

Deliverable Iteration #3 – Improved Functionality, Integration and Technical Debt

For

APP #1 (Group 1-11)

Prepared by Group 11

Alvira Konovalov (40074264)

Hamzah Muhammad (40156621)

William Chittavong (40048632)

Dmytro Chychkov (40034351)

Samuel Chuang (40133237)

April 12, 2022

Table of Contents

Table of Contents	2
Revision History	3
Functionality	4
Github	4
Major libraries and components	4
Modifications	5
Functional testing	5
Screen captures	7
Integration	16
Outcome	16
Challenges	16
Screen captures after integration	17
Technical debt	21
Overview	21
Bugs	21
Vulnerabilities	22
Security Hotspots	22
Code Smells	24
Coverage	26
Duplication	26
References	27

Revision History

Name	Date	Reason For Change	Version
Alvira Konovalov (40074264)	Apr 12	<ul style="list-style-type: none"> - Created deliverable #3 document and filed in all sections (Functionality, Integration and Technical debt). - Created integration diagram. - Helped with the integration process with group 9. - Converted CSV to JSON file and added more workout logs. - Improved functionality and design of use case 1 (Activity recap). 	3.1
Hamzah Muhammad (40156621)	Apr 12	Modifications to Use Case 2: <ul style="list-style-type: none"> - Added time frame filter functionality - Integration with data given from stats team (group 9) 	3.1
Samuel Chuang (40133237)	Apr 12	Modifications to Use case 3: <ul style="list-style-type: none"> - Added monthly/yearly total filter functionality - Read data off of Json file instead of MySQL database - Modified way to calculate weekly total - Integration with group 9 	3.1
Dmytro Chychkov (40034351)	Apr 12	<ul style="list-style-type: none"> - Assisted with Technical Debt analysis; - Modifications to Use Case 4: <ul style="list-style-type: none"> - Improved the viewing experience for charts depicting very long workouts. - The default graph now shows the most recent workout instead of a blank graph. - Updated to work with the data 	3.1

		<p>structure provided by the statistics team (group 9)</p> <ul style="list-style-type: none"> - Made necessary adaptations to facilitate the integration process with group 9. 	
William Chittavong (40048632)	Apr 12	<ul style="list-style-type: none"> - Integration with statistics team (group 9) on Moving Average Chart - Edited input field format. - Moving Average Chart now accepts JSON file - Kept Calculation Method due to misunderstanding with Stats Group 	3.1

1. Functionality

1.1. Github

Working post-integration Github repository is found at:

<https://github.com/ElviraKonovalov/WorkoutApp/tree/graphs-team-integration>

1.2. Major libraries and components

In deliverable #2, two software stack options were considered. After further research and testing, the following stack was chosen as it results in the best performance and integration with the statistics group is smoother.



- *Plotly*: an Open Source Python graphing library



- *Flask*: a Python web framework



- *Pandas*: an Open Source data analysis and manipulation tool



- *React*: an open source JavaScript library for building user interfaces

Google Charts

- *Google Charts*: an Open Source chart generator library
- *HTML5/SVG, Javascript*

1.3. Modifications

In deliverable #2, testing was done on mock data where the data was retrieved from CSV files or a MySQL database. In deliverable #3, data is now retrieved from JSON files. The formatting of the mock data was adjusted to the test data provided in the OneDrive by the professor.

Previously, the Web Interface of Fitness APP #1 was constructed using plain HTML, CSS, and JavaScript. Following the integration with Group 9, to stay consistent, all HTML pages were broken-down as React components instead. In terms of design, the look of the Web Interface has slightly changed.

Some minor design improvements and updates were made to each use case, but no other major features were added since the previous deliverable. The updates are described in detail in section 1.3. Functional testing.

1.4. Functional testing

The following section indicates which functionalities were added, fixed or improved and describes any updates made to each use case since the last deliverable.

Use case 1: Activity recap

The chart's code was adjusted to read data from JSON files. The time unit option was entirely dropped. The y-axis label is now updated when the distance unit is switched (the 'Miles/Km' button is clicked). In deliverable #2, the y-axis label was simply 'Distance'. Option to download the chart as a png is now available (seen in Figure 1).



Figure 1: Download chart as png option

Use case 2: Calories burned

For this use case, we've added the functionality that allows the user to select whether they want to view their total calories burnt for a specific week, month or year if those time frames have workouts entered into. Previously, they were only able to view their calories burnt for four given weeks without the option of filtering the timeframe. This graph was also adjusted to be able to prepare an array based on data from a JSON file instead of mock data from a MySQL database (previous version) and then insert that array into a MySQL Database from where we retrieve the specific statistics that the user wishes to view*. Also, instead of assuming there have been workouts for every single day, we assume there are blank entries in the JSON file.

*As of right now this is only partially working since the segment of my code that inserts the statistics from the JSON file into a MySQL Database isn't functioning. Therefore, that segment has been commented out in the source code and I decided to manually insert the given statistics from the statistics team into the MySQL Database for the sake of showing the functionality of graph generation for calories burned.

Use case 3: Distance travelled

For this use case, we've added the functionality that allows the user to select whether they want to view their total distance travelled of the month or of the year. Previously, they were only able to view their weekly total. This graph was also adjusted to be able to read an array generated based on stats of a JSON file instead of a MySQL database (previous version). As a result of this modification, the way we calculate the weekly total has been modified. Instead of assuming there are stats for every single day, we assume there are blank entries in the JSON file.

Use case 4: Elevation / Ride Intensity

Modified data access and parsing mechanisms to work with the data structure provided by the statistics team (group 9). Altered the graph display representation to allow for bigger workouts to be displayed to enhance viewing experience for the user. Added a functionality to display the most recent workout as the default graph when the Elevation menu tab is selected. Minor design layout changed. Code cleanup.

Use case 5: Moving average

The Moving Average Chart now reads from JSON files instead of a csv file. Changed the design of the input boxes of the chart. The Moving Average Chart does its own calculation of the Moving Average instead of only plotting the values. This is due to miscommunication with the statistics group where they were not calculating the Moving Average. The "rolling" function is kept for calculation.

1.5. Screen captures

Below are screen captures of the new features implemented in each use case.

Use case 1: Activity recap

Figure 2 and Figure 3 show the y-axis label updated when the distance unit is switched by the user (the 'Miles/Km' button is clicked).



Figure 2: Cycling done throughout the week in miles



Figure 3: Cycling done throughout the week in km

The figures below display the cycling distance the user has covered throughout the month, year and all cycling distance covered since the first use of the App.

