Sprint Report

Portfolio task 2/3/4

Unit code: COS40005

Unit Name: Computing Technology Project A

Submission date: 12/10/2025

| Student Name | Student Id | Statement of contribution to the report |
| --- | --- | --- |
| Connor Lack | 103992223 |  |
| Jen Mao | 103821194 | Lesson Learned |
| Jason Vo | 103993653 | Sprint Review Retrospective |
| Kavindu Bhanuka Weragoda | 104860525 | Sprint Progress |
| Md Hridoy Mia | 105077229 | Contributed to writing the Sprint Plan section of this report. |

[**Acknowledgment of Country 2**](#_heading=)

[**Contribution summary (This Sprint) 2**](#_heading=)

[**Contribution Details (This Sprint) 3**](#_heading=h.2ao966uwpe3r)

[Connor Lack 3](#_heading=h.yluqtbiepw3i)

[Document Parsing 4](#_heading=h.ri3v5fs32p40)

[Jason Vo 6](#_heading=h.v11x3dcmz89)

[Jen Mao](#_heading=h.rq43li7f8cgs) [7](#_heading=h.50ue8lkpnpy9)

[Kavindu Bhanuka Weragoda 8](#_heading=h.fc4hj4yqn562)

[Md Hridoy Mia 9](#_heading=h.ilmeopux48uk)

[1. Sprint plan 10](#_heading=h.6vvbf558u6x2)

[**2. Sprint progress 12**](#_heading=h.ca701x7l8nb)

[Technical Progress 12](#_heading=h.yhgzbtzd3hod)

[● Backend development 12](#_heading=h.2zprnx59hjvi)

[● Rule-Based Scoring System: 13](#_heading=h.us3st7zdarme)

[● System Architecture Setup: 14](#_heading=h.77z55tjssn8h)

[FrontEnd & UI Design 14](#_heading=h.7jfwimbd5oxy)

[Team collaboration 15](#_heading=h.xrs8wrtw2mna)

[Summary 16](#_heading=h.98hmj3j6e53n)

[**3. Sprint Demonstration 17**](#_heading=h.2agrsfbiqy0w)

[**4. Sprint review (Critical review of the product) 19**](#_heading=h.4dnaxaacghli)

[**5. Retrospect (Critical review of the process) 20**](#_heading=h.5ikp40u49z1j)

[**6. Lessons learned (Critical review of SPRINT one experience and future plan) 21**](#_heading=h.kj80fsg75agc)

## Acknowledgment of Country

Each team member identifies:

the [Traditional Owners](https://aiatsis.gov.au/explore/map-indigenous-australia) of the land they lived on while completing this work (if living in Australia).

We are honoured to recognise our connection to Wurundjeri Country, history, culture and spirituality through these locations, and strive to ensure that we operate in a manner that respects and honours the Elders and Ancestors of these lands.

We also respectfully acknowledge Swinburne’s Aboriginal and Torres Strait Islander staff, students, alumni, partners and visitors.

We also acknowledge and respect the Traditional Owners of lands across Australia, their Elders, Ancestors, cultures and heritage, and recognise the continuing sovereignties of all Aboriginal and Torres Strait Islander Nations.

## Contribution summary (This Sprint)

|  | Contribution | Initiatives | Communication | | | | | | Respect |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Team Member | Finished tasks in time or on time with an acceptable quality | Self-assigned tasks and proactively found solutions to problems | Attended all team meetings | Attended all supervisor meetings | | Attended all client meetings | Replied every time within an acceptable timeframe | Kept the team updated about the status | The student respected others and listened to different opinions |
| Connor Lack | Yes | Yes | Yes | | Yes | Yes | Yes | Yes | Yes |
| Jason Vo | yes | yes | yes | | yes | yes | yes | yes | yes |
| Jen Mao | Yes | Yes | Yes | | Yes | Yes | Yes | Yes | Yes |
| Kavindu Bhanuka Weragoda | Yes | Yes | Yes | | Yes | Yes | Yes | Yes | Yes |
| Md Hridoy Mia | No | No | No | | Yes | Yes | No | No | Yes |

## 

## Contribution Details (This Sprint)

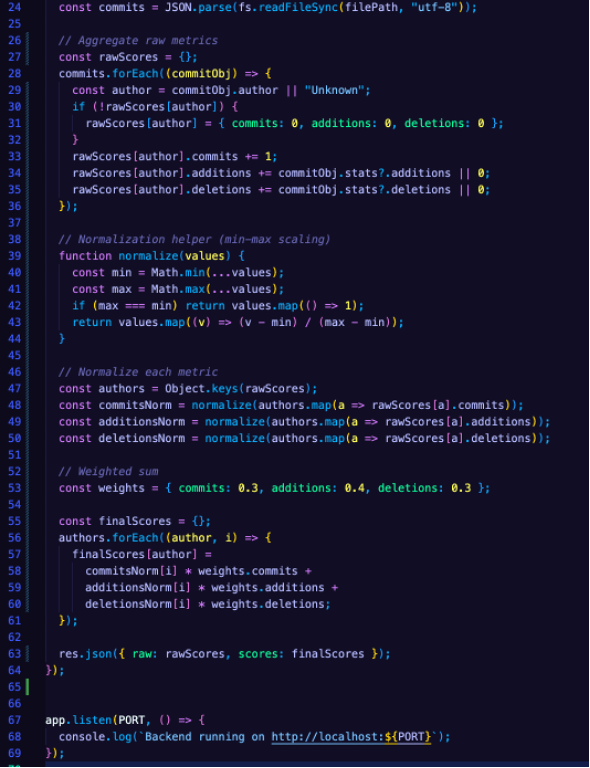
### Connor Lack

During Sprint 2 I contributed to the team through the creation of the [Node.js](http://node.js) and ReactJS project repo, detailing how to install the dependencies and use the backend environment and run the front-end. Express was used to create a server on port 5002.

**GitHub Parsing:**

I created a function fetchData() that fetches data from a specified GitHub repository using their API, then the data is converted into a JSON file from which it can be displayed on the front-end of the system. This is achieved by fetching the list of commits from the repo and then sending a request to the GitHub API to get a specified number of commits.

The data is then formatted into a JSON from which the frontend fetches the /api/scores using a HTTP GET request from React, Node receives the request and executes the route handler before sending a JSON payload over HTTP. Frontend then receives this JSON.

The contribution assessment was based on Jason’s rule-based design for the algorithm that is to be implemented - using normalization, each author has commits, total lines added, and total lines deleted summed up which provides raw numbers. A weighted sum is added which combines the normalized metrics into a single final score using the given weights. 



### Document Parsing

Due to unfamiliarity with the packages, a script was written in a separate directory to experiment with parsing different types of documentation data.

Worklog

I wrote a script that utilized the python package ‘python-dcos’ which allows for the extraction of information from DOCX files. This was utilized to extract all the key information from worklog documents with the hope that it can be displayed neatly on the front-end for the client without them needing to even open the document.

Essentially the code works by creating a dictionary for each week where all the variables are stored (TotalHours, TasksDone, Literature, etc.).

The program then looks for key headings (such as “Key tasks done”, “Key things learnt”, etc.) where the data is written. The block consumes and stores all the information until it reaches the next key heading.

Attendance

This script written using the python package ‘openpyxl’ aims to extract all the key information from the attendance sheet so the data can be easily displayed on the front-end.

The code works by reading each row of the Excel sheet and creating a dictionary for each week that stores the week number, date, absentees, and their reasons. It loops through each team member to check if they were present or absent, records absences, parses the “Name - Reason” text into a dictionary, and then at the end calculates each member’s overall attendance percentage across the semester by dividing the attended meetings by total meetings.



