BILKENT UNIVERSITY

ENGINEERING FACULTY

DEPARTMENT OF COMPUTER ENGINEERING

# CS 399

# SUMMER TRAINING

# REPORT

**Ömer Oktay Gültekin**

**21901413**

**Performed at**

## AKDOĞAN TEKNOLOJİ A.Ş.

**24.07.2023 – 18.08.2023**

# Table of Contents

# 1 Introduction

I did my previous internship at FNSS, a leading company in Turkey in the defense industry. Despite seeing the work environment of a corporate company, I gained a little experience in terms of what engineers do in the defense industry since they gave the toy projects to the computer engineering interns. Therefore, for my second internship, I decided to apply to startups in the defense industry where I can have more hands-on experience. For these reasons, I did my on-site internship at the Web Development department of Akdogan Technology, a young startup located at Ankara University Technopolis. The main focus area of Akdogan Technology is the defense industry; however, it is open to all requests from the other sectors' representatives as long as the requests align with the company's specialties. I chose that company since it integrates modern technologies such as web development frameworks and cloud technologies well into the defense industry.

My expectation from the company is to work on a project which will have an impact on the industry. During my internship, I had a chance to work on an early-stage project that has the potential to revolutionize the data management of defense industry products. That way, the company was able to satisfy my expectations. Thanks to the flexible nature of the startup, I had an opportunity to make critical decisions about the future of the project while developing it. The work that I have done is significant in the project since I created the backbone of the project, and it became nearly demo-ready for potential customers at the end of my internship.

The rest of the report gives much broader information about the company, the work done, and the outcomes of my internship. Also, the conclusion will be made about the internship at the end of the report.

# 2 Company Information

## 2.1 About the Company

Akdoğan Technology is a young startup founded in 2022 [1]. The company combines the innovative ideas and technical know-how gained through its experienced employees in the industry to form a solution partnership with leading defense industry companies [1]. The main focus area of the company is the defense industry [1]. However, it also has capabilities to respond to requests related to the Internet of Things (IoT), unmanned systems, and web technologies. The company gives consultation and software services in these specializations, specifically in the applications of embedded software development, web development, cloud technologies, data analytics, and artificial intelligence [1]. Apart from these, the company has 2 R&D projects. The one I worked on is called Prozek, and it aims to solve the problem of accessing log data of embedded systems and IoT devices by transforming the data into a simple and understandable format [2]. The second one is called Falcon Commander, which aims to be a command center for unmanned vehicles [3]. It has, in fact, started to be used in the Ulaq Armed Unmanned Surface Vessel in partnership with Meteksan Defence.

The company has two departments. One I interned in is the Web Development Department, and the other one is the Embedded Systems Development Department.

## 2.2 About the Department

The work the company does generally consists of both embedded systems and web development parts. That combination is not very common in the defense industry, and that is where the company is distinguished from its competitors. Therefore, the main responsibility of the web development department is developing web dashboards for embedded systems and IoT devices programmed or developed in the embedded systems development department. On the other hand, they are developing some pure web development projects, like the project I worked on.

## 2.3 About the Hardware and Software Systems

### 2.3.1 Hardware Systems

Akdogan Technology is a newly founded startup; therefore, there are not many complicated hardware systems in the office.

Every developer has a monitor, keyboard, and mouse pair but brings their own laptop to the office. There is one laptop and a bunch of Raspberry Pi cards in the office, which are used as a server. Thanks to the solution of the company for CI / CD pipelining, whenever a developer commits to some project, the project on the server gets updated. This way, while working on a project that needs to communicate with another project, the projects become always up to date and the developer gets correct data.

As for development, sometimes coding is done on a Raspberry Pi card. For some projects, clients send the hardware they need to program. Since the company is one of the subcontractors of the Ulaq Vessel Project, there is always some part of the ship found in the office such as a joystick that the ship captain needs to control the ship.

Specifically for the Falcon Commander project, there is one Steam Deck in the office since the software the company developed for Falcon Commander will run on this hardware. That way the operators will be able to control the ship through portable hardware.

### 2.3.2 Software Systems

The company uses two types of software systems to maximize their work efficiency. They either use software from third-party vendors or develop their own utility libraries.

The developers use the VS Code text editor for developing the projects. The company recommends using generative AI products, and the developers are advised to use GitHub Copilot and ChatGPT. The code repos are deployed on either GitHub or AWS CodeCommit Service based on their confidentiality level. AWS Cloud services are fully utilized in the company. The database of the project I worked on is managed on an EC2 machine. For task management, Jira software is used.

The company also has some utility libraries to use in their project. One is implemented so that they can directly access the most commonly used custom functions in C++ through their common library. The other library is developed for getting the hardware info the software runs on. That way, they can directly see IP addresses, MAC addresses, etc., through the functions of the library. The last library is still in active development and aims to collect all custom web components and widgets in one library that will allow them to use them in further projects. Some of the components I have developed during my internship are added to this library.

### 2.4 About my supervisor

- **Name:** Enes Akdoğan
- **Job Title:** Founder
- **Education:**
    - BS in Computer Science, Computer Engineering Department, Bilkent University, 2017.
    - MS in Defence Electronics And Software, Department of Defence Technologies and Systems, Başkent University, 2021.
- **Email:** enes@akdogan.tech

## 3 Work Done

### 3.1 Setup for the Work

On my first day in the office, I was given a work email and Jira account and I was asked to track and issue all tasks in Jira Software. Also, I was added to the organization account of the company in GitHub to access the repository of the project.

### 3.2 Planning the Project

At the start of my internship, the project I worked on was not much developed, and only the proof of concept was done. Even the project specifications were not fully determined. Therefore, the initial documents about the project were given to me to think about how to develop the project. Since there are no clients for the project, we need to come up with what functional requirements it should have, in other words, what features may the potential clients need. We came up with an improved design report for the project with my supervisor, who is the responsible employee for that project. Since Meteksan Defense is the closest partner company of Akdogan Technology and the biggest potential customer of the project, we decided that it would be wise to develop a prototype targeting their Ulaq Vessel, which the company knows a lot about.

