**Planning Document**

EduArchive

**Team Data Access Geeks**

**Project Overview**

Our team has the goal of developing a student-focused database containing articles spanning three main subject categories: Arts, Mathematics, and Technology. Following the creation of this database, we will design a web application that allows users to access and engage with these articles.

**Iteration 1: Documentation (21/08/23 - 27/08/23)**

“*Analyse the data and create a Solution Design document outlining the architecture and database model. Also, create appropriate User Access and UX Design documents to implement the database using a web application. For each part of the Solution Design document, provide reasons as to why you selected a particular technology or design.*”

* **Plan:** Set up the team’s collaboration and development tools, and work on the project’s various documents.
* **Objectives/Goals:** GitHub organisation setup, Team Contract document, System Design document, Requirements document, Planning document (updated each iteration)
* **Decisions:**

The team will use MS Teams as the main communication tool (online meetings, daily stand-up updates, file sharing/storage). MS Teams Tasks by Planner and To Do will be used for the team’s Kanban board. MS Teams was selected because all team members have experience using it and allows much of the team’s collaboration/organisation to be handled within a single application.

For the web application’s implementation, a GitHub organisation and team will be created, along with the GitHub repository for the project. GitHub offers an extensive range of features for teams to collaborate on a codebase including, version control, issue management, discussion forums and project management.

* **Kanban screenshot:**
* **Review of iteration:**

All documentation is being worked on. The Team Contract document and Requirements document have many sections completed and these documents should be finalised soon. The System Design document has been started, but some aspects of the application’s system design have yet to be determined or finalised. Planning document will be added-to throughout the various iterations of the project.

Data Access Geeks was selected as the team’s name, and a Data Access Geeks team created in MS Teams. A Kanban board for managing the project’s tasks was set up in MS Teams.

An organisation, repository and team will be created in GitHub shortly, and team members will be sent an invitation to join the Data Access Geeks GitHub team.

* **Client feedback:** (Maki) I gave Bilal a very brief update. He seemed satisfied with the team’s progress to-date. He asked if the team had selected a DBMS yet; I told him that we were still deciding between a relational (MySQL) and non-relational (MongoDB) database, and that this decision would likely determine what tech stack (Django, MERN, etc.) the team would use to build the web application.
* **Changes made:** No major changes made.
* **Scrum Master Records:**

**Iteration 2: Developing the Database (28/08/23 - 03/09/23)**

“*Create the database in the approved database management system. Follow the approved database model to create the appropriate database structure. Enter the test data provided.*”

* **Plan:** Work on the design of the web application’s UI/UX. Decide on the DBMS (Database Management System), then determine what tech stack to use to implement the web application.
* **Objectives/Goals:** Create application design. Start GitHub repository. Research/select the DBMS and tech stack.
* **Decisions:**

The team selected ***EduArchive*** as the name for our web application, as it conveys the key idea of the application being an archive of educational content.

After considering the sample data that was provided for this project, the team decided that using MongoDB (a non-relational, document-oriented database) would best serve as our DBMS. MongoDB is a leading NoSQL database that offers flexibility and scalability for our platform. Its document-oriented structure allows for varied data formats, making it easier to store articles with different attributes without the need for a rigid schema. This flexibility aids rapid development and iteration, accommodating future expansions or modifications to the platform. Additionally, MongoDB’s scalability ensures that as the user base grows, the database can handle increased traffic and data volume without compromising performance.

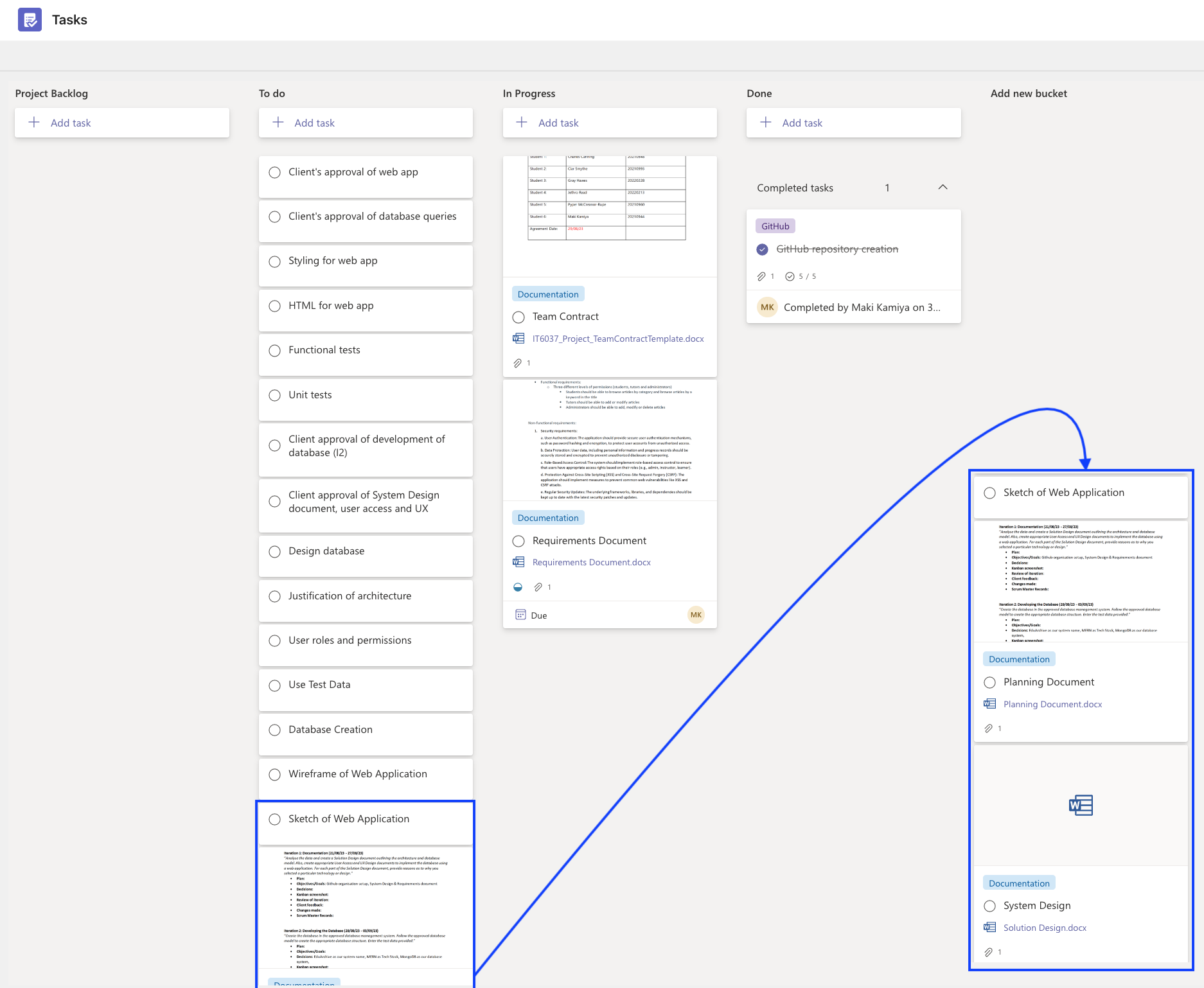
The team toyed with the idea of building the web application using the Django framework. However, Django has been designed to work best with relational databases, such as MySQL, Postgres, SQLite. Given the decision to go with MongoDB as our database solution, the other alternative the team considered was the MERN tech stack (MongoDB, Express, React, Node.js), and this was subsequently selected for this project. MERN is a very popular set of technologies used to build web applications; as such, there are many resources available on building applications utilising this tech stack. Moreover, given its popularity, the team thought it would be good to learn since team members had not had previous experience developing a full-stack web application using MERN.