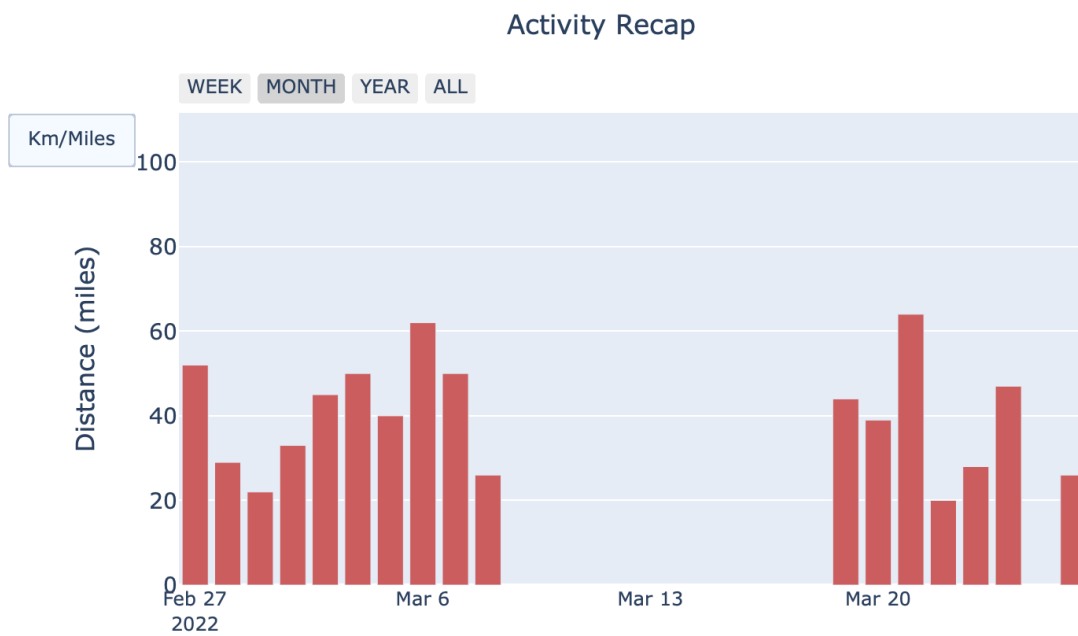


Figure 4: Cycling done throughout the month in miles

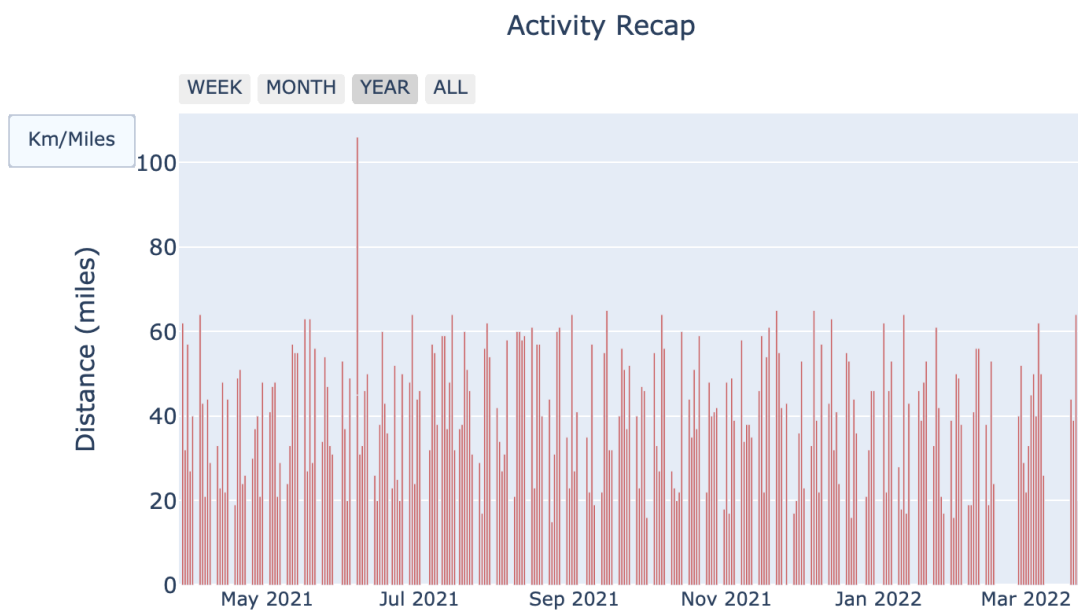


Figure 5: Cycling done throughout the year in miles

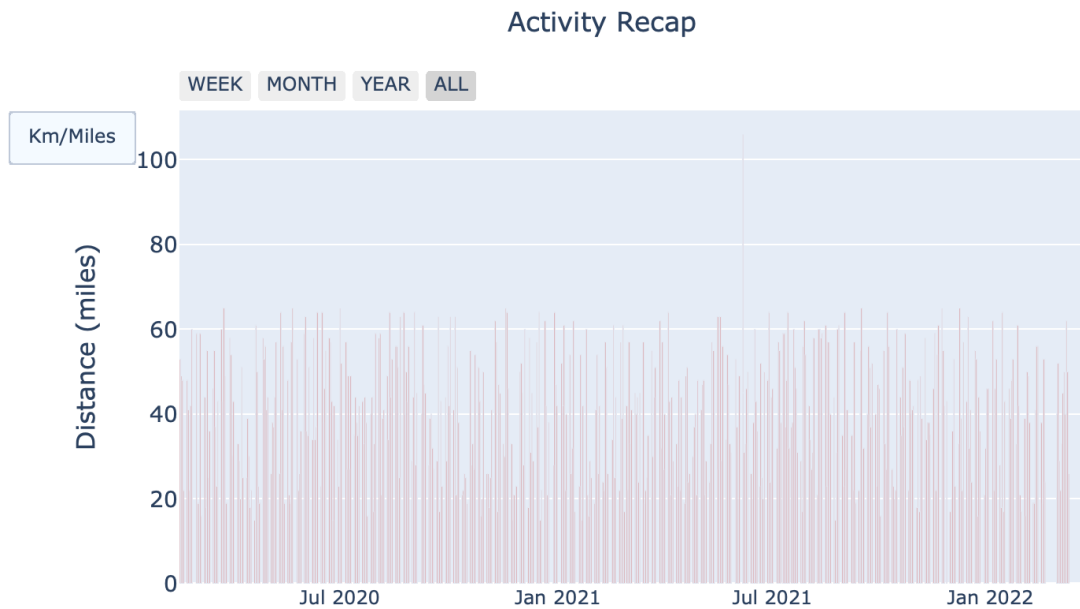


Figure 6: Cycling done since the first workout logged in the App

Use case 2: Calories burned

This is a demonstration of the three types of options available for calories burned. Each filter will showcase the calories burned on the active days within the selected time period.

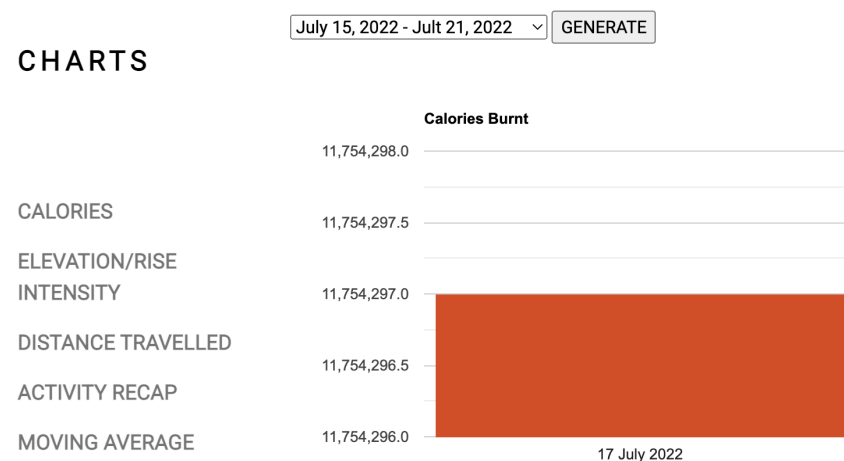


Figure 7: This is the option where the user wishes to see the statistics for a specific week. By default, if no filter is selected, the webpage will display the latest active week as shown above.

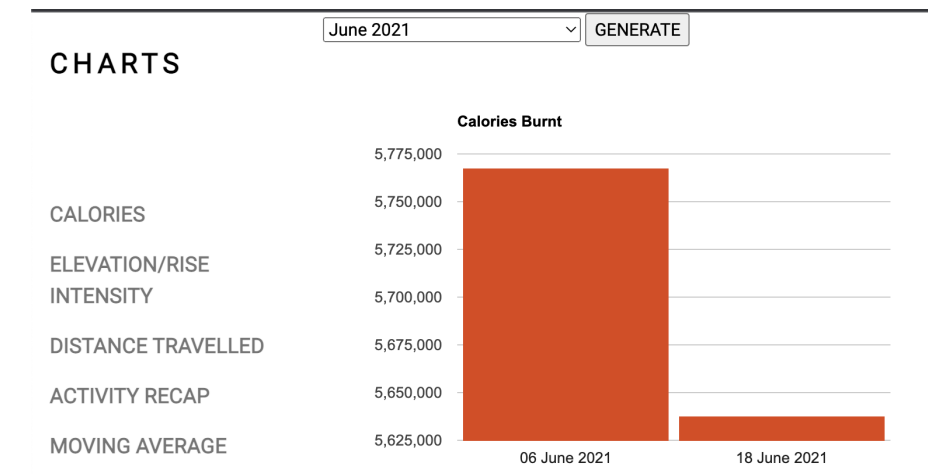


Figure 8: This is the option where the user wishes to see the statistics for a specific month, e.g. June.