### 3.3 Information About the Project

With my additions, the final overview of the project becomes the following: The project, named Prozek, aims to become a tool for collecting, processing, and understanding log data from embedded systems and IoT devices. It will serve as a data warehouse where information retrieved from log data is shown on dashboards through data visualizations to make the data accessible and understandable to even employees with no technical background. At the end of the development, Prozek is expected to be a de facto standard for mission debrief meetings by providing ad hoc reporting and analysis about the flight and detecting anomalies in any of the sensors of the platforms, such as Unmanned Aerial Vehicles (UAV). After my internship, the project received support from 1507 - TUBITAK SME R&D Start-up Support Programme.

## 3.4 Project initial condition

The proof of concept of the project was done using Flask framework for the backend, PostgreSQL for the database, and React framework for the front end. Before me, they cleaned the CSV data consisting of several flight data of one UAV flight in America found on the internet and fed it to the system. The initial dashboard had one page for showing the data in a table, a page for drawing charts based on some parameters from the data, and a map page where flights are shown. Appendix A shows the pages from the initial condition of the project. The map page is not shown since I did not do anything on the map and the reason I did not show the other remaining pages is that they had no significant content to show.

## 3.5 Finding Solutions to the Known Issues

There were many issues even in this small scope. Since the log data had rows for every second of the flight, the points charts needed to draw were enormous and charts were drawn after quite a while the page loads. Also, the constraints of the data were not known fully for how the data should be stored. In other words, is SQL enough to store time series data needed to be answered. Apart from these, the front end needed to be redesigned.

I started with researching React Chart Libraries to integrate into Prozek. The library should have to be fast, lightweight, and support a wide range of chart types. After my research, I found out that the Nivo chart library is approximately thirty times faster than the current chart library, Recharts [4]. Since speed is the top priority and Nivo is faster than all of the other chart library implementations, I decided to integrate the Nivo library into Prozek [5]. It also has the advantage of the modularity of the chart types, which means Prozek does not need to import all the library to use some of the charts [5]. While a wide range of charts type implementation of Nivo was a big plus, the lack of brushing and zoom ability for the charts was a tradeoff I needed to decide [5].

For how to store time series data, I found out that TimeScaleDB is a common solution that distinguishes itself from other alternatives by being open source and being an extension to PostgreSQL [6].

For front-end improvements, I inspected dashboards of various log visualization products and put them together in a document to inspire from them while developing the front end.

Before starting to code, we planned a scrum sprint on Jira software with my supervisor to create a 2-week plan for me to implement the features we decided on while writing the design report.

## 3.6 Implementing Product and Platform Dashboard

I split the application into three parts: Product, Platform, and Mission Dashboards. As an example in the context of Akıncı Bayraktar UAV, the product is the name of the Akıncı itself. It contains all the productions of the Akıncı. A platform is one of the UAVs in this production whereas a mission represents one flight of one platform.

Appendix B shows the tabs I implemented for the Product Dashboard. I started to change the front end by first changing the theme of the app to the dark theme as it is more appealing to the eye. Rather than doing it directly through CSS, I learned to use Ant Design Config Providers to change the CSS of Ant design components (see Appendix D Figure 9) [8]. Then, I integrated the Tabs component from the Ant design into the project to show each product's information in a more neat way. A summary page was needed to show how many platforms the product has, how many missions these platforms conduct, the total flight times in seconds of these missions, and the first and last time the product flew (see Figure 3). I used various Ant design UI components again to implement the tab such as Card and Filters. I used SQLAlchemy and Flask Restful APIs to get the info from the database.

Then, a Calendar tab was implemented to show the flights in a daily manner (see Figure 4). I used the Nivo Calendar chart type to implement this.

A final tab was implemented to provide a way for the user to see a summary of the platforms of the product and a way to reach the platforms page (see Figure 5). In other words, users can see the platforms and their missions and directly open their dashboards. I used Description and Collapse components from Ant design while implementing this tab.

Platform Dashboard has the same pages as the product page with more narrower information in them. For example, its calendar page only shows the days of the flights of that specific platform and in the other tab, it can only see its missions. The UI of the pages is not added to the report for simplicity.

## 3.7 Implementing Mission Dashboard

Appendix C shows the tabs I implemented for the Mission Dashboard. The following were done on this dashboard. A summary tab was added to show a detailed overview of the mission. Users can see the total flight time, distance, and start and end time of the flight (see Figure 10 for how data was sent from the backend). A map was added to show the flight route. Nivo Line Charts were integrated into the project to show speed over ground and longitude data whereas Nivo BoxPlot chart type was added to show the roll and yaw data on it. Lastly, the Nivo heatmap chart type was added to the project so that users can see the relation between Rudder and Throttle (see Figure 6). I needed to implement responsive functionality to Nivo charts because when the browser was resized the labels of the charts became incomprehensible and did not dynamically change their size. I implemented a solution that dynamically changes the font size, angle, and number of labels shown on the axis which I intend to open a pull request to integrate into the library. As a side note, Since the nature of the report does not allow me to put the code into the report, I do not share the source code, otherwise, the report could be more than a hundred pages long.

For the last two tabs, I modified the existing pages which are on Appendix A. The log table was not very useful before since the only thing it did was show the logs. My supervisor asked me to add filter capability to the table and recommended to me to use the Material UI Datagrid component instead of the Ant Design Table. However, after inspection, I decided that Datagrid is not enough for our data since it has no comparison capability. Then, I implement a custom filter for the table where the user can input a number and filter the row for the values less than, greater than, equal, and not equal to that of the input number. I also added a sorting capability for each row and changed the color of the values less than or greater than that of the given number (see Appendix C Figure 7).

Lastly, I changed the previous chart page implementation and greatly reduced the waiting time for the user to see the charts (see Appendix C Figure 8).

After all was done, I noticed that the data fetched from the backend was not optimized, that is every time the user switched tabs the data was loaded again. Since the data is static, I proposed to add useQuery which allows the application to store fetched data in browser storage so that the data is immediately available to use. On the last day of my internship, I learned the usage of useQuery and integrated it into the project (see Appendix C Figure 11 for the interface of useQuery to see the cached data).

# 4 Performance and Outcomes

## 4.1 Solving Complex Engineering Problems

There were many complex engineering problems I needed to solve in my internship starting from the design report. I used my report writing skills from school but we never wrote a report for such a big scope project. I managed to create the backbone of the project and started to implement the prototype.

