By

Checkmate

Submitted to
The University of Roehampton

University of Roehampton London

24th April 2022

Software Engineering Group Report

CMP020N204S

Abstract

The world population has rapidly increased in the past few centuries. According to an article from the Royal Society publishing, in the last 70 years, the world's population increased by a staggering 4 billion people [2]. Companies around the world use databases that help them better understand the different factors that contribute to this growth, the most populated areas, what are the next steps that need to be taken to prevent the overpopulation of different areas and more. Some projections for the world population predict that in the next 50 years the world population growth will stagnate or even decline in the most developed parts of the world, whilst the least developed areas will see rapid growth. This project aims to help the analysts gather specific information regarding populations from around the world.

Declaration

I hereby certify that this report constitutes my own work, that where the language of others is used, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions, or writings of others. I declare that this report describes the original work that has not been previously presented for the award of any other degree of any other institution.

Date: 24/04/2022

Signed by

A.Bogdan T.Head Z.Oulhadj M.K

Acknowledgements

We (“Checkmate group”) would like to thank our stakeholders including the University of Roehampton and our client for their cooperation throughout the development of this project. They have provided us with the necessary requirements that we as a group needed to complete this project. The university has also assisted us by providing resources to us such as video tutorials and lab classes which we used for training purposes.

Table of Contents

1. Introduction

We ("Checkmate group") have been hired by a company to help them acquire population information in a user friendly, easy to use manner by developing a platform which they will be able to use to subsequently increase their productivity and provide many other potential benefits.

As per our client's requirements [1], our application needs to use certain technologies and methodologies for the web application to be fully integrated with its systems. The list of requirements is as follows:

- Front-end must use the Pug framework
- Back-end must use the Node.js framework
- Front and back-ends must communicate via Express.js
- Application must use a database for storing population information
- Application must use a VCS (Version Control System)
- Project needs to be deployed as a Docker container

In addition to these requirements we have also been tasked with working as a Scrum team and therefore, have assigned each team member to a task:

- Zakariya Oulhadj (Team Leader 1)
- Samoil Bogdan Adascalului (Team Leader 2)
- Taylor Head (Scrum Master)
- Kezzy Mohamed (Project Owner)

A Scrum team is often formed by a Scrum Master, Product Owner, Team Leaders and Developers. The entire project will be split into 4 stages which will be called "Sprints" that are one month apart from each other. The Project Owners are responsible for looking over the entire project and making sure everything is on track regarding the deadlines. The Scrum Master is responsible for ensuring that the Scrum guidelines are being followed and the Team Leaders ensure that each team member is doing their respective tasks. The benefit of assigning a role to each team member means that everyone within the group has a clearer understanding of what they are expected to do.

1.1 Problems that will be addressed

The organisation that we work for requires easy access to their population information. Currently, the organisation has little to no access to this information and therefore, we ("Checkmate group") need to address this. We as a group need to gather the requirements from the client [1], design and implement a project to address those requirements which will give the organisation easily and user-friendly access to this population data.

1.2 Aims

The project consists of three main aims that we hope to achieve by the end of the project. The first of these aims is to develop a platform according to the client's requirements that allow the business to use their population information. Secondly, support must be provided during and beyond the project's development. This will ensure that any queries are fully addressed and resolved. Finally, the platform must be developed in such a way that allows the project to be easily adapted if the client’s requirements change. Below is a list of these aims mention:

- Meet all client requirements
- Population information easily accessible
- Easy to use platform
- To meet project deadline
- To provide continued support past the project deadline

1.3 Objectives

For our aims to be completed, we need to set out specific objectives that need to be met. An aim is a general statement that we as a group want to meet. An objective is a more concrete task that will fulfil that desired aim. Below is a list of these objects that we want to complete:

- Implementing an authentication system
- Restrict database access to unregistered users
- Design an easily accessible and user-friendly contact form
- Design a user-friendly application that connects our website to the database
- To meet each of the 4 sprint deadlines
- Implement database interactions including editing data, adding new data etc.
- Making use of the Docker system

1.4 Legal, Social, Ethical and Professional Considerations

Prior to any work commencing we as a group need to take into consideration the legal, social and ethical issues that may affect the project. By taking all of these into consideration we will have a more refined and definitive set of requirements that will be followed.

