Requirements Specification Document

1.Introduction:

The **Yorkshire and Humber Regional Organised Crime Unit (YHROCU)** requires a modern workflow management system for more efficiency with dealing with non-crime related activities. Their current system lacks streamlined task assignment, tracking, and collaboration features, necessitating the development of a web-based solution.

The team is working on a new system that will allow individual or collaborative assignments by assisting staff to manage tasks more effectively and successfully, The system will allow features such as real-time updates and automated emails notifications when assigning and completing tasks, supervisors will have better visibility and control over tasks that are given to staff while tracking the progress made and preventing any unauthorized deletion or lose to any data to ensure safe and smooth progress.

Furthermore, the system will allow more features like dynamic dashboard for all users depending on their position, ex managers will be able to view all tasks assigned to all staff and the progress made, while staff will only be able to see their assigned tasks and the progress they made themselves, more features such as data export capabilities as well as filters and search function to improve overall workflow transparency.
To enhance security in the system, OpenAuth-based authentication will integrate with the exciting structure, making this system a scalable and long-term solution for YHROCU's operational needs.

1.1 Team expertise:

Ahmad's strength as a team leader to excels in organization and support, in managing meetings, assigning/delegating tasks and ensuring accountability by following up until everything is completed. Also, his proactive approach has kept the team on track; working towards our given deadlines. He has also spearheaded the collation of this report; pooling together the team's collective work into one cohesive piece.

Dan's strength is that he's an excellent communicator. He has been the glue that holds the team together while also managing any interpersonal issues that have come up as the project has developed. He was crucial in making sure that people are understanding what needs to be done and work towards given deadlines. He also has prior experience with web design and SQL database implementation.

Fizan's strengths lie in documentation, organizing meetings, recording key details, and ensuring the team stays on track. Additionally, he is proficient in JavaScript, Python, and PHP, allowing him to contribute to the project's functionality in its later stages. Lastly, he

has some prior experience with unit testing from last year's software development module.

Zakir's expertise include Comprehensive Research and Justification writing, Visually Intuitive User Interfaces; he is Proficient in Photoshop & Web Design. Past Experience in PHP, JavaScript, CSS and Python.

Murtaza excels in documentation and organizing the GitHub repository. He is also confident in JavaScript and Python, enabling him to assist with front-end functionalities after the itinerary stage of the project is completed. Additionally, his strong background research skills make him best suited to contributing to the risk assessment aspect of the work.

Tom's key strength is mainly with overseeing the coding and guiding other team members with the coding tasks that were given to them. He has given consistent and detailed updates as the development has progressed. Tom's competency with implementation of key features is what has allowed the team to produce a viable prototype that fit the given brief.

Ali's strength has been in implementing some HTML pages with CSS styling. Ali has had limited coding experience before but has brought his enthusiasm to team meetings and has sought help when needed.

2. Functional (and non-functional) requirements:

Functional Requirements

User Roles and Permissions

Admin Role –

- Can Create, edit and delete user accounts (No sign-up is required as this is a closed system)
- Manage user roles and permissions

Manager/Supervisor Role –

- Can view all projects and sort by the due date.
- Assign tasks/projects to users and track the ongoing progress.
- Separate manager view with a dashboard summarizing project statuses with filters included for urgency and due dates.
- **View a single log per project that consolidates all task updates in date order, providing an overall picture without needing to click through individual tasks.**

User Role –

- Create and update assigned tasks.
- View personal tasks and track progress.

Task/Project management

- Users can create tasks, set due dates and assign them to individuals or teams.
- Tasks cannot be deleted but can be updated to ensure accurate auditing.
- Supervisors can view and manage all tasks and assignments.

Notifications

- Automatic email notifications for task assignment, task updates and overdue tasks.

Reporting/Data exportation

- Supervisors can **export** project data in CSV or PDF format.
- There will be a search system for tasks based upon keywords, due dates or assigned users.

Access Control

- Supervisors can configure task visibility and provide only assigned users to be able to view the task or for it to be viewable by all users.
- The system will use OpenAuth for secure authentication and compatibility with the existing systems in use by the client.
- Data will be stored locally to ensure data availability.

Non-Functional Requirements

UI Design

- The UI design will be minimalistic to allow for ease of use by users.
- There will be a dark mode/high contrast mode to provide better viewability.

Security & Data integrity

- There will be strict access control measures in place to prevent unauthorized access.
- No deletions are allowed, ensuring storability of data.
- Security roles will be separated for Admins, supervisors and users.

Performance & Compatibility

- The system should be optimized for use on laptops and should use a responsive web design.
- The system dashboards should be optimized and not take a long time to load.

Notifications

- Email notifications should be timely and sent at appropriate times (work hours) to avoid spam.

3. Data Description:

This web application was constructed using a mixture of HTML, CSS and PHP with connections to the database using SQL. The styling has been matched to the website mock ups as best as we could achieve given the time constraints of the project. CSS styling was delegated through the implementation of a dummy page without database connection to demonstrate so others could work on CSS without needing database access.

The log in page first used hardcoded usernames and passwords but later was updated to pull from the database of users with IF statements to verify passwords with the database, once verified, the user details are stored in a SESSION variable which can be accessed between pages on the site. SESSION variables are stored on the server and can be accessed by any page using the session_start() method. This allows login details to be consistent across the whole site without the repeated use of GET and POST to pass information from one page to the next.

After a successful login the user is taken to the list view which shows all tasks which that user is authorised to view:
Users can only view tasks which are assigned to them, whereas managers and admins can view all tasks in the database.

Clicking on an entry in the list will take the user to a detailed view of that task which shows additional information and permits managers and admins to edit the details of the selected task, and added a task archival function where it lets staff remove tasks from public view without removing the information entirely which fulfils the functional requirement to have an audit log.

The database contains 2 tables one for tasks and the other for users, the task table contains columns for id, subject, project, assignee, status and priority and the user table contains columns for id, username, password and clearance level.

Several repeated blocks of code, such as the page header and footer or the database connection code, have been sectioned off into separate files which are inserted into each page as necessary to minimize redundancy.