### 

### 

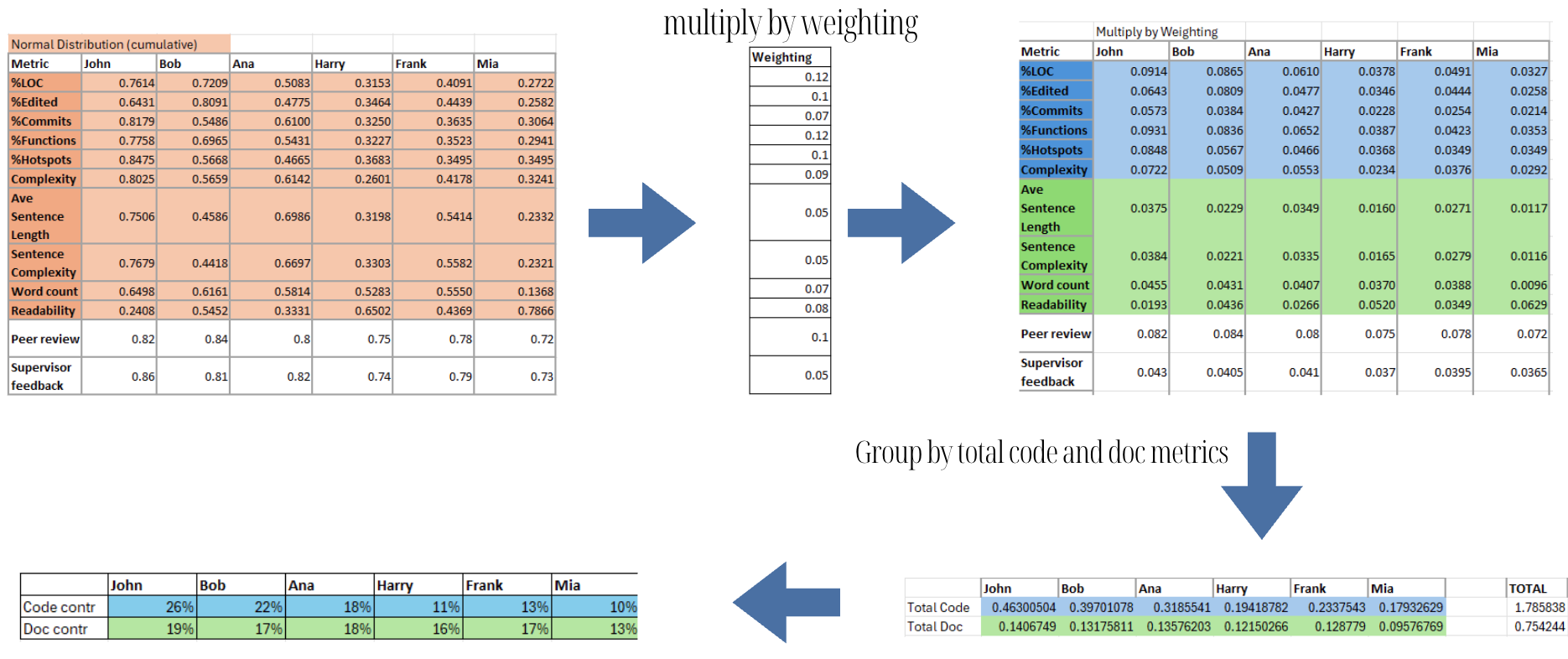
### 

### 



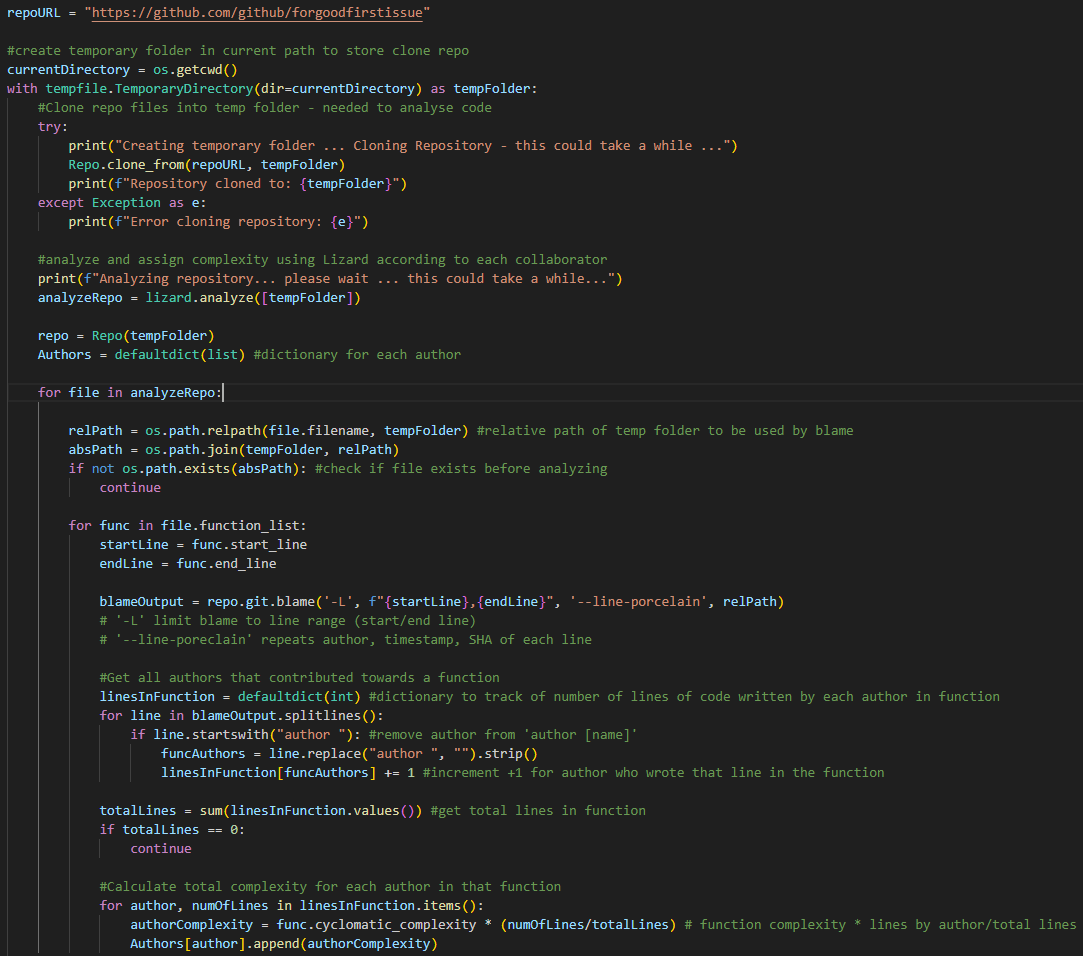
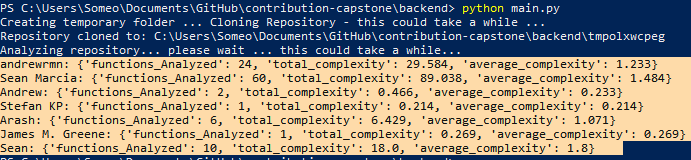
### Jason Vo

In this sprint, I was responsible for coming up with an algorithm approach and beginning the implementation for our backend in python, to gather the metrics necessary for the algorithm.



Our algorithm takes metric data from each group member in several categories. It then creates a normal distribution using that data so that we can compare all categories in a common scale. This is then used to calculate total contribution based on artificial weightings. The full excel sheet with experimentation and modifications can be found [here](https://docs.google.com/spreadsheets/d/1cW4HvujqKeDtwEoGuUheCxgzBZVv36yr/edit?usp=sharing&ouid=103552945780820271304&rtpof=true&sd=true).

I was also able to implement Lizard’s code complexity function in python to analyse a repository via a link to the repository. This would be helpful as we could implement it into our algorithm to use to calculate contribution.

### Jen Mao

### During this sprint, I was primarily responsible for designing and developing the full interactive prototype for our system using Figma. Building upon the feedback received from both the client and our supervisor after Sprint 1, I focused on improving the overall usability, accessibility, and visual consistency of the interface. One of the key recommendations from the supervisor was to implement both light mode and dark mode themes to enhance user comfort and accessibility under different lighting conditions. In response, I created two complete theme variations in Figma — ensuring that all UI components, color palettes, text contrasts, and icons adapted seamlessly between modes.In addition to the theme updates, I refined the structure and layout of several major screens, including the Log In, Dashboard, Rule Settings, Upload Data, and Export Report pages. I carefully organized content sections to improve readability and optimized spacing, alignment, and typography for a more professional and balanced look. The Dashboard and Rule Settings screens were particularly enhanced with new interactive components such as progress indicators, toggle switches, and drop-down menus that make the interface more dynamic and user-friendly.