1.4.1 Legality

The project is licensed under GPL-3.0 Licence (GNU Public Licence version 3) [3]. This means that the client has:

- the freedom to use the software for any purpose,
- the freedom to change the software to suit their needs,
- the freedom to share the software with their friends and neighbours, and
- the freedom to share the changes you make.

We find that this gives us and the client the greatest levels of flexibility during and after the project's development.

1.4.2 Social

There is one social issue that may affect the project and that is accessible to users. Some users may require special needs since they may be disabled in one form or another and therefore, we need to take this into consideration. For example, adding features that help those users.

1.4.3 Ethical

There are multiple ethical concerns that need to be addressed. The first of which is data misuse. We will have access to primarily two pieces of data which are the population information and user data. It’s therefore critical that we ensure this data is not shared outside the scope of this project or used for any other purpose than the development of the project. This has been noted in the group's Code of Conduct [5].

1.4.4 Code of Conduct

The code of conduct is a professional document that aims to guide those whom it may concern regarding the team and projects standards, responsibilities, disciplinary actions and technical details. We as a team will refer to this code of conduct throughout the project to ensure all the standards are being followed.

1.5 Background

1.5.1 Client

The client that we will be developing this project is Dr Sobhan Y. Tehrani. This client has provided us with the initial requirements for the project [1]. In addition to this, we will showcase our progress with our client at each Sprint (every month) during the project’s development.

1.5.2 Project suitability for BSc students?

This project consists of multiple elements including programming, team collaboration and project management. We believe that this is suitable as this gives us the opportunity to develop our skills as a team working on various aspects of this project. In addition to this, it will give us a greater insight into all stages of a project including requirements gathering, design, implementation and finally evaluation.

1.6 Report overview

This report is structured into three main parts, requirements gathering, design and implementation. The requirements gathering is important as it provides the foundation on which the project will be developed. Therefore, it is vital the requirements we gather are accurate and in line with what the client expects. Based on these requirements, we will move on to the design phases. The second part of this report is the project's implementation and evaluation. These sections below will be discussed in great detail.

2. Technology Review

Due to our organisation specification, we were required to make a crud application for smooth and efficient operation of the business.

As the Scrum team, we reviewed the technologies, which consisted of the web front-end using PUG and the backend using Node.js with communications for the front and back via Express.js and the storage using MySQL. This was the base technology review we needed to work with, and we started research with existing sources provided by Roehampton University. As the project went on, we out scoped other technologies listed below to bring the project further together.

2.1 Version Control

Working on the project simultaneously is vital when working as a group and therefore, we had to make an important decision early on regarding which version control system to use. We took into consideration three main options, GitHub, BitBucket and GitLab. During our first official meeting [4] we discussed and analysed all three options and concluded that we should use GitHub. This was primarily due to the wide range of features that GitHub provides such as GitHub Issues for bug reporting, project management via GitHub's project boards, repository wiki page and more. In addition to this, two of the group's members already had prior knowledge of GitHub and therefore, it was already a strong candidate.

2.2 Packages

List of technologies that will be used:

- **Visual Studio Code (1.66.2)** - Text Editor used for our team members
- **GitHub Desktop (2.9.12)** - Git project management
- **Node.js (8.5.5)** - Used because of requirements
- **PUG (3.0.2)** - Used because of requirements
- **Express.js (4.16.1)** - Used because of requirements
- **Bootstrap (5.1.3)** - Style sheets
- **Bootstrap Icons (1.8.1)** - SVG icons by Bootstrap
- **Travis CI** - Continuous Integration
- **Cypress (9.5.4)** - Unit testing
- **Docker (4.5.0)** - Container for project
- **MySQL (2.18.1)** - Database language

A full detailed explanation as to how each package will be used can be seen in the Appendix.