The previous chart implementation code was very long and complicated and I needed to go through the code and understand what parts I needed to change to integrate the Nivo chart library. That is a common engineering problem since reading the bad-quality code written by other engineers is part of the job. In some places, the Nivo chart library did not comply with our requirements, which made me go through the source of the project and understand what part I needed to change. That was quite challenging considering the length of the code written by 182 developers [7]. Having had difficulty reading others' code, I left my code in a manner that everything is component and wrapped all the applicable components to use abstraction so that somebody after me could easily use these components without knowing their complexity. Making everything pluggable was a complex engineering problem to solve.

Lastly, I designed what the database should look like using my database systems course knowledge, which solves the complex engineering problem of storing time series data efficiently.

## 4.2 Recognizing Ethical and Professional Responsibilities

Akdogan Technology, as a defense company, expects high professional responsibilities from its employees in some projects in terms of confidentiality, especially in partnership with defense companies. All employees sign contracts with the company for not mentioning the confidential information of the projects. The company is also responsible for keeping the information of the partner companies secret. It can also be categorized as an ethical issue since some partner companies never check how Akdogan Technology uses their information. Therefore, I observe that trust is leading some of the communications of the company.

Akdogan Technology has professional responsibility for meeting the deadlines assigned by partner companies when the company is a subcontractor. For that, the scrum sprint approach was used and even though no penalty was applied to developers who did not meet the goals within the deadline, it was the developers' professional responsibility to complete the tasks before the deadline. Also, the developers keep track of the time they spend on each project by using Clockify software to better calculate the price of the services they give the other companies. I was impressed by this very ethical approach since the company cares about not gaining unfair money.

The company has flexible working hours and no one actually controls how much time you worked last week. It is employees' professional and ethical responsibility to fulfill their work hours.

Last but not least, every employee is professionally responsible for attending a meeting organized once a week.

## 4.3 Making Informed Judgments

Since I was involved in the decision process of the project I needed to make many informed judgments. My judgments had no effect on the global, environmental, or societal context because of the type of the project; however, they can partly have an effect on the economic context.

I did use the others performance metrics and my own tests on the libraries to decide Nivo chart library was best to use in the project. The tradeoffs were performance versus zoom and brushing abilities, as well as the range of the chart types. However, considering that the customers most of the time will not need zoom and brushing abilities, I decided that response time will be more important than additional abilities. If we use other libraries, we may need to create reports with charts beforehand, which will require time. Using Nivo will save time for Customers and thus, money.

Using useQuery rather than direct fetching using the Axios library was an informed judgment, too. Since the clients will upload their log data beforehand and use the dashboard to use the visualization abilities of the project afterward, all data on the dashboard can be static and once the data is fetched, there is no need to refetch the same data. Therefore, I integrated useQuery into the project by compromising the ability to show dynamic data, assuming there will be no.

I made informed judgments on designing the front end of the project by using my knowledge of similar product dashboards and analyzing them to understand what UI elements may be the most appealing to the clients.

## 4.4 Acquiring New Knowledge by Using Appropriate Learning Strategies

My internship was challenging in terms of expanding my knowledge since the project I worked on was in its initial stages and many decisions had to be taken. The database structure was not known even by my supervisor and many changes and additions were needed on the front end. Also, I had no familiarity with the Ant Design UI Library.

I used generative AI tools for initial research on data and visualization problems to narrow the scope of the research. Then, I compared the given solutions on the internet. I looked at others' comparisons as well as conducted my own comparisons to learn which libraries have the best performance for my specific use case. Then, I carefully examine the documentation of the libraries to learn how to use them.

When I was stuck and could not solve the problems on my own, I looked at programming forums on the internet like Stackoverflow and tutorials from YouTube before asking my supervisor.

## 4.5 Applying New Knowledge as Needed

I used my knowledge to improve the project. Although I did not implement the TimescaleDB solution to the database since the other works were more important for the demo of the project, I presented a report to my supervisor that contained the reasons for using the TimescaleDB and how it should be integrated into the project.

I used my learnings on Ant Design to add new pages and components to the Project which is explained in detail in Section 3. I used my knowledge of the Nivo chart library to solve the slow rendering problem of the previous chart implementation using Recharts. I also learned to useQuery hook in React and integrate it into the project to maximize the efficiency of data fetch from the database.

## 4.6 Awareness About Diversity, Equity, and Inclusion

Akdogan Technology has a narrow diversity, mainly due to being a start-up. Employees are mostly either new graduates or senior students, except my supervisor. Also, the employees are all men.

In terms of equity, all employees are equal. The company provides meals for everyone as long as they are at the office. Everyone has their own space for working and everyone is free to use any company properties.

Inclusivity is at the heart of Akdogan Technology. Anyone can object to the method of the job done for logical reasons. In my internship, I suggested different ideas than my supervisor about the implementation of the project many times. After discussing the pros and cons of both ideas, we always decided to implement the more logical ones. Therefore, even interns can participate in the decision process in Akdogan Technology.

# 5 Conclusions

Being an intern at Akdogan Technology was a wonderful experience for me since I had a chance to get hands-on experience in the real-life TUBITAK-supported project that will have a huge impact on the industry when it is completed. Also, I saw the start-up environment and how things in the defense industry work. I had a question in my mind about founding a start-up and I learned how to manage a start-up from my supervisor in my internship.

I wrote a design report for the project by using my knowledge from Object-oriented Software Engineering, and Technical Report Writing and Presentation courses I took at Bilkent. I used my solid foundation from the Database Systems course to analyze the existing database structure and how to optimize it. I was expected to justify my decisions every time I made critical decisions for the project. I would never have understood the structure of TimescaleDB and how it optimizes time-series data storage without my training at Bilkent. Since the libraries I used on the front end were open-source libraries, their documentation was not comprehensible especially if very specific customizations were needed. In these situations, I used my knowledge from programming language courses to predict how things should be done in that library. Since the solutions were not always on the Internet, that was challenging for me. On the other hand, I gained a lot from working on open-source libraries and trying to improve them. In fact, we will open a pull request for some of my complex implementations on these libraries so that others can benefit from them.

In summary, I learned many things and contributed to an R&D project from its proof-of-concept to nearly the end of the demo phase. I enjoyed my internship and appreciated my supervisor's valuable contribution. I would definitely recommend being an intern at Akdogan Technology.

## 6 References

[1] "Home."
https://akdogan.tech/. [Accessed: Oct 8, 2023].

[2] "Prozek."
https://akdogan.tech/prozek. [Accessed: Oct 8, 2023].

[3] "Falcon Commander."
https://akdogan.tech/falcon-commander. [Accessed: Oct 8, 2023].