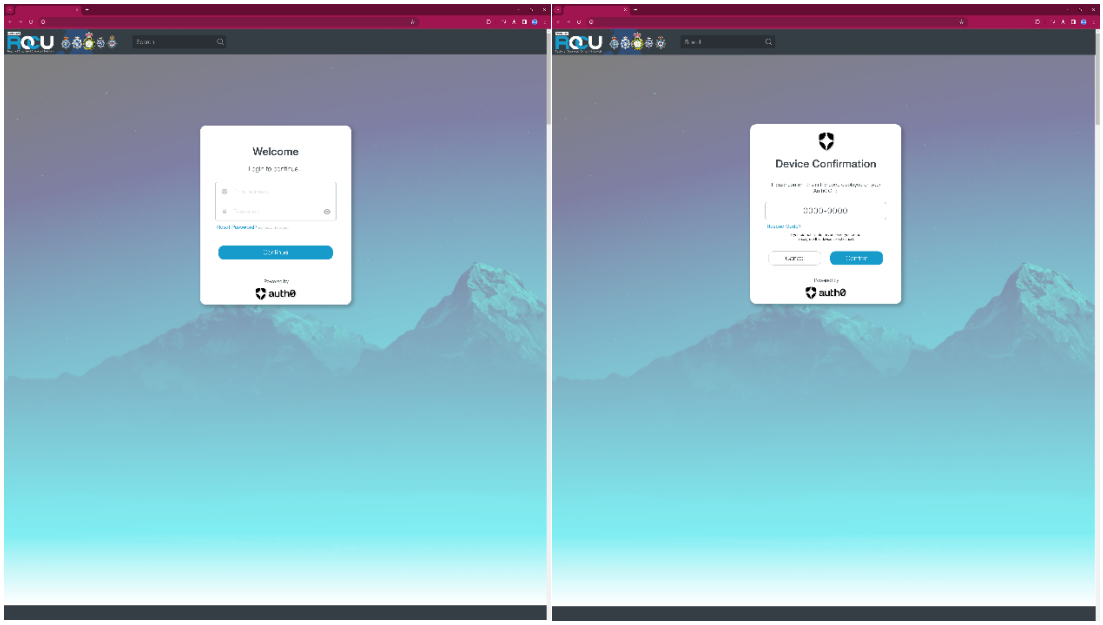
When the user has finished using the site, they can log out of their session which deletes all SESSION variables and ends the session requiring the user to log back in whenever they wish to start a new session.

4. INTERFACE:

4.1 Layout Diagram:

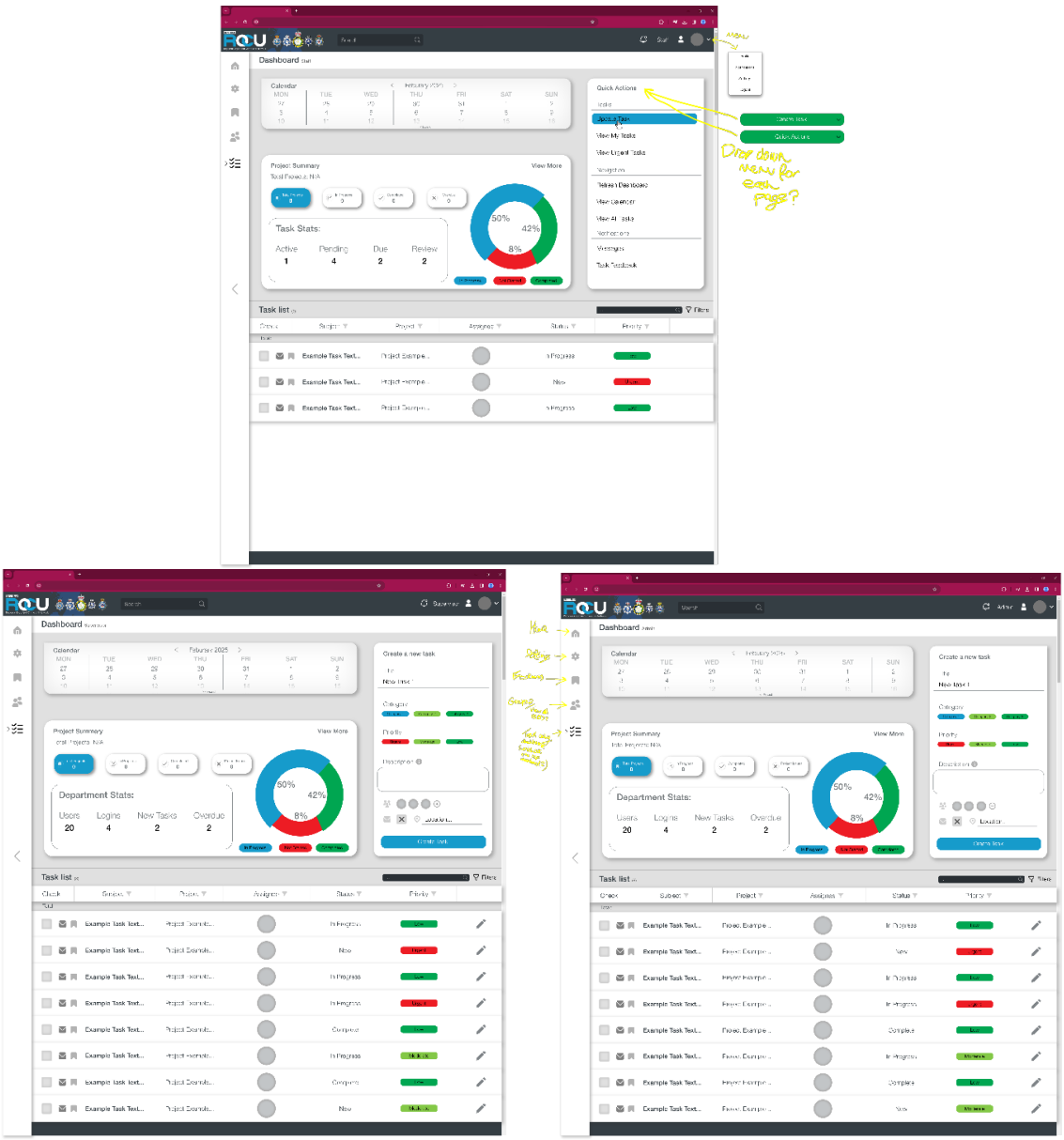
This is a general layout diagram of how the website starts from PAGE I – LOGIN and how it is navigated.