3. Design

The design phase of this project lasted 2 weeks, starting on 23/01/22 to 6/02/22. One of the fundamental aspects that we as a group discussed before any formal development took place was the methodology we will use as a team. Since we are a Scum team we, therefore, agreed to follow the Agile SDLC (Software Development Life Cycle) to give us the ability to quickly design, implement and test each feature during development. This is greatly beneficial, unlike the Waterfall model where we would be restricted in regards to what tasks we could do since every task relies on the previous task being completed.

The "Checkmate Group" model of development is as follows:

1. Discuss a potential feature with team members
2. Design feature

3. Create a new branch for feature
4. Implement feature
5. Test feature
6. Merge branch with main

The next area that we focused our attention on was the platform itself. We designed a website that will consist of a “front-end” and a “back-end”. The front-end will refer to the part of the site that will be visible to the end-user whereas the back-end is the internal process that manages the site.

3.1 Front-end

The front-end will be the area of the site to showcase the project itself. This will include information about the Checkmate group team members, what technologies have been used as well as giving the user a way of contacting the team. As mentioned in section 2.2, the Bootstrap CSS library will be used as this will allow us to shift our attention to others rather than basic styling. The benefit of this is that the site will look great and also provide means of easily changing this styling in the future based on the client’s requirements.

3.2 Back-end

The client listed authentication as an optional requirement but mentioned that it would be a greatly beneficial feature to have. Due to other aspects of the project going to plan, we were in a position that allowed us to undertake such a task and therefore, we agreed that we were going to design and implement an authentication system for the website.

With the report viewer already in the process of being designed we wanted to take this opportunity to link together the database viewer and authentication. We did this by protecting the database viewer behind a login screen. The reason for this is that as mentioned in the Ethical issues section, the database may contain confidential information so, only users with appropriate permissions should be able to view it.

3.3 Report Viewer

The concept of the report viewer came about after an initial redesign near the beginning of the project. Originally, we wanted to have a navbar dropdown menu with all the different reports. Over several meetings, however, we decided against this as this would require a unique page per report. In the short term, this would not be an issue however, in the long term, if the client wanted to add more reports then this would quickly result in an unmanageable amount of pages.

Instead, we decided we would implement a dashboard-like viewer that consists of all the features required on a single page. The benefit is that viewing different types of reports will be contained to a set of consistent tools and features. The reports themselves will simply be tables that get included in that one page and the user can simply select a report from a dropdown menu.

4. Implementation

The entire project was implemented over four months and four sprints. Since the version control system being used is GitHub our code was hosted here: https://github.com/checkmate-group/coursework

4.1 Front-end

The term “front-end” refers to the elements of the site that can be viewed by the user (without authentication) when they visit the site. Figure 1 shows the front end (highlighted in purple) and consists of the home, about, features and contact pages. Each of these pages was developed with the goal in mind being to showcase the project from a clients/user's perspective. For example, the about page showcases each team member and all the features that they have worked on. Furthermore, the features page contains technical information regarding the internal workings of the platform.

4.2 Private front-end

The private “front-end” (highlighted in dark blue) is similar to the front-end with the exception that it is only accessible as an authenticated user. More specifically, the page can only be viewed once the “back-end” of the site has authenticated the user and then redirected them to that page.

4.3 Back-end

A key aspect of this project was the “back-end”. This relates to the internal code of the website that interacts with the webserver such as routing, database communication and authentication. This section is highlighted in green in the image below.