[4] "A Comparison of Data Visualization Libraries for React."
https://www.capitalone.com/tech/software-engineering/comparison-data-visualization-libraries-for-react/. [Accessed: Oct 8, 2023].

[5] "Nivo."
https://nivo.rocks/. [Accessed: Oct 8, 2023].

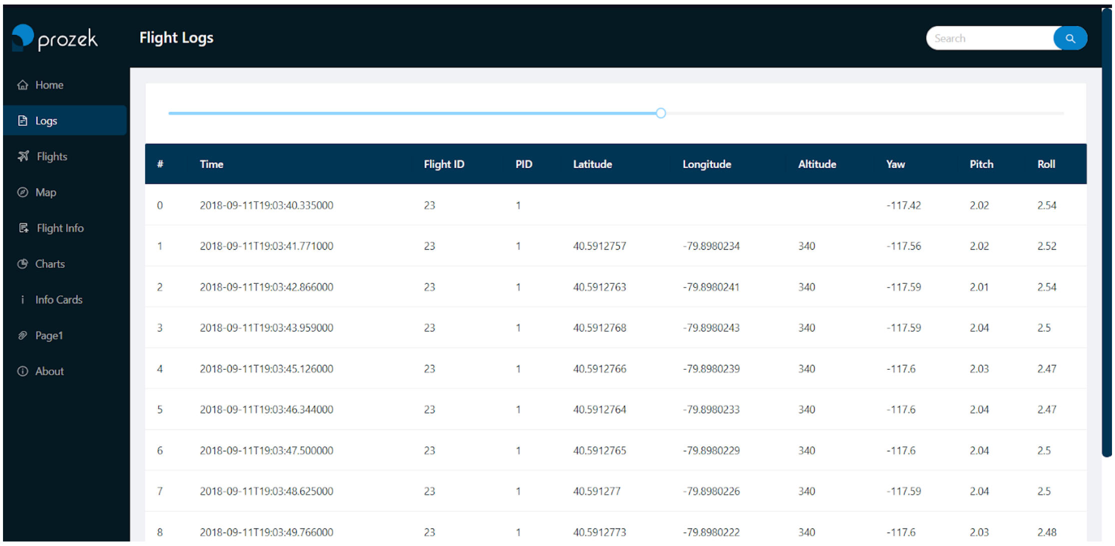
[6] "TimeScale"
https://www.timescale.com/. [Accessed: Oct 8, 2023].

[7] "Nivo GitHub"
https://github.com/plouc/nivo. [Accessed: Oct 8, 2023].

[8] "Ant Design."
https://ant.design/. [Accessed: Oct 8, 2023].

# 7 Appendices

## 7.1 Appendix A



| # | Time | Flight ID | PID | Latitude | Longitude | Altitude | Yaw | Pitch | Roll |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 2018-09-11T19:03:40.335000 | 23 | 1 |  |  |  | -117.42 | 2.02 | 2.54 |
| 1 | 2018-09-11T19:03:41.771000 | 23 | 1 | 40.5912757 | -79.8980234 | 340 | -117.56 | 2.02 | 2.52 |
| 2 | 2018-09-11T19:03:42.866000 | 23 | 1 | 40.5912763 | -79.8980241 | 340 | -117.59 | 2.01 | 2.54 |
| 3 | 2018-09-11T19:03:43.959000 | 23 | 1 | 40.5912768 | -79.8980243 | 340 | -117.59 | 2.04 | 2.5 |
| 4 | 2018-09-11T19:03:45.126000 | 23 | 1 | 40.5912766 | -79.8980239 | 340 | -117.6 | 2.03 | 2.47 |
| 5 | 2018-09-11T19:03:46.344000 | 23 | 1 | 40.5912764 | -79.8980233 | 340 | -117.6 | 2.04 | 2.47 |
| 6 | 2018-09-11T19:03:47.500000 | 23 | 1 | 40.5912765 | -79.8980229 | 340 | -117.6 | 2.04 | 2.5 |
| 7 | 2018-09-11T19:03:48.625000 | 23 | 1 | 40.591277 | -79.8980226 | 340 | -117.59 | 2.04 | 2.5 |
| 8 | 2018-09-11T19:03:49.766000 | 23 | 1 | 40.5912773 | -79.8980222 | 340 | -117.6 | 2.03 | 2.48 |

Figure 1: Initial Page for Displaying Logs

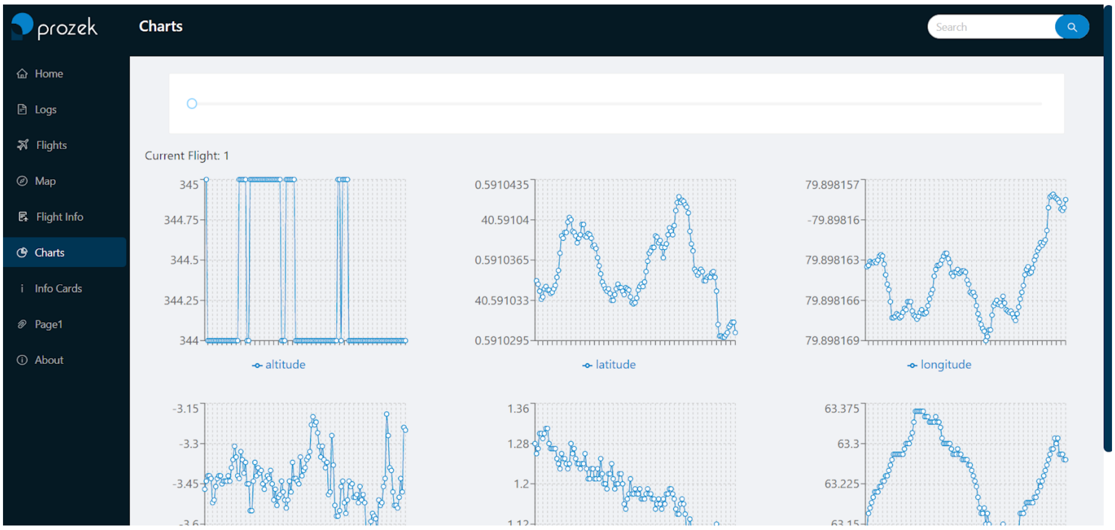
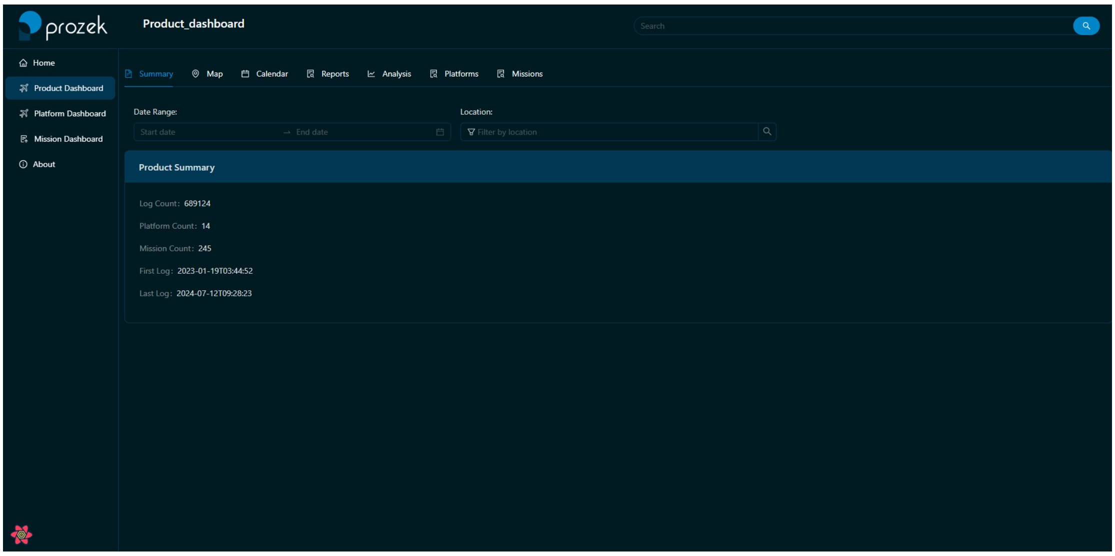


