.argv) > 1:
56         strength_workout_object.duration = int(sys.argv[1])
57         if strength_workout_object.duration > MAX_WORKOUT_DURATION:
58             strength_workout_object.duration = MAX_WORKOUT_DURATION
59             print("Duration exceeds maximum limit of 20 minutes. Setting duration to 20 minutes.")
60         print("Duration set to " + str(strength_workout_object.duration) + " minutes")
61     else:
62         # Default duration is 20 minutes (no argument given)
63         strength_workout_object.duration = 20
64         print("Duration not specified, defaulting to 20 minutes")
65
66 def main():
67     try:
68         # Load environment variables from the .env file
69         env_path = '/home/pi/.env'
70         load_dotenv(env_path)
71
72         # Instantiate StrengthWorkout object and initialize duration to user-set parameter
73         strength_workout_object = StrengthWorkout()
74         strength_workout_object.__init__()
75
76         global mqtt_client
77         global deviceid
78
79         set_workout_duration(strength_workout_object)
80
81         # Initialize MQTT client and subscribe to resistance topic
82         mqtt_client = MQTTClient(os.getenv('MQTT_HOSTNAME'), os.getenv('MQTT_USERNAME'), os.getenv('MQTT_PASSWORD'))
83         deviceid = os.getenv('DEVICE_ID')
84         topic = f"bike/{deviceid}/resistance"
85         print(deviceid)
86         print(topic)
87         mqtt_client.setup_mqtt_client()
88         mqtt_client.subscribe(topic)
89         mqtt_client.get_client().on_message = strength_workout_object.read_remote_data
90
91         mqtt_client.get_client().loop_start()
92
93         # Start the strength workout
94         print("Starting the strength workout...")
95         perform_strength_workout(strength_workout_object)
96         print("Workout complete!")
97
98     except KeyboardInterrupt:
99         pass
100     mqtt_client.get_client().loop_stop()
101
102 if __name__ == "__main__":
103     main()

```

Strength Workout Class:

```

6  class StrengthWorkout():
7      def __init__(self):
8          load_dotenv('/home/pi/.env')
9          self.duration = 0
10         self.resistance_data = [0]
11         self.current_resistance = 0
12
13     def get_duration(self):
14         return self.duration
15
16     def set_duration(self, duration):
17         self.duration = duration
18
19     def get_resistance_data(self) -> list:
20         return self.resistance_data
21
22     def set_resistance_data(self, input_data) -> list:
23         self.resistance_data = []
24         for x in input_data:
25             self.resistance_data.append(x)
26
27     def calculate_strength(self):
28         avg_resistance = sum(self.resistance_data) / len(self.resistance_data)
29         return avg_resistance * 0.95
30
31     # This is a callback function to be used when a message is received via MQTT in the StrengthWorkout.py script.
32     # Its use case is only for the strength workout mode and should not be used in any other context.
33     def read_remote_data(self, client, userdata, msg):
34         payload = msg.payload.decode("utf-8")
35         try:
36             # Attempt to parse the payload as JSON in line with incline and resistance script output
37             dict_of_payload = json.loads(payload)
38             resistance_value = dict_of_payload["value"]
39             temp = self.resistance_data[-1]
40             if temp != resistance_value:
41                 print("Received " + msg.topic + " " + str(msg.qos) + " " + str(msg.payload))
42                 self.current_resistance = resistance_value
43             except json.JSONDecodeError:
44                 # Treat it as a singular string value
45                 resistance_value = payload
46

```

Potential Improvements:

More functionalities could be added to the workout once it is connected to the mobile application like the user can change resistance during the workout if its too hard or low for user. The proof of Concept has also been made and available on the github repository for the enhanced features after it is connected to the mobile application.

<https://github.com/redbackoperations/iot/blob/main/Drivers/Proof%20of%20Concepts/StrengthWorkout.cs>

TEAM MEMBERS: AHAMD RIAZ, MOMIN KHALID BUTT

4.7. Deliverable: Endurance Workout

Description: Cycling endurance can be increased by entering the desired incline at the beginning of the workout and the maximum training time of 20 minutes. After every minute, the resistance will increase by 1 until it reaches 19 (in the range of -10 to 19). Users who want to use cycling to increase their endurance must have it.

Key Progress: completed (to run, check User Manual)

The Endurance Workout has been finished and put into practise. It works with the bike's incline to raise the cyclist's inclination.

To help others understand the use case and justification for a user performing an endurance workout, the Trello board was updated.

Strength Workout contains:

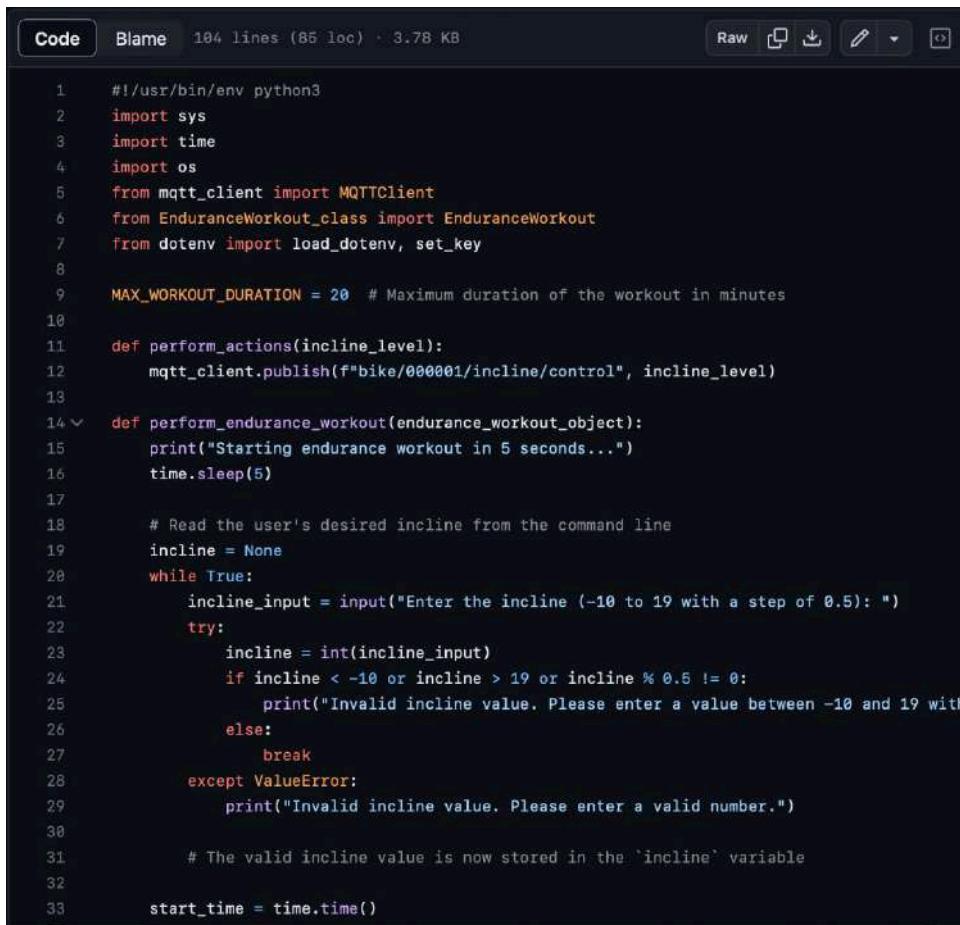
- 20 Minute (or custom set time) non-stop ride.
- Starts with user desired incline.

- Increase incline after every 1 minute by 1 with a step of 0.5.

<https://trello.com/c/PoTyIJ6/113-endurance-workout>

https://github.com/redbackoperations/iot/tree/main/Drivers/endurance_workout

Endurance Workout:



The screenshot shows a code editor interface with a dark theme. At the top, there are tabs for 'Code' (which is selected) and 'Blame'. Below the tabs, it says '104 lines (86 loc) · 3.78 KB'. On the right side of the header, there are several icons: 'Raw', a copy icon, a download icon, an edit icon, and a dropdown menu. The main area contains the following Python code:

```

1  #!/usr/bin/env python3
2  import sys
3  import time
4  import os
5  from mqtt_client import MQTTClient
6  from EnduranceWorkout_class import EnduranceWorkout
7  from dotenv import load_dotenv, set_key
8
9  MAX_WORKOUT_DURATION = 20 # Maximum duration of the workout in minutes
10
11 def perform_actions(incline_level):
12     mqtt_client.publish(f"bike/000001/incline/control", incline_level)
13
14 def perform_endurance_workout(endurance_workout_object):
15     print("Starting endurance workout in 5 seconds...")
16     time.sleep(5)
17
18     # Read the user's desired incline from the command line
19     incline = None
20     while True:
21         incline_input = input("Enter the incline (-10 to 19 with a step of 0.5): ")
22         try:
23             incline = int(incline_input)
24             if incline < -10 or incline > 19 or incline % 0.5 != 0:
25                 print("Invalid incline value. Please enter a value between -10 and 19 with")
26             else:
27                 break
28         except ValueError:
29             print("Invalid incline value. Please enter a valid number.")
30
31     # The valid incline value is now stored in the 'incline' variable
32
33     start_time = time.time()

```

```

34     try:
35         while True:
36             current_time = time.time() - start_time
37             if current_time >= (endurance_workout_object.duration * 60) or current_time >=
38                 break
39
40             # Store the incline, current time, and perform the endurance workout action
41             endurance_workout_object.incline_data.append(incline)
42             perform_actions(incline)
43
44             incline += 1
45             if incline > 19:
46                 incline = 19
47
48             time.sleep(60)
49     except KeyboardInterrupt:
50         print("Workout stopped")
51         print("Count of data points given: " + str(len(endurance_workout_object.incline_da
52
53 def set_workout_duration(endurance_workout_object):
54     # Read the command line argument for setting the duration of the workout
55     if len(sys.argv) > 1:
56         endurance_workout_object.duration = int(sys.argv[1])
57         if endurance_workout_object.duration > MAX_WORKOUT_DURATION:
58             endurance_workout_object.duration = MAX_WORKOUT_DURATION
59             print("Duration exceeds maximum limit of 20 minutes. Setting duration to 20 mi
60             print("Duration set to " + str(endurance_workout_object.duration) + " minutes")
61     else:
62         # Default duration is 20 minutes (no argument given)
63         endurance_workout_object.duration = 20
64         print("Duration not specified, defaulting to 20 minutes")

```

```

66 def main():
67     try:
68         # Load environment variables from the .env file
69         env_path = '/home/pi/.env'
70         load_dotenv(env_path)
71
72         # Instantiate EnduranceWorkout object and initialize duration to user-set parameter
73         endurance_workout_object = EnduranceWorkout()
74         endurance_workout_object.__init__()
75
76         global mqtt_client
77         global deviceId
78
79         set_workout_duration(endurance_workout_object)
80
81         # Initialize MQTT client and subscribe to incline topic
82         mqtt_client = MQTTClient(os.getenv('MQTT_HOSTNAME'), os.getenv('MQTT_USERNAME'), o
83         deviceId = os.getenv('DEVICE_ID')
84         topic = f'bike/{deviceId}/incline'
85         print(deviceId)
86         print(topic)
87         mqtt_client.setup_mqtt_client()
88         mqtt_client.subscribe(topic)
89         mqtt_client.get_client().on_message = endurance_workout_object.read_remote_data
90
91         mqtt_client.get_client().loop_start()
92
93         # Start the endurance
94         # Start the endurance workout
95         print("Starting the endurance workout...")
96         perform_endurance_workout(endurance_workout_object)
97         print("Workout complete.")
98
99     except KeyboardInterrupt:
100         pass
101         mqtt_client.get_client().loop_stop()
102
103 if __name__ == "__main__":
104     main()

```

Endurance Workout Class:

```
1  import time
2  import json
3  import os
4  from dotenv import load_dotenv
5
6  class EnduranceWorkout:
7      def __init__(self):
8          load_dotenv('/home/pi/.env')
9          self.duration = 0
10         self.incline_data = []
11         self.endurance_score = os.environ.get("ENDURANCE_SCORE")
12         self.current_incline = 0
13
14     def set_endurance_score(self, score):
15         self.endurance_score = score
16
17     def get_endurance_score(self):
18         return self.endurance_score
19
20     def get_duration(self):
21         return self.duration
22
23     def set_duration(self, duration):
24         self.duration = duration
25
26     def get_incline_data(self):
27         return self.incline_data
28
29     def set_incline_data(self, input_data):
30         self.incline_data = []
31         for x in input_data:
32             self.incline_data.append(x)
33
```

```
def calculate_endurance_score(self):
    total_incline = sum(self.incline_data)
    self.set_endurance_score(total_incline)

# This callback function is to be used when a message is received via MQTT in the Endurance_Workout mode
# Its use case is only for the endurance workout mode, and it is not to be used in any other context
def read_remote_data(self, client, userdata, msg):
    payload = msg.payload.decode("utf-8")
    try:
        # Attempt to parse the payload as JSON in line with incline data script output
        dict_of_payload = json.loads(payload)
        incline_value = dict_of_payload["value"]
        temp = self.incline_data[-1]
        if temp != incline_value:
            print("Received " + msg.topic + " " + str(msg.qos) + " " + str(msg.payload))
            self.current_incline = incline_value
            self.incline_data.append(incline_value)
    except json.JSONDecodeError:
        # Treat it as a singular string value
        incline_value = payload
        self.current_incline = incline_value
        self.incline_data.append(incline_value)
```

Potential Improvements: Once the workout is linked to the mobile application, more features might be added, such as the ability for the user to adjust the incline if it is too high.

or low for them while exercising. When connected to the mobile application, the proof of Concept for the upgraded functionality will also be made and made accessible on the GitHub repository.

https://github.com/redbackoperations/iot/blob/main/Drivers/Proof_of_Concepts/EnduranceWorkout.cs

TEAM MEMBERS

Suggested and Modified the project, Task assigned and tracked the progress: Ella Zarandi
Coded and Tested: AHAMD RIAZ, MOMIN KHALID BUTT

4.8. Deliverable Threshold Workout

Description: The threshold power workout code provides a framework for implementing a threshold power workout using a Python script.

It can be customized to perform specific actions when the threshold is exceeded, such as calculating the resistance required or triggering additional workout features.

The code provides a structure for organizing and managing the threshold power workout, including MQTT message handling, threshold monitoring, and resistance calculation.

Please note that the code is a starting point and may require customization and further development based on specific needs, equipment, and desired functionality. We can build upon this framework to add additional features, user interactions, data visualization, and more.

The program prints out the power generated every second. It also calculates the resistance required to reach maximum threshold.

Features:

- MQTT communication
- Number of intervals
- Resistance required to reach maximum threshold power
- Power generated every second

```
1 import os
2 from dotenv import load_dotenv
3
4 class ThresholdWorkout:
5     def __init__(self):
6         load_dotenv('/home/pi/.env')
7         self.duration = 0
8         self.power_data = []
9         self.threshold_workout = os.environ.get("THRESHOLD_WORKOUT_SCORE") # Updated variable name
10        self.current_power = 0
11        self.threshold_power = 0
12
13    def set_threshold_workout(self, threshold_workout): # Updated method name
14        self.threshold_workout = threshold_workout
15
16    def get_threshold_workout(self): # Updated method name
17        return self.threshold_workout
18
19    def get_duration(self):
20        return self.duration
21
22    def set_duration(self, duration):
23        self.duration = duration
24
25    def get_power_data(self):
26        return self.power_data
27
28    def set_power_data(self, input_data):
29        self.power_data = []
30        for x in input_data:
31            self.power_data.append(x)
32
33    def calculate_threshold_workout(self): # Updated method name
34        avg_power = sum(self.power_data) / len(self.power_data)
35        self.set_threshold_workout(avg_power * 0.95)
36
37    def read_remote_data(self, client, userdata, msg):
38        payload = msg.payload.decode("utf-8")
39        try:
40            dict_of_payload = json.loads(payload)
41            power_value = dict_of_payload["value"]
42            temp = self.power_data[-1]
43            if temp != power_value:
44                print("Received " + msg.topic + " " + str(msg.qos) + " " + str(msg.payload))
45                self.current_power = power_value
46                self.check_threshold()
```



```
Threshold_class.py
36
37     def read_remote_data(self, client, userdata, msg):
38         payload = msg.payload.decode("utf-8")
39         try:
40             dict_of_payload = json.loads(payload)
41             power_value = dict_of_payload["value"]
42             temp = self.power_data[-1]
43             if temp != power_value:
44                 print("Received " + msg.topic + " " + str(msg.qos) + " " + str(msg.payload))
45                 self.current_power = power_value
46                 self.check_threshold()
47             except json.JSONDecodeError:
48                 power_value = payload
49
50     def set_threshold_power(self, threshold):
51         self.threshold_power = threshold
52
53     def check_threshold(self):
54         if self.current_power > self.threshold_power:
55             print("Threshold power exceeded!")
56             resistance_required = self.calculate_resistance_required(self.current_power)
57             print("Resistance required to reach threshold power: ", resistance_required)
58             # Perform threshold workout action
59             # Replace this with your desired action when the threshold is exceeded
60
61     def calculate_power_usage_per_second(self, start_time, power_usage):
62         elapsed_time = time.time() - start_time
63         if elapsed_time > 0:
64             return power_usage / elapsed_time
65         else:
66             return 0
67
68     def calculate_resistance_required(self, current_power):
69         # Calculate the resistance required to reach the threshold power
70         # I've used a hardcoded method
71         if current_power < 100:
72             resistance_required = 5
73         elif current_power < 200:
74             resistance_required = 8
75         else:
76             resistance_required = 10
77         return resistance_required
78
```

```
❶ Threshold_workout.py
 1  #!/usr/bin/env python3
 2  import sys
 3  import time
 4  import os
 5  from mqtt_client import MQTTClient
 6  from Threshold_class import ThresholdWorkout
 7  from dotenv import load_dotenv, set_key
 8
 9  def perform_threshold_test(threshold_workout_object):
10      print("Current Threshold Workout: ", threshold_workout_object.get_threshold_workout())
11      print("Starting threshold test in 5 seconds...")
12      time.sleep(5)
13
14      start_time = time.time()
15      power_usage = 0
16      try:
17          while time.time() - start_time < (threshold_workout_object.get_duration() * 60):
18              time.sleep(1)
19              if threshold_workout_object.get_power_data() is not None:
20                  current_power = threshold_workout_object.get_power_data()[-1]
21                  power_usage += current_power
22                  print("Current power: ", current_power)
23                  power_usage_per_sec = threshold_workout_object.calculate_power_usage_per_second(start_time, power_usage)
24                  print("Power usage per second: ", power_usage_per_sec)
25                  if current_power > threshold_workout_object.threshold_power:
26                      print("Threshold power exceeded!")
27                      resistance_required = threshold_workout_object.calculate_resistance_required(current_power)
28                      print("Resistance required to reach threshold power: ", resistance_required)
29                  else:
30                      print("No power data received")
31
32      except KeyboardInterrupt:
33          print("Test stopped")
34          print("Count of data points given: " + str(len(threshold_workout_object.get_power_data())))
35          pass
36
37
38
39  def set_workout_duration(threshold_workout_object):
40      if len(sys.argv) > 1:
41          threshold_workout_object.set_duration(int(sys.argv[1]))
42          print("Duration set to " + str(sys.argv[1]) + " minutes")
43      else:
44          threshold_workout_object.set_duration(20)
45          print("Duration not specified, defaulting to 20 minutes")
```

```

47 def set_threshold_power(threshold_workout_object):
48     if len(sys.argv) > 2:
49         threshold_workout_object.set_threshold_power(int(sys.argv[2]))
50         print("Threshold power set to " + str(sys.argv[2]) + " watts")
51     else:
52         threshold_workout_object.set_threshold_power(200)
53         print("Threshold power not specified, defaulting to 200 watts")
54
55 def main():
56     try:
57         env_path = '/home/pi/.env'
58         load_dotenv(env_path)
59         threshold_workout_object = ThresholdWorkout() # Updated class name
60         threshold_workout_object.__init__() # Removed redundant initialization
61         global mqtt_client
62         global deviceId
63         set_workout_duration(threshold_workout_object)
64         set_threshold_power(threshold_workout_object)
65
66         mqtt_client = MQTTClient(os.getenv('MQTT_HOSTNAME'), os.getenv('MQTT_USERNAME'), os.getenv('MQTT_PASSWORD'))
67         deviceId = os.getenv('DEVICE_ID')
68         topic = f'bike/{deviceId}/power'
69         print(deviceId)
70         print(topic)
71         mqtt_client.setup_mqtt_client()
72         mqtt_client.subscribe(topic)
73         mqtt_client.get_client().on_message = threshold_workout_object.read_remote_data
74         mqtt_client.get_client().loop_start()
75
76         print("Starting the threshold test...")
77         perform_threshold_test(threshold_workout_object) # Updated variable name
78
79     except KeyboardInterrupt:
80         pass
81         mqtt_client.get_client().loop_stop()
82
83 if __name__ == "__main__":
84     main()

```

Github Link: <https://github.com/redbackoperations/iot/tree/main/Drivers>

The threshold_class file and the threshold_power workout file are the ones that are integrated into the main repo for redback operations

Github link:
[https://github.com/redbackoperations/iot/tree/main/Drivers/Proof of Concepts](https://github.com/redbackoperations/iot/tree/main/Drivers/Proof%20of%20Concepts)

Threshold workout as proof of concept is available through the link above

Team members:

Suggested and Modified the project, Task assigned and tracked the progress: Ella Zarandi
Coded and tested : Harris Sajjad

4.9. Deliverable IP Address Telegram Bot

Description: Building on the need for remote access, the IP of the Pi must be known. Unfortunately, the Pi is on Deakin's DHCP server and we have been unsuccessful in our request for a static IP. An alternative solution was needed to always have the latest IP of the Pi.

Completed by Ethan Benjamin

Conducted Brainstormed on an alternative solution and played with how to record the IP of the Pi. First aim was publishing the IP to GitHub but this would be a security risk. Final

decision was to use a Telegram bot to publish any IP changes when they happen. The bot publishes the updates to a Telegram group which can be joined using this link - <https://t.me/+YI175e65hdo5M2I0>.

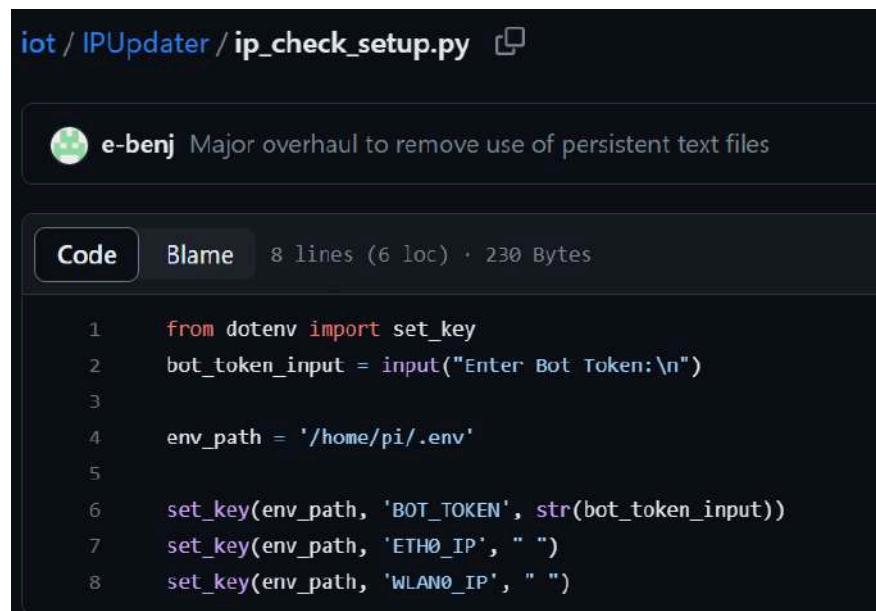
There are a total of three scripts for this system. Two python and one bash script. The first python script allows the user to setup the system by adding the Telegram Bot ID and then creates the necessary env variables.

The bash script is then used to be triggered by a cron task at every designated interval. It simply uses grep to record the network interface details to a temporary .txt file. Once this is done, it triggers the second python script which will read in the details from the .txt file and delete it once done. It then compares the IP from the file to the env variable and if necessary, pushes a Telegram message that the IP has changed.

The scripts were tested on my personal Pi and were then merged onto the Pi in the lab. We ran into an issue where the Telegram API is blocked by the Deakin Firewall but a request to create an allowance was submitted and approved by our tutor so it will be implemented by the IT support desk.

Trello Ticket - <https://trello.com/c/Ay76ZRIf/105-create-and-set-a-static-ip-address-for-the-raspberry-pi>

GitHub Commit - <https://github.com/redbackoperations/iot/pull/78>



A screenshot of a GitHub commit page for the file `ip_check_setup.py`. The commit was made by `e-benj` and is titled "Major overhaul to remove use of persistent text files". The code tab is selected, showing the following Python code:

```
1  from dotenv import set_key
2  bot_token_input = input("Enter Bot Token:\n")
3
4  env_path = '/home/pi/.env'
5
6  set_key(env_path, 'BOT_TOKEN', str(bot_token_input))
7  set_key(env_path, 'ETH0_IP', " ")
8  set_key(env_path, 'WLAN0_IP', " ")
```

[iot](#) / [IPUpdater](#) / **check_ip.sh** 



e-benj Major overhaul to remove use of persistent text files

Code

Blame

12 lines (9 loc) • 283 Bytes

```
1 ifconfig eth0 | grep 'inet' > eth0_temp.txt
2 ifconfig wlan0 | grep 'inet' > wlan0_temp.txt
3
4 new_eth0=$(head -1 eth0_temp.txt)
5 rm eth0_temp.txt
6 echo $new_eth0 > eth0_temp.txt
7
8 new_wlan0=$(head -1 wlan0_temp.txt)
9 rm wlan0_temp.txt
10 echo $new_wlan0 > wlan0_temp.txt
11
12 python3 ip_updater.py
```

iot / IPUpdater / ip_updater.py



e-benj Correct minor bug

Code

Blame 44 lines (34 loc) · 1.33 KB

```
1 import os
2 import telebot
3 from dotenv import load_dotenv, set_key
4 import subprocess
5
6 env_path = '/home/pi/.env'
7 load_dotenv(env_path)
8
9 #get bot token and device ID
10 BOT_TOKEN = os.environ.get('BOT_TOKEN')
11 DEVICE_ID = os.environ.get('DEVICE_ID')
12
13 #create telegram bot instance
14 bot = telebot.TeleBot(BOT_TOKEN, parse_mode=None)
15
16 #get existing IPs
17 curr_eth0 = os.environ.get('ETH0_IP')
18 curr_wlan0 = os.environ.get('WLAN0_IP')
19
20 #get most recent ETH0 IP
21 with open('eth0_temp.txt') as file:
22     new_eth0 = file.readline()
23
24 file.close()
25 subprocess.run("rm eth0_temp.txt", shell=True)
26
27 #get most recent WLAN0 IP
28 with open('wlan0_temp.txt') as file:
29     new_wlan0 = file.readline()
30
31 file.close()
32 subprocess.run("rm wlan0_temp.txt", shell=True)
33
34 #check if eth0 IP has changed and if necessary update env record and send telegram alert
35 if new_eth0 != curr_eth0:
36     set_key(env_path, 'ETH0_IP', new_eth0)
37     message = "Latest eth0 IP config for Redback Pi" + str(DEVICE_ID) + ": " + new_eth0
38     bot.send_message(-1001950776693, message)
39
40 #check if wlan0 IP has changed and if necessary update env record and send telegram alert
41 if new_wlan0 != curr_wlan0:
42     set_key(env_path, 'WLAN0_IP', new_wlan0)
43     message = "Latest wlan0 IP config for Redback Pi" + str(DEVICE_ID) + ": " + new_wlan0
44     bot.send_message(-1001950776693, message)
```


4.10. Deliverable Fitness Tracker

Fitness Tracker Project:

The aim of this project is to implement a foundational tool to collect data from users and their workouts to be able to provide them with a regular fitness routines and workouts to make their exercise performance more effective.

In this project the data set which has been used, called "Master_Data" which is accessible through BigQuery.

This project has different components as below:

- ❖ Established a Data Dictionary
- ❖ Conducted some research over the Company's Data Pipeline
- ❖ Implemented a BMI Calculator
- ❖ Created Feedback Analysis Model
- ❖ Designed Fitness Summary Dashboards

1- Data Dictionary

Established a data dictionary and placed in GitHub to allow others to understand the variable names and values of the data collected from Wahoo sensors

Fields	
Distance	<i>show distance in either miles/feet (statute units) or kilometers/meters (metric units)</i>
Enhanced_altitude/Altitude	<i>To determine the rider's elevation which means the total number of feet the rider climb in a ride)</i> <i>This information is recorded during the activity and is used to report elevation related information in Garmin Connect.</i>
Ascent	<i>A climb or walk to the summit of a mountain or hill/an instance of rising or moving up through the air.</i>
Total Ascent	<i>A total of elevation gain which means the total number of feet the rider climb in a ride</i>
Average Ascent	<i>Provides an average of all ascents recorded during an activity</i>
Maximum Elevation	<i>Provides the highest elevation achieved.</i>
Grade	<i>Data field for Garmin devices that calculates the slope (or grade) of the hill users are walking on</i>
Calories	<i>This is the total of active and resting calories that are calculated during a recorded activity on your device (from</i>

	<i>the moment that you start the timer for the activity to the moment you stop the timer)</i>
Speed/Distance Algorithm:	<i>This is the most basic method of determining calories. It is represented in calories/Kcal.</i>
Speed/Enhanced Speed:	<i>It is distance by total time sent on an activity. It is calculated in m/sec or m/h. If the values of speed are too large to be fit in speed, then enhanced speed is used.</i>
Heart_rate:	<i>heart rate values can be set as absolute or relative values. Absolute values represent beats per minute (bpm) for heart rate, or watts for power.</i>
Temperature:	<i>The Temperature widget will display the ambient air temperature near the barometric altimeter port. This reading can be affected by body heat. It is represented in Fahrenheit.</i>
Cadence	<i>The cadence fields in a FIT file represent RPMs. For cycling 1 RPM equals one full rotation of the cranks Cadence in cycling is the number of revolutions your pedals make per minute as you ride. revolution per minute(RPM)</i>
Power	<i>Power values can be set as absolute or relative values. Absolute values represent watts for power.</i>
Left_right_balance	<i>It shows as a percentage the power separately put out by the left and right leg.</i>
Session_ID	<i>It is unique id generated for each session performed by user</i>
User_ID	<i>It is the unique ID generated for every user</i>
Weight	<i>It gives the weight of the person in kgs</i>
Gender	<i>It Shows gender of registered person of the device</i>
FTP	<i>Functional Threshold Power (FTP) is a measurement from power meters. It is the highest power level you can maintain for one hour without growing fatigued. FTP is beneficial because it provides an outlook on performance ability.</i>
Age	<i>Available on select Garmin watches, Fitness Age is an estimate of how fit you are compared to your actual age. Compatible Garmin watches will measure your Fitness Age differently, depending on which device you have. Fitness age is an estimate of how fit you are compared to your actual age</i>
Product_Name	<i>It describes the product used for recording the activity.</i>

Link to the Wahoo Introduction:

<https://cloud-api.wahooligan.com/#introduction>

[Link to GitHub](#)

[Link to Ms Team](#)

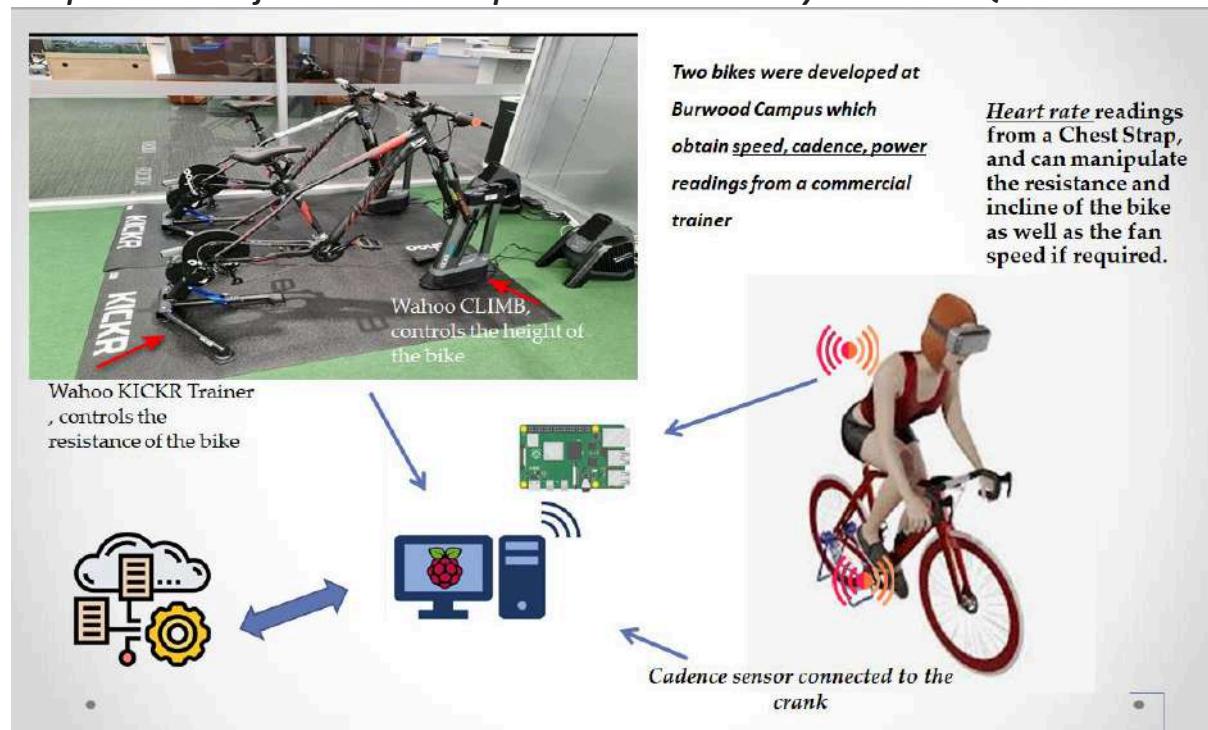
2- Data Pipeline

The Data Origin and Data Pipeline in Redback Company:

At Redback company we utilize “Contact Management System” (CMS). The data is received from bike’s sensors (Wahoo) via Bluetooth using a Raspberry Pi and submitted via MQTT to the CMS, GCP, unity and mobile app.

In coding side, what is done is mostly coding in Python to get data from riders and sending it off to the mobile app to get used.

The protocol used for this data transport mechanism used by drivers is MQTT.



For more detail see the Sensor Architecture report in GitHub, under Tri 3 2023 folder

[Link to Ms Team](#)

[Link to GitHub](#)

3- BMI Calculator

A BMI Calculator has been implemented using python, as we don’t have the height field in our current database to be able to calculate the BMI, so on the model that I’ve

implemented it would ask the user to input their height and weight and then it would calculate the BMI.

5. Summary of BMI and the benefits and limitations:

BMI(Body Mass Index) is one of the most common ways of grouping people by their weight.

5.1. A History of BMI:

BMI was first developed in the mid-1800s by a polymath called Adolf Quetelet. The purpose was to measure people's weight in relation to their height, by dividing their weight in KG by their height in meters squared. Scientists have created BMI categories to group people according to how far their weigh is from the norm for their height.

5.2. The categories of BMI:

- *BMI less than 16, classified as "Severely Underweight",*
- *BMI 16 to <18.5, classified as "Underweight" range.*
- *BMI 18.5 to <25, classified as "Healthy weight" range.*
- *BMI 25.0 to <30, classified as "Overweight" range.*
- *BMI 30.0 or higher, falls within the "Obesity" range.*

5.3. Benefits:

- *Easy to measure,*
- *No expensive equipment needed,*
- *It is quick and cheap to work out,*

5.4. Limitations:

- *BMI is not a way of diagnosing body fat,*
- *Categorizing people based on their BMI could lead to weight stigma,*
- *BMI is not an appropriate measure for children*

The following code will receive "Hight" and "Weight" from user to calculate and round the BMI to 2 decimal places and shows the user's BMI category using a function called "BMI". Also this code will check the entered values for "Hight" and "Weight" and if it's not valid, it will respond "Incorrect Data"

```
In [77]: h = float(input('Enter height in Meter: '))
if (h >= 1.0 and h <= 2.5):
    height = h
else:
    raise Exception("Incorrect Data, Please enter your height in Meter ")

w = float(input('Enter weight in KG: '))
if (w >= 10 and w <= 1000):
    weight = w
else:
    print("Incorrect Data, Please enter your weight in KG ")

def BMI(height, weight):  #BMI Function to calculate BMI
    bmi = round(weight/(height**2), 2)

    if (bmi < 16):
        return 'Severely Underweight', bmi

    elif (bmi >= 16 and bmi < 18.5):
        return 'Underweight', bmi

    elif (bmi >= 18.5 and bmi < 25):
        return 'Healthy', bmi

    elif (bmi >= 25 and bmi < 30):
        return 'Overweight', bmi

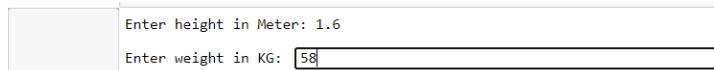
    elif (bmi >= 30):
        return 'Obese', bmi

calculation, bmi = BMI(height, weight)
print('Your BMI is: {} which is {}, the normal BMI is between 18.5 to 25'.format(bmi, calculation))

Enter height in Meter: 1.70
Enter weight in KG: 60

Your BMI is: 20.76 which is Healthy, the normal BMI is between 18.5 to 25
```

After you run the code, it will ask for height and weight



```
Enter height in Meter: 1.6
Enter weight in KG: 58
```

And then it will calculate and return the BMI,

```
Your BMI is: 22.66 which is Healthy, the normal BMI is between 18.5 to 25
```

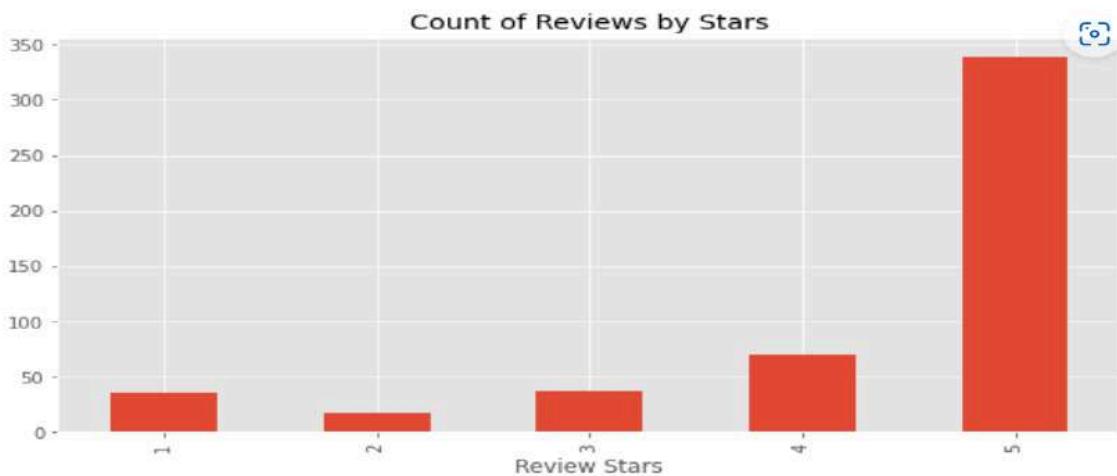
[Link to GitHub](#)

4- Feedback Analysis

This project aims to train and prepare a “feedback analysis model” to be used at the time that we would have the “feedback” feature for any of our products.

- *The Vader Model is developed and tested with different random datasets*
- *Ran NLTK work tokenizer to splits the feedbacks*
- *The required Tokens are generated*
- *Sentiment scores are generated and added to the current dataset*

The Python has been used to create the Model:



- **the plot we can seFrome that most of the reviews are 5 stars**

```
In [83]: # Test 2
sia.polarity_scores('This is the worst thing ever.')
Out[83]: {'neg': 0.451, 'neu': 0.549, 'pos': 0.0, 'compound': -0.6249}

In [4]: sia.polarity_scores('This is the best thing ever.')
Out[4]: {'neg': 0.0, 'neu': 0.543, 'pos': 0.457, 'compound': 0.6369}
```

Below indicates the project progress steps and the whole Python code:

- *The required libraries imported:*

```

In [ ]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

plt.style.use('ggplot')

import nltk
nltk.download()

showing info https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/index.xml

```

```

In [ ]: import nltk
nltk.download('maxent_ne_chunker')

In [ ]: import nltk
nltk.download('words')

In [28]: import nltk
nltk.download('vader_lexicon')

[nltk_data] Downloading package vader_lexicon to
[nltk_data]     C:\Users\ella\AppData\Roaming\nltk_data...

```

```

Out[28]: True

```

- *The data set imported*

```

In [17]: # Read in data
feedback = pd.read_csv('Reviews.csv')
feedback.head()

```

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Good Quality Dog Food	I have bought several of the Vitality canned d...
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Not as Advertised	Product arrived labeled as Jumbo Salted Peanut...
2	3	B000LQOCHO	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1219017600	"Delight" says it all	This is a confection that has been around a fe...
3	4	B000UA0QIQ	A395BORC6FGVXV	Karl	3	3	2	1307923200	Cough Medicine	If you are looking for the secret ingredient i...
4	5	B006K2ZZ7K	A1UQRSCLF8GW1T	Michael D. Bigham "M. Wassir"	0	0	5	1350777600	Great taffy	Great taffy at a great price. There was a wid...

```

In [ ]: feedback.shape

```

```

In [ ]: feedback.shape

```

```

In [18]: feedback = feedback.head(500)
feedback.head(5)

```

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Good Quality Dog Food	I have bought several of the Vitality canned d...
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Not as Advertised	Product arrived labeled as Jumbo Salted Peanut...
2	3	B000LQOCHO	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1219017600	"Delight" says it all	This is a confection that has been around a fe...
3	4	B000UA0QIQ	A395BORC6FGVXV	Karl	3	3	2	1307923200	Cough Medicine	If you are looking for the secret ingredient i...
4	5	B006K2ZZ7K	A1UQRSCLF8GW1T	Michael D. Bigham "M. Wassir"	0	0	5	1350777600	Great taffy	Great taffy at a great price. There was a wid...

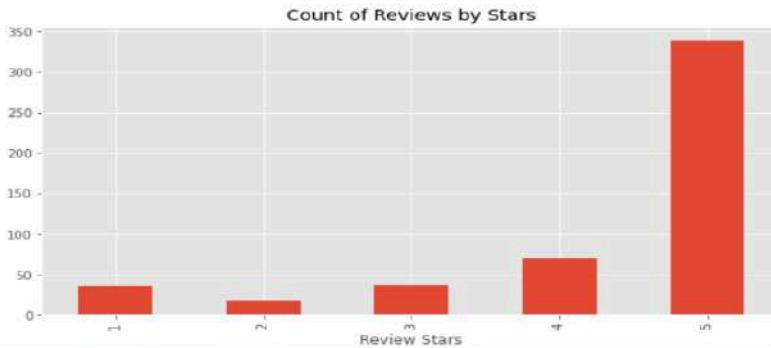
- *Ran a Quick EDA to get an idea of what the data set looks like*

Value count on Score column to see the number of times each score occurs

```
feedback['Score'].value_counts().sort_index()

Out[19]: 1     36
2     18
3     37
4     70
5    339
Name: Score, dtype: int64

In [20]: ax = feedback['Score'].value_counts().sort_index() \
          .plot(kind='bar',
                title='Count of Reviews by Stars',
                figsize=(10, 5))
ax.set_xlabel('Review Stars')
plt.show()
```



From the plot we can see that most of the reviews are 5 stars

- **Running NLTK work tokenizer to splits the sentence into the parts of each word in the sentence**

```
ex = feedback['Text'][50]
print(ex)

This oatmeal is not good. Its mushy, soft, I don't like it. Quaker Oats is the way to go.
```

the result came back negative

```
tokens = nltk.word_tokenize(ex)
tokens

['This',
 'oatmeal',
 'is',
 'not',
 'good',
 '.',
 'Its',
 'mushy',
 ',',
 'soft',
 ',',
 'I',
 'do',
 "n't",
 'like',
 'it',
 '..',
 'Quaker',
 'Oats',
 'is',
 'the',
 'way',
 'to',
 'go',
 '..']
```

- ***Running the nltk pos tag for part of speech tagging***

```
speech_tagged = nltk.pos_tag(tokens)
speech_tagged[:10]

[('This', 'DT'),
 ('oatmeal', 'NN'),
 ('is', 'VBZ'),
 ('not', 'RB'),
 ('good', 'JJ'),
 ('.', '.'),
 ('Its', 'PRP$'),
 ('mushy', 'NN'),
 (',', ','),  
 ('soft', 'JJ')]
```

- ***Grouping the Tokens into chunks of text***

```
entities = nltk.chunk.ne_chunk(speech_tagged)
entities pprint()

(S
  This/DT
  oatmeal/NN
  is/VBZ
  not/RB
  good/JJ
  ./.
  Its/PRP$ 
  mushy/NN
  ,/,
  soft/JJ
  ,/,
  I/PRP
  do/VBP
  n't/RB
  like/VB
  it/PRP
  ./.
  (ORGANIZATION Quaker/NNP Oats/NNPS)
  is/VBZ
  the/DT
  way/NN
  to/T0
  go/VB
  ./.)
```

Implementing the Vader Model:

6. Vader Sentiment Scoring

VADER(Valence Aware Dictionary for Sentiment Reasoning) is an NLTK module that provides sentiment scores based on the words used. It is a rule-based sentiment analyzer in which the terms are generally labeled as per their semantic orientation as either positive or negative.

VADER has the advantage of assessing the sentiment of any given text without the need for previous training as we might have to for Machine Learning models. The result generated by VADER is a dictionary of 4 keys neg, neu, pos and compound: neg, neu, and pos meaning negative, neutral, and positive respectively.

This module uses a “bag of words” approach:

- Stop words are removed
- Each word scored and combined to a total score
- This model is not account for relationship between words

```

from nltk.sentiment import SentimentIntensityAnalyzer
from tqdm.notebook import tqdm

#sia is an created object of "Sentiment Intensity Analyzer"
sia = SentimentIntensityAnalyzer()

```

Testing our object

sia.polarity_scores('I am so happy!')

The result is: negative as zero, neutral point = 0.318 and positive point= 0.682, So this sentence is mostly positive

```

# Test 2
sia.polarity_scores('This is the worst thing ever.')
{'neg': 0.451, 'neu': 0.549, 'pos': 0.0, 'compound': -0.6249}

sia.polarity_scores('This is the best thing ever.')
{'neg': 0.0, 'neu': 0.543, 'pos': 0.457, 'compound': 0.6369}

```

The result is: negative point = 0.451, neutral point = 0.549 and positive point= 0 and compound: -0.6249 So this sentence is mostly Negative

```

# Running the sia on our tokens
sia.polarity_scores(ex)
{'neg': 0.22, 'neu': 0.78, 'pos': 0.0, 'compound': -0.5448}

```

The result is: negative point = 0.22, neutral point = 0.78 and positive point= 0 and compound: -0.5448 So this sentence is mostly Negative

Run the polarity score on the entire dataset, the result is a dictionary which stores the result of the below loop which each raw would

```

result = {}
for i, row in tqdm(feedback.iterrows(), total=len(feedback)):
    text = row['Text']
    myid = row['Id']
    result[myid] = sia.polarity_scores(text)

0% | 0/500 [00:00<?, ?it/s]

```

Here shows the result dictionary with each id that contain the scores of neg, neu, pos and compound which stored in a Panda data frame to make it easier to work with and displayed horizontally

```
pd.DataFrame(result).T
```

	neg	neu	pos	compound
1	0.000	0.695	0.305	0.9441
2	0.138	0.862	0.000	-0.5664
3	0.091	0.754	0.155	0.8265
4	0.000	1.000	0.000	0.0000
5	0.000	0.552	0.448	0.9468
...
496	0.000	0.554	0.446	0.9725
497	0.059	0.799	0.142	0.7833
498	0.025	0.762	0.212	0.9848
499	0.041	0.904	0.055	0.1280
500	0.000	0.678	0.322	0.9811

500 rows × 4 columns

So now we have a data frame that has index which is the IDs and the four neg, neu, pos and compound fields, we call this result Vaders

```
#vaders.reset_index().rename(columns={'index': 'Id'})
vaders["Id"] = feedback["Id"]
#vaders.merge(feedback, how='Left')
#vaders.columns
```

```
vaders.columns
```

```
Index(['neg', 'neu', 'pos', 'compound', 'Id'], dtype='object')
```

```
vaders=vaders.merge(feedback, how='left')
```

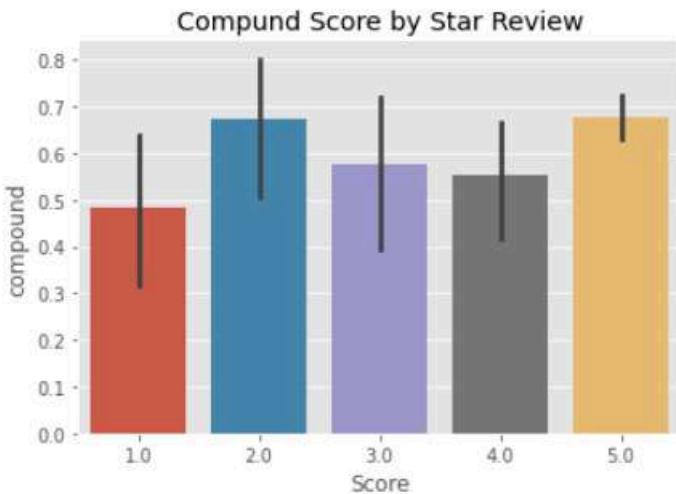
Now we have sentiment score added to the original fields

	neg	neu	pos	compound	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time
0	0.000	0.695	0.305	0.9441	2.0	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0.0	0.0	1.0	1.346976e+09
1	0.138	0.862	0.000	-0.5664	3.0	B000LQOCHO	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1.0	1.0	4.0	1.219018e+09
2	0.091	0.754	0.155	0.8265	4.0	B000UA0QIQ	A395BORC6FGVXV	Karl	3.0	3.0	2.0	1.307923e+09
3	0.000	1.000	0.000	0.0000	5.0	B006K2ZZ7K	A1UQRSCLF8GW1T	Michael D. Bigham "M. Wassil"	0.0	0.0	5.0	1.350778e+09
4	0.000	0.552	0.448	0.9468	6.0	B006K2ZZ7K	ADT0SRK1MGOEU	Twoapennything	0.0	0.0	4.0	1.342051e+09

7. Plot VADER results

Runing the Plot on vaders' data by assigning x the score valuse which is the star review of the the person and then compound is going to be our y value and that's the negative to positive.

```
ax = sns.barplot(data=vaders, x='Score', y='compound')
ax.set_title('Compund Score by Star Review')
plt.show()
```



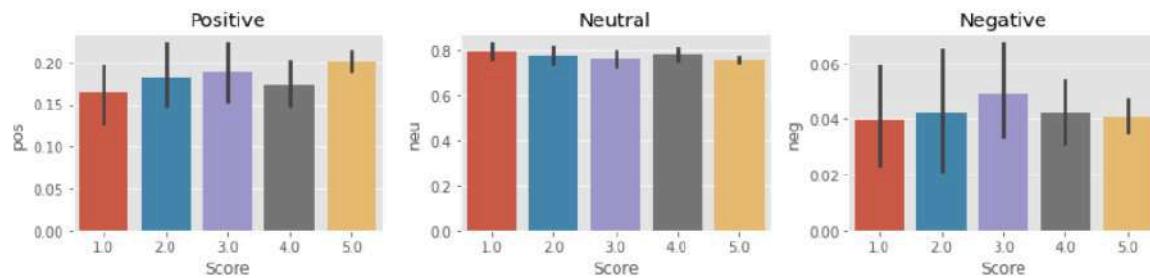
The plot shows that one star review has lower compound score and the five star view is higher

Running the Plot based on the positive neutral and negative scores

```

fig, axs = plt.subplots(1, 3, figsize=(12, 3))
sns.barplot(data=vaders, x='Score', y='pos', ax=axs[0])
sns.barplot(data=vaders, x='Score', y='neu', ax=axs[1])
sns.barplot(data=vaders, x='Score', y='neg', ax=axs[2])
axs[0].set_title('Positive')
axs[1].set_title('Neutral')
axs[2].set_title('Negative')
plt.tight_layout()
plt.show()

```



The plots confirm that our Vader Model is valuable in having the connection between the score of the text and sentiment score and it does relate to the actual rating review of the reviewers

[Link to GitHub:](#)

5- Fitness Summary Dashboards

The aims of this project is to create the summary reports, using SQL queries on the data which are collected using the Wahoo sensors

In this project some dashboards designed using Tableau to visualize the user's workouts reports

Summary Reports – All users

For the best measure of calories burned, the ELEMNT GPS devices series (including the ELEMNT, BOLT, ROAM, and RIVAL) and Wahoo Fitness app default to using calculations based on power output when a trainer or power meter are connected (if no direct power source is connected, heart rate will be used to calculate calories as described further below). Calories from power are calculated based on the following formula:

- **Calorie rate = (4.184kJ) * 0.239GMR**

In the calculation above, Kilo Joules (kJ) are measured from the trainer or power meter and multiplied by the Gross Metabolic Rate (GMR, an average of gross metabolic efficiency), estimated at 23.9% since 1 Calorie is equal to 4.184kJ of work and the human body is about 24% efficient at burning calories.

The MAX calories – By Month

```

SELECT DISTINCT EXTRACT(MONTH from date_AEST) AS month, userId, max(calories) as max_calories
FROM `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data`
WHERE calories IN

(SELECT max(calories) as max_calories
FROM `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data`
WHERE date_AEST BETWEEN '2021-01-01' AND '2022-01-01'
GROUP BY EXTRACT(MONTH from date_AEST)
)

GROUP BY userID, month
ORDER BY month

```

The MAX Distance – By Month

```

SELECT f.date_AEST, f.userId, max(f.distance) as max_distance
FROM `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data` 
inner join

(SELECT s.userID, max(s.distance) as max_distance
FROM `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data` 
WHERE date_AEST BETWEEN '2021-01-01' AND '2022-01-01'
GROUP BY EXTRACT(MONTH from s.date_AEST), s.userID
)

ON f.userID = s.userID and
GROUP BY f.userID, date_AEST

```



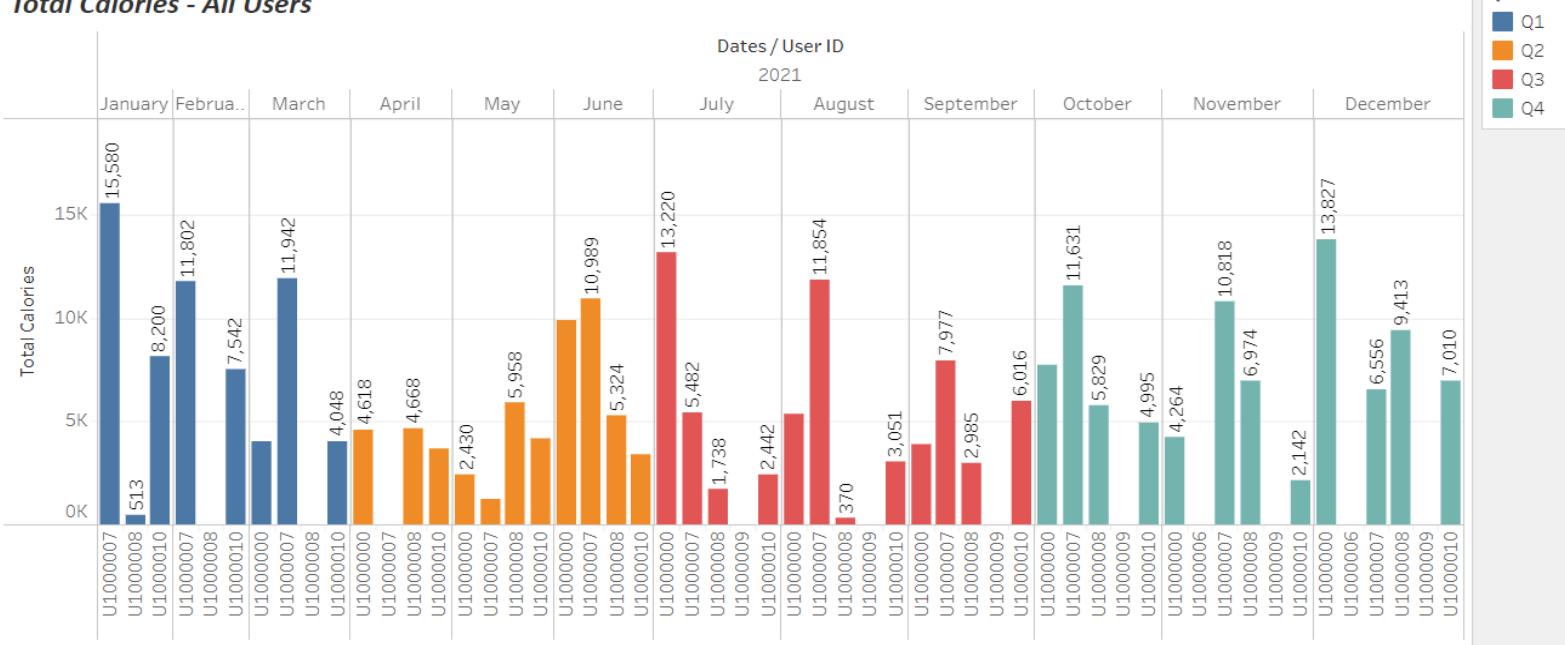
Total Calories and Distance for all users by each date – 2021

```

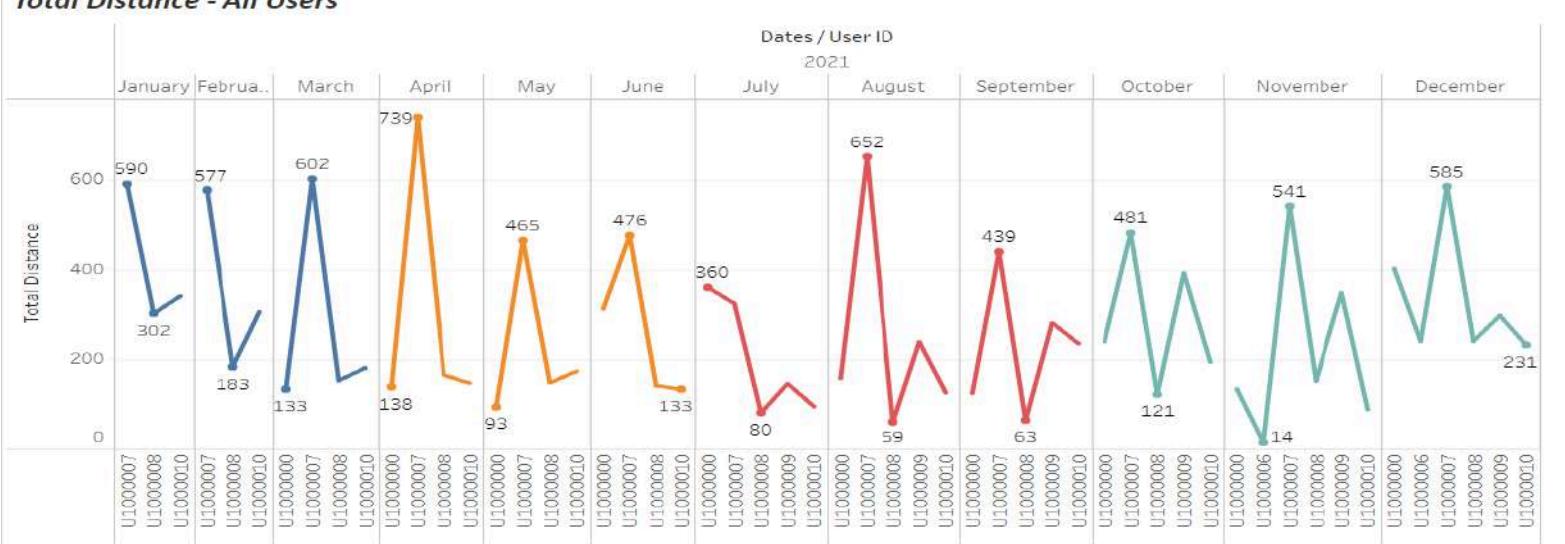
SELECT userID, date_AEST AS dates, Round(SUM(calories)/4.18) as total_calories,
       Round(SUM(distance)/1000)AS total_distance
FROM `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data`
WHERE (date_AEST BETWEEN '2021-01-01' AND 2021-12-31')
GROUP BY date_AEST, userID
ORDER BY date_AEST
    