4.2 Login & Authentication:



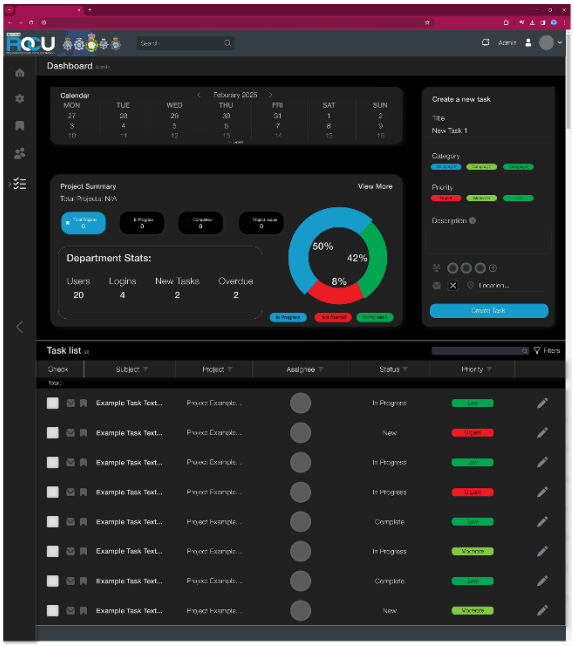
Starting at Page 1, you are asked to log in.
Provide the users email address and password. Login triggers database connection if the details exist.
Password reset will send an email/notification and is done through admin user access level for security.
Page 2, using Open Authentication, will send a code to your device for login security.

4.3 Dashboard:



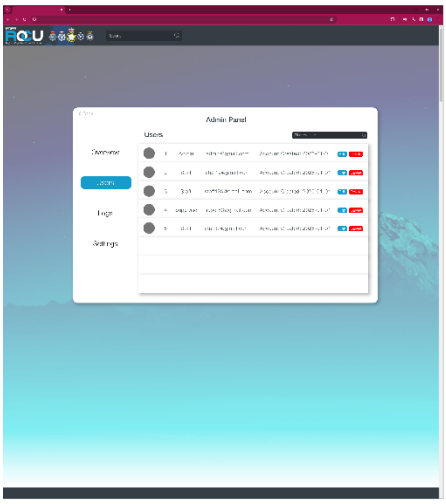
After user authentication, depending on the access level of that user's account, the site will redirect to Page 3 which is the dashboard.
Here the user can see stats based on the access level (for admins & supervisors all task and project data are shown, for staff their task and project data are shown).
Create task column is only shown to admins & supervisors, task details, assigned users and tags can be an added here.
Task list will show all tasks relevant to that users access level (all tasks are shown for admin & supervisors, only their tasks are shown to staff)

4.4 Dark Mode:



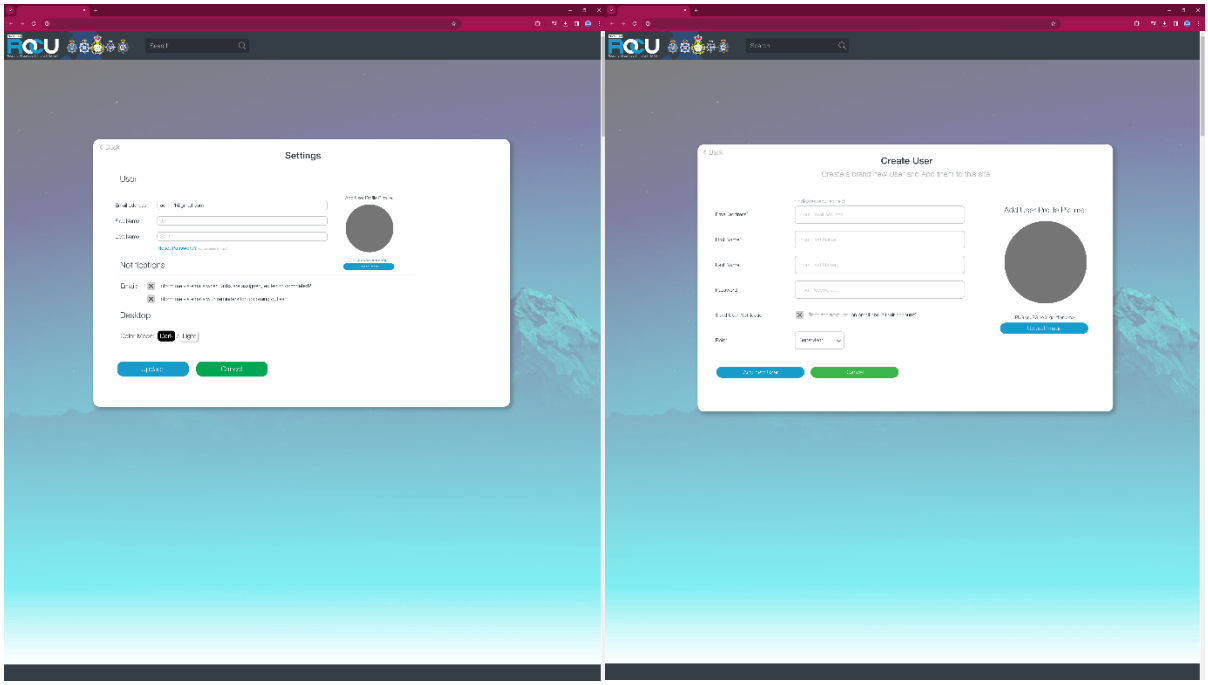
Alternate lighting styles, Light & Dark Mode, will be available to switch from.
Here is an example of the dashboard with Dark Mode enabled.