Figure 2: Initial Page for Displaying Charts

## 7.2 Appendix B



Date Range: Start date → End date

Location: Filter by location

Product Summary

Log Count: 689124

Platform Count: 14

Mission Count: 245

First Log: 2023-01-19T03:44:52

Last Log: 2024-07-12T09:28:23

Figure 3: Product Dashboard Summary Page

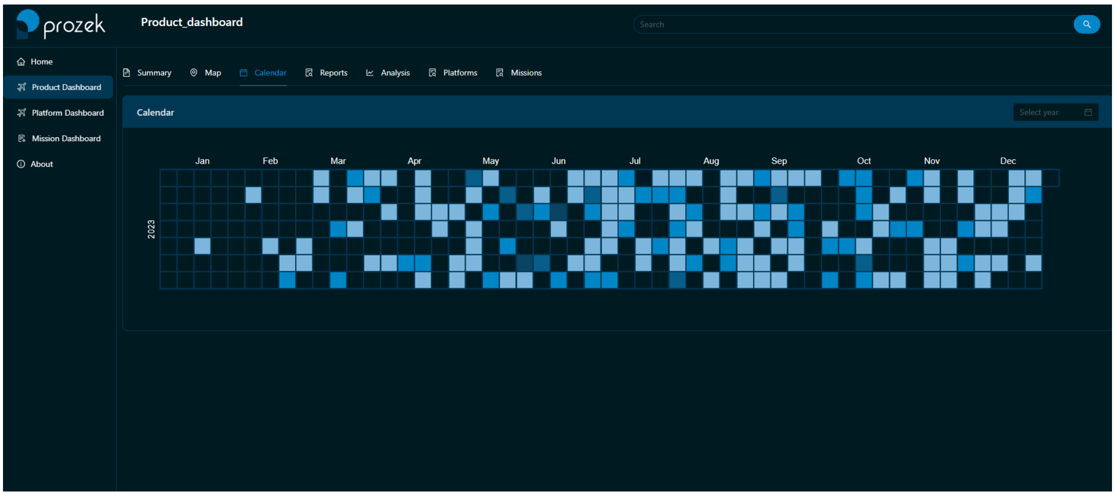
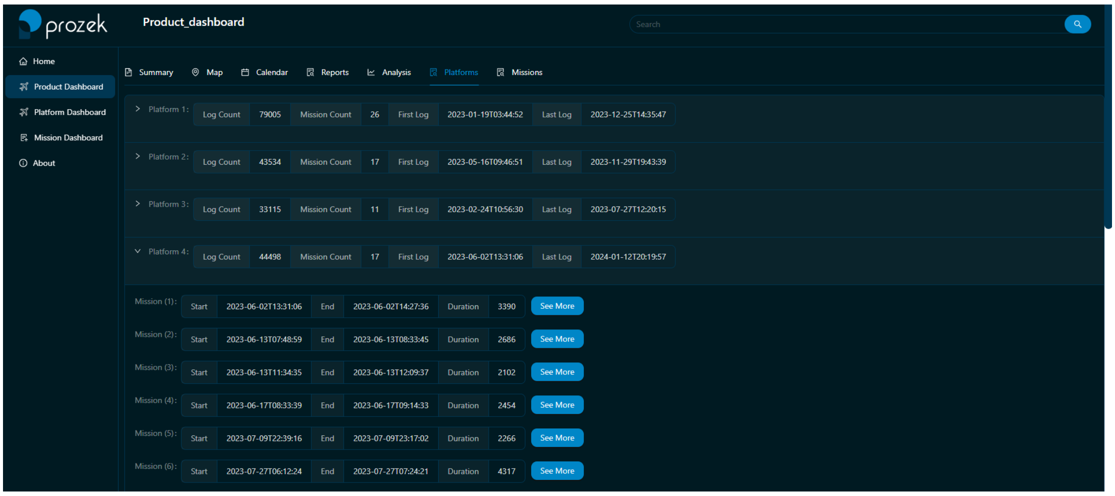