```

Row	userID	dates	total_calories	total_distance
1	U1000007	2021-01-01	3293.0	66.0
2	U1000007	2021-01-02	3791.0	70.0
3	U1000010	2021-01-02	101.0	5.0
4	U1000007	2021-01-03	122.0	8.0
5	U1000007	2021-01-05	749.0	16.0
6	U1000010	2021-01-07	1308.0	54.0
7	U1000010	2021-01-08	77.0	4.0
8	U1000007	2021-01-08	1205.0	28.0
9	U1000008	2021-01-08	null	3.0
10	U1000010	2021-01-09	568.0	22.0
11	U1000007	2021-01-09	null	106.0
12	U1000008	2021-01-09	233.0	31.0
13	U1000008	2021-01-10	null	11.0
14	U1000007	2021-01-10	1291.0	49.0

Total Calories - All Users



Total Distance - All Users



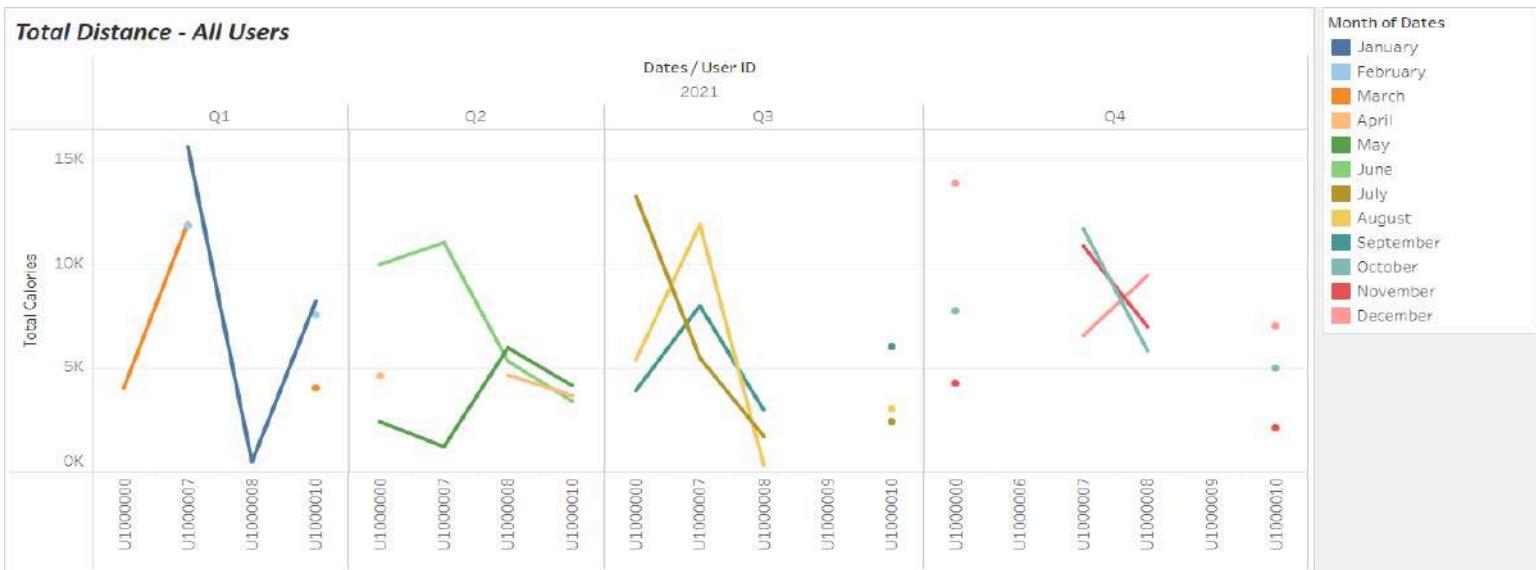


Tableau Dashboard - Total Report by Month (All Users)



- Average, MAX and MIN distance and calories for all riders – By Month

SELECT

```
EXTRACT(MONTH from date_AEST) AS month,  
round(AVG(calories)) as avg_calories,  
Round(AVG(distance))AS agv_distance,  
round(MAX(calories)) as max_calories,  
Round(MAX(distance))AS max_distance,  
round(MIN(calories)) as min_calories,  
Round(MIN(distance))AS min_distance,  
FROM `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data`
```

WHERE date_AEST BETWEEN '2021-01-01' AND '2021-12-30'

GROUP BY month

ORDER BY month;

**Monthly Calories and Distance Report - All Users
(Average, MAX and MIN)**

Month of d..	AVG Calories	AVG Distance	Max Calories	Max Distance	Min Calories	Min Distance
January	8,563	597	42,613	2,826	0	0
February	7,682	535	36,740	2,563	0	0
March	7,880	562	39,198	2,690	0	0
April	5,527	516	26,110	2,515	0	0
May	7,039	468	33,424	2,253	0	0
June	12,655	575	58,925	2,748	0	0
July	8,427	518	39,149	2,462	0	0
August	8,867	657	41,919	3,116	1	0
September	10,285	673	48,721	3,161	4	0
October	12,836	799	58,991	3,714	5	0
November	10,449	673	49,094	3,140	87	0
December	13,909	936	65,280	4,373	44	0

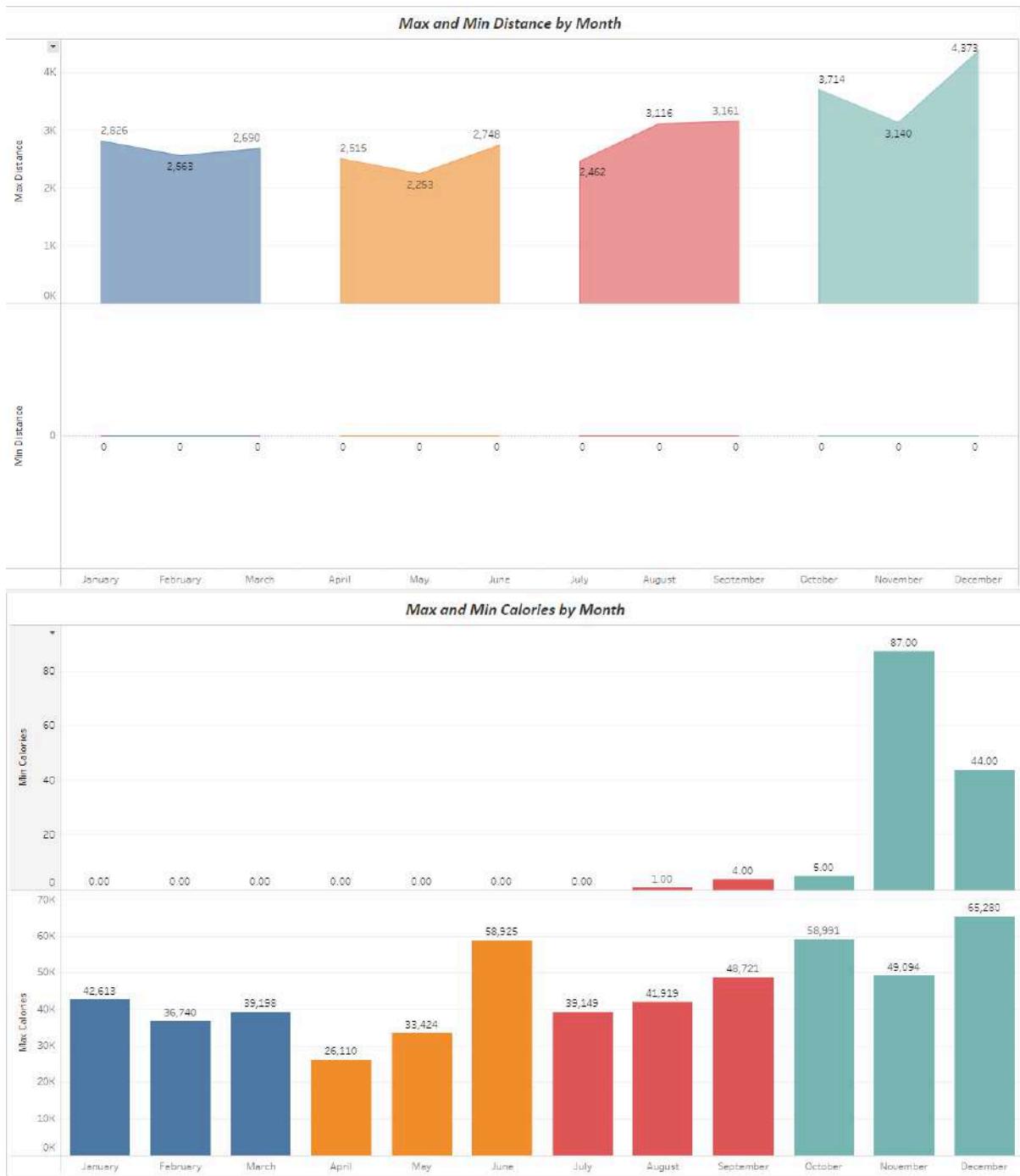


Tableau Dashboard - Monthly Calories and Distance Report
All Users (Average, MAX and MIN)

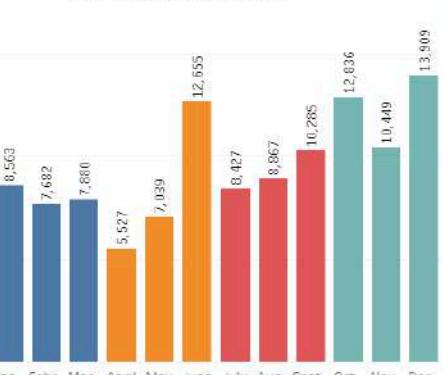
Monthly Calories and Distance Report - All Users (Average, MAX and MIN)

Month of date_AEST	Avg Calories	Avg Distance	Max Calories	Max Distance	Min Calories	Min Distance
January	8,563	597	42,613	2,826	0	0
February	7,602	535	36,740	2,563	0	0
March	7,800	562	39,198	2,690	0	0
April	5,527	516	26,110	2,515	0	0
May	7,039	460	33,424	2,253	0	0
June	12,655	575	58,925	2,749	0	0
July	8,427	510	39,149	2,462	0	0
August	8,067	657	41,919	3,116	1	0
September	10,205	673	40,721	3,161	4	0
October	12,036	799	58,991	3,714	5	0
November	10,449	673	49,094	3,140	87	0
December	13,909	936	65,280	4,373	44	0

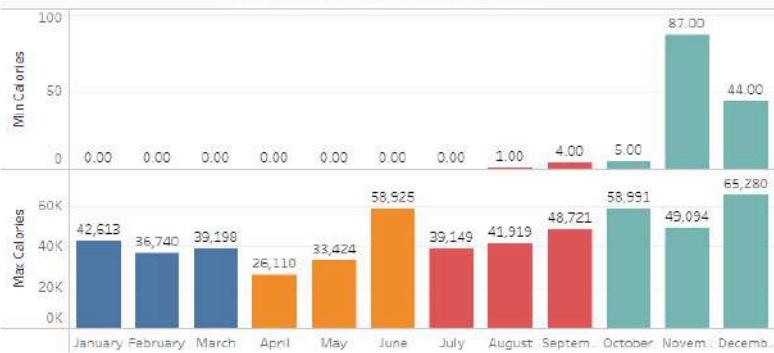
AVG Distance by Month - All Users



AVG Calories by Month



Max and Min Calories by Month



Max and Min Distance by Month



Summary Cycling Tracking 2021

Monthly Totals for each user:

The below query would calculate the Monthly total distance in KM, total calories and the total hours, for each users in 2021,

Calculation For User U1000000

```

SELECT EXTRACT(MONTH from date_AEST) AS month,
       ROUND(SUM(calories)/4.18) AS total_calories,
       Round(SUM(distance)/1000) AS total_distance,
       ROUND(count(date_AEST)/4663) AS total_hour
  FROM `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data`
 WHERE userID = "U1000000"
   and date_AEST BETWEEN 2021-12-31' AND '2022-01-01'
   and date_AEST
 GROUP BY month
 ORDER BY month;
    
```

month	total_calories	total_distance	total_hour
3	4050.0	136.0	13.0
4	4619.0	140.0	15.0
5	2429.0	92.0	14.0
6	9950.0	313.0	23.0
7	13220.0	362.0	23.0
8	5389.0	157.0	18.0
9	3920.0	124.0	15.0
10	7730.0	241.0	16.0
11	4264.0	133.0	14.0
12	13828.0	402.0	26.0

Total Calories - By Day

Day of Dates	Dates										
	March	April	May	June	July	August	Septem..	October	Novem..	Decemb..	
1		755	179	55	1,195	122	284			214	
2			33	461	293	540	268		651		
3		269		84	417			1,098			
4		114		159			276			69	
5				294			288		67		
6		292		219	313			113			
7			795		29		308		270		
8					29	633	93	2,172			
9					530	161			288		
10	818				461		1,095		434		
11				130	2,488					631	
12	28	146			234		172	99			
13				1,962		553	114	186		140	
14	910		1	170		154	200				
15	10			327	5,461	182	27				
16	351	118	297			190				678	
17		1,407		788						1,976	
18	107					184	73				
19	490		28	984		165		359			
20	9		48	2,308					358		
21			351		950	554			27	593	
22		2	19	299		773	60	320	159	581	
23					215		300		284		
24		1,335					179	1,126	888	410	
25								179	563		
26	833	16			570	125		72		820	
27		27	320	830	35	342	184			3,106	
28		26				571		481	131	1,769	
29		46	202	355			67			1,694	
30	14	65		524			74		144	1,146	
31	481		157					1,525			

Calculation For User U1000007

```

SELECT EXTRACT(MONTH from date_AEST) AS month,
       ROUND(SUM(calories)/4.18)   AS total_calories,
       Round(SUM(distance)/1000)   AS total_distance,
       ROUND(count(date_AEST)/4663) AS total_hour
FROM `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data`
WHERE userID = "U1000007"
      and date_AEST BETWEEN 2021-12-31' AND '2022-01-01'
  
```

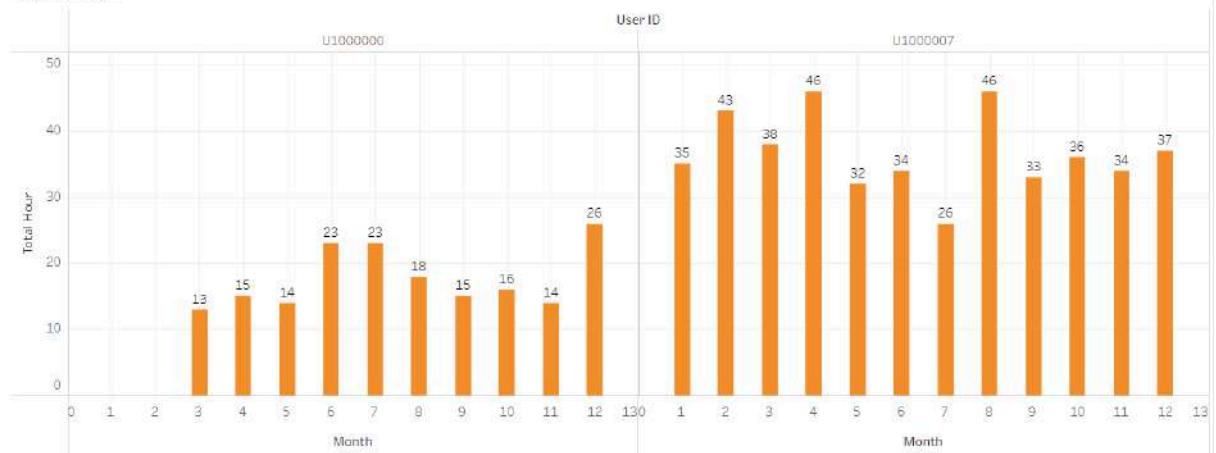
and date_AEST
 GROUP BY month
 ORDER BY month;

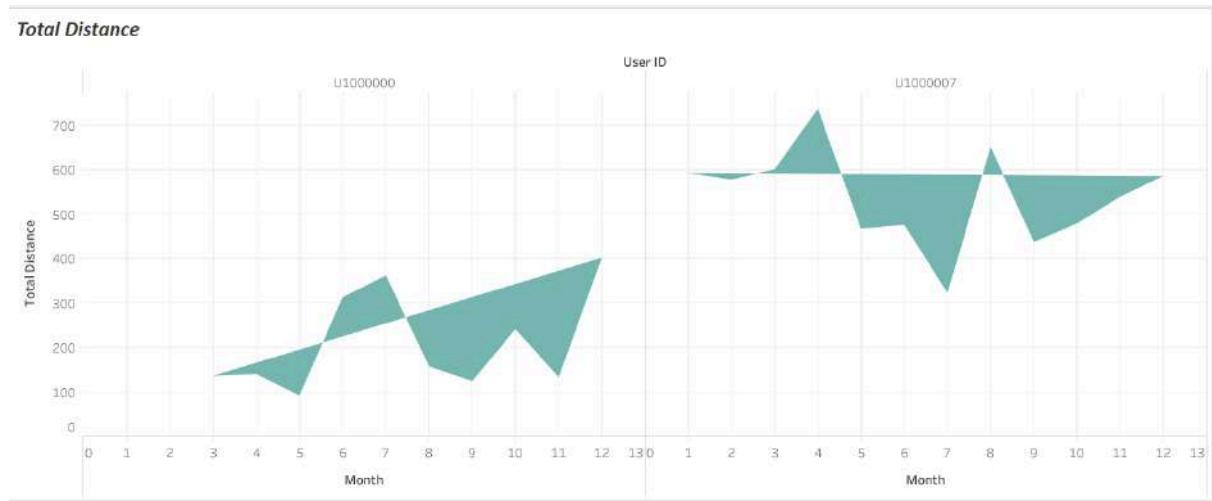
month	total_calories	total_distance	total_hour
1	15579.0	593.0	35.0
2	11801.0	578.0	43.0
3	11943.0	601.0	38.0
4	null	737.0	46.0
5	1237.0	467.0	32.0
6	10990.0	476.0	34.0
7	5482.0	323.0	26.0
8	11855.0	651.0	46.0
9	7975.0	437.0	33.0
10	11632.0	479.0	36.0
11	10817.0	539.0	34.0
12	6556.0	586.0	37.0

Total Calories



Total Hours





Daily Total Report for each user:

The below query would calculate the total distance in KM and total calories for each day of the year 2021:

For user U1000000

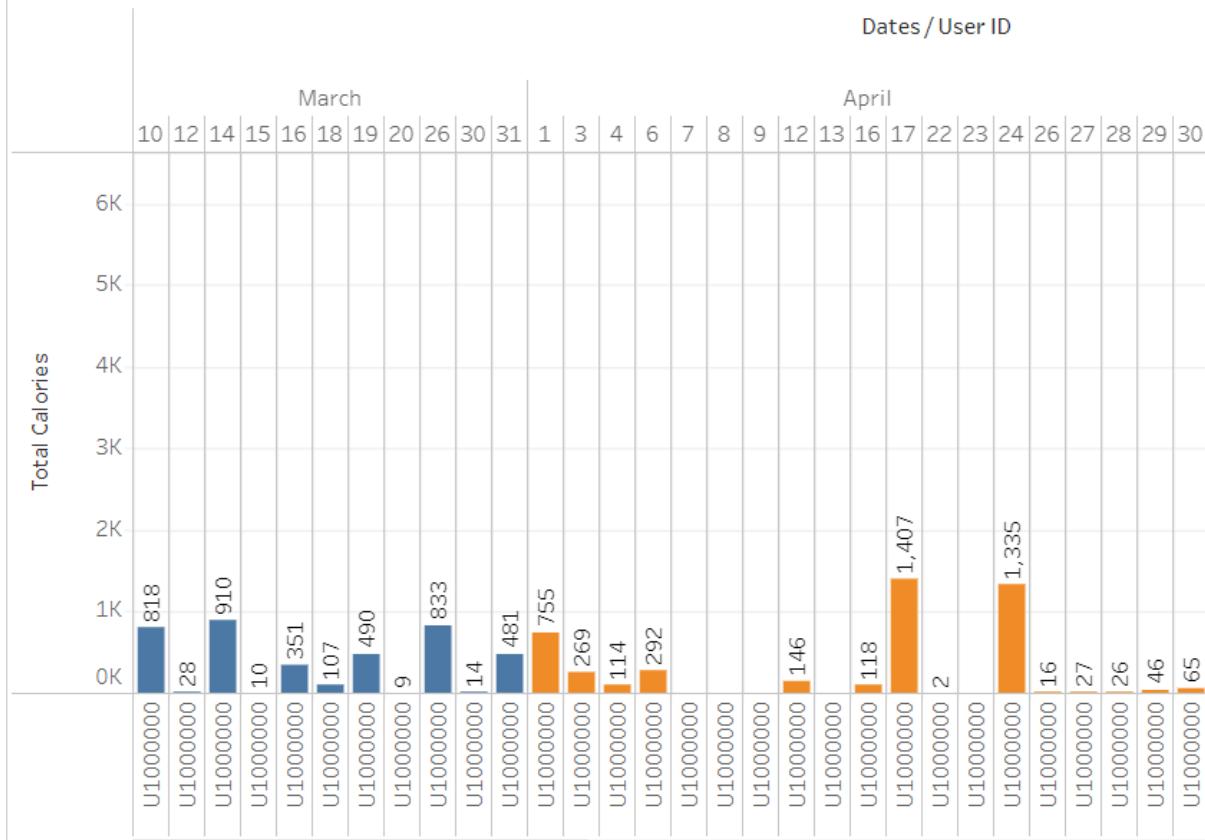
```
SELECT userID,
       date_AEST AS dates,
       ROUND(SUM(calories/4.18)) AS total_calories,
       Round(SUM(distance/1000))   AS total_distance
FROM `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data`
WHERE (date_AEST BETWEEN '2021-01-01' AND 2021-12-31)
      AND userID = 'U1000000'
GROUP BY date_AEST, userID
ORDER BY date_AEST
```

Row	userID	dates	total_calories	total_distance
1	U1000000	2021-03-10	818.0	26.0
2	U1000000	2021-03-12	28.0	1.0
3	U1000000	2021-03-14	910.0	31.0
4	U1000000	2021-03-15	10.0	0.0
5	U1000000	2021-03-16	351.0	10.0
6	U1000000	2021-03-18	107.0	3.0
7	U1000000	2021-03-19	490.0	19.0
8	U1000000	2021-03-20	9.0	0.0
9	U1000000	2021-03-26	833.0	26.0
10	U1000000	2021-03-30	14.0	0.0
11	U1000000	2021-03-31	481.0	17.0
12	U1000000	2021-04-01	755.0	25.0
13	U1000000	2021-04-03	269.0	7.0
14	U1000000	2021-04-04	114.0	3.0
15	U1000000	2021-04-06	292.0	9.0

Total Calories - By Day

Day of Dates	Dates											
	March	April	May	June	July	August	Septem..	October	Novem..	Decemb..		
1		755	179	55	1,195	122	284				214	
2			33	461	293	540	268				651	
3		269		84	417						1,098	
4		114		159							276	69
5				294							288	67
6		292		219	313						113	
7			795		29		308				270	
8					29	633	93	2,172				
9					530	161					288	
10	818				461		1,095				434	
11				130	2,488							631
12	28	146			234		172	99				
13				1,962		553	114	186			140	
14	910		1	170		154	200					
15	10			327	5,461	182	27					
16	351	118	297			190					678	
17		1,407		788							1,976	
18	107					184	73					
19	490		28	984		165		359				
20	9		48	2,308							358	
21			351		950	554					27	593
22		2	19	299		773	60	320	159		159	581
23					215		300				284	
24		1,335					179	1,126	888		410	
25							179	179	563			
26	833	16			570	125		72			820	
27		27	320	830	35	342	184				3,106	
28		26				571		481	131		1,769	
29		46	202	355							1,694	
30	14	65		524		67			144		1,146	
31	481		157			74		1,525				

Total Calories By Day



Created the following Summary Tracking Dashboards using Tablue:

Monthly Total – for User1 and User7

Summary Tracking 2021 - User U000000

Month of Dates	Total Calories	Total Distance
March	4,051	133
April	4,618	138
May	2,430	93
June	9,949	312
July	13,220	360
August	5,390	157
September	3,921	124
October	7,730	239
November	4,264	133
December	13,827	402

Total Calories by Month



Total Hours by Month



Total Distance by Month



Tracking Summary Report 2021 - User U000007

Month of D..	Total C..	Total Di..
January	15,580	590
February	11,802	577
March	11,942	602
April	739	
May	1,237	465
June	10,989	476
July	5,482	324
August	11,854	652
September	7,977	439
October	11,631	481
November	10,818	541
December	~10,000	~500

Total Calories by Month



Total Hours by Month



Total Distance by Month



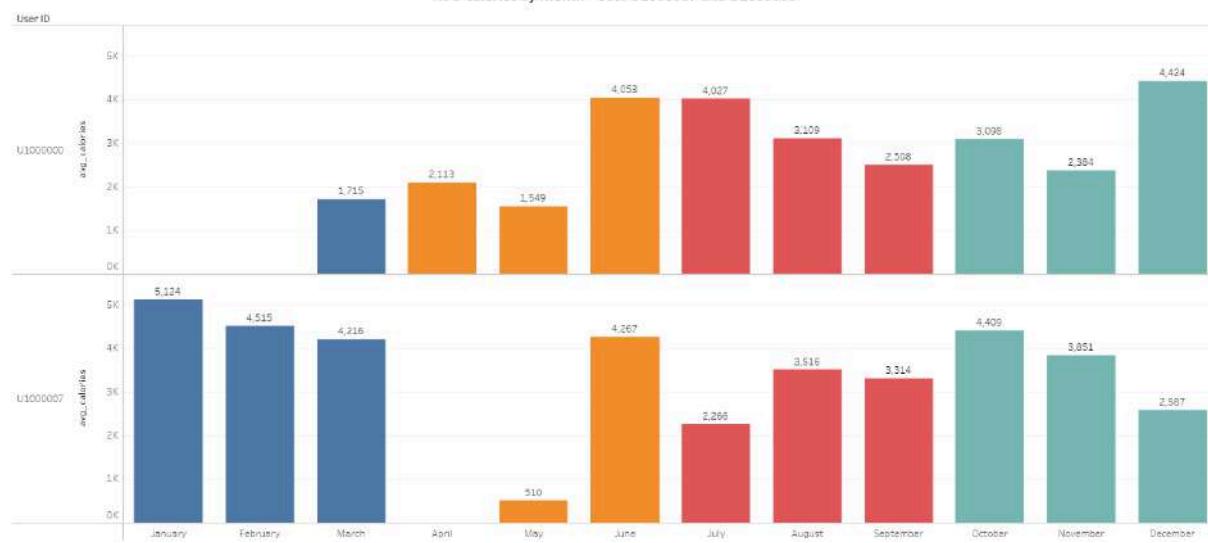
The below query would calculate the Monthly Average distance (KM) and calories for each user – in year 2021:

```
select userID, EXTRACT(MONTH from date_AEST) AS month, round(AVG(calories)) as avg_calories, Round(AVG(distance))AS agv_distance
FROM `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data`
```

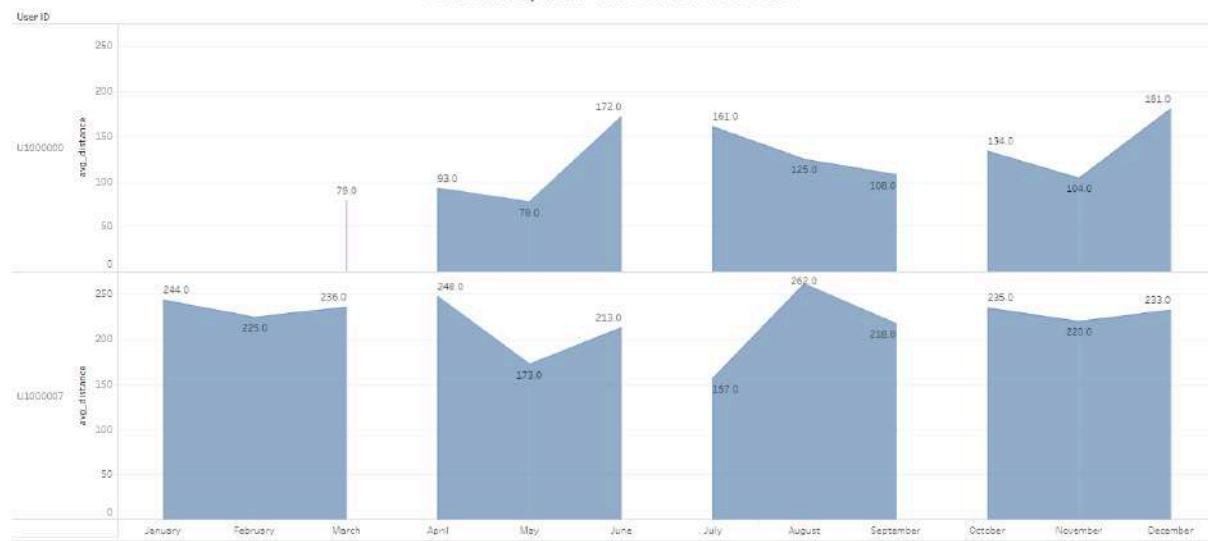
```
WHERE (date_AEST BETWEEN '2021-01-01' AND 2021-12-31) AND userID = 'U1000000'
GROUP BY month, userID
```

ORDER BY month;

Avg Calories by Month - User U1000007 and U1000000



Avg Distance by Month - User U1000007 and U1000000



**Tableau Dashboard – Monthly report
(Avg, Max, Min) – U000000**

Max, Min, and Avg Monthly Report 2021 - U000000

**Monthly Calories and Distance Report - UI000000
(Average, MAX and MIN)**

Month of date AEST	Avg Calories	Avg Distance	Max Calories	Max Distance	Min Calories	Min Distance
March	1,715	80	8,962	407	0	0
April	2,113	93	10,235	458	0	0
May	1,549	78	7,527	389	0	0
June	4,053	169	19,276	814	0	0
July	4,027	161	18,668	758	0	0
August	3,109	127	15,411	615	1	0
September	2,508	108	12,522	528	0	0
October	3,098	134	14,147	625	0	0
November	2,384	105	11,401	501	0	0
December	4,424	181	20,751	866	0	0

Max, Min and Avg Distance Report by Month (U000000)



Monthly Calorly Report (Max, Min and Avg) - U000007



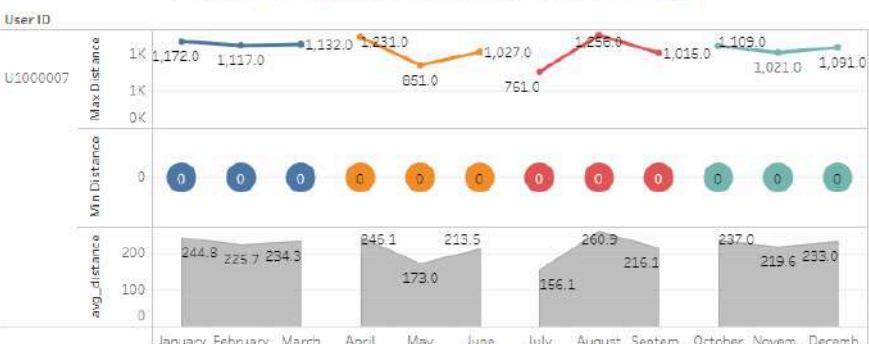
**Tableau Dashboard – Monthly report
(Avg, Max, Min) – U000007**

Max, Min, and Avg Monthly Report 2021 - U000007

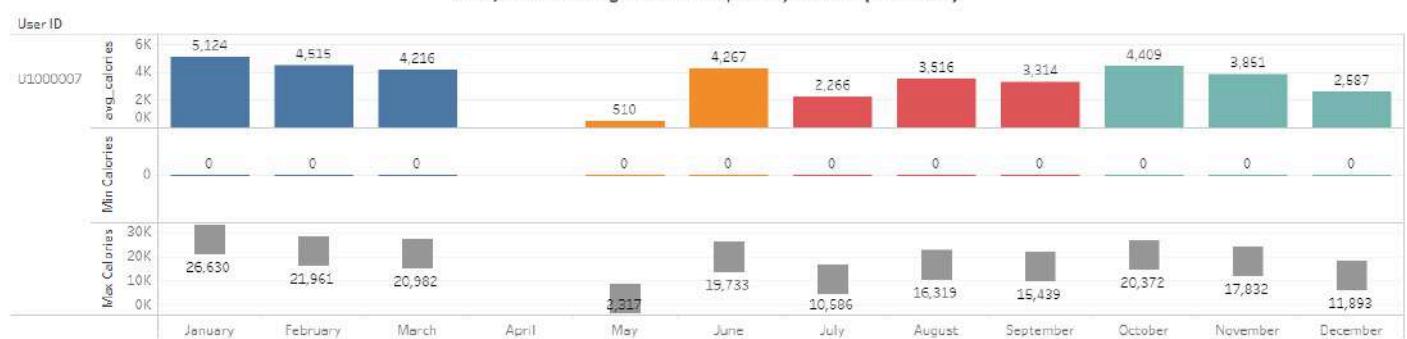
**Monthly Calories and Distance Report - UI000007
(Average, MAX and MIN)**

Month of date AEST	Avg Calories	Avg Distance	Max Calories	Max Distance	Min Calories	Min Distance
January	5,124	285	26,630	1,172	0	0
February	4,515	226	21,961	1,117	0	0
March	4,216	234	20,982	1,132	0	0
April	246	—	—	1,231	—	0
May	510	173	2,317	851	0	0
June	4,267	213	19,733	1,027	0	0
July	2,266	156	10,586	761	0	0
August	3,516	261	16,319	1,256	0	0
September	3,314	216	15,439	1,015	0	0
October	4,409	237	20,372	1,109	0	0
November	3,851	220	17,832	1,021	0	0
December	2,587	233	11,093	1,091	0	0

Max, Min and Avg Distance Report by Month (U000007)



Max, Min and Avg Calories Report by Month (U000007)



Daily Calorie Tracking 2021 - User U000000



Daily Calorie Tracking 2021 - User U000007



[Link to GitHub](#)

Team Member: Ella Zarandi

8. Roadmap

8.1. *GUI (Graphical User Interface) application that displays bikes current metric values/updates resistance/incline in real-time (AKA ‘free roam-mode’ for user):*

8.1.1. Description:

- *This would allow users/testers in the lab or remotely to be able to read every value (workout, speed, cadence, power, resistance, incline, FTP, etc) in real-time and not depend on the terminal that outputs MQTT callback messages, also enable buttons that will set resistance and incline manually*

8.1.2. Requirements:

- *Developed in python, incorporating the already implemented MQTT client, that subscribes to every topic in schema including workout*
- *During runtime, displays values of received topic values, updates value in GUI as its being received through MQTT message callbacks.*
- *Set resistance/incline manually from GUI.*
- *Manually enter the bike’s device id in the GUI, such that if we have multiple bikes we can subscribe to any specific bike at a time.*
- *No Terminal input/output needed during runtime; all visuals handled by GUI outputs.*

8.2. Create custom workouts via GUI (feature extension for GUI application):

8.2.1. Description:

- *Create an extension or a separate GUI window application that allows the user to set a custom workout, this could mean setting duration, intervals, unique resistances and inclines per interval etc This can allow the user to set specific workout regimes tailored to the users’ preferences.*

8.2.2. Requirements:

- *Developed in python, incorporating the already implemented MQTT client/dotenv for reading environment variables.*
- *Tkinter??*
- *User friendly interface*
- *Entries for user to set intervals, duration of intervals, and unique resistances/incline per interval (publish new MQTT value to bike per interval)*

Trello: <https://trello.com/c/qzyvvqN/117-create-custom-workouts-via-gui-feature-extension-for-gui-application>

8.3. Integrating Mobile App and IoT processes to start workouts via MQTT (Integration with mobile team):

8.3.1. Description:

- We want to begin integrating the Mobile Application and IoT processes, such that we can start workouts on the app via button presses, and have the pi handle the hardware manipulation and conveyancing of data between Kickr devices<->pi<->MQTT Broker<->Mobile App

8.3.2. System Requirements:

- Mobile app to be able to publish values to MQTT broker at bike/00000x/workout when particular start workout button pressed on app, IoT system running a scripts that listens for published values and starts related workout based on value received from MQTT topic

8.3.3. Sub System Requirements:

8.3.4. Overall:

- Each system to adhere to MQTT schema structure
- Required values published will be formalised to specific values and shared/documentated across teams via PDF schema {FTP, Ramped, Threshold, etc}
- Every value received from workout topic to be validated for correct type.
- Once a value received, other workouts cannot be started until the current one ends (terminate listener script processes once workout begins??)

8.3.5. Idea for Mobile App to:

- Create a MQTT client with the required Redback Hostname, Username and PW (same host as the IoT pi for communicating topic values)
- Publish a value to the workout topic in redbacks MQTT broker when some arbitrary button' is pressed
- Topic to subscribe/publish to = bike/000001/workout

Trello: <https://trello.com/c/cKs3wu3O/109-integrating-mobile-app-and-iot-processes-to-start-workouts-via-mqtt>

8.4. Throttle KICKR script from sending zero value data (Save GCP spending further)

8.4.1. Description:

We have previously and successfully throttled the publishing of zero value data from the fan and heartrate, this is important when we are connecting to the sensors yet are not riding the bike. We do not want to flood the CMS or GCP with redundant data as this can cost up to \$20 per day if left running.

8.4.2. Requirements / Recommendations

- Create a fix similar to fan/heartrate publishing fix (`incline_and_resistance_control.py` contains different MQTT implementations so research will need to be made)
- Only throttle when speed is 0 for a particular amount of time (the speed is always the last to drop to 0)
- Create a counter for bike speed == 0, if counter == 10, don't publish any bike data until the speed value changes)

Trello: <https://trello.com/c/vnwtDBI/97-throttle-kickr-script-from-sending-zero-value-data-save-gcp-spending-further>

8.5. Create Source Control/integration documentation/video for using a Branch style Workflow compared to a Fork/PR workflow for Team

8.5.1. Description

In current trimesters, the main Workflow for integrating changes to the codebase is via Forking and creating Pull Requests. This current style of DevOp makes it very difficult to checkout test branches on the RPI in the lab.

A more efficient workflow is to Create a new branch in the main IoT repo i.e “new_feature_12” or “specific_bug_fix_11”, etc. This means team members can push changes to these branches, then create a pull request to merge to main. From there, we can checkout this branch into the software/hardware and test.

git needed some config setting changed in order for git to accept commits to the branch repo

8.5.2. Requirements:

Create tutorial documentation with how to create and sustain this type of workflow, we want each member to be able to:

- Successfully clone main repo (upstream)
- fetch or create new branches designed for specific tasks
- add, stage, commit and push update to these specific branches
- Create pull requests that can be merged in the main branch on successful review

Known Issues (needs tutorial for):

- Workshop needed for demonstrating source control workflow
- User must have PAT (Personal Access Token) to use as git password whenever pushing (needs tutorial)
- specific config settings need changing:
 - run:
 - `git config --global credential.helper cache`

- `git config lfs.https://github.com/redbackoperations/iot.git/info/lfs.locksverify false`

Trello: <https://trello.com/c/xw4p6w4M/106-create-source-control-integration-documentation-for-using-a-branch-style-workflow-compared-to-a-fork-pr-workflow-for-team>

9. Open Issues

- *FTP workout needs to set initial resistance value to Kickr, currently it has a resistance of 0. (Set to 25? (Publish to resistance topic at start of FTP script))*
- *Any guest of Trello can move their task to ‘completed’ before being reviewed and approved. This muddies the workflow and timeline for leadership to understand current stages of tasks.*
- *The current workouts’ functionality to output resistance required to reach the threshold power is hard coded. The goal is to make the code so responsive that it prints out the resistance required based on calculations in the function. MQTT communication is used to read values from the bike. The program prints out the power used every second and lets the user know how much resistance is required to reach the maximum threshold power. The proof of concept file for the threshold power workout contains a working program that showcases what we need to do to deliver the final project. It contains a threshold power workout script that includes warm-up periods, cool-down period, rest period, and number of intervals*
- *These should be edited in future, updated for more modular workouts.*
- *Improve the workouts to able the users to modify the level of intensity of the workout starts at 1 but can be increased or decreased (between level 1 and 20) using the + or – buttons on the app.*

10. Lessons Learned

- *The main accessible resources are the Handover document and GitHub repo*
- *Identified the existing project’s issues and Suggested the possible solutions*
- *Scoped the projects and sprint planning to set timeframes and goals*
- *Delegated tasks appropriately and enforced realistic deadlines.*
- *Delegated tasks appropriately and enforced realistic deadlines.*
- *Made sure everyone were clear with their task and the expectations*
- *Provided everyone with the required tools and resources*
- *Fostered an environment that encouraged collaboration.*
- *Promoted clear and meaningful communication through team meetings or 1:1 catch up sessions to keep the communication lines open.*
- *Monitored and evaluated the project progress across the company, address potential issues and made sure the deadlines are met*
- *Conducted research and up-skilling when needed*
- *Tested out the implemented projects and created showcase videos*

11. Product Development Life Cycle

11.1. New Tasks

New tasks can be created, modified or dropped based on

- The existing project's issues and the possible solutions,*
- Adding new features to the company's products*

For each task/project, a new list should be added to the Trello board.

- *Build a workflow by click "Add a list" to add your first list to the board and title it to represent the stage or information that it will hold such as "To Do," "Doing," and "Done" or as detailed as needed for the work you are doing*
- *Add cards to the list, cards are used to represent tasks and ideas, so each card to include the projects documentation (Project overview, goals and objectives)*
- *Cards can be customized to hold a wide variety of useful information such as members, ideas, tasks, and due dates*
- *There's no limit to the number of cards you can add to a board*
- *If you want to add a team member to the task, click "Invite" and select a member*

11.2. Definition of Done

A task is done when it's tested and working status is confirmed with the team leader.

The member who implemented that task should include the required details about the task into the card, in the Trello board such as:

- A description about the task*
- What values has been added by implementing this feature*
- The process steps has been taken*
- The link to the Trello board*

The status of the card should be marked as Done,

11.3. Task Review

All contributions made to the remote repository in forms of updates, fixes, implementations, improvements, and new features are reviewed by Tech Lead Thomas Morgan. Reviews are made via Pull Requests in the Company GitHub and are reviewed and tested thoroughly.

Recommendations, comments, and requests for improvements had been made where needed throughout every PR made to the repository.

11.4. Testing

Testing is handled from repo branches in the hardware, where each feature and update is tested for validation, feasibility and efficiency via terminal monitoring, CMS output and debugging.

Unit Tests should be developed in future when workouts become more complex to ensure modular operation and validation.

Changes and edits that had to be made to the codebase on the pi was done via nano, as there is no current feature branch-style workflow to utilise in the company GitHub.

11.5. Branching Strategy

11.5.1. Current system:

We used an open-source strategy of forking the main repository and then cloning it to your local computer. work on each feature in its own branch on your computer and submit a pull request once it is complete. Upon review, the pull request will be merged into the main repository.

Note:

This workflow has proved to be extremely inefficient, as we are working with constantly testing and updating on hardware, its made it impractical to test implementations on hardware without overwriting main repo files. Typical industry practise for working directly with internal software/hardware in a large team with sprint style work methods do not operate in this “open source” manner.

Plans for future system:

11.5.2. Future Recommendations:

Each member should have access to committing and pushing to feature/bug fixing branches stemming from the main remote repo, (members who are not leadership will not have access to the main branch), and then create pull requests to merge their updates from this branch into the main repository. Only leadership team has the authority to accept and merge a branch into the main repository.

Reason:

This enables members to work on specific problems **on the hardware**, isolated from the main repository, without the need for technical leadership to handle every PR for small changes. This also allows team members to make significant contributions and for leadership team to easily maintain and observe commits

Process for members in this workflow:

Each member outside of leadership should have authority to commit to feature branches stemmed from main repo. (no authority to merge to main)

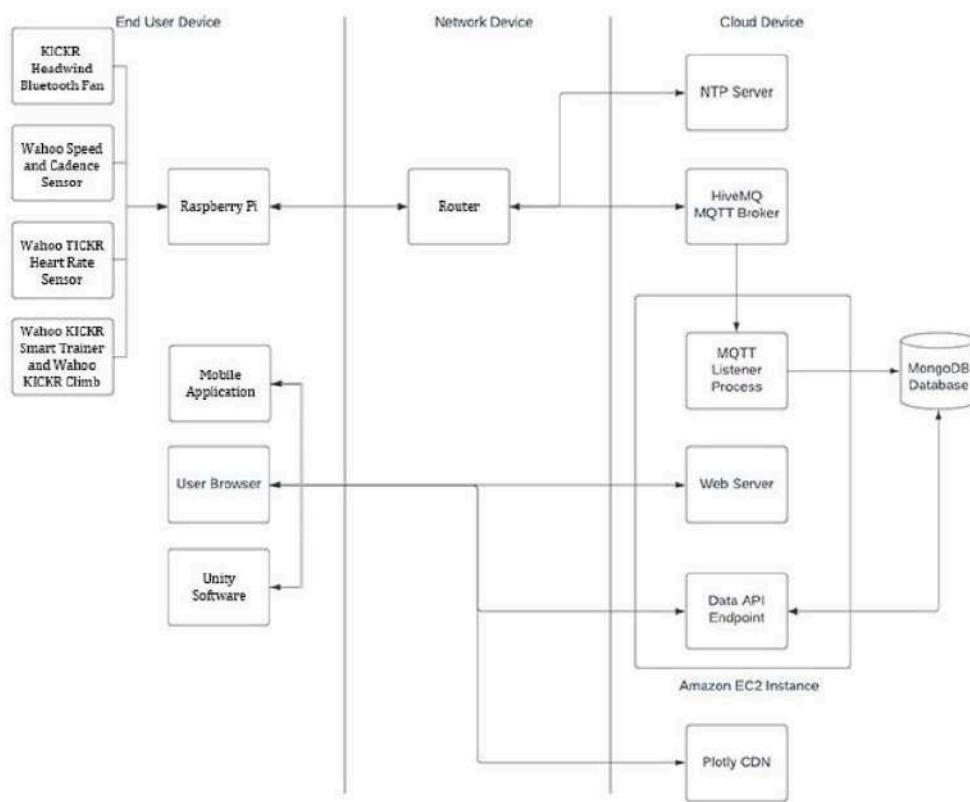
- Successfully clone main repo
- fetch or create new branches designed for specific tasks
- add, stage, commit and push update to these specific branches
- Create pull requests that can be merged in the main branch on successful review

Note:

- User must have PAT (Personal Access Token) to use as git password whenever pushing to branch
- specific config settings need changing:
 - run:
 - `git config --global credential.helper cache`
 - `git config lfs.https://github.com/redbackoperations/iot.git/info/lfs.locksverify false`

12. Product Architecture

12.1. UML Diagram



12.2. Tech Stack

- Hardware:
 - Wahoo KICKR Trainer (controls the resistance of the bike and provides sensor data. We chose this rather than an existing smart bike so that it would be easier to access the data)
 - Wahoo CLIMB Incline Simulator (controls the height of the bike)
 - Wahoo TICKR Heart Rate Sensor
 - Wahoo HEADWIND Fan
 - Random bike

- *Raspberry Pi*
- *A Mac is needed for compiling the mobile application for iPhone. A regular PC can create the Android version.*
- *Cloud hosting platforms:*
 - *MongoDB*
 - *HiveMQ*
 - *GCP (for data store and CMS)*
- *Software:*
 - *Most drivers were written in Python or NodeJS, as these have easy integration with MQTT libraries.*
 - *Paho-mqtt was chosen for use on the Raspberry Pi as there is help provided in the HiveMQ documentation on using this version.*
 - *Paho-mqtt-m2mqtt was wrapped for use with Unity as it supports MQTTS, many other libraries only support plain MQTT.*
 - *ReactJS was used for the CMS*

13. Source Code

All source code for the IoT team is in the repository
<https://github.com/redbackoperations/iot>

TODO: Provide a backup of the SD card. While it is mainly an install of Rasbian with a clone of the iot repository, it has had some set up to install required software and add the environment variables and wifi passwords.

14. Login Credentials

We use an open-source strategy of forking the main repository and then cloning it to your local computer.

You should work on each feature in its own branch on your computer and submit a pull request once it is complete.

Upon review, the pull request will be merged into the main repository. This system is in place because it is easy to accidentally create commits with thousands of unnecessary changes, and rolling this back is difficult.

If this occurs and you can't fix it easily, you can always back up the few files you did need to change and then delete your local repository, then clone your fork again and put your changes back in to give you a clean working copy again.

Reach out to your team leader if you need help with this.

As a result, only the team leaders will have direct access to the Github repository, others will be able to fork it and submit pull requests but not commit directly.

The team leaders should also try to ensure that the other team leader reviews their code rather than merging their own commits without review.

1) First log in to github with your account and create a fork of <https://github.com/redbackoperations/iot>

If your github login is ramiru then the fork will be at <https://github.com/ramiru/iot>

2) I like to work on a local copy of my fork, so then I would clone it to my computer
git clone <https://github.com/ramiru/iot>

Then create a new branch in the fork for your work, say marketresearch
git checkout -b marketresearch

Add the files to the new branch and commit them.

(copy the files into iot/Research on your computer)

git add .

git commit -m "NEW: Created market research report"

Then push the branch from your computer back to your fork on github
git push origin marketresearch

OR if you prefer, I think you can create a branch within your fork using the github website and upload your files directly to it and commit them. But you won't have a local copy on your computer that way for your future work.

3) Finally, log back into github and go to your branch within <https://github.com/ramiru/iot> Click on Compare and Pull Request and create the pull request. This will then come up for the team leaders to review

4) Since you are done, you can also move your Trello tickets to Review.

How Update the Local clone files:

Syncing GitHub Repos

When you are collaborating with others on a project, there are often changes being made to the repo that you (and others) are contributing to. It is important keep your fork up to date or in sync with those changes as you work. Keeping your fork in sync with the central repo will reduce the risk of merge conflicts.

Syncing Your GitHub Repo Reduces the Chances of a Merge Conflict

A merge conflict occurs when two people edit the same line in a file. Git does not know how to resolve the conflict (i.e. which changes to keep and which to remove).

*When **git** does not know how to resolve a conflict, it will ask you to manually fix the conflict. If you sync your files regularly, you will ultimately reduce the risk of a merge conflict.*

An Example Workflow Where Syncing Is Important

Pretend that you are working on a fork of your colleague's repo. Your colleague's repo is the final home for the code and content that you are working together on collaboratively.

Your colleague and others in your group may be updating code while you are working. It is important to ensure that your fork is in sync with your colleague's repo, ideally before making a new pull request to that repo.

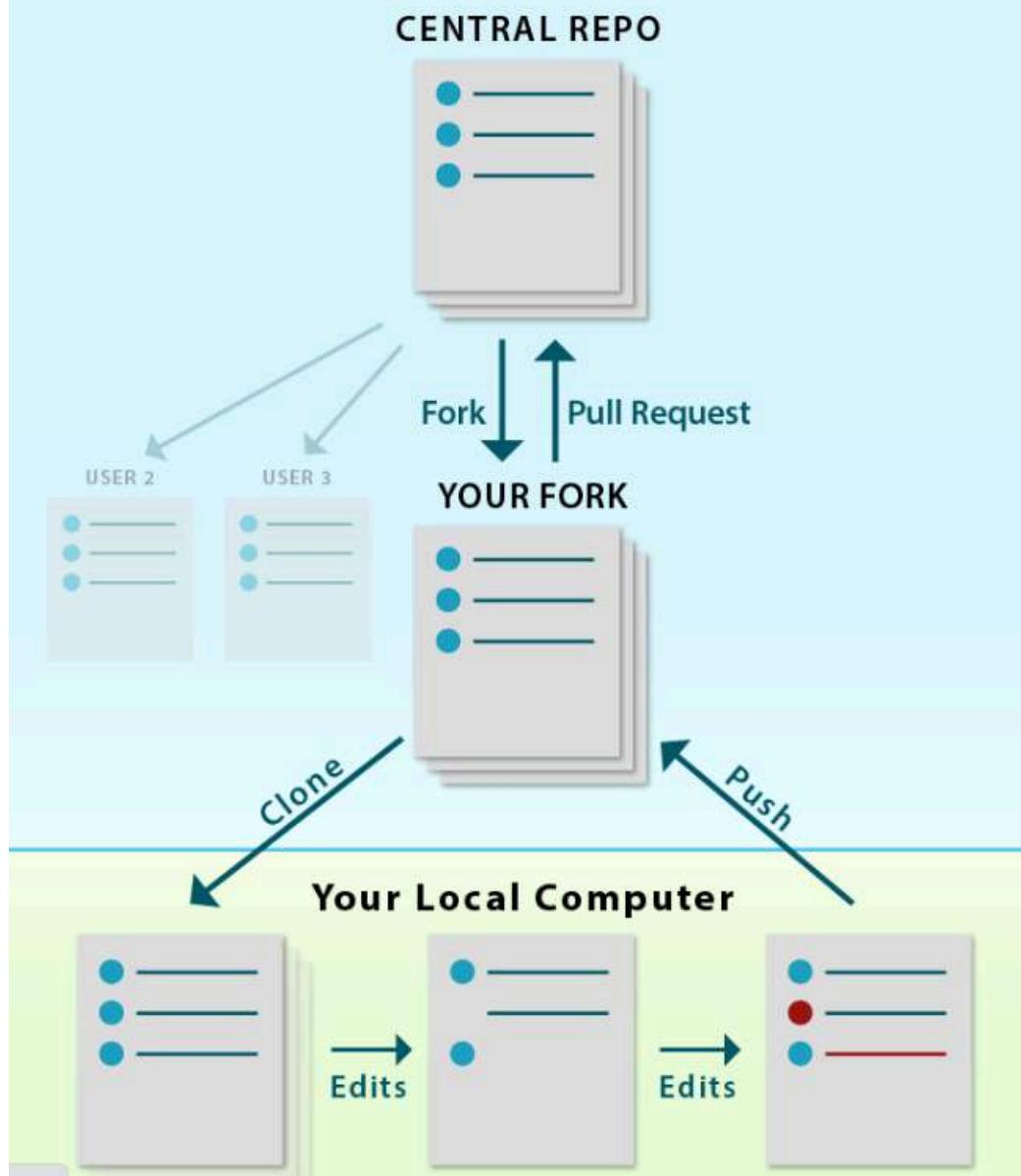
Your repo being in sync refers to your fork having all of the commits or changes to the code and files that have been made to the parent repo.

Two Ways to Sync A Repo - Command Line and on GitHub

There are a few ways to update or sync your repo with the central repo (e.g. your colleague's repo).

1. *You can perform a "Reverse Pull Request" on GitHub. A reverse pull request will follow the same steps as a regular pull request. However, in this case, your fork becomes the **base** and your colleague's repo is the **head**. If you update your fork this way, you will then have to PULL your changes down to your local clone of the repo (on your computer) where you are working.*
2. *You can manually set or pull down changes from the central repo to your clone locally. This can be done in the Terminal. When you update your local clone, you will then need to push the changes or commits back up to your fork on [GitHub.com](#).*

Github.com (Cloud)



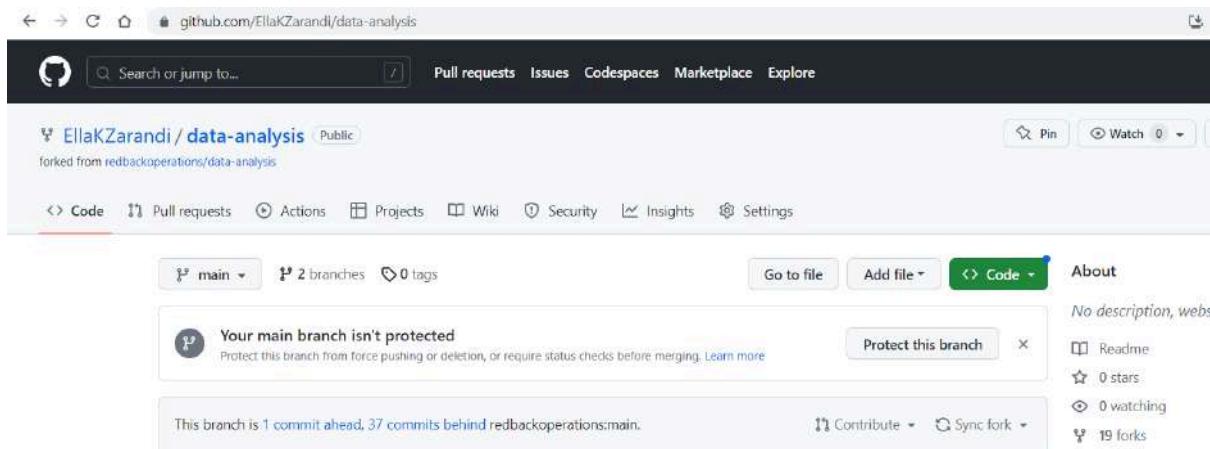
Sync Your Forked GitHub Repo Using a Reverse Pull Request

To sync your forked repo with the parent or central repo on GitHub you:

1. Create a pull request on **GitHub.com** to update your fork of the repository from the original repository, and
2. Run the `git pull` command in the terminal to update your local clone. The following sections review how to complete these steps.

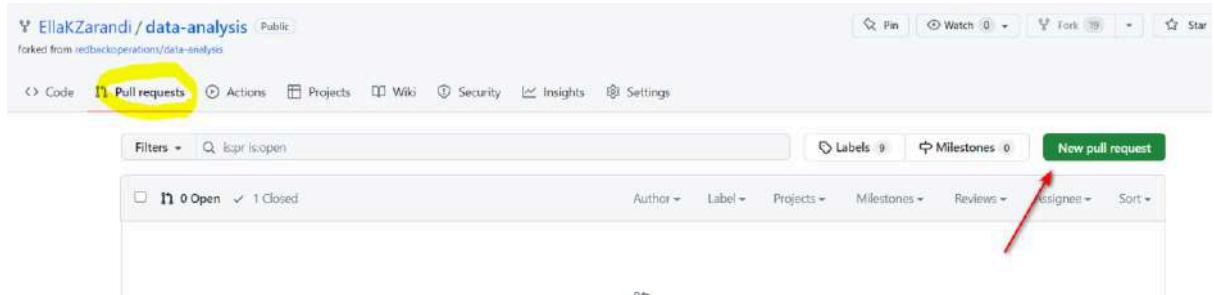
How to Sync or Update Your Forked Repo Using the Github Website

To update your fork on **GitHub.com**, navigate in your web browser to the main **GitHub.com** page of your forked repository (e.g. <https://github.com/your-username/example-repository>]



Then create a pull request by following these steps:

1. Click on the New pull request button to begin the pull request.



2. On the new page, choose your fork as the **base fork** and the original repository (e.g. your colleague's repo) as the **head fork**.
 - o **IMPORTANT:** You need to click on the text **compare across forks** to be able to select the base and head forks appropriately.

A screenshot of a GitHub fork page. At the top, it shows the repository 'EllaKZarandi / data-analysis' (Public) forked from 'redbackoperations/data-analysis'. Below the header are navigation links: Code, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. A yellow circle highlights the 'Compare across forks' link in the top right corner of the main content area. Red arrows point from the 'base repository' dropdown and the 'head repository' dropdown to their respective labels.

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks.

base repository: EllaKZarandi/data-analysis ▾ base: main ▾ head repository: redbackoperations/data-analysis ▾ compare: main ▾
✓ Able to merge. These branches can be automatically merged.

Discuss and review the changes in this comparison with others. [Learn about pull requests](#)

3. Then, click on *Create pull request*.

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks.

base repository: EllaKZarandi/data-analysis ▾ base: main ▾ head repository: redbackoperations/data-analysis ▾ compare: main ▾
✓ Able to merge. These branches can be automatically merged.

Discuss and review the changes in this comparison with others. [Learn about pull requests](#)

[Create pull request](#)

4. On the new page, add a title and comment and then click on *Create pull request* once more to finish creating the pull request.

A screenshot of the GitHub pull request creation page. The title field is filled with 'Synced fork Repo'. A red arrow points from the 'Preview' button to the title field. Another red arrow points from the 'Allow edits by maintainers' checkbox to the 'Create pull request' button. A yellow circle highlights the title input field.

You have now updated your local clone with the updates that you merged into your fork from original **GitHub** repository.

When you create this pull request, you will see what files will be updated in your fork.

Synced fork Repo #2

[! Open](#) EllaKZarandi wants to merge 37 commits into `EllaKZarandi:main` from `redbackoperations:main`

Conversation 0 Commits 37 Checks 0 Files changed 103

EllaKZarandi commented 1 minute ago

No description provided.

tejasvarun and others added 30 commits last month

- Add files via upload
- body posture check updated
- Merge branch 'redbackoperations:main' into main
- New dataset_Random Forest_stanford
- Add files via upload
- Add files via upload
- Merge pull request #25 from redbackoperations/Ella-Zarandi ...
- Delete BMI_Calculator.pdf
- Create Consultation Platform

Owner [...](#)

Reviewers
No reviews
Still in progress? [C](#)

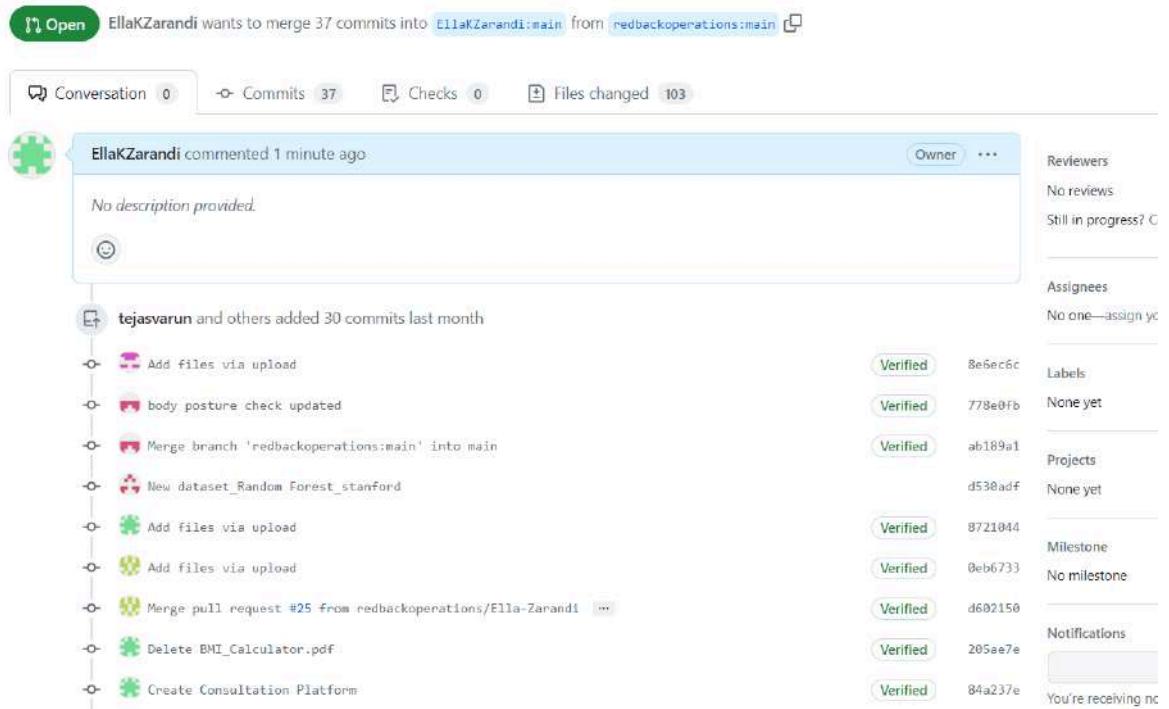
Assignees
No one—assign [you](#)

Labels
None yet

Projects
None yet

Milestone
None yet

Notifications
You're receiving no notifications



After creating the pull request, you need to **merge the pull request**, so that the changes in main repo are merged into your fork. The next section explains how to merge a pull request.

How to Merge a Pull Request

To merge a pull request:

1. Open up the pull request if it is not already open on **GitHub**.
2. Click on the green button at the bottom of the pull request page that says **Merge pull request**.

Synced fork Repo #2
EllaKZarandi wants to merge 37 commits into [EllaKZarandi:main](#) from [redbackoperations:main](#)

- o Merge pull request #33 from SaeedAlnaqeeb/main ... Verified b91be13
- o Merge pull request #32 from whosthis-svg/main ... Verified bd81a31
- o Merge pull request #31 from Nick-Manny/main ... Verified 4e41bfd
- o Merge pull request #30 from SindhujaManduru/main ... Verified 6a3d6e4

Add more commits by pushing to the [main](#) branch on [redbackoperations/data-analysis](#).

🔧 **Require approval from specific reviewers before merging**
Branch protection rules ensure specific people approve pull requests before they're merged.

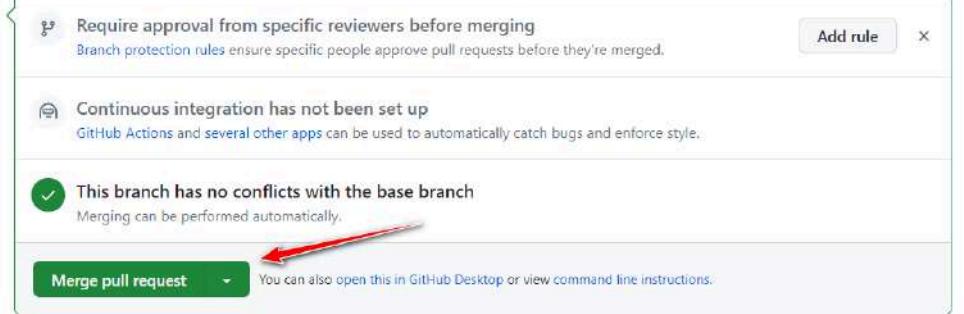
⌚ **Continuous integration has not been set up**
GitHub Actions and [several other apps](#) can be used to automatically catch bugs and enforce style.

✓ **This branch has no conflicts with the base branch**
Merging can be performed automatically.

Merge pull request You can also open this in GitHub Desktop or view command line instructions.

Write Preview

Leave a comment



3. Click on the *Confirm merge* button.

Synced fork Repo #2
EllaKZarandi wants to merge 37 commits into [EllaKZarandi:main](#) from [redbackoperations:main](#)

- o Merge pull request #33 from SaeedAlnaqeeb/main ... Verified b91b
- o Merge pull request #32 from whosthis-svg/main ... Verified bd81
- o Merge pull request #31 from Nick-Manny/main ... Verified 4e41
- o Merge pull request #30 from SindhujaManduru/main ... Verified 6a3d

Add more commits by pushing to the [main](#) branch on [redbackoperations/data-analysis](#).

Merge pull request #2 from redbackoperations/main

Synced fork Repo

This commit will be authored by 129282750+EllaKZarandi@users.noreply.github.com

Confirm merge Cancel

Write Preview

Leave a comment



Once you have conformed the merge, all of the changes from main repo are in your repo. When you return to your fork on [GitHub.com](#), you will see the changes that you have just merged into your fork.

When you update your fork using a reverse pull request on [GitHub.com](#), you then need to update your files locally. The steps to do that are below.

How to Update Your Local Clone

Once you have synced (i.e. updated) your fork on [GitHub.com](#), you are ready to update your cloned repo on your local computer.

To pull down (i.e. copy) the changes merged into your fork, you can use the Terminal and the `git pull` command.

To begin:

- 1. On your local computer, navigate to your forked repo directory.*

```
$ cd path-to-repo/repo-name  
$ git pull
```

- 2. Once you have changed directories to the forked repo directory, run the command `git pull`.*



The screenshot shows a terminal window titled 'MINGW64:/d/AIRepo'. The command \$ git pull is typed at the prompt. The terminal output shows the command being run and a vertical scroll bar on the right side of the window.