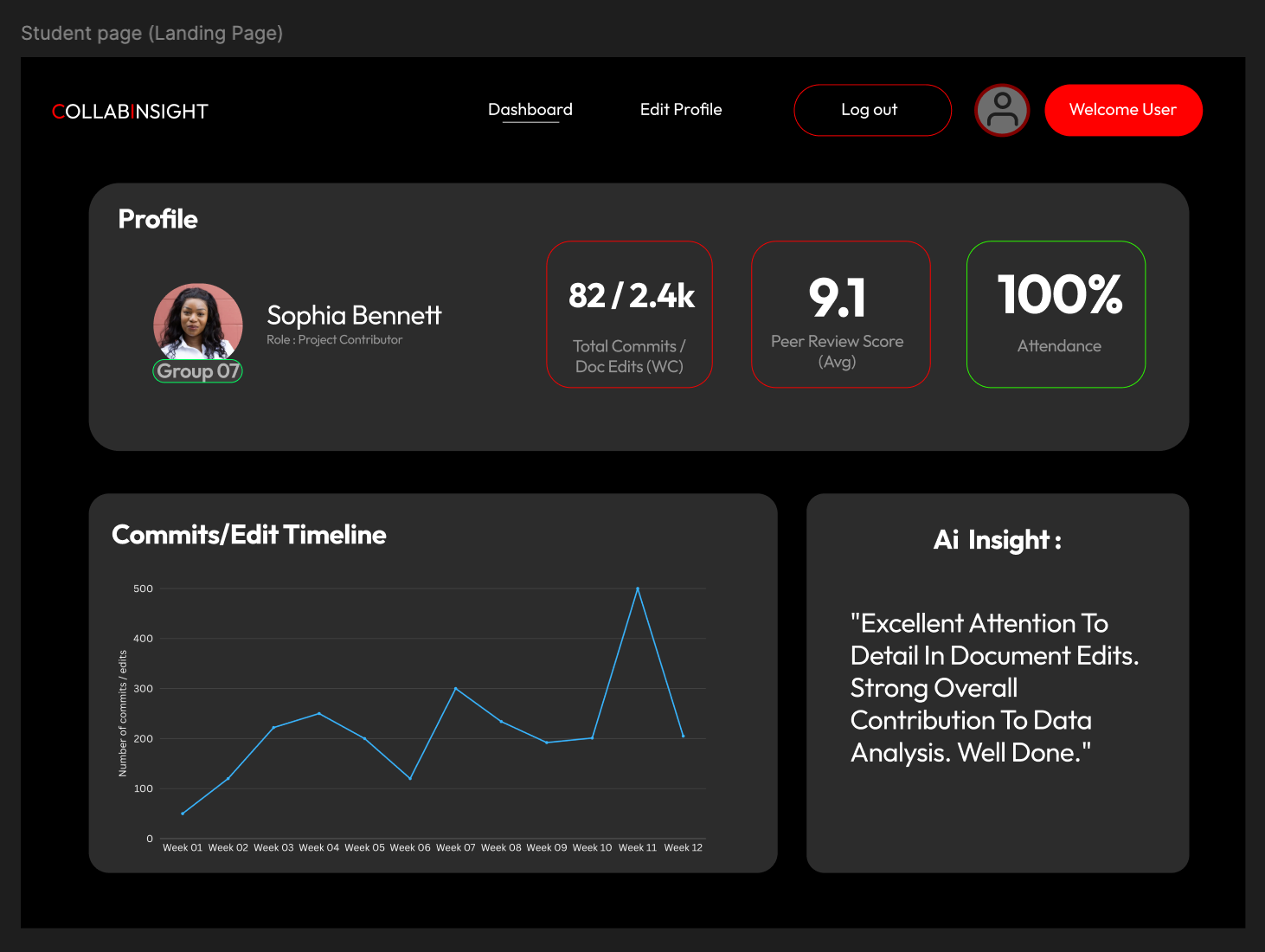
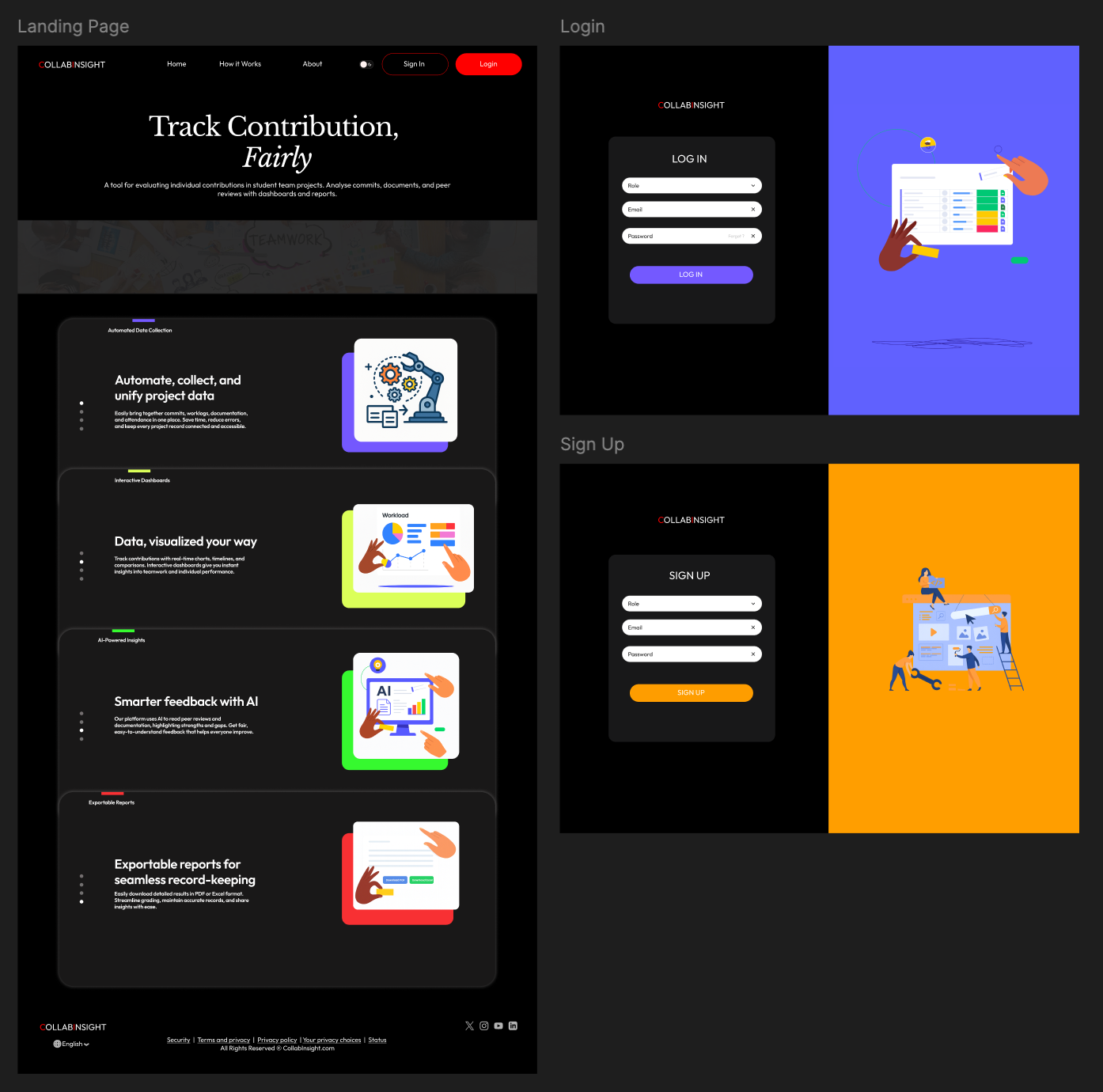
### Link to figma: <https://rb.gy/zwc278>

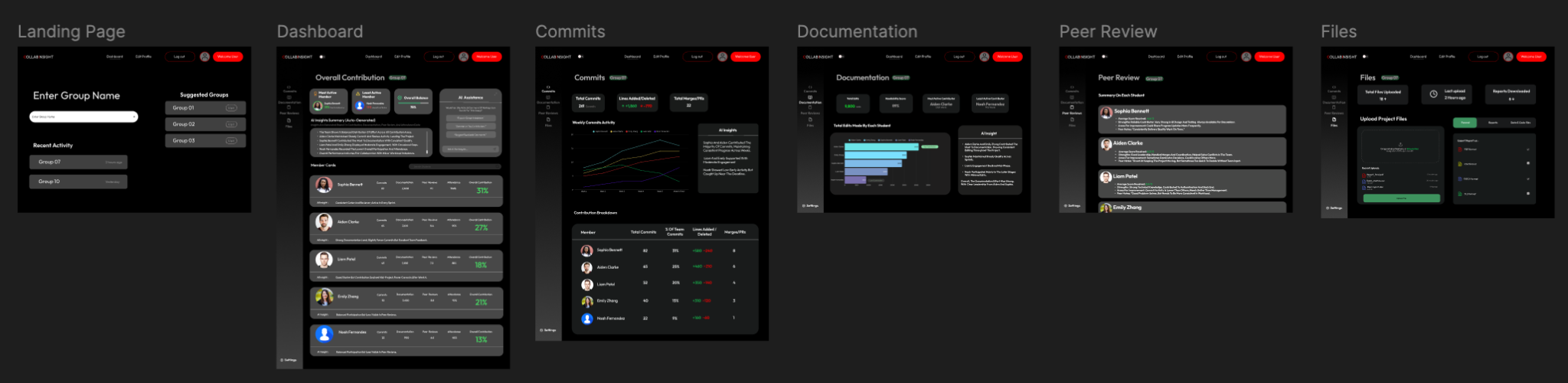
### 

### Kavindu Bhanuka Weragoda

During Sprint 2, my main task was to design the Figma prototype for our Rule-Based Contribution Evaluation System to present to the client. I focused on creating a clean and user-friendly interface that clearly displays the most important information about student contributions. My design emphasized visual clarity and simplicity, allowing users to quickly understand key data without being overwhelmed. I included important visual elements such as graphs, charts, and an AI Insight Summary section on each page, which helps explain the meaning behind the data. I also made sure that the student view page was well-structured so that each student could easily see their own contribution details. The overall goal of my work was to make the interface visually appealing, easy to navigate, and effective for supervisors and students to interpret results. This prototype was shown during the Sprint 2 presentation and received positive feedback from the client for its simple and clear layout.