Figure 4: Product Dashboard Calendar Page

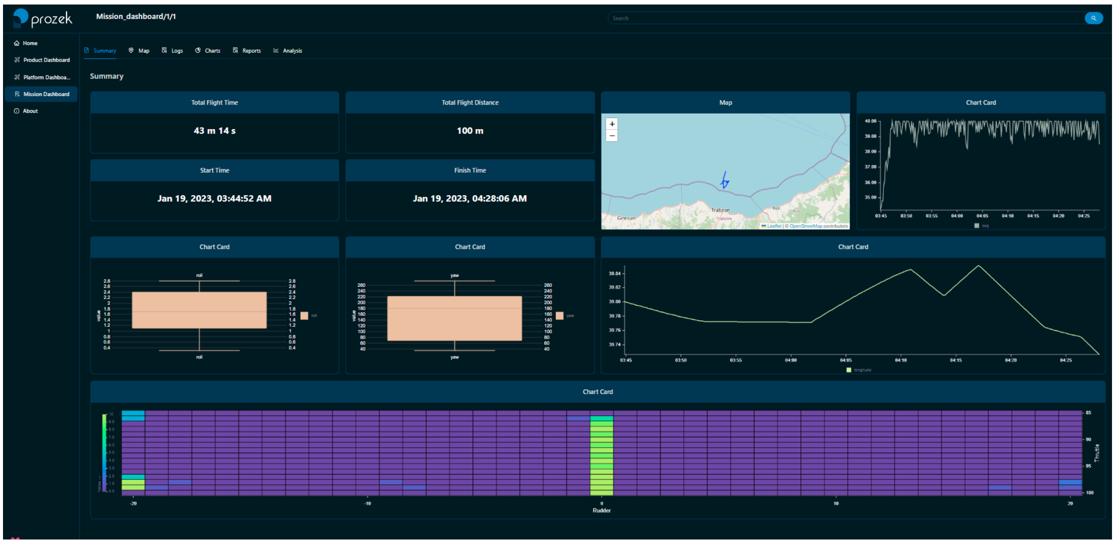


| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Platform 1: | Log Count | 79005 | Mission Count | 26 | First Log | 2023-01-19T03:44:52 | Last Log | 2023-12-25T14:35:47 |
| Platform 2: | Log Count | 43534 | Mission Count | 17 | First Log | 2023-05-16T09:46:51 | Last Log | 2023-11-29T19:43:39 |
| Platform 3: | Log Count | 33115 | Mission Count | 11 | First Log | 2023-02-24T10:56:30 | Last Log | 2023-07-27T12:20:15 |
| Platform 4: | Log Count | 44498 | Mission Count | 17 | First Log | 2023-06-02T13:31:06 | Last Log | 2024-01-12T20:19:57 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Mission (1): | Start | 2023-06-02T13:31:06 | End | 2023-06-02T14:27:36 | Duration | 3390 | See More |
| Mission (2): | Start | 2023-06-13T07:48:59 | End | 2023-06-13T08:33:45 | Duration | 2686 | See More |
| Mission (3): | Start | 2023-06-13T11:34:35 | End | 2023-06-13T12:09:37 | Duration | 2102 | See More |
| Mission (4): | Start | 2023-06-17T08:33:39 | End | 2023-06-17T09:14:33 | Duration | 2454 | See More |
| Mission (5): | Start | 2023-07-09T22:39:16 | End | 2023-07-09T23:17:02 | Duration | 2266 | See More |
| Mission (6): | Start | 2023-07-27T06:12:24 | End | 2023-07-27T07:24:21 | Duration | 4317 | See More |