```
e11a@DESKTOP-LULUHRL MINGW64 ~  
$ cd D:\AIRepo  
e11a@DESKTOP-LULUHRL MINGW64 /d/AIRepo  
$ git pull|
```

*You have now updated your local clone with the updates that you merged into your fork from original **GitHub** repository.*

Dev/Ops Team

***Company ----Redback Operations
Project -----Deployment of Redback
website on GCP (Google Cloud Platform)***

Trimester 1, 2023

15. Project Information

15.1. Company Acting Director

Kevin Lee

Director

kevin.lee@deakin.edu.au

15.2. Project Team

Deployment of Redback website on GCP (Google Cloud Platform)

Redback Operations

Name	Level	Role
Arjodh Singh	Junior	Team Lead
Astifo Paul	Junior	Team Member
Adhish Anand	Junior	Team Member
Sam TitusMenacheery	Junior	Team Member
Steve Lee	Senior	Team Member

16. Project Overview

The main objective of the project was the deployment of Redback website on Google Cloud Platform. It entails the creation of CI/CD pipeline which would trigger every time any new changes to code are pushed to Git Hub and would be automatically deployed on Google Cloud. This would help the website to get updated with new code changes without undergoing any downtime.

17. User Manual

CI/CD Pipeline has been built for the Redback Website frontend on Goggle Cloud Platform. The pipeline is configured to be triggered every time new code was pushed into the minor/cloud branch of the GitHub repository. Once that happens, the pipeline:

- *Builds the Docker image*
- *Pushes the built Docker image into Google Cloud Artifact Registry*
- *Deploy the Docker image on Google Kubernetes Engine*

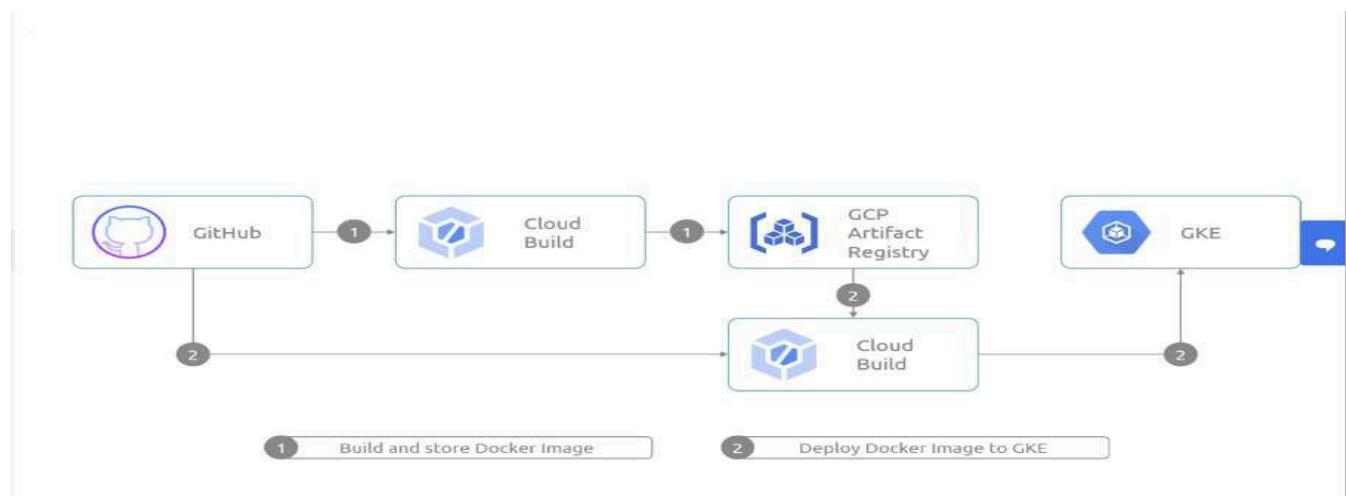
18. Completed Deliverables

CI/CD Pipeline is created for the redback website frontend.

Configuration files for deployment are stored on Git Hub.

Services such as Cloud Build, Artifact Registry and Google Kubernetes Engine are provisioned on GCP.

The architecture diagram for the entire process is:



Configuring Cloud Build Pipeline:

- The first step is responsible for building and tagging the Docker image of our application
- The second step is responsible for pushing the Docker image built on the step one to Google Artifact Registry
- The third step is responsible for deploying the Docker image on Google Kubernetes Engine

Arjodh-Singh / website-frontend Public

forked from redbackoperations/website-frontend

< Code Pull requests Actions Projects Wiki Security Insights Settings

Code

minor/cloudbuild + ⌂

Go to file ⌂

- .vs
- .vscode
- public
- src
 - .gitignore
 - Dockerfile
 - README.md
 - cloudbuild.yaml
 - gke.yaml
 - package-lock.json
 - package.json

website-frontend / cloudbuild.yaml

Arjodh Singh updated location 324e5de · 3 weeks ago History

Code Blame 16 lines (13 loc) - 436 Bytes

```

1 steps:
2
3   - name: "gcr.io/cloud-builders/docker"
4     args: ["build","-t","gcr.io/$PROJECT_ID/website-frontend","."]
5
6   - name: "gcr.io/cloud-builders/docker"
7     args: ["push","gcr.io/$PROJECT_ID/website-frontend"]
8
9   - name: "gcr.io/cloud-builders/gke-deploy"
10    args:
11      - run
12      - --filename=gke.yaml
13      - --image=gcr.io/$PROJECT_ID/website-frontend
14      - --location=us-central1
15      - --cluster=website-frontend
16      - --namespace=website-frontend-test

```

Create cloud build:

Google Cloud sit-23t1-cicd-pipeline-b7292e5 cloud build Search

Region us-central1

This dashboard shows recently triggered builds. You can see all builds, including manually submitted builds, on the History page.

Filter Enter property name or value

Successful: Arjodh-Singh/website-frontend - website-frontend

Latest Build	Duration	Trigger description	Source	Commit
5/5/23, 9:52 AM	00:02:37	This is the trigger for redback website	Arjodh-Singh/website-frontend	9844d11

Build History Latest View all Average Duration 00:02:47 Pass-Fail % 67% - 33%

Connect with your GitHub Repository:

The screenshot shows the Google Cloud Build Repositories interface. At the top, there are tabs for '1ST GEN' and '2ND GEN' (which is selected). A prominent message box titled 'Introducing Cloud Build repositories (2nd gen)' contains text about the second generation of Cloud Build repositories, mentioning better management of connections between Cloud Build and third-party source code providers. Below this, there are buttons for 'VIEW REPOSITORIES (2ND GEN)' and 'LEARN MORE'. A dropdown menu for 'Region' is set to 'us-central1'. Under 'Host connections', there is a table with one row: 'Arjodh Singh/website-frontend' connected to 'Cloud Build GitHub App'. There are also sections for 'Repositories' and 'CONNECT REPOSITORY'.

Setting up the Cloud Build Trigger:
Create Trigger on your branch minor/cloudbuild:

The screenshot shows the 'Edit trigger' page. The 'Source' is set to 'Arjodh-Singh/website-frontend'. The 'Name' is 'website-frontend'. The 'Region' is 'us-central1 (Iowa)'. The 'Description' is 'This is the trigger for redback website'. The 'Event' section is expanded, showing options for repository events: 'Push to a branch' (selected), 'Push new tag', and 'Pull request'. It also lists 'Or in response to': 'Manual invocation', 'Pub/Sub message', and 'Webhook event'. There are also 'DISABLE' and 'DELETE' buttons at the top left.

Create Google Kubernetes Cluster:

Make any small change in the code files and push changes to branch. This push will trigger the pipeline and all the steps are executed sequentially and new changes are deployed on Kubernetes cluster.

Let's say you change the code for Redback Operations Testing Version to Redback Operations:

Before:



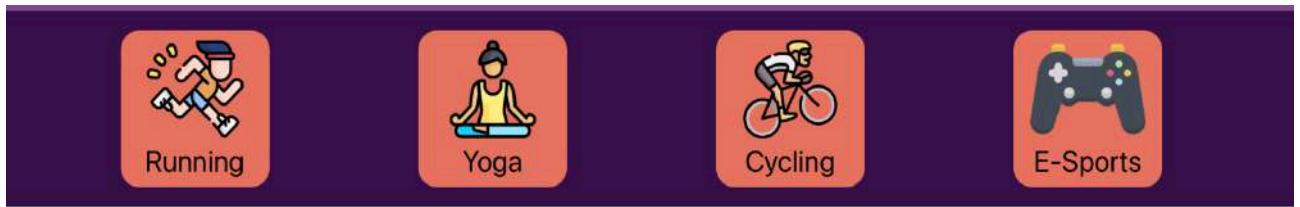
Welcome to Redback Operations Testing Version

Redback Operations builds cutting-edge technologies for connected health, fitness and sport to enable safe smart exercise. It is focused on providing a community-driven exercise and well-being gaming experience with competitive elements.

```
Article.jsx – website-frontend
1 import React from "react";
2 import "./HomeStyles.css"
3 import logo from "../img/Logo.png"
4
5 export default function Article() {
6   return (
7     <div className="background">
8       <div className="redback-banner">
9         <img className="home-logo" src={logo}/>
10        <div className="info">
11          <h3>Welcome to Redback Operations</h3>
12          <p>Redback Operations builds cutting-edge technologies for connected health, fitness and sport to enable safe smart exercise. It is focused on providing a community-driven exercise and well-being gaming experience with competitive elements.</p>
13        </div>
14      </div>
15    </div>
16  );
17}
18
19 }
```

The screenshot shows the VS Code interface with the Article.jsx file open. The code is a React component named Article. It imports React, HomeStyles.css, and the logo image. The component returns a div with a background class containing a redback-banner div with an image of the logo and an info div containing a h3 title and a paragraph describing the company's focus on connected health, fitness, and sport.

After:



Welcome to Redback Operations

Redback Operations builds cutting-edge technologies for connected health, fitness and sport to enable safe smart exercise. It is focused on providing a community-driven exercise and well-being gaming experience with competitive elements.

Check build history for build summary:

Build history								STOP STREAMING BUILDS		LEARN	
		Filter Enter property name or value									
	Status	Build	Source	Ref	Commit	Trigger Name	Created	Duration	Actions	Actions	Actions
	Success	21cd326d	Arjodh-Singh/website-frontend	minor/cloudbuild	9544d11	website-frontend	5/5/23, 9:52 AM	2 min 37 sec	⋮	⋮	⋮
	Success	47994a85	Arjodh-Singh/website-frontend	minor/cloudbuild	6ff6232	website-frontend	5/4/23, 1:35 PM	3 min 6 sec	⋮	⋮	⋮
	Success	a79bd364	Arjodh-Singh/website-frontend	minor/cloudbuild	3f33beb	website-frontend	5/4/23, 12:27 PM	3 min 3 sec	⋮	⋮	⋮
	Success	19f5a9ac	Arjodh-Singh/website-frontend	minor/cloudbuild	e92dab5	website-frontend	5/3/23, 8:22 PM	2 min 58 sec	⋮	⋮	⋮
	Success	6a8024d2	Arjodh-Singh/website-frontend	minor/cloudbuild	a93d632	website-frontend	5/1/23, 10:14 PM	2 min 54 sec	⋮	⋮	⋮
	Success	5c885cod	Arjodh-Singh/website-frontend	minor/cloudbuild	5ab18aa	website-frontend	4/29/23, 10:43 PM	2 min 49 sec	⋮	⋮	⋮
	Success	92a70005	Arjodh-Singh/website-frontend	minor/cloudbuild	baf5342	website-frontend	4/29/23, 10:18 PM	2 min 28 sec	⋮	⋮	⋮
	Success	81fae619	Arjodh-Singh/website-frontend	minor/cloudbuild	ecaaef5	website-frontend	4/28/23, 9:30 PM	2 min 55 sec	⋮	⋮	⋮
	Success	cb5e2c7a	Arjodh-Singh/website-frontend	minor/cloudbuild	1a3c48d	website-frontend	4/28/23, 8:58 PM	2 min 53 sec	⋮	⋮	⋮
	Success	529bf581	Arjodh-Singh/website-frontend	minor/cloudbuild	f0fa197c	website-frontend	4/28/23, 8:51 PM	3 min 4 sec	⋮	⋮	⋮
	Success	53c14cc3	Arjodh-Singh/website-frontend	minor/cloudbuild	2245dc6	website-frontend	4/28/23, 8:30 PM	6 min 32 sec	⋮	⋮	⋮
	Failure	c1432761	Arjodh-Singh/website-frontend	minor/cloudbuild	75646cf	website-frontend	4/28/23, 1:18 PM	2 min 33 sec	⋮	⋮	⋮
	Failure	f281da73	Arjodh-Singh/website-frontend	minor/cloudbuild	75646cf	website-frontend	4/28/23, 1:02 PM	2 min 14 sec	⋮	⋮	⋮
	Failure	d76dc1bb	Arjodh-Singh/website-frontend	minor/cloudbuild	75646cf	website-frontend	4/28/23, 10:07 AM	2 min 19 sec	⋮	⋮	⋮
	Failure	3bb8bee8	Arjodh-Singh/website-frontend	minor/cloudbuild	75646cf	website-frontend	4/28/23, 1:33 AM	1 min 51 sec	⋮	⋮	⋮
	Failure	a1ce33fe	Arjodh-Singh/website-frontend	minor/cloudbuild	82ec04d	website-frontend	4/28/23, 1:17 AM	1 min 44 sec	⋮	⋮	⋮
	Failure	454f79fc	Arjodh-Singh/website-frontend	minor/cloudbuild	f1add0c	website-frontend	4/28/23, 12:55 AM	2 min 8 sec	⋮	⋮	⋮
	Failure	3b6ffbf0	Arjodh-Singh/website-frontend	minor/cloudbuild	48d1aaf0	website-frontend	4/28/23, 12:51 AM	-	⋮	⋮	⋮
	Failure	3a1528ca	Arjodh-Singh/website-frontend	minor/cloudbuild	2ed344b	website-frontend	4/28/23, 12:48 AM	-	⋮	⋮	⋮
	Success	87ea59c0	Arjodh-Singh/website-frontend	minor/cloudbuild	4225981	website-frontend	4/27/23, 11:47 PM	1 min 59 sec	⋮	⋮	⋮
	Success	21fb26ec	Arjodh-Singh/website-frontend	minor/cloudbuild	71589f3	website-frontend	4/27/23, 11:39 PM	2 min 1 sec	⋮	⋮	⋮

Rows per page: 50 ▾ 1 – 21 of 21 < >

Successful: 21cd326d
Started on May 5, 2023, 9:52:02 AM

Trigger: website-frontend **Source:** Arjodh-Singh/website-frontend **Branch:** minor/cloudbuild **Commit:** 9044611

Steps	Duration	BUILD LOG	EXECUTION DETAILS	BUILD ARTIFACTS
Build Summary 3 Steps	00:02:37	<input type="checkbox"/> Wrap lines <input type="checkbox"/> Show newest entries first 1: starting build "21cd326d-6d9f-4ce9-bd4c-07a7f9de1a7a" 2: FETCHSOURCE 3: Hint: Using 'master' as the name for the initial branch. This default branch name 4: hint: is subject to change. To configure the initial branch name to use in all 5: hint: of your new repositories, which will suppress this warning, call: 6: hint: 7: hint: git config --global init.defaultBranch <name> 8: hint: 9: hint: 10: hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and 11: hint: 'development'. The just-created branch can be renamed via this command: 12: hint: 13: hint: git branch -M <name> 14: Initialized empty Git repository in /workspace/.git/ 15: From https://github.com/Arjodh-Singh/website-frontend 16: * branch 9844d1b775a1ce63926adcc0d5500cd098b892e8... -> FETCH_HEAD 17: HEAD is now at 9044611 demo version 2.2 18: BUILD 19: Starting Step #0 20: Step #0: Already have image (with digest): gcr.io/cloud-builders/docker 21: Step #0: Sending build context to Docker daemon 31.86MB 22: 23: Step #0: Step 1/6 : FROM node:12.18.1 24: Step #0: 12.18.1: Pulling from library/node 25: Step #0: 81fcf19181915: Pulling fs layer 26: Step #0: ec43ec6a2361: Pulling fs layer 27: Step #0: 828519924538: Pulling fs layer 28: Step #0: a6f58c47cna0: Pulling fs layer 29: Step #0: 3369007d7af21e: Pulling fs layer 30: Step #0: 922785fffa8f8: Pulling fs layer 31: Step #0: 0f4af08e16b: Pulling fs layer 32: Step #0: 2f3ae993e48c2: Pulling fs layer		<input type="checkbox"/> EXPAND <input type="checkbox"/> VIEW RAW

All the configuration files for cloud build and deployment are saved at:
<https://github.com/Arjodh-Singh/website-frontend/tree/minor/cloudbuild>

External end points for website deployed on GKE:

<http://34.67.157.179>

Name	Status	Type	Endpoints	Pods	Namespace	Clusters	Pods Running	Pods Desired
gcp-website-frontend-gke-service	OK	External load balancer	34.67.157.179.80	1/1	website-frontend-test	website-frontend	1	1

Created documentation to show how to setup a Jenkins server on GCP.
[How to Setup a Jenkins Server.docx](#)

Updated the Trello board.

Trello Board link: <https://trello.com/b/SvbARmmY/devops>

Key Achievements:

Name	Level	Achievements	Role
Arjodh Singh	Junior	<ul style="list-style-type: none"> Upskilled on DEVOPS stack and deployed Jenkins on VM on private and tested it for integration and deployment Conducted regular meetings and updated all the documents related to project Created end-end pipeline for redback website front end and deployed on GKE 	Team Lead
Astifo Paul	Junior	<ul style="list-style-type: none"> Upskilling on Jenkins, research and testing done to deploy it on VM. Also, research of CICD pipelines, Kubernetes engine, etc to continue work in DevOps. Organization and attendance of meetings to let all members know of the tasks/deadlines that are due. 	Team Member
Adhish Anand	Junior	<ul style="list-style-type: none"> Deployed Jenkins server on GCP vm instance and created a documentation on the same. Tested the working on Jenkins server using a sample job. Created docker image for the flutter application on the local along with its documentation 	Team Member
Sam TitusMenacheer y	Junior	<ul style="list-style-type: none"> Created a google cloud project. Created VM in google cloud for running Jenkins. Installed Jenkins in VM. Configured Jenkins into VM external link. Downloaded the Jenkins Plugins. Researching on building a Docker image of Git hub application from the git repository. and push the image into Google cloud Container Registry Preparing for the next step by researching Kubernetes cluster technology to host the application 	Team Member
Steve Lee	Senior	<ul style="list-style-type: none"> Tested the working on jenkins server using a sample job. Set up the yaml file for installing the monitoring tool (grafana) on the linux server so that it can be applied immediately after installing 	Team Member

		<p><i>kubernetes.</i></p> <ul style="list-style-type: none">• <i>Pre-preparation for Kubernetes installation completed.</i>	
--	--	---	--

19. Roadmap

The next trimester we will aim to complete the deployment of backend of Redback website.

20. *Open Issues*

The main issue that we encountered this trimester was limited privileges for deployment of Kubernetes cluster on Google Cloud Platform.

Re: FOLLOW UP: Capstone GCP - 2023 T1 - Arjodh Singh - Redback Operations - New Project Request

Friday, 28 April 2023 at 1:19 PM

ARJODH ARJODH SINGH <s221530673@deakin.edu.au>
To: _SIT Technical Team; **Cc:** Kevin Lee

From: ARJODH ARJODH SINGH <s221530673@deakin.edu.au>
Sent: Friday, April 28, 2023 2:31 AM
To: _SIT Technical Team <sit-technical-team@deakin.edu.au>
Cc: Kevin Lee <kevin.lee@deakin.edu.au>
Subject: Re: FOLLOW UP: Capstone GCP - 2023 T1 - Arjodh Singh - Redback Operations - New Project Request

Hi Team,
I hope you are doing fine.
I am getting stuck with the below error when trying to deploy containers on GKE. It's related to permission for the service account. Could you please look into this?



The screenshot shows a Google Cloud Build interface. The 'Build details' tab is selected, displaying a log of build steps. Step 175, titled 'Deploying configuration files', fails with the following message:
"F1: Failed to get image 'gcr.io/sit-23t1-cloud-pipeline-b7292eb/website-FrontendNodejs256_c9325877f656ab5b92cbcd' from container repository 'gcr.io/sit-23t1-cloud-pipeline-b7292eb/website-FrontendNodejs256_c9325877f656ab5b92cbcd'. Error: failed to get cluster credentials: failed to authorize access: command to get cluster credentials failed: Fetching cluster endpoint and token failed: failed to get token from endpoint 'https://container.googleapis.com/v1beta1/projects/sit-23t1-cloud-pipeline-b7292eb/locations/us-central1/clusters/gke-sit-23t1-cloud-pipeline-b7292eb'. ResponseError{code=403, message='Required 'container.clusters.get'' permission(s) for 'projects/sit-23t1-cloud-pipeline-b7292eb/locations/us-central1/clusters/gke-sit-23t1-cloud-pipeline-b7292eb' are missing or insufficient.'} F2: Saving suggested configuration files to 'output/suggested'."
The log also shows other steps like 'Pushing' and 'Creating deployment' which were successful.

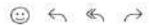
But that was resolved after working with Technical Team.

Re: FOLLOW UP: Capstone GCP - 2023 T1 - Arjodh Singh - Redback Operations - New Project Request

AA

ARJODH ARJODH SINGH <s221530673@deakin.edu.au>
To: _SIT Technical Team; Cc: Kevin Lee

Friday, 28 April 2023 at 1:19 PM



Good morning Arjodh,

I have now enabled the **Kubernetes Engine Developer** role for the 671060301813@cloudbuild.gserviceaccount.com service account. Please re-test and let me know if this resolves the issue.

If problems persist, can you re-send any errors / warnings in full? (your screenshot below is a bit cut-off in places)

Best of luck!

Service account permissions

Cloud Build executes builds with the permissions granted to the [Cloud Build service account](#) tied to the project. You can grant additional roles to the service account to allow Cloud Build to interact with other GCP services.

Service account email: 671060301813@cloudbuild.gserviceaccount.com

GCP Service	Role	Status
Cloud Functions	Cloud Functions Developer	● DISABLED ▾
Cloud Run	Cloud Run Admin	● DISABLED ▾
App Engine	App Engine Admin	● DISABLED ▾
Kubernetes Engine	Kubernetes Engine Developer	● ENABLED ▾
Compute Engine	Compute Instance Admin (v1)	● DISABLED ▾
Firebase	Firebase Admin	● DISABLED ▾
Cloud KMS	Cloud KMS CryptoKey Decrypter	● DISABLED ▾
Secret Manager	Secret Manager Secret Accessor	● DISABLED ▾
Service Accounts	Service Account User	● DISABLED ▾
Cloud Build	Cloud Build WorkerPool User	● DISABLED ▾

Roles not listed here can be managed in the [IAM section](#).

Regards,

Scott Blackburn
Senior Technical Officer, Cloud Computing & AI
School of Information Technology

21. Lessons Learned

The most important lesson was to have a fixed schedule for the meetings and that would be agreed by everyone. This would allow each team member to come into the meeting and speak his mind. If he is not technical strong enough to contribute, he must listen carefully about the progress of the project. This would enable him to give presentations or create documents for the team.

22. Product Development Life Cycle

22.1. New Tasks

The weekly meeting was done at 10 pm on Sunday and all the tasks were planned for next week and introspection was done on previous tasks. The tasks were created in Trello.

Trello Board link: <https://trello.com/b/SvbARmmY/devops>

22.2. Definition of Done

The task was consider done if it was able to get its objective with proper code and documentation.

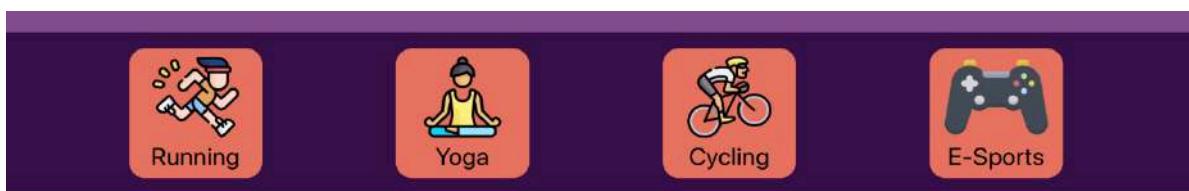
22.3. Task Review

Team leader was used to review the task of teammates by running the code on its machine and giving the required feedback to assignee.

22.4. Testing

Let's say you change the code for Redback Operations Testing Version to Redback Operations:

Before:



Welcome to Redback Operations Testing Version

Redback Operations builds cutting-edge technologies for connected health, fitness and sport to enable safe smart exercise. It is focused on providing a community-driven exercise and well-being gaming experience with competitive elements.



The screenshot shows the VS Code interface with the title bar "Article.js – website-frontend". The left sidebar displays the project structure:

- WEBSITE-FRONTEND
 - VS
 - website-frontend
 - VSCode
 - launch.json
 - public
 - index.html
 - logo92.png
 - logo512.png
 - manifest.json
 - robots.txt
- src
 - .vs
 - app
 - components
 - aboutUs
 - contactUs
 - home
 - img

After:



Welcome to Redback Operations

Redback Operations builds cutting-edge technologies for connected health, fitness and sport to enable safe smart exercise. It is focused on providing a community-driven exercise and well-being gaming experience with competitive elements.

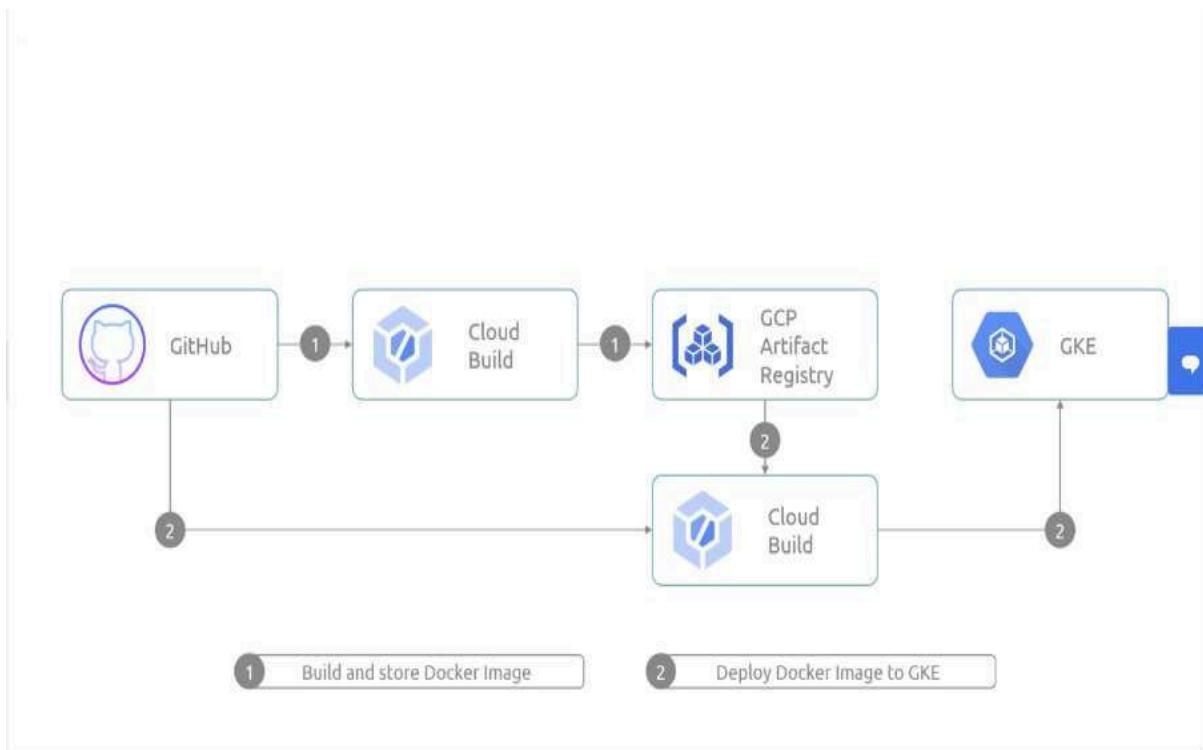
22.5. *Branching Strategy*

We use an open-source strategy of forking the main repository and then cloning it to your local computer. You should work on each feature in its own branch on your computer and

submit a pull request once it is complete. Upon review, the pull request will be merged into the main repository. This system is in place because it is easy to accidentally create commits with thousands of unnecessary changes, and rolling this back is difficult. If this occurs and you can't fix it easily, you can always back up the few files you did need to change and then delete your local repository, then clone your fork again and put your changes back in to give you a clean working copy again. Reach out to your team leader if you need help with this. As a result, only the team leaders will have direct access to the GitHub repository, others will be able to fork it and submit pull requests but not commit directly. The team leaders should also try to ensure that the other team leader reviews their code rather than merging their own commits without review

23. Product Architecture

23.1. UML Diagram



23.2. Tech Stack

Git Hub:

GitHub is an online software development platform. It's used for storing, tracking, and collaborating on software projects.

Google Cloud Build:

Cloud Build is a service that executes your builds on Google Cloud. Cloud Build can import source code from a variety of repositories or cloud storage spaces, execute a build to your specifications, and produce artifacts such as Docker containers or Java archives.

Google Artifact Registry:

Artifact Registry provides a single location for storing and managing your packages and Docker container images. You can: Integrate Artifact Registry with Google Cloud CI/CD services or your existing CI/CD tools. Store artifacts from Cloud Build.

Google Kubernetes Cluster:

Google Kubernetes Engine is a powerful cluster manager and orchestration system for running your Docker containers. It's built on the open-source Kubernetes system, giving you the flexibility to take advantage of on-premises, hybrid, or public cloud infrastructure.

24. *Source Code*

*All the configuration files for cloud build and deployment are saved at:
<https://github.com/Arjodh-Singh/website-frontend/tree/minor/cloudbuild>*

25. *Login Credentials*

N/A

26. *Appendices*

Project Handover Document

Redback Operations

Mobile App

Trimester 1, 2023

27. Project Information

27.1. Company Acting Director

Name Imali Dias

Title Dr

Email imali.dias@deakin.edu.au

Name Kevin Lee

Title Dr

Email kevin.lee@deakin.edu.au

27.2. Project Team

Mobile App

Redback Operations

<i>Name</i>	<i>Level</i>	<i>Role</i>
<i>Tung Truong The</i>	<i>Senior</i>	<i>Team Lead</i>
<i>Pengyu Xiao</i>	<i>Senior</i>	<i>Team Member</i>
<i>Raveen Yashod</i>	<i>Junior</i>	<i>Team Member</i>
<i>Manusha Umayanga</i>	<i>Junior</i>	<i>Team Member</i>
<i>Seth Tan</i>	<i>Junior</i>	<i>Team Member</i>

28. Project Overview

The Mobile App is one of the Redbacks products that the user can engage with. It is about developing an app that will comprise one of the possible experiences Redback customers use when they are exercising. The problem it aims to solve is providing customers with an accessible and portable solution that can be used to track their fitness. A mobile app can easily be downloaded and set up on exercise machines or for workouts when the user is on the move. The overall goal of the project is to provide a friendly and engaging user experience that will impact stakeholders by building a brand image for Redback giving it value while also strengthening customer loyalty. The main deliverable for the project will be an app with all the necessary features so that there is a complete experience from when the app is opened, and a workout is completed.

29. User Manual

The first step to do is asking for the permission for the [Mobile Team Trello](#), [GitHub](#) and Google Cloud Platform account from the company (for the map API and the Firebase else you might be charged).

This is the mobile team [Figma](#).

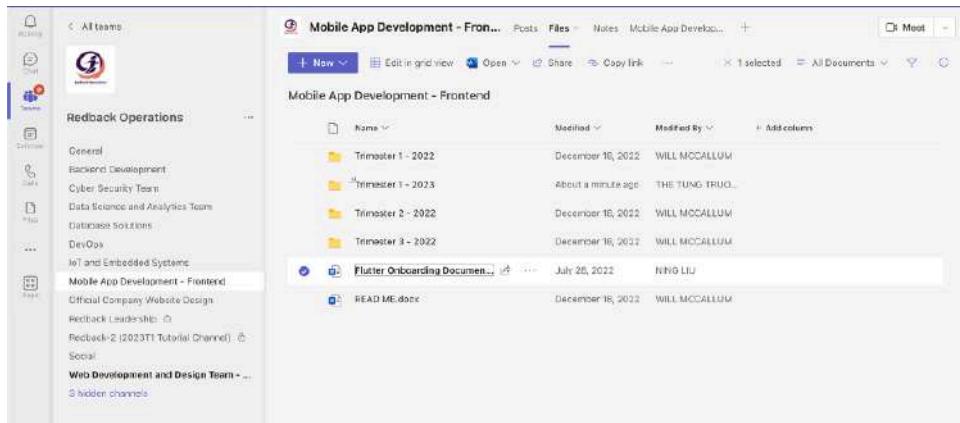
The instructions of how to currently set up the app can be found in the readme file inside the project [GitHub](#).

However, in this trimester, we made some changes to the backend which is rebased the previous backend to the Firebase Database so the instruction for the “Setting up the backend” might not work. In order to understand the step for setting up Firebase, please read [this](#). For further information about the Firebase credential, please contact Manusha Umayanga (s22231396@deakin.edu.au) through Teams.

This is the admin Gmail for the Firebase. May need to change to another Gmail because this is just a placeholder Gmail.

Username: redbackoperation@gmail.com

Password: redback123



The screenshot shows a Microsoft Teams channel interface titled "Mobile App Development - Frontend". The left sidebar lists several teams under "Redback Operations", including General, Backend Development, Cyber Security Team, Data Science and Analytics Team, Database Solutions, DevOps, IoT and Embedded Systems, Mobile App Development - Frontend, Official Company Website Design, Redback Leadership, Redback-2 (2023T1 Tutorial Channel), and Social. The main area displays a grid view of files and folders. The columns are Name, Modified, and Modified By. The visible items are:

Name	Modified	Modified By
Trimester 1 - 2022	December 16, 2022	WILL McCALLUM
Trimester 1 - 2023	About a minute ago	THE TUNG TRUONG
Trimester 2 - 2022	December 16, 2022	WILL McCALLUM
Trimester 3 - 2022	December 16, 2022	WILL McCALLUM
Flutter Onboarding Document	July 26, 2022	NING LIU
README.docx	December 16, 2022	WILL McCALLUM

The newest code has been updated inside the “Main” branch, future student can start by cloning this branch.

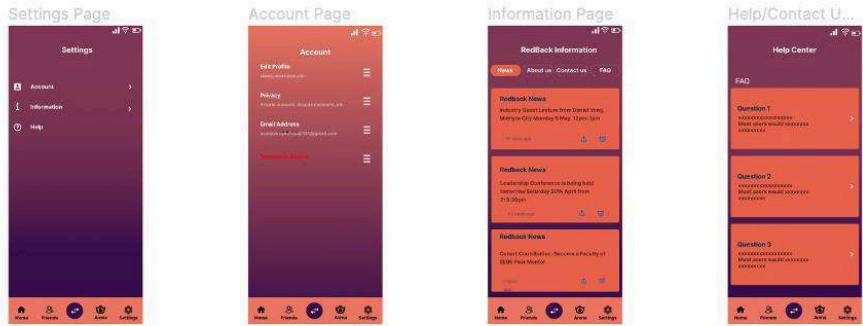
For further instruction on Flutter command or setting up device can be found in the ‘Mobile App Dev Private’ channel within the company Teams.

30. Completed Deliverables

This trimester we finished some screens from the Figma:

Role 4

Settings Page
→ Account page
→ Information page
→ Help/Contact Us page

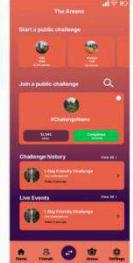


Role 3

Friends Page
Arena Page



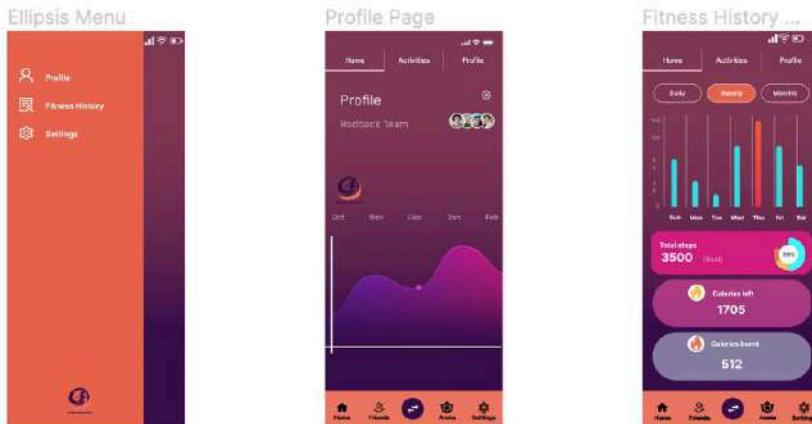
Arena Page:



Role 5

Ellipsis Menu

- Profile page
- Fitness History page
- Settings page



The screen design has been fixed to match the previous trimester Figma.

Besides that, we set up the connection to Firebase, users can sign-up for an account in the signup screen or log in using their Gmail account.

https://deakin365-my.sharepoint.com/personal/s222184591_deakin_edu_au/_layouts/15/stream.aspx?id=%2Fpersonal%2Fs222184591%5Fdeakin%5Fedu%5Fau%2FDocuments%2FMicrosoft%20Teams%20Chat%20Files%2FApp%20Demo%20Raveen%20%2D%20HD%201080p%2Emp4&referrer=Teams%2ETEAMS%2DELECTRON&referrerScenario=p2p%5Fns%2Dbim&qa=1

However, if you are using the simulator to use this feature, it might crash if you haven't added the computer certificate to the Firebase console.

RAVEEN YASHOD KULATHUNGA SILMAN ARACHCHILAGE Sunday 18:01

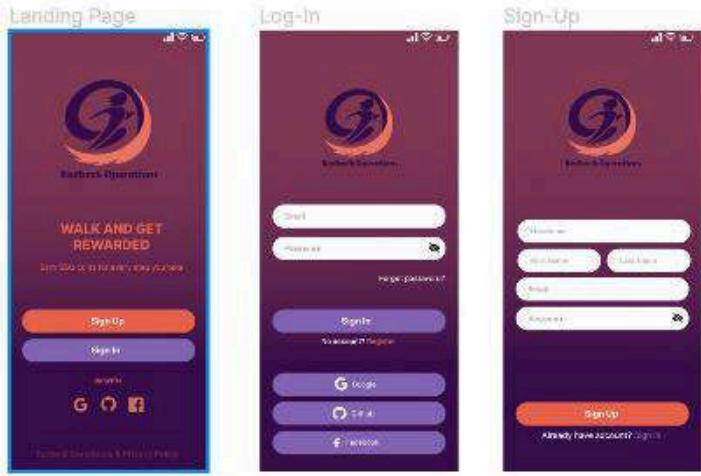
The issue is we are running the app on debug mode in the Android studio, so it does not recognise the authentication genuinely. We must add our computer certification to the Firebase console for that. On my computer, I did that but on other computers not. That's an issue I found. If anyone needs to log in using a Google account in the debug mode, they must add their computer certifications files to Firebase. It should be ok when the mobile app is completely installed on a mobile device..

31. Roadmap

I will demonstrate the roadmap based on the app flow.

Role 1

Landing Page
→ Log-In page
→ Sign-Up page



For the first flow, we implemented the login using the Google account, however, logging in through GitHub and Facebook is not working now.

Role 2

Home Page
→ Workout Type page
→ Workout Component Feature 1
→ Workout Component Feature 2
→ Workout Component Feature 3
→ Complete Workout Component Feature 1
→ Complete Workout Component Feature 2
→ Complete Workout Component Feature 3



The second flow needs to be redesigned to match the Figma. Please contact the Design team to choose the Figma design that they want or that you think can be implemented. Most of our work this trimester is based on the previous trimester because the designs from students this trimester created are not finished when we start.

The middle screen of the flow needs to implement paging features between the workout screen and the map screen. We created a "map.dart" file but it is not finished yet.

Please research how to connect Flutter with Google API framework (`google_maps_flutter`). An API key is needed, please put the API in the following directory:

- For Android: `/app-frontend/android/app/src/main/AndroidManifest.xml`
- For iOS: `app-frontend/ios/Runner/AppDelegate.swift`

- Inside the `constant.dart`: `app-frontend/lib/Utils/constants.dart`

Besides that, we need to connect with the IOT team to display information such as average speed, distance, or heart rate in the workout screen. Please work with the IOT team and research this topic.

Role 3

Friends Page
Arena Page

Friends P...



Arena Pa...



Role 4

Settings Page
→ Account page
→ Information page
→ Help/Contact Us page

Settings ...



Account ...



Informatio...



Help/Con...



Role 5

Ellipsis Menu
→ Profile page
→ Fitness History page
→ Settings page



The flow 3, 4, and 5 don't have much to do. Our team this trimester finished all the designs for these flows. The logic for these flows needs to be discussed with the design team or the company.

32. Open Issues

- Many students are not experienced with Flutter framework so it can slow down development. A Flutter Onboarding Document has been made to address this and can be found in 'Mobile App Dev Private' channel.
- The Figma and the company workflow are ambiguous.
- The project doesn't have a good assessment process. Some students from the previous trimester just created a screen to match the design and didn't even merge to the main branch or connect to the correct flow order. This trimester, we had to spend a lot of time doing this. Besides that, the project doesn't have a good design pattern and architecture since it started, so the code on some screens is a big mess.
- The coding skill between members is not equivalent.

33. Lessons Learned

The biggest lesson learned this trimester is the importance of communication between team members/ other teams in the company. In order to develop the app, we need strong communication with the Design Team and the IOT team. The emphasis of communication is because last trimester it was lacking, and it led to differences which affected progress.

34. Product Development Life Cycle

34.1. New Tasks

Tasks created based on requirements set out by the directors and company leads as well as other teams. Tasks need to be created from the Trello “To-do” column, then from that we assign them to the suitable member.

There is a weekly meeting where we discuss the task that we should do/assign with other members. Besides that, the meeting should be a place where team members discuss issues and work progress.

34.2. Definition of Done

Once the task is done, they communicate with the team lead to get their approval and once it is all good the task is moved to ‘Done’ list in Trello.

A successfully completed task depends on its requirements. If it is just a design of a page and it matches the corresponding Figma page along with code that is efficient, then it is done. Please don’t create fixed size components, components should be set based on device screen ratio. I created a common in Utils/size_config.dart. Use it to retrieve the device screen size. If the page needs functionality like pressing a button and going to the next page, then its implementation is checked to make sure it is being done the same way as every other page so that the code is consistent.

34.3. Task Review

The tasks are reviewed by the team lead who can communicate it to the leads of the other teams for validation if they need to.

A GitHub action called Flutter Lint has also been set up which checks the code and flags errors when trying to commit to the repository. This is the final review after the team lead and then the code is ready to push once there are no more errors.

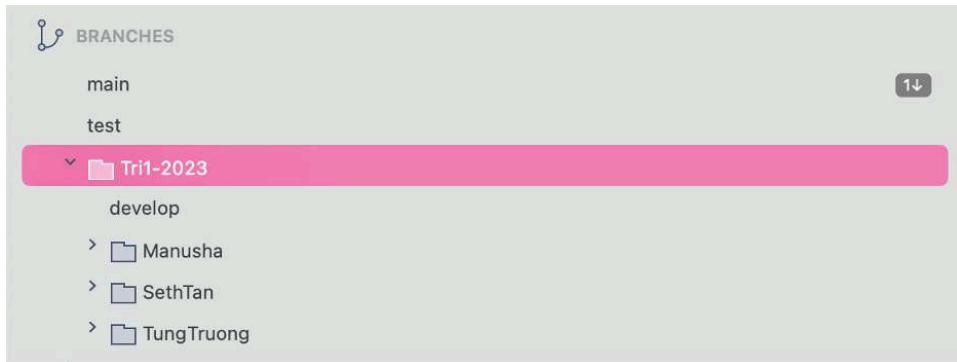
34.4. Testing

The product is tested locally through a browser, emulator, or physical device to make sure it meets desired functionality. Once that functionality works and the code is pushed to a branch, it is merged with other pages and tested in a similar way but on a slightly larger scale with all pages to make sure it integrates well with the app.

Besides that, please read [this](#), it is a tool for UI testing. I created a demo on [Trello](#). After running the tool, if it passed all cases then it can be considered as done.

34.5. Branching Strategy

All the code is merged inside the Main branch. Each member should have a different branch to avoid conflict. The recommendation here is that the leader should create a folder with hierarchy like this.



The first folder should indicate the current trimester, next is the folder based on the member's name and then the working branch is put inside.

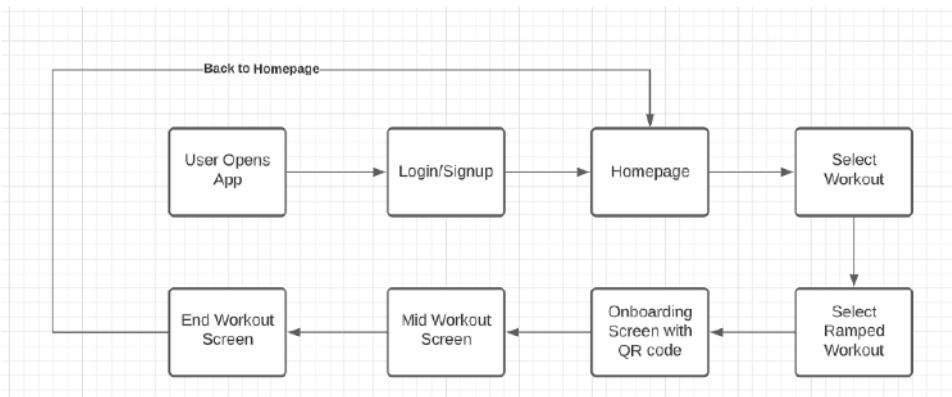
The "trimester_name" folder should have a "develop" branch in order for members to merge their code. If the code merged to the "develop" branch passes through all the test and cause no conflict, then you can merge to the main branch.

When first starting the "develop" branch must be branched from the main branch.

The "develop" branch should be pulled frequently. Each member should fetch and pull code from the develop branch to their branch every time they are working. This can help you avoid conflict, and all the code will be up to date.

35. Product Architecture

35.1. UML Diagram



35.2. Tech Stack

- Flutter – as it is a cross platform software with a lot of capability and resources available online.
- VS Code/ Android Studio – IDE to code Flutter, Android Studio also helps dealing with Android Configuration
- XCode – Handle the iOS configuration
- GitHub – most common version control system that anyone can learn
- Teams – makes it easy to communicate with all students and share files
- Trello – allows for tasks to be managed clearly
- Firebase – Database

- *Maestro – Testing tools*

36. Source Code

App front-end code: <https://github.com/redbackoperations/app-frontend>

Figma: <https://www.figma.com/file/tvpePl4wuKneyD9faU1yw/Mobile-App-Flow?type=design&node-id=743-239&t=uSUDt4OxkRvy3XiZ-0>

Trello: <https://trello.com/b/cNxiN0iZ/mobile-app-development>

37. Login Credentials

App Credential:

- Username: user@user.com
- Password: user@123

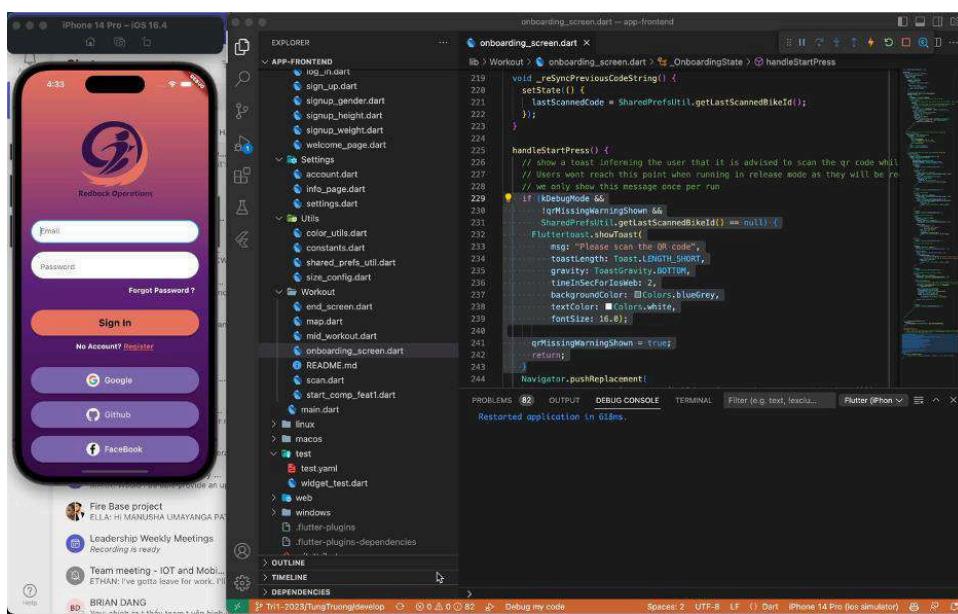
Or you can create your wanted account on the sign up screen.

Admin Gmail account for Firebase:

- Username: redbackoperation@gmail.com
- Password: redback123

38. Other Relevant Information

This is the current app flow for the second flow. To access the workout screen, please comment on this part of the code inside the “onboarding_screen.dart”. This is a tip if you are using simulators.



Else, you can use the real device for scanning the QR code inside this path:
app-frontend/assets/QRCodes.pdf

39. Appendices

Project Handover Document

***Cyber Team
Company Redback***

Project 2-factor authentication

Trimester 1, 2023

40. Project Information

40.1. Company Acting Director

Kevin Lee

Associate Head of School, Planning and Strategy

kevin.lee@deakin.edu.au

40.2. Project Team

2-factor authentication project

Redback Operations

Team Members

Tahlea Grant

Jikuan Liu

2-factor authentication project team leader

s222034126@deakin.edu.au

TAHLEA GRANT

2-factor authentication project team member

41. Project Overview

41.1. What is the project about?

The project aims to continue the previous trimester's 2-factor authentication implementation, with the motivation being secure and safe logins which ensure user authenticity and prevent malicious users gaining access to the company websites and databases.

41.2. What problem is the project solve?

This positively impacts stakeholders as by implementing a 2FA system the attack space is reduced, and the likelihood of attack is minimised. Therefore, the cost of recovery to the company is diminished greatly, as well as the risk for negative publicity which may lead to distrust from clients.

41.3. What are the aims of the project?

The overall goal is to develop new functions based on previous works and ensure that a functional 2FA system is implemented at the website without any bugs or implications.

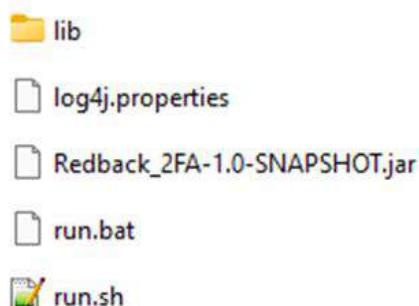
41.4. What are the deliverables?

- provide a randomly generated QR code for users to scan using their authenticator apps such as DUO app and Google Authenticator app to gain a time-based one-time password(TOTP) which is input on Redback's log-in page when they log in.
- develop an authentication API which accepts user-input TOTP password and verifies it.
- contacted frontend team to integrate all above code.
- develop a program using atmospheric Noise to generate truly random secret keys which can't be guessed by attackers.

42. User Manual

Use maven to build the project, and then uncompress the 'Redback_2FA-1.0-SNAPSHOT-assembly.tar.gz' archive file. The structure of extracted files looks like this:

Unzip The

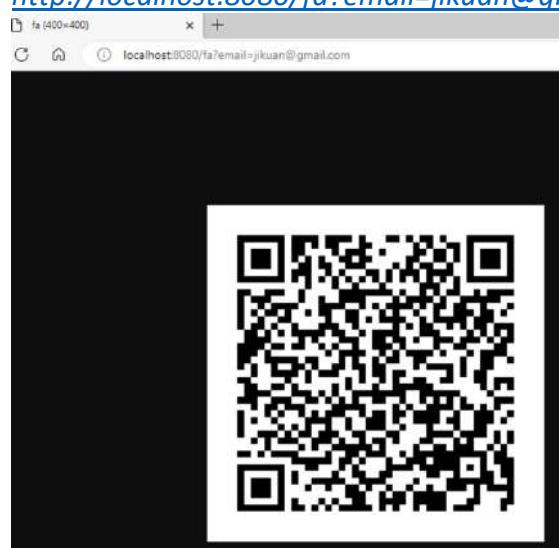


In windows, open a command terminal, then input "./run.bat"; In linux, open a command terminal, then input "sh run.sh":