Team members that will be involved in designing the web app’s UI/UX have chosen Figma as their preferred design application. Figma features an extensive set of design and collaboration tools that make it an ideal choice for designing the application, from wireframes to mock-ups and interactive prototypes.

We also discussed project roles and decided on the allocations listed below. Notably, we divided the Scrum Master role into two: one for the Frontend and another for the Backend. We also agreed to rotate this responsibility, ensuring every team member gets a chance to serve as a Scrum Master.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Role** | | | **Team Member Name** | | | |
| First Iteration: **Documentation** (21/08/23 - 27/08/23) | Second Iteration: **Developing the Database** (28/08/23 - 03/09/23) | Third Iteration: **Developing Database Queries** (04/09/23 - 10/09/23) | Fourth Iteration: **Implementing Database using Web Application** (11/09/23 - 15/-09/23) |
| **Team Leader** | | | Maki | Maki | Maki | Maki |
| **Scrum Master (Frontend)** | | | - | Jethro | Pyper | Gray |
| **Scrum Master (Backend)** | | | - | Charles | Ciar | Maki |
| **Designer** | Gray & Jethro | Gray & Jethro |  | | |  |
| **GitHub Manager** | | | Maki | Maki |  |  |
| **Main Developer** | | | Charles, Pyper, Ciar | Charles, Pyper, Ciar |  |  |