Figma Link : [Figma Link](https://www.figma.com/design/mVJSj8tKOPOtZOYvvkLNYN/ColllabInsight--Project-A-?node-id=24-26&p=f&t=qVDFJSO1ugEDNbWu-0)





### Md Hridoy Mia

### 

### **1.** **Sprint plan**

The goal of Sprint 2 was to develop the foundational components of the Rule-Based Contribution Evaluation System by creating data processing pipelines, defining contribution scoring rules, and producing initial user interface concepts. This sprint aimed to deliver visible system progress while aligning with supervisor expectations regarding transparency, fairness, and explainability.

**Sprint Objectives**

* Establish core rule-based logic for contribution evaluation
* Build prototype parsers for project artefact extraction
* Set up initial system architecture for future integration
* Design user interface prototypes for contribution reporting
* Prepare system demonstration for Sprint 2 review

**Sprint Backlog**

| ID | Sprint Task | Assigned To | Status |
| --- | --- | --- | --- |
| SP2-01 | Define contribution rule set and weighting logic | Team Lead | Completed |
| SP2-02 | Develop Git repository commit parser | Backend Developer | Completed |
| SP2-03 | Develop Worklog parser | Backend Developer | Completed |
| SP2-04 | Develop Attendance parser | Backend Developer | Completed |
| SP2-05 | Build basic system architecture | Full Stack Developer | Completed |
| SP2-06 | Design Contribution Dashboard UI (Figma) | UI/UX Designer | Completed |
| SP2-07 | Design Student Contribution Profile UI (Figma) | UI/UX Designer | Completed |
| SP2-08 | Prepare demonstration workflow and Sprint 2 presentation | Whole Team | Completed |

**Sprint Deliverables**

* Rule-based scoring documentation
* Parser outputs for Git, Worklog, and Attendance data
* System architecture and data flow diagram
* Figma interface prototypes
* Sprint 2 presentation slides
* Updated backlog and Sprint 3 roadmap

**Sprint Scope** The team focused on artefact processing and evaluation logic to demonstrate measurable system progress. AI/ML automation was deliberately excluded in this sprint to keep the phase explainable and aligned with client expectations.

**Definition of Done (DoD)** A task was considered complete when:

* Functionality met acceptance criteria
* Output was validated by the team
* Work was documented and presented
* Code or designs were stored in the shared repository

**Dependencies**

* Parser development depended on sample datasets
* Dashboard UI depended on expected output format from parsers
* System flow design required agreed rule logic