```
T> .\run.bat
the default listening port is 8080
port:
```

Input a new port or leave it.

After pressing enter key, open a browser, input "<http://localhost:8080/fa?email=jikuan@gmail.com>".



Scan the QR code using any authenticator apps(e.g., google authenticator, DUO Mobile, and so forth), and then get the Time-based one-time password(TOTP) from the apps. Remember the TOTP.

Open a browser again, and then input the following address:

<http://localhost:8080/verify?email=jikuan@gmail.com&code=640122>

where the value of the ‘email’ parameter must be the same as the one input to generate the QR code, the value of the code must be the TOTP got just now. Therefore, if the TOTP is 134366, then the input address must be:

<http://localhost:8080/verify?email=jikuan@gmail.com&code=134366>

If the TOTP is input timely, the page will show the “pass” word; otherwise, it will show the “fail” word.



43. Completed Deliverables

- Developed two different application programming interfaces(API) done by Jikuan Liu. The first API is to provide a randomly generated QR code for users to scan using their authenticator apps such as DUO app and Google Authenticator app. Accordingly, they can gain a time-based one-time password. Then they can input this password on a log-in page when they need to log in to the company’s website. The second API called authentication API accepts the user-input password and verifies it. If verification is successful, then grant the user the privilege to access the internal information of the company’s website. This practice can help authenticate the user’s identity and prevent non-legitimate access. The code was uploaded to the Redback Company’s GitHub page
- Reached out to front-end team leaders after all code jobs are finished. They were informed that the 2-factor authentication(2FA) project is ready to be integrated into their project. They also already received the instruction manual for the 2FA project.
- Published a research report on how to use Atmospheric Noise to generate truly random secret keys which largely enhances the security of users’ accounts. This report was written by the team member Tahlea.

Important links can be found at

- Implementation of 2-factor authentication – unknown [Accessible via this link](#) (to github)
- Cyber Security Team 2.1P Updated - Cyber Team Trimester 1 2023 [Accessible via this link](#) (to sharepoint)
- Code [Accessible via this link](#) (to github)
- Random Key Gen Project code [Accessible via this link](#) (to github)
- The Document for the Random Key Gen Project code [Accessible via this link](#) (to pdf on github)
- Key Gen specific Documents [Accessible via this link](#) (to Teams)

44. Roadmap

- Currently the frontend team promised they will integrate the 2FA project to their projects. The integration job should be done by the frontend team members in the next trimester.
- Currently the Atmospheric Noise research report written by Tahlea should be implemented using JAVA and integrated into the 2FA program. This job should be done by the cyber security members in the next trimester.
- pairs of a email and a key should be stored in a database. This job should be done by backend team in the next trimester.

45. Open Issues

To make it run continuously in productive environments which requires to be able to simultaneously handle multiple requests, a thorough test should be done before publishing it to the corporate go-live environment. Currently, the program cannot process simultaneously multiple requests.

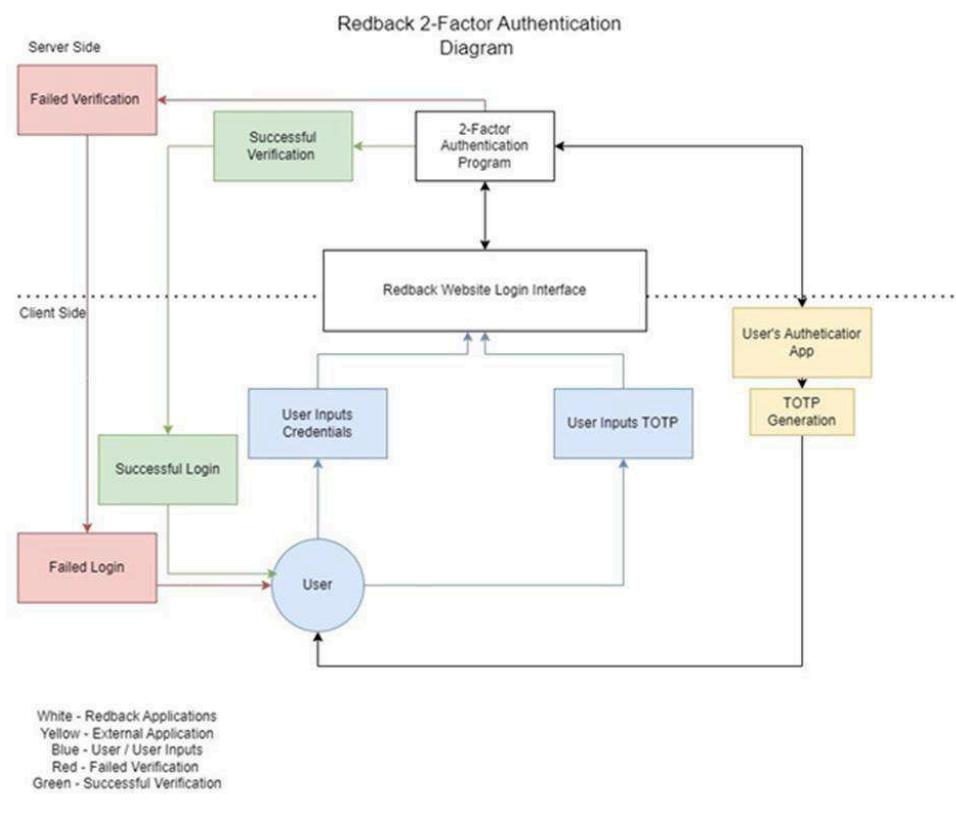
Additionally, the truly random key gen currently only exists within its own project and has yet to be integrated into the Redbacks 2FA.

46. Lessons Learned

Keeping track of each member's progress and reminding them of critical date points are very important.

47. Product Architecture

47.1. UML Diagram



47.2. Tech Stack

The 2FA project is written in the latest version of JAVA 8; IntelliJ idea is an Integrated Development Environment which was also used.

Redback_2FA-1.0-SNAPSHOT-assembly.tar.gz is a built parcel using Maven.

48. Source Code

Source code of 2FA, instruction manual of 2FA, Atmospheric Noise document are as follows:

[https://github.com/redbackoperations/Cyber-Security/tree/main/T1_2023/Project%202-Factor%20Authentication%20\(2FA\)](https://github.com/redbackoperations/Cyber-Security/tree/main/T1_2023/Project%202-Factor%20Authentication%20(2FA))

Trello:

<https://trello.com/b/kj9wulDB/cyber-security>

binary distributions built by using source code of 2FA:

[https://github.com/redbackoperations/Cyber-Security/tree/main/T1_2023/Project%202-Factor%20Authentication%20\(2FA\)/Binary%20Distributions](https://github.com/redbackoperations/Cyber-Security/tree/main/T1_2023/Project%202-Factor%20Authentication%20(2FA)/Binary%20Distributions)

49. Appendices

Installation and run online video tutorial:

<https://screenrec.com/share/iu0ApzNR6T>

Project Handover Document
Cyber Team

Company: Redback Operations

Project: Implementation of SIEM

Trimester 1, 2023

50. Project Information

50.1. Company Acting Director

Kevin Lee

Associate Head of School, Planning and Strategy

kevin.lee@deakin.edu.au

50.2. Project Team

Implementation of SIEM

Redback Operations

Team Members

Adam Bainey

Caitlin Parker

Carla Estella

Stephen Tobechuku

51. Project Overview

This project is the implementation and of a Security Incident and Event Management (SIEM) system into the Redback Operations environment. Once the SIEM is set up and the rules have been put into place this will allow the cyber team to monitor for anything that could be flagged as an issue or malicious. This will involve setting up the SIEM environment and then populating it with rules that Redback Operations can then use to trigger on key events, such as unauthorised access.

52. User Manual

This instruction is not exhaustive in nature but will provide a baseline and all information has been reference for [Wazuh](#) & [OpenCTI-Platform](#) & [ElasticSearch](#).

It enables the ability to create consistent development, testing, and production environments, ensuring that these applications run the same way across different environments.

Allows ease of orchestration and to manage containers, including starting, stopping, and scaling containers as needed.

The installation will refer to a Docker-Compose and an .env these files and additional sources to install Installing Elasticsearch

52.1. Pre-requisites

- Access to GCP instance for the project
- Administrative privileges or access to the root user.
- Some prior knowledge in Linux systems
- Access to previous docker compose and .env files
- Access to Git command
- Requirements for Wazuh host
- Requirements for OpenCTI host
- Requirements for ElasticSearch host

Host requirements of Wazuh will depend on the number of agents required.

Simple guide

Agents	CPU	RAM	Storage (90days)
1-25	4 vCPU	8 GiB	~50GB

Host requirements of OpenCTI

CPU	RAM	Storage
2 vCPU	8 GiB	~10GB

Host requirements of ElasticSearch

CPU	RAM	Storage

52.2. Tasks Description – Wazuh Docker Deployment

Deploying Wazuh as a single-node which will include:

This section deploys Wazuh manager, Indexer, and Dashboard node on Docker using sudo privileges

Step	Command to execute
1.	<code>apt update && apt upgrade -y</code>
2.	<code>echo "vm.max_map_count=262144" sudo tee -a /etc/sysctl.conf</code>
3.	<code>curl -sSL https://get.docker.com/ sh</code>
4.	<code>curl -L <a data-bbox="493 586 1377 691" href="https://github.com/docker/compose/releases/download/v2.12.2/docker-compose-\$(uname -s)-\$(uname -m)">https://github.com/docker/compose/releases/download/v2.12.2/docker-compose-\$(uname -s)-\$(uname -m)" -o /usr/local/bin/docker-compose</code>
5.	<code>chmod +x /usr/local/bin/docker-compose</code>
6.	<code>#Docker and Docker compose should now be deployed</code>
7.	<code>cd /opt/ && mkdir deploy && cd deploy</code>
8.	<code>git clone https://github.com/wazuh/wazuh-docker.git -b v4.4.1</code>
9.	<code>cd wazuh-docker/single-node/</code>
10.	<code>docker-compose -f generate-indexer-certs.yml run --rm generator</code>
11.	<code>docker-compose up -d</code>
12.	<code>#Wazuh should be running</code>

52.3. Post-installation

The installation instructions should contain a check list that details

- Navigate to https://{{IP_ADDRESS}}
- Login with the default Credentials and changed
- Check all functionality from the dashboard. Wazuh conducts automatic prechecks as it loads.

52.4. Tasks Description – OpenCTI Docker Deployment

Deploying OpenCTI which will include:

This section deploys OpenCTI node on Docker using sudo privileges its is expected to be run on a separate host to the Wazuh instance.

Step	Command to execute
13.	<code>apt update && apt upgrade -y</code>
14.	<code>echo "vm.max_map_count=1048575" sudo tee -a /etc/sysctl.conf</code>
15.	<code>curl -sSL https://get.docker.com/ sh</code>
16.	<code>curl -L https://github.com/docker/compose/releases/download/v2.1</code>

	<code><u>2.2/docker-compose-\${uname -s}-\${uname -m}" -o /usr/local/bin/docker-compose</u></code>
17.	<code>chmod +x /usr/local/bin/docker-compose</code>
18.	<code>#Docker and Docker compose should now be deployed</code>
19.	<code>cd /opt/ && mkdir deploy && cd deploy</code>
20.	<code>#copy the docker docker-compose.yml for from the repository and the .env file. Modify values for username and passwords.</code>
21.	<code>set -a ; source .env</code>
22.	<code>docker-compose up -d</code>
23.	<code>#OpenCTI should be running</code>

52.5. Post-installation

The installation instructions should contain a check list that details

- Navigate to http://{IP_ADDRESS}
- Login with the Credentials set in the .env file
- Check all functionality OpenCTI conducts automatic prechecks as it loads.

52.6. Tasks Description – ElasticSearch Docker Deployment

Deploying ElasticSearch which will include:

This section deploys ElasticSearch node on Docker using sudo privileges it is expected to be run on a separate host to the ElasticSearch instance.

Step	Command to execute
1.	<code>apt update && apt upgrade -y</code>
2.	<code>echo "vm.max_map_count=1048575" sudo tee -a /etc/sysctl.conf</code>
3.	<code>curl -sSL https://get.docker.com/ sh</code>
4.	<code>curl -L "<a -o="" a="" bin="" docker-compose<="" href="https://github.com/docker/compose/releases/download/v2.1.2/docker-compose-\${uname -s}-\${uname -m}" local="" usr=""></code>
5.	<code>chmod +x /usr/local/bin/docker-compose</code>
6.	<code>#Docker and Docker compose should now be deployed</code>
7.	<code>cd /opt/ && mkdir deploy && cd deploy</code>
8.	<code>#copy the docker docker-compose.yml for from the repository and the .env file. Modify values for username and passwords.</code>
9.	<code>set -a ; source .env</code>
10.	<code>docker-compose up -d</code>

11.	# ElasticSearch should be running
-----	-----------------------------------

*Note The .env file sets environment variables that are used when you run the docker-compose.yml configuration file. Ensure that you specify a strong password for the elastic and kibana_system users with the ELASTIC_PASSWORD and KIBANA_PASSWORD variables. These variables are referenced by the docker-compose.yml file.

52.7. Post-installation

The installation instructions should contain a check list that details

- Navigate to <http://{IP ADDRESS}>
- Login with the Credentials set in the .env file
- Check all functionality ElasticSearch conducts automatic prechecks as it loads.

52.8. Tasks Description – Wazuh Agent Installation

In this section, Wazuh dashboard is used to deploy Wazuh agents on the desired hosts for monitoring, allowing communication with the Wazuh server. As an example, a test server agent is deployed in the following instructions.

1. Go to Wazuh > Agents and click on “Deploy new agent”, refer to the screenshot below.
2. Specify the operating system of the host to be monitored.
3. The Wazuh server address refers to the internal IP address of the server where the agent will report back to.
4. Copy the generated script onto the host’s command line for installation and deployment.
5. Ensure that the Wazuh Manager IP address in file /var/ossec/etc/ossec.conf is correct on the selected host.
6. Start the agent using the given commands. *Note: this command is executed on the host’s command line.
7. Refresh the page and wait for the agent to fully connect.

53. Completed Deliverables

This trimester the project team was able to deliver two virtual machines.

The first is ‘opensiem’ that is able to handle the different applications running on it at the same time for the SIEM set up.

On this virtual machine Docker, Wazuh Dashboard and Elasticsearch are all installed. All these installations have been tested to make sure they run together without causing crashes or conflicting with one another.

The second virtual machine is called ‘opencti’, which has OpenCTI and Wazuh Agent and threat intel feed installed on the server.

These virtual machines can be found in the google cloud instance ‘sit-22t2-reddback-infra-612f86e’.

54. Roadmap

The following items can be archived for the SIEM project in later trimesters.

- *Develop a UML diagram*
- *See if there is way to get access to the GitHub to the cyber security and/or create a new repository if unable to do so. [GitHub - SssandMan/Cyber-Security: Redback Operations Cyber Security Repository](#)*
- *Identify the areas of concern that need to be monitored in security for Redback Operations environment.*
- *Create rules and playbooks that can be used for monitoring in the SIEM VM.*
- *Implement other data sources into the SIEM VM for monitoring*

55. Open Issues

There is no proper UML design for this solution and should be done next trimester to make sure that the project does not get off track.

56. Lessons Learned

Get access to Google Cloud to instance ‘sit-22t2-reddback-infra-612f86e’ early. Nominate someone in the project team to submit the google cloud form on behalf of everyone in the project.

57. Product Development Life Cycle

57.1. New Tasks

New tasks are created in the project team meetings when updates on the project/tasks are given by the individual team members. When tasks have been completed or issues have raised the team will discuss what needs to be done and adjust the Trello board and planning to reflect these discussions.

Once a new task has been agreed upon by the team the task will then go onto the Trello board as either backlog or in progress.

57.2. Definition of Done

A task can be marked as completed in a couple of ways depending on the nature of the task.

- *Documentation tasks are marked as completed when another team member has looked over the documentation and the documentation meet the following criteria.*
 - *Concise and legible*

- *Steps are logical and can be followed by someone just pick up the project for the first time*
 - *If everything is lost the documentation can be followed from start to finish to recreate everything*
- *Tasks that involving installing or implementing into the SIEM VM should be tested in a test SIEM VM so make sure that nothing that is being installed will cause conflicts with what has already been installed.*

57.3. Task Review

Task reviewing is a team effect and as such no one person is assigned to review tasks but instead it should rotate through the team, so everyone is reviewing. When a task is under review the team member reviewing should assign it to themselves and make note that the task had been reviewed. This can be a checklist item or in the description of the task.

57.4. Testing

Testing should be done in a test virtual machine that is copied from the production virtual machine. This will make sure that whatever is being tested is done so in an environment that is the same to production.

This is makes sure that we can notice any issues before production.

58. Product Architecture

58.1. Tech Stack

Google Cloud Project Instance

Docker

Wazuh

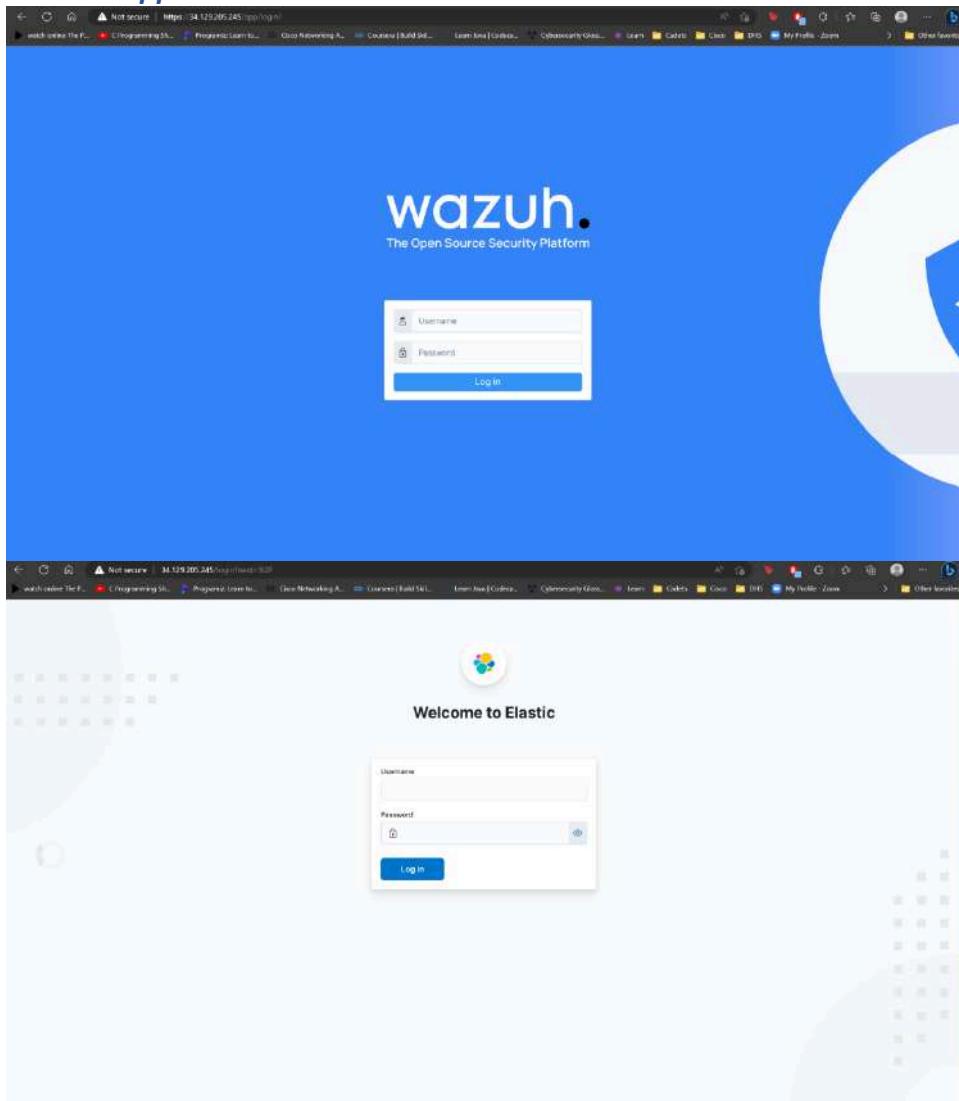
OpenCTI

ElasticSearch

59. Login Credentials

Every project team member has their own login details for google cloud. This is done with the Deakin username and the Deakin sync SSO.

60. Appendices



Project Handover Document

Redback Operations

Cyber Project 3 – Vulnerability Management

Trimester 1, 2023

Table of Contents

- **Project Information**
 - *Company Acting Director*
 - *Project Team*
- **Project Overview**
- **User Manual**
 - *Google Cloud Security Command Centre*
 - *SCC integration with SIEM (Wazuh)*
 - *SIEM integration*
 - *Vulnerability Code Scanner*
- **Completed Deliverables**
 - *System Requirements Document:*
 - *Vendor Analysis:*
 - *Google Cloud Security Command Centre:*
 - *SCC Integration with SIEM (Wazuh)*
 - *Vulnerability Code Scanner Engine*
- **Roadmap**
 - *Vulnerability Code Scanner Engine*
 - *Penetration Testing*
- **Open Issues**
 - *Google Cloud Security Command Centre:*
 - *SCC Integration with SIEM (Wazuh)*
- **Lessons Learned**
- **Product Development Life Cycle**
 - *New Tasks*
 - *Definition of Done*
 - *Task Review*
 - *Testing*
 - *Branching Strategy*
- **Product Architecture**
 - *UML Diagram*
 - *Tech Stack*
- **Source Code**
- **Login Credentials**
- **Other Relevant Information**
- **Appendices**

- *Project Information*
 - *Company Acting Director*

Kevin Lee
Director
kevin.lee@deakin.edu.au

- *Project Team*

Vulnerability Management
Redback Operation

Name:	Role:
NASIM EMADI	<i>Junior - Team Lead</i>
SANJAY MEDIKONDURU	<i>Junior - Team Member</i>
Harshana Thilanga	<i>Junior - Team Member</i>

Project Github Reposirotypes:

https://github.com/redbackoperations/Cyber-Security/tree/main/T1_2023/Project%203%20-%20Vulnerability%20Management

- *Project Overview*

Vulnerability management is one of the essential 8 recommended by the Australian government, and a key security control to track and monitor security vulnerabilities in information technology systems. A vulnerability is a weakness in an IT system that can be exploited by an attacker to deliver a successful attack.

A vulnerability management system commonly, discovers network-connected services and enumerates hard hardware, firmware/software of discovered resources and correlates with a database of known bugs and vulnerabilities (Vendor fed in Intelligence) to a reporting dashboard that provides visibility for service owners to remediate issues under their own process flow (e.g. install a patch) otherwise mitigate the risk (either accept/transfer the risk or consider a preventive control).

Here we initiated this project to provide minimum viable product with introducing 3 major functionalities:

Functionality 1 – Continuous vulnerability monitoring across Redback operation IT Systems.

This capability includes a periodic automated vulnerability scanner engine to discover Redback Operations' IT resources deployed in Google cloud Platform (GCP) and discover security issues and report findings enriched with CVSS (The Common Vulnerability Scoring System) rated format into a dashboard.

Functionality 2 – Integration with Security Incident and Event Management (SIEM) system.

Vulnerability Management security findings to be reported to SIEM as centralised security event management, So not VM findings can enrich correlation and analysis in SIEM, but also built-in SIEM incident workflow can be utilised to have vulnerability finding records converted into an actionable incident passed to service owners for remediation.

Functionality 3 – Code Scanner integration with CI/CD pipeline.

As one of maturity proactive measure in vulnerability management system, the project decided to investigate and provide work instructions, advisory and awareness on a static code scanner engine that supports majority of programming languages currently used in Redback Operations' code development CI/CD pipeline. The aim is a tool that provides analysis results for developers and discovers any security flaws in code development process.

- *User Manual*

- *Google Cloud Security Command Centre*

Google Cloud Security Command Center (SCC) is a comprehensive security management and data risk assessment dashboard offered by Google Cloud Platform (GCP). It provides centralized visibility into the security posture of your GCP resources, enabling you to monitor, detect, and respond to security threats and vulnerabilities. Here is an overview of the SCC dashboard and its important tabs:

Dashboard: The Dashboard tab provides an overview of your organization's security status, including the number of assets, security findings, and ongoing security sources.

Assets: The Assets tab displays a comprehensive inventory of your organization's GCP resources, such as virtual machines, databases, storage buckets, and more. You can view detailed information about each asset, including its configuration and associated security findings.

Findings: The Findings tab provides insights into security findings identified by SCC. These findings may include vulnerabilities, misconfigurations, and suspicious activities detected across your GCP resources. You can prioritize and investigate these findings for remediation.

Security Health Analytics: The Security Health Analytics tab focuses on providing proactive security insights and recommendations. It leverages machine learning and rule-based detection to identify potential security risks, misconfigurations, and compliance violations.

Security Marks: The Security Marks tab allows you to tag assets with custom labels for better organization and tracking. You can mark assets as favorites, assign severity levels, and add notes or comments for collaboration with your security team.

Security Sources: The Security Sources tab displays a list of security sources integrated with SCC. These sources include Google-native services like Cloud Security Scanner, Forseti Security, and partner solutions. You can configure and manage these sources to enable continuous monitoring and automated security assessments.

Regarding scanning configurations, Google Cloud SCC offers various scanning capabilities to enhance your security posture. These include:

Web Security Scanner: This scanner automatically detects common web application vulnerabilities by crawling and scanning your applications hosted on GCP. It checks for issues like cross-site scripting (XSS), SQL injection, and insecure direct object references.

Forseti Security: Forseti Security is an open-source security toolkit for GCP. It conducts continuous monitoring and analyzes your GCP resources for security vulnerabilities, misconfigurations, and policy violations. Forseti Security helps you enforce compliance with best practices and industry standards.

Cloud Security Scanner: Cloud Security Scanner is a web application vulnerability scanner provided by Google. It examines your application's public-facing endpoints and reports potential security risks, such as cross-site scripting (XSS) and mixed content vulnerabilities.

Partner Integration: Google Cloud SCC allows integration with various third-party security solutions to enhance scanning capabilities. These partners provide additional security assessments and insights tailored to specific requirements.

These scanning configurations help you detect and mitigate security risks across your GCP resources, ensuring a more robust and secure infrastructure.

Note: This system does not resolve or fix issues itself. It is just reporting system. It is up to the service owner to get a patch or remediation action applied to IT resources so next scanning will not have risk items reported.

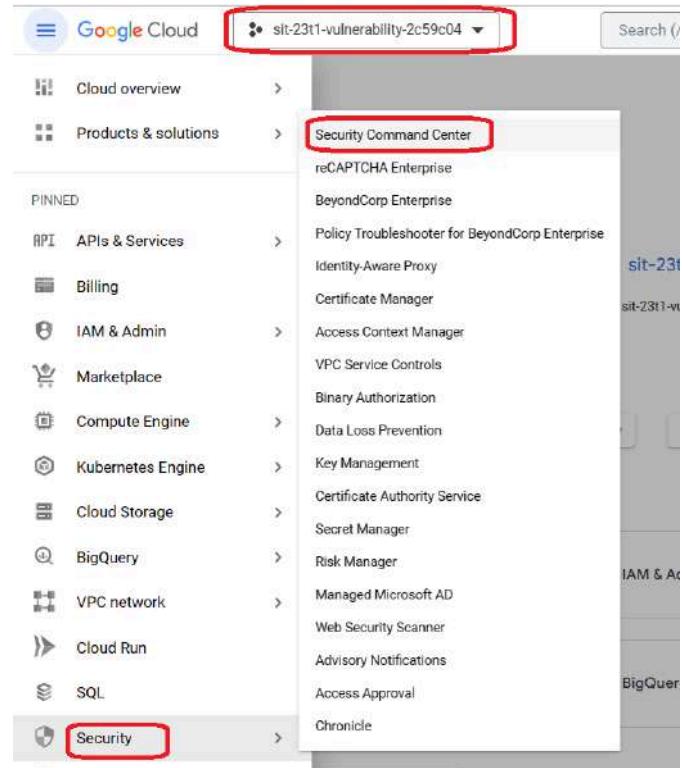
This system also does not need necessarily to have end-user data to work on. It is a tool which upon enablement, any system that gets provisioned in the future also will automatically be onboarded and monitored (though SCC service account needs access to new project so resources can be discovered).

- *SCC integration with SIEM (Wazuh)*

Google Cloud SCC module has a web interface that provides visibility over scanning scope and published results in dashboard.

First you need to get access to the Deakin project [sit-23t1-vulnerability-2c59c04](#)

Open SCC console:



The Overview tab, provides summary of findings for desired period.

OVERVIEW

VULNERABILITIES

ASSETS

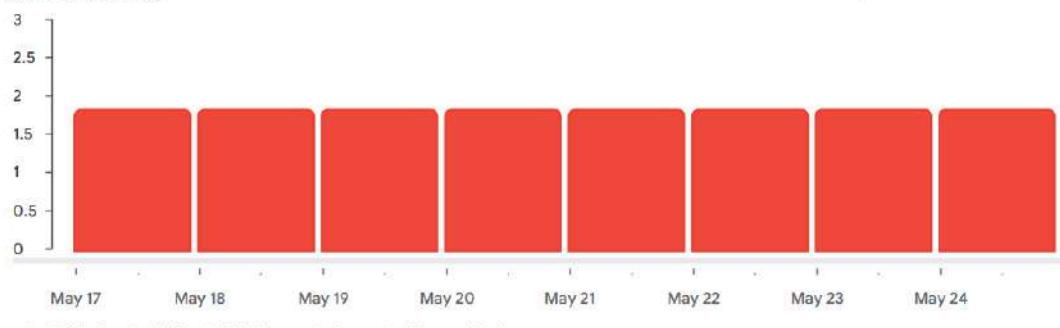
FINDINGS

SOURCES

EXPLORE

Active vulnerabilities over time by severity

2 active vulnerabilities



Active vulnerabilities

2 active vulnerabilities

FINDINGS BY CATEGORY

FINDINGS BY RESOURCE TYPE

FINDINGS BY PROJECT

Filter Categories

?

Vulnerabilities tab list findings against violated well known security standards.

OVERVIEW

VULNERABILITIES

ASSETS

FINDINGS

SOURCES

EXPLORE

Use Security Command Center's vulnerabilities dashboard to find potential weaknesses in your Google Cloud resources.

This page displays results only for projects in which Security Health Analytics, Rapid Vulnerability Detection, and Web Security Scanner are enabled. These services may not be enabled for all of your projects. Enable these services in settings.

[EDIT SETTINGS](#)[DISMISS](#)

Filter Vulnerabilities

Status	Last scanned	Category	Module ID	Recommendation	↓ Active findings	Standards
⚠	May 24, 2023 at 3:02:35 AM GMT+10	Open RDP port	OPEN_RDP_PORT	Firewall rules should not allow connections from all IP addresses on TCP or UDP port 3389	!! 1	CIS 1.0 : 3.7 CIS 1.1 : 3.7 CIS 1.2 : 3.7 CIS 1.3 : 3.7 PCI 1.2.1 NIST : SC-7 ISO A.13.1.1
⚠	May 24, 2023 at 3:02:33 AM GMT+10	Open SSH port	OPEN_SSH_PORT	Firewall rules should not allow connections from all IP addresses on TCP or SCTP port 22	!! 1	CIS 1.0 : 3.6 CIS 1.1 : 3.6 CIS 1.2 : 3.6 CIS 1.3 : 3.6 PCI 1.2.1 NIST : SC-7 ISO A.13.1.1
✓	May 24, 2023 at 3:00:56 AM GMT+10	Dataproc ima...	DATAPROC_IMAGE_OUTDATED	Dataproc clusters should not use images affected by Log4j vulnerability	!! 0	
✓	May 24, 2023 at 3:01:51 AM GMT+10	MFA not enfor...	MFA_NOT_ENFORCED	Multi-factor authentication should be enabled for all users in your org unit	!! 0	CIS 1.0 : 1.2 CIS 1.1 : 1.2 CIS 1.2 : 1.2 CIS 1.3 : 1.2 PCI 8.3 NIST : IA-2 ISO A.9.4.2
⚠	May 24, 2023 at 3:02:37 AM GMT+10	Non org IAM ...	NON_ORG_IAM_MEMBER	Corporate login credentials should be used instead of Gmail accounts	!! 0	CIS 1.0 : 1.1 CIS 1.1 : 1.1 CIS 1.2 : 1.1 CIS 1.3 : 1.1 PCI 7.1.2 NIST : AC-3 ISO A.9.2.3
✓	May 24, 2023 at 3:02:37 AM	Open discose...	OPEN_CISCOSECURE_WEBSM_PORT	Firewall rules should not allow connections from all IP	!! 0	PCI 1.2.1 NIST : SC-7 ISO A.13.1.1

Assets tab provides list of discovered IT resources that has been analysed so far.

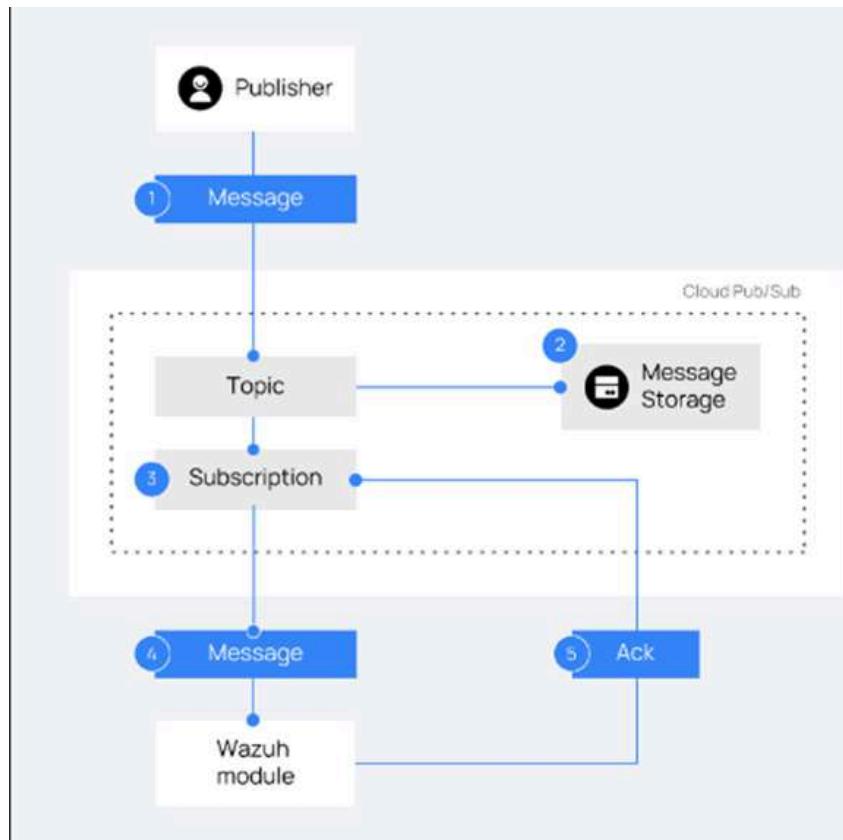
OVERVIEW		VULNERABILITIES	ASSETS	FINDINGS	SOURCES	EXPLORE		
Use Security Command Center's assets display to review your Google Cloud resources. Learn more								
View by		RESOURCE TYPE	PROJECT	ASSETS CHANGED		EXPORT		
Find resource type		No assets selected SET SECURITY MARKS						
Resource type ↑	Count	Filter Attributes, properties and marks						
All	1	<input type="checkbox"/> resourceProperties.name	securityCenterProperties.resourceType ↑	securityCenterProperties.resourceOwners	securityMarks.marks			
compute.Project	1	<input type="checkbox"/> sit-23t1-vulnerability-2c59c04	google.cloud.resourcemanager.Project	user.blac_adm@deakin.edu.au	-			
Firewall	4	<input type="checkbox"/> default-allow-internal	google.compute.Firewall	user.blac_adm@deakin.edu.au	-			
LogBucket1	2	<input type="checkbox"/> default-allow-icmp	google.compute.Firewall	user.blac_adm@deakin.edu.au	-			
LogSink	3	<input type="checkbox"/> default-allow-ssh	google.compute.Firewall	user.blac_adm@deakin.edu.au	-			
Network	1	<input type="checkbox"/> default-allow-rdp	google.compute.Firewall	user.blac_adm@deakin.edu.au	-			
resourcemanager.Project	1	<input type="checkbox"/> default	google.compute.Network	user.blac_adm@deakin.edu.au	-			
Route	39	<input type="checkbox"/> default-route-7bae5d25f8436998	google.compute.Route	user.blac_adm@deakin.edu.au	-			
ServiceAccount	2	<input type="checkbox"/> default-route-4d3932c5fa065c36	google.compute.Route	user.blac_adm@deakin.edu.au	-			
ServiceAccountKey	3	<input type="checkbox"/> default-route-452db41ed72a2b76	google.compute.Route	user.blac_adm@deakin.edu.au	-			
serviceusage.Service	18	<input type="checkbox"/> default-route-a90d6ed542531d1f	google.compute.Route	user.blac_adm@deakin.edu.au	-			
Subnetwork	37	<input type="checkbox"/> default-route-505adfe032574de	google.compute.Route	user.blac_adm@deakin.edu.au	-			
Subscription	1	<input type="checkbox"/> default-route-e1c2f872b1202999	google.compute.Route	user.blac_adm@deakin.edu.au	-			
Topic	2	<input type="checkbox"/> default-route-08d5d763e1895d00	google.compute.Route	user.blac_adm@deakin.edu.au	-			
		<input type="checkbox"/> default-route-66c15f0e58f03c9a	google.compute.Route	user.blac_adm@deakin.edu.au	-			

Findings tab provides detailed information about discovered vulnerabilities.

Security Command Center		GET PREMIUM		Edit Query		Time range
OVERVIEW		VULNERABILITIES	ASSETS	FINDINGS	SOURCES	EXPLORE
Query preview state="ACTIVE" AND NOT muted="MUTED"						
EDIT QUERY Last 30 days						
Quick filters	CLEAR ALL	I	Findings query results		SET SECURITY MARKS	MUTE OPTIONS
State			<input type="checkbox"/> Category	Severity	Event time ↓	Create time
<input type="checkbox"/> Show inactive			<input type="checkbox"/> Open RDP port	High	May 7, 2023, 9:45:32 PM	May 7, 2023, 9:45:32 PM
<input type="checkbox"/> Show muted						
Category			<input type="checkbox"/> Open SSH port	High	May 7, 2023, 9:45:31 PM	May 7, 2023, 9:45:32 PM
<input type="checkbox"/> Open RDP port	1					
<input type="checkbox"/> Open SSH port	1					
Finding class						
<input type="checkbox"/> Misconfiguration	2					
<input type="checkbox"/> Observation	0					
<input type="checkbox"/> SCC error	0					
<input type="checkbox"/> Threat	0					
<input type="checkbox"/> Vulnerability	0					
Project ID						

60.1. SIEM integration

The SIEM integration is a one time configuration has been done in SCC project. messages are published to Pub/Sub and with provided Credential key, SIEM team supposed to configure Wazuh (SIEM engine) to subscribe and fetch findings periodically.



Because SIEM project is not completely operational, Wazuh configuration has not completed yet, but whenever the configuration at SIEM is completed, the integration will be automatically completed. Eventually, the user interface for reported messages will be SIEM (Wazuh) portal which will be out of the scope for this project.

For work instruction about SIEM integration, refer to Project Github file:

https://github.com/redbackoperations/Cyber-Security/blob/main/T1_2023/Project%203%20-%20Vulnerability%20Management/Deployment/4-%20SCC%20Integration%20with%20SIEM/SCC_Wazuh_integration_work_instruction.docx

Trello Card: <https://trello.com/c/KT9HqG49>

- Vulnerability Code Scanner

Bandit is an open-source vulnerability scanning tool that is designed to find security issues in python. The way the tool does this is by processing each file, building an AST (Application Security Testing), and then runs the plugins against those AST nodes. Bandit then provides the user with a vulnerability report against the source code after scanning through all those files.

To install bandit, pip tool should be used. To download the tool, open up terminal and type the command:

Command: pip install bandit safety

To use the tool, firstly a python script must be written and saved for scanning. Then once the python file is saved, and ready to be scanned, go to the terminal, and use the bandit command:

Command: bandit <python file>

Bandit then generates a report with any possible security threats.

For details, refer to the user manual document in project Github

https://github.com/redbackoperations/Cyber-Security/blob/main/T1_2023/Project%203%20-%20Vulnerability%20Management/Deployment/5-%20VM%20Code%20Scanner/WI%20-%20Bandit%20-%20User%20Guide.docx

- *Completed Deliverables*

- *System Requirements Document:*

At week 1 project, we have reviewed Redback Operation handovers and extracted IT resources we should consider in project scope and product selection.

Here are identified system requirement of the project:

- As data is generated and stored in GCP, The database management in GCP needs to be monitored and be in the scope.
- There is infrastructure such as firewalls, DNS, Web service, and so on provisioned in GCP which also need to be monitored and scanned periodically for vulnerabilities.
- There are applications and codes developed or under development in which VM is good to have a scanning tool for code analysis testing before code deployments and to be available for integration into CICD pipeline.
- Reported dashboard should be able to be visible for service owners as well as periodic reports or incidents generated to communicate discovered vulnerabilities with service owners.

For details of identified resources and project requirements, refer to project Github file:

https://github.com/redbackoperations/Cyber-Security/blob/main/T1_2023/Project%203%20-%20Vulnerability%20Management/Requirement/VM%20System%20Requirement.docx

- *Vendor Analysis:*

We analysed 3 vendors:

- Google Cloud "Security Command Centre" as Deakin cloud provide native VM system.
- 2 other industry known vendors for VM systems: "Tenable" and "Rapid 7".

we did side by side product comparison for different features and capabilities, then decided to proceed with Google Cloud "Security Command Centre" based on following facts:

- *SCC module in GCP provides not only Vulnerability Management capability but also it provides Threat Detection, Compliance Reporting, and advanced Vulnerability scanning on endpoints. However, the Vulnerability management basic functionality is available free out-of-the-Box under in the Standard tier.*
- *SCC is a built-in module so it provides the most forward compatibility with new IT resources Redback project is currently deploying or will deploy in GCP.*
- *SCC also seems to have better compatibility with any SIEM product which currently has a live project to deploy into GCP. We hope we integrate SCC with SIEM so it can utilise SIEM workflows to raise an incident for SCC findings.*
- *We believe free module SCC deployment is essential and it can always extend to advanced features like remote endpoint scanning or penetration testing features if needed in the future.*

Refer to Project Github folder for vendor analysis documents:

https://github.com/redbackoperations/Cyber-Security/tree/main/T1_2023/Project%203%20-%20Vulnerability%20Management/Vendor%20Selection

Trello Board Cards:

SCC: <https://trello.com/c/zqoMluGZ>

Rapid 7: <https://trello.com/c/ruYWUQRM>

Tenable: <https://trello.com/c/JZd59UEm>

Vendor Analysis Report: <https://trello.com/c/MaWDqYpM>

- *Google Cloud Security Command Centre:*

Google Cloud Security Command Center (SCC) offers two tiers: Standard tier and Premium tier. The selection of the tier depends on the specific needs and requirements of your organization. Here is an explanation of each tier and the reasons for selecting the Standard tier:

Standard tier: The Standard tier of Google Cloud SCC provides essential security monitoring and management capabilities. It offers the following features:

a. Asset inventory: The Standard tier enables you to gain visibility into your organization's assets deployed on Google Cloud Platform (GCP). It provides an inventory of your virtual machines, databases, storage buckets, and other resources.

b. Security findings: This tier helps you identify and track security findings such as vulnerabilities, misconfigurations, and suspicious activities across your GCP resources.

c. Security health analytics: The Standard tier includes security health analytics, which provides proactive insights and recommendations to improve your security posture. It leverages machine learning algorithms and rule-based detection to identify potential risks and compliance violations.

d. Integration with security sources: Standard tier allows you to integrate with various security sources, including Google-native services and partner solutions, to enhance your security capabilities. These sources can provide additional scanning and monitoring functionalities.

Reasons for selecting the Standard tier:

i. Basic security monitoring: The Standard tier is suitable for organizations that require fundamental security monitoring and management capabilities without advanced features. It provides visibility into assets and security findings, helping you maintain a basic level of security.

ii. Cost-effective: The Standard tier is typically less expensive compared to the Premium tier. If your organization has budget constraints or does not require advanced security features, the Standard tier can be a cost-effective option.

iii. Scalability: The Standard tier is designed to scale with your organization's needs. It can accommodate growing resource inventories and security findings without significant performance impacts.

- [*SCC Integration with SIEM \(Wazuh\)*](#)

SCC integration configuration in VM project is completed with creation of Pub/Sub as well as service account credential to be used by SIEM as subscriber.

SIEM integration document is prepared with details of every steps required along with screenshots and contains:

- Generating GCP service account required for Publishing SCC findings.
- Configuring Pub/Sub in GCP SCC project
 - Create Topic.
 - Create Pub/Sub.
 - Create Sink (log Router)
- Wazuh Configurations.
 - Installing pre-requisite applications for gcp-pubsub.

- Adding Subscription to Wazuh config.
- Sample of expected result

The document contains all work instructions project team as well as Deakin cloud management team (for access provision) and SIEM team (sibling project as subscriber) needed to be done.

Refer to Github repo for work instruction file:

https://github.com/redbackoperations/Cyber-Security/blob/main/T1_2023/Project%203%20-%20Vulnerability%20Management/Deployment/4-%20SCC%20Integration%20with%20SIEM/SCC_Wazuh_integration_work_instruction.docx

SIEM integration team engagement

https://github.com/redbackoperations/Cyber-Security/blob/main/T1_2023/Project%203%20-%20Vulnerability%20Management/Deployment/4-%20SCC%20Integration%20with%20SIEM/SIEM%20team%20-%20engagement%20.msg

Trello Board Card: <https://trello.com/c/KT9HqG49>

- Vulnerability Code Scanner Engine

User Manual: https://github.com/redbackoperations/Cyber-Security/blob/main/T1_2023/Project%203%20-%20Vulnerability%20Management/Deployment/5-%20VM%20Code%20Scanner/WI%20-%20Bandit%20-%20User%20Guide.docx

Test cases:

https://github.com/redbackoperations/Cyber-Security/blob/main/T1_2023/Project%203%20-%20Vulnerability%20Management/Deployment/5-%20VM%20Code%20Scanner/Bandit-Code%20Scanner-Test%20casess%20results.docx

- Roadmap

- Vulnerability Code Scanner Engine

The Vulnerability code scanner engine Bandit has to be deployed locally and it needs to be an integral part of the developing team. The plan for the integration of Bandit is to create an awareness program where it can be integrated with the development team and the tools

they use. The program would consist of how to use the tool, how to read the report generated by the tool, and why the tool is important.

- *Penetration Testing*

As an enhancement for this project, we suggest adding an automated pen testing scanner to be integrated with CI/CD pipeline for web development verification testings. It can also be used for periodic pen testing for applications in production state.

- *Open Issues*

- *Google Cloud Security Command Centre:*

Nasim Comment: Google Cloud SCC service account in vulnerability management project requires access at enterprise level to all projects otherwise to number of target projects which this solution should be able to scan resources in those projects and report the findings. Without this cross project privilege, the scanner does not have permission to scan other projects. This privilege is pending approval by Redback GM, so the cloud team can grant the cross projects privilege.

Refer to corresponding in Github for further details:

https://github.com/redbackoperations/Cyber-Security/blob/main/T1_2023/Project%203%20-%20Vulnerability%20Management/Deployment/3-%20SCC%20Discovery/Requesting%20access%20security%20health%20check%20cross%20all%20Redback%20projects..msg

- *SCC Integration with SIEM (Wazuh)*

Although VM project has completed the configurations, but end to end application testing is pending other Cyber project, SIEM to be operational and config their Wazuh module as per provided work instruction and subscription service account key.

So, we have listed this item to remind the team to follow up with sibling project for testing results.

SIEM integration team engagement

https://github.com/redbackoperations/Cyber-Security/blob/main/T1_2023/Project%203%20-%20Vulnerability%20Management/Deployment/4-%20SCC%20Integration%20with%20SIEM/SIEM%20team%20-%20engagement%20.msg

Trello Board Card: <https://trello.com/c/KT9HqG49>

- *Lessons Learned*

This project was an effective opportunity to work in a real project and contribute to cyber security controls Redback operation needed.

We did learn a lot how to do vendor analysis and consider different parameters and most importantly to keep the cost in first place when we have tight budget in such projects.

Moreover, we learnt how to collectively work on different part of the project through collaborate and help each other and keep focused on overall view of project and deliverables.

We also experienced a mistake where accidentally uploaded one of service account keys to Github and deleted it after 5 minutes, however got notified from Github later on as well as Digital Services team following day, urged us to remove the key from gitgub commit history and rotate the key. The resolution action is completed and replied to Digital Team as following:

https://github.com/redbackoperations/Cyber-Security/blob/main/T1_2023/Project%203%20-%20Vulnerability%20Management/Deployment/4-%20SCC%20Integration%20with%20SIEM/Service%20Account%20rotated%20after%20Creation%20file%20leaked.msg

So, we got an important lesson how a simple mistake can have massive consequences. We also got a technical learning, how to remove a git revision from history as well as service account key rotation in GCP.

- *Product Development Life Cycle*

- *New Tasks*

We utilised team formation and brain storming methods to individually investigate about our weekly meeting agenda (when new subject was in the list) and then brain stormed about the agenda and discussed about action plan. Then we used to create task(s) in Trello board to describe what needs to be done, the time frame and agree on assigned resources.

We used scrum meeting and project management style and formed our project Github and stages starting with Project scope definition (system requirement collection), product selection, deployment, and handover.

- *Definition of Done*

We marked Trello board as closed where described activity was completed with documented outcome and in our weekly scrum meeting review, all team members were happy to close it.

As part of the closure, we also were keeping our Github repository and respected Trello card updated with artifacts.

- *Task Review*

In our weekly Scrum meeting, not only we were discussing what needs to be done in next week, but also we had follow up discussion about tasks were ready to be closed. The assignee commonly were presenting an overview and answered any questions other team members had. Then we were get the task marked as done, if all team members agreed and happy by task review results.

- *Testing*

We had quite simple testing as we deployed available tools in our projects, however for some instances like code scanner, we did number of use case testings and fuzzy testing with different codes our development team use that language for coding.

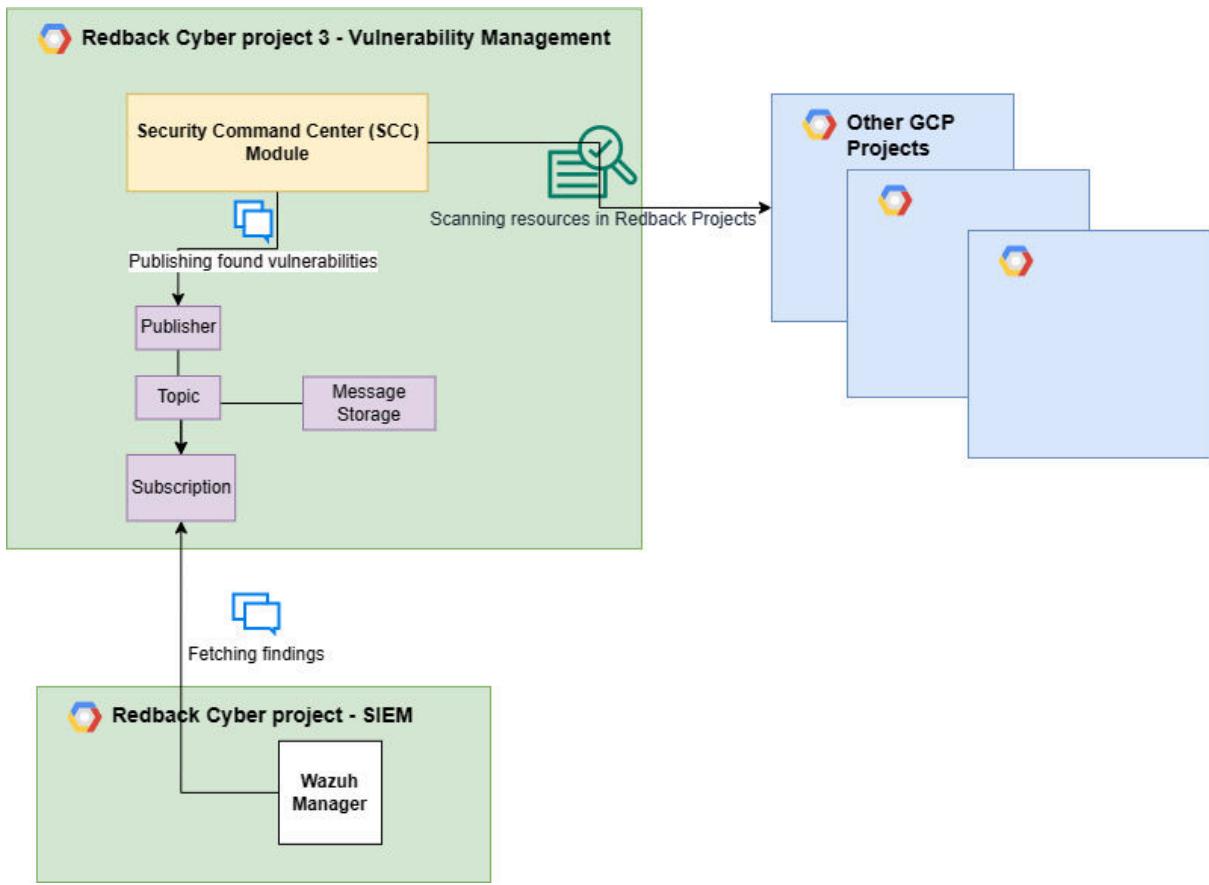
We also has a Proof Of Concept (POC) test environment created in our personal GCP account and to model limitations we had in Deakin cloud tenant with different projects and segregation of projects. This testing environment was required to provide work instruction and make sure the solution works in Redback project GP tenant.

- *Branching Strategy*

Our project focus is not code development and most usecase for Github repo is for document control. So we have one main branch which we were merging the files directly to main. As soon as documents or files were ready and attached the Trello board card, Our team lead had the file uploaded to Github repo.

- *Product Architecture*

- *UML Diagram*



- *Tech Stack*

List all of the software and hardware utilised in this project. For each tool, give a short description and explain why it was chosen.

Bandit: Proposed as code vulnerability scanner which has most compatibility with developers coding language: Python

Python: Worked when testing Bandit.

Git Bash: Used for interacting with Github and upload files and couments of the project to Cyber Security Repo.

MS Visio: Drawing architectural diagrams.

GCP Console: Configuring SCC module and integration with SIEM.

- *Source Code*

Project Github Repository:

https://github.com/redbackoperations/Cyber-Security/tree/main/T1_2023/Project%203%20-%20Vulnerability%20Management

- *Login Credentials*

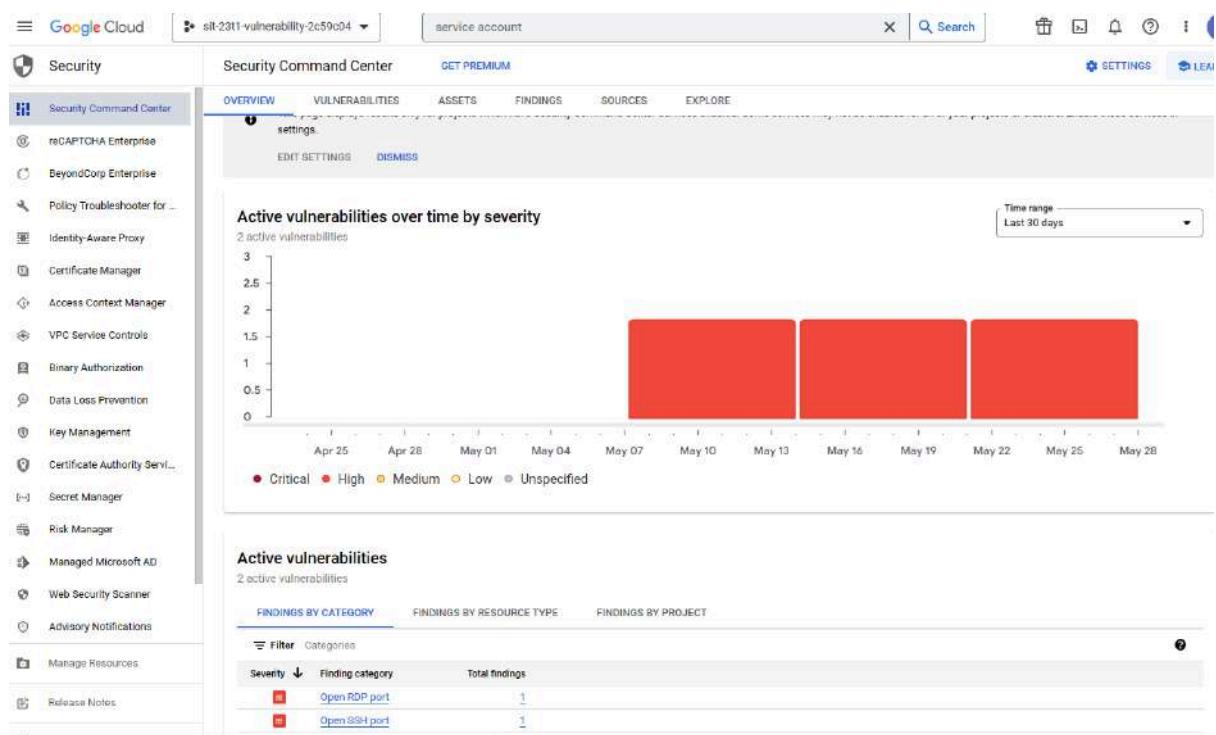
No login credential is used in our project.

Though in GCP IAM, there is a service account for SCC Pub/Sub to be used by SIEM when integrates and fetch findings.

- *Other Relevant Information*

- *Appendices*

Google Cloud Security Command Centre (SCC):



Google Cloud | sit:23t1-vulnerability-2c59c04 | service account | Search | X

Security Command Center

OVERVIEW VULNERABILITIES ASSETS FINDINGS SOURCES EXPLORE

This page displays results only for projects in which Security Health Analytics, Rapid Vulnerability Detection, and Web Security Scanner are enabled. These services may not be enabled for all of your projects. Enable these services in settings.

EDIT SETTINGS **DISMISS**

Status	Last scanned	Category	Module ID	Recommendation	Active findings	Standards
⚠️	May 27, 2023 at 11:28:14 PM GMT+10	Open RDP port	OPEN_RDP_PORT	Firewall rules should not allow connections from all IP addresses on TCP or UDP port 3389	1	CIS 1.0.3.7 CIS 1.1.3.7 CIS 1.2.3.7 CIS 1.3.3.7 PCI 1.2.1 NIST SC-7 ISO A.13.1.1
⚠️	May 27, 2023 at 11:29:05 PM GMT+10	Open SSH port	OPEN_SSH_PORT	Firewall rules should not allow connections from all IP addresses on TCP or SCTP port 22	1	CIS 1.0.3.6 CIS 1.1.3.6 CIS 1.2.3.6 CIS 1.3.3.6 PCI 1.2.1 NIST SC-7 ISO A.13.1.1
🟢	May 27, 2023 at 11:27:09 PM GMT+10	Dataproc image	DATAPROC_IMAGE_OUTDATED	Dataproc clusters should not use images affected by Log4j vulnerability	0	
🟢	May 27, 2023 at 11:28:08 PM GMT+10	MFA not enforced	MFA_NOT_ENFORCED	Multi-factor authentication should be enabled for all users in your org unit	0	CIS 1.0.1.2 CIS 1.1.1.2 CIS 1.2.1.2 CIS 1.3.1.2 PCI 8.3 NIST IA-2 ISO A.9.4.2
🟡	May 27, 2023 at 11:28:45 PM GMT+10	Non org IAM member	NON_ORG_IAM_MEMBER	Corporate login credentials should be used instead of Gmail accounts	0	CIS 1.0.1.1 CIS 1.1.1.1 CIS 1.2.1.1 CIS 1.3.1.1 PCI 7.1.2 NIST AC-9 ISO A.9.2.3

Google Cloud | sit:23t1-vulnerability-2c59c04 | service account | Search | X

Security Command Center

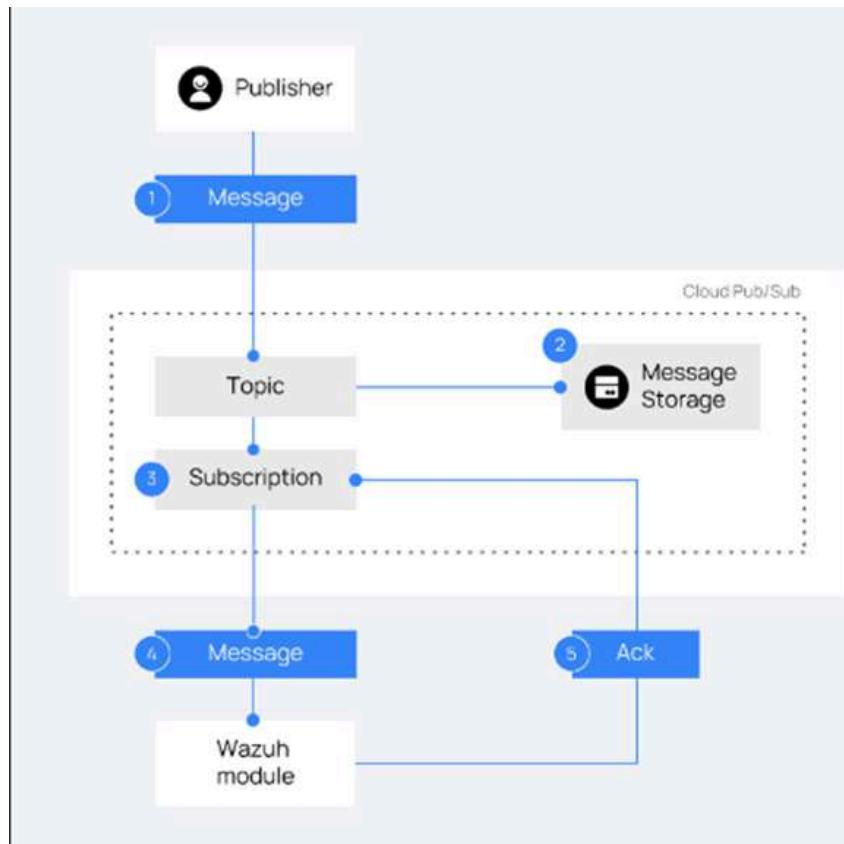
OVERVIEW VULNERABILITIES ASSETS FINDINGS SOURCES EXPLORE

Query preview
state="ACTIVE" AND NOT muted="MUTED"

EDIT QUERY **Time range** Last 30 days

Quick filters		Findings query results																													
		<input type="checkbox"/> SET SECURITY MARKS <input type="checkbox"/> MUTE OPTIONS <input type="checkbox"/> EXPORT <input type="checkbox"/> COLUMNS																													
<input type="checkbox"/> State <input type="checkbox"/> Show inactive <input type="checkbox"/> Show muted		<table border="1"> <thead> <tr> <th>Category</th> <th>Severity</th> <th>Event time</th> <th>Create time</th> <th>Resource display name</th> <th>Resource full name</th> </tr> </thead> <tbody> <tr> <td>Account has leaked credentials</td> <td>Critical</td> <td>May 25, 2023, 12:32:35 AM</td> <td>May 25, 2023, 12:32:35 AM</td> <td>sit:23t1-vulnerability-2c59c04</td> <td>/cloudresourc...ects/558552435</td> </tr> <tr> <td>Open RDP port</td> <td>High</td> <td>May 7, 2023, 9:45:32 PM</td> <td>May 7, 2023, 9:45:32 PM</td> <td>default-allow-rdp</td> <td>/compute/goo...vulnerability-2c59c04/global/020</td> </tr> <tr> <td>Open SSH port</td> <td>High</td> <td>May 7, 2023, 9:45:31 PM</td> <td>May 7, 2023, 9:45:32 PM</td> <td>default-allow-ssh</td> <td>/compute/goo...vulnerability-2c59c04/global/668</td> </tr> </tbody> </table>						Category	Severity	Event time	Create time	Resource display name	Resource full name	Account has leaked credentials	Critical	May 25, 2023, 12:32:35 AM	May 25, 2023, 12:32:35 AM	sit:23t1-vulnerability-2c59c04	/cloudresourc...ects/558552435	Open RDP port	High	May 7, 2023, 9:45:32 PM	May 7, 2023, 9:45:32 PM	default-allow-rdp	/compute/goo...vulnerability-2c59c04/global/020	Open SSH port	High	May 7, 2023, 9:45:31 PM	May 7, 2023, 9:45:32 PM	default-allow-ssh	/compute/goo...vulnerability-2c59c04/global/668
Category	Severity	Event time	Create time	Resource display name	Resource full name																										
Account has leaked credentials	Critical	May 25, 2023, 12:32:35 AM	May 25, 2023, 12:32:35 AM	sit:23t1-vulnerability-2c59c04	/cloudresourc...ects/558552435																										
Open RDP port	High	May 7, 2023, 9:45:32 PM	May 7, 2023, 9:45:32 PM	default-allow-rdp	/compute/goo...vulnerability-2c59c04/global/020																										
Open SSH port	High	May 7, 2023, 9:45:31 PM	May 7, 2023, 9:45:32 PM	default-allow-ssh	/compute/goo...vulnerability-2c59c04/global/668																										
<input type="checkbox"/> Category <input type="checkbox"/> Account has leaked credentials <input type="checkbox"/> Open RDP port <input type="checkbox"/> Open SSH port																															
<input type="checkbox"/> Finding class <input type="checkbox"/> Misconfiguration <input type="checkbox"/> Threat <input type="checkbox"/> Observation <input type="checkbox"/> SCC error <input type="checkbox"/> Vulnerability																															
<input type="checkbox"/> Project ID <input type="checkbox"/> sit:23t1-vulnerability-2c59c04																															

SCC integration with SIEM



Bandit Code Scanner:

```
PS C:\Users\sanjay.medikonduru\Desktop\Security> bandit test.py
[main] INFO profile include tests: None
>> Issue: [B105:hardcoded_password_string] Possible hardcoded password: 'fn94uhfjhfn9fgb9b430'
Severity: Low Confidence: Medium
CWE: CWE-259 (https://cwe.mitre.org/data/definitions/259.html)
More Info: https://bandit.readthedocs.io/en/1.7.5/plugins/b105\_hardcoded\_password\_string.html
Location: test.py:1:8
1     token = "fn94uhfjhfn9fgb9b430"
2
3     temp_dir = "/tmp"

----->> Issue: [B108:hardcoded_tmp_directory] Probable insecure usage of temp file/directory.
Severity: Medium Confidence: Medium
CWE: CWE-377 (https://cwe.mitre.org/data/definitions/377.html)
More Info: https://bandit.readthedocs.io/en/1.7.5/plugins/b108\_hardcoded\_tmp\_directory.html
Location: test.py:3:11
2
3     temp_dir = "/tmp"

-----
Code scanned:
    Total lines of code: 2
    Total lines skipped (#nosec): 0

Run metrics:
    Total issues (by severity):
        Undefined: 0
        Low: 1
        Medium: 1
        High: 0
    Total issues (by confidence):
        Undefined: 0
        Low: 0
        Medium: 2
        High: 0
Files skipped (0):
```

Project Handover Document

*Company ----Redback Operations
Project -----VR Sun cycle*

Trimester 1, 2023

1. Project information	154
1.1 Company Acting Director	154
1.2. Project Team	154
1.3. Members	154
2. Project Overview	155
3. User Manual.....	155
3.1. Microsoft Teams	155
3.1.1. Communication.....	155
3.1.2. File Storage	155
3.2. Trello.....	157
3.3. Unity (Editor Version: 2020.3.37f1)	162
3.3.1 Installation and setup.....	162
3.3.2. Basics of Unity.....	165
3.3.3 Build settings.....	168
3.3.4. Using VR Headset	169
3.4. GitHub	171
3.4.1. Setup Github	171
3.4.2. Using GitHub with Unity.....	180
3.5. Blender / Maya / 3D Max	186
3.5.1 Installation and setup 3DMAX	186
4. Deliverables.....	189
5. Open Issues	189
6. Lessons Learned	190
7. Product Development Life Cycle	191
1. New Tasks.....	191
2. Definition of Done	191
3. Task Review.....	192
4. Testing	192
5. Branching Strategy	192
8. Product Architecture	192
1. UML Diagram.....	192
2. Tech Stack.....	193
9. Appendices.....	194
9.1 Showcase Video	194
9.2 Documentation Deliverables	194
9.3 unity Deliverables.....	194
10. Content Contribution.....	199

1. Project information

60.2. 1.1 Company Acting Director

*Kevin Lee
Director
kevin.lee@deakin.edu.au*

60.3. 1.2. Project Team

*Company: Redback Operations
Team: Virtual Reality
Channel: VR modelling and World Building Hub
Project: Sun Cycle*

60.4. 1.3. Members

Name	Senior/Junior/Position	Role
<i>Daniel Isla</i>	<i>Junior - Regular Member</i>	<i>Game Engine/Programming</i>
<i>Hilal Irshad</i>	<i>Junior - Team Lead</i>	<i>Game Engine/Programming</i>
<i>Jiahao Feng</i>	<i>Senior - Regular Member</i>	<i>Design/Asset Creation</i>
<i>Nathan James Fairchild</i>	<i>Junior - Regular Member</i>	<i>Design/Asset Creation / Game Engine/Programming</i>
<i>Steven Makris</i>	<i>Junior - Regular Member</i>	<i>Design/Asset Creation / Game Engine/Programming</i>

2. Project Overview

Project Sun Cycle is a Virtual Reality based game which is played in combination with an exercise bike. In this single player experience the player is able to explore and travel across the post-apocalyptic tribal world completing quest such as delivering goods between the different tribes on their hover skater-bike. The aim of this project is to create a fun and immersive experience that will encourage people to exercise more with the increased motivation of being able to play a video game simultaneously.

An overall goal of incorporating enjoyable yet challenging quests within the game will help test the endurance/strength of the players and keep them exercising on a regular basis. Another aspect of this is incorporating a beautiful in game environment to visually impress the player as they travel around the game world.

In conclusion as stated above this project aims to solve the problem of low motivation to exercise by gamifying an otherwise boring workout.

3. User Manual

60.5. 3.1. Microsoft Teams

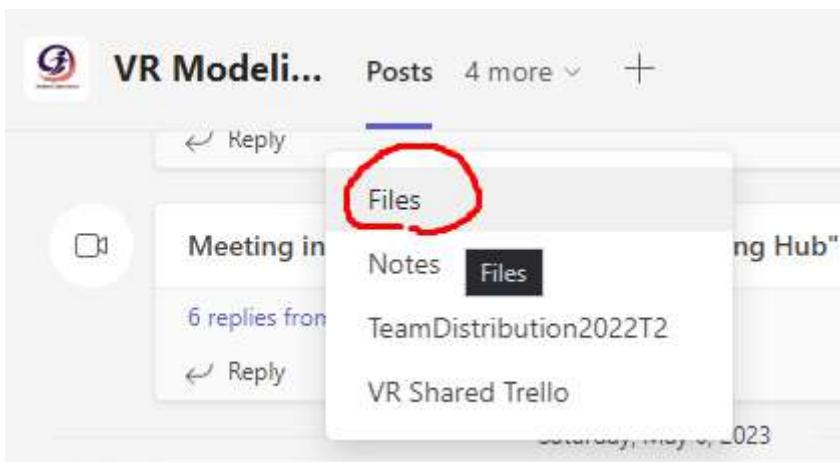
Once added to the Redback Operations Company use the “VR Modelling and World Building Hub” public channel for communication and file storage.

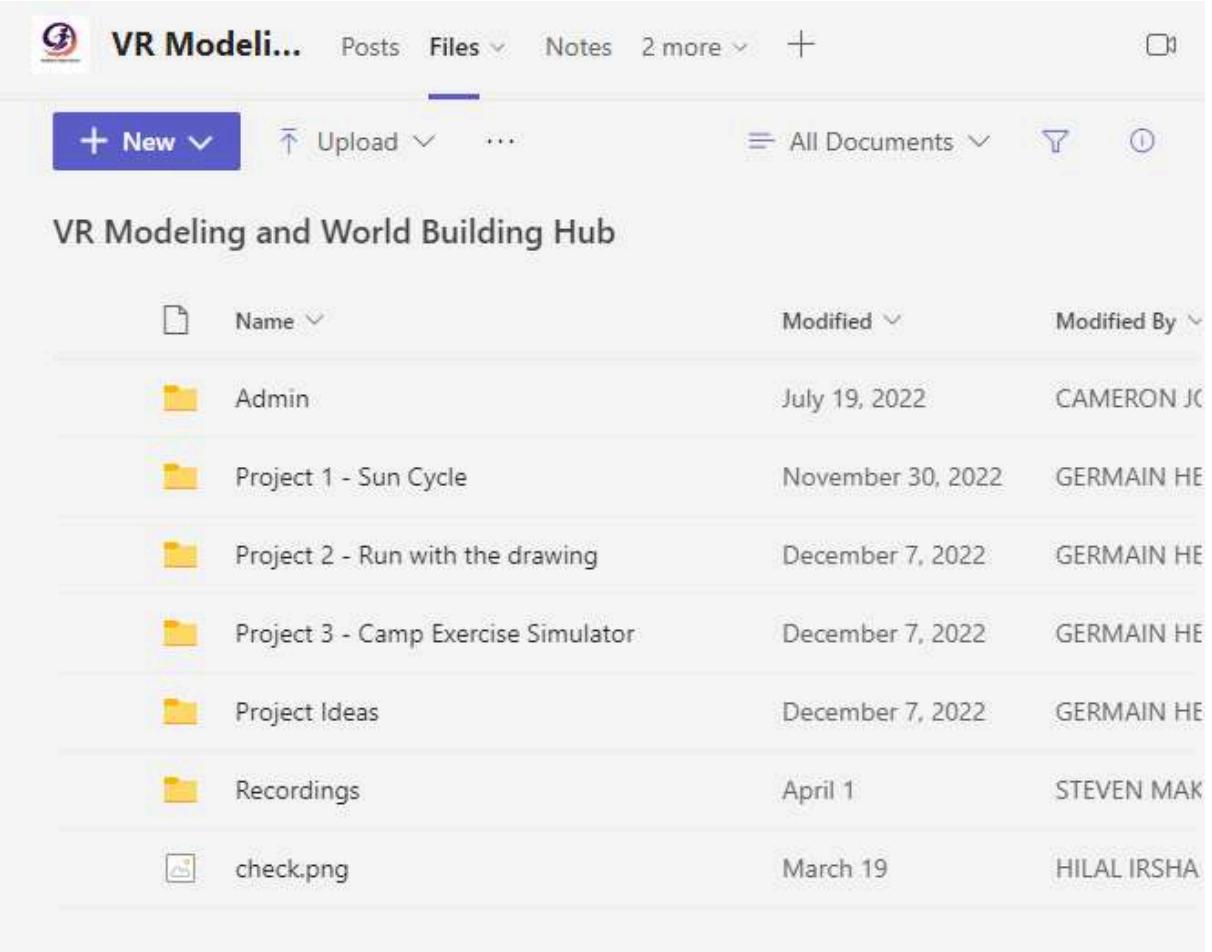
60.6. 3.1.1. Communication

Communicate with your team using posts, personal messages and making sure to engage in scheduled meetings to have discussions about the project.

60.7. 3.1.2. File Storage

Store any contributes, ideas and important files within the files section of the channel. Make sure to organize your content by categorizing it into its respective folder.





The screenshot shows a digital workspace titled "VR Modeling and World Building Hub". At the top, there are navigation links for "Posts", "Files", "Notes", and "2 more". Below the header are buttons for "+ New", "Upload", and "All Documents". The main area displays a list of files and folders:

	Name	Modified	Modified By
📁	Admin	July 19, 2022	CAMERON JO
📁	Project 1 - Sun Cycle	November 30, 2022	GERMAIN HE
📁	Project 2 - Run with the drawing	December 7, 2022	GERMAIN HE
📁	Project 3 - Camp Exercise Simulator	December 7, 2022	GERMAIN HE
📁	Project Ideas	December 7, 2022	GERMAIN HE
📁	Recordings	April 1	STEVEN MAK
🖼️	check.png	March 19	HILAL IRSHA

Opening the file "Project - Sun Cycle"

VR Modeling and World Building Hub > Project 1 - Sun Cycle



This screenshot shows the contents of the "Project 1 - Sun Cycle" folder. The list includes various sub-folders and files:

	Name	Modified	Modified By
📁	2022 T1	December 7, 2022	GERMAIN HE
📁	2022 T2	December 7, 2022	GERMAIN HE
📁	2022 T3	November 30, 2022	GERMAIN HE
📁	2023 T1	March 18	HILAL IRSHA
📁	Archived Files from Private Channels Copie...	December 14, 2022	GERMAIN HE
📁	MISC images	December 7, 2022	GERMAIN HE
📁	Unity solutions to problems	December 13, 2022	GERMAIN HE

Files to the project on Teams (This are individual files and will not be shared with anyone If you make any changes to them):

VR Modeling and World Building Hub > Project 1 - Sun Cycle > 2023 T1 > Project				
	Name	Modified	Modified By	+ Add column
○	Unity.zip	A few seconds ago	STEVEN MAKRIS	
○	Usefull Files.zip	29 minutes ago	STEVEN MAKRIS	

60.8. 3.2. Trello

We use Trello to organise the tasks that need to be completed, the status of said tasks and who is assigned to each of the tasks.

Trello: <https://trello.com/b/aXyBMyGf/project-sun-cycle>

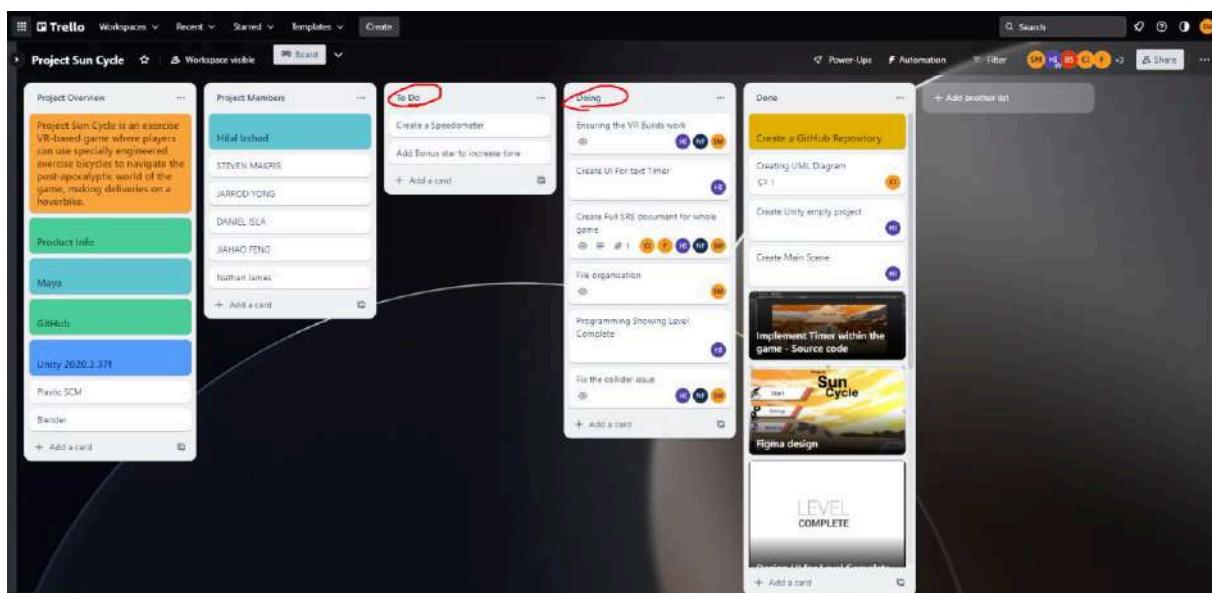
Overview

This image demonstrates how we categorize our cards (tasks) into their respective statuses such as “To Do”, “Doing” and “Done” which contains tasks from the roadmap (past trimesters) that we could decide to work on. To move these cards to a different list you can just click and drag them.

To do: add a trello image of our overall board

New Tasks

1. When you want to create a new task, you can click the “Add a card” button under each list.



2. You can then add a task title to that card.

Enter a title for this card...

Add card



...

3. Once you have a new card you can select that card to go into the details and add a description to better define the workload of the task.

Ensuring the VR Builds work
in list Doing

Members: HI, NF, SM

Notifications: Watching

Description: Add a more detailed description...

New Lists

1. If you want to add a new list to help categorize your cards, click "Add another list".

+ Add another list

2. Once you have clicked that enter the name of list you want and then click "Add list".

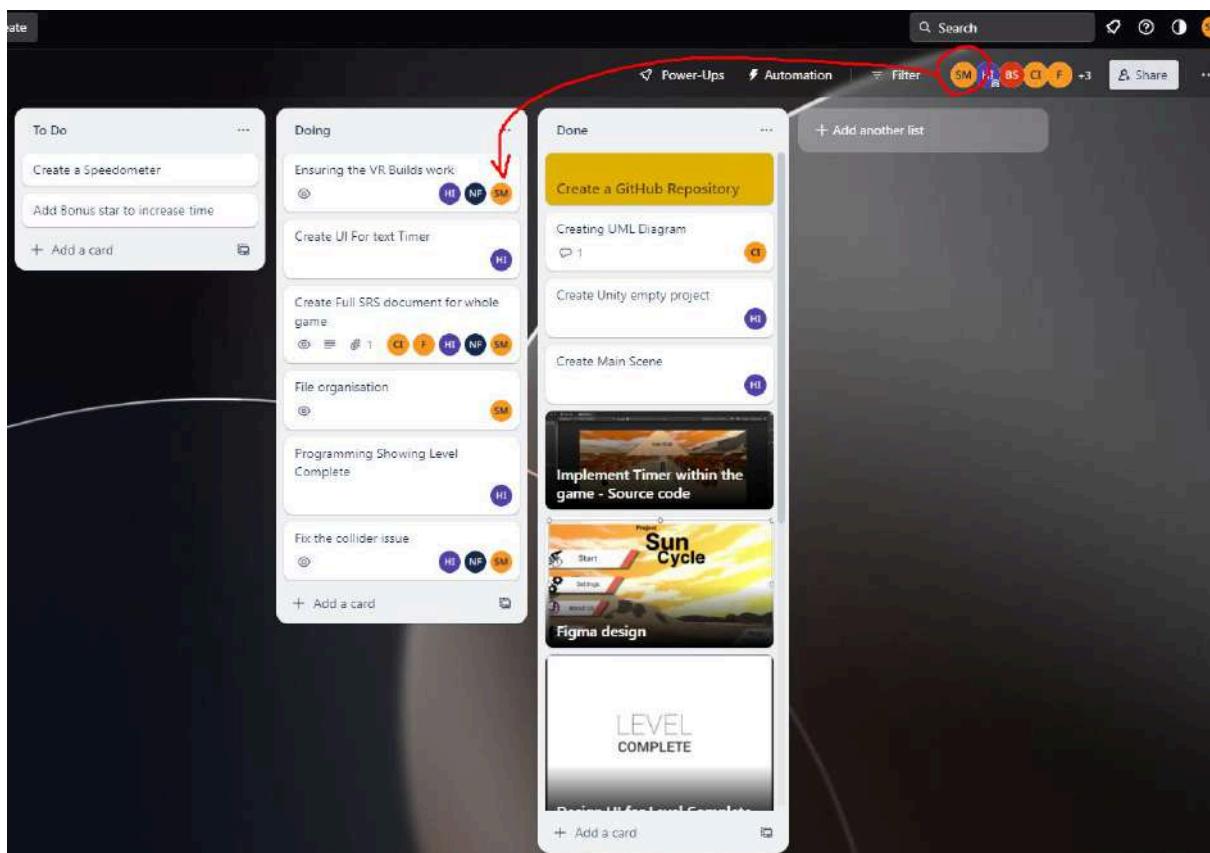
Enter list title...

Add list



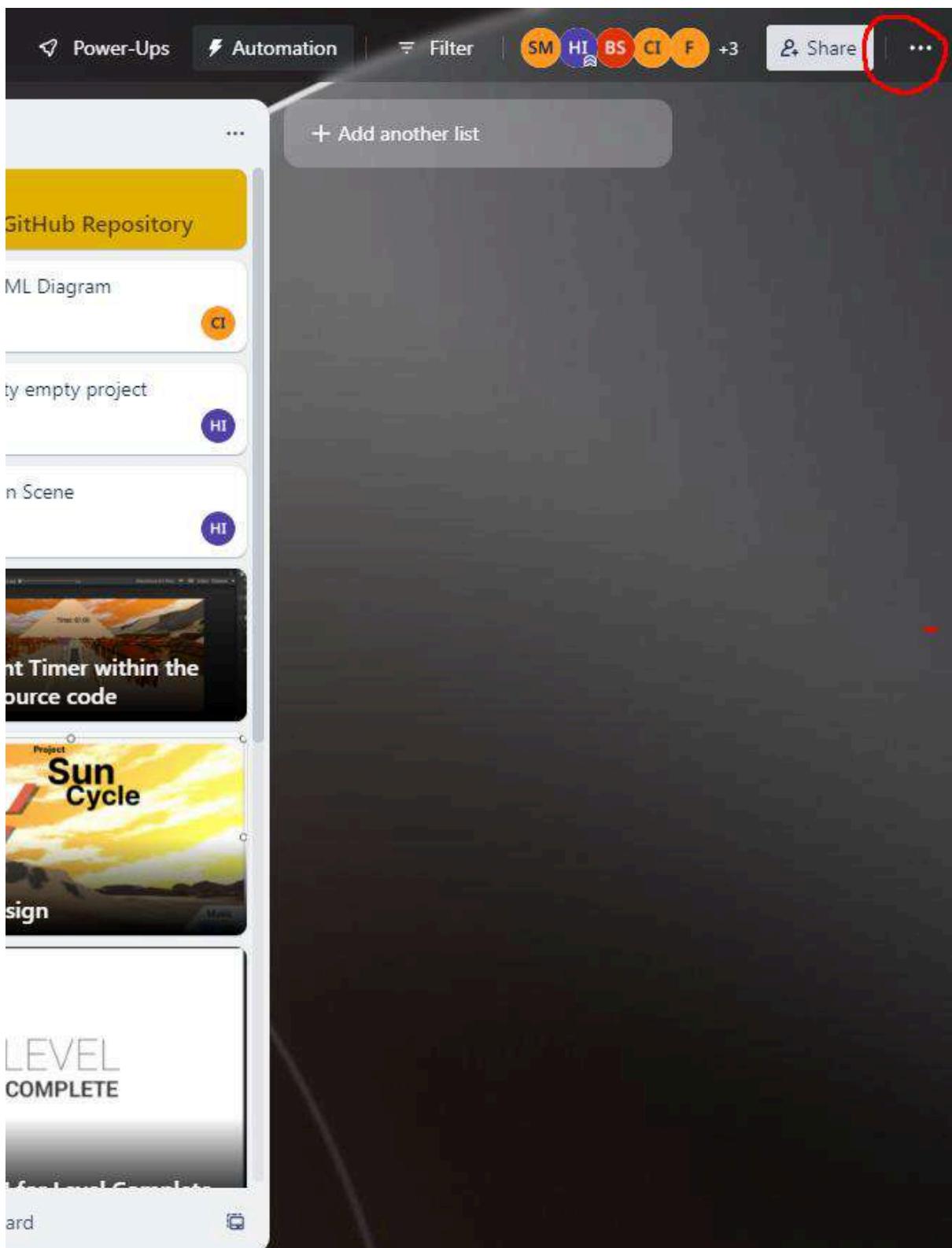
Assigning Members

When you have decided on a task you want to contribute to you can drag a user from the top right onto a card to assign that team member to the task.



Trello activity history

1. To see the activity in Trello, click the three-dot icon in the top right.



2. You can then see the history of your Trello board's activity.

Menu

- About this board**
Add a description to your board
- Change background**
- Custom Fields**
- Stickers**
- More**

- Automation**
Automate cards and more...

- Power-Ups**
Google Drive and more...

Activity

- Hilal Irshad** attached TimerUI.png to [Implement Timer within the game-Frontend](#) 21 May at 16:16

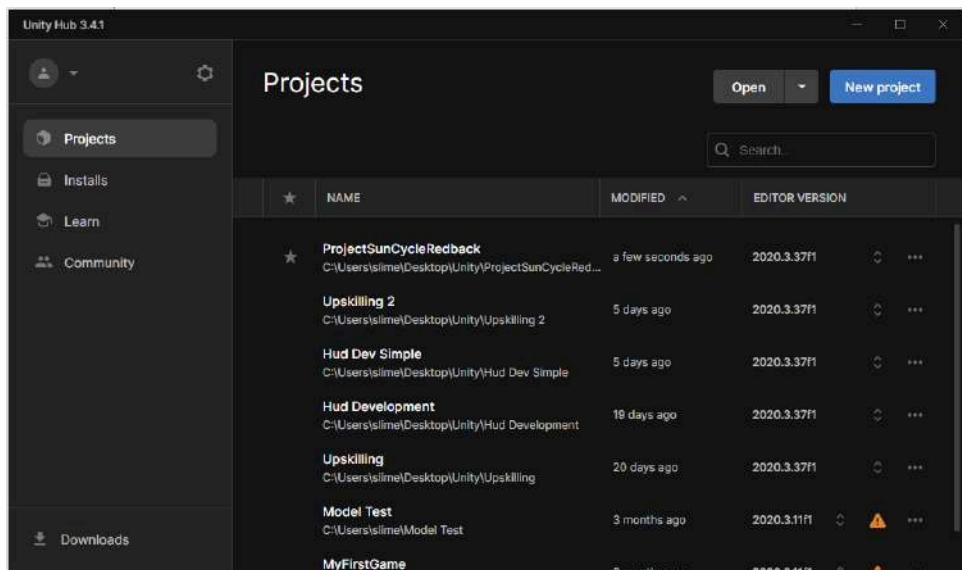
- Hilal Irshad** joined [MissionFailUI](#) 21 May at 16:15
- Hilal Irshad** joined [Implement Timer within the game- Frontend](#) 21 May at 16:14
- Hilal Irshad** moved [Implement Timer within the game- Frontend](#) from To Do to Done 21 May at 16:13

60.9. 3.3. Unity (Editor Version: 2020.3.37f1)

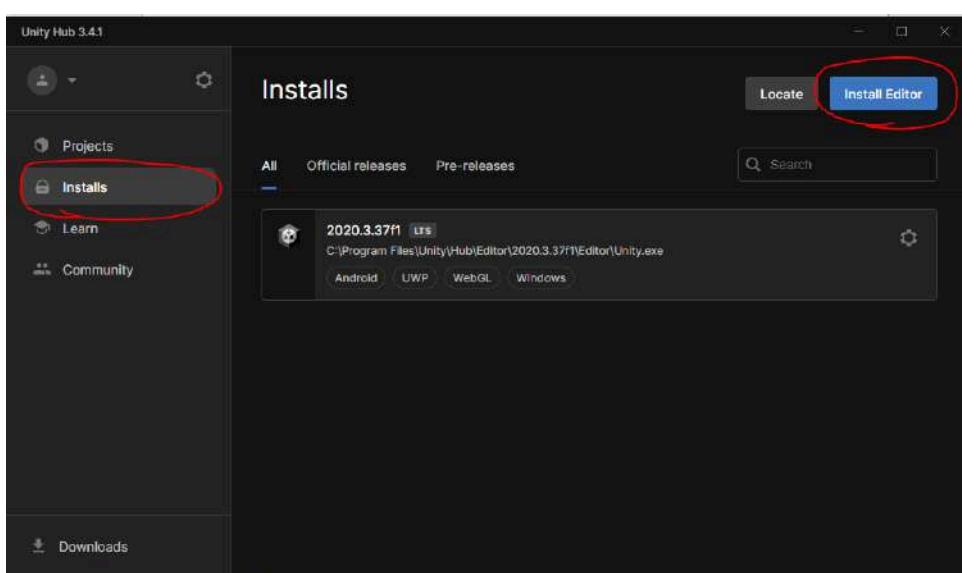
Unity is the game engine we use to develop our project. We are using Editor version

60.10. 3.3.1 Installation and setup

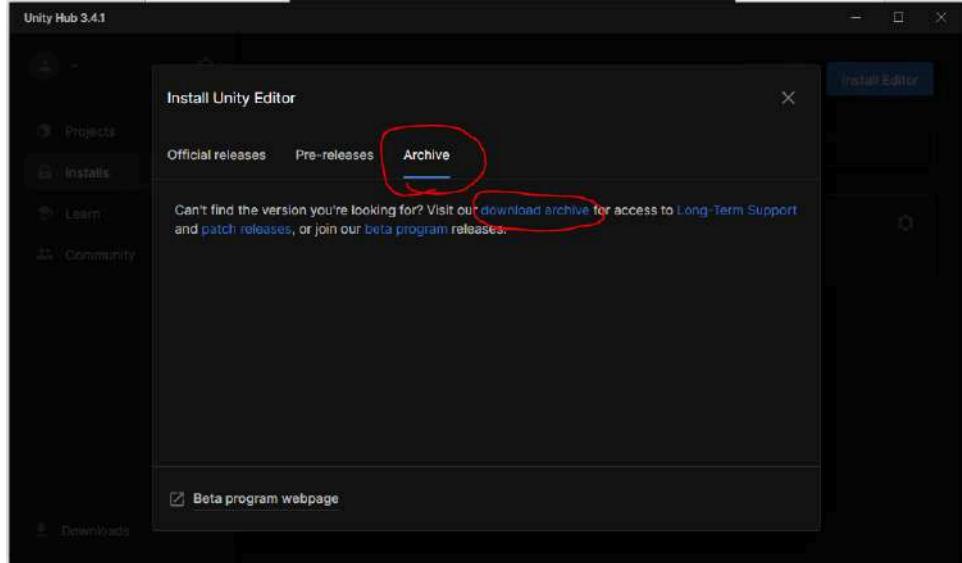
1. The First step is to create a Unity account. Go to the Unity website and “Create a Unity ID” (make an account).
2. The second step to using the Unity software is downloading it. Go to this link: <https://unity.com/download> and select and download the version that matches your circumstances (Operating System). Once downloaded execute the setup for the Unity Hub.
3. Launch Unity Hub and after you have logged in you will get something like this.



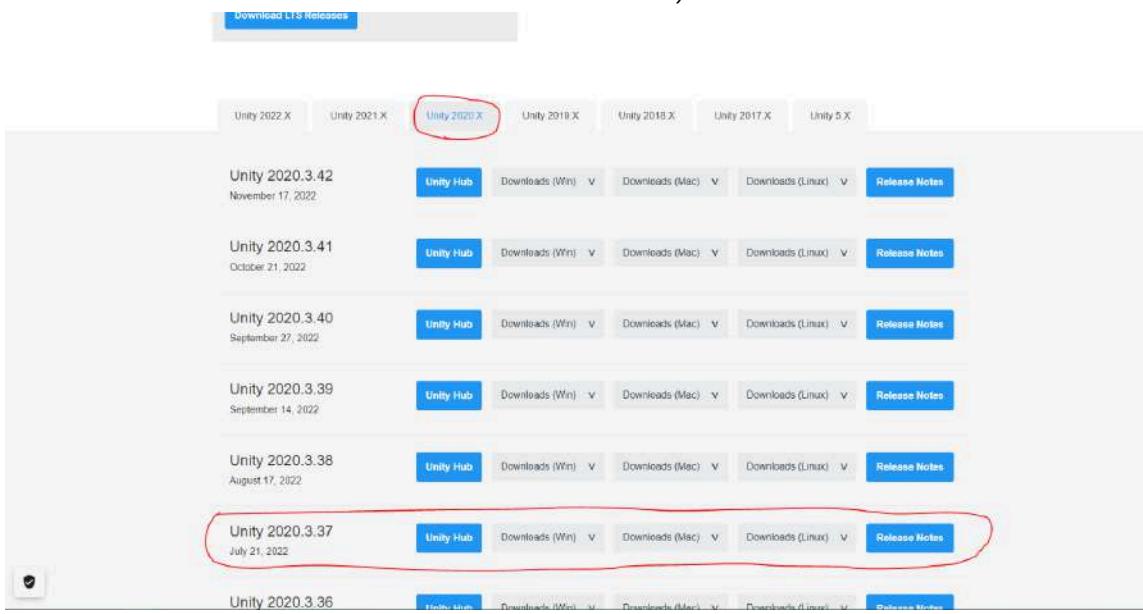
4. Now we will need to install the Editor Version that we are using for the Sun Cycle project. Go to the “Installs” tab on the left and then click the “Install Editor” button in the top right.



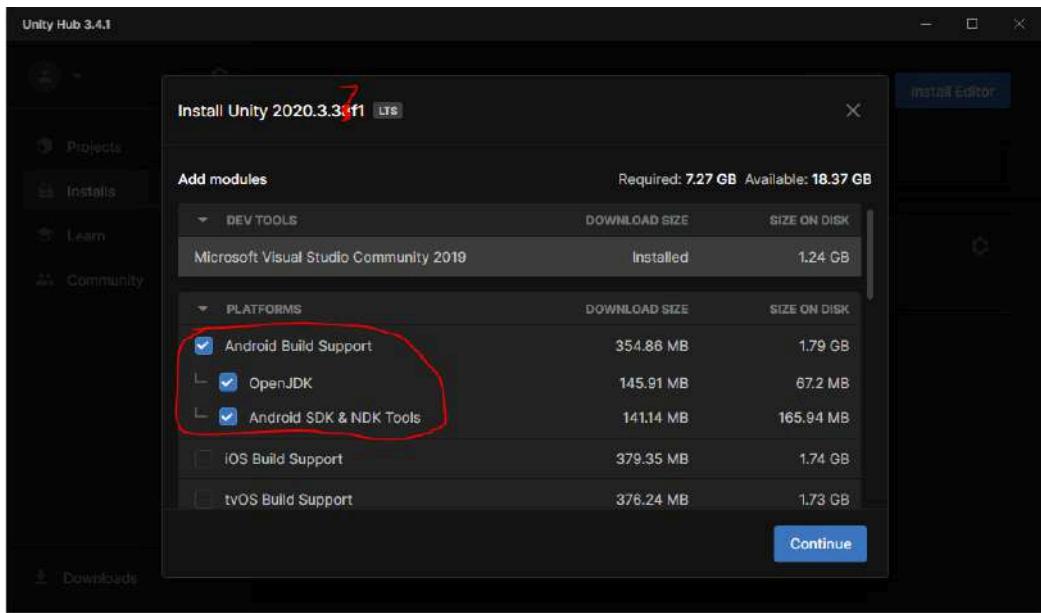
5. Then click on the "Archive" tab and click the download archive link (<https://unity.com/releases/editor/archive>).



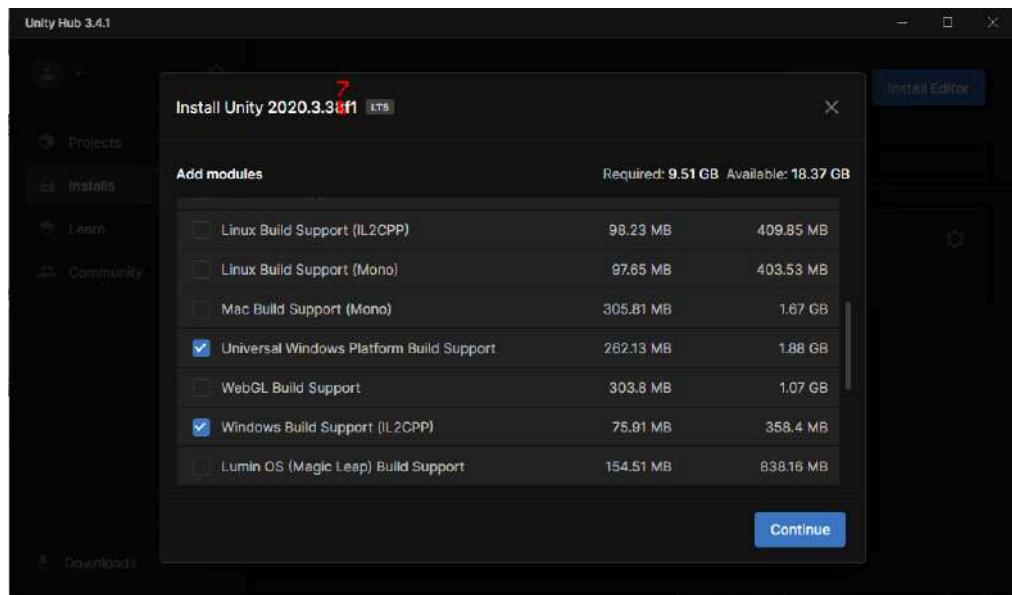
6. Then scroll down and select the "Unity 2020.X" tab. Then scroll down to the Unity Editor Version 2020.3.37 and click the "Unity Hub" button.



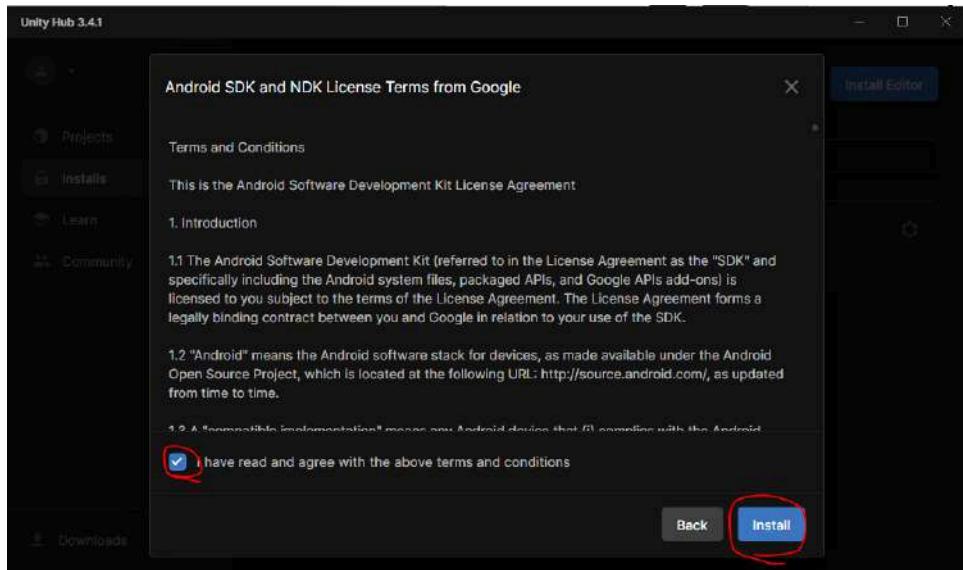
7. This will take you to the Unity Hub again where we will get to choose the Modules we want to add to the Editor. In this case we will want to add the Android Build Support for the VR headset.



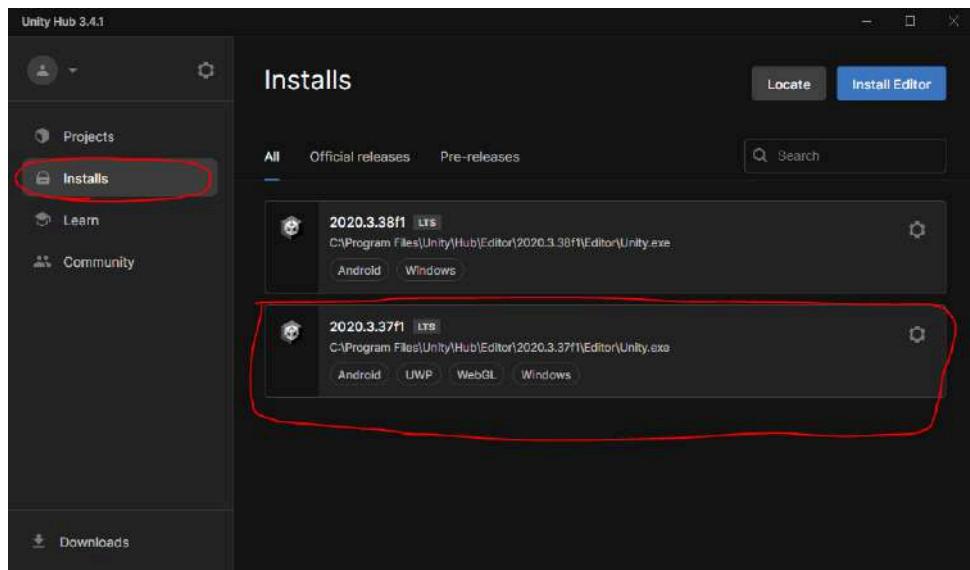
8. You can also choose to add other modules if you wish to test or build the project for other platforms. For example, I might select these modules as I use Windows OS:



9. Then click "Continue" and accept the terms and conditions and click "Install"



10. Once it is done installing should show up in the “Installs” tab.

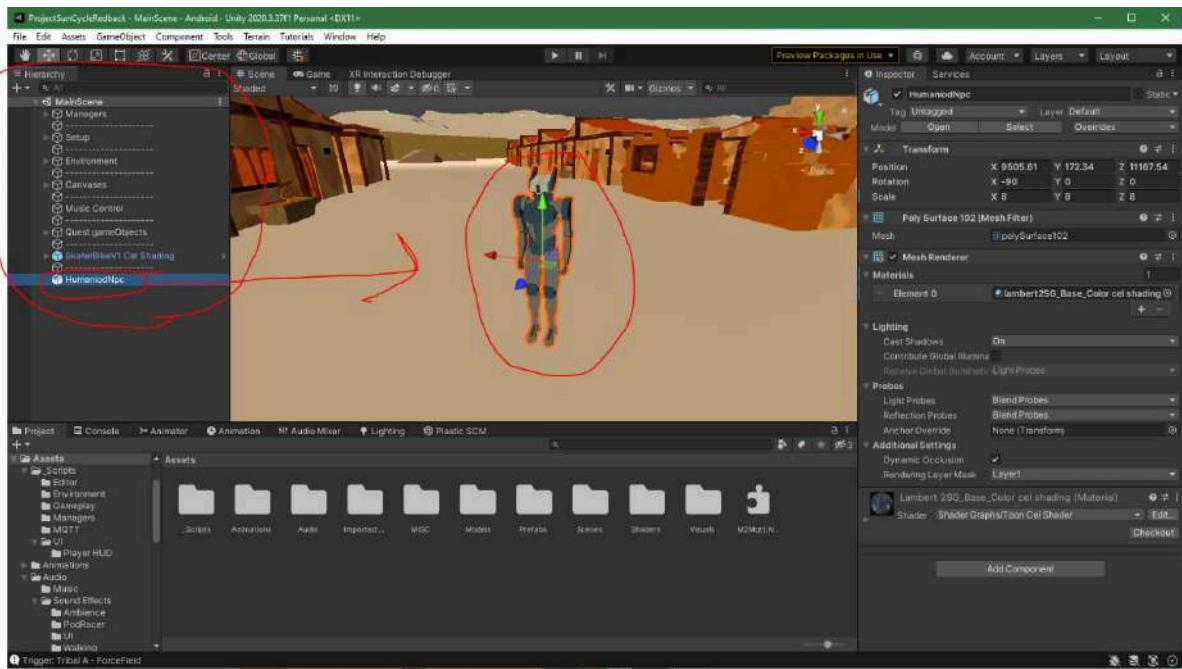


60.11. 3.3.2. Basics of Unity

Below explains the functionality of some of the most important windows/panels in Unity but there are many more you can use for specific things. It's a good idea to Upskill in Unity before usage.

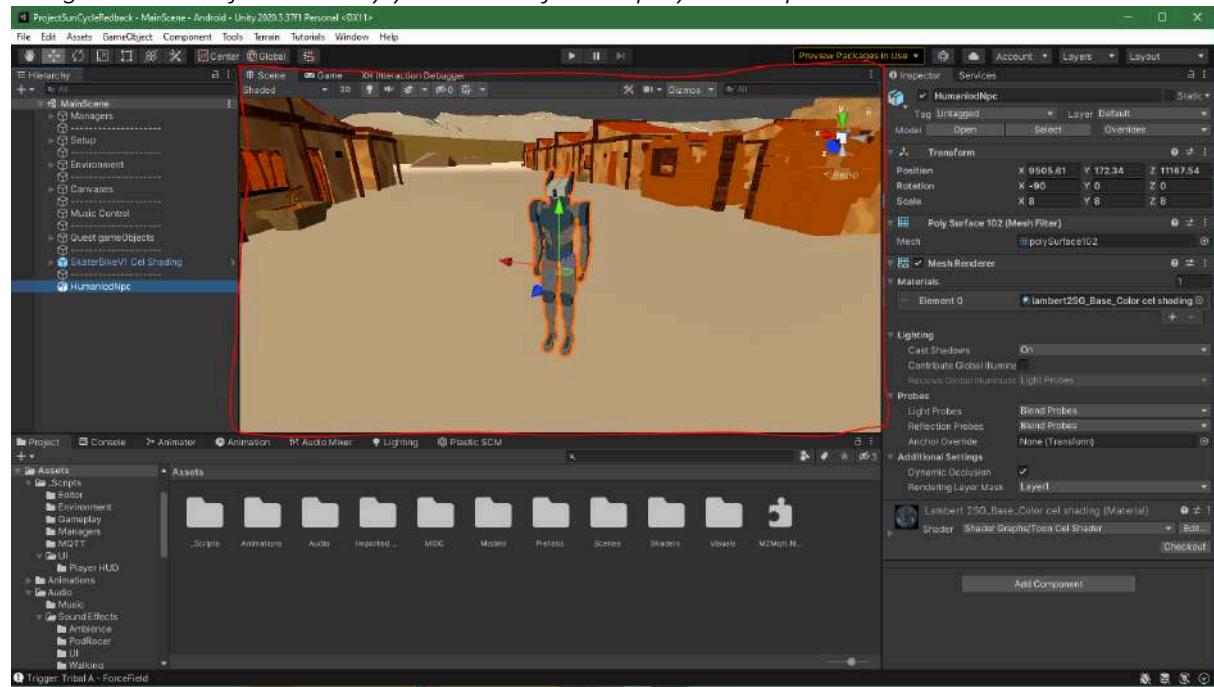
Hierarchy

This contains all of the game object that are currently within your scene, in this case “MainScene” contains “HumanoidNPC” which you can see in the middle.



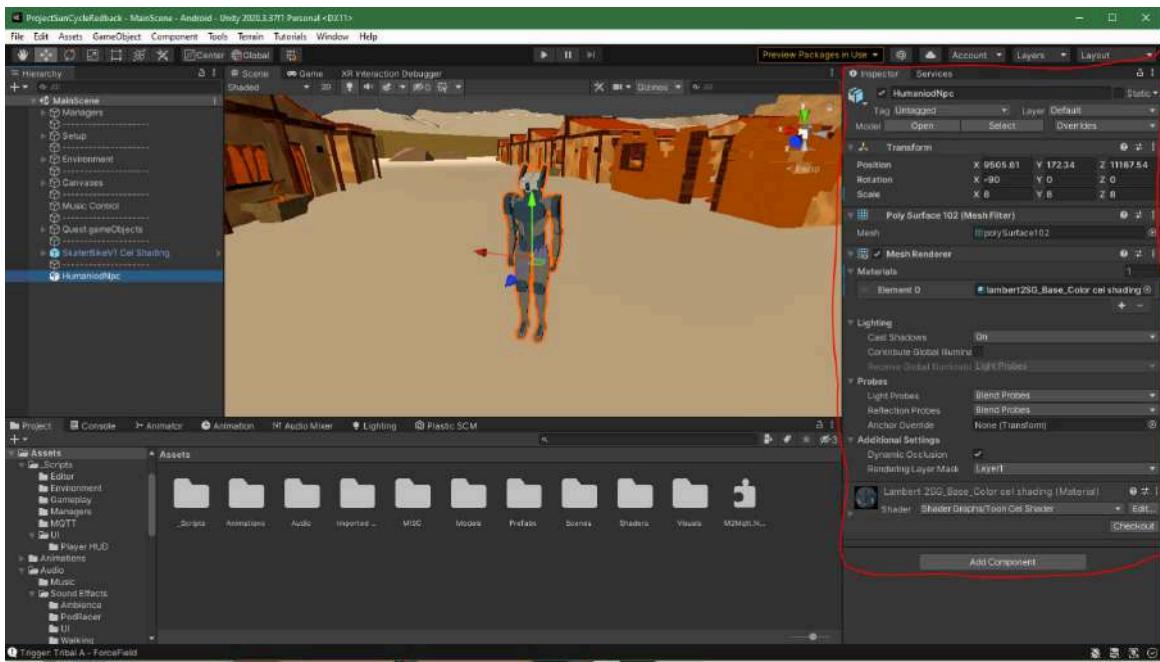
Scene

The scene is kind of like a level in a game, you add things to the scene that you want in that level. Through this screen you can add and manipulate the game objects in the level and design the world just the way you want it for the player to experience it.



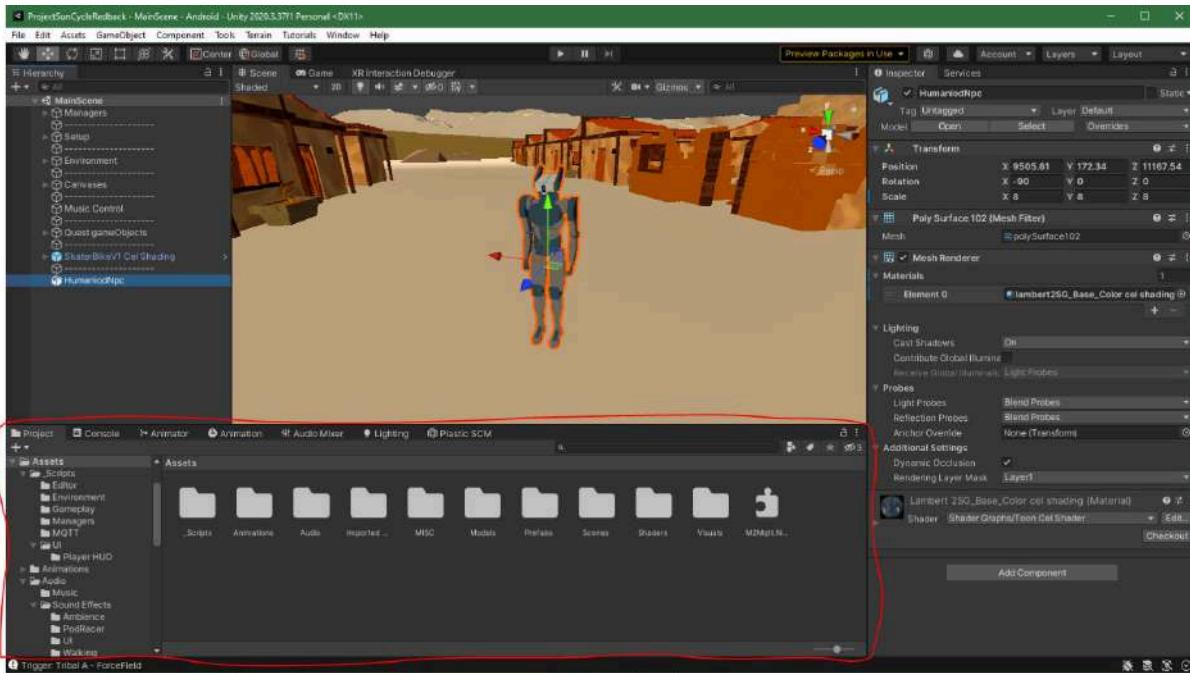
Inspector

Once you have clicked on something in either your Hierarchy or your Scene to select it the Inspector will pop up with the details of the selected object. It will show you all the different components of that game object, in this case the robot has a transform (position, rotation and scale in world space) and mesh renderer (renders the 3D mesh and material). Though this panel you can add or manipulate components.



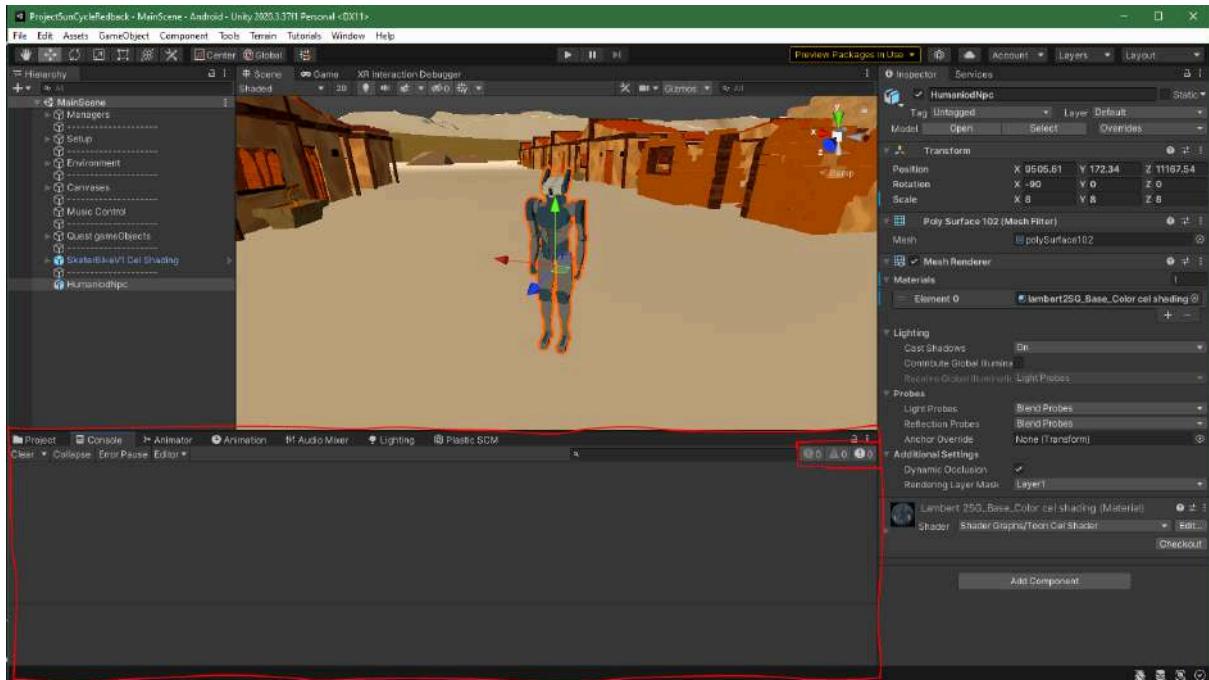
Project Directory

The project directory stores all of the assets that you are using in your project, things such as Audio, 3D models, textures, scripts, shaders, etc. You can use these assets to create game objects in any way you want. You should try to keep these files organized with simple understandable names.



Console

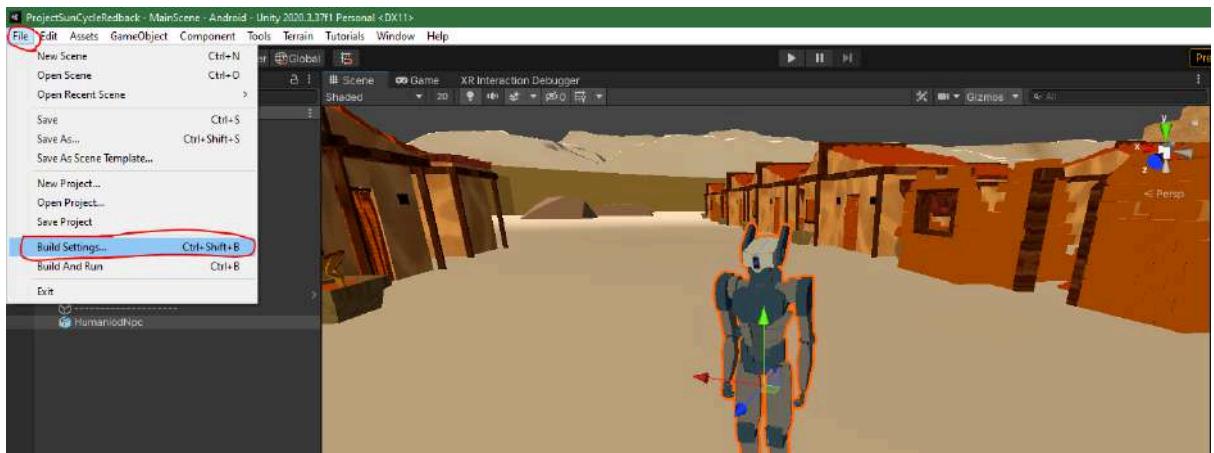
You can use the console panel while testing your game to debug your game and ensure it is free of errors. The three clickable boxes in the top right of the panel allow you to turn on debug messages, warning messages and error messages. You use these messages to diagnose and fix problems with the project.



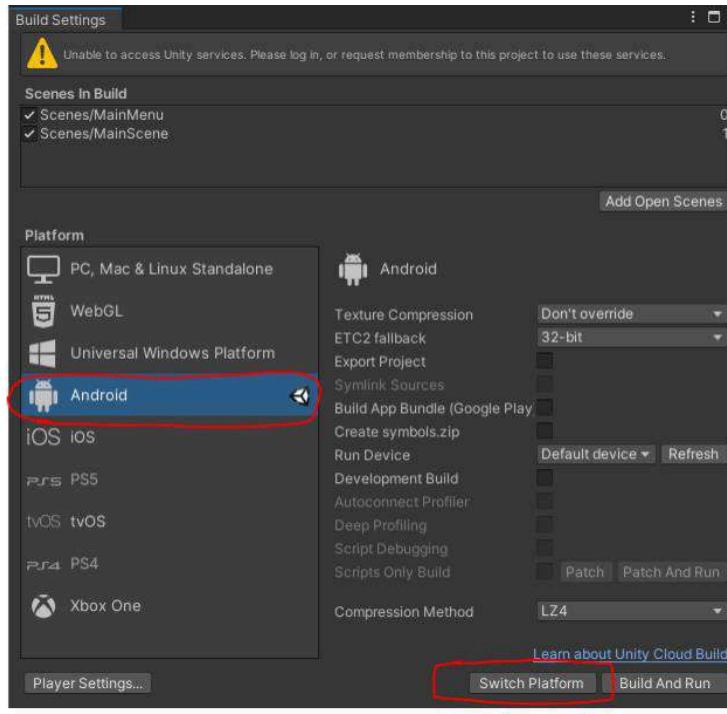
60.12. 3.3.3 Build settings

Building the project is basically the process of creating the stand-alone version of the game that can be run on a specific platform as an executable. Because we want this game to run on a VR headset, we need change the platform to Android.

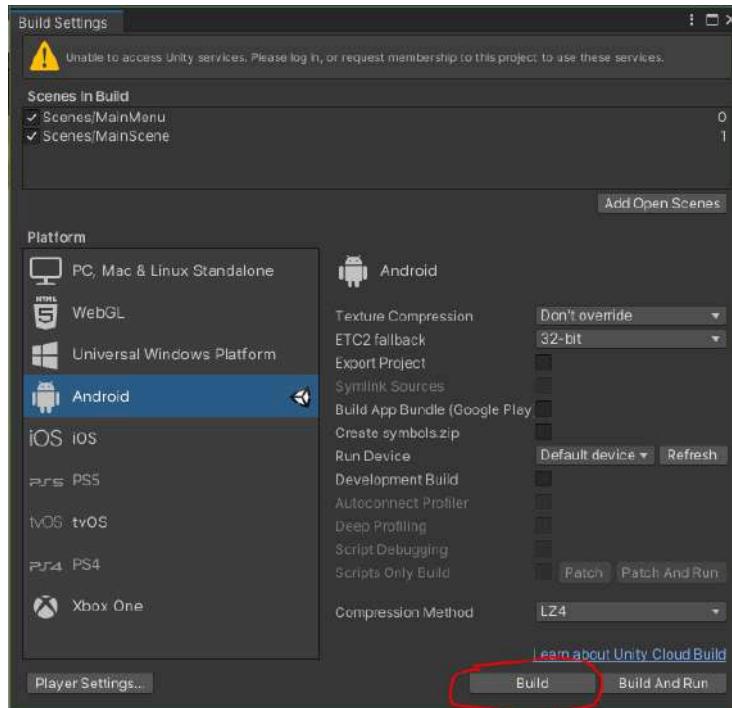
1. Go to “File” -> “Project Settings...”



2. Once you are in the project setting click on the “Android” tab and click “Switch Platform”



3. After this process is finished you will be able to make a build of the project by clicking the "Build" button, this will be stored as an executable file wherever you choose the location to be.

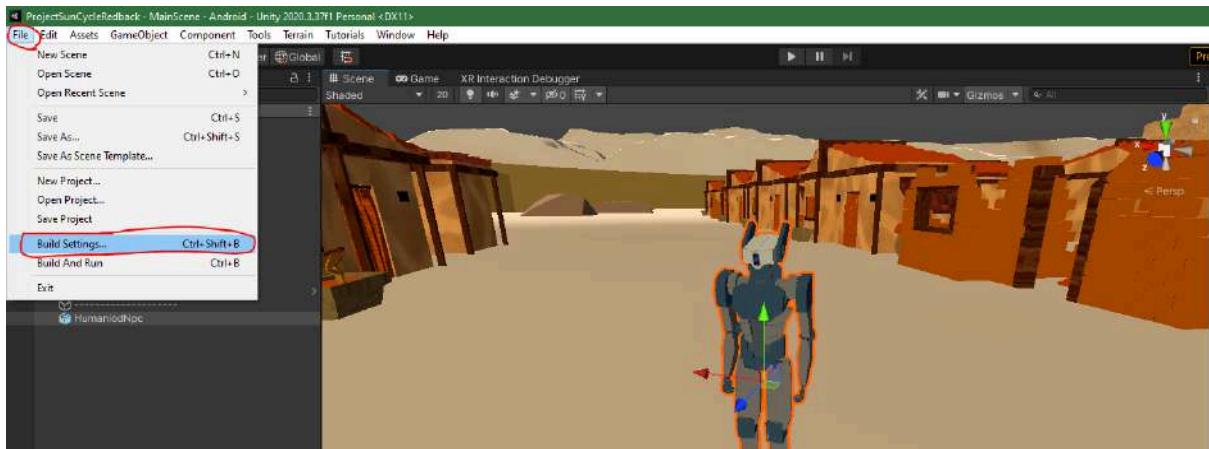


60.13. 3.3.4. Using VR Headset

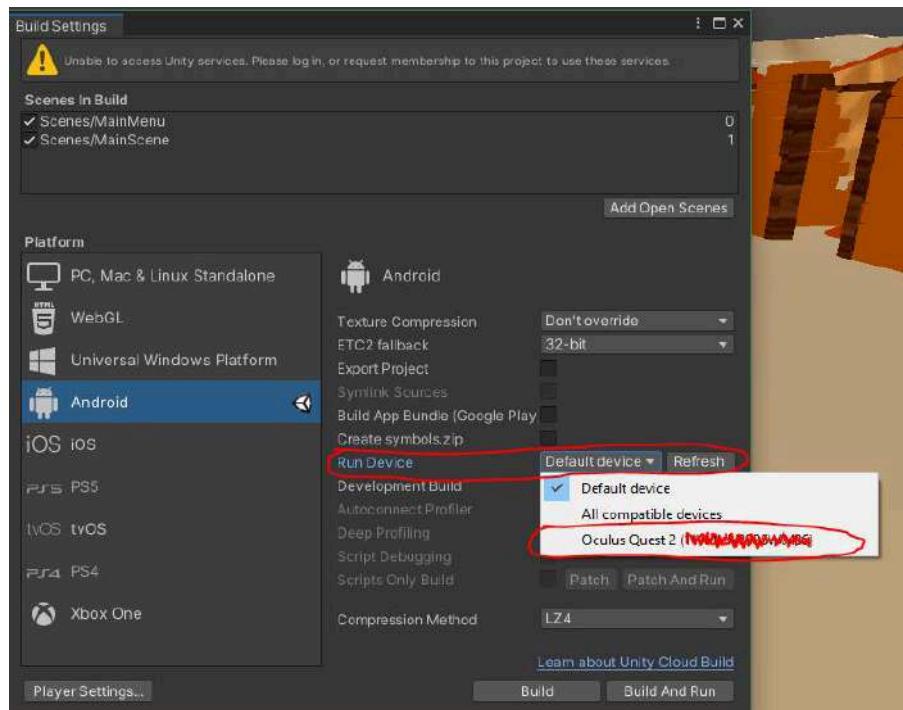
Note: this section contains biased information based around the Oculus/Meta Quest 2 headset.

1. Have Oculus application open.

2. Have Unity project open.
3. Install & Setup oculus app on your phone.
4. Watch this short video to turn on developer mode on your headset:
https://www.youtube.com/watch?v=75dHXROt4mE&ab_channel=HowToGeek
5. Connect VR headset to computer using cable (or use air link, not recommended).
6. Accept popups on the headset.
7. Launch Quest link on headset.
8. In Unity go to “File” -> “Build Settings...”

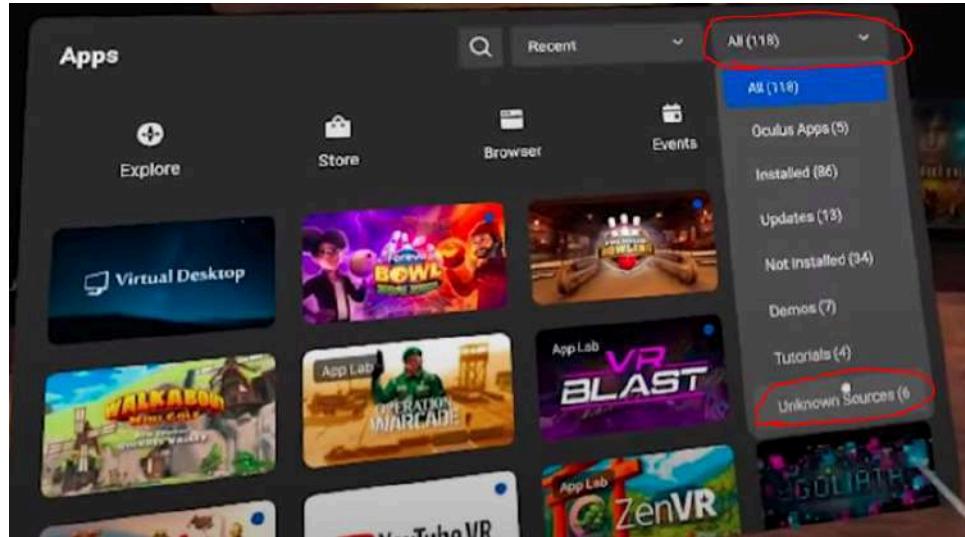


9. Then go to the “Run Device” section and hit “Refresh”, once that is done click the “Default device” dropdown and select “Oculus Quest 2”.



10. Click play button in Unity to test the game in the editor on your headset.
11. If you want to build a stand-alone version of the game on your headset, just click the “Build” button in the Project Settings and after it is done building you can unplug your headset.

12. To find the stand alone build go to your apps on your headset and click the dropdown menu in the top right and then sort by “Unknown Sources”.



60.14. 3.4. GitHub

A central centre for version control and developer collaboration, GitHub is a web-based platform. It offers Git, a distributed version control system that enables multiple users to work on the same project at once without overwriting one another's modifications.

Github Link: <https://github.com/redbackoperations/VR-T12023>

60.15. 3.4.1. Setup Github

A contributor access is required in order to operate on your GitHub repository and guarantee that all files are preserved. Without contributor status, certain files may be unavailable or missing during the collaboration process.

Ask the company leader to gain contributor access.

1. Now that you have contributor access. Open Github and download “Download only me & READ ME FIRST” & “Download only me & READ ME FIRST”

I will include the contents of these files here.

2. “Download only me & READ ME FIRST” file:

VR-T12023

To work on this project please read 'GitHub how to upload 100mb+.docx' first, because Github won't allow you to upload more than 100mb.

On the files Unity. You will find our game.

Do not Use the Backup one as it is in a safe state of the current game.

Backup should be used only when all the changes made are safe to be transferred without breaking the current state of the game.

Instructions

You cannot clone it until you have read 'GitHub how to upload 100mb+.docx' and follow the steps

3. "GitHub how to upload 100mb+" file:

This Document is to show you how to upload more than 100mb on GitHub and work on the project.

1. Download Git. Don't change any option during the installation except the default editor for Git if you want.

<https://git-scm.com/downloads>

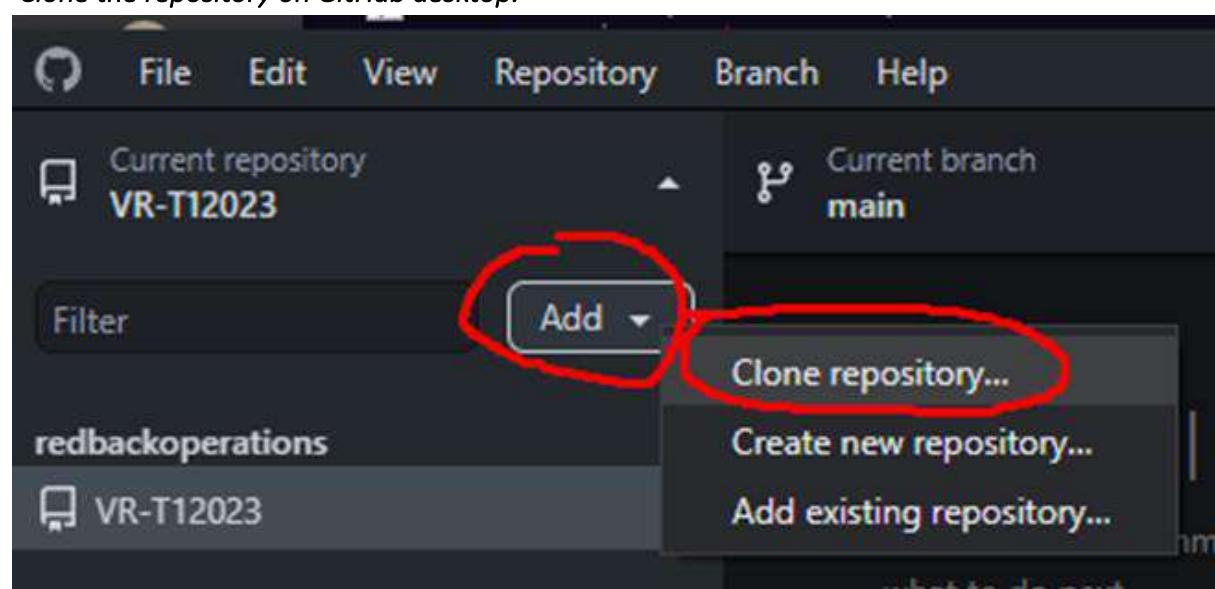
2. Download Git LFS

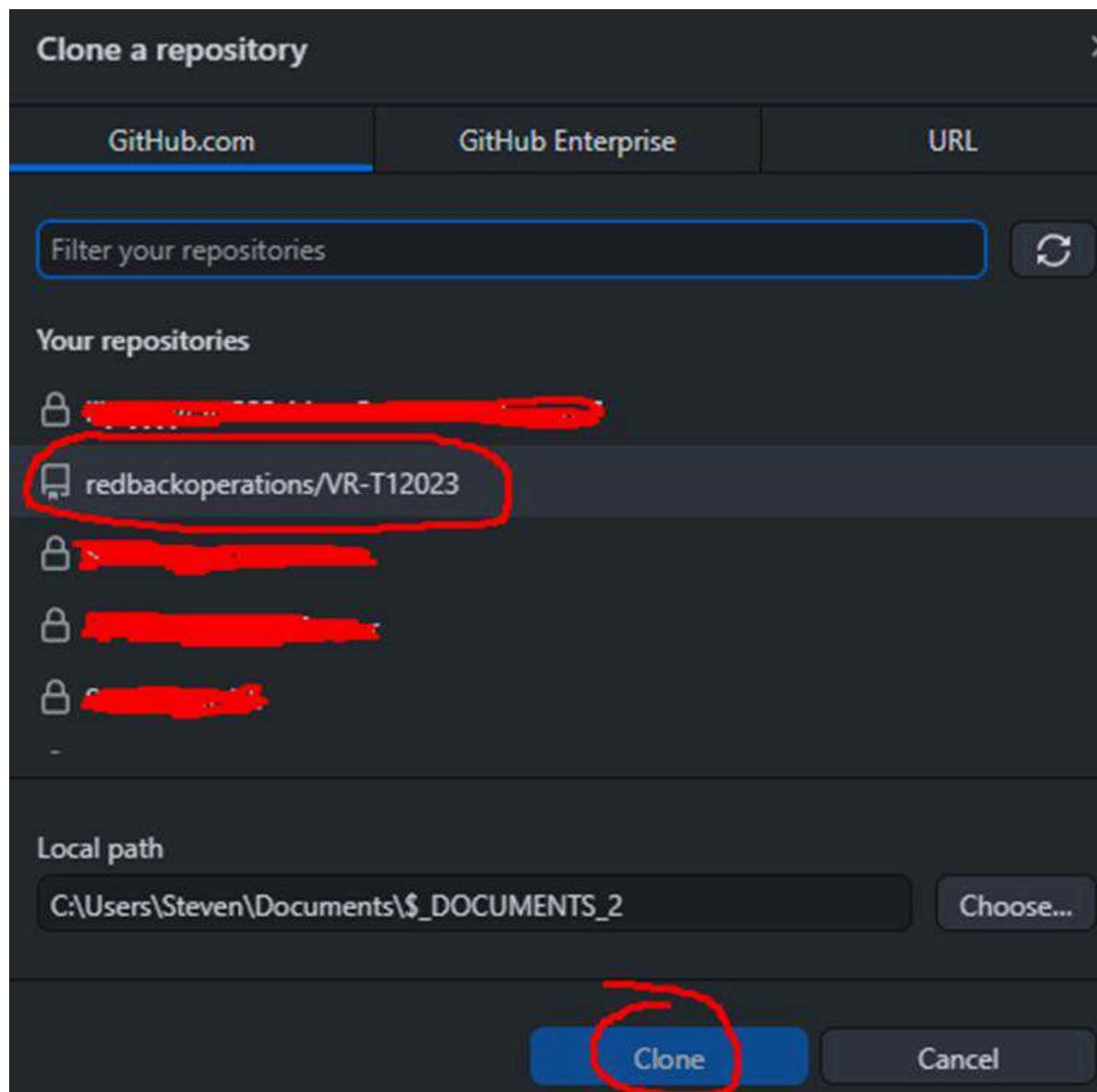
<https://git-lfs.com/>

3. Download GitHub Desktop

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

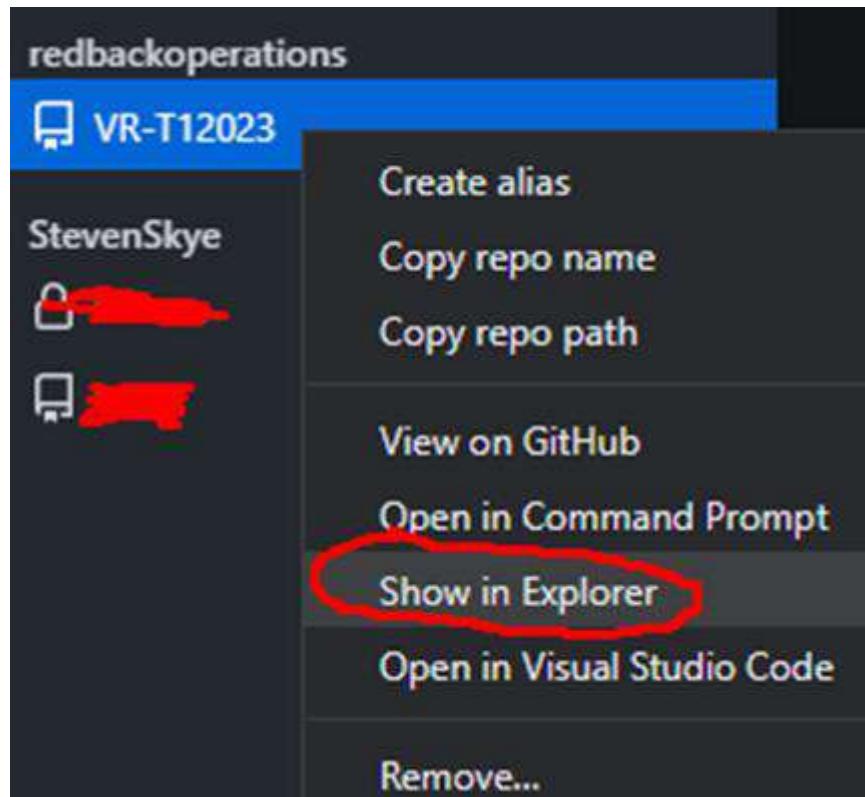
4. Clone the repository on GitHub desktop.





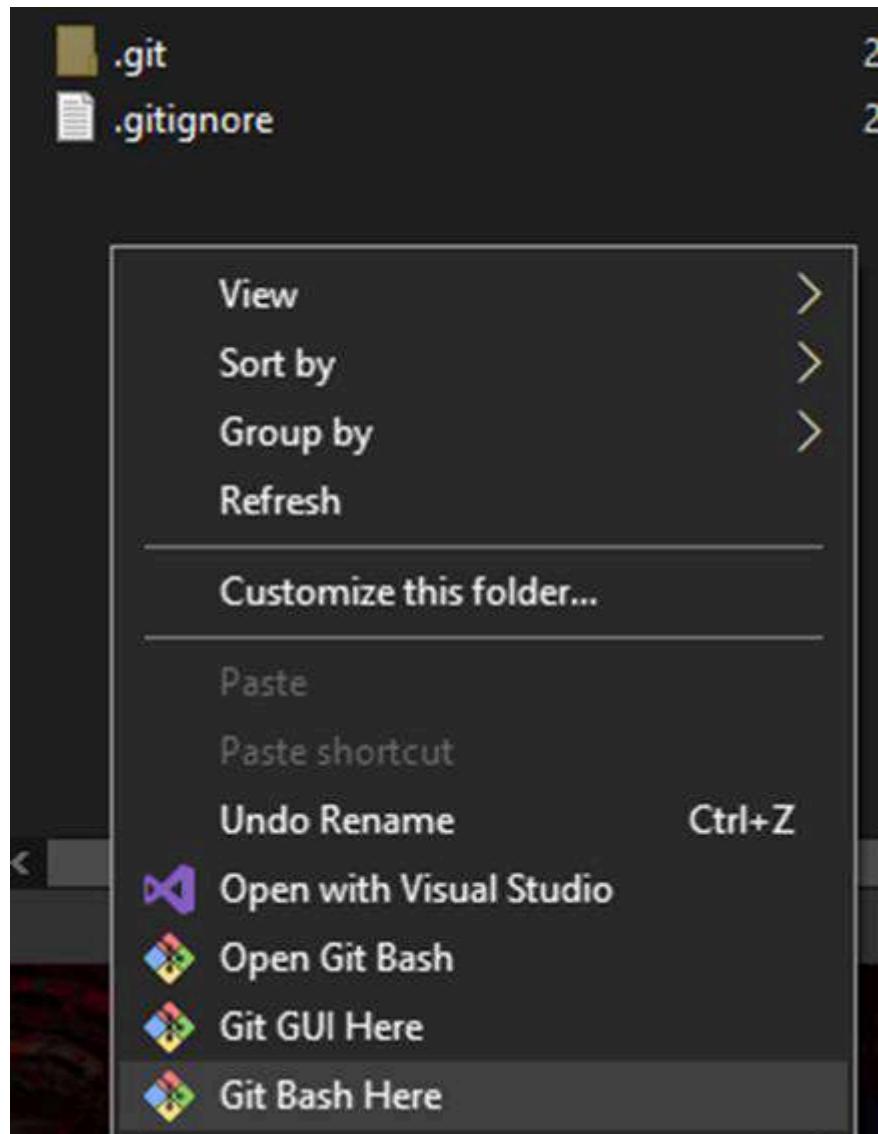
It will ask you if you want to initialize GIT LFS. You need to Initialize it and accept it if you want all the files to be there and working.

Now you can Open the Explorer:

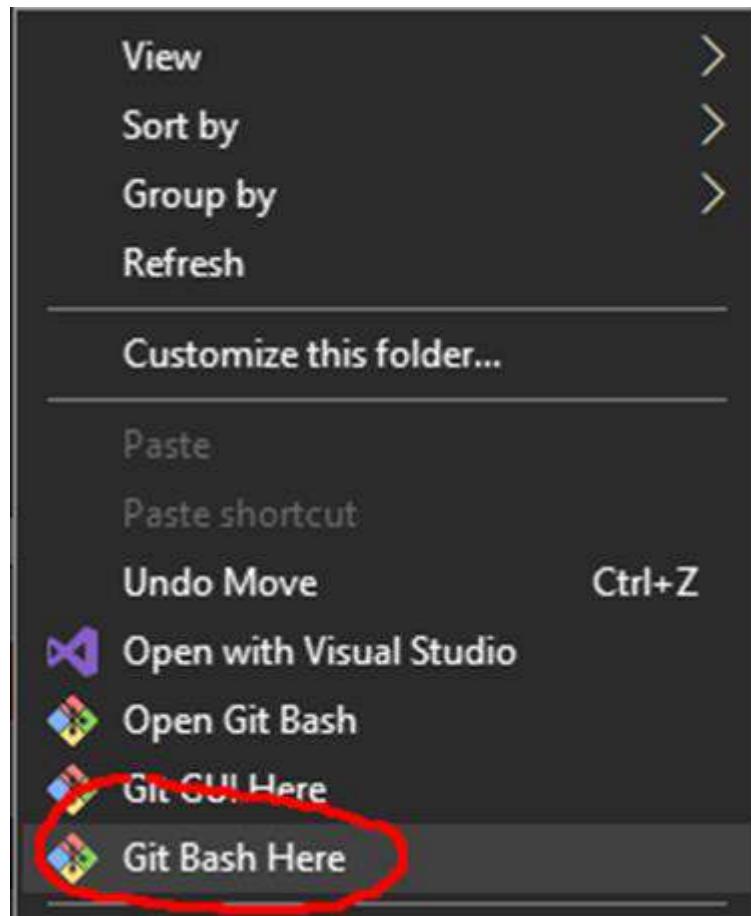


5. Now you should be able to work on the repository and make changes. You don't have to read the following steps except if you want to know how Git LFS work.

6. Right click on the Explorer file (where the GitHub repo is)



7. Click on Git Bash Here

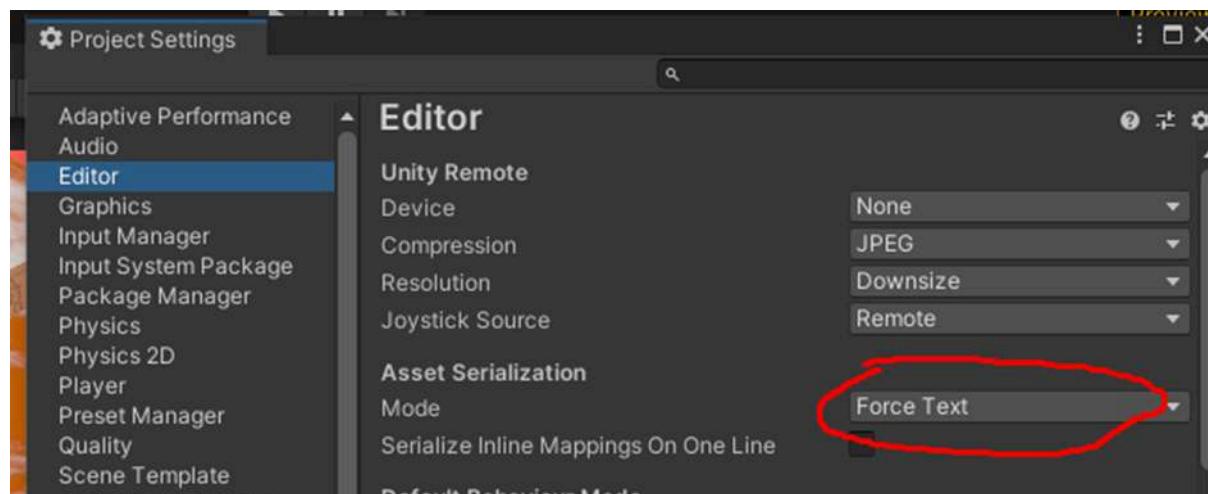


8. Mingw64 will open and write the command: git lfs install