* **Kanban screenshot:** Captured on 30/08/2023



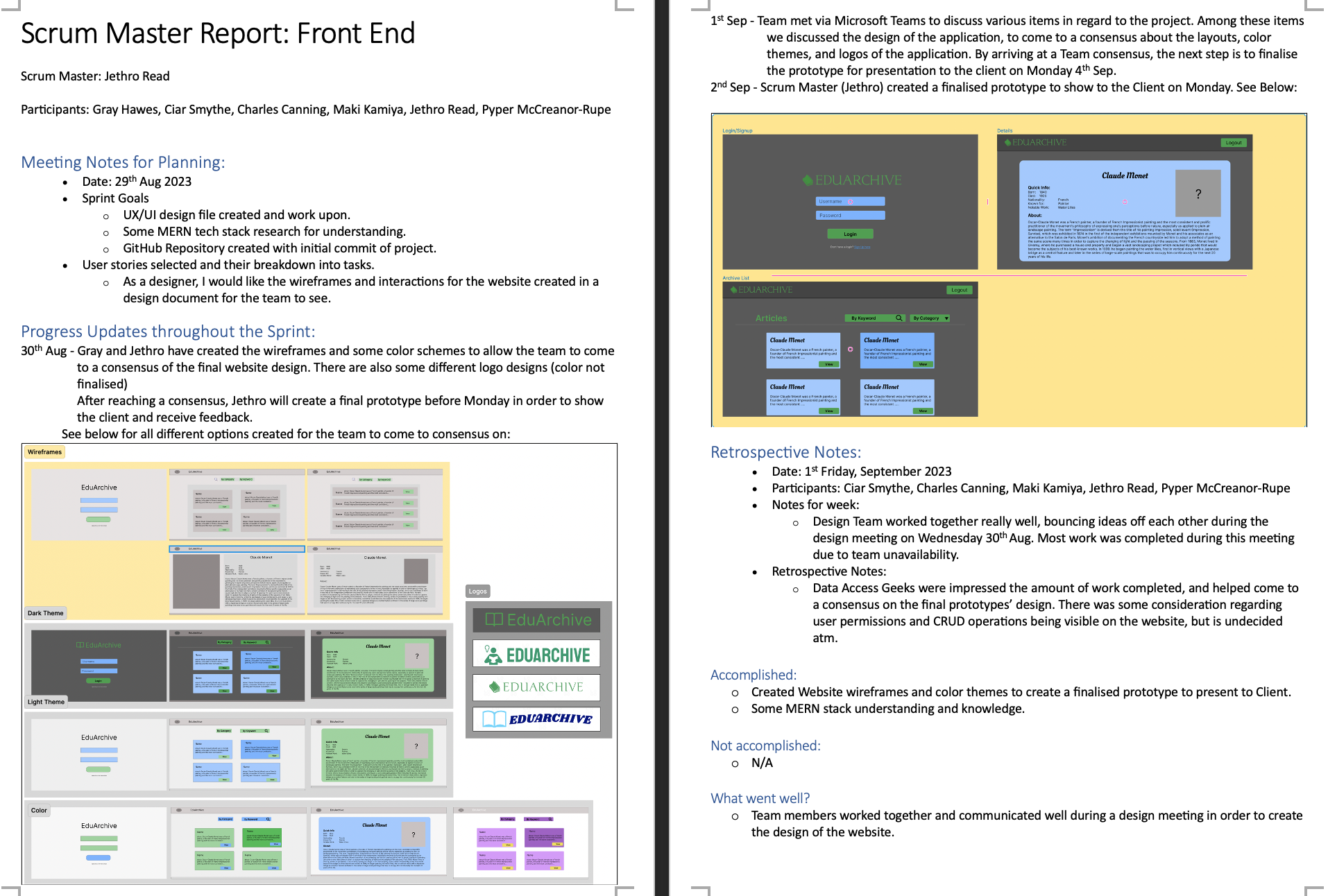
* **Review of iteration:**

Jethro and Gray worked on the design of the web application and produced wireframes and mock-ups for several design layouts and colour schemes. The team discussed the various design options and settled on a preferred layout and colour palette, and a finalised prototype was subsequently produced.

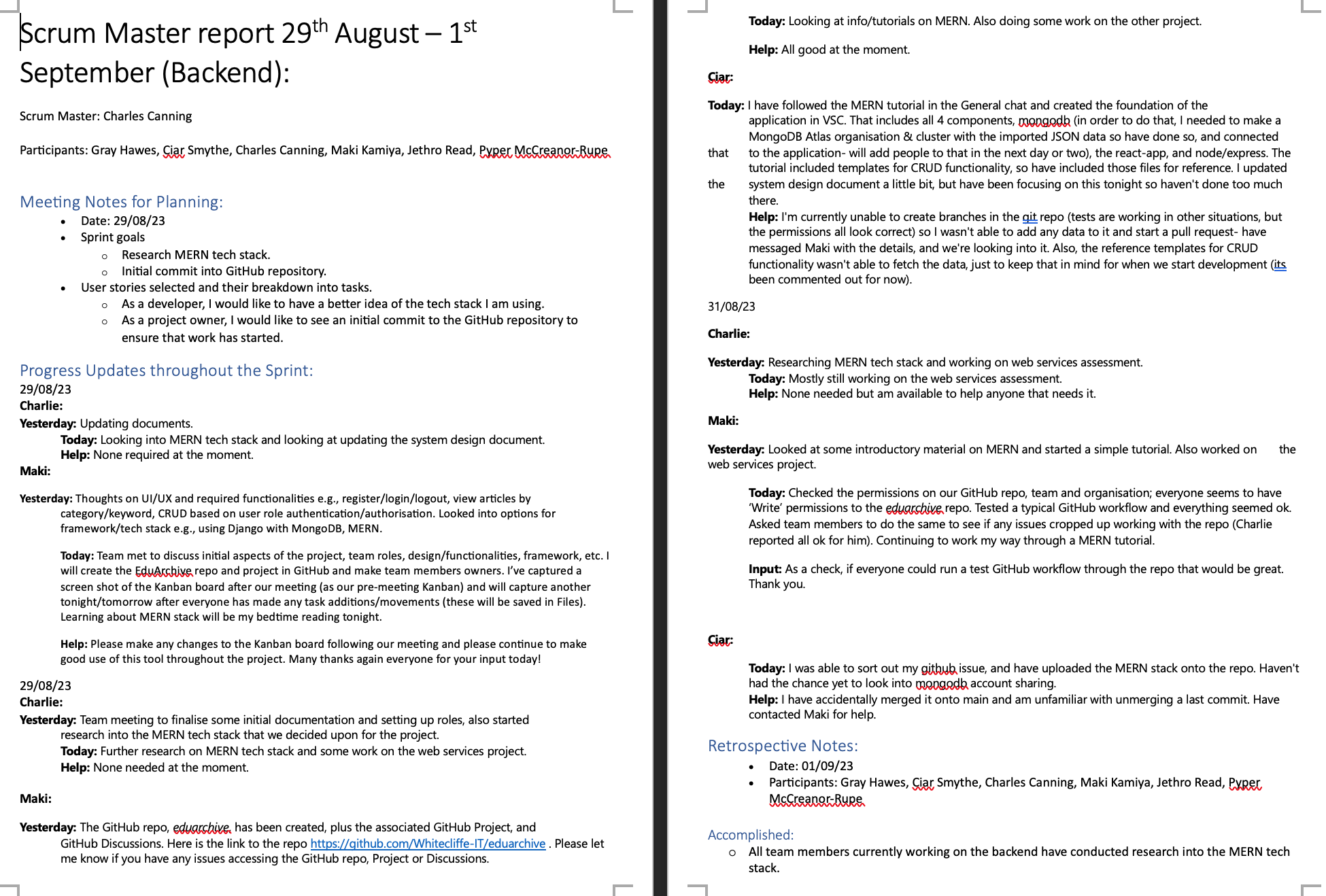
A GitHub organisation, Whitecliffe IT, and a repository, *eduarchive* (<https://github.com/Whitecliffe-IT/eduarchive>), were created to enable the team to collaborate on the project’s codebase. A GitHub team, *Data-Access-Geeks*, was set up for the project’s team members to join, and then the *Data-Access-Geeks* team was assigned to the *eduarchive* repository. Additionally, GitHub’s Project and Discussions features were initiated for the *eduarchive* repo. Team members were able to confirm that they could perform a typical GitHub workflow on the *eduarchive* repo, i.e., create a new branch, commit to that branch, submit a pull request, and merge the branch into the ‘main’ (default) branch of the *eduarchive* repo.

The team spent time researching the MERN stack its use in creating web applications. Ciar established a MongoDB Atlas database and enabled access to it for all team members. She also carried out the initial set up stages for the MERN application and its connection to the MongoDB database. This work was committed to the *eduarchive* repository.