4.5 Admin Panel:



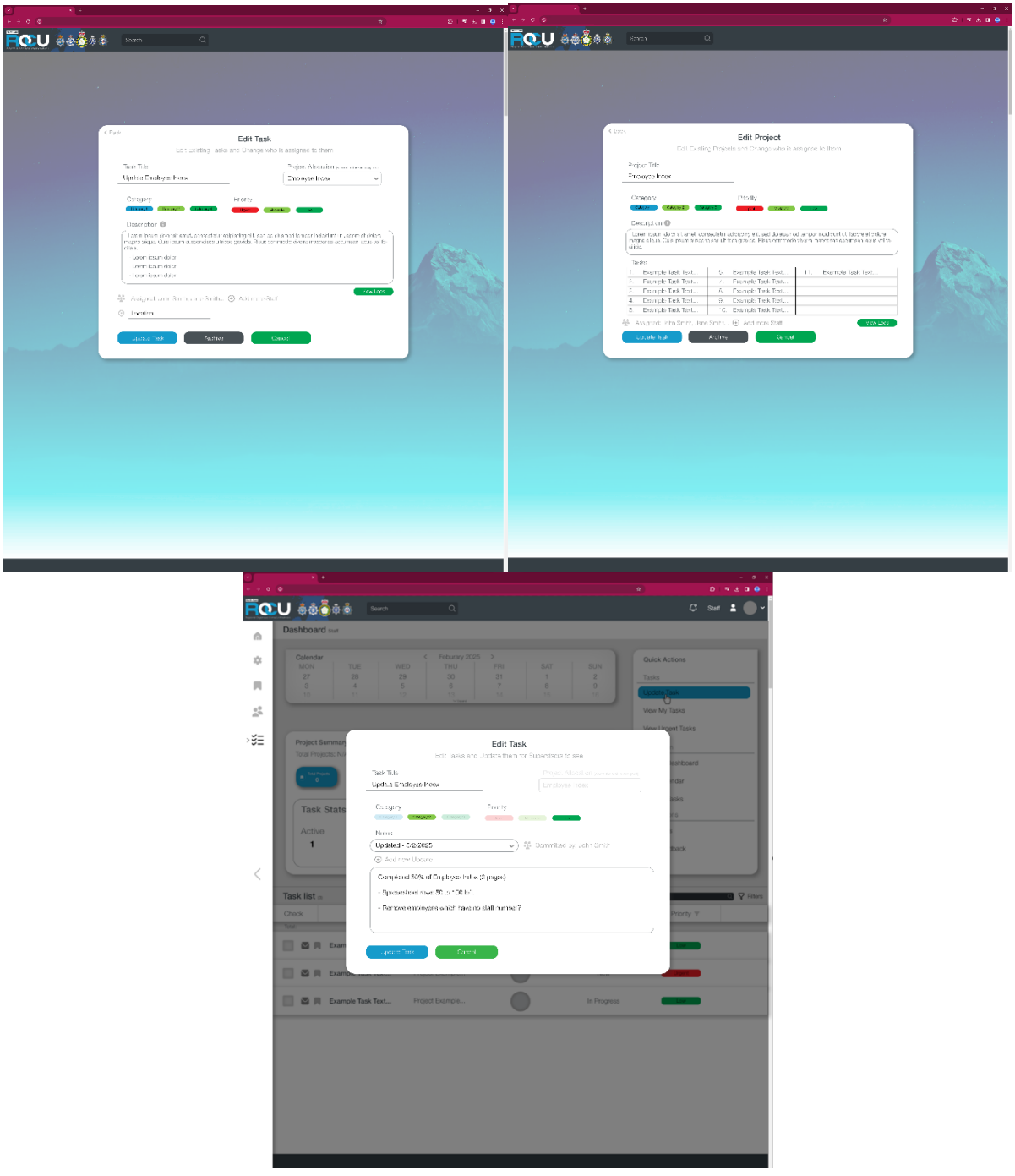
The admin panel can be accessed through Admin user access level.
This panel allows admins to see an overview of all users, projects, tasks.
Users tab will allow admins to see a list of all users in the database and edit, delete them.
Logs tab will allow admins to see a list of logs for every task or project, time, date and who the actor is. Exporting all logs from the database will create a CSV file for the user to download.
Settings, like normal user settings, will also allow global notification settings to be written, download data or backup and restore.

4.6 Settings & Create User:



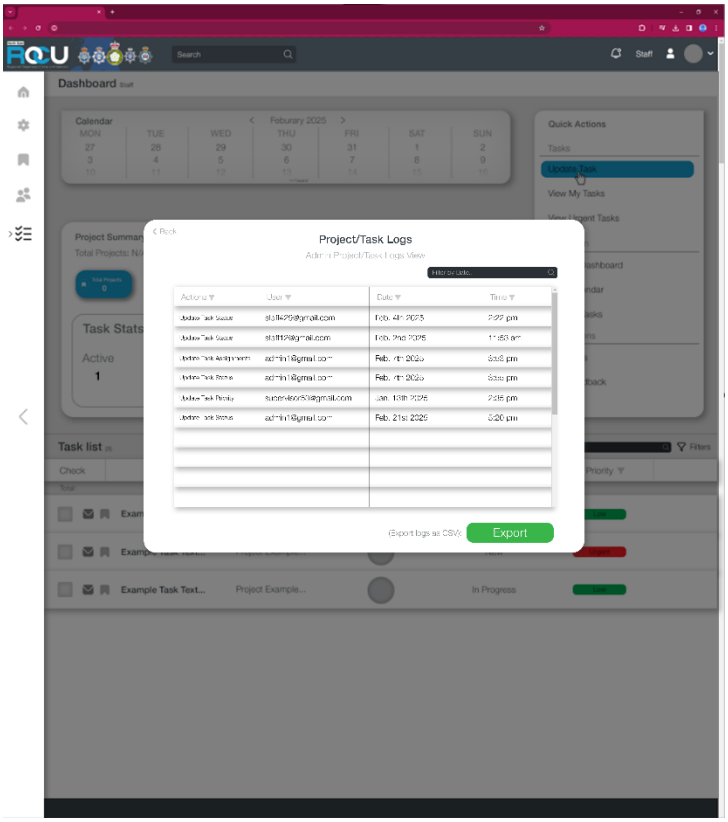
Settings can be accessed via the drop-down menu on the right of the top bar. (4.3 Dashboard)
Users email address and profile picture can be changed while First & Last name cannot and is restricted to Admin user access level.
Notifications to inform the user via email can be opted in or out of here.
Colour modes can be changed here also, setting stored in the database per user