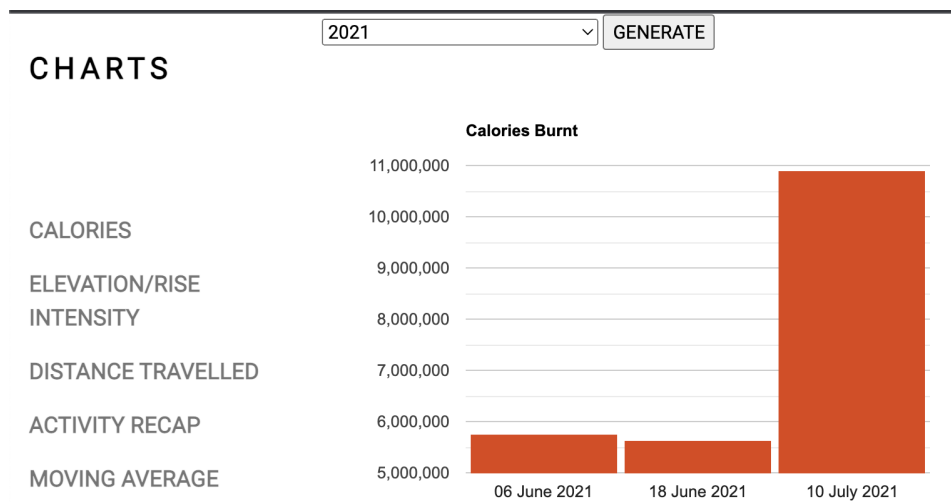


Figure 9: This is the option where the user wishes to see the statistics for a specific year, e.g. 2021.

Use case 3: Distance travelled

This is a demonstration of the 3 options available for distance travelled.

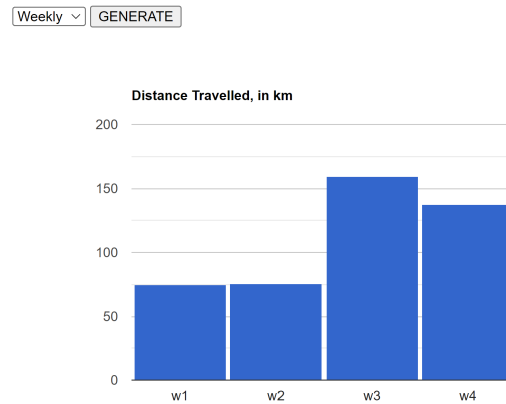


Figure 10: Default stats when user first load distance travelled

This is shown by default when the user first logs in. It is showing the total weekly distance travelled.

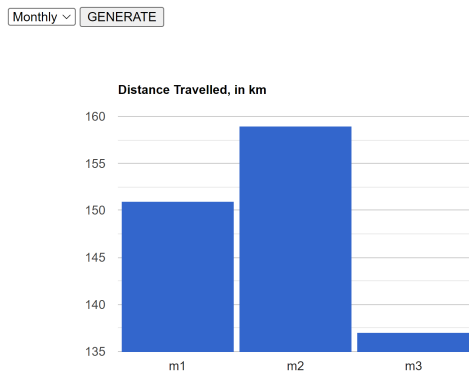


Figure 12. Monthly total

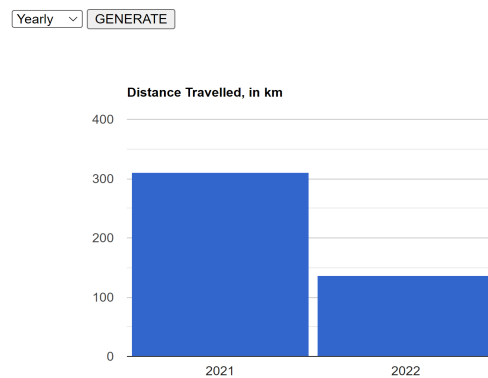


Figure 13. Yearly total

On the top left corner, users can then choose to view either “Monthly ” or “Yearly” total. Of course, the option to view weekly total is available as well.

Use case 4: Elevation / Ride Intensity

The figures below feature a marathon style workout of over 20 kilometers featuring a chart controls menu in the top right corner. The second picture showcases the region select feature for close-in examination of the graph.

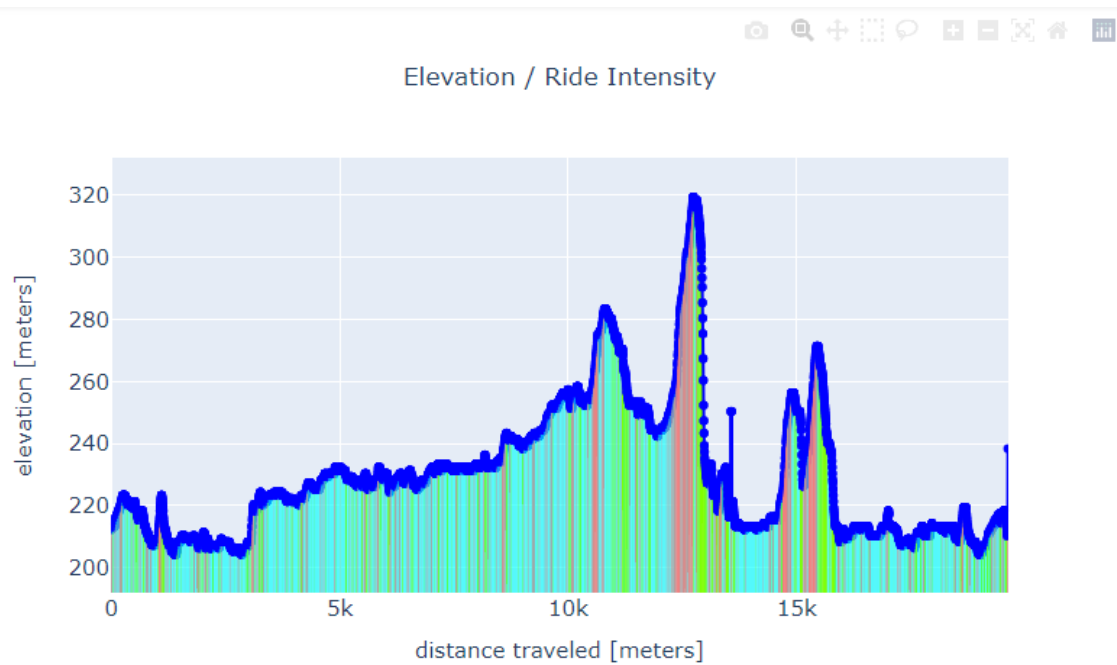


Figure 14: Elevation / Ride Intensity 20 km marathon

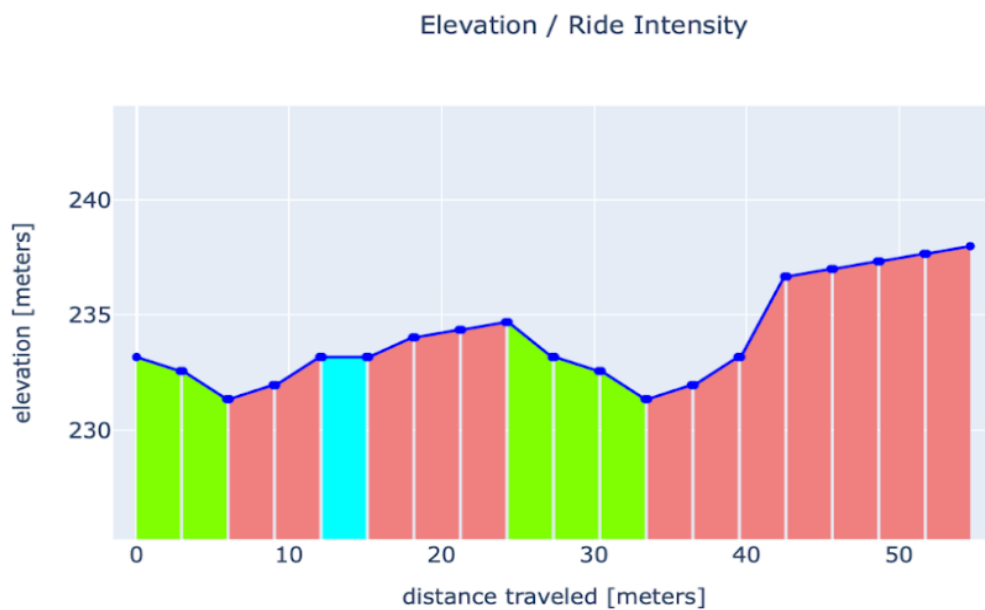


Figure 15: Elevation / Ride Intensity zoomed-in view

Use case 5: Moving average

The figures below show the different periods of time (days, months, a year) that the moving average is calculated for. Each one has a different window size. The user can see the trends over different periods of time under different windows (incremental average sizes).