* **Client feedback:** (Maki) I gave Bilal a brief update informing him of the team’s decisions to go with MongoDB as our database solution and MERN as our tech stack for implementing the web application. I also told him that the team had looked over several design options to arrive at a finalised prototype design. He seemed happy with the choices the team had made.
* **Changes made:** No major changes made.
* **Scrum Master Records:**



****



**A white page with black text

Description automatically generated**

# Iteration 3: Developing Database Queries (04/09/23 - 10/09/23)

“*Create and test the appropriate queries as per client requirements.*”

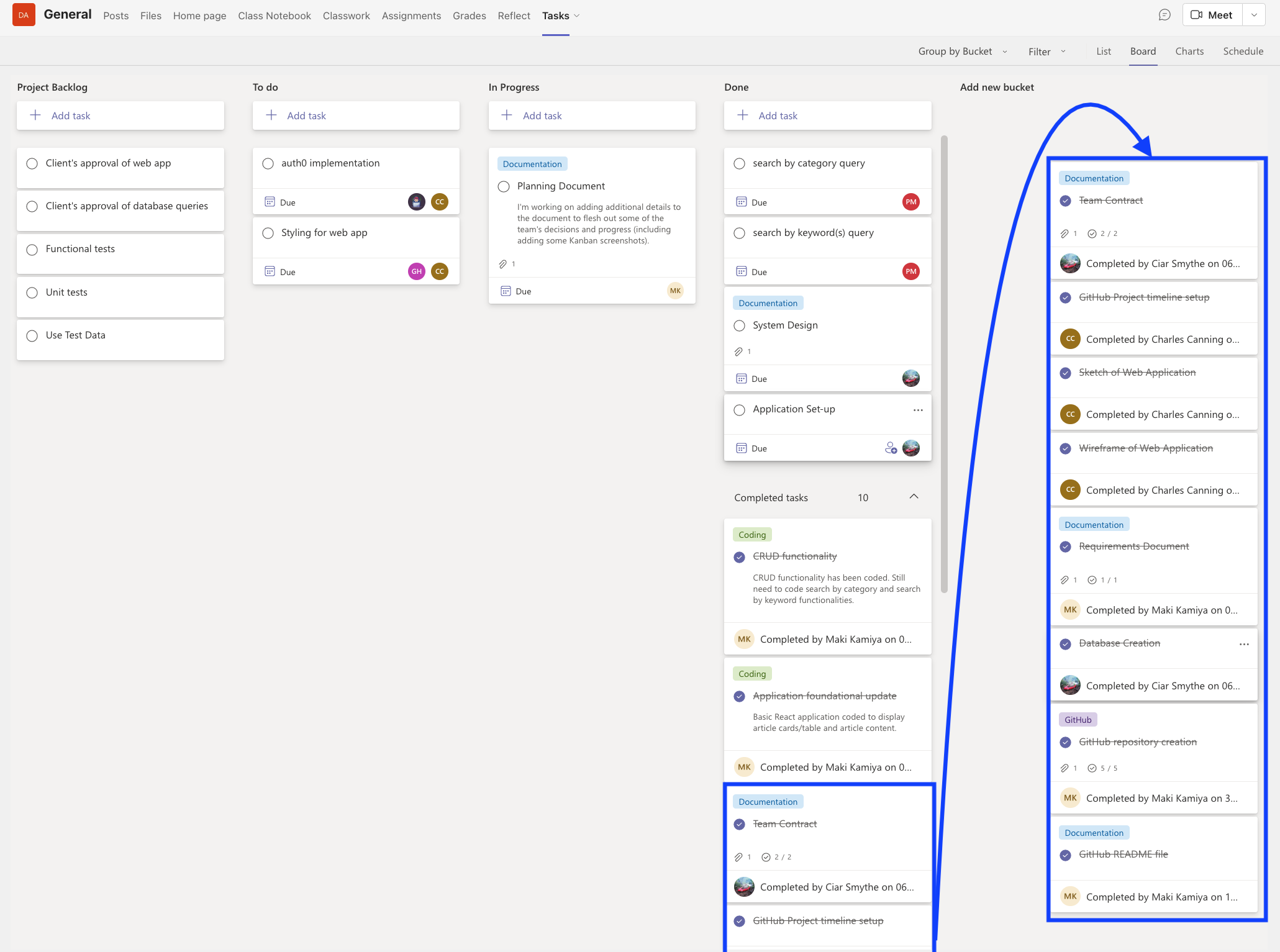
* **Plan:** Extend the initial web application that has been set up by adding CRUD operations, search by ‘category’ and search by keyword in ‘name’ functionalities.
* **Objectives/Goals:** Code CRUD and search queries. Begin CSS styling. Look at implementing authentication and authorisation.
* **Decisions:**

Before extending the web application with CRUD and search functionalities, the codebase was refactored from using create-react-app (CRA) to using **Vite** (<https://vitejs.dev/>). There are several reasons why Vite is a better React tooling solution than CRA (<https://vitejs.dev/guide/why.html>). Overall, Vite provides a much better developer experience (DX) and is significantly leaner and more performant. Moreover, the official React docs no longer recommend CRA as a solution for creating React applications.

For CSS styling, we have gone with **Tailwind CSS** (<https://tailwindcss.com/>), which is a utility-class CSS framework that enables CSS styling to be composed directly in the HTML rather than in a separate CSS file. This allows styling to be implemented more rapidly and the inline utility classes makes it easier to see what styling is being applied to each of the page elements. Tailwind CSS is also supported in Vite (<https://tailwindcss.com/docs/guides/vite>).

For authentication and authorisation, the team has decided to utilise a third-party identity validator, **Auth0 by Okta** (<https://auth0.com/>). The use of such third-party auth providers is increasingly common in web development. They have expertise in ensuring systems are secure and up to date with the latest security standards and practices, offering regular updates and protection against common web threats, e.g., CSRF, XSS, SQL injection attacks. Moreover, they offer SDKs and libraries that allow developers to quickly implement authorisation and authentication into their applications.