4.7 Edit Project /Task:



For admins & supervisors editing tasks & projects will allow for every detail to be changed: title, project allocation, category, priority, description, assigned users, location.
Project allocation can be changed via drop down menu showing all current projects available.
Tasks & Projects cannot be deleted, only archived.
For staff, editing tasks will allow for certain details to be changed: Add notes/updates (shows actor who committed and date of commit) Can be navigated via drop down menu for each task. This will be a permanent change log for each task.

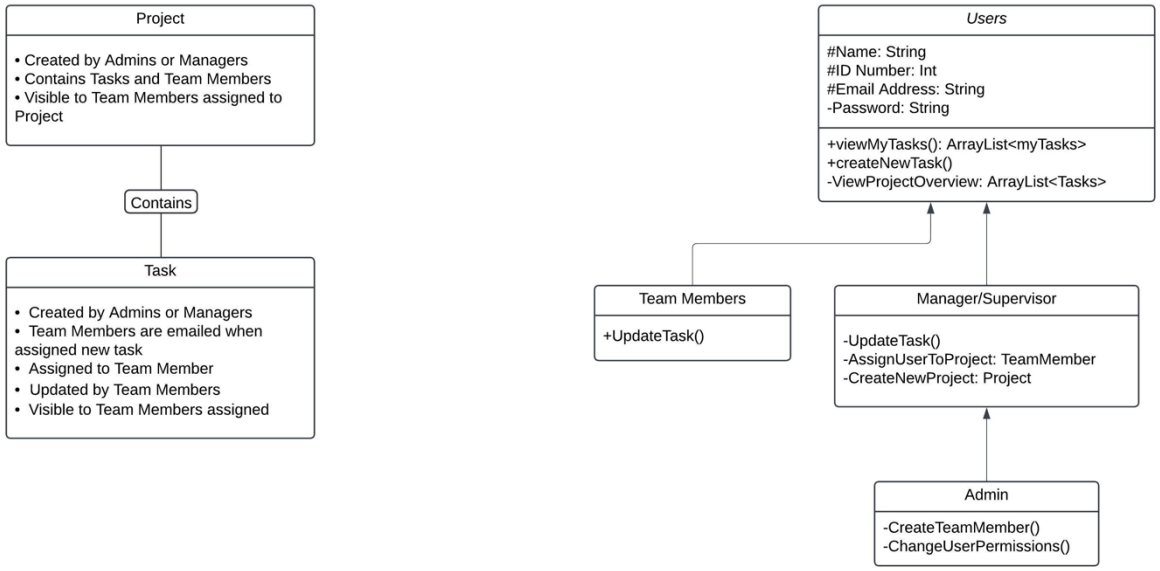
4.8 Project/Task Logs:



The Project/Task Logs page allows administrators & supervisors to check the activity of a specific project or task. This extends (4.7 Edit Project/Task) and will be displayed on both pages in that section.
The activity or actions are shown in date/time priority order with a search bar to allow for finding specific actions. Users & Actions are displayed for each activity.
Activity for specific projects or tasks can be exported to CSV.

UML Class Diagram:

UML Class Diagram



The UML class diagram shows the relationship between the different subclasses of users and their respective functionality. Admins are a subclass of Managers and have the exclusive power to add users to the system and to change a Team Member's permissions to that of a Manager.

The class diagram also shows the relationship between Tasks and Projects; Tasks are contained within the projects and tasks are assigned to Team Members.

5. LSEPI and Risk Assessment:

The Yorkshire and Humber Regional Organised Crime Unit (YHROCU) require a workflow management system to manage non-crime-related projects. This risk assessment will outline potential risks associated with the deployment of the workflow system and then provide solutions to mitigate or remove those risks. This will lead to the efficient and secure operation of the system while also adhering to the YHROCU's requirements.

Risk 1 – Unauthorised Access to the System
This risk means a party that is not linked to any of the system actors gains unauthorised entry to the system.
Prevention to Risk #1 – Implementation of OpenAuth Verification and Minimalistic User Privileges
To mitigate this risk, OpenAuth will be used to verify user logins. Additionally, since actors are given privileges strictly related to their role—such as staff only being able to update task progression and input into a log. This ensures staff accounts do not have significant authority in the system. In a worst-case scenario where a staff's user account is compromised, the administrator can delete the affected account to restore security to the system.

Risk 2 – Data Loss
This risk means that tasks, logs, or other system data could become lost or inaccessible due to hardware failure, software corruption, or malicious attacks—especially critical given that all servers are maintained locally.
Prevention to Risk #2 – Scheduled Backups and Local Recovery Measures
To mitigate this risk, the client will be informed on how the system can incorporate regular automated backups using Windows Server Backup, storing the backup data on a separate local storage device. It is assumed that there are already backup protocols in place that the client uses, and the above prevention method is a generic method that can easily be implemented.

Risk 3 – Email Notification Spoofing
This risk means that malicious actors could spoof system-generated email notifications, misleading users by impersonating YHROCU communications.
Prevention to Risk #3 – Official Email Affiliation and Verification Measures
To mitigate this risk, notifications should ideally be sent from an email address affiliated with YHROCU, minimising the chance of spoofing. Due to the nature of the client, it is assumed that such email protocols are already in place.

Risk 4 – SQL Injection
This risk means that malicious actors could exploit vulnerabilities in input fields by injecting SQL commands, potentially allowing unauthorised access or manipulation of the system's database.
Prevention to Risk #4 – Secure Coding Practices and Input Validation
To mitigate this risk, secure coding practices will be followed, and all user inputs will be strictly validated. This includes using prepared statements and sanitizing input data to ensure that no harmful code can be executed.