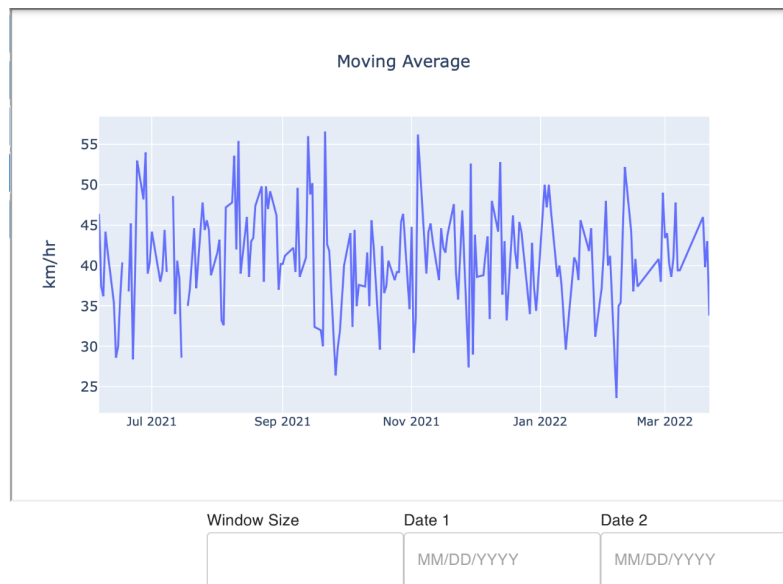


Figure 16: Moving Average chart with default window size of 5

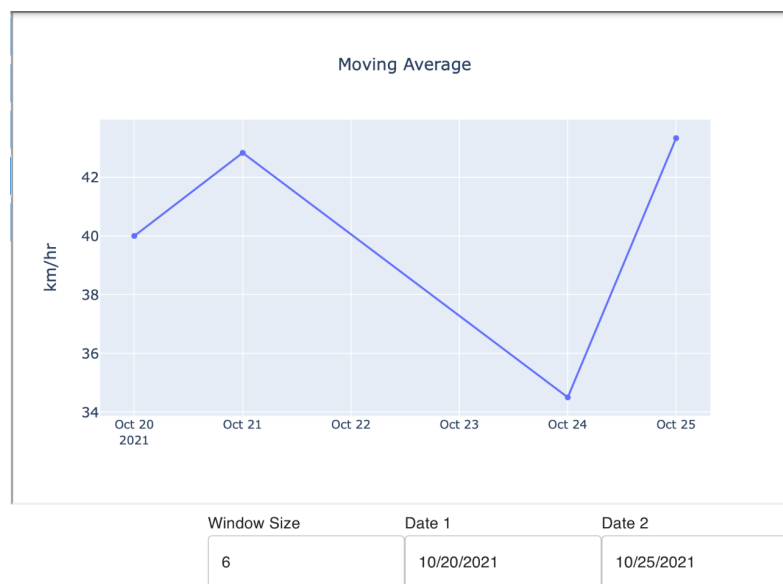


Figure 17: Moving Average chart with window size of 6 and specified range of dates

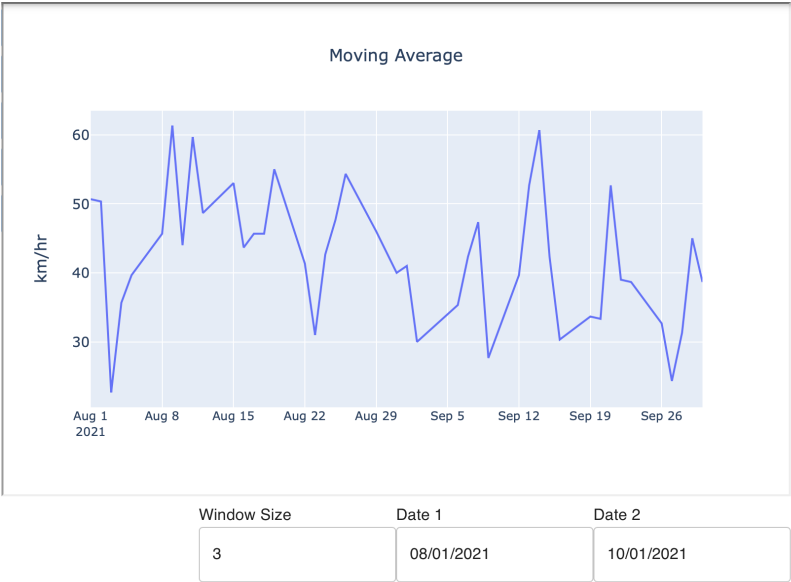
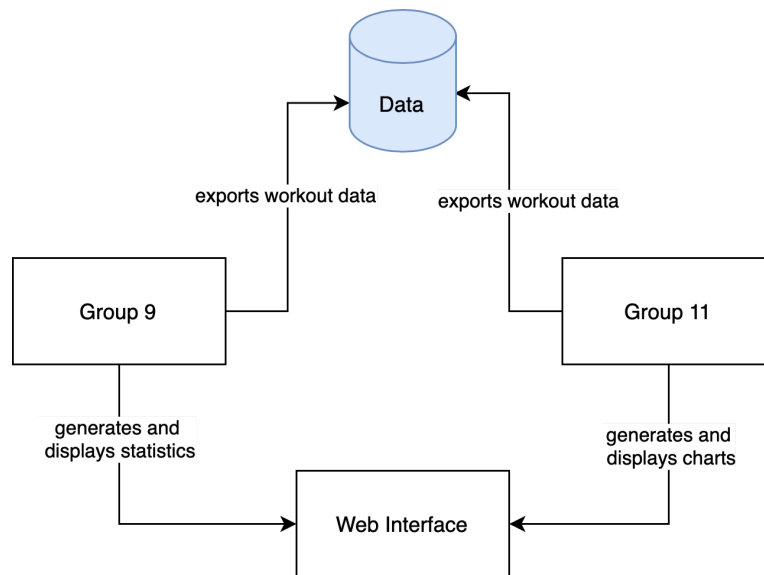


Figure 18: Moving Average chart with window size of 3 and specified range of dates

2. Integration

2.1. Outcome

Overall, the integration with Group 9 Statistics was *successful*. Our group managed to host our charts onto the statistics (Group 9) webpage. All charts were modified to work with the data structure provided by the statistics team (see JSON files).



The resulting program consists of 5000 lines of code, contains 507 statements and 76 distinct functions across 95 files.

2.2. Challenges

One of the challenges our group faced when integrating with Group 9 was a misunderstanding of what each group was responsible for. Our group had the impression that the statistics group was responsible for generating the data for our charts. However, the statistics group had a different task in mind. It became apparent that both groups (Group 9 and Group 11) have each created Web Interfaces to display the charts or the tables with the statistics. This issue was resolved by integrating our HTML pages as React components, as was done by Group 9.

Since group 9 did not provide our group with 'real' statistics to create our charts, we decided to keep using mock data. The formatting of the mock data was adjusted to the test data provided in the OneDrive by the professor.

Lastly, all data is now retrieved from JSON files. This adds some extra challenge for some use cases such as distance travelled, since originally the code was assuming that there are entries for every single day in the stats database (JSON file). It turns out that there wasn't so the code has to be adapted for the situation to be able to properly calculate the weekly, monthly, and yearly total.