## 

## **2.** **Sprint progress**

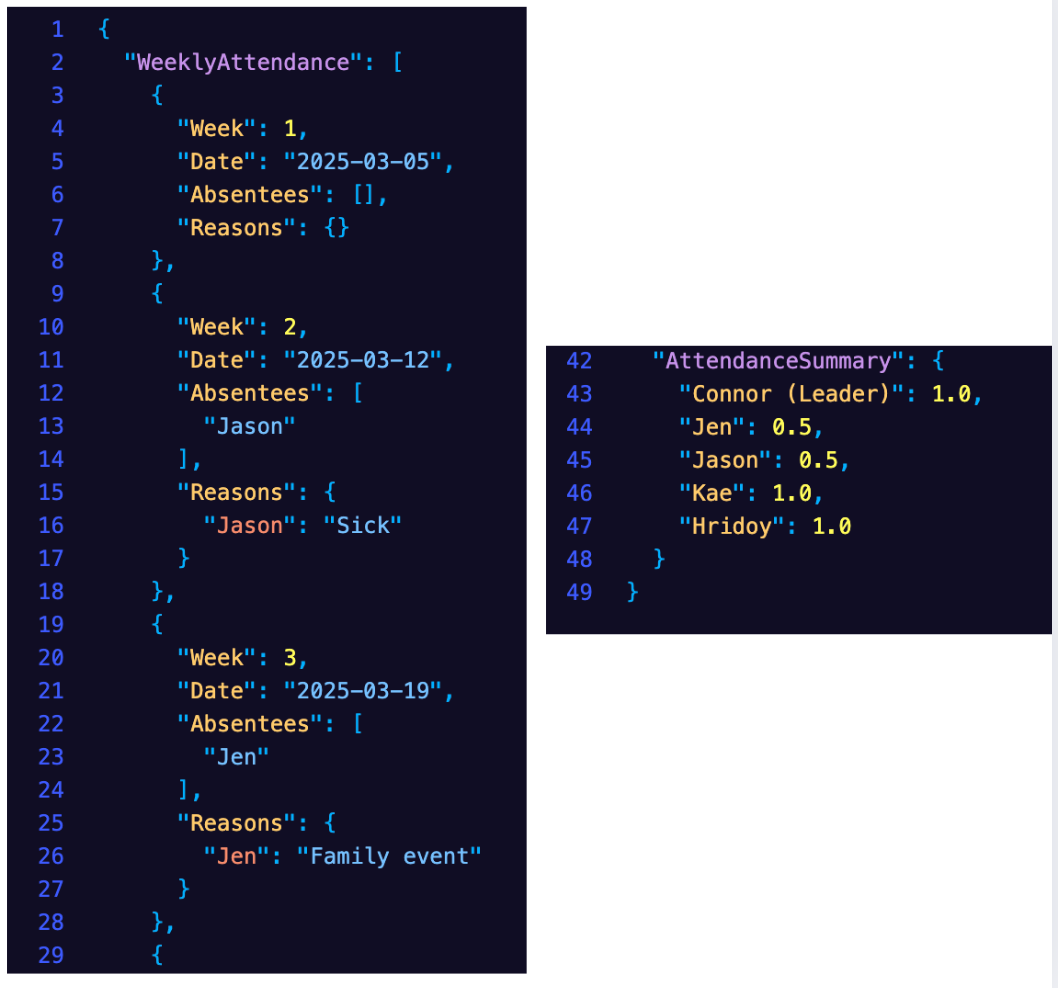
### Technical Progress

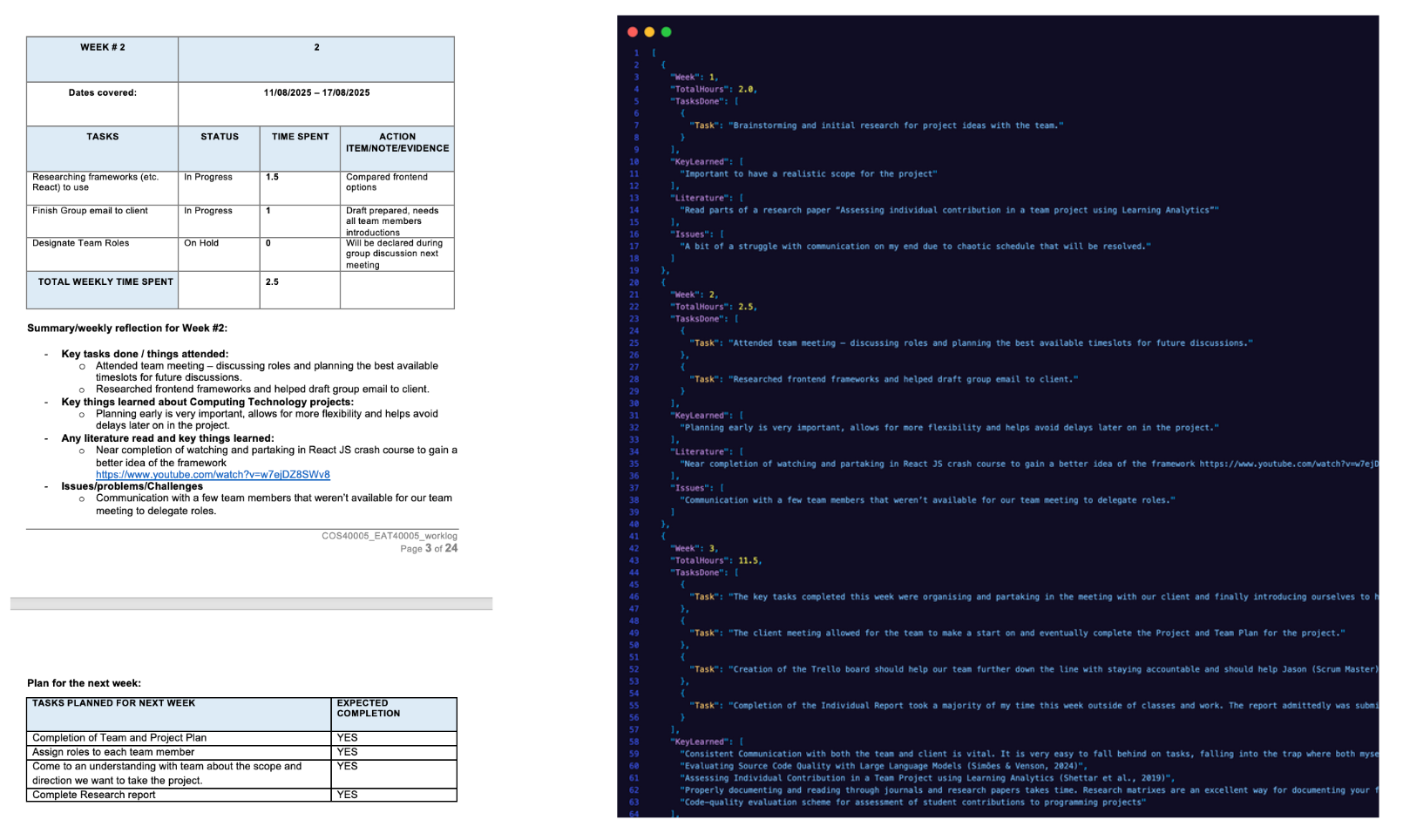
#### Backend development

The backend team created 3 main data phrases

* **GitHub Parser:** Collects commit data from GitHub using its API and saves it in a structured JSON format.
* **Worklog Parser:** Uses the Python package python-docx to read student worklog documents and extract weekly hours, key tasks, and learning points.
* **Attendance Parser**: Uses openpyx1 to read Excel attendance sheets, calculate attendance rates, and show absences.

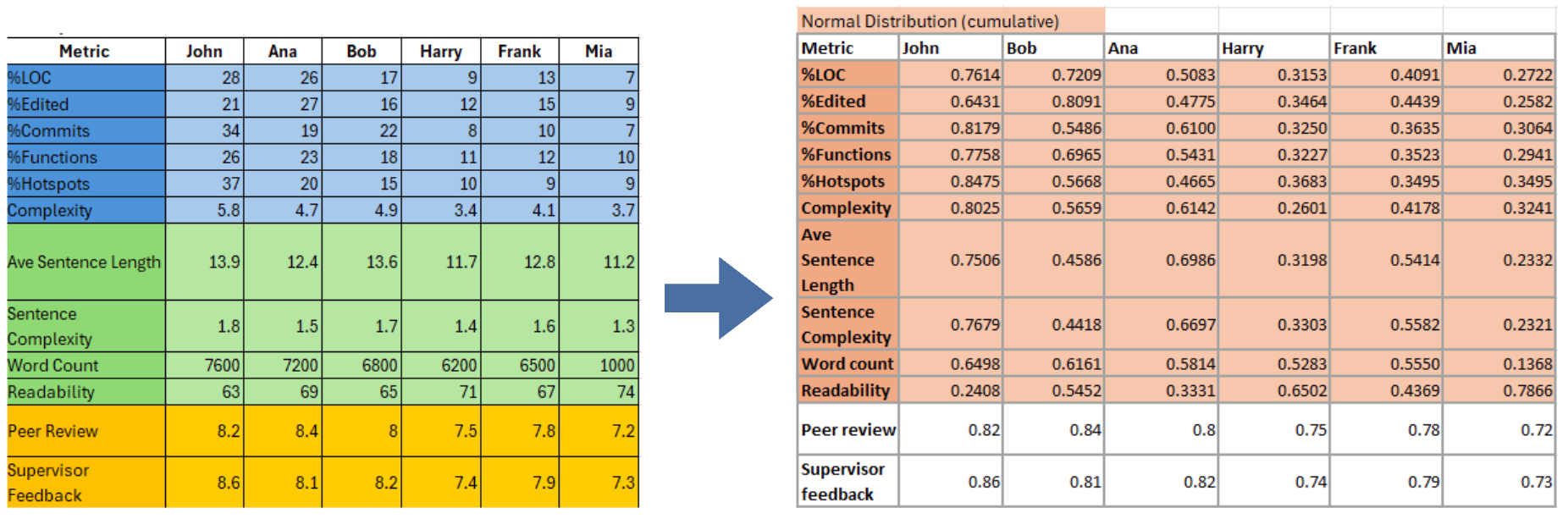
These parsers make it easier to automatically collect and process data from different project artefacts.

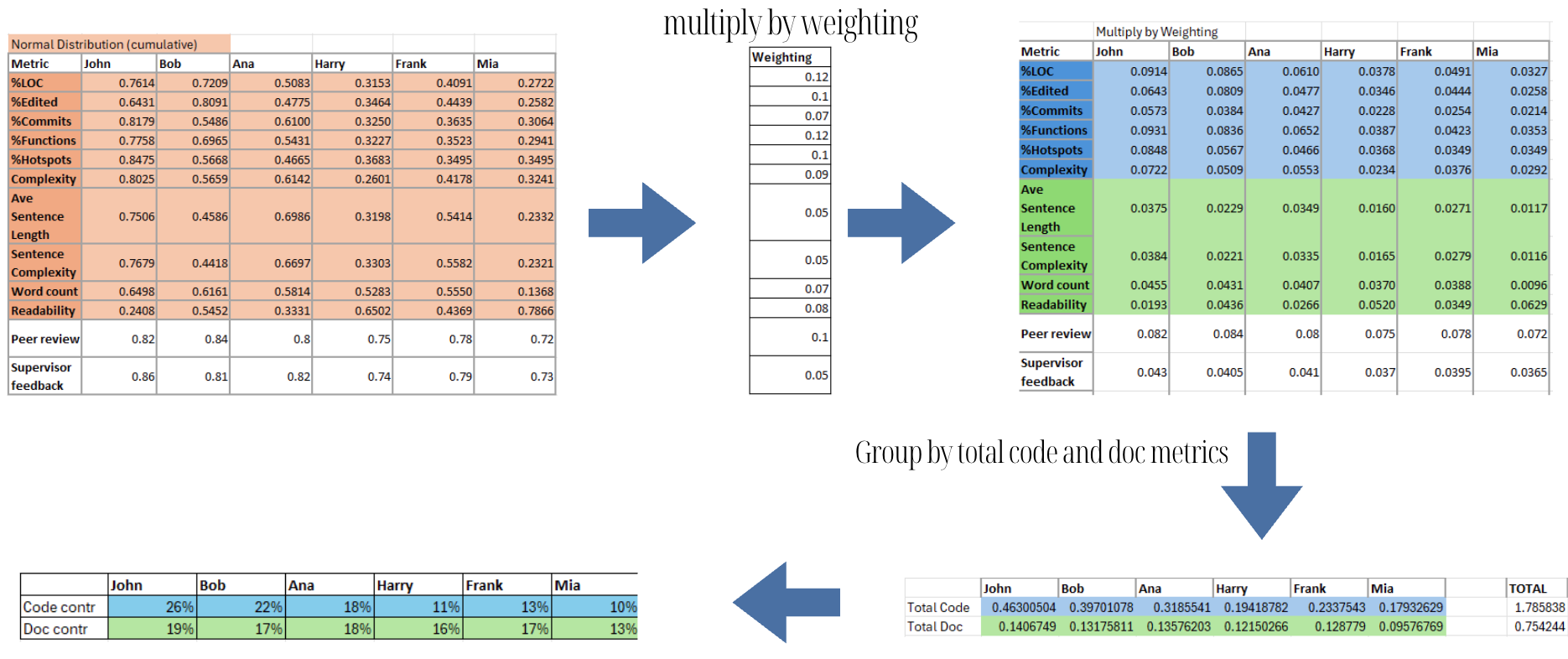




#### Rule-Based Scoring System:

The team built a set of clear rules to calculate each member's contribution. The team designed and trialed a rule-based scoring algorithm to calculate contribution percentages across multiple artefacts (code, documentation, attendance). The logic applies normal distribution normalization to balance data variability and ensure fairness when evaluating groups of different sizes. Weightings were assigned to each artefact type, following the principle of triangulation - cross-verifying contribution evidence from multiple sources.





#### System Architecture Setup:

The basic Node.js and React system structure was set up. This follows the same style as modern MERN-based platforms (like ScriptSaga) and allows us to easily add new features later, such as peer reviews or supervisor feedback.