Risk 5 – Search Privacy Risk
This risk means that the search function might allow staff users to view tasks or projects that they are not assigned to, potentially exposing sensitive or unauthorized data.
Prevention to Risk #5 – Enforce Role-Based Access Control in Search
To mitigate this risk, the search functionality will be designed to return only those tasks or projects that are explicitly assigned to the querying user. This ensures that staff cannot see data beyond their permitted scope, while managers—with broader access—retain the ability to view all tasks as needed.

Risk 6 – Exported CSV/PDF Data Privacy Risk
This risk means that sensitive data contained in exported CSV or PDF files could be compromised if these files are intercepted or accessed by unauthorized parties.
Prevention to Risk #6 – Encrypt Exported Files
To mitigate this risk, all exported CSV/PDF files will be encrypted. This ensures that even if the files are accessed outside the system, their contents remain protected and unreadable by unauthorised users.

Risk 7 – Malicious Staff/Manager

This risk means that a staff or supervisor may deliberately misuse their privileges to compromise system integrity or disrupt operations.
Prevention to Risk #7 – Administrative Oversight
To mitigate this risk, the administrator can promptly delete or disable any staff or manager found to be acting maliciously, ensuring that harmful access is quickly removed from the system.

Risk 8 – Inadequate User Training and Operational Misuse
This risk means that staff may inadvertently misuse the system due to insufficient training, potentially leading to data errors or security vulnerabilities.
Prevention to Risk #8 – Comprehensive Training and Documentation
To mitigate this risk, provide thorough user training sessions and develop clear, accessible documentation. This ensures that all staff understand how to use the system properly and securely, reducing the likelihood of accidental misuse.

Legal Issue 1 – Data Protection and Privacy Compliance

This issue means that the system will handle personal and sensitive work-related data, which falls under the UK GDPR and the Data Protection Act 2018. Failure to comply could incur legal penalties and lead to breaches of confidentiality.

Prevention of Legal Issue #1 – Access Controls
To mitigate this issue, the system will ensure access is strictly controlled through role-based authentication, and only authorised personnel can view relevant tasks. OpenAuth authentication will be implemented to integrate with existing security infrastructure.

Legal Issue 2 – Compliance with Electronic Communications Regulations

This issue means that the system's email notifications must comply with UK regulations to prevent spam-like behaviour or unauthorised email handling.

Prevention of Legal Issue #2 – Official Email Domains and Consent-Based Notifications
To mitigate this issue, the system will use YHROCU's official email domain to send notifications. Users will also have clear guidelines on the nature of email alerts to ensure compliance with legal standards.

Social Issue 1 – Staff Privacy and Monitoring Concerns

This issue means that staff may feel their work is excessively monitored due to the system logging all task updates and progress tracking.

Prevention of Social Issue #1 – Transparency in System Usage
To mitigate this issue, clear policies will be established to explain the purpose of logging, ensuring that it is for task management rather than personal surveillance. Staff will be informed that the system is designed to improve workflow efficiency rather than to track individual performance.

Social Issue 2 – Accessibility and Ease of Use

This issue means that not all users may have the same level of technical expertise, leading to difficulties in system adoption.

Prevention of Social Issue #2 – User-Friendly Design and Accessibility Compliance
To mitigate this issue, the system will be designed with an intuitive interface that follows accessibility standards such as WCAG 2.1. User training sessions and support documentation will also be provided to ensure all staff can effectively use the system.

Ethical Issue 1 – Fair Task Distribution and Management

This issue means that task assignments could be unfairly distributed, either through bias or mismanagement, potentially leading to overburdening of certain employees.

Prevention of Ethical Issue #1 – Transparent Task Assignment and Manager Oversight
To mitigate this issue, managers will have oversight of all tasks, and an auditing function will be included to ensure fair distribution.

Ethical Issue 2 – Security vs. User Convenience

This issue means that while strict security measures are necessary, they must not make the system overly complex or difficult to use.

Prevention of Ethical Issue #2 – Role-Based Access Control with Minimal Friction
To mitigate this issue, the system will balance security and usability by assigning role-based access that ensures users only have the permissions they need without unnecessary restrictions. Two-factor authentication (2FA) will be considered for the administrator and managers handling sensitive data.

Ethical Issue 3 – Privacy in Task Searches

This issue means that if search functionality is not properly controlled, staff may be able to view tasks they are not assigned to, exposing unauthorised data.

Prevention of Ethical Issue #3 – Search Results Restricted by Role-Based Access
To mitigate this issue, the search function will only return tasks that the user has explicit permission to view. Managers will have broader search capabilities but only within their authorised scope.

Professional Issue 1 – System Reliability and Data Integrity

This issue means that, as a critical tool for YHROCU, the system must be reliable and capable of securely storing all task data without loss or corruption.

Prevention of Professional Issue #1 – Regular Backups and Local Data Storage
To mitigate this issue, the system will implement scheduled backups using Windows Server Backup or a similar tool, ensuring that data is safely stored on a separate local device. Local storage will be prioritised to align with YHROCU's existing IT policies.

