**GROUP REPORT**

**Software Engineering Group Presentation**

Georgy SEK22555793

Assia IBRAH22520673

Jonny EMI22529075

Michael MEY22555175

Piri RAT22528167

**Declaration**

We hereby certify that this report constitutes our own work, that where the language of others is used, quotation marks so indicate, and that appropriate credit is given where we have used the language, ideas, expressions, or writings of others. We 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.

We have implemented resources provided by Roehampton University in the labs and seminar notes. All credits go to the institution and the providers on Moodle.

Signed by the whole group

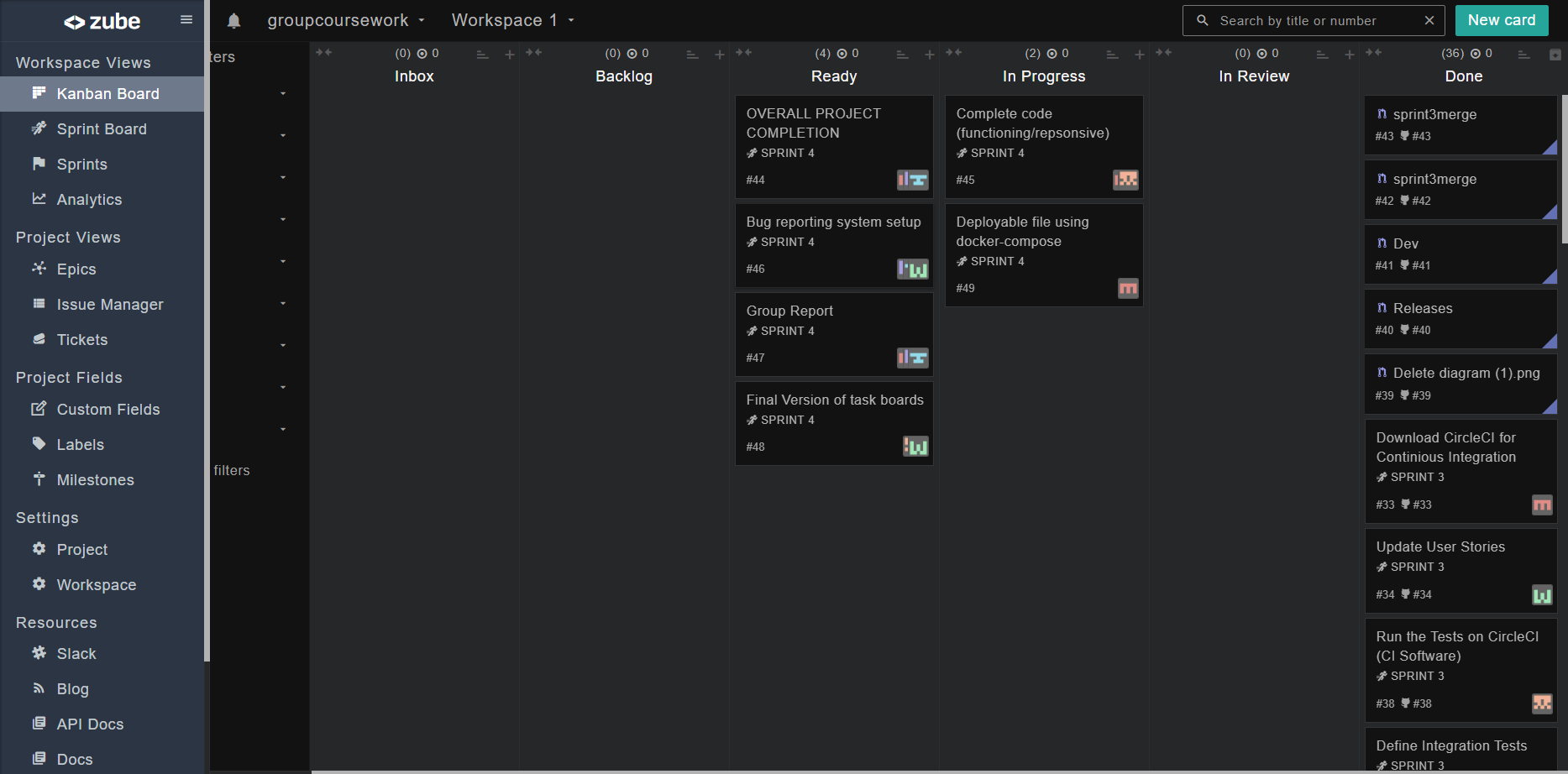
Date: 21/04/2024

**Summary**

In this assignment we have worked as a group to complete all Sprints from 1 to 4. We have used SCRUM to work together and complete this project in the most efficient and effective manner. We implemented team working tools and strategies to ensure everyone equally participated and completed their role to complete this project.