### 

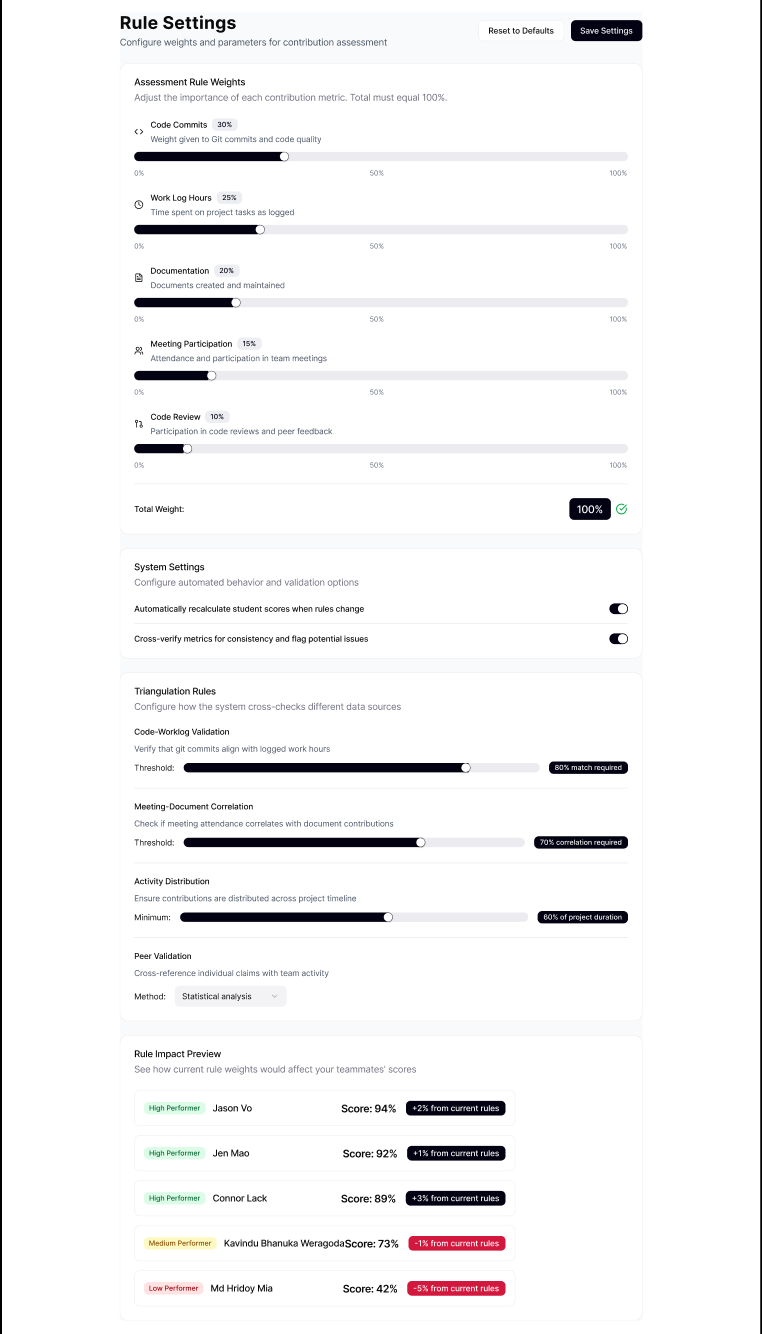
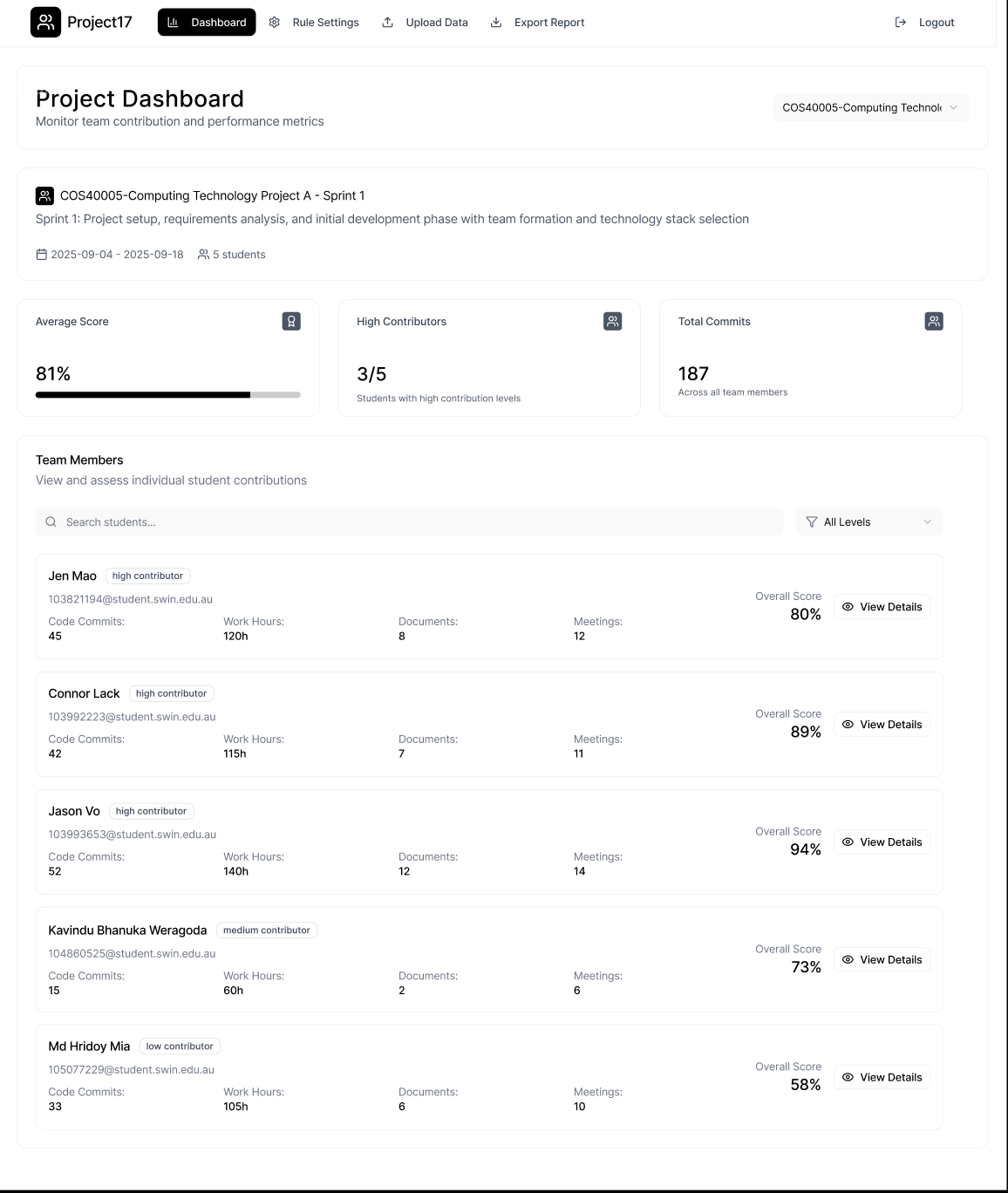
### FrontEnd & UI Design

We managed to create 2 figma prototype designs

**Prototype 1 :**

This version takes a more simple and traditional dashboard approach.