```
Steven@DESKTOP-IR89E08 MINGW64 ~/Documents/$_DOCUMENTS_2/VR-T12023 (main)
$ git lfs install
Updated Git hooks.
Git LFS initialized.
```

This will initialize Git LFS

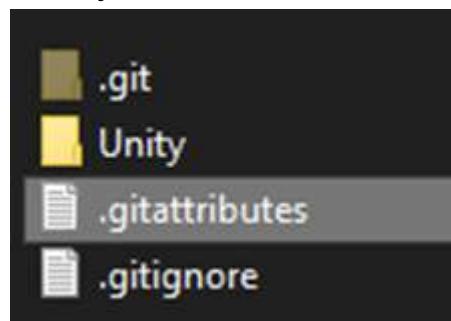
9. Open the Unity Project and make sure the project is "Force Text" (Usually this is the default option)



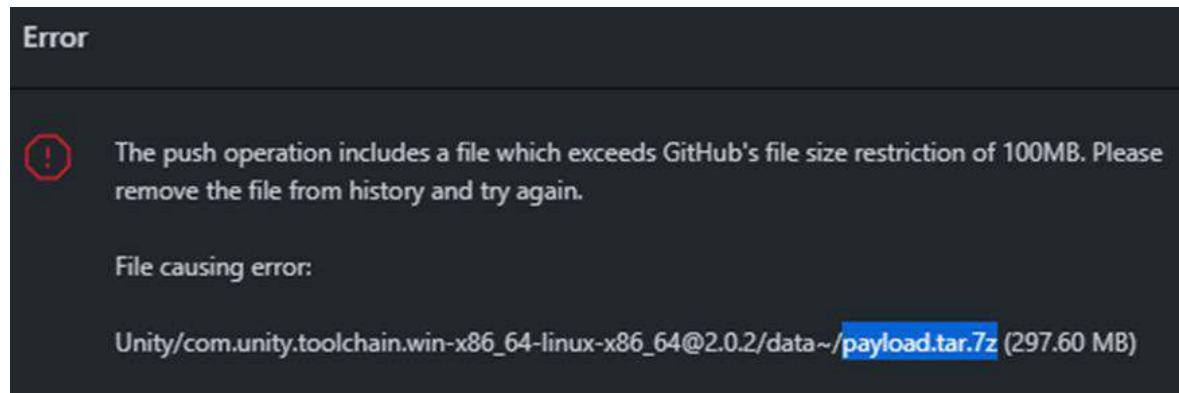
10. Now we want Git LFS to track files that can be over 100mb. For now, we use .psd (If [.gitattributes](#) already exist. You do not have to do this part Go to step 13.)

```
Steven@DESKTOP-IR89E08 MINGW
$ git lfs track "*.psd"
Tracking "*.psd"
```

11. A new file will be created that tracks the specified files.



12. Now when you try to commit a new change to the project and a file is over 100mb. Check its file extension.



13. Open .gitattributes and write this code

```
*.1xo filter=lfs diff=lfs merge=lfs -text  
*.7z filter=lfs diff=lfs merge=lfs -text
```

14. Now Commit .gitattributes first and then commit the change that was more than 100mb. If you try to commit the change that include 100mb. It will not accept it. You need to commit the change of .gitattributes to accept your file that is more than 100mb

15. This code will show all the files that have been track by Git LFS

```
STEVENDESKTOP-1K9GJ
$ git lfs ls-files
$ git lfs ls-files
bf196a8dd7 * Unity/ProjectSunCycleRedback/Assets/Assets/Bad House.fbx
f60430f1cb * Unity/ProjectSunCycleRedback/Assets/Assets/Port House.fbx
641efbe557 * Unity/ProjectSunCycleRedback/Assets/Assets/Port House.fbx
fa3105934 * Unity/ProjectSunCycleRedback/Assets/Assets/Port House.fbx
Keo_game_sounds - 01 Start - Crows Constant.wav
39e79b27a4 * Unity/ProjectSunCycleRedback/Assets/Assets/Port House.fbx
Keo_game_sounds - 01 Start - Crows Approach.wav
262a04d047 * Unity/ProjectSunCycleRedback/Assets/Assets/Port House.fbx
Keo_game_sounds - 01 Start - Pod Racer Start.wav
8564179d47 * Unity/ProjectSunCycleRedback/Assets/Assets/Port House.fbx
Keo_game_sounds - 01 Start - Pod Racer Fin.wav
437c580f8a * Unity/ProjectSunCycleRedback/Assets/Assets/Port House.fbx
Keo_game_sounds - 01 Start - Accepted Quest.wav
499d080c1a * Unity/ProjectSunCycleRedback/Assets/Assets/Port House.fbx
Keo_game_sounds - 01 Start - Menu Clicking.wav
ad41e114c0 * Unity/ProjectSunCycleRedback/Assets/Assets/Port House.fbx
Keo_game_sounds - 01 Start - Menu Selecting.wav
a238634a40 * Unity/ProjectSunCycleRedback/Assets/Assets/Port House.fbx
Keo_game_sounds - 01 Start - Notification.wav
0418495f39 * Unity/ProjectSunCycleRedback/Assets/Assets/Port House.fbx
Keo_game_sounds - 01 Start - Text Scroll.wav
dc92038d75 * Unity/ProjectSunCycleRedback/Assets/Assets/Port House.fbx
Keo_game_sounds - 01 Start - Indoor Walk.wav
c2ca4b18b7 * Unity/ProjectSunCycleRedback/Assets/Assets/Port House.fbx
Keo_game_sounds - 01 Start - Outdoor Walk.wav
34c5eb0414 * Unity/ProjectSunCycleRedback/Assets/Buttons/Buttons-Exit Game[3328].png
9f0f715af5 * Unity/ProjectSunCycleRedback/Assets/Buttons/Buttons-Main Menu[3329].png
7669efdfb1 * Unity/ProjectSunCycleRedback/Assets/Buttons/Buttons-Options[3330].png
afd8575c49 * Unity/ProjectSunCycleRedback/Assets/Buttons/Buttons-Play[3331].png
a45f498b6a * Unity/ProjectSunCycleRedback/Assets/Buttons/Buttons-Quit[3332].png
63ffb5e8e0 * Unity/ProjectSunCycleRedback/Assets/Buttons/Buttons-Resume[3333].png
baef7d8bd92 * Unity/ProjectSunCycleRedback/Assets/Buttons/Buttons-Settings[3334].png
b34f75dff * Unity/ProjectSunCycleRedback/Assets/Buttons/Play Button.png
9cccf ea7aa * Unity/ProjectSunCycleRedback/Assets/Buttons/Quit Button.png
9e6470bb3c * Unity/ProjectSunCycleRedback/Assets/Buttons/Settings Button.png
ecc2becbf * Unity/ProjectSunCycleRedback/Assets/Custom Quest/Assets/GFX/UI/Buttons/Plus Down Test.png
3dfb79efcf * Unity/ProjectSunCycleRedback/Assets/Custom Quest/Assets/GFX/UI/Buttons/active (1).png
2d21793c4a * Unity/ProjectSunCycleRedback/Assets/Custom Quest/Assets/GFX/UI/Buttons/button150 down.png
95790d18e7 * Unity/ProjectSunCycleRedback/Assets/Custom Quest/Assets/GFX/UI/Buttons/button150 hover.png
c2076bd1fe * Unity/ProjectSunCycleRedback/Assets/Custom Quest/Assets/GFX/UI/Buttons/button150x.png
80e751627f * Unity/ProjectSunCycleRedback/Assets/Custom Quest/Assets/GFX/UI/Buttons/button16 down.png
61dd776ba1 * Unity/ProjectSunCycleRedback/Assets/Custom Quest/Assets/GFX/UI/Buttons/button16 hover.png
77a14eda4c * Unity/ProjectSunCycleRedback/Assets/Custom Quest/Assets/GFX/UI/Buttons/button16x.png
04b2c9b049 * Unity/ProjectSunCycleRedback/Assets/Custom Quest/Assets/GFX/UI/Buttons/down (1).png
eebe352fc3 * Unity/ProjectSunCycleRedback/Assets/Custom Quest/Assets/GFX/UI/Buttons/down.png
a12865c34b * Unity/ProjectSunCycleRedback/Assets/Custom Quest/Assets/GFX/UI/Buttons/down hover (1).png
4292c74857 * Unity/ProjectSunCycleRedback/Assets/Custom Quest/Assets/GFX/UI/Buttons/down hover.png
418010c38b * Unity/ProjectSunCycleRedback/Assets/Custom Quest/Assets/GFX/UI/Buttons/in background (1).png
79ad53af16 * Unity/ProjectSunCycleRedback/Assets/Custom Quest/Assets/GFX/UI/Buttons/in background (1).png
```

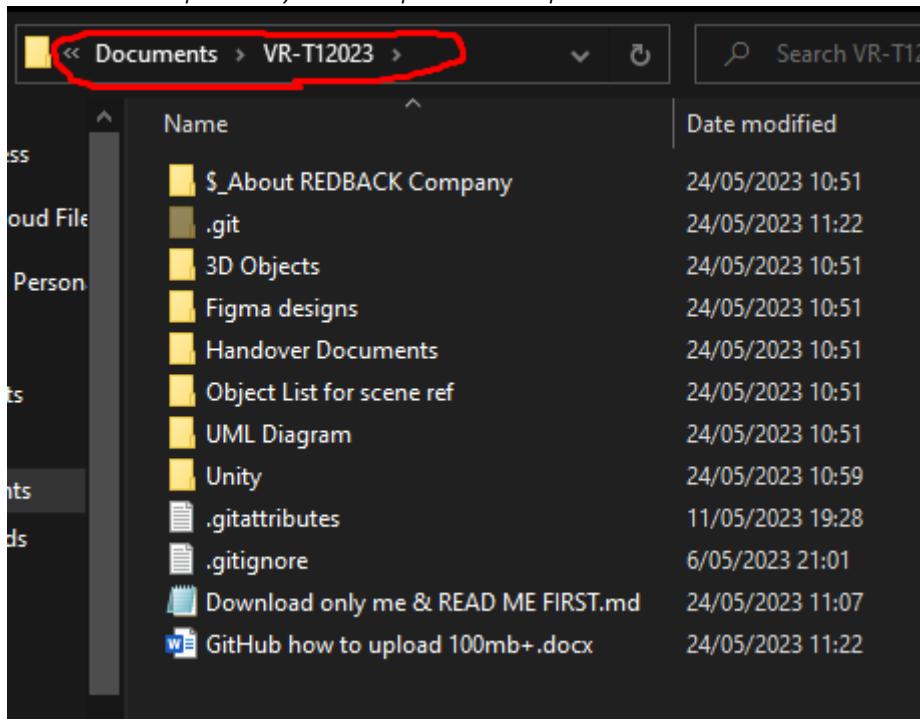
16. Now you should be able to upload more than 100mb

60.16. 3.4.2. Using GitHub with Unity

Todo: How to use github with unity and how it works

After completing 3.4.1. We can now use Github Desktop to share files.

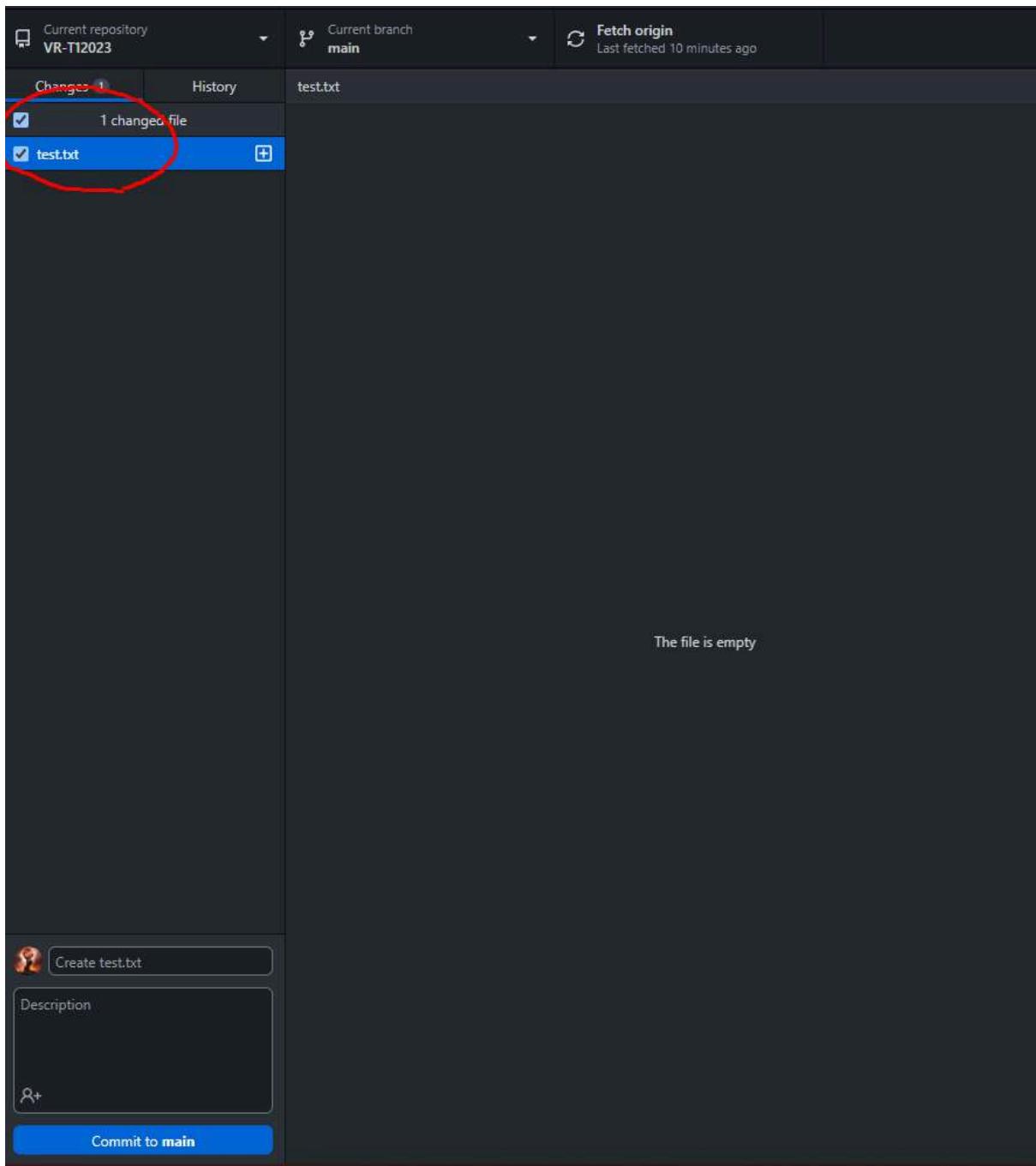
Find the local path in your computer and open it:



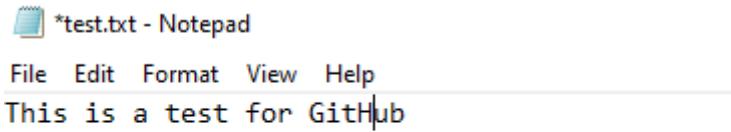
Here, any changes you make. Will also happen on “GitHub Desktop”

For example if you create any file (I will create a text file called “Test”). The change will appear on “GitHub Desktop”

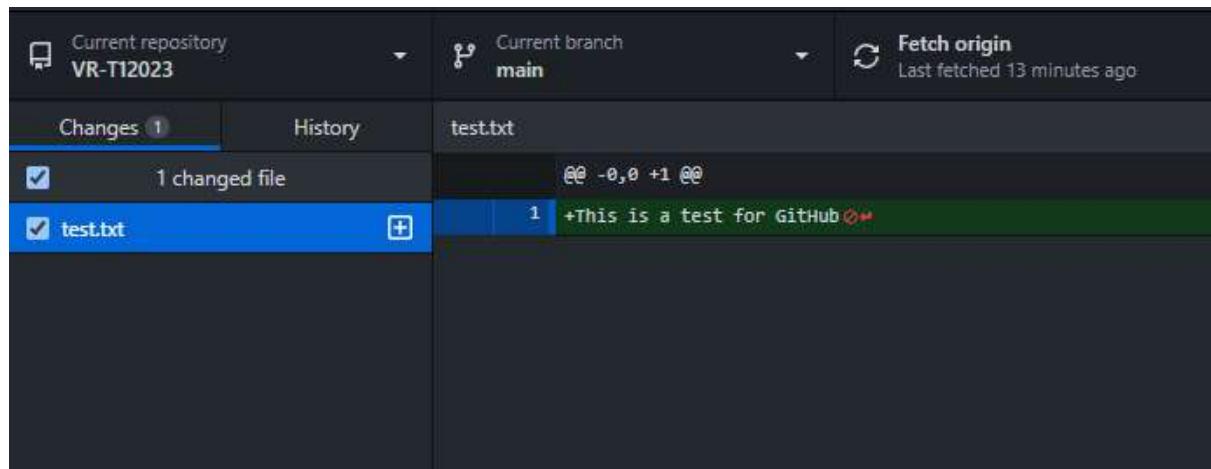
Name	Date modified	Type	Size
\$_About REDBACK Company	24/05/2023 10:51	File folder	
.git	24/05/2023 11:22	File folder	
3D Objects	24/05/2023 10:51	File folder	
Figma designs	24/05/2023 10:51	File folder	
Handover Documents	24/05/2023 10:51	File folder	
Object List for scene ref	24/05/2023 10:51	File folder	
UML Diagram	24/05/2023 10:51	File folder	
Unity	24/05/2023 10:59	File folder	
.gitattributes	11/05/2023 19:28	Text Document	2 KB
.gitignore	6/05/2023 21:01	Text Document	2 KB
Download only me & READ ME FIRST.md	24/05/2023 11:07	MD File	1 KB
GitHub how to upload 100mb+.docx	24/05/2023 11:22	Microsoft Word D...	262 KB
test.txt	24/05/2023 11:31	Text Document	0 KB



Now you can write anything on that file and it will automatically go to 'GitHub Desktop' if you save it.



```
*test.txt - Notepad
File Edit Format View Help
This is a test for GitHub
```



```
Current repository VR-T12023
Current branch main
Fetch origin
Last fetched 13 minutes ago