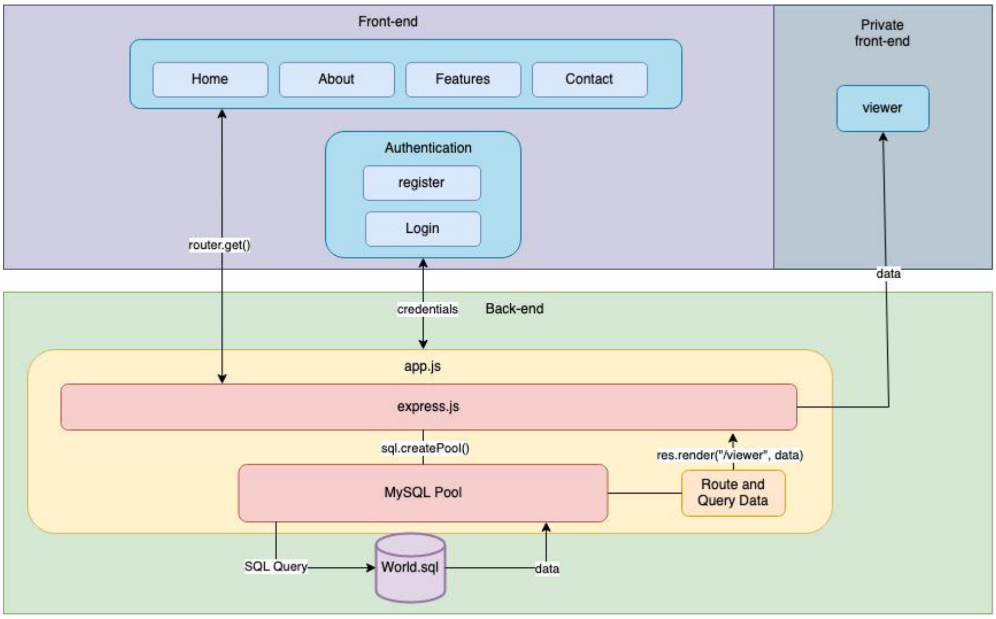

Figure 1: Technical overview

5. Evaluation

Nearing the end of the project we have close to 600 commits, 45 merges and 16 issues closed. For sprints 1, 2 and 3 we received grades 68, 75, and 72. We as a group believe that the project has been an overall success due to the high grades that we have received so far and therefore, the client is satisfied with the progress being made.

5.1 Performance

In terms of the performance of the site, we believe that it is of a satisfactory level meaning that for the time being it is ok, however, can be further improved over time. Shown in Appendix (Figure 4) is a table that consists of the loading times for each page on the site. What it shows is that on average the static web pages take 6.10ms to load. On the other hand, the dynamic web pages i.e. the page that reads data from a database take 20.78ms.

Regarding the report loading times, we had a meeting discussing how we should reduce this time and concluded that we should change the code such that we only connect once to the database at the start and then reuse that existing connection rather than establishing a new connection per report.

5.2 Team collaboration

Regularly the team held group meetings in which we discussed the progress we have made as well as future work that had to be completed. This was greatly beneficial throughout the project's development as this ensured the team was on the same page and that all team members were aligned with the client's requirements. Each meeting that took place was recorded within the meeting minutes document [4].

The figure below shows the commit distribution i.e. the proportion of work each of the team members made throughout the project's development.