2.3. Screen captures after integration

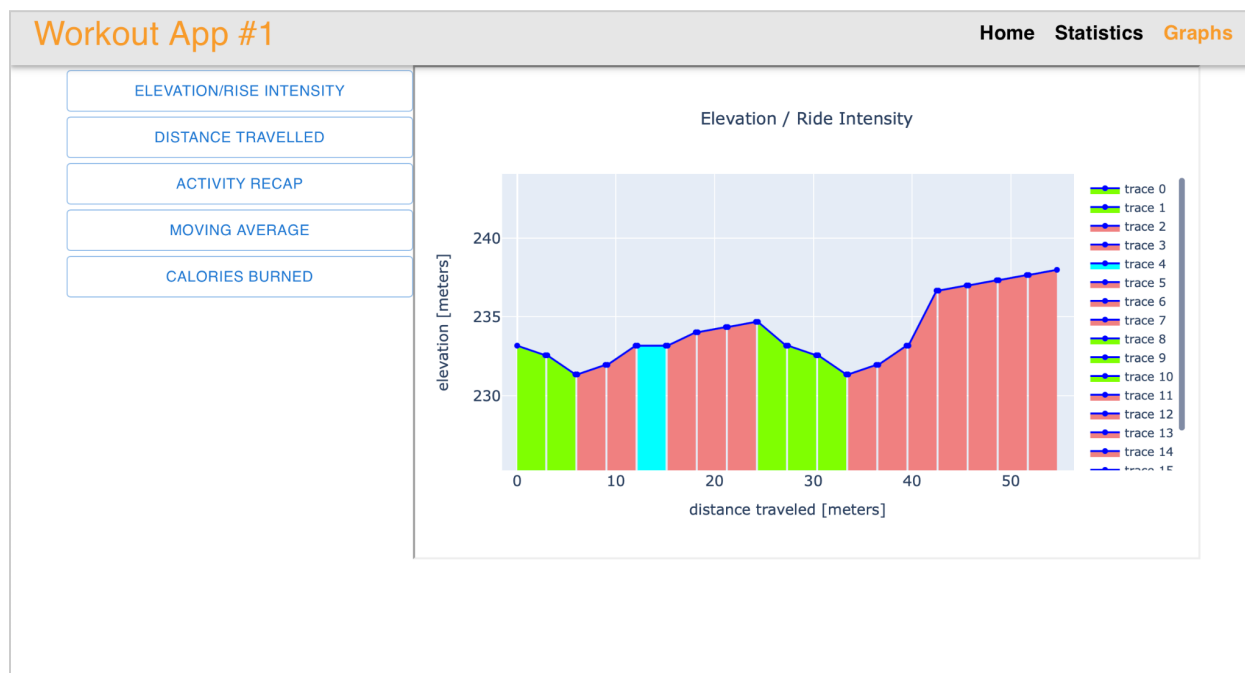


Figure 19: Integrated Web Interface design displaying the Elevation/Ride Intensity chart

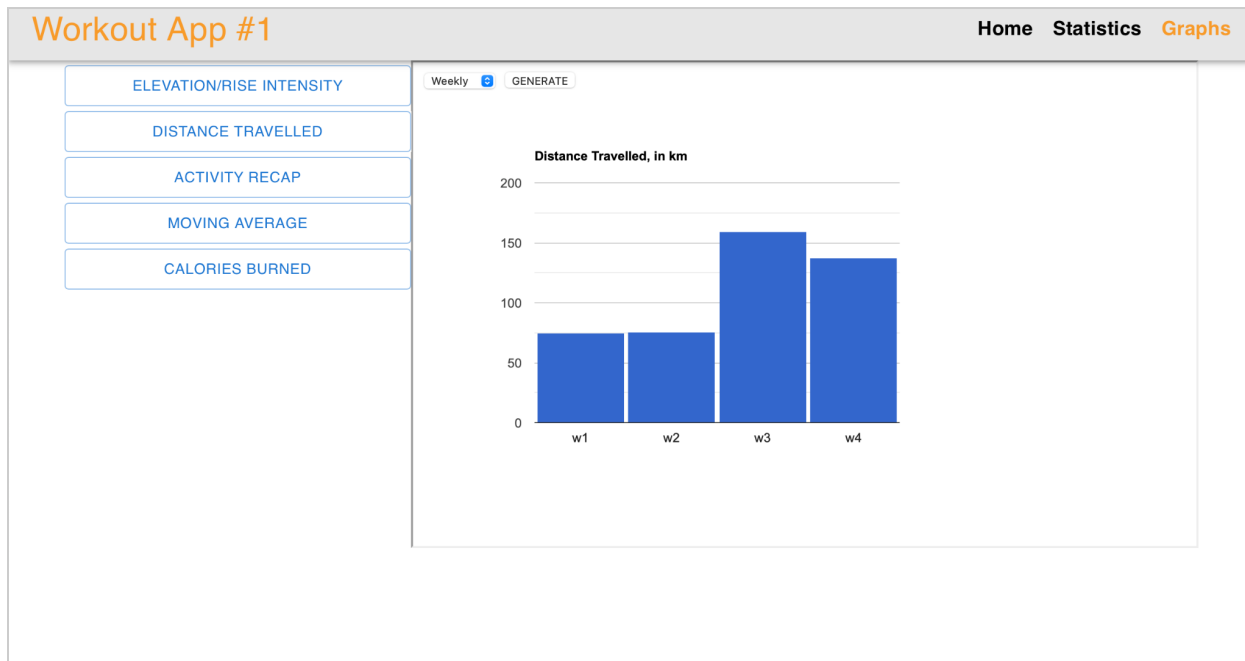


Figure 20: Integrated Web Interface design displaying the Distance Travelled chart