Figure 5: Product Dashboard Platforms Page

## 7.3 Appendix C



Summary

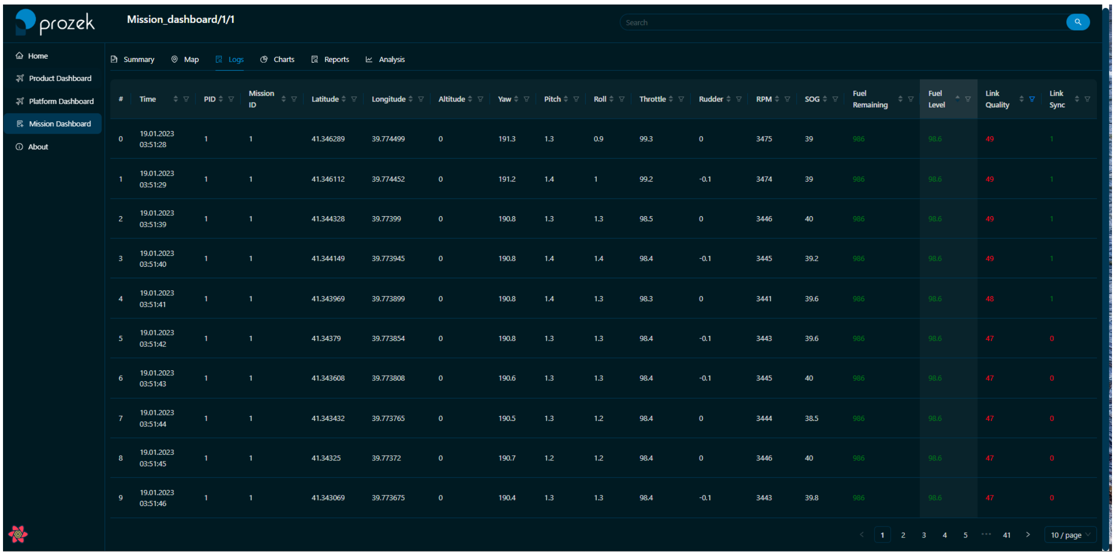
Total Flight Time: 43 m 14 s

Total Flight Distance: 100 m

Start Time: Jan 19, 2023, 03:44:52 AM

Finish Time: Jan 19, 2023, 04:28:06 AM

Figure 6: Flight Dashboard Summary Page



Mission_dashboard/1/1

| # | Time | PID | Mission ID | Latitude | Longitude | Altitude | Yaw | Pitch | Roll | Throttle | Rudder | RPM | SOG | Fuel Remaining | Fuel Level | Link Quality | Link Sync |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 19.01.2023 03:51:28 | 1 | 1 | 41.346289 | 39.774499 | 0 | 191.3 | 1.3 | 0.9 | 99.3 | 0 | 3475 | 39 | 986 | 98.6 | 49 | 1 |
| 1 | 19.01.2023 03:51:29 | 1 | 1 | 41.346112 | 39.774452 | 0 | 191.2 | 1.4 | 1 | 99.2 | -0.1 | 3474 | 39 | 986 | 98.6 | 49 | 1 |
| 2 | 19.01.2023 03:51:39 | 1 | 1 | 41.344328 | 39.77399 | 0 | 190.8 | 1.3 | 1.3 | 98.5 | 0 | 3446 | 40 | 986 | 98.6 | 49 | 1 |
| 3 | 19.01.2023 03:51:40 | 1 | 1 | 41.344149 | 39.773945 | 0 | 190.8 | 1.4 | 1.4 | 98.4 | -0.1 | 3445 | 39.2 | 986 | 98.6 | 49 | 1 |
| 4 | 19.01.2023 03:51:41 | 1 | 1 | 41.343969 | 39.773899 | 0 | 190.8 | 1.4 | 1.3 | 98.3 | 0 | 3441 | 39.6 | 986 | 98.6 | 48 | 1 |
| 5 | 19.01.2023 03:51:42 | 1 | 1 | 41.34379 | 39.773854 | 0 | 190.8 | 1.3 | 1.3 | 98.4 | -0.1 | 3443 | 39.6 | 986 | 98.6 | 47 | 0 |
| 6 | 19.01.2023 03:51:43 | 1 | 1 | 41.343608 | 39.773808 | 0 | 190.6 | 1.3 | 1.3 | 98.4 | -0.1 | 3445 | 40 | 986 | 98.6 | 47 | 0 |
| 7 | 19.01.2023 03:51:44 | 1 | 1 | 41.343432 | 39.773765 | 0 | 190.5 | 1.3 | 1.2 | 98.4 | 0 | 3444 | 38.5 | 986 | 98.6 | 47 | 0 |
| 8 | 19.01.2023 03:51:45 | 1 | 1 | 41.34325 | 39.77372 | 0 | 190.7 | 1.2 | 1.2 | 98.4 | 0 | 3446 | 40 | 986 | 98.6 | 47 | 0 |
| 9 | 19.01.2023 03:51:46 | 1 | 1 | 41.343069 | 39.773675 | 0 | 190.4 | 1.3 | 1.3 | 98.4 | -0.1 | 3443 | 39.8 | 986 | 98.6 | 47 | 0 |

< 1 2 3 4 5 ••• 41 > 10 / page

Figure 7: Flight Dashboard Summary Page

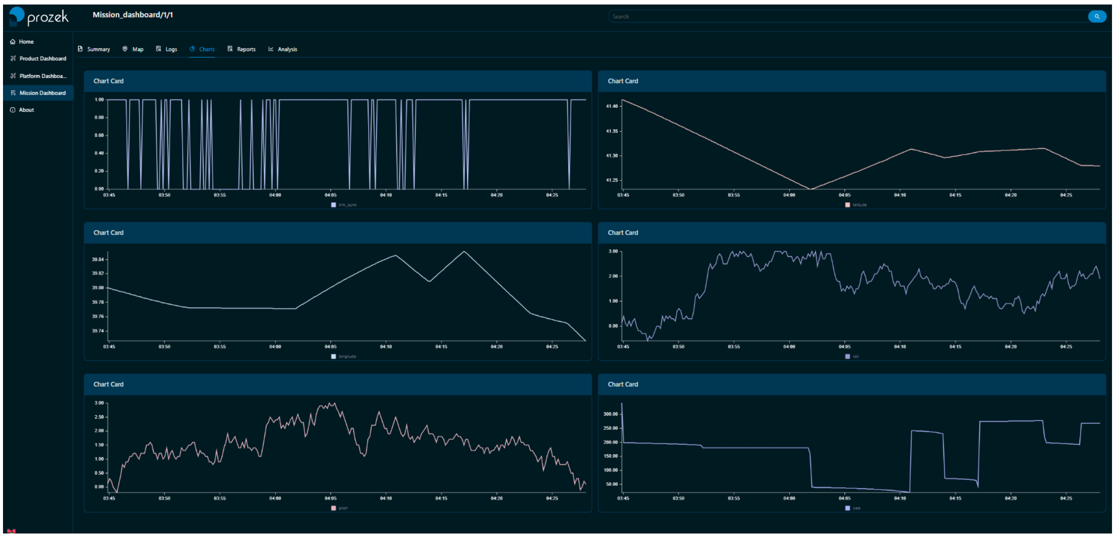
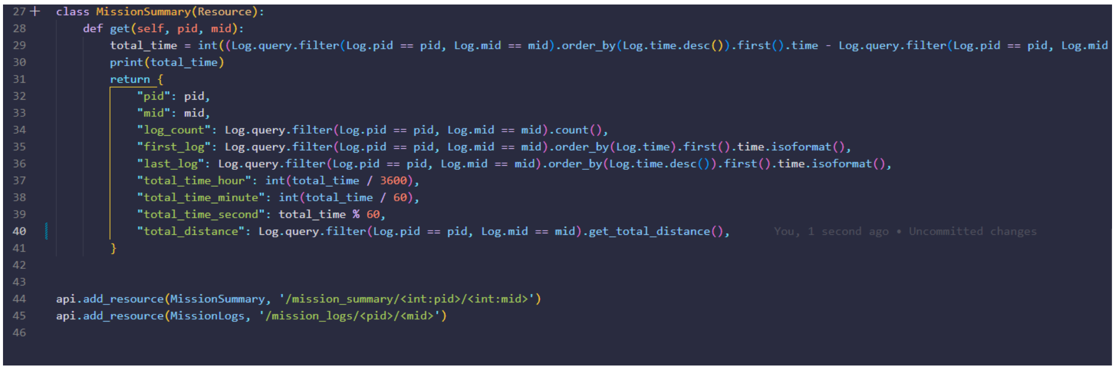