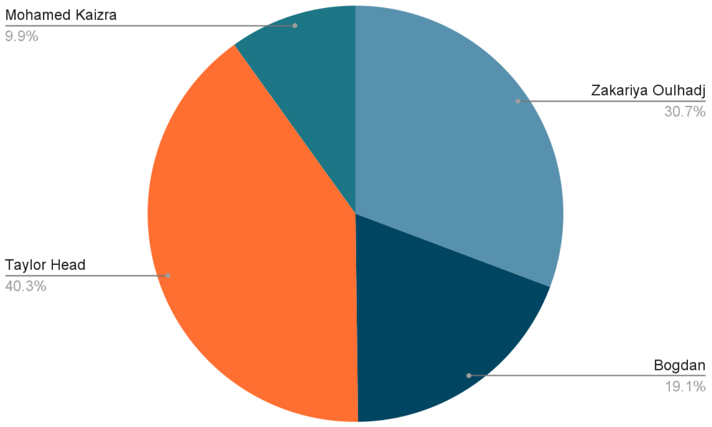

Figure 2: Commit distribution

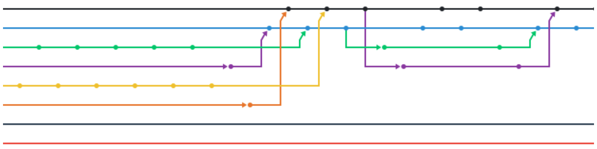

Figure 3: Merging

On one occasion the team leader (Zakariya) and the scrum master (Taylor) were working on a particular feature related to the core of the website. They had both worked on the same set of files and this had resulted in a merge conflict. For 5 hours, they worked together to resolve this issue which is another show of collaboration within the team.

5.3 Related Work

As we tried to develop our web app, we found out there are quite a few different apps online that do almost the same things as ours. So our main concern was how to make it different.
One of the features we added is security. Only registered users can have access to our database.This way we can limit the number of users that have access to our system.Another addition is our Contact Us feature. This is our way of communicating with the clients. As soon as the user enters the details (name, email, message) our partner, FormSubmit.io, will redirect all the information to all the members' emails.

Another difference between our application and the rest is that ours have an edit feature.The users can add, delete, and edit information.This way we can update the database straight from the user interface without having any background in coding.

6. Conclusion

The project is now nearing its conclusion, therefore, we now want to discuss how we feel the project went, the positives and negatives as well as the areas of the project that need to be further developed for anyone who decides to continue working on this platform.

6.1 Reflection

6.1.1 What went well?

At the end of this project, we were able to gauge how everything went through the client's feedback as well as through regular team meetings. As a team, we have concluded that the project went as expected with little to no hiccups along the way. During the early phases of development, we had no experience working together and therefore, we were unsure as to how well our collaboration will go. Over time, however, we learnt each other's strengths and weaknesses and as a team, we were able to work collectively we realised the following:

- Improves productivity
- Increases innovation
- Increases collaboration
- Builds trust among team members

6.1.2 What did not go well?

6.1.2.1 Team Cooperation

Because we didn't know each other too well and didn't know what to expect from each other, it took some time for us to find our rhythm. Because of it, we had team members that did not contribute equally and we had encountered a few problems due to being short-staffed.

6.1.2.2 Database Connection

Within our App.js, we had problems with our connection pool where we couldn't connect to the localhost IP automatically with Docker. This meant you had to keep updating the IP address manually, which significantly increased development time as a result of having to repeatedly rebuild the Docker project container.

6.1.2.3 Version Control

We also ran into many problems with our version control system (GitHub), which had a steep learning curve for our group since this was the first time using this technology which came with many issues including merging and handling data and more.

6.1.3 What would we have done differently?

6.1.3.1 Time Allocation

Looking back, we realised that there are a few things we could have done differently. One of them being time allocation. At first, we allocated too much time to jobs that didn’t require as much time as we allocated. One example is that we took 2 weeks to finish Sprint 1 when we could have worked more on the actual coding part of the project rather than spending so much setting up the GitHub and all the other tools we used.

6.1.3.2 Work Assignment

One other thing that could be changed in the future is the job assignment. After this experience, we realise that rather than letting everyone do whatever they feel comfortable doing, the most beneficial way to do it is for everyone to do a bit of everything. The experience of doing a bit of frontend and backend could be very beneficial for each of the members' careers.
Another issue that we encountered and that could have been handled differently is enforcing the code of conduct. Sticking more rigorously to the code of conduct standards, we could have handled the problems that we encountered more efficiently.

6.1.3.3 Prior Research

Also doing our research regarding the technologies that we are going to use, before starting the actual coding, would have made us more confident, more reliable and also faster in solving the problems that we encountered during the implementation process.