Kanban Boards and Sprint Boards

* This included the use of Kanban boards and Sprint boards to easily visualise the tasks we needed to complete within which time frames as well as review who was assigned each task. We found it easiest to use Zube.io which automatically integrated the issues into the boards as well as the assignees.



Code of Conduct

* We also had a Code of Conduct which stated ground rules and obligations to maintain a civil and effective workflow within the group. This was set before the project began to ensure every member complied throughout. Every member agreed to the regulations to ensure equality and fairness.

Project Backlog

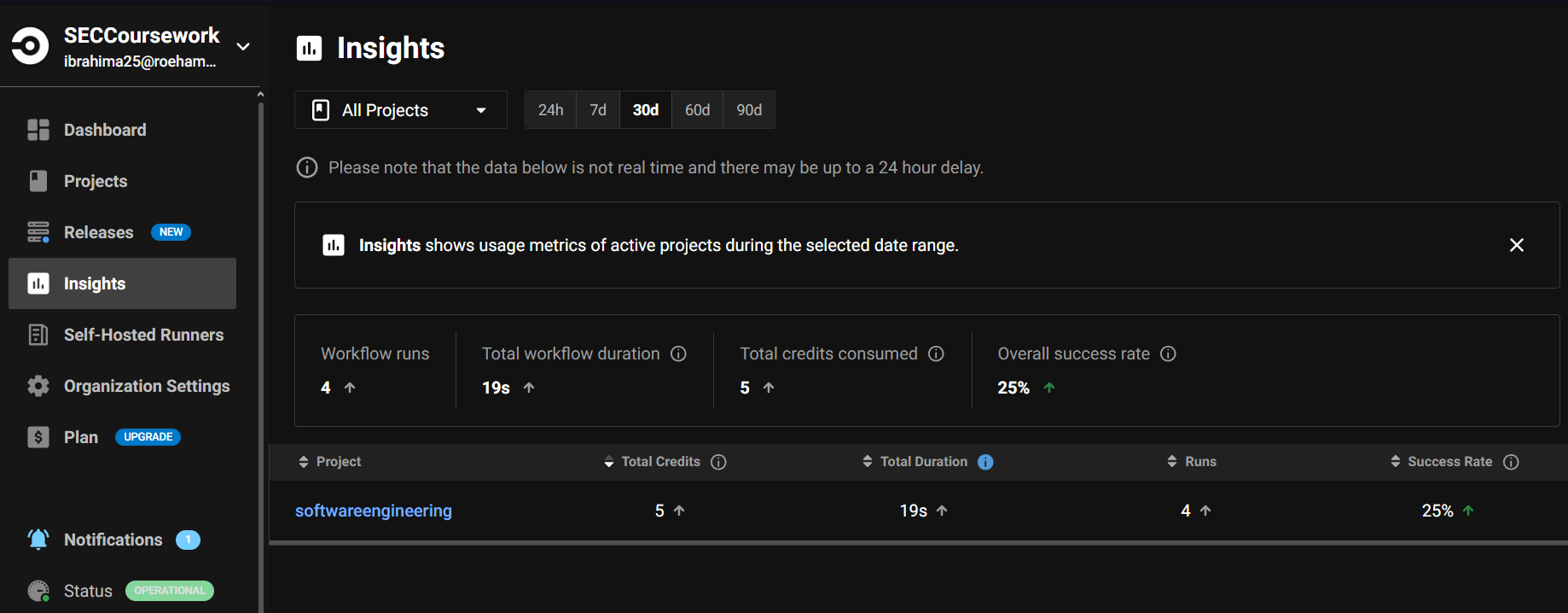
* The project backlog was a vague way to bullet point tasks we require to complete for each sprint. This also helped to overlook how much time would need to be spent on the different sprints.