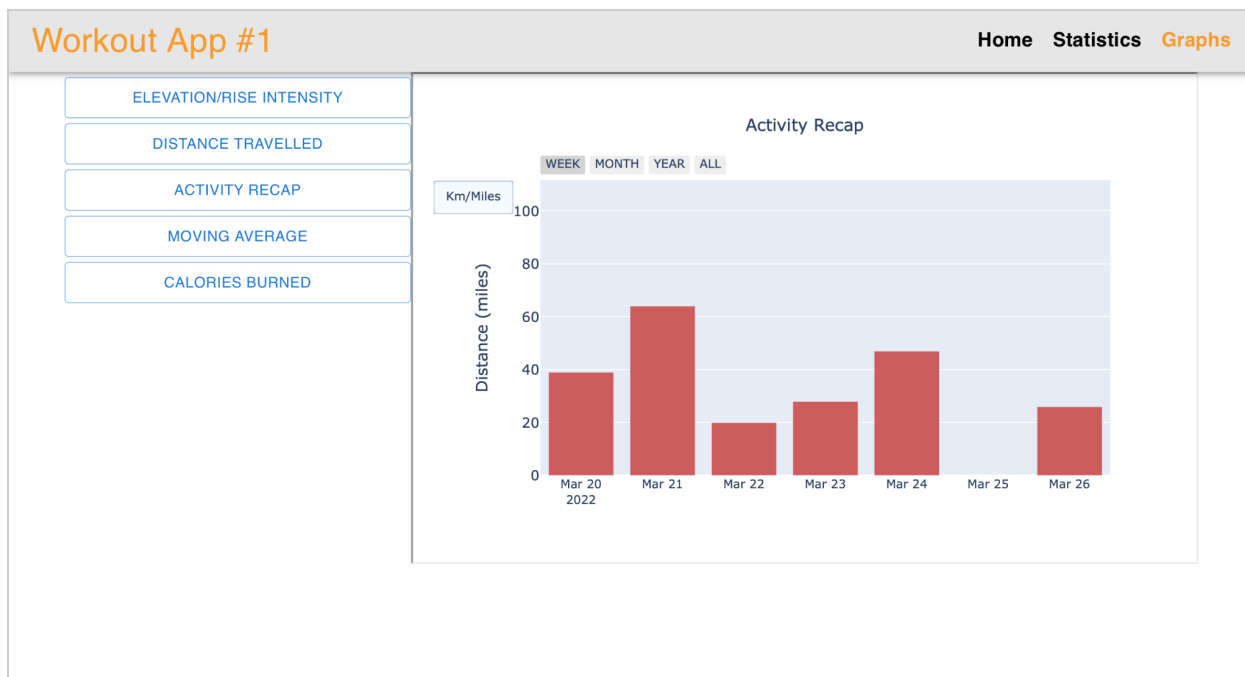


Figure 21: Integrated Web Interface design displaying the Activity Recap chart

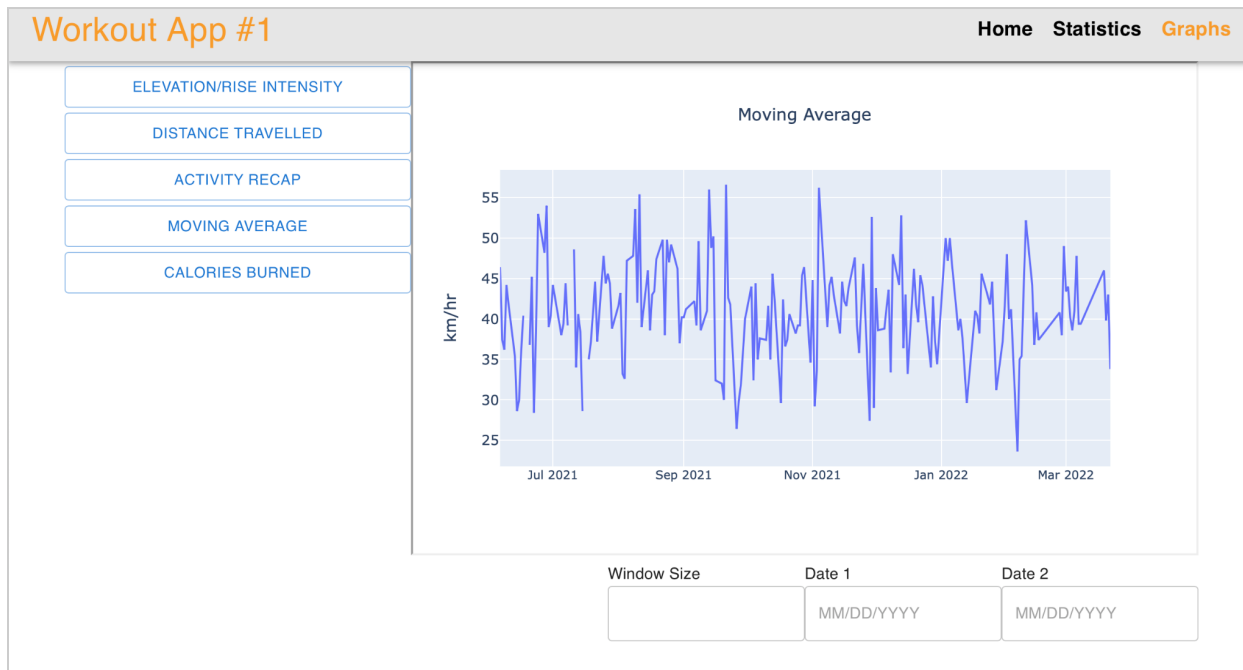


Figure 22: Integrated Web Interface design displaying the Moving Average chart

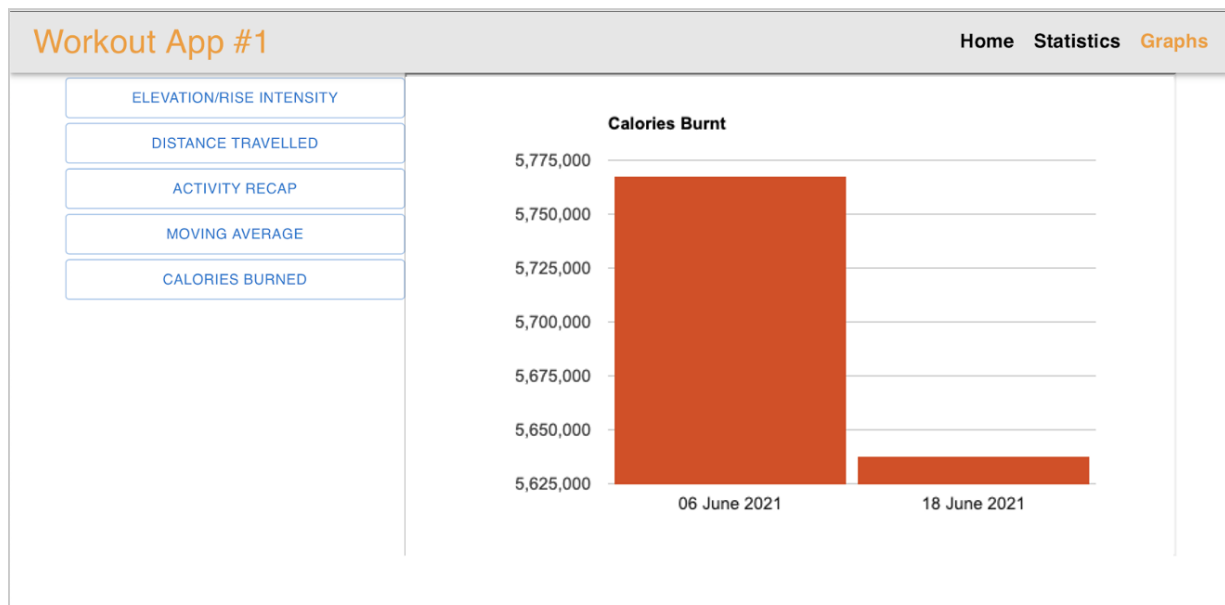


Figure 23: Integrated Web Interface design displaying the Calories Burnt chart

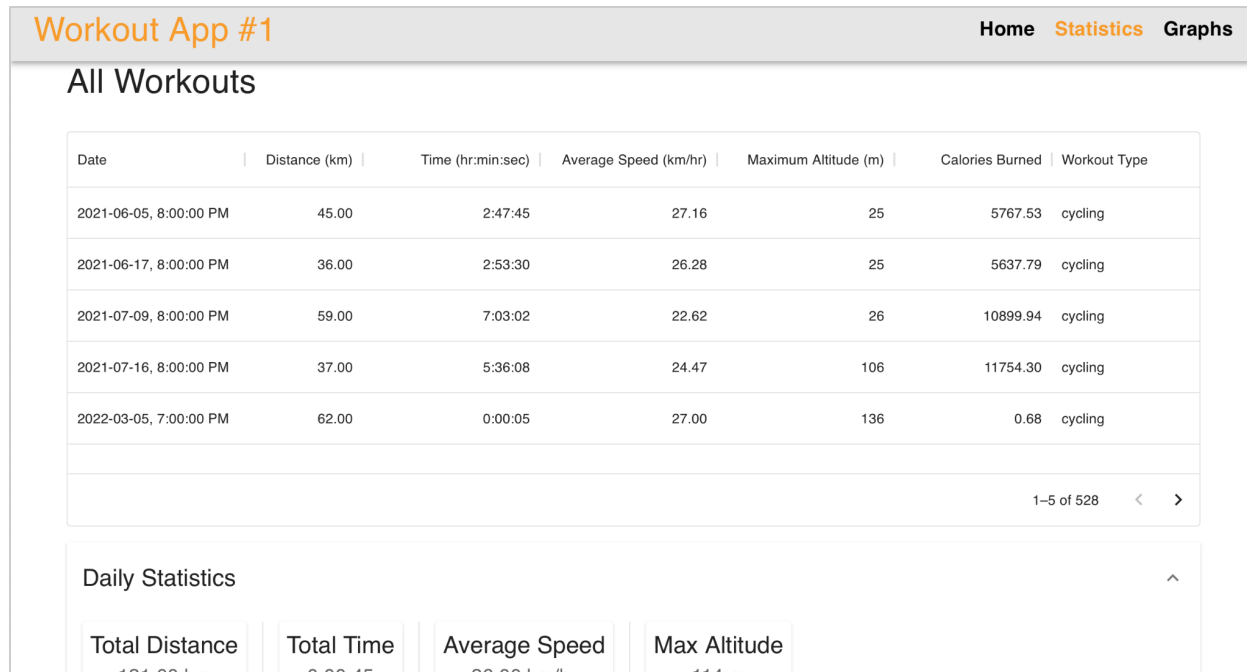
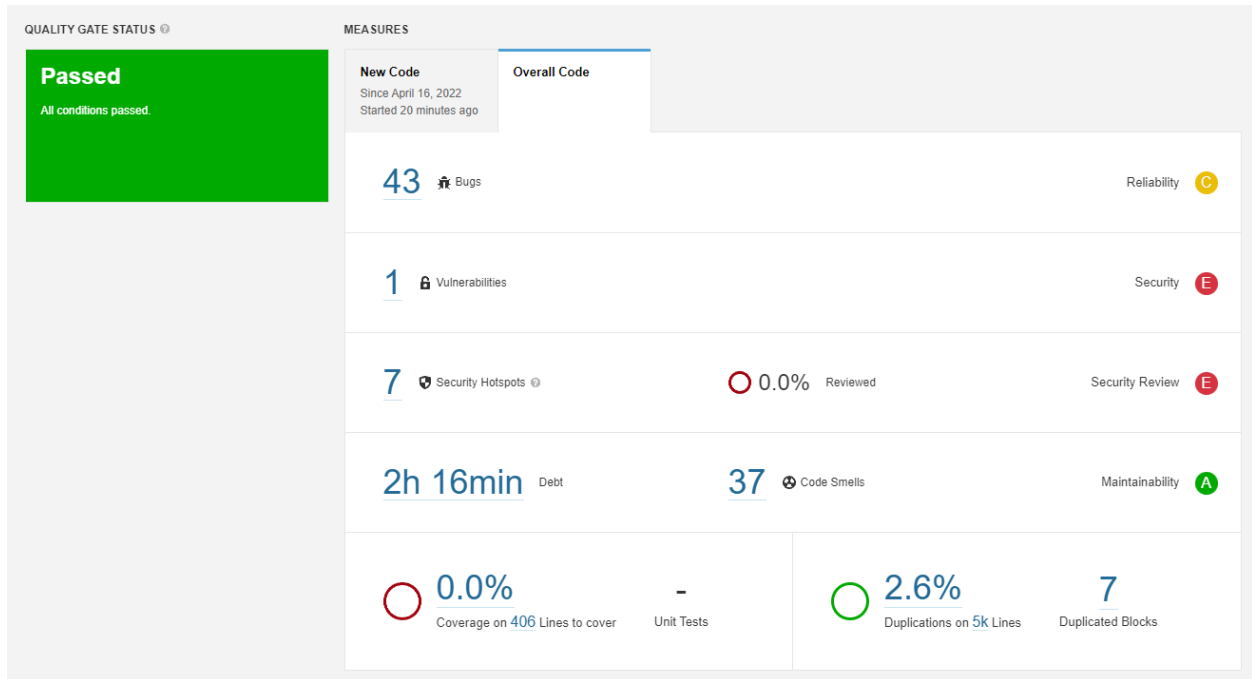