* **Kanban screenshot:** Captured 10/09/2023



* **Review of iteration:**

CRUD operations have been coded into an early-stage implementation of the web application. The UI displays individual cards for each article in the MongoDB database and allows users to create/read/update/delete article content.

A screenshot of a computer

Description automatically generated

Pyper has coded the search functionalities (search by category and search by keyword in the name field). The search by category feature allows users to select a particular category (All Categories, Arts, Mathematics, Technology) from a dropdown menu; this updates the page to display only the article cards matching the category that has been selected.

A screenshot of a computer

Description automatically generated

The search by keyword in the name field involves entering a search term into a search bar; article cards with matches in the name field are then displayed.

A white background with black dots

Description automatically generated

Charles has started coding the CSS styling of the web application to implement the design of the finalised prototype that Jethro and Gray produced during Iteration 2.

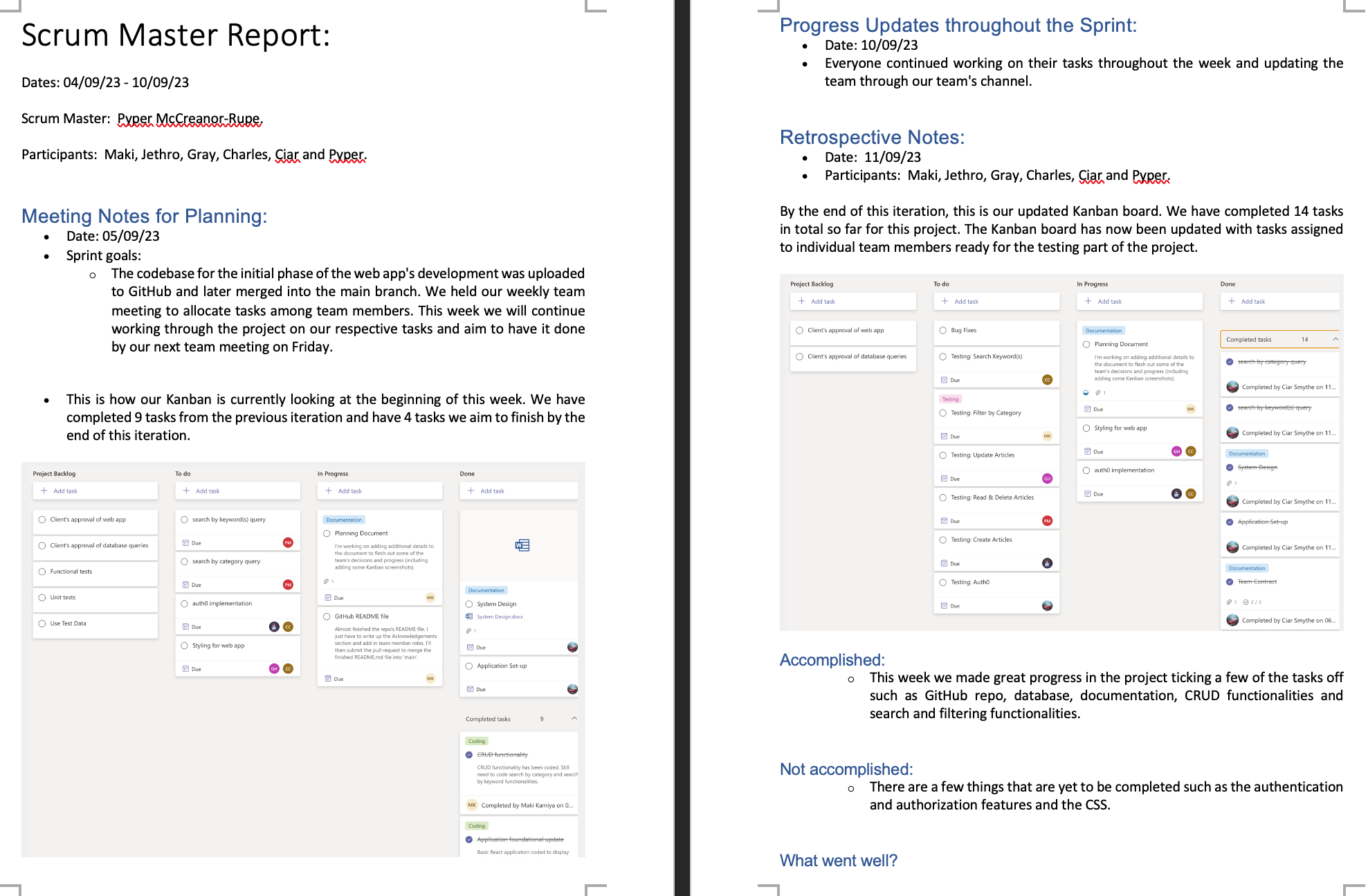
A screenshot of a computer

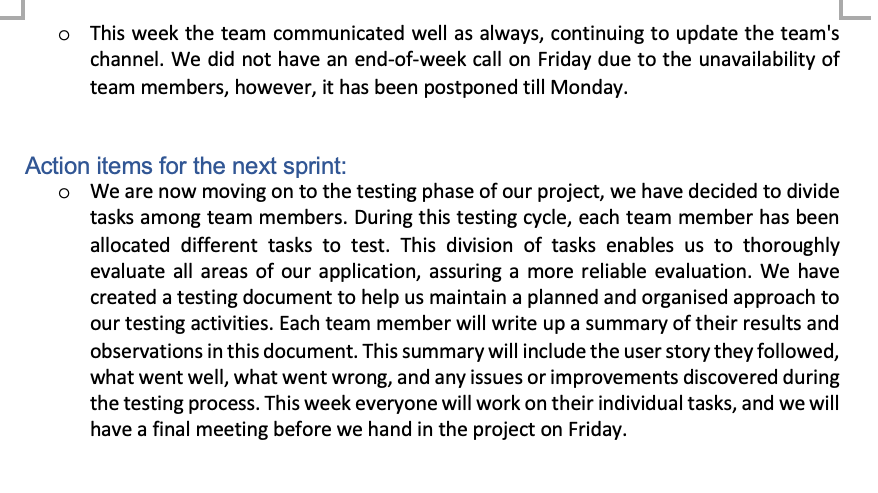
Description automatically generated

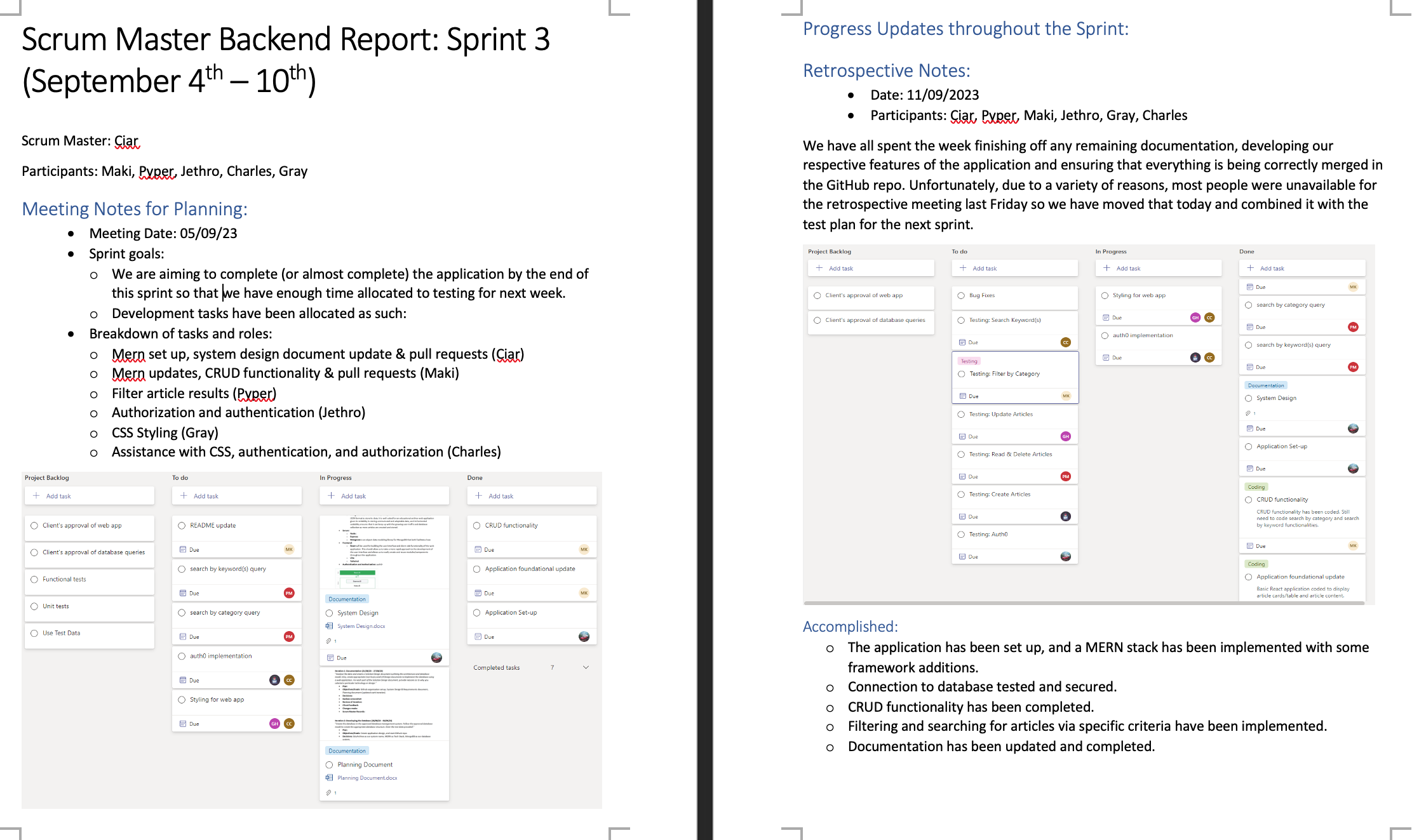
Jethro has been progressing with the implementation of the third-party Auth0 authentication/authorisation into the application.

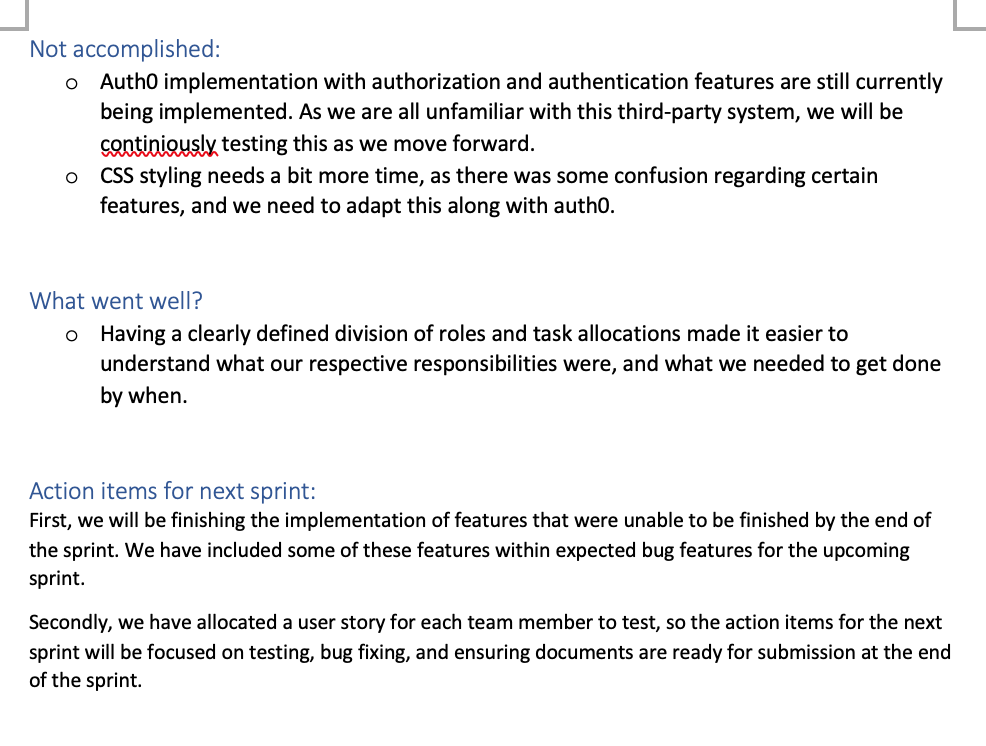
Lastly, the GitHub README for the *EduArchive* repository has been essentially completed (may need minor additions/amendments before final project submission).