**Aims and Objectives**

The overall aim of the project is to utilise the SQL file of population reports and develop a website to showcase the results in a table. The website is written in PUG rather than HTML and uses CSS to design the layout as well as Javascript for interaction.

Continuous Integration - CircleCI

* We have chosen CircleCI to be our platform for our continuous integration. One member has a login and controls the access of CircleCI and ensures tests are running to keep the code working effectively.



**Implementation results**

Having followed the design and development methodologies guidelines as discussed above, the group has produced the system with the proposed solution in the SQL database, PUG, CSS, and JavaScript. The site will now be in a position to present to the viewers the population reports successfully and in an accessible and interactive manner. CircleCI integration brought along continuous testing and continuous integration, hence reducing the live environment's number of bugs to the lowest possible level.

**Assessment and Challenges**

The project fulfils most of the set goals; although, there have been a few bumps along the road. For instance, fitting very different technologies together took a lot of debugging and optimization.

However, with continuous integration using CircleCI, it has amazing benefits compared to others. The result of the evaluation was that the solution turned out to be strong in different test conditions, but some features fail to produce the results in a case where

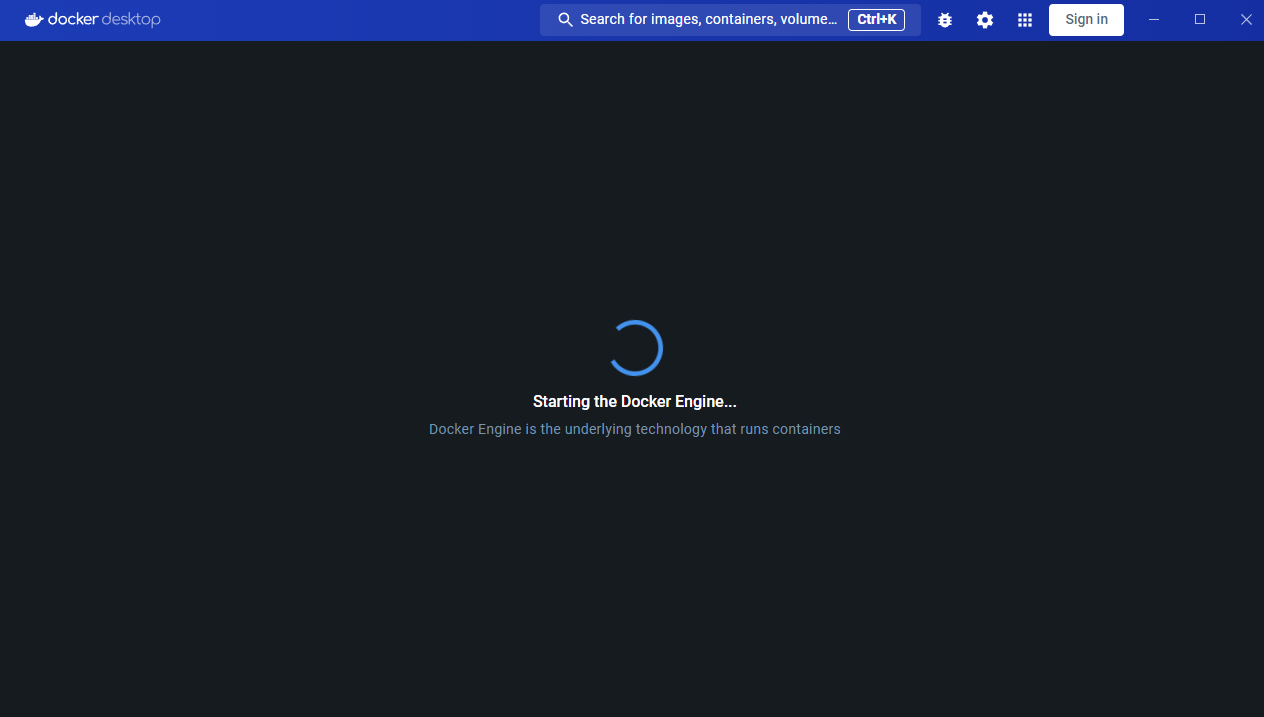
they were subjected to the stress tests, hence further rounds of the refinement. Related Work

The project borrowed much from the already existing tools to visualise the data on population and added a few things uniquely, for example, interactive charts and dynamic filtering. This comparison makes it clear that the project, from the user's point of view, is even more responsive and involved than the existing solutions.