Changes 1 History test.txt
1 changed file
test.txt +This is a test for GitHub
```

To actually share your files with the team you have to commit and push the change:
Commit:

Current repository VR-T12023

Current branch main

Changes 1 History

1 changed file

test.txt

@@ -0,0 +1 @@

1 +This is a test for GitHub

Create test.txt

Description

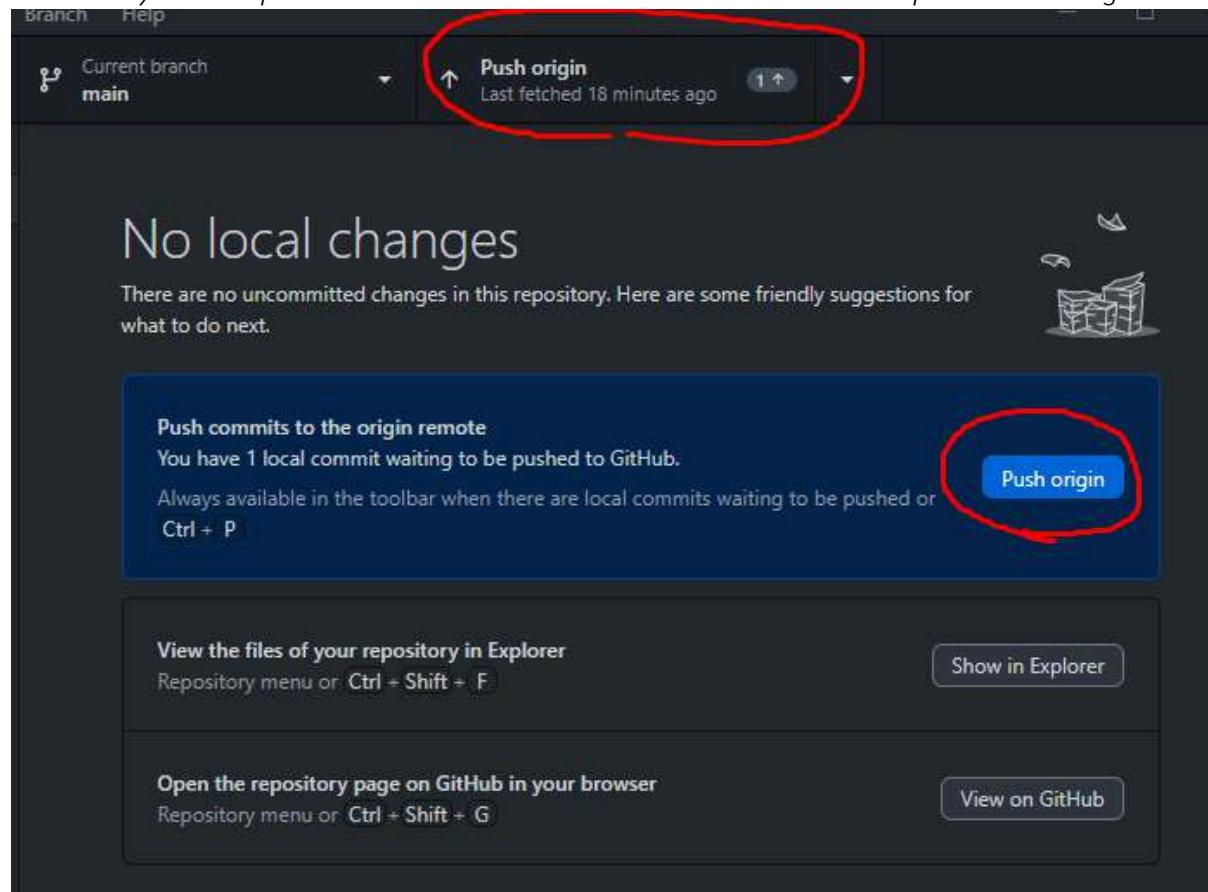
R+

Commit to main

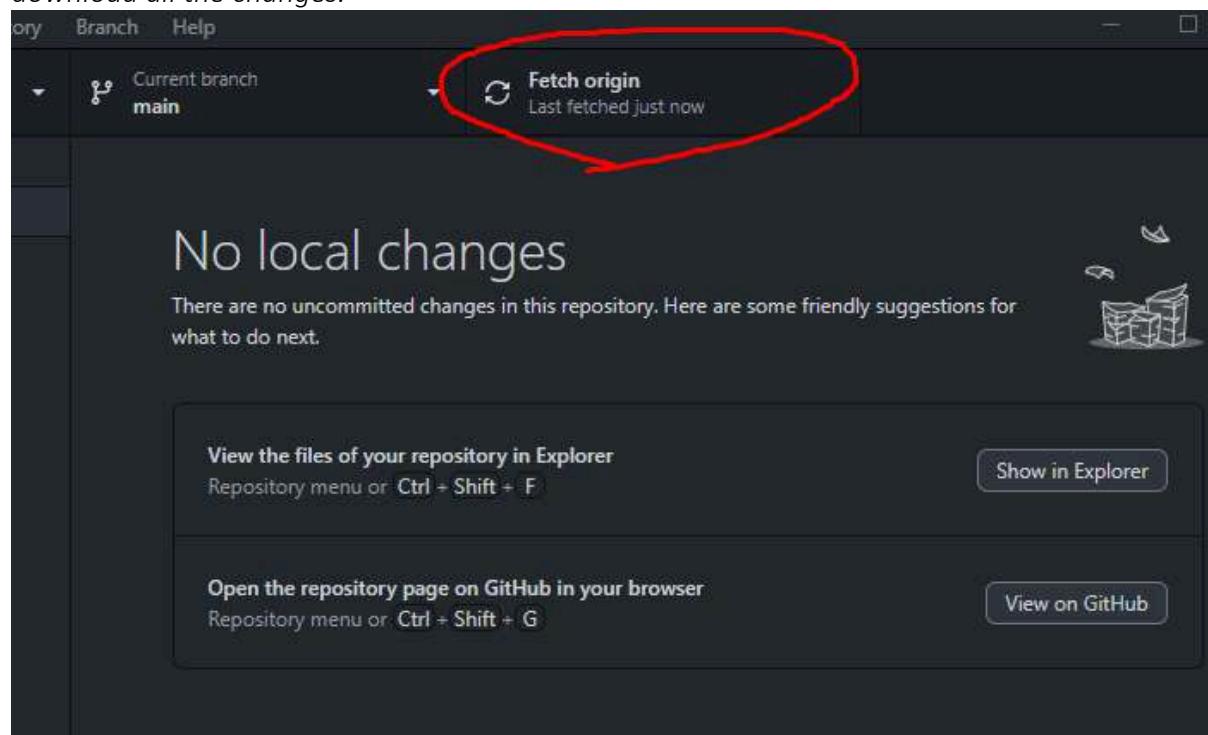
The screenshot shows a dark-themed Git commit interface. At the top, it displays the current repository as "VR-T12023" and the current branch as "main". Below this, there are two tabs: "Changes" (which is selected) and "History". Under the "Changes" tab, it indicates "1 changed file". A list of files shows "test.txt" with a checkmark next to it. On the right, the content of "test.txt" is displayed with a single line: "+This is a test for GitHub". Below the list of changes, there is a large, semi-transparent red circle highlighting the bottom-left corner of the interface. Inside this highlighted area, there are three input fields: "Create test.txt", "Description", and "R+", each with a small icon to its left. At the very bottom of the interface is a blue button labeled "Commit to main".

Usually you have to write a small summary above the Description for the “Commit to main” activate.

Now that you have pressed the “Commit to main” button. You have to press “Push origin”



To receive a change from other members. You have to press the “Fetch origin” button and download all the changes.



Important Note: Always “Push origin” and “Fetch origin” all the time for everyone to be up to date with the files. Otherwise “Conflicts” within files may appear that can set back team’s project

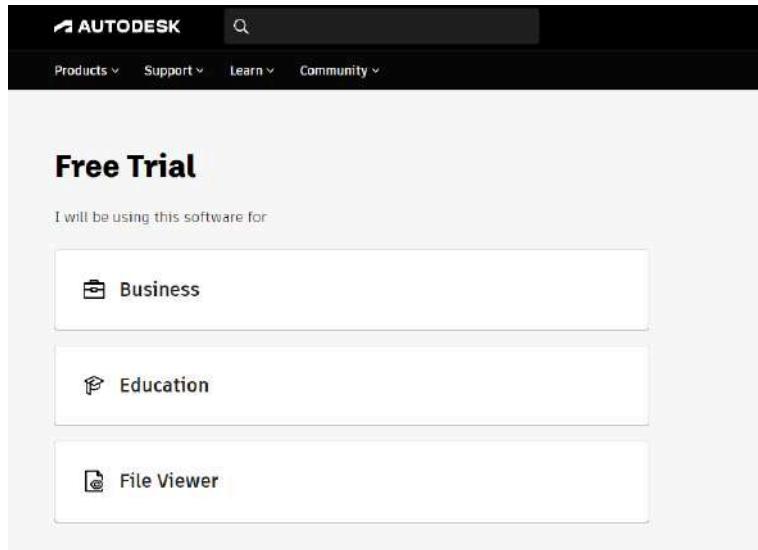
60.17. 3.5. Blender / Maya / 3D Max

Note: this section contains biased information based around the 3D Max software. It's your choice however for which 3D software you decide to use.

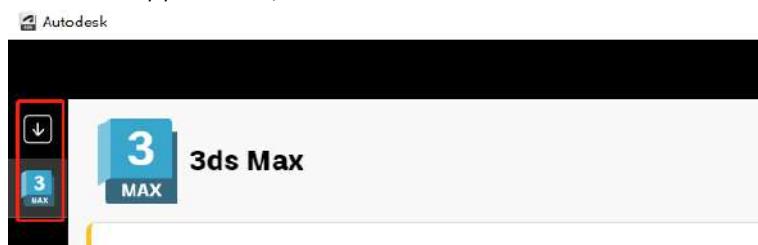
This section will contain basic information on how to use a 3D software application to create and export an asset.

60.18. 3.5.1 Installation and setup 3DMAX

1. Create an Autodesk account if you do not have any, go to 3DMAX website and create a new account.
2. You can get a Free trial if you are a university student and did not use before, then choose the operating version and download Autodesk.

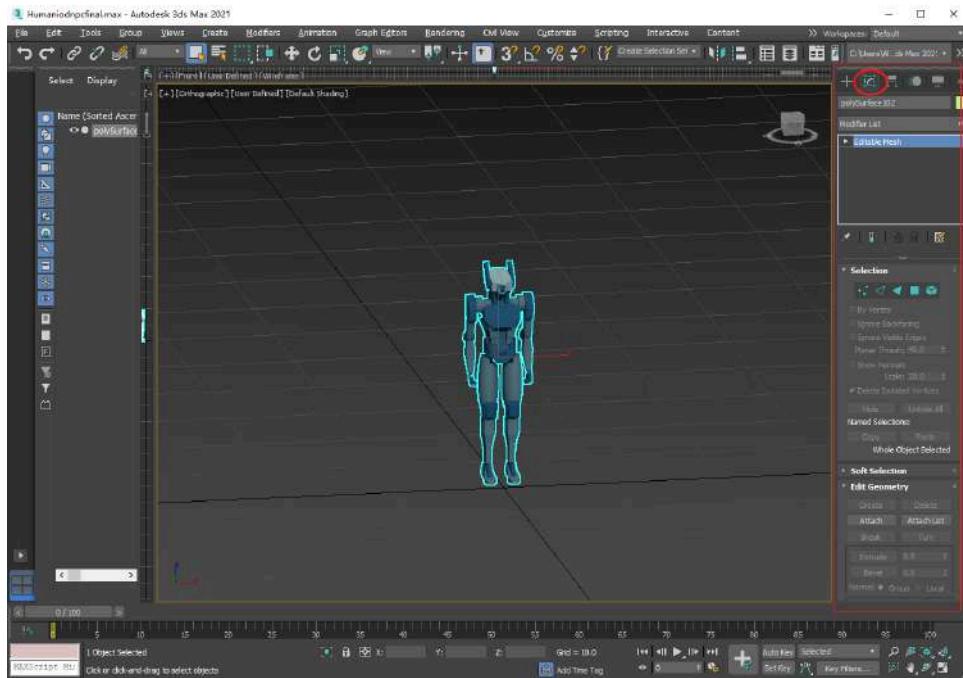


3. Open the Autodesk application, download the 3DMAX tool.

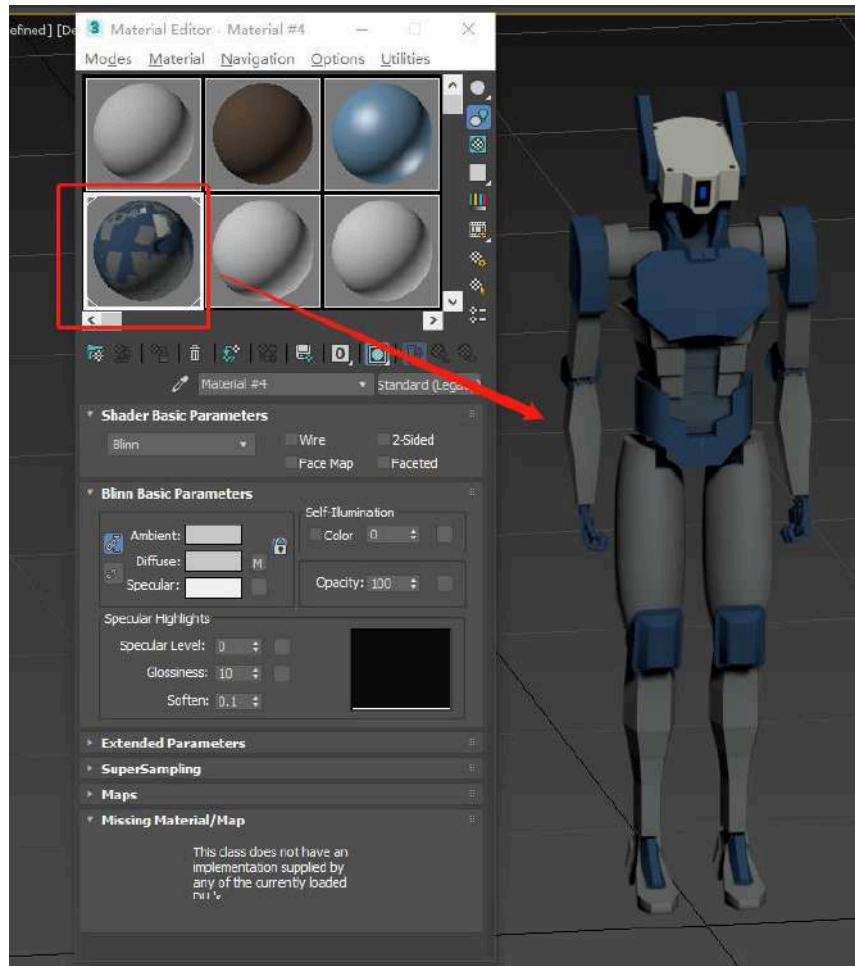


4. Open the 3D MAX software and use the tools provided to create your model.

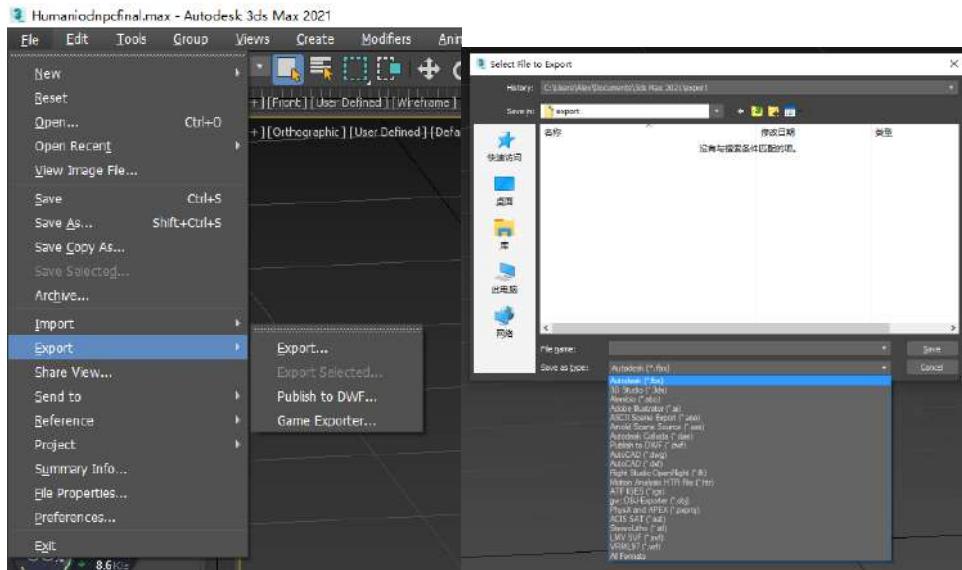
5. You can then modify the model through the edit bar on the right while observing the model, but please note that model modification will affect the application of textures



6. Click the shortcut key "M" to open the material editor, view the texture and apply the model by dragging the shader.



7. Export the model and convert it to any format you want.



8. Store the exported file somewhere easily accessible by the Unity Team (Microsoft Teams Files) who can then import your model into the project.

61. 4. Deliverables

Trimester Deliverables	Status	where
Added GitHub for better accessibility	Done	GitHub: https://github.com/redbackoperations/VR-T12023
2 new 3D Models for Bicycle option	Done	Github
3D models (objects) for the Desert Map	Done	Github
Better Organization of Github Files	Done	Github
New map called "City"	Done	Github
3D Model of the City	Done	Github
User is able to choose a bicycle	Done	Github
User is able to choose a Map	Done	Github
End trigger	Done	Github
Timer	Done	Github
Collision	Half-done	Github
Movement	Half-done	Github
Add Bonus star to increase timer	Not done	N/A
Create a Speedometer	Not done	N/A
UML settings/controls	Done	Github

62. 5. Open Issues

- Issue that will come again and again: Merging scene's can bring conflicts. The reason is that two users use the same "asset/model" to do different things. That brings a pandemonium to the scene that can't be resolved. To resolve it, one scene must be accepted as the current one. The second scene that has the changes needs to be put

back one by one to the first scene. Put back the scripts/models/assets etc. at the hierarchy.

- *Desert map has collision issues*
- *City map the collisions aren't fully implemented*
- *Setting is not functional*
- *Quit is not functional*

63. **6. Lessons Learned**

Our team started working on a Unity game development project during this trimester, and as a result, they learned a lot. Following are some of the most important takeaways from the project, along with suggestions for incoming teams:

Managing the scope:

Throughout the course of the project, we came to understand the value of efficient scope management. When we first started, our team had big goals and wanted to include a lot of features and functionality. But as time went on, it became clear that our scope was too broad, which limited our options and reduced the quality of our work.

Future teams should give careful consideration to realistic scope planning and early project goal setting. It's crucial to establish the fundamental gameplay and features in accordance with the time and resources available. As a result, the development process can be more narrowly focused, and the intended quality requirements are met by the finished product.

Optimization and Performance:

As our project was ongoing and expanded in complexity, we ran into performance problems that impacted the overall smoothness and enjoyment of the game. This was mostly caused by ineffective resource management, inefficient code, and a lack of early performance consideration.

Future teams should start the project by giving optimization and performance considerations a top priority. This entails putting in place effective algorithms, controlling memory use, and profiling the game to find and fix obstacles. Early speed optimization can help teams avoid laborious refactoring later on and guarantee a smooth gaming experience.

Sharing of knowledge and documentation:

We came to understand the value of thorough documentation and team members exchanging information. Confusion and inefficiencies resulted from our initial failure to properly explain critical design choices, code structure, and project guidelines.

The culture of documenting and information sharing should be established by future teams. In order to do this, it is necessary to keep technical requirements, architectural blueprints, coding standards, and user manuals and guides current. Future teams will gain from the quicker onboarding of new members and easier collaboration during various project phases.

Issue with User Interface (UI) Design:

Initially, we undervalued the significance of usability testing and UI design that is intuitive. As a result, there was user confusion, poor navigation, and a poor user experience. Future teams must allocate enough time for usability testing and UI design. It is essential to design a user interface that is both aesthetically pleasing and efficient at guiding players through the game. Usability testing with representative users can assist identify navigation issues, highlighting problems with users, and enhancing overall UI design.

Regular Knowledge and skill development:

We as a team came to the conclusion that creating games with Unity is a lifelong learning process. We faced difficulties that compelled us to go beyond our original capabilities in terms of knowledge and skills.

Future teams should actively participate in ongoing skill development and learning throughout the project. This can entail studying online guides, taking part in Unity forums and communities, watching tutorials or doing online courses that can help them, and encouraging team members to share their knowledge. Teams may overcome obstacles and produce high-quality games by keeping up with new Unity features, industry trends, and best practices.

Issue of loading assets into Scene:

Managing game elements, such as music, sound effects, and art assets, proved challenging.

Keeping track of asset versions, ensuring consistency in style and quality, and effectively integrating them into the game all become difficult tasks.

Future teams ought to put in place a formal asset management framework. This entails classifying assets into logical folders or categories, adhering to consistent naming practises, and managing assets using version control systems. Teams could also think about utilising the asset management solutions found in the Unity Asset Store to speed up the procedure and guarantee effective teamwork.

64. 7. Product Development Life Cycle

1. New Tasks

Each week we would meet as a team to discuss either current tasks or possible new ideas, anyone who isn't there can watch the recordings. Within the meetings we would go over our trello board that has all our tasks listed, each week we talk about new features we could add and take a vote on if they are good ideas to add or not.

Each person who comes up with the idea adds it to trello and adds the names of the people to work on it with them.

2. Definition of Done

Each task is designated to a team member or a number of members, each member can check the trello board for tasks that need to be done. Once a task is finished we show it to the rest of the team in either our weekly meetings or one on one chat, once we agree it's done to an acceptable level we can change the task in trello into the done category.

A completed task needs to work as intended, and that the feature is tested by others.

3. Task Review

Once a task has been said to be finished, the member/members working on it will post a message announcing the completion of the task. Typically someone would set a meeting with that person and get them to show that the new feature is working as intended before we push it in the repository.

4. Testing

When we talk about a new feature we have a level of understanding of what it is meant to do and look like. Each developer does their own testing of the feature to make sure its up to par, after the developer tests it we pass it to the team to have them test and see if anything needs fixing or needs to change.

5. Branching Strategy

As a team we use the repository to work more seamlessly as a team, keeping all our files for the project within the repository for ease of access and to keep organised.

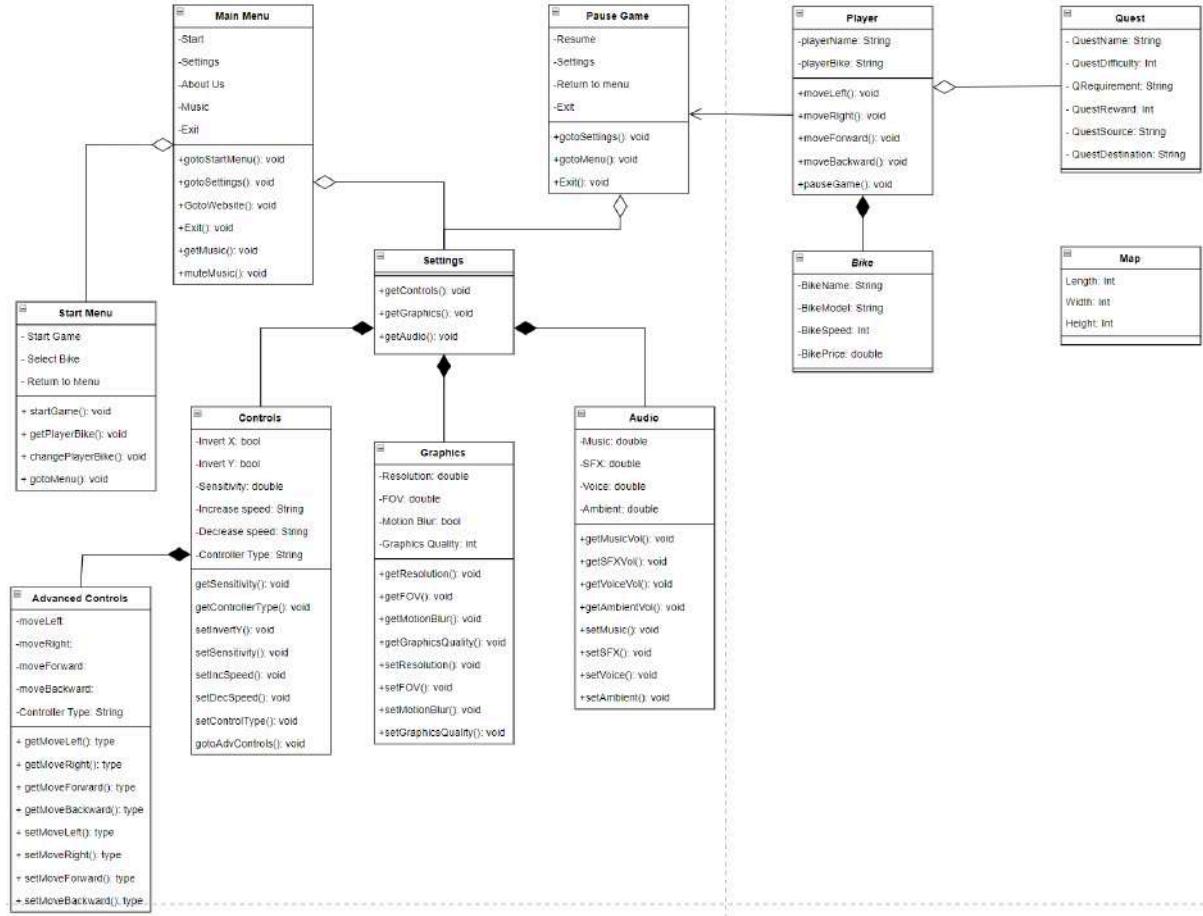
Commits have to be looked over to make sure you're not deleting something very important. Besides that, we don't have any set rules for committing or pushing.

Each person has to name their changes for exactly what they have done, we try not to do the edit in multiple commits, we want changes to be in one commit to keep it cleaner.

After working with the repository I would suggest that the new team have the main branch for project work, but create sub branches for the extras of the project that might not need to go inside the actual unity file.

65. 8. Product Architecture

1. UML Diagram



2. Tech Stack

Unity Editor 2020.3.37f1:

A powerful game engine tool that allows users to build and develop gaming applications, which can be useful when creating games within a 2D or a 3D environment. This version is used due to its stability, and to its relatively low minimum requirements.

Draw.io:

A free and open source graphic designer used to create diagrams, flowcharts and UML diagrams.

This program was chosen due to its ease of use, its accessibility and availability.

Figma design:

A collaborative program for designing interfaces for our project.

This program is used as it provides an easy way of creating visualisations and is collaborative, allowing multiple users to contribute.

Github:

An application that provides version control for projects in development, allowing users to continuously update a project.

This is used as it is best for software development for both in person and online. It is also collaborative as multiple people can update files within a project.

Blender:

A strong open-source 3D creation tool is Blender. It is frequently used for many different things, such as video editing, simulation, rendering, and 3D modelling. Blender is a flexible tool for animators, designers, and artists since it supports a large variety of features and functionalities.

Maya:

The entertainment business makes extensive use of Maya, a potent 3D computer graphics programme, to produce stunning 3D models, animations, simulations, and visual effects. It is created and maintained by Autodesk, a top software provider for computer-aided design and the production of 3D content.

66. *9. Appendices*

66.1. *9.1 Showcase Video*

Showcase Video:

[showcase.mp4](#)

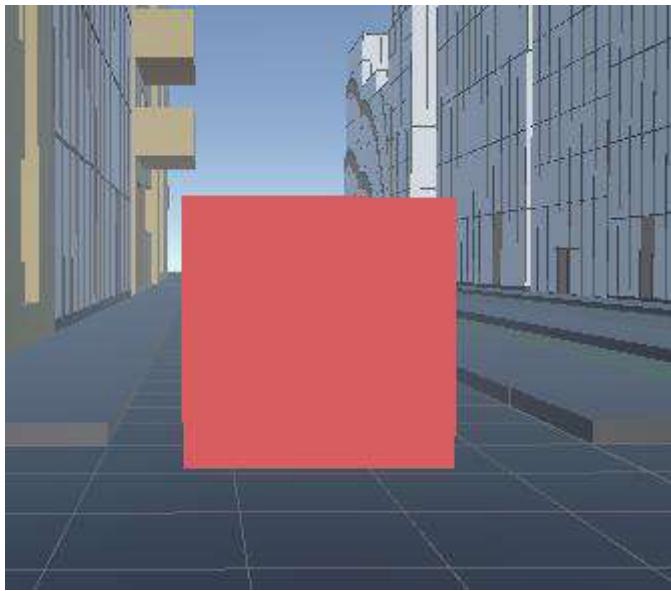
66.2. *9.2 Documentation Deliverables*

Organization of GitHub files

Name	Date modified	Type	Size
📁 \$_About REDBACK Company	24/05/2023 10:51	File folder	
📁 .git	27/05/2023 20:38	File folder	
📁 3D Objects	24/05/2023 10:51	File folder	
📁 Figma designs	24/05/2023 10:51	File folder	
📁 Handover Documents	24/05/2023 10:51	File folder	
📁 Object List for scene ref	24/05/2023 10:51	File folder	
📁 UML Diagram	27/05/2023 19:00	File folder	
📁 Unity	24/05/2023 10:59	File folder	
📄 .gitattributes	11/05/2023 19:28	Text Document	2 KB
📄 .gitignore	6/05/2023 21:01	Text Document	2 KB
📄 Download only me & READ ME FIRST.md	24/05/2023 11:07	MD File	1 KB
WORD GitHub how to upload 100mb+.docx	24/05/2023 11:22	Microsoft Word D...	262 KB

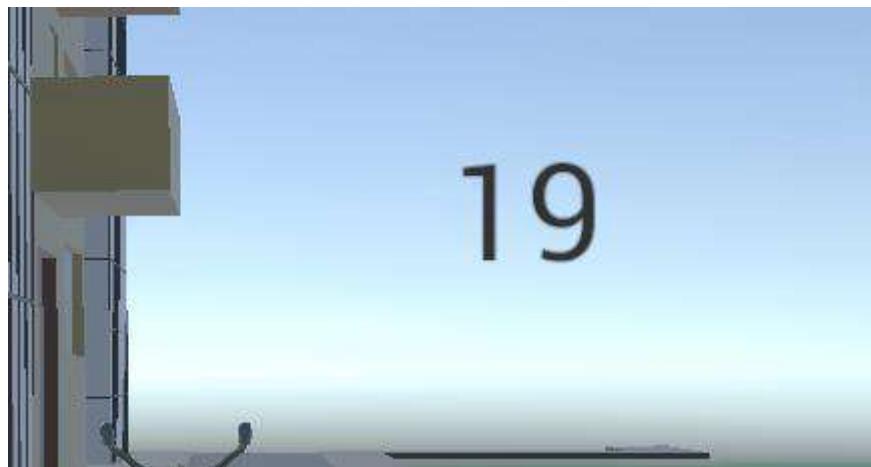
66.3. *9.3 unity Deliverables*

End trigger



LEVEL
COMPLETE

Timer



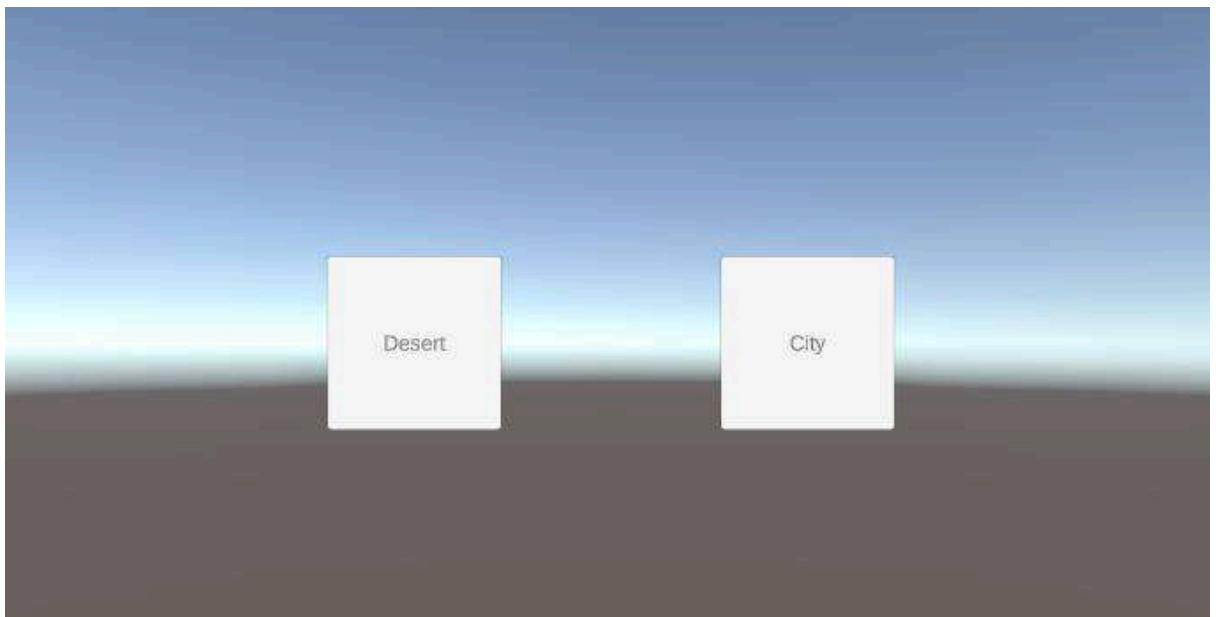
Mission **FAIL**

Bicycle selection





Map Selection



City Map



3D assets



<i>Name</i>	<i>Content</i>
<i>Steven Makris</i>	<i>1,2,3,4,5,8.2(blender,Maya), 9</i>
<i>Daniel Isla</i>	<i>8(UML, Unity, Draw.io, Github, figma)</i>
<i>Hilal Irshad</i>	<i>Section 6, (unity and programming)</i>
<i>Nathan</i>	<i>7</i>

68. Project Team

Project Name: RedBack Design and Development

Team: WebDev and Design Team

Avinash Maurya	Main Team Lead
Enock Nyaundi	Team Member
Yiannis Doulgerakis	Team Member
Jack Cousens	Team Member
HAN XIA	Team Member
Brian Dang	Sub Team Lead - Marketplace
Anno Gomes	Team Member
Hokky Kurnia	Team Member
HARSH H PATEL	Team Member
Nicholas James Stavropoulos	Team Member
Mohit Jurani	Team Member
Nicolas Andres Tomas	Team Member
Evan Mitropoulos	Sub Team Lead – Mobile Design
Ben Stephens	Sub Team Lead
Qianhui Meng Meng	Team Member
Robert Nicolas Bajan	Team Member
Yesitha Henagama Liyanage	Team Member
Dio Kyrou	Sub Team Lead - Website

69. Project Overview

The project aim towards making the existing UI/UX design more user friendly, and device optimised, even across all applications, and creating and developing new features and UI/UX of the website and mobile app pages. The project was able complete the set goals by implementing better UI/UX designing to the existing structure for both web and mobile applications. Moreover, new pages and features were designed, like night mode, marketplace pages for app etc. The existing webpages were optimised and were made device friendly, and all the existing pages across all our applications were changed to reflect evenly. Moreover, the Marketplace website was made, and the foundation has been laid for the upcoming cohorts to develop on it. Other major tasks involved research UI/UX design, make low fidelity designs of the website and the mobile application, make a design flow to show how the website and the application would execute and then make the final high-fidelity designs of the website and the application. With project 4, do some upskilling by practising making a website and functionalities through code and then once most of the web designs are finished, start making the web pages of the Redback website.

70. User Manual

The Figma board can be found here:

<https://www.figma.com/file/tvpePld4wuKneyD9faU1yw/Mobile-App-Flow?type=design&node-id=743-239>

Showcase video can be found here:

https://www.youtube.com/watch?v=78Gabb9L2ho&ab_channel=nickko

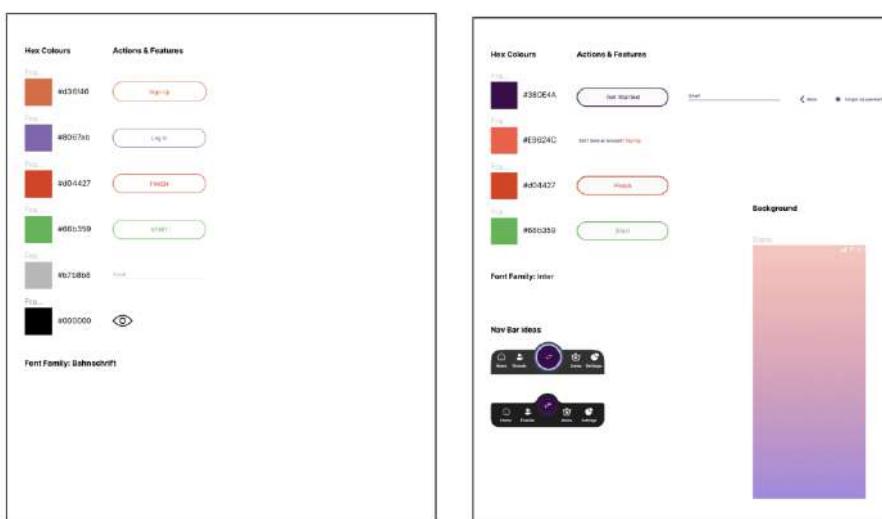
Main Webpage: <https://redbackoperations.netlify.app/>

GitHub Frontend: <https://github.com/redbackoperations/website-frontend>

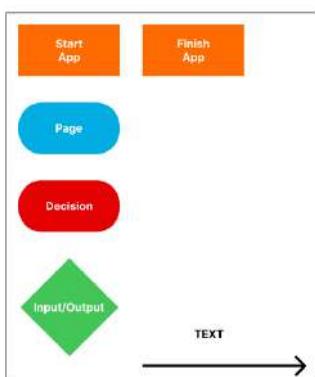
GitHub Backend: <https://github.com/redbackoperations/backend>

GitHub Book: <https://github.com/redbackoperations/Book>

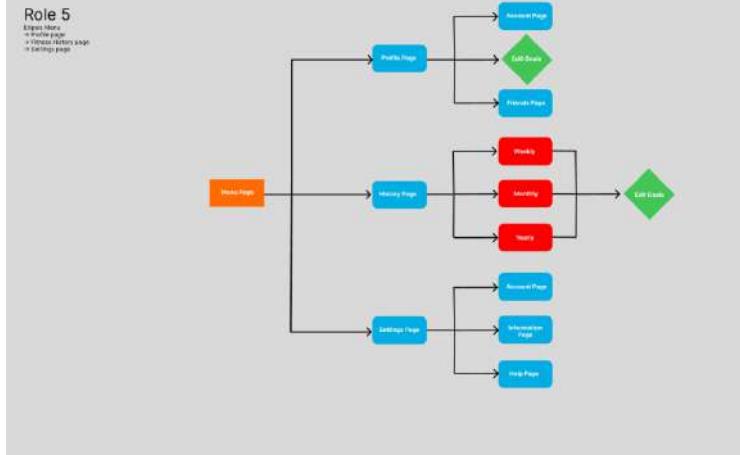
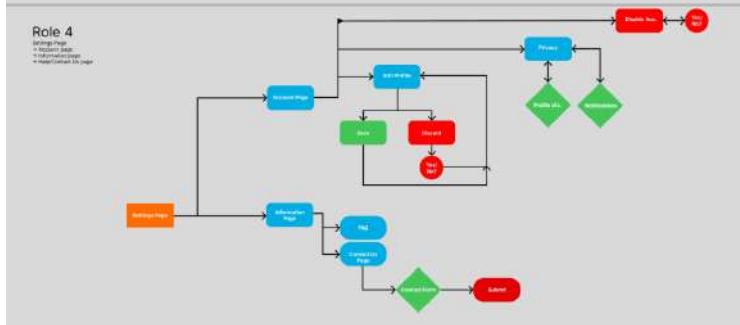
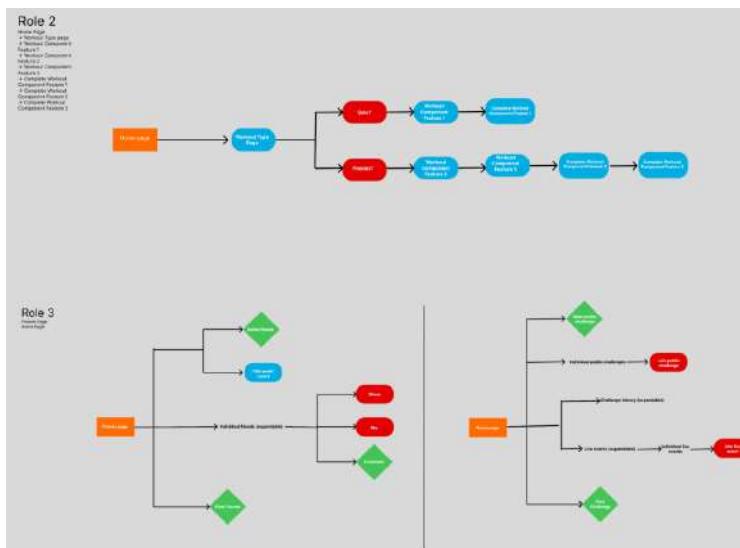
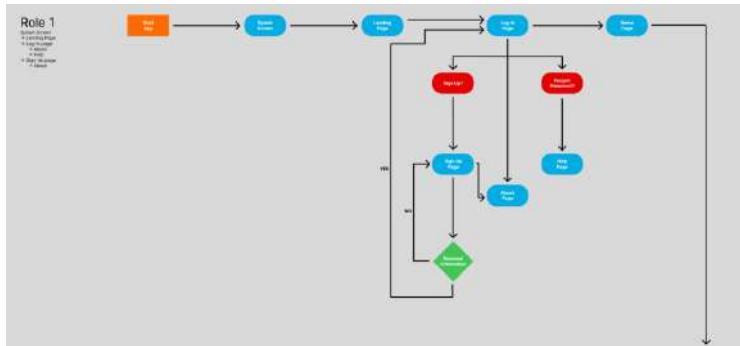
Moodboard:



Key:

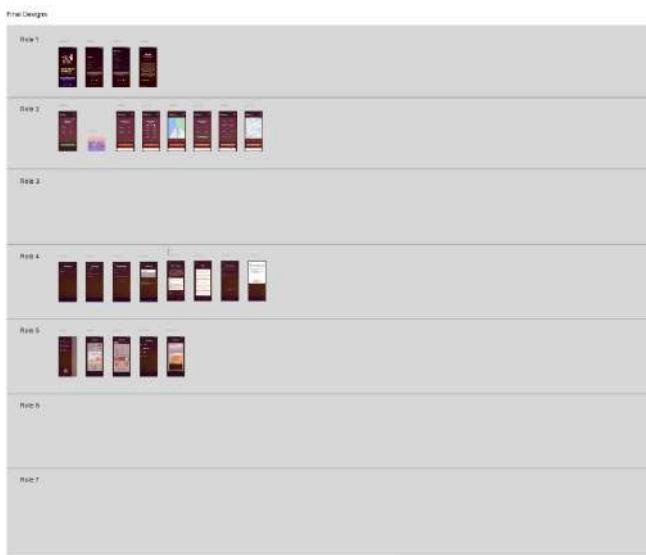


Flow:



New Night Mode:

Night Owl Theme



Book

71. Completed Deliverables

Provide a list of product features and/or deliverables, including a brief description, that have been completed this trimester. Please relate these deliverables to their corresponding Trello cards if this is possible.

Only include features and/or deliverables that are fully complete – incomplete work is to be listed in section 4. Roadmap.

Make sure to explicitly highlight which features and/or deliverables were completed this Trimester and which team member(s) were primarily responsible for their completion.

Also, please indicate where each of the completed deliverables can be found (E.g., MS Teams, GitHub repository) and make sure to include a URL link to the resource.

Completed Deliverables include:

Optimized UI/UX - Website and Mobile Application: All the webpages and mobile app pages were revamped to reflect the same across all the platforms. The UI/UX designs were improved and made more user friendly and efficient.

RedBack History Book: A better version of a “Redback” book has been created to reflect the progress of the company, its team, and its projects throughout the cohorts. It works like a normal notebook, and things can be added to it as a backend functionality has been implemented.

Marketplace Mobile App Design: The marketplace designs have been created for the mobile

developers to build on.

Marketplace Website: The marketplace website has been developed to a base level and is ready for deployment to go live.

Optimized Website for different devices: The existing webpages have been made device friendly (mobile friendly).

Night Mode feature: A new night mode feature was designed by the team which has been integrated into the mobile app.

Main Webpage: <https://redbackoperations.netlify.app/>

GitHub Frontend: <https://github.com/redbackoperations/website-frontend>

GitHub Backend: <https://github.com/redbackoperations/backend>

GitHub Book: <https://github.com/redbackoperations/Book>

Figma: <https://www.figma.com/file/tvpePlD4wuKneyD9faU1yw/Mobile-App-Flow?type=design&node-id=743-239>

Development Tasks:

Name	Task 1 – Optimise the webpages and make it device friendly	Task 2 – Update the pages to reflect the recent information	Task 3 – Update the progress and any recent updates of the company in the book.	Task 4 – Help in integrating the 2FA feature to our frontend	Task 5 – Create and Manage the Trello Board
Enock Nyaundi	Our Team	Our Team	NA	NA	NA
Yiannis Doulgerakis	Home page	Home	NA	NA	NA
Jack Cousins	About Us	About Us	NA	NA	NA
HAN XIA	Contact Us	About Us	DevOps	NA	NA
Brian	Our Team	Home	WebDev	NA	NA
Anno Gomes	Home Page	Our Team	Cyber Security	NA	NA
Hokky Kurnia	Our Project	Our Project	VR Team	NA	NA
HARSH H PATEL	Home	Contact us	Design team	NA	NA
Nicholas James Stavropoulos	Our Team	Our project	Data/AI team	NA	NA
Mohit Jurani	Our Project	Our Team	Mobile Team	NA	NA
Nicolas Andres Tomas	Oversea and Assist for all	Oversea and Assist for all	DevOps, WebDev &	Yes	Yes

	pages	pages	Design, CyberTeam, Cyber Team, VR Team, Data/AI team, mobile app		
Evan Mitropoulos	Oversea and Assist for all pages	Oversea and Assist for all pages	Yes	Yes	

Marketplace Website: Design and Development – Avinash Maurya
Task Allocation, Review, Feedback and Development, Skill Up Sessions – Avinash Maurya

72. *Roadmap*

The Marketplace Website needs to develop on as it is currently at a base level with limited functionality. A major update would be enabling multiple users to sell multiple items.

The RedBack book needs to be updated with all the major changes and to reflect the most recent journey at any given point. It needs to be embedded in the main website for public access as well. It needs to update cohort team members. The book has limited functionality. The Redback Website needs to be updated with recent changes and any modifications. Changes made this trimester needs to be developed on in terms of improving accessibility and reachability. The SEO of the website needs to be improved.

Design Team to be made redundant and merged with Dev Team. The developers will be responsible for both designing and developing.

73. *Open Issues*

Major issue that the team faced was regarding skilling up and knowledge transfer. The team needed to be given skill up tasks until week 8 so that everyone could be comfortable with the tech stack they were using. The Web Design team has learnt from this project is to create low fidelity designs to help us improve on our skills in UI/XU designs along with using Figma as a tool to use to create website designs which we all improved on overtime. What we would recommend for future teams to do differently is to plan what pages are required to make the website and then if people in future teams don't know how to do UI designs, then they should do research first on how to do UI/UX designs step by step and make low-fidelity designs as an upskill task to improve on their skills in UI/UX design.

74. *Lessons Learned*

Increase the number of skill up and knowledge transfer sessions, decrease the time spent on team allocation. Have more developers than designers, so the workload is well distributed.

75. *Product Development Life Cycle*

75.1. *New Tasks*

The goals for the trimester are set by the Main Team leader, who then creates sub team and sub team leaders for smaller projects for the same product. New tasks are created through having team meetings and everyone telling about their progress with their ongoing tasks to see if they have finished their tasks. If they have finished their tasks, then they can work on the new pages that needs to be created that I have thought of or noted down for organisation. The main team leader have also given us new tasks to do as well, which is upskilling and related tasks to each team. We form new ideas through discussing in team meetings about the process we have done and once that has been discussed, we think of new ideas on what we all should do for next week.

75.2. Definition of Done

A task is finished by stating it on the Trello board and marking the task as complete and also through team meetings where a team member would state if their tasks are finished, which then reviewed and ticked off by the team leader.

75.3. Task Review

In the Web Design team, the sub team leader reviews the tasks when its marked as done such the designs, the design flow and more. When the sub team leader thinks that the work looks good enough, the sub team leader would show the work to the main team leader of the overall web development and design team to assess if the designs are efficient enough for the web development team to create. We have a team meeting to chat about our work we have done. The sub team leader would ensure that he would look at all of the work and if its good enough, he would send it to the main team leader, and then the web development process would begin.

75.4. Testing

We would assess our designs to see if it looks quality and efficient enough to use them as confirmed designs for the Web Development team to develop on. The Web Design team would develop low fidelity designs as a plan on how the web pages would run with the functionalities required and how it would look like as well. The main leader and sub team leader review and verify the work for testing.

Team project Handover Document

Data and AI TEAM

Trimester 1, 2023

<https://github.com/redbackoperations/data-analysis>

Projects

- *User Ranking - Engagement*
- *FIT File Handling and Data Pipeline*
- *Corporate Reporting*
- *Sentiment analysis (language processing) and Community standards
User/Community comments*
- *Performance Ranking (User)*
- *Workout Categorisation*
- *Data Warehouse*
- *Google Analytics/Hotter Analytics/MixPanel/App Analytics (Marketing and UX)*
- *Posture Analysis*

76. Project Name: User Ranking - Engagement

Company: Redback Operations

Team Members: Saeed Alnaqeeb (Lead)

77. 1. Project Overview

In general, this is a ranking system project that make use of the data produced from rides to evaluate riders' engagement and based on that, creates a competitive fun environment by ranking them. Each rider can either earn or lose points, depending on how they engage within the application and how regularly they ride. In this project, we aim to analyse the produced data, use that to build the system, and analyse users progress by building machine learning models. The deliverables of this project consist of a brief analysis extracting data that relates to users' engagement and a ranking system based on points.

78. 2. User Manual

This project is currently being developed using Python and can be navigated by viewing the created files under project 15 of the Data/AI team repository on GitHub. There will be two files demonstrating the approach followed while working on this project, a python file containing analysis of the sample dataset, and another python file where the system's algorithm is developed. Also, a folder that holds the documentation of the project.

Available at:

https://github.com/redbackoperations/data-analysis/tree/main/Trimester_1_2023/Project%20User%20Ranking%20Engagement

79. 3. Completed Deliverables

- *Data analysis of users' activities and finding produced data do be used for evaluating a user's score.*
- *Ranking Criteria definitions.*
- *Algorithm that assigns points to users based on their engagement.*
- *Produced dataset of the users with their score and rank*
- *Post-ranking analysis to find patterns and trends for future ML implementation*

All available at:

https://github.com/redbackoperations/data-analysis/tree/main/Trimester_1_2023/Project%20User%20Ranking%20Engagement

80. 4. Roadmap

Roadmap for Trimester 2, 2023:

- *More analysis on the produced dataset*
- *Search machine learning models to implement*
- *Build different ML models for user progress prediction and projected path*

81. 5. Open Issues

- *The system needs more testing, as it was only tested on a sample dataset.*
- *Not yet implemented on the original dataset.*
- *Ranking criteria can be modified in the next trimester.*

82. 6. Lessons Learned

One of the main lessons learned is that the continuous use of Trello board would absolutely help organize both individual and team's work. Recommended technology for future machine learning models is skikit-learn library in Python.

83. 7. Product Development Life Cycle

83.1. 7.1. New Tasks

We do have two weekly meetings where we discuss the ongoing work. One of the meetings is at the start of the week which is a stand up meeting to present your plan for the week, and another meeting later for the progress and any encountered issues.

83.2. 7.2. Definition of Done

Work is considered done if it is reviewed with the lead and committed successfully to GitHub.

83.3. 7.3. Task Review

Tasks are generally reviewed by the team lead.

83.4. 7.4. Testing

In this project, all the work and testing were on a sample dataset, so I runned different methods to test if the product works as it intended to work or not.

83.5. 7.5. Branching Strategy

In GitHub, every member commits their work to a forked repository of the team's repo. With each update, we create a pull request so that it gets reviewed and branched to the main repo.

84. 8. Product Architecture

84.1. 8.1. Tech Stack

- Python: analysis, visualisation and algorithms
- SQL: Querying data
- GitHub
- Trello Board

85. 9. Source Code

All available at:

https://github.com/redbackoperations/data-analysis/tree/main/Trimester_1_2023/Project%20User%20Ranking%20Engagement

86.

87. 10. Appendices

Documentation of the project and approach along with some important files are available to view on the project GitHub repository:

https://github.com/redbackoperations/data-analysis/tree/main/Trimester_1_2023/Project%20User%20Ranking%20Engagement

Showcase Video:

https://video.deakin.edu.au/media/1_blu17zp1

88. Sentiment Analysis

Redback Operations

89. 11. Project Overview

SunCycle users have the option to leave comments on each other's activities as means of increasing engagement. To ensure that the comments are appropriate and to keep track of language usage, this project uses machine learning tools to analyse comment data and classify user comments.

The goal of the project is to build and maintain a safe, friendly environment by creating community guidelines, performing sentiment analysis, and monitoring users' comments. To help the company compete with other game developers in the market, it is very important to have a good environment and improve user experience from every aspects. More specifically, by preventing users from leaving toxic comments, this project helps protect users, foster inclusivity and diversity, and gather user feedback.

Deliverables include:

- I. *Community guidelines.*
- ii. *Data sets for training purposes.*
- iii. *Machine learning models that can classify user comments.*
- iv. *Showcase video*

90. 12. User Manual

This project is in the development phase. Currently the RNN model can classify a dataset with 1 label and 3 category values and achieve an accuracy of approximately 90%, and the DistilBert model can classify a dataset with 6 labels and achieve an accuracy of approximately 70%. Depending on the future requirement of Redback Operations, more accurate models might need to be constructed.

This project is fairly easy to use. When the model is finished, it will be added to SunCycle website as a new feature and will be used to process posted comments everyday after extracting them from the platform's database. The model is expected to analyze the data and produce a dataset with users' comments and their corresponding sentiment analysis result (positive/negative). Analysts can then use the dataset to monitor user comments.

91. 13. Completed Deliverables

Deliverables:

- I. *RNN model:*
 - a. *Completed by Yvette Liang*
 - b. *A deep learning model that can classify positive, neutral, negative comments and achieve an accuracy of 90%.*
 - c. *Github link: https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2014%20Sentiment%20analysis/NLP_06_RNN-copy1.ipynb*
 - d. *Finished on May 11 (Trello)*
- ii. *DistilBert model:*
 - a. *Completed by Yvette Liang*

- b. A deep learning model that can classify a toxic comment into one of six labels and achieve an accuracy of 70%.
- c. Github link: https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2014%20Sentiment%20analysis/NLP_08_DistilBert-copy1.ipynb
- d. Finished on May 11 (Trello)

iii. Community guidelines:

- a. Competed by Yvette Liang
- b. Simple community guidelines that can be posted on SunCycle's platform for users to follow.
- c. Attached in the appendix.

92. 14. Roadmap

1. NLP basics: Learned the basics of natural language processing and constructed the first **Logistic Regression** model on a dataset downloaded from Kaggle.
2. Researched into and construct NLP models: Finished researching; Constructed two models (**Multinomial Naive Bayes** and **SVM**).
3. Researched into and construct multiclass classification: The **Random Forest** model is complete.
4. Researched into and constructed **Stanford CoreNLP**.
5. Documented process, compared results, tried to improve accuracy: **RNN**
6. Transformer model on multilabel classification: **Bert** and **DistilBert**.

93. 15. Open Issues

- I. Software compatibility issues: All the models are developed on Google Colab. Team members need to be careful when they use the code locally or run in a different system.
- II. Trello board: This board lists general topics that the project focused on in a certain period. If members want to check more specific details, they need to click on the cards and read the comments.
- III. The modeling process: The models are moving toward the field of deep learning. Members might need the corresponding skillset to develop more accurate models.
- IV. Project requirement changes: Both the data (languages other than English) and the company requirement (classification threshold) may change in the future, members need to work on data preprocessing when they have new data come in.

94. 16. Lessons Learned

1. Machine learning concepts: When I worked on NLP models, one of the biggest problems is the large dimension of data. It is very challenging to select the right corpus. It is also required the team member to master the machine learning concepts. For example, I should have tried other dimensionality reduction techniques

other than PCA, especially when I realized that method required linear relationship of data.

2. *Create fast models: Users' comments can sometimes be very lengthy, and the dataset can get very large, so it is very important to keep track of a model when it processes data. If members want to use a loop, make sure to print something at the end so that they know the amount of data left to be processed. Google Colab sometimes can stop automatically if the user does not edit the webpage, so it is important to always keep an eye on the code.*
3. *Language issues: Members should familiarize themselves with people's language usage online to better preprocess data and tokenize them. For example, as a student whose English is not the first language, I could only create simple functions to preprocess user comments. I focused more on different machine learning models, and I realized I might have done a much better job if only I knew how people communicate online.*

95. 17. Product Development Life Cycle

The product development life cycle for this project is the agile process flow: moving through concept, inception, iteration and construction, release, production, and retirement.

Currently, as the model is still in the phase of iteration and construction, the introduction here is focused on the explanation of my workflow.

As there is limited computational capability in a personal laptop, I started from simple models such as Logistic Regression. As I learned more about machine learning, along the track of our unit progress, I could try more complex models and more feature engineering techniques such as Multinomial Naive Bayes, SVM, and Random Forest.

I created 4 models but then I realized their limit on improving accuracy. Therefore I started to try multi-label algorithms. I read about other people's articles online and tried Stanford CoreNLP model. I also tried deep learning models such as RNN, which was able to classify positive, neutral, negative comments and achieve an accuracy of 90%. I then moved toward multilabel classification. Bert was slow, so I constructed DistilBert, which could classify a toxic comment into one of six labels and achieve an accuracy of 70%.

Overall, the workflow during the construction phase is moving from simple to complex models, directed by the need of increasing accuracy.

95.1. 17.1. New Tasks

In this project, tasks are created based on three rules: member's current skillset, model accuracy, and onTrack tasks (documents and presentations).

We have team meetings every week and each member would discuss about the current progress, ask questions, and offer help. For this project, tasks are goal-oriented. A team member needs to have adequate knowledge and skills to work on NLP models, so it is necessary to research and improve skill. The models should be effective, so accuracy is an

important measurement to create tasks when necessary. OnTrack tasks are required in this unit, so they are also on the task list.

95.2. 17.2. Definition of Done

- I. A classification model is done when it produces an accuracy score that is high enough. In this case, RNN and DistilBert can be called completed.
- II. A code file is done when it can be run all the way down, with comments, and with accuracy scores at the end. It is finished when other team members can understand it easily.
- III. A report is done when it is submitted and accepted by the instructor.

95.3. 17.3. Task Review

As there is only one person working on this project this trimester, this person alone looks for dataset, writes code, and reviews it before uploading to Github. Team leader of the data science team would review it before merging. The code uploaded have all been checked and made sure that there is no confusion. Comments are added so that future team members understand the documents.

95.4. 17.4. Testing

During the iteration and construction period, testing is done after training the model. In the future, the testing can be done by letting the model produce a data table, with users' original texts on the left-hand side and sentiment scores / label on the right-hand side. This way, our team members can take a sample and check how accurate the score / label is in terms of categorizing the corresponding texts. If the score is not correct, then they can improve and update the model.

95.5. 17.5. Branching Strategy

Currently, there is only member pushing files to the origin. In the future, team members should use team chat to share their work and revise a final copy, and designate a person to submit the final copy and push it to the origin.

96. 18. Product Architecture

96.1. 18.1. UML Diagram

96.2. 18.2. Tech Stack

Google Colab: It has Jupyter notebook and relevant packages. It is flexible and allows multiple person to edit.

97. 19. Source Code

Github code:

https://github.com/redbackoperations/data-analysis/tree/main/Trimester_1_2023/Project%202014%20Sentiment%20analysis

Dataset:

<https://www.kaggle.com/datasets/kazanova/sentiment140>
<https://www.kaggle.com/datasets/charunisa/chatqpt-sentiment-analysis?select=file.csv>
<https://www.kaggle.com/datasets/julian3833/jigsaw-toxic-comment-classification-challenge?select=test.csv>

Other helpful information:

<https://scikit-learn.org/stable/modules/multiclass.html>
https://colab.research.google.com/github/stanfordnlp/stanza/blob/master/demo/Stanza_CoreNLP_Interface.ipynb#scrollTo=xiFwYAqW4Mss
<https://github.com/practical-nlp/practical-nlp-code/pull/38>
<https://huggingface.co/bert-base-uncased>
<https://huggingface.co/distilbert-base-uncased>

98. 20. Appendices

Showcase video link:

<https://youtu.be/BOQfavLfvAQ>

Community guidelines:

Maintaining a healthy and friendly online environment is essential for creating a positive and safe space for everyone. Here are some simple rules that the team can refer to when they draft community guidelines:

1. *No hate speech: Any comments that are racist, sexist, homophobic, or discriminatory are not allowed.*
2. *No personal attacks: Personal attacks and insults towards other individuals are not allowed.*
3. *No bullying or harassment: Any form of bullying or harassment is not allowed, regardless it's directed towards another user or not.*
4. *No trolling: Comments that are posted with the intent of provoking an emotional or negative response from other users are not allowed.*
5. *Stay on topic: Comments should be relevant to the topic being discussed.*
6. *No spamming: Posting the same comment repeatedly or promoting products or services excessively is not allowed.*

99. 21. Posture Analysis

Mark Tolley

Samuel Kamau

22. Project Overview

This project centers on a comprehensive, real-time posture analysis system for cyclists, designed to significantly reduce injuries and energy wastage resulting from incorrect form during cycling. Utilizing the power of Python with the OpenCV, PyQt, and MediaPipe libraries, the application encompasses three primary modules: pre-workout analysis, cycling analysis, and post-workout analysis.

The project aims to foster healthier cycling habits, enable more efficient training at home, streamline coaches' tasks in training clients, reduce posture-related injuries, and improve a cyclist's performance over time. It distinctively targets cyclists' needs, which sets it apart from more generalized posture analysis applications available in the market.

The key deliverables of the project are:

Real-time pose estimation and cycling posture analysis module: It provides real-time feedback on the cyclist's posture and form, allowing instant adjustments. It also includes a pedaling technique analysis and an aerodynamic analysis, which can be leveraged to optimize the cyclist's efficiency and performance.

Pre-Workout and Post-Workout Demonstration videos with real-time user tracking: This interactive feature empowers the cyclist to follow along with the demonstration videos while the system tracks and analyzes their form in real-time. This further facilitates more flexible training at home and aids in preparing for and winding down from cycling sessions.

Data and video recording of sessions: Each cycling session is recorded and stored both as video data and as quantitative data. This allows for tracking of progress over time, and helps to highlight areas of improvement.

Data visualization tool: A tool that presents the collected data from each session in an easily digestible format, such as graphs. This empowers cyclists and coaches to analyze the data and understand trends in performance and posture improvements.

100. 23. User Manual

Set-Up and Activation

Refer to the ReadMe file for a guide

UI

The user interface has been designed for ease of use, providing only essential navigation options to streamline the user experience. Just run the User Interface.py file after the initial set-up and you're good to go

101. 24. Completed Deliverables

The product's successful deliverables at this stage include the following components:

- An analyzer for pre-workout and post-workout activities
- The feature to log both data and video during workout sessions
- A function to erase recorded data when necessary
- A tool for visualizing user data

All aforementioned deliverables can be accessed at the product's GitHub repository: [GitHub Repository](https://github.com/redbackoperations/data-analysis/tree/main/Trimester_1_2023/Project%202020%20Posture%20Analysis)

102. 25. Roadmap

The projected pathway for the product encompasses the following enhancements:

- Integration of AR features for immersive interaction
- Utilization of edge detection for precise spine and aerodynamics analysis
- Automation to switch from demonstration mode to live feed upon detection of a human and bike
- Machine learning algorithms to detect user performance trends and offer suitable advice
- NLP implementation for real-time audio feedback from users
- Facial tracking technology to monitor user energy levels and emotional states

103. 26. Open Issues

One persistent issue hampers seamless project execution, specifically the Github Desktop's inability to access the project repository due to file naming compatibility issues. An interim solution has been implemented, utilizing Co-Pilot and a separate repository to track progress. Once significant modifications are complete and rigorously tested, the changes are then transferred to the main repository.

104. 27. Lessons Learned

The project has underscored the importance of various aspects like efficient project management, time management, detailed task division, incorporation of feedback loops for iterative product enhancement, and meticulous project planning. It is advised for the future teams to adhere strictly to the project planning to eliminate unnecessary time expenditure. Nonetheless, retaining a degree of flexibility is beneficial, given the unpredictability of certain issues.

105. 28. Product Development Life Cycle

The creation of new tasks is managed via Trello cards and the associated checklists. A task is deemed complete when it is free from both syntax and logical errors, and the outcome aligns with the pseudocode initially outlined. The solo nature of the project means code reviews are conducted by the developer, focusing on runtime error checks and functional tests. For testing, live, recorded, or downloaded video footage is utilized.

106. 29. Product Architecture

The project architecture comprises five primary classes:

- *Pose Analyser: This class is responsible for the real-time analysis and logging of cycling data.*
- *PoseComparison: This class compares stretching workout videos with user live feed, incorporating methods from Pose Analyser.*
- *UserVisualizer: It handles the visualization of user data and facilitates data deletion when required.*
- *Instructions: This class overlays instructional captions onto stretching videos.*
- *WorkoutMode: This is used by the UI to interchange between the two analysers.*

Additionally, a script named CDataset is present that converts an image folder into a standardized dataset of images, serving as a training set for the machine learning model.

The entire architecture utilizes Python with significant reliance on libraries including OpenCV, PyQt, and MediaPipe.

107.

108. 30. Appendices

108.1. 30.1. Software Dependencies

The following libraries and software are required to run and develop the application. Each of these plays a crucial role in the functioning of different aspects of the project.

TensorFlow: TensorFlow is an end-to-end open-source platform for machine learning. It is utilized in this project for processing intensive tasks and machine learning models.

OpenCV: OpenCV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library. In this project, it's used for image processing tasks.

MediaPipe: MediaPipe is a cross-platform framework for building multimodal applied machine learning pipelines. It's utilized in this project for real-time pose estimation.

PyQt5: PyQt5 is a set of Python bindings for The Qt Company's Qt application framework and runs on all platforms supported by Qt. It is used for creating the graphical user interface for this application.

PyDub: PyDub is a simple and easy-to-use Python library for audio manipulation. It is used for handling audio-related tasks in the project.

Seaborn: Seaborn is a Python data visualization library based on Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. It is used for data visualization tasks in this project.

Tkinter: Tkinter is Python's de-facto standard GUI (Graphical User Interface) package. It is used in this project for creating the user interface.

Pandas: Pandas is a software library written for data manipulation and analysis in Python. It is used in this project for data handling tasks.

Matplotlib: Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter. It is used in this project for data plotting tasks.

109. 31. Project 10 - Google Analytics/ Hotter Analytics/ MixPanel/ App Analytics (Marketing and UX)

1. Project Overview

We are using existing data from Google Analytics to create reports to study user's behaviour and get meaningful insights about their out-of-game engagement in order to create a feedback loop for product owners which will enable them to view and address product statistics, bounce rates and various other issues.

Aims for Trimester:

1. To formalise report on key data insights
2. Attempt to harmonise various data sources
3. To capture the behavioural & consumption pattern
4. To collect details of other products of same specification

Deliverables:

5. To formalise report on key data insights
6. Attempt to harmonise various data sources
7. To capture the behavioural & consumption pattern
8. To collect details of other products of same specification

Completed Deliverables

Used GA(Google Analytics) for a travel website to find the constitution of the in-flow of users, detailed statistics (acquisition, behaviour and conversions, etc.) of the site for various durations and their comparisons, demographic (Age, gender, etc.) of the users as well as the visualization of all the data from all the popular browsers.

[https://github.com/ktripathi04/data-analysis/tree/main/Trimester 1 2023/Project%2019%20App%20Analytics](https://github.com/ktripathi04/data-analysis/tree/main/Trimester%201%202023/Project%202019%20App%20Analytics)

2. Roadmap

1. Define Objectives and Requirements:
 1. Clearly articulate the objectives and requirements of the custom tool.
 2. Identify the specific data needs and gaps in the existing data collection methods.
 3. Determine the key metrics and insights required to address the task effectively.
2. Research and Data Exploration:
 4. Conduct research on available data sources and technologies that can provide the desired data.
 5. Explore different APIs, databases, or data providers that offer relevant data.
 6. Evaluate the feasibility, reliability, and cost implications of accessing and integrating the identified data sources.
3. Design Data Collection Strategy:
 7. Develop a comprehensive data collection strategy that aligns with the project's objectives.
 8. Determine the appropriate data collection methods, such as web scraping, API integration, data feeds, or data partnerships.
 9. Consider data privacy and compliance requirements while designing the data collection strategy.
4. Build Data Collection Pipeline:
 10. Implement the necessary infrastructure and tools to support data collection and storage.
 11. Develop scripts or code modules to automate data retrieval from various sources.
 12. Ensure the scalability, reliability, and security of the data collection pipeline.
5. Data Processing and Integration:

13. Clean, preprocess, and normalize the collected data to ensure consistency and accuracy.
 14. Develop data integration processes to merge the new data with existing datasets, if applicable.
 15. Implement data quality checks and validation mechanisms to identify and address any anomalies or errors.
6. Data Analysis and Visualization:
 16. Apply appropriate analytical techniques and algorithms to extract insights from the collected data.
 17. Develop data visualization components or dashboards to present the insights in a clear and meaningful manner.
 18. Enable interactive exploration and filtering of the data to facilitate deeper analysis.
7. Testing and Validation:
 19. Conduct rigorous testing to ensure the accuracy, reliability, and performance of the custom tool.
 20. Validate the results against known benchmarks or ground truth data, if available.
 21. Seek feedback from relevant stakeholders to identify any areas of improvement or fine-tuning.
8. Deployment and Maintenance:
 22. Deploy the custom tool in a production environment or integrate it into the existing workflow.
 23. Establish proper monitoring and error handling mechanisms to ensure smooth operation.
 24. Plan for regular maintenance and updates to accommodate evolving data sources or changing requirements.
9. User Training and Support:
 25. Provide comprehensive training and documentation to users on how to effectively utilize the custom tool.
 26. Offer ongoing support and address any user inquiries or issues promptly.
 27. Encourage user feedback and suggestions for further enhancements or feature additions.
10. Continuous Improvement:
 1. Regularly evaluate the effectiveness and relevance of the data collected and insights generated.
 2. Incorporate user feedback and iterate on the custom tool to enhance its performance and usability.
 3. Stay updated with new data sources, technologies, and analytical methods to continuously improve the tool's capabilities.
3. Open Issues
 9. Data was preset as per the designers view point and therefore the project had to be tailored according to the designers views.
 10. Limitation on acquiring the data as it wasn't enough to come to a conclusion.
 11. Lack of a team since the project is relatively new and I'm the only one working on it.

7. Lessons Learned

Proper Planning and Scope Definition

1. *It's important to clearly define the project scope, objectives, and deliverables at the beginning. Without a clear plan, teams may face scope creep, lack of focus, or unrealistic expectations.*
2. *Recommendation: Invest time in detailed project planning, including defining goals, identifying key metrics to track, and setting realistic timelines. Regularly reassess and refine the project scope to ensure alignment with the team's capabilities and available resources.*

Effective Communication and Collaboration

1. *Communication breakdowns can hinder progress and lead to misunderstandings. Ineffective communication among team members, stakeholders, or during panel presentations can impact project outcomes.*
2. *Recommendation: Establish regular communication channels, such as team meetings or project management tools, to ensure everyone is aligned, progress is shared, and challenges are addressed promptly. Practice clear and concise communication during panel presentations to effectively convey progress and insights.*

Data Quality and Preprocessing

3. *Inadequate data quality or improper preprocessing can lead to unreliable insights and inaccurate conclusions. Failure to clean and validate data can introduce biases or skew results.*
4. *Recommendation: Prioritise data quality assurance and establish robust preprocessing procedures. Implement data cleaning techniques, handle missing values, and ensure data integrity. Document data preprocessing steps to maintain transparency and reproducibility.*

Training and Knowledge Enhancement

5. *Lack of familiarity with Google Analytics or inadequate training can limit the team's ability to leverage its full potential. Insufficient understanding of the tool's features and capabilities may result in underutilisation or misinterpretation of data.*
6. *Recommendation: Invest in comprehensive training and knowledge enhancement on Google Analytics. Ensure team members have a solid understanding of the tool's functionalities, data interpretation, and analysis techniques. Encourage continuous learning through online resources, tutorials, and hands-on practice.*

Continuous Monitoring and Iterative Improvement

7. *Failing to monitor analytics data continuously and iterate on insights can limit the effectiveness of the project. Neglecting to review and act upon analytics insights in a timely manner may lead to missed optimisation opportunities.*
8. *Recommendation: Establish a process for ongoing monitoring and analysis of analytics data. Regularly review key metrics, track performance against goals, and identify areas for improvement. Encourage an iterative approach to implement changes based on data insights and continuously optimize the website's performance.*

8. Product Development Life Cycle

The team was tasked to find the footfall for the travel website .

Here we deployed the analytics tool to complete the given task.

The tool was deployed in the website to generate the data in a certain pattern.

We then compared the analytics report with that of another tool to ascertain the data's authenticity.

109.1. 8.1. New Tasks

The team meets twice a week to discuss the progress and assign new tasks to the members.

109.2. 8.2. Definition of Done

How does the team know when a task is done?

What are criteria for a successfully completed task?

This may seem obvious, but it in a software development project having a definition of done can ensure a certain standard of work that holds all team members accountable. For example, messy, clunky code that "just works" is very different to clean, well-commented code that works AND is easy to understand. Which would you prefer to be your team's definition of done?

109.3. 8.3. Task Review

Who reviews a task once it's been marked as done?

How does the team ensure that all work is looked over before it's contributed to the main repository or working prototype?

If you don't currently have a system for reviewing tasks, make sure to flag this for next trimester's team to work on as soon as they begin.

109.4. 8.4. Testing

How do you test your product to see if it does what it was originally planned to do?

If your product isn't heavily comprised of software, how can you build in testing to your team's product development life cycle to ensure that "stuff works as it should"?

109.5. 8.5. Branching Strategy

How does your team currently use GitHub repository?

What rules for commits and pull-requests have been put in place so far?

How should new members use GitHub repository in a way that doesn't result in all commits being dumped in a messy Master branch?

Again, if your team hasn't formally discussed a branching strategy, this a great opportunity to describe what your current system is and how it could be improved going forward.

For example, if you currently have all members of the team commit directly to the Master branch, can you recommend any tutorials for the future team to review that might lead to a cleaner, more organised and more efficient repository?

9. Product Architecture

109.6. 9.1. UML Diagram

In the given diagram, the website is connected to Google Analytics, representing the integration between them. The "Google Analytics" component further connects different components representing various aspects of data analysis and tracking, such as "Data Collection", "User Acquisition", "User Behaviour", "Conversions", "Duration and comparison analysis", "Demographic Analysis" and "Browser Visualization".

109.7. 9.2. Tech Stack

Google Analytics and a few scripts of python

10. Source Code

All source code should be found on your team's GitHub repository, unless your project has unique constraints that require you to store your code elsewhere. This includes any resources (e.g., wireframes, designs) that need to be transferred over to the new team as well.

Please provide all of the necessary instructions to accessing your source code. This includes URLs of online hosted repositories, links to any software dependencies, database components, or external libraries.

If your code is hosted on a server external to Deakin, make sure to also transfer digital copies of your code over to your client and the next team as a backup.

11. Login Credentials

Please provide all credentials (usernames and passwords) for any of the resources, websites, or platforms being utilised for this project. Please make sure that none of these credentials share passwords or usernames with any of your team's private credentials.

12. Other Relevant Information

This section is an invitation to add any additional information that you think will help to onboard new members. If you choose not to add any extra sections to this document, this section should be deleted.

Please edit this entire document as you see fit. If you think adding 5 extra sections that aren't listed here will help to communicate the nuances of your project to future members, go ahead! We want you to take full ownership of your handover and this document.

13. Appendices

Include all relevant artefacts delivered during the course of the project. Anything that will paint a clearer picture of your team's progress this trimester, the things that informed decisions, and the evolution of your product.

Please also include a link to your team's showcase video.

110. Project User Analysis

Project Leader: Miriam Llause Cotrina

Team member & Database Responsible: Tejas Varun Baskar

Team member & Visualization Responsible: Miriam Llause Cotrina

111. 32. Project Overview

The User Analysis project's main goal is to provide access keep users engage with our product (Smart Bikes) by giving them access to their current and historical performance information. User's performance analysis will be available in real time and in multiple platforms; everyone who trains with our smart bikes will be able to analyse their performance evolution in a dynamic and user-friendly way.

The project was discussed on the first company meeting of this trimester. After the leaders explain what the company was about, we thought we needed to offer some sort of post-sale-experience to our users where they not only visualise the outcomes of their workouts but also it is a way to keep them engage with product by creating a competitive in-game environment.

During the development of this project, we used platforms such: MS Excel, Python, Big Query, Google Cloud, and Power BI. The reason we selected Power BI as the main platform to create the visualizations and final deliverables of this project is due to its property to turn unrelated sources of data into visually and interactive reports.

The final deliverables of this project will be dashboards where users can see their performance evolution and interact with its own data. Users can track their progress and see if it is going according to their personal goals. However, we also had to do research about cycling metrics and visualization, these metrics and why we selected them will be also detailed in this report (User Manual).

112. 33. User Manual

The final deliverable of this Project contains four different dashboards in Power BI. These dashboards explore six different cycling metrics. It is important to result that the metrics were selected considering the research with have done and the data we had available.

Within each dashboard we have graphics that help the users observe their progress on a specific metric, in multiple and dynamic ways, each graphics was selected targeting the potential desire of our users. We aimed for these dashboards not to only look attractive but also to bring meaningful and valuable information that contributes to our user's fitness journey with our devices.

The Calories Dashboard:

This measure is noted to estimate the number of calories (unit of energy provided by food) that we are burning. Measuring this value is helpful while aiming for weight loss. The focus of weight loss is burning more calories and less intake of calories than burning. Most of the application and fitness devices use algorithms to calculate calories which is not exact most of the times. The better way to measure calories is by using power meter and heart rate sensors. The ration between power and calories is almost 1:1 with a 5% margin for error.

In this dashboard we can visualise four different graphics divided into two segments, the top two relates to Today's metrics compared to the daily average, and the other two at the bottom relates to the historical record (including today's).

The appropriate lecture of the below would be:

- “Today you have burnt 1,360 calories, the average calories you have burnt daily since you have joined is 942 Cal, your lowest in a day have been 18 Cal and your highest record is 3,740cal. – This is an indicator that today our User have burnt more calories than their daily average, but not yet reached their maximum capacity”.

The Distance Dashboard:

Distance is used to measure the endurance of an individual. But calculating distance varies depending on the type of track/road they are riding on. Because riding on a flat track and riding on an elevated track is different because there is more effort put in when the track is a bit elevated. Consider a nominally fit person the average distance covered in kilometres is 20.

This dashboard also presents four different graphics divided into 2 segments, the top two relates to Today's metrics compared to the daily average, and the other two at the bottom relates to the historical record (including today's).

The appropriate lecture of the below would be:

- “Today you have ridden 64 kilometres, the average kilometres you have ridden daily since you have joined is 36.55 km, your lowest in a day have been 0.61 km and your highest record is 166km. – This is an indicator that today, our User have ridden a distance longer than their daily average, but not yet reached their maximum capacity”.

The Heart Rate & Speed Dashboard:

Measuring the heart rate is one of the important measures because it helps us to know about our pressure levels to the heart. When the heart rate is high it is a sign that we are

pressurizing the heart too much in such cases we will have to reduce our work and give it a rest. Measuring high, low, and average is helpful as it helps us to analyse how we progress with our fitness. Maintaining lower and average heartrate is always better because it helps us avoid sudden heart problems.

Speed is majorly used for self-satisfaction. While see the speed and comparing it on a daily or monthly we can see the growth. Measuring the average speed is the best measure because while seeing a long-time review, we can consistently see the growth with average speed as current, min and max keeps varying a lot daily.

Both metrics were placed together due to their relation between each other, and this is probably one of the most complex dashboards that brings a lot of information to our users.

In this dashboard we can visualise the heart pulse and speed on a period of 1 year, and on the side graphics we can find most recently data such today's metrics and maximum reached since the user joined to the platform.

The Power & Cadence Dashboard:

Power is measured to determine how much effort you have put in to for the training session. Power is measured in watt. Measuring the average power is a better value considering a long-term analysis. As the average power increases, we can say that there a steady increase in performance and fitness. The average power for a beginner is around 75 – 100 watts.

Cadence is the number of revolutions per minute or RPM a person completes at a given speed while riding a bike. Generally, a good number to achieve in cycling cadence is between 80 – 100 rpm. Beginners' user will start pedalling as lower as 60 – 80 rpm and pro-users can do over 100rpm and 110 rpm during springs.

In this dashboard, we can visualise both the Power and Cadence. We have linear graphics with the last 12-month progression data, where the users can spot peaks and lowest points, and for a more specific data there is also tables with the numeric values per month.

113. 34. Completed Deliverables

Data Cleaning & Handling

Primarily responsible: Tejas Varun Baskar

Status: Fully Completed

Brief Description:

As far as the data cleaning is concerned the completed deliverables for this this trimester is processed data set of a single user of the wahoo devices. For this we converted the data that was given as each second's data and then it is now converted into a daily data that consists of metrics for a single day.

The other process done to the dataset is eliminating the NAN values and generalizing the given data set into 2 decimal points for better interpretation. Finally, after the selection of important metrics we find the minimum, maximum and the average for the appropriate metrics. Both, the code we use for data cleaning and the process ("How we did it") were stored on the below locations.

Locations:

Data Cleaning Procedure: https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2016%20Performance%20Ranking/Research%20on%20the%20important%20KPI%20and%20cleaning%20procedure..docx

Code for Data Cleaning: https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2016%20Performance%20Ranking/temp%20code%20for%20clearing%20the%20null%20values

Data Visualization & Dashboards

Primarily responsible: Miriam Llause Cotrina

Status: Fully Completed

Brief Description:

The main goal of this project was to create dashboards where users can visualise their fitness progress. At first, we needed to identify the appropriate cycling metrics, and then looked into our datasets to make sure we had stored the data we needed. After all of these was sorted, I uploaded the processed data into Power BI and created the dashboard.

The process of dashboard creation started before we jumped into Power BI, it started with Cycling metrics research as we needed first to understand what our Users would be interested to see. Both, the metrics research, and dashboards are part of the deliverables and they both can be found on the below locations.

Locations:

Cycling Metrics Research: https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2016%20Performance%20Ranking/Research%20on%20the%20important%20KPI.docx

Dashboards: https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2016%20Performance%20Ranking/RedBack%20Operations_Project%2016%20User%20Analysis%20FINAL%20PRODUCT.pbix

114. 35. Roadmap

By the end of this trimester, all deliverables planned have been successfully finished, however there are plenty features that we identify during the process of creating our dashboards that could make our product more attractive and enhance the user experience by making our platform more useful.

Please note that the features will be described below were not planned to be developed during this trimester, hence we do not have them on our current Trello board, as they are features to be analyse and develop over the next trimesters.

- User Data Handling

In terms of data cleaning the aim for next trimester deliverables are processing of data for all other user in the original dataset and finding a few more important metrics to analyse the performance in a much more efficient.

- Data Visualization

It would be interesting if we could get feedback from user's based on their interaction with our product and improve the visualizations on our dashboard, and/ or incorporate new metrics for the user analysis.

- User Interaction

One of the topics to work on next trimester is to incorporate interaction to the visualization. Users should be able to add their goals for the day, week, and month and track their current metrics with what they were expecting. For example, if a customer was expecting to burn 1,000 calories per day, we want them to input it into our system, and based on the data registered we should be able to tell them how far they are from achieving their goal. This feature will create an on-game competitive environment and will increase the engagement with our users.

- Predictive Analytics

We believe this is the last goal for our dashboards. Based on user's previous data, we should be able to apply predictive analytics and forecast their metrics. This feature will help users to make a training plan to achieve their goals. For example, if a user would like to increase their speed from 25 km/h to 35 km/h based on their current development how many hours should

this user train per day and what their training should look like to achieve this goal in 30 days?

115. 36. Open Issues

Data Access & Cleaning

- *The major issue that we encounter at the start of the trimester is not having access to the actual dataset to extract the data and do the work. – This was solved after a couple of weeks of having constant communication with one of our Company Leaders: Mark Telleys.*
- *Access to Google Cloud Platform – After the permissions were granted, one team member: Miriam Llauce had issues accessing to the google cloud platform. It was complicated to Log in into Google Cloud with the @deakin.edu.au account. Every time I tried to do it would take me back to my personal Google account. I had to Log off and delete my personal account saved details from the browser and then log in into Google Cloud with my Deakin account.*
- *Another issue was a confusion regarding the data we received, we had many cells filled as NaN and we did not know “how to eliminate the NaN values without affecting the dataset in large fashion” - at the end we overcome this using Python collab.*

116. 37. Lessons Learned

This trimester has been very insightful, and we both feel like we have accomplished a lot. Having our Company Team Leader: Mark and our Mentor: Ben support through this experience helped us to keep ourselves engage and motivated with the Company and Project.

Technology and Upskilling

In terms of technology, Power BI was the platform selected to develop our dashboards, and we both had very little knowledge. We both used it before, but we did not have the required experience, so we decided to upskill on it, we spend individual time upskilling and then we had weekly meetings where we both shared our learnings and put it into practise creating the dashboard.

We suggest to the next team members to dedicate time Upskilling if they think they need to, and to do so as soon as they start working on the project and not to leave it for the last minute, the better skilled they are the easier the work will be.

Teamwork

As a team, we believe that the major lesson learnt from this project in this trimester is the importance of teamwork. Since this project only consisted of two team members, there was a large amount of work that was split among us and most of these tasks were dependent on the work of another team. We both showed responsibility, interest,

and commitment with the project, we met every deadline, and we kept an open and constant communication which made this experience very pleasant.

We recommend to the next members to keep working as a team, to open different channels of communication and keep it ongoing.

117. Project Data Warehouse

Prastut Sapkota – Project Lead

Saransh Gupta – Developer

Ankit Mehta - Developer

118. 38. Project Overview

The data for the Data and AI teams of Redback Company are temporarily stored in BigQuery environment. As BigQuery is a cloud-based environment and therefore has a limited database security option. As there is no infrastructure to manage and does not need a database administrator it can further expose the data into more vulnerabilities. So, we aim to provide an effective and long-term solution for the temporary measure that we have opted. The data will be pipelined into a Relational Database Management System (RDMS), and we aim to provide it through MSSQL. The data warehouse will consist of various layers from extracting Raw data to creating data marts for the business. In this project the primary focus would be providing an effective data warehouse architecture following data integration strategies, governance and security but not limited to modelling and analytics strategies.

118.1. 38.1. Aims

1. Defining an architecture for the warehouse which includes data ingestion methods, storage options, and software and hardware specification among others.
2. Establishing data privacy, data security and data quality process.
3. Development of logical models, data schemas and data marts.
4. Establishing pipeline for the raw data connection and designing ETL pipelines.
5. The ability of handling large chunks of data with continuous monitoring and optimizing.
6. Development of master database.

118.2. 38.2. Deliverables

118.2.1. 38.2.1. Long-term

- A permanent solution to the extraction and storage of data.
- Integration of Extract, Transform and Load (ETL) workflows which extracts data from the source, transform as per the architecture and requirements of the data warehouse, and loads into the warehouse.

- An effective data quality and governance framework.

118.2.2. 38.2.2. Trimester

- A visual representation of the data structure in the data warehouse.
- A detailed description of how data from various sources are integrated and transformed.
- A master documentation that consists of comprehensive documentation of the data in the warehouse along with system and user documentation.
- A beta version of the integration of ETL through the proposed data warehouse architecture.

119. 39. User Manual

119.1. 39.1. System Requirements

As we are mainly using major tools like Alteryx and MS SQL, we need to know their system requirements which are as follows:

119.1.1. 39.1.1. Alteryx System Requirements:

- Operating System: Alteryx Designer is primarily available for Windows. As of my knowledge cut-off in September 2021, Alteryx Designer is not officially supported on Linux or macOS. However, you can use virtualization or containerization solutions like Parallels or Docker to run Alteryx on those platforms.
- Processor: 64-bit dual-core (x64) processor or higher
- RAM: 8 GB of RAM or higher (16 GB or more recommended for larger workflows and advanced analytics tools)
- Disk Space: 2 GB of available hard-disk space for installation
- Screen Resolution: 1280x800 or higher

119.1.2. 39.1.2. Microsoft SQL Server System Requirements:

The system requirements for Microsoft SQL Server can vary depending on the specific version and edition you are installing. Here are the general system requirements:

- Operating System: Microsoft SQL Server is available for Windows, but there are also editions available for Linux and macOS.
- Processor: 64-bit processor with a speed of 1.4 GHz or faster (2 GHz or faster recommended)
- RAM: At least 1 GB (4 GB or more recommended)
- Disk Space: Minimum of 6 GB of available hard-disk space for installation
- Screen Resolution: Minimum of 1024x768 pixels

It's important to note that the system requirements can vary based on the specific version, edition, and workload you plan to run. It is recommended to refer to the official

documentation provided by Alteryx and Microsoft for the most accurate and up-to-date system requirements for their respective products.

119.2. 39.2. Downloading Alteryx

To download Alteryx, follow these steps:

- *Visit the Alteryx website: Go to the Alteryx website at <https://www.alteryx.com>.*
- *Navigate to the Products section: Click on the "Products" tab in the top navigation menu.*
- *Choose Alteryx Designer: Select "Alteryx Designer" from the list of products. This is the primary tool for data preparation, blending, and advanced analytics. Follow the prompts to download Alteryx.*

119.3. 39.3. Setting up SQL Server

- *Download SQL Server: Visit the official Microsoft website or the SQL Server product page to download the version of SQL Server that suits your needs. Choose the appropriate edition and ensure it is compatible with your operating system.*
- *Run the Installer: Once the SQL Server installation file is downloaded, run the installer by double-clicking on it. This will start the SQL Server Installation Centre.*
- *Choose Installation Type: In the SQL Server Installation Centre, select "New SQL Server stand-alone installation or add features to an existing installation" to begin the installation process.*

119.4. 39.4. Running the SQL Script

The [link](#) to the GitHub repository where the MS SQL script is located which is to be executed in the MS SQL located in the local machine.

119.5. 39.5. Setting up Database Connections in Alteryx

To set up a database connection in Alteryx, follow these steps:

- *Launch Alteryx: Open Alteryx Designer on your computer.*
- *Open Workflow: Create a new workflow or open an existing one where you want to set up the database connection.*
- *Drag Input Tool: From the "Connectors" tab in the toolbar, locate and drag the "Input Data" tool onto the workflow canvas.*
- *Configure Input Tool: Double-click on the Input Tool to open its configuration window.*
- *Select Database: In the configuration window, select the "Database" option from the left panel.*
- *Choose Database Type: Choose the appropriate database type from the drop-down menu.*

119.6. 39.6. Troubleshooting

119.6.1. 39.6.1. Alteryx:

- 1. Verify Database Connection Details:** Double-check the server's name, authentication mode, username, password, and database name you provided in the database connection configuration. Ensure they are correct and match the settings of your SQL Server.
- 2. Test the Connection:** Use the "Test Connection" button in the Alteryx database connection configuration window to check if the connection can be established successfully. If the test fails, review the connection details and verify network connectivity to the SQL Server.
- 3. Firewall and Network Settings:** Ensure that the necessary firewall ports are open to allow communication between your computer running Alteryx and the SQL Server. Check your network settings and consult with your network administrator if needed.
- 4. Driver Compatibility:** Confirm that you have the appropriate database drivers installed for the version of SQL Server you are connecting to. Outdated or incompatible drivers can cause connection issues. You can usually download the required drivers from the database vendor's website.
- 5. Permissions and Credentials:** Verify that the user credentials provided for the database connection have the necessary permissions to access the SQL Server and the specified database. Check with your database administrator to ensure the user has the required privileges.

119.6.2. 39.6.2. Microsoft SQL Server:

- 1. SQL Server Service Status:** Ensure that the SQL Server service is running on the server. You can check the service status using the SQL Server Configuration Manager or Services console.
- 2. Check SQL Server Error Logs:** Examine the SQL Server error logs for any error messages or warnings that may indicate issues with the server. The error logs are typically located in the "Log" folder within the SQL Server installation directory.
- 3. Network Connectivity:** Verify that your computer can reach the SQL Server by pinging the server's IP address or hostname. If there is no response, check your network configuration, firewall settings, and network connectivity.
- 4. Authentication Mode:** If you are using SQL Server authentication, confirm that the provided username and password are correct. If using Windows authentication, ensure that the Windows account has the necessary permissions to access the SQL Server.

5. *Check Database Availability: Ensure that the target database is online and accessible. Use SQL Server Management Studio (SSMS) or a similar tool to connect to the server and verify the database status.*

If you encounter any specific error messages or issues, referring to the Alteryx and SQL Server documentation, online forums, or reaching out to their respective support channels can provide more detailed troubleshooting guidance.

120. 40. Completed Deliverables

We have successfully completed the development of an ETL (Extract, Transform, Load) model for the three datasets provided to us. Each dataset was assigned to a team member, and by the end of this trimester, we have delivered the completed ETL model.

To begin the process, we imported the data from Google Big Query to our individual machines. Next, we established databases using MS SQL on our respective machines to store and manage the data. We utilized Alteryx, a powerful data analytical tool, to perform data cleaning and transformation tasks. By leveraging Alteryx and MS SQL, we created Raw, Staging, and Master databases for each dataset, ensuring an organized and structured data flow.

All the relevant files and folders associated with our project can be found on GitHub. We have committed our final changes and deliverables to the repository, allowing for easy access and version control.

In summary, the completed deliverables for this trimester include Alteryx Workflow for each of the dataset:

1. *Bike Data:*
 - o *Responsible Team Member: Saransh Gupta*
 - o *Location: https://github.com/redbackoperations/data-analysis/tree/main/Trimester_1_2023/Project%2018%20Data%20Warehouse/Development/Alteryx/Bike_Data*
2. *Review Data:*
 - o *Responsible Team Member(s): Ankit Mehta*
 - o *Location: https://github.com/redbackoperations/data-analysis/tree/main/Trimester_1_2023/Project%2018%20Data%20Warehouse/Development/Alteryx/Review_data*
3. *Fitness Data:*
 - o *Responsible Team Member(s): Prastut Sapkota*
 - o *Location: https://github.com/redbackoperations/data-analysis/tree/main/Trimester_1_2023/Project%2018%20Data%20Warehouse/Development/Alteryx/Fitness%20Data*

121. 41. Roadmap

121.1. 41.1. Next Trimester:

- *Enhanced ETL Workflows: Continuation of the development and improvement of the Extract, Transform, and Load (ETL) workflows. This involves refining and optimizing*

the existing ETL processes, incorporating additional data sources, and enhancing data transformation logic.

- *Data Mart Development: Creation of specific data marts tailored to the needs of different business units or analytical requirements. Data marts provide subsets of data optimized for specific analysis or reporting purposes, enabling efficient and targeted data retrieval.*
- *Data Governance Implementation: Implementation of data governance practices and mechanisms to ensure ongoing data quality, compliance, and security. This includes establishing data stewardship roles, defining data standards, implementing data validation, and monitoring processes, and enforcing data privacy regulations.*
- *Advanced Analytics and Reporting: Development of advanced analytics capabilities, such as predictive modelling, machine learning, and data visualization tools. This will enable the Data and AI teams to derive valuable insights from the data warehouse and generate meaningful reports for decision-making purposes.*
- *Data Security Enhancements: Implementation of additional security measures to safeguard sensitive data within the data warehouse. This may include encryption, access controls, audit trails, and data anonymization techniques to ensure compliance with privacy regulations and protect against potential data breaches.*
- *Automated Data Pipeline Monitoring: Integration of monitoring and alerting systems to proactively identify and address issues within the data pipelines. This includes setting up automated alerts for data quality anomalies, performance bottlenecks, or pipeline failures, allowing timely intervention and minimizing potential disruptions.*

121.2. 41.2. In Progress (Incomplete Work Items):

- *ETL Workflow Optimization: Currently, the ETL workflows are in progress and being optimized for improved performance, scalability, and reliability. This involves identifying and resolving any bottlenecks, fine-tuning data transformations, and ensuring efficient data loading processes.*
- *Data Governance Framework Development: The development of the data governance framework is ongoing. This includes the definition of data standards, establishment of data stewardship roles and responsibilities, and the implementation of data validation processes. The framework is being designed to ensure data quality and compliance.*

These features and deliverables provide an overview of the planned work for future phases of the project, including those that are currently in progress. They demonstrate a progression towards a mature and efficient data warehouse solution that addresses Redback Company's data integration, governance, security, and analytics needs.

122. 42. Open Issues

122.1. 42.1. Potential issues and challenges the team is currently facing, along with progress made to address them:

- *Data Quality and Availability: The team is encountering challenges with the data provided, such as null values and bugs in the data. This hampers their ability to clean and analyze the data effectively using Alteryx. To address this, the team has initiated data quality checks and validation processes. They are working on identifying and addressing null values and data bugs to improve the overall data quality. It is crucial to continue investing in data cleansing and validation techniques to ensure reliable and accurate data for analysis.*
- *Alteryx Compatibility and Platform Limitations: The team has identified that Alteryx does not work on macOS, which affects the ability of team members using Mac computers to utilize Alteryx for data processing. Additionally, different versions of Alteryx have different tools and configurations, and Windows versions may not support the latest versions of Alteryx or MS SQL. While progress may have been limited in addressing these issues, it is important for future teams to explore alternative data preparation and analytics tools that are compatible with macOS. They should also ensure that all team members are using compatible versions of Alteryx and have the necessary system requirements to support the desired functionality.*
- *Unclear Task Review Process: The team is facing challenges with an unclear process for reviewing completed tasks on Trello, resulting in a backlog of work that is between unfinished and finished. To address this, the team should establish a clear and well-defined task review process. This process should include criteria for task completion, designate responsible reviewers, and outline the necessary steps for closing tasks in Trello. Clear communication and regular updates on task statuses are essential to avoid confusion and efficiently manage the workflow.*
- *Team Member Availability: Availability of team members is a challenge that can impact project progress. If team members have conflicting schedules or other commitments, it can delay task completion and coordination. To address this, it is important to establish regular communication channels, set clear expectations for availability and responsiveness, and distribute tasks and responsibilities based on team members' availability and expertise. Regularly monitoring and adjusting workloads can help mitigate the impact of limited availability.*

122.2. 42.2. Recommendations for Future Teams:

- *Data Quality Assurance: Establish data quality assurance processes from the early stages of the project. Implement data cleansing and validation techniques to identify and address data issues. Work closely with data providers to improve data quality and ensure accurate and reliable data for analysis.*
- *Compatibility and Platform Considerations: Assess the compatibility of tools and platforms with team members' systems and project requirements. Explore alternative software solutions that are compatible with different operating systems. Stay updated with the latest software versions and system requirements to ensure compatibility and access to necessary functionalities.*
- *Well-Defined Processes: Define and document clear processes for task management, including task review and closure. Ensure all team members are familiar with and follow these processes consistently. Regularly review and refine the processes based on feedback and evolving project needs.*

- *Resource Planning and Communication: Plan resources and workloads effectively, considering team members' availability and expertise. Establish regular communication channels and set clear expectations for responsiveness and availability. Proactively address conflicts and adjust workloads to ensure smooth project execution.*
- *System Security and Maintenance: Implement regular backups, system monitoring, and maintenance practices to prevent system corruption and ensure data integrity. Develop a disaster recovery plan and invest in security measures to protect project data and minimize potential disruptions.*

By addressing these challenges and following the recommendations, future teams can mitigate risks, enhance project efficiency, and create a more conducive working environment.

123. 43. Lessons Learned

123.1. 43.1. Key Lessons Learned:

- *Clear Communication Channels: One important lesson learned is the significance of establishing clear communication channels within the team and with stakeholders. It is crucial to have regular and transparent communication to ensure everyone is aligned, progress is effectively conveyed, and any challenges or roadblocks are addressed in a timely manner. Clear communication helps in managing expectations and maintaining project momentum.*
- *Continuous Testing and Quality Assurance: A valuable lesson is the importance of incorporating continuous testing and quality assurance processes throughout the project lifecycle. This ensures that data transformations, ETL workflows, and system functionalities are thoroughly tested and validated, minimizing the risk of errors or issues in the production environment. Early and frequent testing helps in identifying and addressing issues promptly.*
- *Agile Project Management: Adopting an agile project management approach enables flexibility and adaptability to changing requirements and evolving project needs. Agile methodologies, such as Scrum or Kanban, facilitate iterative development, frequent feedback cycles, and continuous improvement. This allows for better responsiveness to emerging challenges and ensures that the project stays on track.*
- *Documentation and Knowledge Sharing: Comprehensive documentation and knowledge sharing are essential for future reference, onboarding new team members, and facilitating seamless handovers. Maintaining updated documentation about the data warehouse architecture, data models, workflows, and system configurations helps in ensuring continuity and reduces dependency on specific individuals.*

123.2. 43.2. Recommendations for Future Teams:

- *Emphasize Collaboration: Encourage a collaborative work environment where team members actively collaborate and share their knowledge and expertise. Foster a culture of open communication, regular stand-up meetings, and collaborative*

decision-making. This helps in leveraging the diverse skill sets within the team and promotes shared ownership of the project's success.

- *Prioritize Data Governance from the Start: Establish a robust data governance framework early in the project to ensure data quality, security, and compliance. Define data standards, implement data validation processes, and establish clear roles and responsibilities for data stewardship. Proactively address data governance considerations rather than treating them as an afterthought.*
- *Test Automation: Invest in test automation tools and frameworks to streamline and expedite the testing process. Automated testing helps in reducing human errors, ensuring consistent testing practices, and accelerating the feedback loop. It allows for more comprehensive test coverage and supports regression testing as the project evolves.*
- *Continuous Integration and Deployment: Implement continuous integration and deployment (CI/CD) practices to automate the build, testing, and deployment processes. This allows for rapid and reliable deployments, reduces the risk of introducing errors during deployment, and ensures a more efficient release cycle.*
- *Knowledge Transfer and Succession Planning: Plan for knowledge transfer and succession planning to ensure the smooth transition of project ownership and knowledge sharing when team members transition out. Encourage documentation, conduct knowledge sharing sessions, and assign mentors to facilitate the transfer of knowledge and expertise.*

By incorporating these recommendations, future teams can overcome potential challenges and increase the effectiveness and efficiency of their project. Clear communication, agile practices, strong data governance, automation, and knowledge sharing will contribute to the project's success and ensure a more streamlined and sustainable data warehouse solution.

124. 44. Product Development Life Cycle

Over the course of the trimester, as a team, our work methods, habits, and processes have evolved organically and while we may not be able to clearly defined Product Development Life Cycle, we have set up practices that would guide our work.

The first set of practices that we followed is planning and setting up the goals. We started the week from the planning phase itself and identified tasks that we are going to complete on those weeks. Furthermore, we set up weekly objectives and defined what can be the key milestones and deliverables.

Following Planning and goal setting, we do follow the task allocation and Responsibilities through the Trello platform. Once our goals and deliverables has been set, the project lead would assign themselves and other colleagues based on their expertise and availability.

Apart from the weekly meetings, we have a constant communication with the team where we would discuss if any hurdles that may arise during performing the tasks and find out solutions to solve those hurdles. We do follow the agile approach of product development, as the project that we had needs us to adapt to the changing requirements and iterate on our work.

124.1. 44.1. New Tasks

The new tasks are created through teams collaborating process. Usually in our planning session we have a brainstorming and idea generation session where all the team members are encouraged to contribute their ideas and suggestion regarding the tasks. Once these ideas are finalized, we then evaluate them based on the impact and alignment to the project goals. Once we have the well-defined requirements, we break down the work into smaller, manageable tasks and then these tasks are assigned to the respective team members based on their weekly availability, workload, and skills. Some of the tasks may be reprioritized later and to tackle this requirement we have been involved in an iterative approach.

124.2. 44.2. Definition of Done

We try to define the definition of done before starting the task itself which includes the specific criteria and expectations required for the task completion. The task should meet the objectives and delivers the expected outcomes. We set up the task deliverables to exhibit high level of quality and functionality following the global standard code, features, and components. The task should be free from significant bugs and should be listed in the bug section in Trello board.

124.3. 44.3. Task Review

The tasks are reviewed by the team lead or the project manager who supervises the tasks and review process. The team themselves creates a unit testing environment to help the team lead to identify any bugs or error arising in the tasks and solves them before sending up for review.

The team lead performs a code review on the tasks that have been pushed into the fork repository. Once there is a green light from the team lead, the team members then prepare themselves for the next tasks. Since, we haven't had a proper designated Quality Assurance (QA) team or testing team we followed the process of unit testing environment. We are hoping for the Readback Operations to assign a designated team for QA and testing to have a quality product with bug free.

124.4. 44.4. Testing

As the developer themselves must perform the testing phases, we opted for unit testing approach. Unit testing approach helps in assuring that the code and data from the source file to the databases are accurate.

As the team is growing in Redback Operation, once we have a designated QA or tester, we intend to follow the following steps:

1. Creating a comprehensive test plan which outlines the testing approach, objectives, and scope.
2. Based on the planned requirements, various test cases are to be developed.
3. Checking any changes or additions made to the product do not impact existing functionalities.
4. A bug tracking system that stores all the bugs and errors identified.

124.5. 44.5. Branching Strategy

The current team has created a fork where they perform their development tasks. Once, the code has been developed the developer push their code into the designated folder inside the forked repository. Once it has been developed and approved by the project lead, the project

lead performs pull request where the team lead perform a code review and merge it into the main branch.

The commits and pull-requests are only performed once the development is done, and unit testing is performed. Then, the project lead performs a code review and if only a green signal is given from the project lead the pull-requests is performed.

A good approach towards avoiding messy branch is to create branching by each task. The new member should perform regular committing and pushing into the branch. Also, team collaboration and communication can play an important role as any concerns are addressed properly by the team members regarding the GitHub repository.

125. 45. Product Architecture

125.1. 45.1. Data Architecture

125.2. 45.2. Tech Stack

We have used two tools: Alteryx and SQL server. Here's is short description on both tools:

- *Alteryx: Alteryx is a powerful data preparation and analytics platform that provides a range of functionalities for data blending, cleansing, and advanced analytics. It offers a visual interface that enables users to design and execute data workflows without the need for coding. Alteryx was likely chosen for this project because of its ability to handle complex data transformations and automate repetitive tasks, allowing for efficient data preparation and analysis.*
- *SQL Server: SQL Server is a relational database management system (RDBMS) developed by Microsoft. It provides a robust platform for storing, managing, and retrieving structured data. SQL Server supports the SQL (Structured Query Language) standard and offers features such as data integration, data warehousing, and business intelligence. SQL Server was likely chosen for this project due to its scalability, security features, and extensive toolset for data management and analysis.*

126. 46. Source Code

Please find the below link to the source code for the project:

[Github Source Code Link](#)

127. 47. Login Credentials

The project was developed and tested in local environments. Please download the following two tools:

- *Alteryx*
- *MS SQL server*

For running the Alteryx workflow, create a database connection to your local MSSQL server to the input and output of the workflow to run the workflow.

128. 48. Raw Database

A raw database is referred to as the initial or source database that contains the original, unprocessed data collected from various sources. The main purpose of it is to provide

reliable and persistent storage solution for the data before it goes through the Staging or ETL process.

Alteryx:

129. 49. Master Database

A centralized repository that stores accurate and comprehensive data about various entities is termed as Master Database. The master database contains key elements of the data which are vital and consistent across different applications. The main aim of master database is to maintain data integrity, eliminate redundancy, and promote data consistency.

Alteryx:

130. 50. Staging Database

The purpose of staging data base is to provide data cleansing, transformation, and integration before loading the data into production stage and thus is referred as the intermediate step between the source and the target data warehouse which in our case is Raw and Production database respectively.

Alteryx:

131. 51. Production Database

The production database consists of live production environment data that contains aggregated and normalized data.

Alteryx:

132. 52. Appendices

Project Video Link: [Video Showcasing the project](#)

Project Research Link: [Github Data Warehousing Research](#)

Project Trello Link: [Trello](#)

133. Workout Categorisation

Redback Operations

Team Member: Nicholas Manning (Project Lead)

134. 53. Project Overview

Smart Bike users exercise for many different purposes: general fitness, fat loss, endurance etc. This project aims to develop a clustering model to categorise these workouts into groups of workout types to better tailor the user's experience and further gamify their training sessions. Each workout completed provides different outcomes in terms of distance covered, power generated, speed maintained, hills climbed etc. The model's goal is to identify the workout type of a particular workout so that we can decompose the benefits of the workout, offer workout suggestions or track their goals more accurately.

This project will use second by second breakdowns of user workout session data to develop an unsupervised machine learning clustering model to group workouts together with other similar workouts, then analyse those groups to determine what sort of similarities and features are identified. Once these labels are determined, they can be used for workout recommendations in the future.

The project is divided into four parts:

- 1. Research - Research and testing of clustering models and performance metrics.*
- 2. Data Acquisition and Cleaning – Accessing and importing Smart Bike data for the model, followed by cleaning, filtering, standardising and extracting relevant features.*
- 3. Modelling – Testing models and performance metrics to determine most successful outcome and applying those to identify appropriate clusters.*
- 4. Cluster Analysis – Decomposing and visualising characteristics of clusters to identify similarities observed by the model and workout types.*

Deliverables:

- 1. An aggregated dataset exemplifying standard user sessions.*
- 2. An unsupervised clustering model trained and tuned on the dataset.*
- 3. A set amount of workout types identified in the data.*
- 4. A prediction model for assigning future sessions to groups.*
- 5. Handover documentation for Trimester 2 – 2023.*

135. 54. User Manual

Access to files and scripts described can be found in the below:

GitHub:

https://github.com/redbackoperations/data-analysis/tree/main/Trimester_1_2023/Project%2017%20Workout%20Categorisation

Google Cloud Console - BigQuery:

<https://console.cloud.google.com/bigquery?project=sit-23t1-fit-data-pipe-ee8896e&ws=!1m4!1m3!8m2!1s891680982841!2sbb3a9d3a3feb4a9ea56a5c035945d280>

Utilising the workout categorisation clustering model requires a Fit file dataset, made of workout sessions, which can be accessed through Deakin's Google Cloud Console - BigQuery environment and the Smart Bike fitness data:

Dataset ID = sit-23t1-fit-data-pipe-ee8896e.fitness_data.

To retrieve the training dataset, I submitted the below SQL query:

Once the dataset is exported to your local computer, it can be run through the ‘Session-By-Second.ipynb’ python script. This script aggregates the second-by-second user data into individual workout sessions, cleans the data, standardises the distribution and removes any unnecessary datapoints, creating useful features for the final model, such as session length, total distance, total ascent, average heart rate, average cadence and average power.

The future user will then be able to recreate the results of the model and visualise the clusters. The whole dataset can be visualised using dimensionality reduction, turning the dataset 2D.

Along with other metrics, the clusters can be assessed to confirm there are distinct workout types in the data. See below, the Principal Component Analysis (PCA) scatterplot with 4 clusters highlighted.

A future user may run the model and choose a different amount clusters if they feel there is justification in the data, for example, if new sessions in the training dataset show a new workout type has emerged. Currently there appears to be clear delineations in the data to justify 4 clusters, as shown by the session Length vs session Distance scatterplot below:

KDE plots, which highlight the distributions of individual features, also can be used to show a clear difference in clusters, example below of the Speed KDE plot and Heart Rate KDE plot.

Once a number of clusters is chosen, the model and the scaler must be saved down in ‘.sav’ files for use in the prediction model. The current model files can be found in the project GitHub.

The prediction script, ‘Import and Apply Clustering Model.ipynb’, takes a single second-by-second fitfile dataset and applies the same transformations required for entry into the model,

It then runs the data through the model and assigns the new workout session to a cluster.

136. 55. Completed Deliverables

Completed deliverables include:

1. BigQuery SQL query for exporting data for model training.

https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2017%20Workout%20Categorisation/BigQuery%20FitFile%20Master%20Query%20for%20Workout%20Categorisation.docx

2. Master Fit file aggregation, cleaning, feature extraction, modelling, data visualisation and analysis tools.

https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2017%20Workout%20Categorisation/Session-BySecond%20Pipeline.ipynb

3. Cluster prediction/allocation model.

https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2017%20Workout%20Categorisation/Import%20and%20Apply%20Clustering%20Model.ipynb

4. Model and Scaler files

https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2017%20Workout%20Categorisation/clustering_model.sav

https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2017%20Workout%20Categorisation/clustering_scaler.sav

137. 56. Roadmap

The next step in the project requires the implementation of the model in a user interface:

- 1) Import recently completed or uploaded training session fitfile into the workout categorisation model to predict/allocate the session into a workout type.
- 2) Take the predicted workout type and determine what workouts to recommend.
- 3) Present the recommendation back to the user in a format they can utilise. For example, a pre-set training session focused on hills, sprints, average power etc.

138. 57. Lessons Learned

From a process and technology perspective, the major lesson learnt from this project is to ensure that the data utilised in the machine learning model is in a format that correctly represents the action being modelled and is therefore able to be used in a model effectively.

When the trimester began our team did not have access to the BigQuery environment and could not get the full Fit file dataset from the Smart Bikes. Initially this was not an issue, given Mark Telley, the cohort leader was able to provide a useful test dataset, however, this dataset had some aggregation and feature extraction methods already applied to it, which skewed the data. It also meant that when access to the site was granted, a whole new approach was required and a new focus back on simply getting the data into the right shape.

This is not to say that the early part of the trimester was wasted, it just was not used as efficiently as it could have been. Future teams building machine learning models should

prioritise the data because without the right data the model cannot learn and function properly.

139. 58. Product Development Life Cycle

139.1. 58.1. New Tasks

The idea for the project was part of a list of suggestions offered by the cohort lead at the start of the trimester, focusing on delivering new features to the Smart Bike project in the Data and AI space. As I was the only one who was interested in the workout categorisation project, I took on the task of outlining what could be done within the timeframe given and what broader expectations could be set for delivering insights to the user in the long term.

I bounced ideas and plans back and forth with cohort lead, Mark Telley, around what could be achieved and how the current data needed to be transformed to result in a deliverable product. I then set about working through the required steps to deliver the clustering model: researching, acquiring, and cleaning the data, modelling and analysing the results.

When problems were encountered, I referred to Mark for assistance and advice, but overall, I was able to work through problems and come up with new approaches to resolve issues on my own volition.

139.2. 58.2. Definition of Done

This project had a simple yet complex goal. Build or extract a dataset reflective of the workouts and build a model based on the dataset. The difficulty in these two steps was not necessarily in the coding, but more so in the confirmation that the data was an accurate reflection of the workouts, and the model represented that fact. The task was deemed done when the visualisations and analysis performed on the results justified that there were distinct clusters that could then provide useful recommendations to the user in the future.

139.3. 58.3. Task Review

Work was reviewed by the cohort lead Mark Telley when GitHub pull requests were submitted to merge updates with the main branch. Mark provided feedback on direction and offered support if needed. In future teams if there were more members working on this problem a different review process may be more beneficial.

139.4. 58.4. Testing

The model testing involved analysis and visualisation of the clusters to ensure that the model grouped datapoints into distinct and relevant clusters. The analysis and visualisation highlighted what features the model deemed most important. It not only validated the

model itself, but also the steps taken to manipulate, standardise and extract features from the data to create the dataset.

The final testing involved taking a single workout session Fit file from the source master data used to train the model, performing the requisite transformations, and inserting the session into the model. The final cluster allocation of the session matched the cluster allocated as part of the training set.

139.5. 58.5. Branching Strategy

The Data and AI team GitHub had branches for each project. Each user created a fork of the main branch, but only needed to amend their branch. When pull requests were submitted to merge the changes on a project only that branch was affected. Pull request were reviewed and approved by cohort leader Mark Tolley.

140. 59. Product Architecture

140.1. 59.1. UML Diagram

140.2. 59.2. Tech Stack

1. *Google Cloud Console – BigQuery: Existing infrastructure used to house and query Smart Bike Fit file master dataset.*
2. *Anaconda3 – Jupyter Notebook 6.3.0: Python 3.8.8 selected for programming the model due to comprehensive machine learning libraries.*
3. *GitHub – Version control and web hosting service for the project.*
5. *MS Excel – Export/Import datasets from BigQuery into models.*

141. 60. Source Code

Workout Categorisation Project GitHub:

https://github.com/redbackoperations/data-analysis/tree/main/Trimester_1_2023/Project%202017%20Workout%20Categorisation

Data and AI team Google Cloud Console - BigQuery environment:

<https://console.cloud.google.com/bigquery?project=sit-23t1-fit-data-pipe-ee8896e&ws=!1m4!1m3!8m2!1s891680982841!2sbb3a9d3a3feb4a9ea56a5c035945d280>

142. 61. Login Credentials

Access to Google Cloud Console – BigQuery and Data and AI team GitHub Repository must be granted through the company lead.

143. 62. Appendices

1. Data and AI Trello Board

<https://trello.com/b/NSuF3z83/data-analytics>

2. Workout Categorisation GitHub

<https://github.com/redbackoperations/data-analysis/tree/main/Trimester%201%202023/Project%201%20Workout%20Categorisation>

3. Project Showcase Video

144. Project 11 FIT File Handling and Data Pipeline

144.1. Project Team

Mark Telley

Company Redback Operations

145. 63. Project Overview

The FIT File Handling and Data Pipeline project aims to handle FIT files from the Wahoo KICKR Live, convert them to CSV format, and upload the data to a database. It will aim to provide real-time performance metrics through a rudimentary user interface using basic JS, HTML, and CSS as an MVP (only data points). The project will offer guidance to the web/application team on integrating the data within the game experience. A Python script will communicate with the KICKR, download, and convert FIT files to CSV, and integrate the data into the data warehouse project for storage. The project will provide a comprehensive solution for handling KICKR Live FIT files, making the data easily accessible for analysis and real-time performance metrics.

What is the project about?

What problem is the project solve?

- *Efficiently handles FIT files from the Wahoo KICKR Live.*
- *Converts FIT files to CSV format to support analysis and data manipulation.*

- *Unlocks the ability to upload the converted data to a database for storage and accessibility.*
- *Supports the ability to develop user interface to display real-time performance metrics (see all the other T2 projects)*
- *Offers guidance to the web/application team on integrating the FIT file data within the game experience.*
- *Streamlines the process of retrieving, converting, and analysing FIT file data.*
- *Eliminates the need for manual data extraction and conversion.*
- *Enables users to monitor their performance during workouts and make informed training decisions + allows the data science and AI team to conduct analysis etc.*
- *Provides a comprehensive solution for handling and integrating KICKR Live FIT files.*

What are the aims of the project?

Aims for Trimester

- *Develop a Python script for communicating with the Wahoo KICKR Live via Bluetooth connectivity and Wahoo API, downloading, and converting FIT files to CSV, and uploading data to a database.*
- *Create a rudimentary user interface using basic JS, HTML, and CSS to display real-time performance metrics as an MVP.*
- *Provide guidance to the web team on integrating the data within the game experience.*

What are the deliverables?

Long-term Deliverables:

- *A comprehensive solution for handling KICKR Live FIT files, ensuring easy accessibility for analysis and real-time performance metrics.*
- *Implementation of a scalable and secure data pipeline that seamlessly integrates with the game experience.*

Trimester Deliverables:

- *Development and completion of a Python script capable of downloading, converting, and uploading FIT file data utilising the Wahoo API and Bluetooth connectivity with the KICKR Live.*
- *Creation of a rudimentary user interface as a minimum viable product (MVP) to display real-time performance metrics.*
- *Documentation providing clear instructions on the usage and deployment of the Python script.*
- *Provision of guidance to the web team on effectively integrating the FIT file data within the game experience, including instructions on how to access the data through the implemented data pipeline.*

146. 64. User Manual

Three key elements make up the Project: Setting up a GCP and BigQuery environment, connecting to Bluetooth devices such as but not limited to Wahoo's Kickr bike trainer and connecting to Wahoo via API to retrieve information.

146.1. 64.1. 1.1. GCP & BigQuery

The GCP project provides a sandbox environment and essential datasets to support various projects. From handling FIT files and corporate reporting to sentiment analysis and user ranking, the project covers a wide range of data-related activities. It also facilitates access to app analytics data, demographics analysis, fitness data, user data, and Wahoo Kickr data. Also note original fitness data is stored securely in Google Cloud Storage buckets. It's an opportunity for the team to explore and innovate while paving the way for future developments in data management and analytics.

Refer to the readme.MD for comprehensive instructions.

https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2011%20FIT%20File%20Handling%20and%20Data%20Pipeline/GCP_Bigquery/GCP_BigQuery_Documentation.md

146.2. 64.2. Direction Connection to Wahoo Kickr

The BLE Cycling Power Data Collection sub project is a python script that captures cycling power data from a Bluetooth Low Energy (BLE) device and saves it in a CSV file. Utilising libraries like asyncio, bleak, and pandas, the script provides a straightforward process for data collection. After setting up the necessary libraries, connecting to the BLE device (Wahoo Kickr), and specifying the session length, running the script captures data such as timestamps, power output, energy accumulation, pedal power balance, torque, wheel and crank revolutions, force and torque magnitudes, as well as dead spot angles. The collected data is stored in a pandas dataframe and saved in a CSV file. Documentation also focuses on the use of the PyCycling package as a valuable resource for cycling-related data analysis.

Refer to the readme.MD for comprehensive instructions.

https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2011%20FIT%20File%20Handling%20and%20Data%20Pipeline/Kickr_Connection/README.md

146.3. 64.3. Direction Connection to Wahoo Kickr

The Wahoo API/FIT Handling is a comprehensive solution that utilises the Wahoo API to facilitate interactions with Wahoo products and services. It offers several functionalities to enhance the integration and handling of FIT files:

1. *Authentication: The API allows developers to authenticate with Wahoo by providing the necessary client ID and client secret. These credentials can be obtained from the Wahoo Developer Portal.*
2. *User Details: The API enables the retrieval of user details, providing access to relevant information associated with the authenticated user.*

3. *Select Workout: Users can select specific workouts using the API, making it convenient to fetch and work with targeted exercise data.*
4. *Workout Summary / FIT File: The API allows users to retrieve workout summary information and obtain the associated FIT file. FIT files are widely used in the fitness industry to store detailed workout data, and accessing this information opens possibilities for analysis and further processing.*
5. *FIT Conversion and Handling: The script includes functionality to convert and handle FIT files. By using the "fitparse" and "csv" libraries, FIT files can be transformed into CSV format, allowing for easier manipulation and analysis of the data.*
6. *Data Warehouse / Table: The script also provides an example of integrating the FIT file data into a data warehouse or table. Documentation demonstrates connecting to a data warehouse, creating a table, inserting data, and closing the connection. This functionality streamlines the process of storing and organising the FIT file data for future analysis and retrieval.*
7. *API Endpoints: The Wahoo API offers various endpoints that facilitate communication with the Wahoo system. These endpoints include authentication, user details retrieval, workout listing, workout summary retrieval, and more.*
8. *Security Considerations: The script emphasises the importance of client secret security, ensuring that sensitive information is handled with care and stored in secure locations. Documentation discusses token refresh mechanisms to ensure seamless authentication without compromising user data.*
9. *Database Handling: While providing an example of data insertion into a SQL Server database, the script acknowledges the need to address potential SQL injection issues and handle data types properly. Adhering to best practices, such as using parameterised queries and data type considerations, is crucial when interacting with databases to ensure data integrity and security.*

Full Documentation: https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2011%20FIT%20File%20Handling%20and%20OData%20Pipeline/Wahoo_Cloud_API/readme.md

146.4. 64.4. Pipeline Ideation

The below aims to provides an initial data pipeline that could service the Data/AI team:

- *The Blue = GCP Environment with a future to switch to the data warehouse (3.1).*
- *The Green = Directly connecting to the Bluetooth device (3.2).*
- *The Yellow / Purple / and Red = FIT File and Wahoo API handling (3.3).*

147. 65. Completed Deliverables

GCP

- Set up a Google Cloud Account and established a sandbox BigQuery database for project use.

Wahoo Device Connection

- Successfully recorded data from a Wahoo Kickr during a live session
- Thoroughly documented the steps involved and tested code.

Wahoo API and FIT File Handling

- Obtained a Wahoo API development account for the data pipeline project.
- Developed a working Wahoo API to retrieve information.
- Finalised the FIT handling script, ensuring efficiency and error-free operation.
- Thoroughly documented the steps and tested code.

Refer to the GITHUB LOCATION FOR ALL UPDATES:

https://github.com/redbackoperations/data-analysis/tree/main/Trimester_1_2023/Project%2011%20FIT%20File%20Handling%20and%20Data%20Pipeline

148. 66. Roadmap

Collaboration with IoT Team:

- Goal: Retrieve user-generated data from IoT devices.
- Milestone: Enable the generation of FIT files upon exercise completion.
- Action: Work closely with the IoT team to establish seamless data integration and mirror Wahoo's CLOUD API data model.

Integration with other Redback Teams and support data and analytical requirements:

- Goal: Utilise the retrieved data from different teams to drive analytics and visualisation efforts across the entire company.
- Milestone: Incorporate the IoT-generated data into the existing project framework Project 11)
- Action: Align with the to identify the specific data requirements and adapt the project scope accordingly.

Completion and Deployment of Data Pipeline:

- Goal: Establish a robust and efficient data pipeline using the Data Warehouse.
- Milestone: Successfully deploy the pipeline for seamless data processing and storage.

- Action: Collaborate with the other teams to design and implement the pipeline architecture, ensuring scalability, security, and efficiency.

Expansion of Data Sources:

- Goal: Capture additional data sources to enrich the project insights.
- Milestone: Incorporate data from app/website usage, corporate reporting, social interactions, and user ranking.
- Action: Engage with respective teams (e.g., UX, Cyber, Mobile) to define data collection mechanisms, integrate APIs or data connectors, and design data processing workflows.

Refinement of the Project Roadmap:

- Goal: Align the project with Trimester 2 objectives and priorities.
- Milestone: Evaluate the project roadmap and adapt it to future goals.
- Action: Engage in strategic discussions with stakeholders to identify Trimester 2 project requirements, dependencies, and resources.

By following this roadmap, we can as a team and company aim to enhance our data capabilities by incorporating real user-generated data, expanding the scope of our analytics and visualisation project, and establishing a robust data pipeline that supports other teams in the Company. This roadmap promotes cross-team collaboration and ensures that the project remains aligned with the evolving goals and objectives of Redback.

149. 67. Open Issues

Wahoo developer Account

- . Formal application from Deakin needs to be confirmed/completed.
- . Data model from IoT side of things needs to be established to mirror Wahoo's.

150. 68. Lessons Learned

Data Model Agreement:

Lesson: It is crucial to establish a clear and agreed-upon data model between IoT devices and external platforms like Wahoo or Garmin.

Recommendation: Future teams should prioritise defining and aligning the data model early on to avoid compatibility issues and ensure seamless data integration.

Streamlined Communication Channels:

Lesson: Effective communication channels are vital for project coordination and progress tracking.

Recommendation: Future teams should establish streamlined communication channels to facilitate regular updates, feedback sharing, and efficient collaboration. Utilise tools like project management software and regular meetings to keep the team informed and aligned.

Thorough Documentation:

Lesson: Comprehensive documentation is essential for knowledge transfer and project continuity.

Recommendation: Future teams should prioritise thorough documentation of project scope, technical details, and any challenges encountered. Documenting processes, code

repositories, and project resources will aid future team members and ensure smooth transitions.

Embrace Agile Methodologies:

Lesson: Agile methodologies enable adaptability and iterative development.

Recommendation: Future teams should consider adopting agile project management practices, such as scrum or Kanban, to promote flexibility, regular feedback, and continuous improvement. Agile approaches help teams navigate changing requirements effectively.

Testing and Quality Assurance:

Lesson: Early testing and quality assurance are crucial for identifying and addressing issues promptly.

Recommendation: Future teams should emphasise early testing and quality assurance throughout the development process. Implement automated testing frameworks, conduct regular code reviews, and involve stakeholders in the validation process to ensure high-quality deliverables.

151. 69. Product Development Life Cycle

Team Collaboration: We prioritise working as a team to deliver tangible value to the project and the company.

Scheduled Stand-ups: We conduct two weekly stand-up meetings (start, and end of the week) to synchronise our progress and address any issues or additional tasks required for the project. Ad hoc discussions also take place via the Teams app for quick decision-making.

Task Planning and Progress Tracking: We utilise Trello boards to plan and track our tasks, updating our progress accordingly. We frequently create pull requests (PRs) and ensure timely merging by the team lead.

151.1. 69.1. New Tasks

We come up with new tasks along the way while we are working on existing planned tasks or from each stand-up meeting time. Any new tasks will be created in the [Trello](#) board.

151.2. 69.2. Definition of Done

A DoD list is normally clearly defined in each Trello card, so the card assignee will be able to know exactly when a task is treated as completed by meeting all the DoD items. Additionally, we also have different status labels on each task on the Trello board to indicate their completeness.

151.3. 69.3. Task Review

All tasks' updates are reviewed by the Team Lead prior to being committed/merged in the Github Repo

151.4. 69.4. Testing

Testing was conducted manually – this involved creating test cases and working through them. All task work provided has been tested, and retested.

151.5. 69.5. Branching Strategy

We never directly push any changes into the company's `main` branch. To make any changes, we either create a new branch based off the latest `main` branch or fork the company's `main` branch into our own repo. After we've finished the changes, we create a PR against the company's `main` branch, and have it reviewed and merged by the team lead. We also ensure to resolve conflicts (if there's any) before merging back to the latest `main` branch.

152. 70. Product Architecture

152.1. 70.1. UML Diagram

Refer to point 3.4 Pipeline Ideation

152.2. 70.2. Tech Stack

	<ul style="list-style-type: none">. Google BigQuery. Google Storage Cloud (Buckets). Google Looker Studio (Visualisation). Google Colab (Python Coding)
	<ul style="list-style-type: none">. Wahoo Cloud API
	<ul style="list-style-type: none">. Version control. Project management and documentation
<i>Code Languages:</i>	<ul style="list-style-type: none">. SQL. Python. C (Curl request)

153. 71. Source Code

Refer to the following links

FIT Handling and API:

https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2011%20FIT%20File%20Handling%20and%20Data%20Pipeline/Wahoo_Cloud_API/Wahoo_FIT_Handling.ipynb

Bluetooth Connection:

https://github.com/redbackoperations/data-analysis/tree/main/Trimester_1_2023/Project%2011%20FIT%20File%20Handling%20and%20Data%20Pipeline/Kickr_Connection/Wahoo_Kickr_Connection

https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2011%20FIT%20File%20Handling%20and%20Data%20Pipeline/Kickr_Connection/Wahoo_Kickr_Connection/main.py

154. 72. Login Credentials

BigQuery / GCP refer to project documentation:

To gain access to the GCP Project / BigQuery, please contact your team lead. They will need to coordinate with:

- Scott Blackburn (Senior Technical Officer, Cloud Computing & AI, School of Information Technology)
- GCP Project: SIT-23t1-fit-data-pipe-ee8896e

Wahoo Developer:

Request Use of the Cloud API

The Cloud API uses the public Wahoo server and authorised user data. Because of this, Wahoo Fitness is currently limiting the use of the API to those who request it, as well as providing more information about the scopes involved and the purpose of the application.

When you apply request to Wahoo, it will show up on your Developer Portal as pending approval. Be sure to include as much information as you can about your application so Wahoo can be confident in approving your use of the Cloud API.

Refer to project documentation [here](#)

155. 73. Appendices

Refer to the Github Repo for all project documentation.

https://github.com/redbackoperations/data-analysis/tree/main/Trimester_1_2023/Project%2011%20FIT%20File%20Handling%20and%20Data%20Pipeline

Corporate Reporting

1. Project Plan: Project Pan is designed and is uploaded at Trello board at [Link](#)

2. The Data fields of fitness data are studied and analysed to know their functionality and scope for reporting and below is the analysis:

Time Stamp: The FIT Profile defines the date_time type as an uint32 that represents the number of seconds since midnight on December 31, 1989 UTC. (UTC is AEDT + 10).

Position_lat/Position_long: Garmin GPS devices are set by factory default to lat/long DM. This means it is set to latitude and longitude in degrees and minutes, with decimal minutes.

i.e., it is represented as 156° 44', 72° 10'.

Distance: The Garmin automotive devices can show distance in either miles/feet (statute units) or kilometres/meters (metric units)
Enhanced_altitude/Altitude: The device will measure changes in air pressure to determine your elevation. This information is recorded during your activity and is used to report elevation related information in Garmin Connect. Elevation calibrated by GPS is accurate to +/-400 feet with a strong GPS signal. If the values of altitude are too large to be fit in Altitude, then enhanced altitude is used.
Ascent: a climb or walk to the summit of a mountain or hill/an instance of rising or moving up through the air. Total Ascent provides a total of all increases to elevation (also known as elevation gain). Average Ascent provides an average of all ascents recorded during an activity. Maximum Elevation provides the highest elevation achieved.
Grade: Data field for Garmin devices that calculates the slope (or grade) of the hill you are walking on. It publishes the grade value (in %) to Garmin Connect so you can have a timeline inside your activity.
Calories: This is the total of active and resting calories that are calculated during a recorded activity on your device (from the moment that you start the timer for the activity to the moment you stop the timer). Speed/Distance Algorithm: This is the most basic method of determining calories. It is represented in calories/Kcal.
Speed/Enhanced Speed: It is distance by total time sent on an activity. It is calculated in m/sec or m/h. If the values of speed are too large to be fit in speed, then enhanced speed is used.
Heart_rate: heart rate values can be set as absolute or relative values. Absolute values represent beats per minute (bpm) for heart rate, or watts for power.
Temperature: The Temperature widget will display the ambient air temperature near the barometric altimeter port. This reading can be affected by body heat. It is represented in Fahrenheit.
Cadence: The cadence fields in a FIT file represent RPMs. For cycling 1 RPM equals one full rotation of the cranks. For running 1 RPM represents a step.
Power: Power values can be set as absolute or relative values. Absolute values represent watts for power.
Left_right_balance: It shows as a percentage the power separately put out by the left and right leg.
Gps_accuracy: It represents the drift with accurate gps values. GPS location accuracy is around 3 meters (10 feet), 95% of the time on Garmin devices. This means, at any given time, your device will save your location within 3 meters of your actual location.
Product_Name: It describes the product used for recording the activity.
Serial_Number: Most Garmin devices will have a unique serial number listed on the back or bottom of the device.
Age: Available on select Garmin watches, Fitness Age is an estimate of how fit you are compared to your actual age. Compatible Garmin watches will measure your Fitness Age differently, depending on which device you have. Fitness age is an estimate of how fit you are compared to your actual age.
Gender: It Shows gender of registered person of the device.
Weight: It gives the weight of the person in kgs.
FTP: Functional Threshold Power (FTP) is a measurement from power meters. It is the highest power level you can maintain for one hour without growing fatigued. FTP is

beneficial because it provides an outlook on performance ability.

Session_ID: It is unique id generated for each session performed by user.

User_ID: It is the unique ID generated for every user.

3. Queries for Views are created to fetch out different fields:

All the Queries were done on `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data_copy` table. A copy of `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data` is made and data is updated such that there is current year data to have visual output of all queries.

DISTANCE: Total Distance travelled by the user in last month, last week, previous day are fetched out by using the below queries.

MONTHLY_REPORT:

```
SELECT userID,sum(distance) AS TOTAL_DISTANCE
FROM `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data_copy`
where EXTRACT (MONTH from DATE_AEST) = EXTRACT(MONTH FROM CURRENT_DATE) - 1
and EXTRACT (YEAR from DATE_AEST) = EXTRACT(YEAR from CURRENT_DATE)
group by userID
order by userID;
```

WEEKLY_REPORT:

```
SELECT userID,sum(distance) AS TOTAL_DISTANCE
FROM `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data_copy`
where EXTRACT (DAY from DATE_AEST) > EXTRACT(DAY FROM CURRENT_DATE) - 7 and
EXTRACT (DAY from DATE_AEST) <= EXTRACT(DAY FROM CURRENT_DATE)
and EXTRACT (MONTH from DATE_AEST) = EXTRACT(MONTH FROM CURRENT_DATE)
and EXTRACT (YEAR from DATE_AEST) = EXTRACT(YEAR FROM CURRENT_DATE)
group by userID
order by userID;
```

DAILY_REPORT:

```
SELECT userID,sum(distance) AS TOTAL_DISTANCE
FROM `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data_copy`
where EXTRACT (DAY from DATE_AEST) = EXTRACT(DAY FROM CURRENT_DATE) - 1 and
EXTRACT (MONTH from DATE_AEST) = EXTRACT(MONTH FROM CURRENT_DATE)
and EXTRACT (YEAR from DATE_AEST) = EXTRACT(YEAR FROM CURRENT_DATE)
group by userID,date_AEST
order by userID;
```

CALORIES: Total Calories burned by the user in last month, last week, previous day are fetched out by using the below queries.

MONTHLY_REPORT:

```
SELECT userID,sum(Calories) AS TOTAL_Calories_Burned
FROM `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data_copy`
where EXTRACT (MONTH from DATE_AEST) = EXTRACT(MONTH FROM CURRENT_DATE) - 1
and EXTRACT (YEAR from DATE_AEST) = EXTRACT (YEAR from CURRENT_DATE)
group by userID
```

```
order by userID;
```

WEEKLY_REPORT:

```
SELECT userID,sum(Calories) AS TOTAL_Calories_Burned
FROM `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data_copy`
where EXTRACT (DAY from DATE_AEST) > EXTRACT(DAY FROM CURRENT_DATE) - 7 and
EXTRACT (DAY from DATE_AEST) <= EXTRACT(DAY FROM CURRENT_DATE)
and EXTRACT (MONTH from DATE_AEST) = EXTRACT(MONTH FROM CURRENT_DATE)
and EXTRACT (YEAR from DATE_AEST) = EXTRACT(YEAR FROM CURRENT_DATE)
group by userID
order by userID;
```

DAILY_REPORT:

```
SELECT userID,sum(Calories) AS TOTAL_Calories_Burned
FROM `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data_copy`
where EXTRACT (DAY from DATE_AEST) = EXTRACT(DAY FROM CURRENT_DATE) - 1 and
EXTRACT (MONTH from DATE_AEST) = EXTRACT(MONTH FROM CURRENT_DATE)
and EXTRACT (YEAR from DATE_AEST) = EXTRACT(YEAR FROM CURRENT_DATE)
group by userID,date_AEST
order by userID;
```

WEIGHT: Average weight maintained by the user in last month, last week, previous day are fetched out by using the below queries.

MONTHLY_REPORT:

```
SELECT userID,avg(weight) AS Weight
FROM `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data_copy`
where EXTRACT (MONTH from DATE_AEST) = EXTRACT(MONTH FROM CURRENT_DATE) - 1
and EXTRACT (YEAR from DATE_AEST) = EXTRACT(YEAR from CURRENT_DATE)
group by userID
order by userID;
```

WEEKLY_REPORT:

```
SELECT userID,avg(weight) AS Weight
FROM `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data_copy`
where EXTRACT (DAY from DATE_AEST) > EXTRACT(DAY FROM CURRENT_DATE) - 7 and
EXTRACT (DAY from DATE_AEST) <= EXTRACT(DAY FROM CURRENT_DATE)
and EXTRACT (MONTH from DATE_AEST) = EXTRACT(MONTH FROM CURRENT_DATE)
and EXTRACT (YEAR from DATE_AEST) = EXTRACT(YEAR FROM CURRENT_DATE)
group by userID
order by userID;
```

DAILY_REPORT:

```
SELECT userID,avg(weight) AS Weight
FROM `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data_copy`
where EXTRACT (DAY from DATE_AEST) = EXTRACT(DAY FROM CURRENT_DATE) - 1 and
EXTRACT (MONTH from DATE_AEST) = EXTRACT(MONTH FROM CURRENT_DATE)
and EXTRACT (YEAR from DATE_AEST) = EXTRACT(YEAR FROM CURRENT_DATE)
```

```
group by userID,date_AEST  
order by userID;
```

4. Views are created for the above drafted queries:

Steps to Create Views:

1. Copy the query on to new tab as below:

2. Click on Save view in the drop down next to save:

3. Now Give Dataset and View Name (as per choice) and click on save:

4. Now views are created, click on each view to see their details:

5. BMI is calculated for all the users in the table. To do this there are 2 columns added to the table Height and BMI.

Height Column is updated with few random values for different users using below query:

```
update `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data_copy`
```

```
SET height = 165
```

```
where USERID= 'U1000006'
```

Now BMI value is calculated as follows:

```
update `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data_copy`
```

```
SET BMI = weight*10000/(height*height)
```

```
where userid IN (SELECT distinct USERID from `sit-23t1-fit-data-pipe-  
ee8896e.fitness_data.master_data_copy`)
```

6. Queries are created to fetch out views for Monthly and Yearly BMI:

BMI_MONTHLY_REPORT: (BMI of all users for previous month is generated)

```
select userid,  
avg(weight)AS weight,avg(height)as height,avg(BMI)as BMI,  
case  
when avg(BMI) < 18.5 then "Under weight"  
when avg(BMI) >=18.5 and avg(BMI) < 25 then "Normal"  
when avg(BMI) >=25 then "Obese"  
end as Coach,  
EXTRACT(month from date_AEST) AS MONTH,  
EXTRACT(YEAR from date_AEST) AS YEAR  
from  
`sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data_copy`  
where EXTRACT (YEAR from DATE_AEST) = EXTRACT (YEAR from CURRENT_DATE) and  
EXTRACT (MONTH from DATE_AEST) = EXTRACT (MONTH from CURRENT_DATE)-1  
group by userid,month,year
```

BMI_YEARLY_REPORT: (BMI of users for all months in the year so far is generated)

```
select userid,  
avg(weight)AS weight,avg(height)as height,avg(BMI)as BMI,  
case  
when avg(BMI) < 18.5 then "Under weight"  
when avg(BMI) >=18.5 and avg(BMI) < 25 then "Normal"  
when avg(BMI) >=25 then "Obese"  
end as Coach,  
EXTRACT(month from date_AEST) AS MONTH  
from  
`sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data_copy`  
where EXTRACT (YEAR from DATE_AEST) = EXTRACT (YEAR from CURRENT_DATE) and  
EXTRACT (MONTH from DATE_AEST) <= EXTRACT (MONTH from CURRENT_DATE)  
group by userid,month
```

7. Views are created using the above queries:

8. Visual Representation of BMI is Developed:

BMI_YEARLY_REPORT:

1. Click on BMI_YEARLY_REPORT view in the big query explorer and View opens in the right pane as below:

2. Click on drop down next to export label in the right pane and there will be option explore with sheets as below:
3. Click on Explore with sheets and new sheets opens with preview data from view result. Name that sheet to **BMI_YEARLY_REPORT**:
4. Now click on Chart Label to create charts and new sheet opens next to it:
5. Above Report is created by selecting userid in x-axis and BMI in series and sorted by MONTH and filtered by userid columns.
6. Select one user id from filter to fetch BMI of user for all months in the year.
7. Explore the customize options to style the dashboard.
8. Sheets for different users are created as below by selecting each user in each sheet:

9. In the first sheet where data preview is present click on Schedule Refresh option on top to schedule the refresh of reports:
10. All the sheets in the report can be scheduled for refresh:
11. Reports can be published to audience by clicking on share on top right and giving access to Deakin group only in view of security:

12. Now this report can be accessed using URL:
https://docs.google.com/spreadsheets/d/1xOl-TcMc_0_mlxDijQKkvOOvC21zWWOXIEvgsvVZUo/edit?usp=sharing

BMI_MONTHLY_REPORT:

BMI_MONTHLY_REPORT is also visualised using the above steps. But here data of all users for previous month is shown.so only one chart is created with userid in x-axis and BMI in series and sort by user id:

This report can be accessed using:

<https://docs.google.com/spreadsheets/d/1Cq5NDIH2vKCdIj6nt2cYlkssPWfPipauB3Zjih1osLA/edit?usp=sharing>