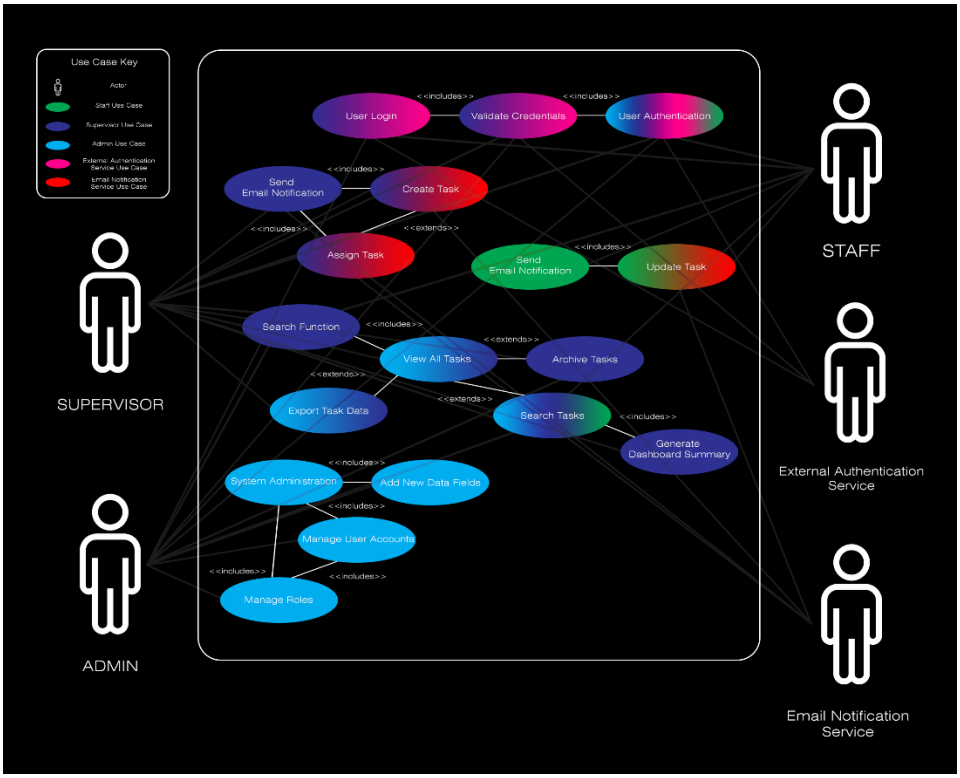
Professional Issue 2 – Training and User Adoption

This issue means that if users are not properly trained, they may struggle to use the system effectively, leading to mistakes or resistance to adoption.

Prevention of Professional Issue #2 – Comprehensive Training and Documentation
To mitigate this issue, user training sessions will be conducted, and a detailed system manual will be provided. This will help ensure that all staff understand how to use the system efficiently.

6. UML and Use Cases:

1 Use Case Diagram:



This Use Case Diagram provides a top-level view of the system's functionality. This illustrates how each actor (Staff, Supervisor, Administrator, External Authentication Service, and Email Notification Service) interacts with various use cases. The use cases are color-coded by role to help visually distinguish which actions belong to which actor.

2 Actors:

This table explains the purpose and basic background functionality of each actor's privileges & features.

Actor	Purpose
Staff	can view, update, manage their assigned tasks in each project.
Supervisor	has elevated responsibility which reflects in their privileges e.g. create, assign, finalize tasks. Also has access to overview features.

Administrator	manages system configuration e.g user accounts, adding/editing projects.
External Authentication	handles login and credential validation via **Open Auth**.
Email Notification Service	sends notifications when tasks are created, assigned or updated.

3 Justification:

Use Case 01: User Login
The goal of this use case is to authenticate user logins & gain access to the system based on their role within the system. In case of invalid credentials, the system displays an error message and asks the user to retry.
Actors include Staff, Supervisor, Administrator, External Authentication Service
Includes: **User Authentication**

Use Case 02: User Authentication
The goal of this use case is to authenticate any user login using an external authentication service such as **OpenAuth**.
Actors include Staff, Supervisor, Administrator, External Authentication Service
Includes: **Validate Credentials**

Use Case 03: Create Task
The Supervisors can create new tasks, set details and can choose to assign Staff.
Actors include Supervisor, Email Notification Service
Includes: **Send Email Notification** (automatically notifies staff)
Extends: **Assign Task** (optionally assign task to staff and send notification to those staff members)

Use Case 04: Assign Task
Supervisor can link a specific task to a Staff member which will define visibility to that user.
Actors include Supervisor
Includes: **Send Email Notification** (alerts assigned staff members of the task)
Extends: **Create Task**

Use Case 05: Update Task
Staff updates the status or review date and adds progress notes to the rolling log for that project.
Actors include Staff, Email Notification Service
Includes: **Send Email Notification** (alerts assigned staff members of the task)

Use Case 06: View All Tasks
Supervisors or Administrators view a full list of all tasks, with filtering and sorting tabs.
Actors include Supervisor, Administrator
Includes: **Search Function**
Extends: **Export Task Data, Search Tasks, Archive Tasks**

Use Case 07: Archive Tasks
Supervisor or Administrator archives a task, will no longer be shown on the dashboard or available to any staff.
Actors include Supervisor Administrator
Extends: **View All Tasks**

Use Case 08: Export Task Data
Users export tasks or project data to CSV for record-keeping.
Actors include Supervisor, Administrator
Extends: **View All Tasks**

Use Case 09: Add New Data Fields
Administrators modify the task structure by adding new data fields like Priority Level.
Actors include Administrator
Includes: **System Administration**

Use Case 10: Manage User Accounts
Administrator creates, updates or removes user accounts with any role.
Actors include Administrator
Includes: **System Administration**

Use Case 11: Search Tasks
Users search by keyword or other data fields to quickly locate tasks.
Actors include Staff, Supervisor, Administrator
Include: **Generate Dashboard Summary**
Extends: **View All Tasks**

Use Case 12: Generate Dashboard Summary
Shows statistics for aggregated functions e.g. Number of Tasks, Users etc
Actors include Supervisor, Administrator
Includes: **Search Tasks**

Use Case 13: Manage Roles
Change or Assign roles (Supervisor, Administrator, Staff) for new or existing users.
Actors include Administrator
Includes: **Manage User Accounts**

Use Case 14: Send Email Notification
Sends users via email a notification when a task is created, assigned to them.
Actors include Email Notification Service
Includes: **Update Task, Assign Task**
Extends: **Create Task**

Use Case 15: Validate Credentials
Takes user login details and validates them using external authentication service.
Actors Include Staff, Supervisor, Administrator, External Authentication Service

Includes: **User Login, User Authentication**

Use Case 16: Search Function
Users search for tasks, projects using search bar with filtering
Actors include Staff, Supervisor, Administrator
Includes: **View All Tasks**