* **Client feedback:**
* **Changes made:**
* **Scrum Master Records:**







****

# Iteration 4: Implementing Database using Web Application (11/09/23 - 15/09/23)

“*Create a single page interface (web or mobile) application to implement and test the client requirement for the database queries.*”

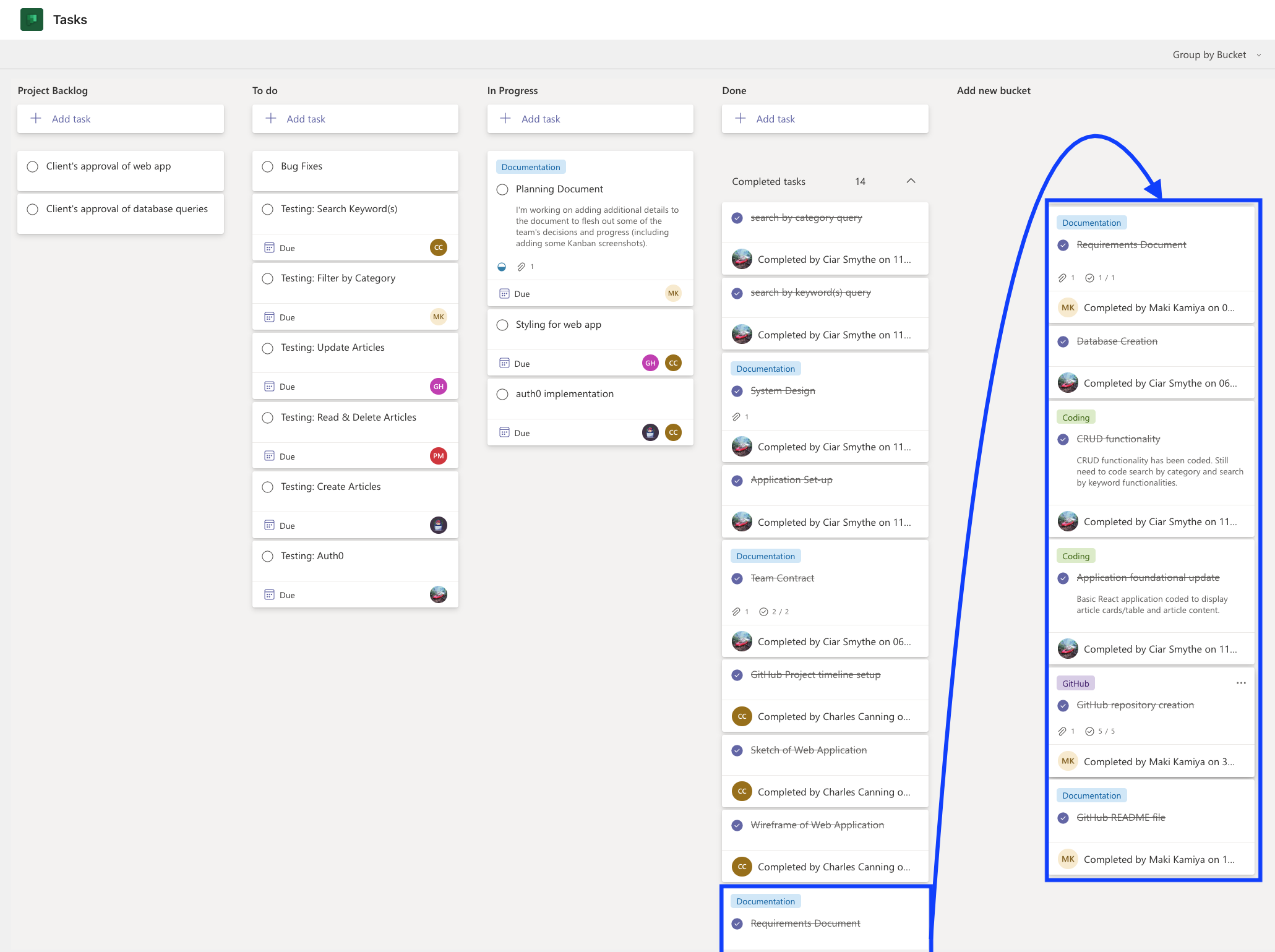
* **Plan:** Complete the web application, implementing the auth functionalities and CSS styling elements. Conduct testing of the application’s functional requirements.
* **Objectives/Goals:** Code authentication/authorisation functionality using Auth0 as the auth provider. Carry out testing of the various functional requirements. On Friday 15th September, submit the Team (Project) and Individual (Reflective Journal with Technical Commentary) components, along with Evaluation Forms (Team-, Peer-, Self-evaluation forms).
* **Decisions:**

Setting up Auth0 for our authentication needs has been more complex than we initially thought. Jethro will keep pushing forward to integrate it for both authentication and authorisation in our app. Meanwhile, Charles will handle the application's CSS styling based on the final prototype's design. However, be aware that some design tweaks might occur if they enhance the user experience (UX).

To avoid potential merge conflicts between the authentication work and CSS styling, we've decided to merge both tasks into one feature branch in the repository. This should reduce the risk of conflicts.