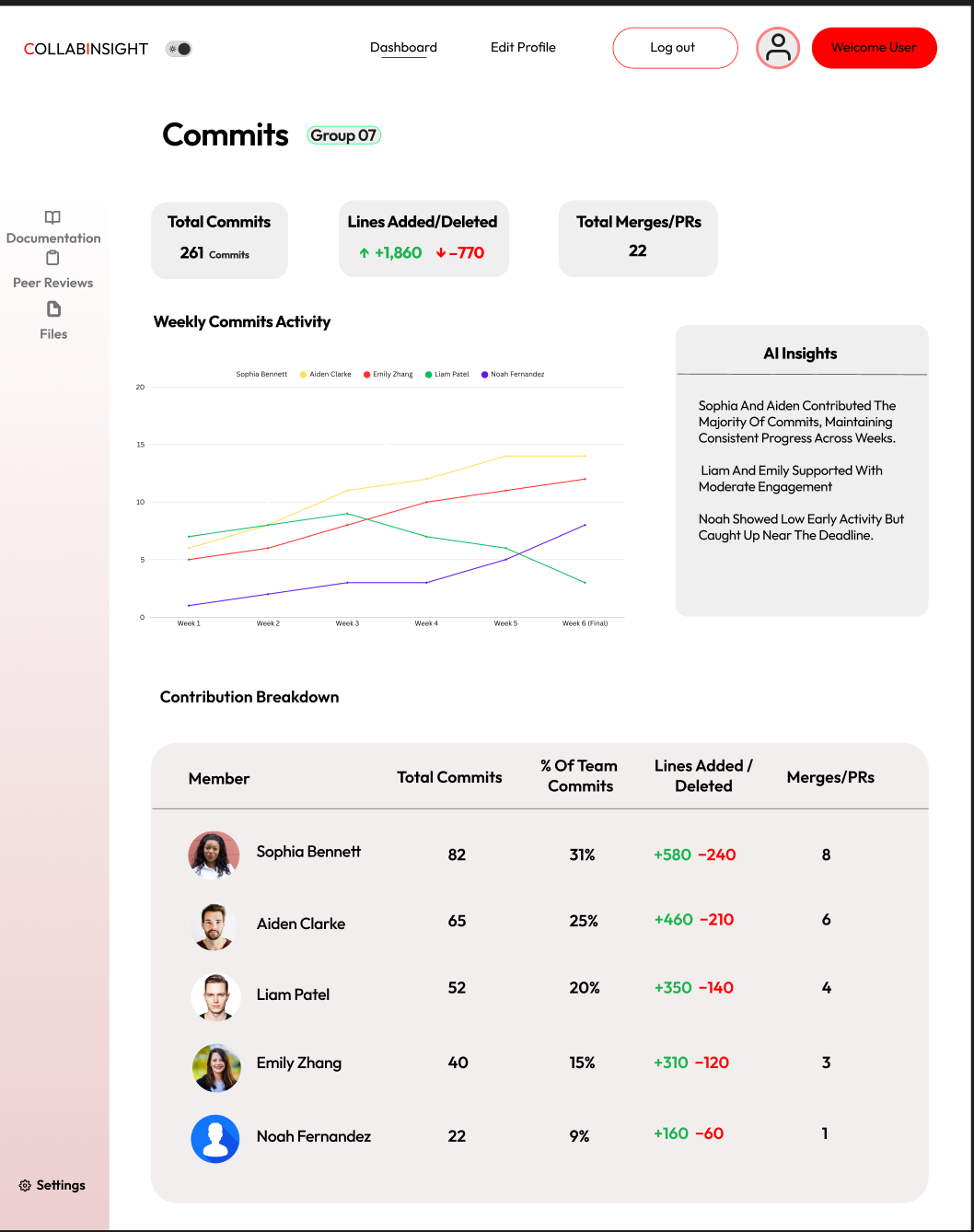
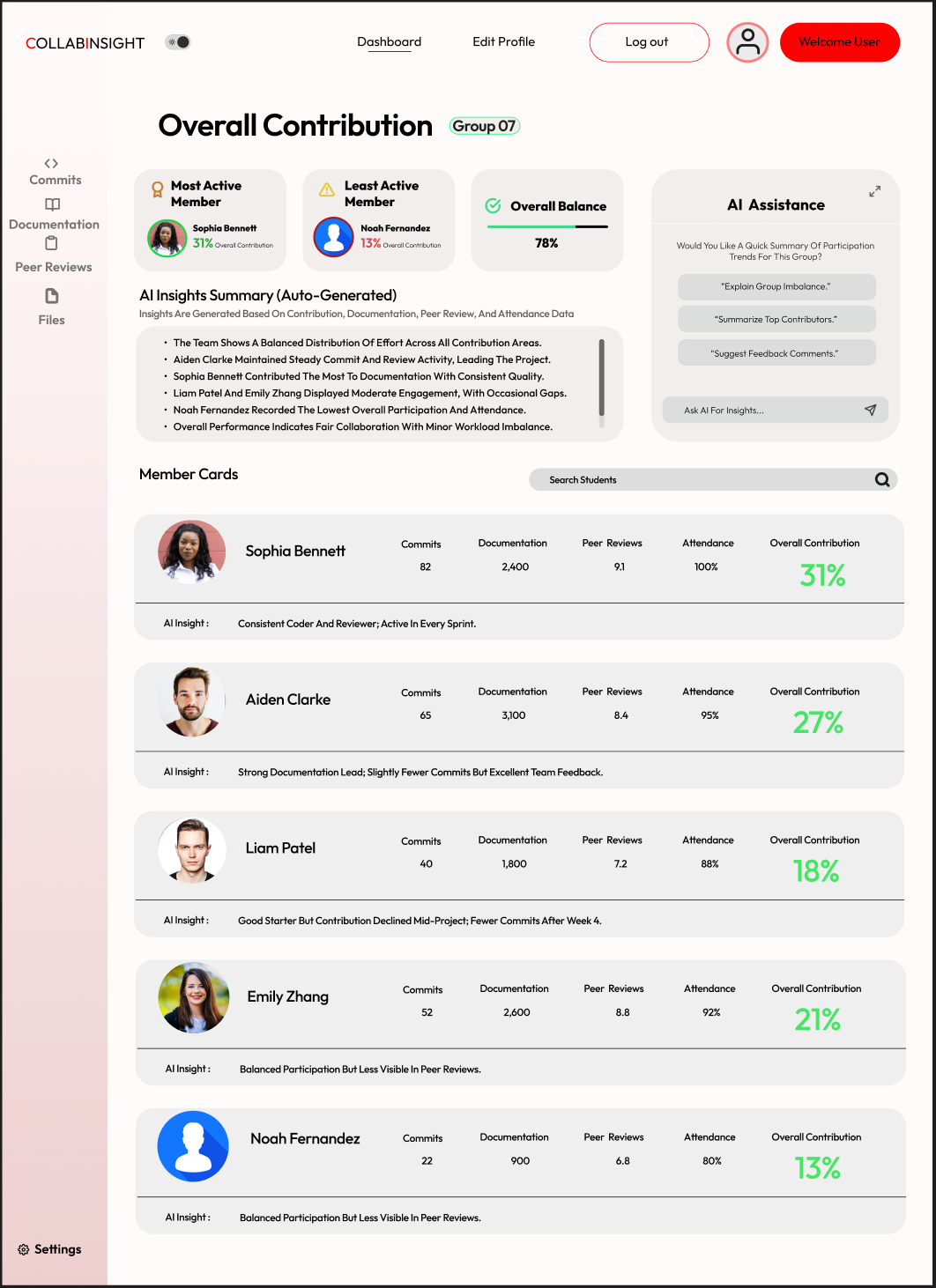
* It focuses on showing detailed data like graphs, tables, and multiple sections for different contribution categories.
* The layout gives supervisors full visibility of all metrics, making it suitable for data-heavy analysis.



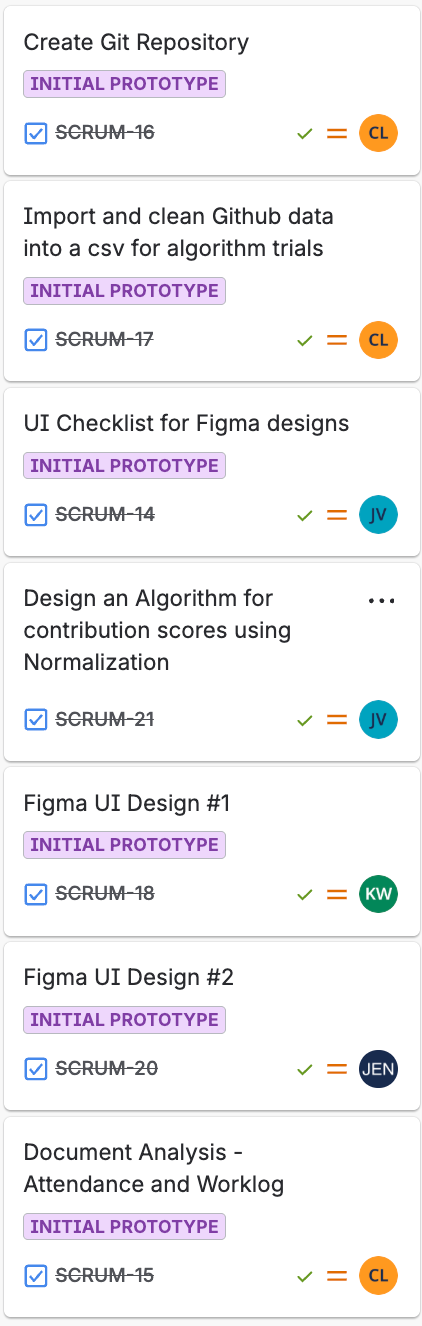
**Prototype 2 :**

This design takes a clean and minimal approach.

* It focuses on clarity and simplicity - showing only the most important visuals and statistics at a glance.
* Each page includes an Al Insight Summary, which automatically explains what the data means (for example, summarizing who contributed most or least).



### Team collaboration

* **Discord:** Main platform for daily communication, sharing updates, and discussing progress.
* **Google Sheets (via google drive) :** Used to update and share our documents
* **Jira :** Managed sprint tasks, assigned responsibilities, and tracked completion.
* **GitHub:** Used to share, review, and merge code among team members.  
  

### Summary

During Sprint 2, the team focused on building the foundation of the Rule-Based Contribution Evaluation System, making strong progress in both backend development and front-end design. The sprint included setting up the Git repository, developing scripts to import and clean GitHub data, and analysing artefacts such as attendance records and worklogs. The backend team implemented automated parsers that extract and organize key data, allowing the system to evaluate contributions from multiple sources. A normalization-based algorithm was also designed to calculate contribution scores fairly and ensure balanced evaluation among team members.