Figure 24: Integrated Web Interface design displaying Statistics of Group 9

3. Technical debt

3.1. Overview



3.2. Bugs

43 bugs were found in the SonarQube analysis. These are code quality bugs that do not harm the functionality of the application. The project's bug rating is 'C' (at least one Major Bug).

We recognised 7 different types of bugs, all of which are found in the temporary mock up display webpage:

chart-php/includes/header.php

<input type="checkbox"/>	Insert a <!DOCTYPE> declaration to before this <html> tag.	Why is this an issue? 21 minutes ago ▾ L1 🔗 🏠
🐞 Bug ▾	🔴 Major ▾	🔵 Open ▾ Not assigned ▾ 5min effort Comment
		🔗 user-experience ▾
<input type="checkbox"/>	Add "lang" and/or "xml:lang" attributes to this "<html>" element	Why is this an issue? 21 minutes ago ▾ L1 🔗 🏠
🐞 Bug ▾	🔴 Major ▾	🔵 Open ▾ Not assigned ▾ 2min effort Comment
		🔗 accessibility, wcag2-a ▾
<input type="checkbox"/>	Replace this <i> tag by .	Why is this an issue? 21 minutes ago ▾ L20 🔗 🏠
🐞 Bug ▾	🟢 Minor ▾	🔵 Open ▾ Not assigned ▾ 2min effort Comment
		🔗 accessibility ▾
<input type="checkbox"/>	Replace this tag by .	Why is this an issue? 21 minutes ago ▾ L21 🔗 🏠
🐞 Bug ▾	🟢 Minor ▾	🔵 Open ▾ Not assigned ▾ 2min effort Comment
		🔗 accessibility ▾
<input type="checkbox"/>	Replace this <i> tag by .	Why is this an issue? 21 minutes ago ▾ L35 🔗 🏠
🐞 Bug ▾	🟢 Minor ▾	🔵 Open ▾ Not assigned ▾ 2min effort Comment
		🔗 accessibility ▾
<input type="checkbox"/>	Add a <title> tag to this page.	Why is this an issue? 22 minutes ago ▾ L3 🔗 🏠
🐞 Bug ▾	🔴 Major ▾	🔵 Open ▾ Not assigned ▾ 5min effort Comment
		🔗 user-experience ▾

workout-app/node_modules/tailwindcss/components.css

<input type="checkbox"/>	Unexpected unknown at-rule "@tailwind"	Why is this an issue? 23 minutes ago ▾ L1 🔗 🏠
🐞 Bug ▾	🔴 Major ▾	🔵 Open ▾ Not assigned ▾ 1min effort Comment
		🔗 No tags ▾

3.3. Vulnerabilities

One vulnerability was found in the SonarQube analysis. The project's vulnerability rating is 'E' (Blocker).

chart-php/includes/db-connect.php

<input type="checkbox"/>	Add password protection to this database.	Why is this an issue? 24 minutes ago ▾ L7 🔗 🏠
🔒 Vulnerability ▾	🔴 Blocker ▾	🔵 Open ▾ Not assigned ▾ 45min effort Comment
		🔗 cwe, owasp-a2, owasp-a3 ▾

1 of 1 shown

This error however is due to the lack of password protection on a placeholder file used for communication with a mock up database, therefore it can be ignored.

3.4. Security Hotspots

7 security hotspots of different risk levels were found in the project by SonarQube. The project's security hotspot rating is 'E' (less than 30% vulnerabilities reviewed).

1) High risk

14 `app = Flask(__name__)`

Make sure disabling CSRF protection is safe here. [Comment](#)

Where is the risk?	What's the risk?	Assess the risk	How can you fix it?
<p>A cross-site request forgery (CSRF) attack occurs when a trusted user of a web application can be forced, by an attacker, to perform sensitive actions that he didn't intend, such as updating his profile or sending a message, more generally anything that can change the state of the application.</p> <p>The attacker can trick the user/victim to click on a link, corresponding to the privileged action, or to visit a malicious web site that embeds a hidden web request and as web browsers automatically include cookies, the actions can be authenticated and sensitive.</p>			

Where is the risk?	What's the risk?	Assess the risk	How can you fix it?
<p>Recommended Secure Coding Practices</p> <ul style="list-style-type: none"> Protection against CSRF attacks is strongly recommended: <ul style="list-style-type: none"> to be activated by default for all unsafe HTTP methods. implemented, for example, with an unguessable CSRF token Of course all sensitive operations should not be performed with safe HTTP methods like <code>GET</code> which are designed to be used only for information retrieval. 			

The easiest solution would be to use a local copy of a library rather than invoking a network call.

2) Low risk

247 `def _corsify_actual_response(response):`
248 `response.headers.add("Access-Control-Allow-Origin", "*")`

Make sure this permissive CORS policy is safe here. [Comment](#)

Where is the risk?	What's the risk?	Assess the risk	How can you fix it?
<p>Having a permissive Cross-Origin Resource Sharing policy is security-sensitive. It has led in the past to the following vulnerabilities:</p> <ul style="list-style-type: none"> CVE-2018-0269 CVE-2017-14460 <p>Same origin policy in browsers prevents, by default and for security-reasons, a javascript frontend to perform a cross-origin HTTP request to a resource that has a different origin (domain, protocol, or port) from its own. The requested target can append additional HTTP headers in response, called CORS, that act like directives for the browser and change the access control policy / relax the same origin policy.</p>			

Where is the risk?	What's the risk?	Assess the risk	How can you fix it?
<p>Recommended Secure Coding Practices</p> <ul style="list-style-type: none"> The <code>Access-Control-Allow-Origin</code> header should be set only for a trusted origin and for specific resources. Allow only selected, trusted domains in the <code>Access-Control-Allow-Origin</code> header. Prefer whitelisting domains over blacklisting or allowing any domain (do not use * wildcard nor blindly return the <code>Origin</code> header content without any checks). 			

A list of trusted domains should be specified to address this security check.

3) Other (same error 5 times in html files)

193	<code><script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script></code>		
<p>Make sure not using resource integrity feature is safe here.</p>			Comment

Where is the risk?	What's the risk?	Assess the risk	How can you fix it?
<p>Fetching external resources, for example from a CDN, without verifying their integrity could impact the security of an application if the CDN gets compromised and resources are replaced by malicious ones. Resources integrity feature will block resources inclusion into an application if the pre-computed digest of the expected resource doesn't match with the digest of the retrieved resource.</p>			

Where is the risk?	What's the risk?	Assess the risk	How can you fix it?
<p>Recommended Secure Coding Practices</p> <ul style="list-style-type: none"> implement resources integrity checks for all static resources (where "static" means that the resource's content doesn't change dynamically based on the browser) use versioned resources instead of using "latest" version of the resources 			

This refers to “server side” local js nodes without a signature that can be trusted.