Every team member will be responsible for testing a particular functionality of the application. They'll also document their findings in the Testing Document. The breakdown is as follows:

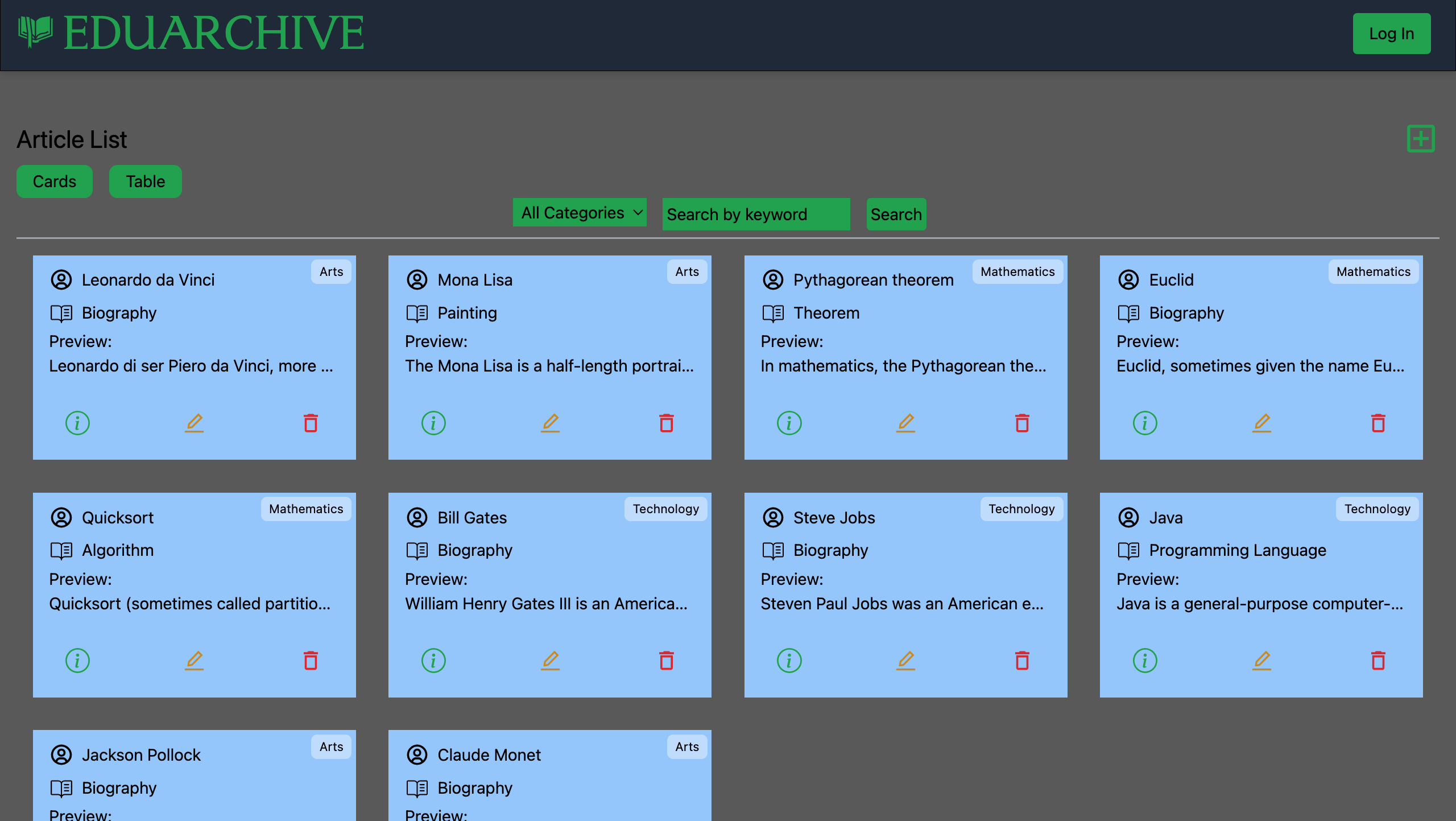
* Ciar: Auth0
* Jethro: Article Creation
* Pyper: Article Reading and Deletion
* Gray: Article Updates
* Maki: Filtering by Category
* Charles: Keyword Search
* **Kanban screenshot:** Captured 11/09/2023



* **Review of iteration:**

The implementation of the authentication/authorisation requirements using Auth0 as the third-party auth provider was unable to be fully completed, as it proved to be more complex than initially anticipated. The team are confident that had we had a little more time we would have been able to complete the auth implementation. As a record of the work that Jethro did for this, his code will remain on the open ‘9-Auth0’ feature branch of the repo but will not be merged into the ‘main’ (default) branch until such time coding the auth functionality may be completed; Jethro or another team member may choose to work on this at a later date perhaps.

The final CSS styling elements for our web application have been completed on the ‘8-css’ feature branch and has merged into the ‘main’ branch of the repo. With this merge, the codebase of our application has been completed for project submission.



Team members have carried out testing of the functional requirements of our web application, the results of which have been documented in the project’s Testing Document.

After the completion of the project’s documentation and GitHub repository requirements, the following will be carried out for the project’s submission:

* Individual component (submit individually):
  + Reflective Journal
  + Evaluation forms (Team-, Peer-, Self-)
* Team component submission (Maki will submit on behalf of the team):
  + All documents in MS Teams will be uploaded to the GitHub repo (except recordings).
  + A copy of the *articles* collection in MongoDB will be saved to the GitHub repo.
  + The above will also be saved to a ZIP file and submitted as the Team Component submission on Canvas with a submission note to Bilal advising that this is the combined work of all members of the Data Access Geeks team (with all team members named). A message will be posted on Teams advising the Team Component has been submitted.
  + A Teams message will also be posted @Bilal with the link to our GitHub repo, <https://github.com/Whitecliffe-IT/eduarchive>
* **Client feedback:** (Maki) I informed Bilal that our team will be submitting the project today, Friday 15th September 2023.
* **Changes made:** No major changes made.
* **Scrum Master Records:**