On the design side, two Figma interface prototypes were developed - one focusing on detailed data visualizations and the other featuring a clean, minimal layout supported by Al-generated summaries for quick interpretation. A Ul checklist was also completed to maintain design consistency and clarity. Together, these developments created a strong and functional base for the project, transitioning it from the planning phase to an early working prototype ready for integration and testing in Sprint 3.

## 

## **3.** **Sprint Demonstration**

In Sprint 2, the team met with the client on Thursday 9th October. A demonstration was conducted which involved a detailed presentation which included a response directly to the client feedback from Sprint 1 and the showcase of early progress in the system’s core components - predominately from the data processing and visualization section.

Feedback and Adaptations

* The client stated that in our first sprint that we needed to finalize an approach for the analysis of individual student contribution. Upon further investigation, we came to the joint decision that machine learning wouldn’t be possible due to data privacy and not having a large enough dataset to train the model.
* The first sprint presentation was also too individualised and technical. This resulted in the client being overwhelmed with information and the creation of an incohesive presentation which felt like 5 small separate presentations (in fact we didn’t even get to present all our information!). However, we improved upon this, providing a helicopter view of the work we had contributed during Sprint 2 instead of a deep dive. We also rehearsed multiple times and dedicated roughly 10-12 minutes of wiggle room for the client to ask and answer questions during the presentation.

System Architecture and Data Flow

* The system architecture and flow was briefly touched upon again to refresh the clients memory on how the data flows through the system. It was also detailed which parts of the design we had worked on during Sprint 2 - this being the rule-based logic, data parsing for GitHub and Documentation, as well as development of the front-end design.

Rule-Based Logic Approach

* The rule based approach presented to the client details how contribution is assessed using data normalization. Normal distribution is applied to smooth out the data variation and cyclomatic complexity was used to estimate the complexity of the code contributed. Peer and supervisor feedback will be implemented as project artifacts.

Data Collection Scripts

* Three different forms of data collection were demonstrated to the client.
  + Git Parser: Fetches commits using the GitHub API and stores the key commit data in a JSON format.
  + Document Parser: Uses python-docx to extract the structured data from weekly reports.
  + Attendance Parser: Uses openpyxl to read the excel spreadsheet to fetch the attendance logs. This allows for calculation of individual weekly presence.

Interface Prototypes

* Two different prototypes were showcased to the client. Both focused on usability, presenting a clean and intuitive design of how the student contribution data could be displayed. The interface will eventually display metrics from commits, documents, attendance, peer feedback, and potentially other project artifacts too.

Sprint 3 Roadmap

* The upcoming roadmap was presented with the main goals of:
  + Implementing the rule-based contribution metric system
  + Enabling manual file uploads in a localhosted environment
  + Integrating document and peer review data
  + Developing the contribution dashboard visualization

## 

## **4.** **Sprint review (Critical review of the product)**

During the sprint, the team was able to produce a demonstration for our client with a presentation which showcased what we have been working on. Our presentation included our algorithm approach, documentation and parsing approach along with several front-end designs for our client to choose from. In terms of our sprint plan for our product, we had planned to show our client a detailed look into what our algorithm could look like as well as how we can obtain code/documentation metric data and data parsing. We also planned on starting the development of our front end as well. We were able to achieve these goals which allows us to progress to the next stage for Sprint 3, a simple prototype.

During the sprint we faced challenges that limited our quality for our overall product. The delay of the completion of our front-end figma designs proved to be a big concern as we were not able to get early client feedback from our designs, limiting our ability to refine them. This resulted in a lot of idle waiting time and a lower standard of work, as we didn’t get feedback from our client until the demonstration. Our team also agreed on our algorithm’s general structure although it had to be modified a few times to better meet requirements. The previous iterations of the algorithm skewed metric data by a large amount despite being only slightly larger or smaller.

The client provided us with positive feedback that suggested we were on the right path, and our presentation was significantly better than our previous one. Our client was happy with our designs and provided feedback about our algorithm approach suggesting more research could be done on methods to gather metrics such as code complexity. They also mentioned that our presentation felt more like a team effort rather than individual work as well as that we were able to complete our presentation at a reasonable time that gave enough time for feedback from the client. Overall, the client was happy with our progress and was excited to see what we will present for our next sprint.

## 

## **5.** **Retrospect (Critical review of the process)**

Throughout the sprint, our team used Jira and Discord to communicate tasks and discussions on the progression of our project. This allowed us to communicate any updates, assign responsibilities and discuss any issues that may appear.

In this sprint we ran into a few challenges such as the completion of one of our front-end designs which delayed our process significantly, as we were expecting to finish our designs within the first week of our sprint. This would have allowed us to get our clients’ feedback on the designs so we could refine them. Not only was the front-end Figma design delayed for weeks, but the initial design also submitted was a blatant copy and paste of an online template that was not relevant to the specific needs of our project. Not only this, upon specific request to modify the design from our unit supervisor, the end result was not only delayed again but the output was a sloppy design change with elements copied and pasted from other team members' designs. This bottleneck resulted in our group having one less design to show our client. If team members are unsure about the project description or how to implement/use tools or design, they are encouraged to ask questions, which was said several times beforehand. It is also expected, as outlined in our team’s code of conduct, to complete tasks on time otherwise communication is needed to inform the team of the delay. As this wasn’t followed, they were left out of the presentation as there was nothing left for them to present.

In future, we plan to combat these delays and low quality of work by reinforcing accountability by having less tolerance for missed deadlines and having mid-progress reports on tasks. The supervisor will also be let known if these conditions are not followed and communication of delays or challenges is limited. Despite these delays, we still managed to provide a satisfactory presentation to the client and completed our key tasks.

## 

## **6.** **Lessons learned (Critical review of SPRINT one experience and future plan)**

**Key Lesson Learned:**

1. **Time Management:**

Each team member was responsible for creating a prototype; however, one was delayed and did not fully meet the expected standard. Because of this, the team was only able to present two prototypes during the Sprint 2 demonstration. This experience reinforced the importance of managing time effectively, clarifying task requirements early, and reviewing progress regularly to avoid last-minute issues in future sprints. Moving forward, the team will focus on better planning and checking in more frequently to ensure all tasks are completed on schedule and meet expectations.

1. **Structured Presentation and Collaboration:**

During Sprint 1, the team’s demonstration lacked structure and flow, which made it difficult for the client to follow the project’s direction. Learning from that experience, the Sprint 2 presentation was planned and delivered with a clear structure—each member presented their part in a connected and logical order. This approach made the demonstration more engaging and cohesive, helping the client clearly understand our project goals. The client gave positive feedback on this improvement and appreciated how well the team communicated and worked together during the presentation.

1. **Team Plan and Communication:**

Team discussions and collaboration improved greatly in this sprint. The group consistently used Discord for daily communication and Jira for tracking progress and assigning tasks. Members were more open in discussing challenges, sharing updates, and offering support to one another. This stronger communication and teamwork created a more positive working environment and helped the team stay aligned throughout the sprint. As a result, collaboration became more efficient, and the overall workflow improved significantly.

**Recommendation and Future Plan:**

1. **Plan Ahead :**

Learning from our previous sprint, we realised how important it is to plan and structure everything properly before starting new tasks. In the next sprint, we should try to plan ahead more effectively and make sure everything we do aligns with the client’s feedback and expectations. Having a clearer structure and timeline will help us stay organised, reduce confusion, and avoid last-minute stress. This also gives us more time to make adjustments based on feedback and deliver stronger results as a team.

1. **Understand and Adapt Client’s Needs:**

One of the main lessons from Sprint 2 was understanding what the client really wants and adjusting our work to match that. We need to focus on shaping our designs, logic, and presentations according to the client’s feedback so that everything we build aligns with their vision for the project. Making changes based on their suggestions not only improves the quality of our work but also shows that we value their input and are moving in the right direction. Keeping an open line of communication with the client will help us make sure we’re always meeting their expectations.

1. **Implement Mid Sprint Progress Check:**

To prevent future delays and improve accountability, we plan to have mid-sprint progress checks where each member gives a quick update on their assigned tasks during a Teams meeting. This check-in will happen around the middle of the sprint to make sure everyone is on track and tasks are being completed as expected. It also gives us a chance to discuss any challenges early and get help before they turn into bigger issues. This will help us maintain a steady workflow, reduce last-minute pressure before demonstrations, and make sure all our deliverables meet the project’s standards on time.

1. **Prepare for the next stage- Full Prototype Development**

Our next stage will be turning the current prototype into a working system, so it’s really important that we stay on track with everything—from back end to front end. Each member needs to manage their time well and complete their tasks before the due date so integration runs smoothly. We’ll also start testing and reviewing code more regularly to make sure everything works as planned. Staying consistent and communicating regularly will help us reach our goal of having a complete, functional prototype ready for the next demonstration.