Use Case 17: System Administration
Administrator access level has elevated permissions such as editing users, creating/editing data fields.
Actors include Administrator
Includes: **Add Data Fields, Manage User Accounts, Manage Roles**

7. Work Plan:

https://github.com/Murtaza-2x/Team6-YHROCU-workflow/blob/main/DOCUMENTATION/WORK%20PLAN/GANTT%20CHART%20ENTERPRISE%20PRO.xlsx

8. GitHub:

https://github.com/Murtaza-2x/YHROCU-workflow

9.Literature Review:

Task management within project management has been evolving over time and will continue to do so, from early Gantt charts and linear "Waterfall" approaches to modern and adaptive frameworks. Traditional methods, such as the Critical Path Method and Waterfall approaches (Seymour, 2025) are very effective in well-defined projects but can become messy and unorganised in dynamic or fast paced environments. This gave rise to Agile methodologies (Nelson, 2023), including Scrum, which emphasizes iterative development such as sprints, frequent feedback, and continuous improvement.

Kanban focuses on visualizing work, limiting work-in-progress, and reducing bottlenecks. Unlike Scrum's timeboxed sprints, Kanban (Oivo, 2013) manages tasks as a flow, with cards moving across columns (e.g "To Do," "In Progress," "Done") (Atlassian, n.d.). Both methods prioritize adaptability, transparency, and short feedback loops. Hybrid models like "Scrum-ban" combine elements of both frameworks, offering additional flexibility.

In parallel, project management software has grown more sophisticated, with tools such as Microsoft Project, Jira, and Trello supporting different approaches. These platforms offer collaboration features like real-time tracking and often integrate

automation or analytics to optimize workflow (Choban, 2024). Best practices include breaking tasks into manageable components, maintaining visibility, fostering communication, and regularly reviewing progress. This ensures teams can respond effectively to changing requirements and deliver consistent value to stakeholders.

References:

Atlassian. (n.d.). Retrieved from Atlassian: https://www.atlassian.com/agile/project-management/kanban-principles#:~:text=Bite%20off%20only%20what%20you,can%20chew

Choban, J. (2024, December 12). Retrieved from HeroCoders: https://www.herocoders.com/blog/jira-time-tracking-everything-you-need-to-know

Nelson, J. (2023, August 5). Retrieved from slideshare: https://www.slideshare.net/slideshow/agile-project-management-a-systematic-literature-review-of-adoption-drivers-and-critical-success-factors/259641666

Oivo, M. (2013, September). Retrieved from Research Gate: https://www.researchgate.net/publication/260739586_Kanban_in_Software_Development_A_Systematic_Literature_Review#:~:text=identified%20as%20primary%20studies%20relevant,practitioners%20to%20gain%20a%20better

Seymour, S. (2025, March 5). Retrieved from monday.com: https://project-management-software.financesonline.com/history-of-project-management-software-how-it-developed/#:~:text=In%201917%2C%20one%20of%20the,of%20the%20project%20manager%E2%80%99s%20kit

10. Peer review:

Ahmad Ghrewi 22011384:

As the team leader, organized meetings and attended them, delegated tasks amongst the team and followed up with team members to make sure we worked to a timeline. Identified challenges and facilitated discussions to resolve issues efficiently.
Ahmad completed the introduction for the report and provided oversight as the report has come together. Ahmad met with Tom to support him to do the data description section, as well as helping with the HTML coding tasks and Database tasks. Ahmad also helped lead the client meeting, making sure that there was clarity for the task and making sure that the team presented themselves smartly by wearing suits and being professional.
Also, Ahmad's strength is with managing the team and following up with team members until task is done. Team score 10/10.

Dan Nicholson 21010733:

Completed the UML class diagram, lead the meeting with the client making sure to present professionally. Dan coordinated with team members to wear suits to look appropriate for the client meeting, helped contribute code to the website application's styling and functionality and contributed to the report documentation for the app,

attended meetings consistently, communicated well throughout the project and helped manage/contact team members that weren't communicating consistently.
Also, Dan's strength is being the speaker of the team as he is good with talking and presenting the team in the best way possible. Team score 9/10.

Zakir Khan 21010492:

worked on the Interface Mock-up for the client interview, then continued to flesh out the design for all webpages + justification that would be implemented, then he made alterations to this considering the clients requests and feedback. He also worked the Use Case Diagram + explanation for this project and refined the HTML & CSS for the Client demo. He attended 99% of meetings & has communicated well when collaborating with other team members.
Also, Zakir's strength is with being punctual and doing tasks perfectly as they should be, especially graphic designs related tasks. Team score 10/10.

Fizan Anjum 19009890:

He worked on collecting team minutes for all the meetings and arranged meeting times for the group, worked on the functional requirements document by himself. Alongside this he adjusted the NDA to use for project 1 and chased & distributed it amongst the team to sign.
He also structured the GitHub by creating some folders to adhere to the guidance provided by Savas in the lectures.
Also, notes were taken as the team secretary in the meeting with the client and relayed back to the team.
He has also been in touch with the client as the main person chasing for feedback/agreements.
In terms of punctuality Fizan has attended 99% of meetings (missing only 1 last minute online meeting) and contributed significantly to the team.
Also, Fizan's strength is being secretary of the team and documenting and recording everything and having work done on time in the right way. Team score 10/10.

Mohammed Murtaza 20006765:

Murtaza, attended meetings consistently, communicated well throughout the project, completed the risk assessment, contributed to the client meeting, completed the LESPI section of the report and followed up with other team members to make sure they had the support they needed.
Also, Murtaza's strength is with how seriously he takes the task that is been assigned to him and giving a very good effort towards doing it and making it in the best way possible. Team score 9/10.

Tom Kershaw 22008855:

Tom implemented the mock-up website using HTML and PHP and documented the development process by using GitHub to track design iterations.

Also, Tom’s strength is mainly with overseeing the coding and guided other team members with the coding tasks that were given to them. Team score 9/10.

Ali Haider 23056767:

Ali attended 3 meetings, including the client meeting and coded a simple login page using HTML and CSS.
Team score 3/10.