3.5. Code Smells

37 code smells were found in the project in the SonarQube analysis.

Critical:

 charts-api/servers/server.py		
<input type="checkbox"/>	Define a constant instead of duplicating this literal 'workouts.json' 4 times.	Why is this an issue? 33 minutes ago ▾ L25 🔗 ⚙️ ▾ 🔗 Code Smell ▾ 🔴 Critical ▾ 🔵 Open ▾ Not assigned ▾ 8min effort Comment 🔗 design ▾
 workout-app/node_modules/flatted/php/flatted.php		
<input type="checkbox"/>	Add curly braces around the nested statement(s).	Why is this an issue? 33 minutes ago ▾ L41 🔗 ⚙️ ▾ 🔗 Code Smell ▾ 🔴 Critical ▾ 🔵 Open ▾ Not assigned ▾ 2min effort Comment 🔗 pitfall ▾
<input type="checkbox"/>	Add curly braces around the nested statement(s).	Why is this an issue? 33 minutes ago ▾ L43 🔗 ⚙️ ▾ 🔗 Code Smell ▾ 🔴 Critical ▾ 🔵 Open ▾ Not assigned ▾ 2min effort Comment 🔗 pitfall ▾
<input type="checkbox"/>	Add curly braces around the nested statement(s).	Why is this an issue? 33 minutes ago ▾ L79 🔗 ⚙️ ▾ 🔗 Code Smell ▾ 🔴 Critical ▾ 🔵 Open ▾ Not assigned ▾ 2min effort Comment 🔗 pitfall ▾
<input type="checkbox"/>	Add curly braces around the nested statement(s).	Why is this an issue? 33 minutes ago ▾ L87 🔗 ⚙️ ▾ 🔗 Code Smell ▾ 🔴 Critical ▾ 🔵 Open ▾ Not assigned ▾ 2min effort Comment 🔗 pitfall ▾
<input type="checkbox"/>	Add curly braces around the nested statement(s).	Why is this an issue? 33 minutes ago ▾ L96 🔗 ⚙️ ▾ 🔗 Code Smell ▾ 🔴 Critical ▾ 🔵 Open ▾ Not assigned ▾ 2min effort Comment 🔗 pitfall ▾
<input type="checkbox"/>	Add curly braces around the nested statement(s).	Why is this an issue? 33 minutes ago ▾ L132 🔗 ⚙️ ▾ 🔗 Code Smell ▾ 🔴 Critical ▾ 🔵 Open ▾ Not assigned ▾ 2min effort Comment 🔗 pitfall ▾

Major:

<input type="checkbox"/>	Remove this commented out code.	Why is this an issue? 35 minutes ago ▾ L29 🔗 ⚙️ ▾ 🔗 Code Smell ▾ 🔴 Major ▾ 🔵 Open ▾ Not assigned ▾ 5min effort Comment 🔗 unused ▾
<input type="checkbox"/>	Remove the unused function parameter "e" or rename it to "_e" to make intention explicit.	🔴 Why is this an issue? 36 minutes ago ▾ L18 🔗 ⚙️ ▾ 🔗 Code Smell ▾ 🔴 Major ▾ 🔵 Open ▾ Not assigned ▾ 5min effort Comment 🔗 unused ▾
<input type="checkbox"/>	Unexpected duplicate name monospace	Why is this an issue? 35 minutes ago ▾ L55 🔗 ⚙️ ▾ 🔗 Code Smell ▾ 🔴 Major ▾ 🔵 Open ▾ Not assigned ▾ 1min effort Comment 🔗 No tags ▾
<input type="checkbox"/>	Remove this deprecated "name" attribute.	Why is this an issue? 35 minutes ago ▾ L229 🔗 ⚙️ ▾ 🔗 Code Smell ▾ 🔴 Major ▾ 🔵 Open ▾ Not assigned ▾ 5min effort Comment 🔗 html5, obsolete ▾
<input type="checkbox"/>	Unexpected empty block	Why is this an issue? 35 minutes ago ▾ L132 🔗 ⚙️ ▾ 🔗 Code Smell ▾ 🔴 Major ▾ 🔵 Open ▾ Not assigned ▾ 1min effort Comment 🔗 No tags ▾
<input type="checkbox"/>	Remove the unused function parameter "username" or rename it to "_username" to make intention explicit.	🔴 Why is this an issue? 36 minutes ago ▾ L4 🔗 ⚙️ ▾ 🔗 Code Smell ▾ 🔴 Major ▾ 🔵 Open ▾ Not assigned ▾ 5min effort Comment 🔗 unused ▾

Minor:

☐
Remove the literal "true" boolean value.

Why is this an issue?
40 minutes ago
L111
🔗
🔍

🔗 Code Smell
🟢 Minor
🔵 Open
Not assigned
5min effort
Comment

🔗 clumsy

☐
Unnecessary semicolon.

Why is this an issue?
37 minutes ago
L28
🔗
🔍

🔗 Code Smell
🟢 Minor
🔵 Open
Not assigned
2min effort
Comment

🔗 unused

☐
Remove the unused local variable "num".

Why is this an issue?
40 minutes ago
L158
🔗
🔍

🔗 Code Smell
🟢 Minor
🔵 Open
Not assigned
5min effort
Comment

🔗 unused

☐
Remove the unused local variable "df".

Why is this an issue?
40 minutes ago
L162
🔗
🔍

🔗 Code Smell
🟢 Minor
🔵 Open
Not assigned
5min effort
Comment

🔗 unused

☐
Remove the unused local variable "df_ft".

Why is this an issue?
40 minutes ago
L163
🔗
🔍

🔗 Code Smell
🟢 Minor
🔵 Open
Not assigned
5min effort
Comment

🔗 unused

☐
Immediately return this expression instead of assigning it to the temporary variable "w".

Why is this an issue?
37 minutes ago
L39
🔗
🔍

🔗 Code Smell
🟢 Minor
🔵 Open
Not assigned
2min effort
Comment

🔗 clumsy

☐
Immediately return this expression instead of assigning it to the temporary variable "date_diff".

Why is this an issue?
37 minutes ago
L33
🔗
🔍

🔗 Code Smell
🟢 Minor
🔵 Open
Not assigned
2min effort
Comment

🔗 clumsy

All of the abovementioned code smells relate to code styling or unused variables and can be ignored at the current stage of development.

3.6. Coverage

Coverage is 0% as no unit testings were created.

3.7. Duplication















There are 7 repeated blocks of code found in the project.

Duplicated Lines (%) 2.6%		New Code: Since April 16, 2022	
		Duplicated Lines (%)	Duplicated Lines
charts-api/servers/templates/elevation.html		100%	64
charts-api/servers/templates/activity_recap.html		98.4%	63
charts-api/servers/templates/index.html		72.7%	48
<div> There are 92 hidden components with a score of 0.0%. <div>Show Them</div> </div>			

Most of the duplicate code comes from the template generating mechanism and can be addressed at the later stages of development to make the program lighter.

3.8. Technical Debt

SonarQube observed two debt hours with a 0.1% debt ratio. The project's maintainability rating is 'A'.

Technical Debt 2h 16min 		New Code: Since April 16, 2022
 charts-api/servers/server.py		33min
 workout-app/node_modules/flatted/php/flatted.php		32min
 chart-php/distance_travelled.php		26min
 workout-app/src/hooks/useWorkouts.js		15min
 charts-api/servers/templates/index.html		10min
 workout-app/src/hooks/useWorkoutsOld.js		7min
 workout-app/node_modules/istanbul-reports/lib/html/assets/base.css		3min
 workout-app/src/pages/graphs.js		2min
 workout-app/node_modules/@csstools/normalize.css/normalize.css		2min
 workout-app/node_modules/@csstools/normalize.css/opinionated.css		2min
 workout-app/node_modules/sanitize.css/sanitize.css		2min
 workout-app/src/pages/statistics.js		2min
 There are 83 hidden components with a score of 0. Show Them		

3.9. Conclusion

By analyzing the Sonar report, our group noticed that post-integration, many files and blocks of code are now unused and redundant and should be removed in future deliverables. Security issues, miscellaneous warnings and other suggestions were brought to our attention as well and will be addressed at later stages of development.

4. References

SonarQube documentation. SonarQube Documentation | SonarQube Docs. (n.d.). Retrieved April 16, 2022, from <https://docs.sonarqube.org/latest/>