6.1.3.4 Git Branches

An issue which was discussed above in section 6.1.2, new features were implemented in unique branches to eliminate any potential conflicts when working together. An improvement we could make for any future work is to reduce the size of work per branch. For example, we created a branch called “core-redesign” that was responsible for reworking the entire core of the project. The issue with this is that once we had to merge this branch back with all of the other branches we received Git merge conflicts since a lot of changes were made and therefore, some features were incompatible.

6.2 Future Work

This project was limited to around four months and therefore, we could only implement the core requirements laid out by the client. The list below shows a few of the ideas that we as a group could implement in the future.

- Password hashing for authentication (SHA-256)
- Minifying assets for increased website performance
- Implement a chat room feature so that users can access support from one of our devs
- Implement Google analytics to the site to view traffic on the site
- Mobile app
- Custom CSS styles
- Solving any outstanding GitHub issues
- Increase database connection efficiency

Additional avenues could be explored such as implementing APIs that would update the dataset in real-time so that the website doesn't get outdated. This would increase the ease of use when working with the population information dataset.

Another feature that could be implemented that we were unable to do within the given timeframe was a sidebar instead of a navbar which was suggested by our Team Leader 2 (Bogdan). The idea behind this is that it's a unique feature that other projects may not have implemented.

Currently, the project was developed locally. However, once the site is hosted online, it would be a good idea to focus on Search Engine Optimisation to increase traffic to the site and subsequently benefit the company.

Before this can be done, however, a suitable website hosting platform must be found as many different companies provide different features. Any team that takes this project on in the future will need to analyse this extensively to make an informed decision.

References

[1] “Software Engineering (CMP020N204S) - Coursework requirements”, *University of Roehampton*.
https://moodle.roehampton.ac.uk/pluginfile.php/2635285/mod_resource/content/1/Coursework.pdf

[2] “Human population and demographic transition”, *Royal Society Publishing*.
https://royalsocietypublishing.org/doi/10.1098/rstb.2009.0137

[3] “The GNU General Public Licence v3.0 - GNU Project,” *Free Software Foundation*.
https://www.gnu.org/licenses/gpl-3.0.en.html

[4] “Meeting minutes”, *Checkmate group*.
https://github.com/checkmate-group/coursework/blob/develop/documents/meetings.md

[5] “Code of Conduct”, *Checkmate group*.
https://github.com/checkmate-group/coursework/blob/develop/CODE_OF_CONDUCT.md

[6] “Use case Diagram”, *Checkmate group*.
https://github.com/checkmate-group/coursework/blob/develop/documents/use_cases/use_case.md

[7] “User stories”, *Checkmate group*.
https://github.com/checkmate-group/coursework/blob/develop/documents/use_cases/user_story.md

Appendix

The appendix here includes text, images and tables that support our report.

Package Usage

Text Editor

The primary development platform we ("Checkmate group") used as a group was Visual Studio Code. We decided to use this because of its built-in packages, wide range of language support as well as first-class packages for Docker and Express.js. It also provides built-in tools such as the terminal which we can use to interact with Git via the command line.

Version Control

GitHub was used for the project management where the project is hosted, and all members can contribute at the same time with a lot of inbuilt features where we can manage and review what members have done and manage the project as a whole.

Package Management

Node.js was used for the backend of the project because it provides the core framework and will contribute to the smooth and efficient operation as it also provides numerous packages. As well as being safe, this is an industry known framework that companies will have support and confidence with.

Templating Engine

Pug was used for the front end of the project over HTML; due to a few factors, we found that PUG is very simple for the client to understand and has simplified syntax, allowing the code to be more readable. Pug already has a template engine for Node, which could be very easy for the client to expand.

Web Framework

Express.js has been used to connect the front and backend of the project; we used this as within the industry Express.js is a leading technology for backend development with its easy Library which is very large and helped the project a lot as well as being server ready.