Figure 8: Flight Dashboard Charts Page

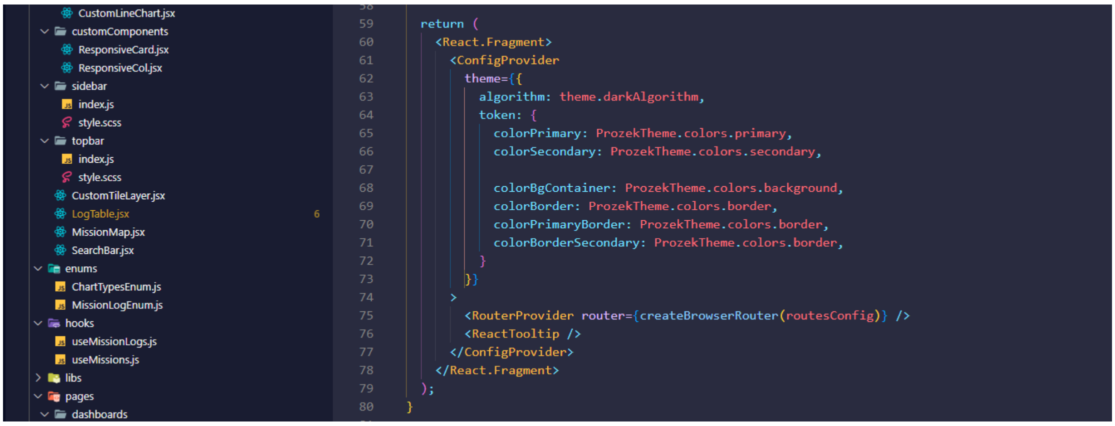
## 7.4 Appendix D



```python
class MissionSummary(Resource):
    def get(self, pid, mid):
        total_time = int((Log.query.filter(Log.pid == pid, Log.mid == mid).order_by(Log.time.desc()).first().time - Log.query.filter(Log.pid == pid, Log.mid
        print(total_time)
        return {
            "pid": pid,
            "mid": mid,
            "log_count": Log.query.filter(Log.pid == pid, Log.mid == mid).count(),
            "first_log": Log.query.filter(Log.pid == pid, Log.mid == mid).order_by(Log.time).first().time.isoformat(),
            "last_log": Log.query.filter(Log.pid == pid, Log.mid == mid).order_by(Log.time.desc()).first().time.isoformat(),
            "total_time_hour": int(total_time / 3600),
            "total_time_minute": int(total_time / 60),
            "total_time_second": total_time % 60,
            "total_distance": Log.query.filter(Log.pid == pid, Log.mid == mid).get_total_distance(),      You, 1 second ago • Uncommitted changes
        }


api.add_resource(MissionSummary, '/mission_summary/<int:pid>/<int:mid>')
api.add_resource(MissionLogs, '/mission_logs/<pid>/<mid>')
```

Figure 9: App.js File showing applying the dark theme to the project

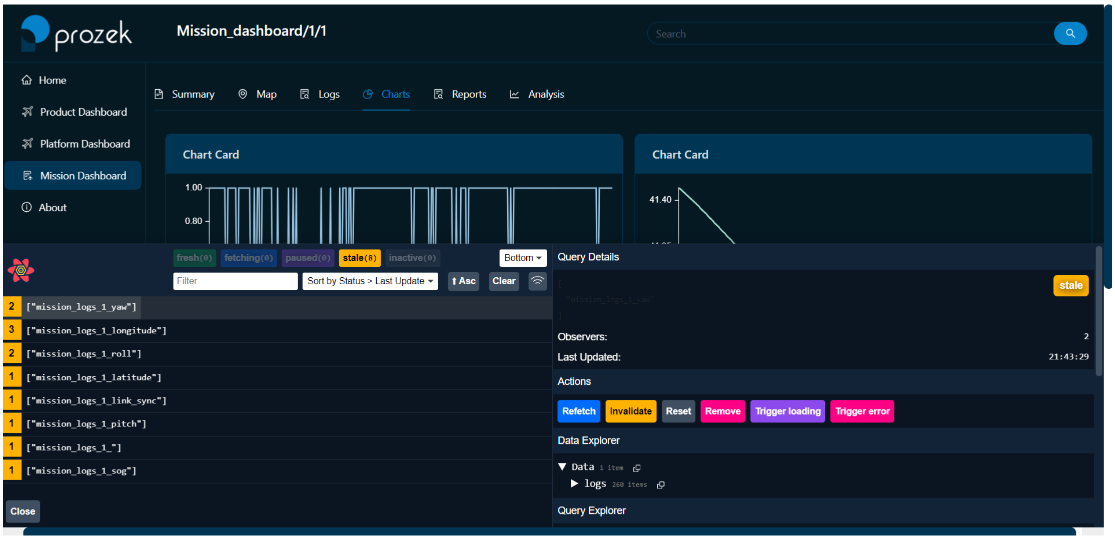


```
    CustomLineChart.jsx
  customComponents
    ResponsiveCard.jsx
    ResponsiveCol.jsx
  sidebar
    index.js
    style.scss
  topbar
    index.js
    style.scss
  CustomTileLayer.jsx
  LogTable.jsx                6
  MissionMap.jsx
  SearchBar.jsx
enums
  ChartTypesEnum.js
  MissionLogEnum.js
hooks
  useMissionLogs.js
  useMissions.js
libs
pages
  dashboards
```

```jsx
  return (
    <React.Fragment>
      <ConfigProvider
        theme={{
          algorithm: theme.darkAlgorithm,
          token: {
            colorPrimary: ProzekTheme.colors.primary,
            colorSecondary: ProzekTheme.colors.secondary,

            colorBgContainer: ProzekTheme.colors.background,
            colorBorder: ProzekTheme.colors.border,
            colorPrimaryBorder: ProzekTheme.colors.border,
            colorBorderSecondary: ProzekTheme.colors.border,
          }
        }}
      >
        <RouterProvider router={createBrowserRouter(routesConfig)} />
        <ReactTooltip />
      </ConfigProvider>
    </React.Fragment>
  );
}
```

Figure 10: Sending basic flight info from the backend



```
2 ["mission_logs_1_yaw"]
3 ["mission_logs_1_longitude"]
2 ["mission_logs_1_roll"]
1 ["mission_logs_1_latitude"]
1 ["mission_logs_1_link_sync"]
1 ["mission_logs_1_pitch"]
1 ["mission_logs_1_"]
1 ["mission_logs_1_sog"]
```

```
[
  "mission_logs_1_yaw"
]
Observers:        2
Last Updated:     21:43:29
▼ Data 1 item
  ▶ logs 260 items
```

Figure 10: The interface of useQuery for seeing the cached data on browser

# Self-Checklist for Your Report

*Please check the items here before submitting your report. This signed checklist should be the final page of your report.*

- ☑ Did you provide detailed information about the work you did?
- ☑ Is supervisor information included?
- ☑ Did you use the Report Template to prepare your report, so that it has a cover page, has all sections and subsections specified in the Table of Contents, and uses the required section names?
- ☑ Did you follow the style guidelines?
- ☑ Does your report look professionally written?
- ☑ Does your report include all necessary References, and proper citations to them in the body?
- ☑ Did you remove all explanations from the Report Template, which are marked with yellow color? Did you modify all text marked with green according to your case?

Signature: ______________________