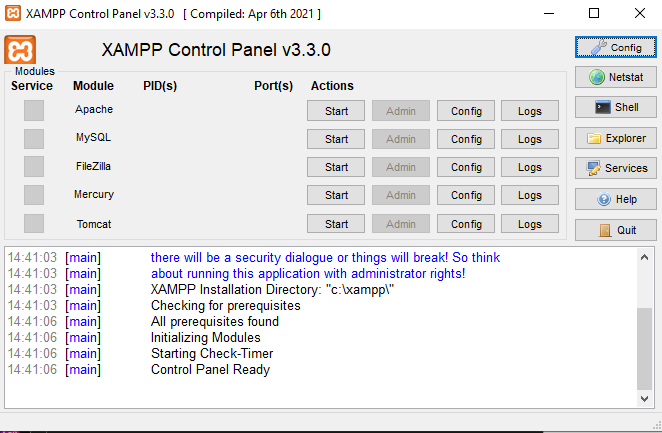
**Coding and Setup**

1. Firstly we had to install the necessary softwares to ensure the code was working.

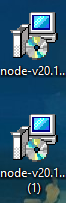
**Docker Desktop:** This is one way to run the website, however after a few weeks we decided to run the website using Node.js.

****

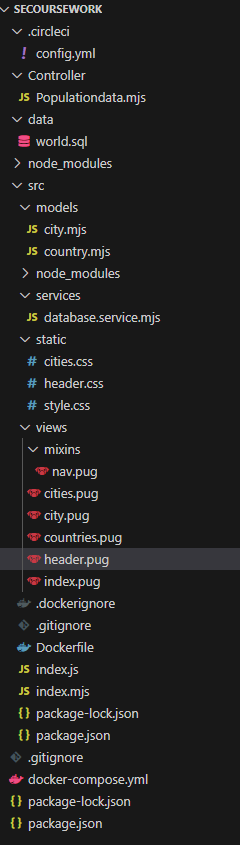
**XAMPP:** Here we installed XAMPP on our devices. This is on a Windows device and it allowed us to run our website using Node.js through the ‘npm run start’ command.



**Node.js:** We also had to ensure Node.js was installed as well to ensure the entire operation is working in the first place. This is the latest Node.js Windows package and allowed us to run the Node terminal on vscode as well as run the necessary commands.



1. Next we had to gather the code from the CRUD template and alter it to align it with our requirements to build our website.



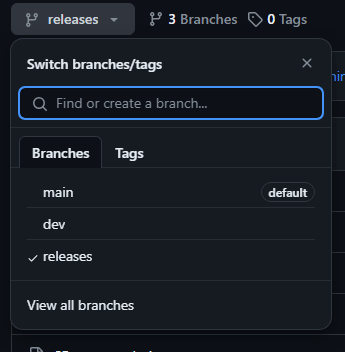
* MVC code is present. (Model, View, Controller)
* Data file is within folder
* Dockerfile code is present
* Docker Compose file to execute with docker-compose up
* Pug files used instead of HTML
* Config.yml present for continuous integration
* Additional files such as header.pug
* .gitignore and .dockerignore to avoid running invalid files

**GITHUB**

Our collaboration can be overlooked on the Github repository with all group members participating and keeping track of tasks on the issues page.

* Branches

There are 3 branches in our Github which each serve an important purpose in our project. The main branch represents the latest stable version of the project, where completed work is merged. The development branch used for ongoing development work, integrating features, fixes, and changes before they're ready for production. Finally the release branch includes all the finalised content which is being released to third party members.



**Conclusion and Reflection**

The project has realised the main goal: to create a dynamic and interactive platform for the visualisation of population data. The entire period of the project we have followed with the agile methodology of development and, hence, adaptive planning and improvement. Reflecting on the project, the team agrees that if more effective time management had been in place, then this would help in accelerating some of the phases, but more so on the code development.

Our journey through sprint 1 to 4 has involved a lot of time management as well as team assigning in order to ensure every task is completed on time and reviewed by the whole group during the group meets.