Styling

Bootstrap/Bootstrap Icons was used since it's the most familiar and has some added benefits and that's why it's the default technology to pick for customizable frontend was mainly due to the fact it has fewer cross-browser bugs since we want our project to be multi-platform.

Continuous Integration

Travis CI was used so that we could monitor our project by getting a test, and generating an email alert when it passes or fails, which helps the project make sure that everything is running and bugs/errors are not pushed to the main branch. This was mainly picked to check the code quality of all members and to set a standard.

Unit Testing

Cypress.js Unit testing environment was used for an end-to-end testing area, which allows us to check and test every area of our code, and the benefit is automated, allowing the client to test the sites and hold records of tests. This was used over Jest for being more user friendly and developer friendly with it also being open source.

Container

Docker was used to allow the project a fast deployment, getting it all hosted within a cluster of containers and allowing us to have easy migrations for when we deliver this to the client. Security is always our main goal, and with docker, everything is running within a container which is hard to open from the outside once set up correctly.

Database

MySQL was used to it already having enterprise-grade management, and when dealing with sensitive data like world populations and user logins we need a solid base with being globally renowned. This was the best option due to it being flexible and being able to attach to our backend.

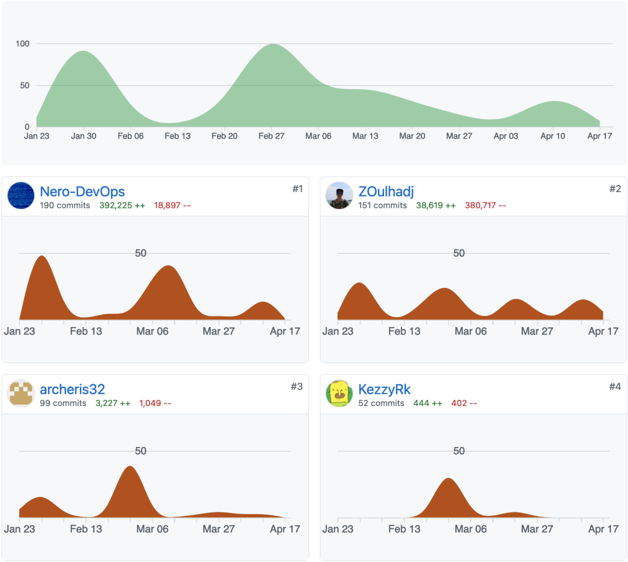

Figure 2: GitHub personal contributions

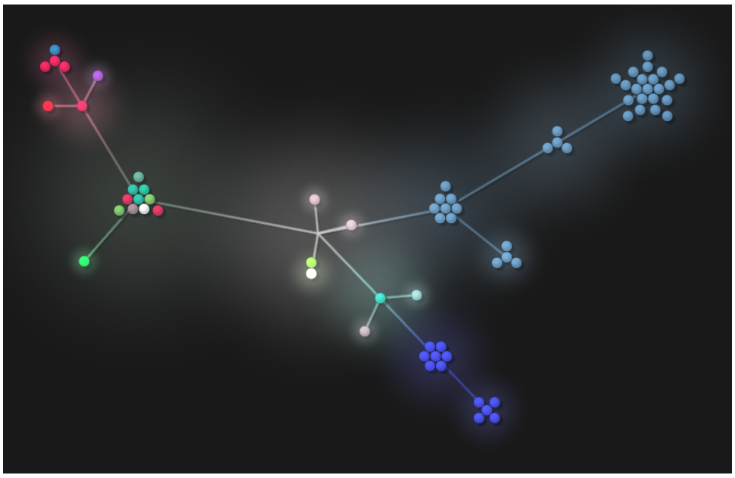

Figure 3: GitHub repository visualisation

Website Performance

Page	Load Time
Home	5.5ms
About	2.424ms
Viewer	4.32ms
Reports	20.78ms
Features	9.2ms
Contact	9.1ms

Figure 4: Website Performance