

# Enterprise Pro:

Coursework 1: Documentation and Demonstration of Software

Project 4: Interactive real-time Visualisation Dashboard – Team 25

23039430/1 AFZAL, Ehtasham [eafzal2@bradford.ac.uk](mailto:eafzal2@bradford.ac.uk)

23015945/1 BISWAS, Kian [k.biswas@bradford.ac.uk](mailto:k.biswas@bradford.ac.uk)

23040217/1 KHALID NACIRI, Lina [l.khalidnaciri@bradford.ac.uk](mailto:l.khalidnaciri@bradford.ac.uk)

21007967/3 KHAN, Tayyib [t.khan34@bradford.ac.uk](mailto:t.khan34@bradford.ac.uk)

23045736/2 NAZAR, Mahad [m.nazar2@bradford.ac.uk](mailto:m.nazar2@bradford.ac.uk)

23021200/1 QAZI, Areesha [a.qazi5@bradford.ac.uk](mailto:a.qazi5@bradford.ac.uk)

22008496/2 RASHEED, Adam [a.rasheed6@bradford.ac.uk](mailto:a.rasheed6@bradford.ac.uk)

Github: <https://github.com/Hedgeturd/ep-team-repo>

06/03/2025

Software Requirements Specifications  
Interactive real-time Visualisation Dashboard

## Contents

1. Introduction .....	5
1.1 Purpose .....	5
1.2 Document Conventions.....	5
1.3 Intended Audience and Reading Suggestions .....	5
1.4 Product Scope .....	5
1.5 Rationale .....	6
1.6 Team Expertise .....	6
1.7 References.....	6
2. Overall Description.....	6
2.1 Product Perspective .....	6
2.2 Product Functions .....	6
2.3 User Classes and Characteristics .....	7
2.4 Operating Environment.....	7
2.5 Design and Implementation Constraints.....	7
2.6 User Documentation .....	7
2.7 Assumptions and Dependencies .....	7
3. External Interface Requirements .....	7
3.1 User Interfaces .....	7
3.2 Hardware Interfaces.....	7
3.3 Software Interfaces .....	7
3.4 Communication Interfaces.....	8
3.5 User Interface Design .....	8
Main Dashboard.....	8
Historical Data Page .....	8
Alerts Page Desktop .....	9
Alerts Page Mobile .....	10
Item Dropdown .....	11
4. System Features .....	11
4.1 Real-Time Dashboard .....	11
4.1.1 Description and Priority .....	11

4.1.2 Stimulus/Response Sequences .....	12
4.1.3 Functional Requirements .....	12
4.2 Historical Data View .....	12
4.2.1 Description and Priority .....	12
4.2.2 Stimulus/Response Sequences .....	12
4.2.3 Functional Requirements .....	12
4.3 Alerts & Notifications .....	12
4.3.1 Description and Priority .....	12
4.3.2 Stimulus/Response Sequences .....	13
4.3.3 Functional Requirements .....	13
4.4 User Authentication & Role Management .....	13
4.4.1 Description and Priority .....	13
4.4.2 Stimulus/Response Sequences .....	13
4.4.3 Functional Requirements .....	13
4.5 AI/ML Anomaly Detection.....	13
4.5.1 Description and Priority .....	13
4.5.2 Stimulus/Response Sequences .....	13
4.5.3 Functional Requirements .....	14
4.6 User Experience & Accessibility .....	14
4.6.1 Description and Priority .....	14
4.6.2 Stimulus/Response Sequences .....	14
4.6.3 Functional Requirements .....	14
4.7 Use Case Diagram .....	15
4.7.1 User .....	15
4.7.2 Admin.....	16
5. Non-functional Requirements.....	16
5.1 Performance Requirements .....	16
5.2 Security Requirements .....	16
5.3 Scalability & Maintainability .....	16
6. Legal, Social, Ethical, and Professional Issues .....	17
6.1 Legal .....	17
6.1.1 Data Protection .....	17
6.1.2 Industry and Regulatory Compliance .....	17
6.1.3 Contractual Agreements and Intellectual Property .....	17
6.2 Social .....	17
6.2.1 Fair Workload Distribution .....	17

6.2.2 Work-Life Balance .....	17
6.2.3 User Experience and Accessibility.....	18
6.3 Ethical.....	18
6.3.1 Data Ethics and Accountability.....	18
6.3.2 Employee Wellbeing and Communication .....	18
6.3.3 Ethical Compliance .....	18
6.4 Professional.....	18
6.4.1 Clear Responsibilities and Assignments .....	18
6.4.2 Administrative Responsibilities .....	19
7. Work Plan.....	19
8. Peer Review & Contributions .....	21

# 1. Introduction

## 1.1 Purpose

This document specifies the requirements for the Interactive Real-Time Visualisation Dashboard for Rakusens. The system is designed to monitor temperature data from multiple sensors, providing real-time insights for production operators and managers. This will allow them to make data-driven decisions to improve product quality, energy efficiency, and operational transparency.

## 1.2 Document Conventions

This document follows structured software requirement specifications. Sections are numbered, and key information is highlighted for clarity.

## 1.3 Intended Audience and Reading Suggestions

- Developers – To understand system design and implementation requirements.
- Project Managers – To track system progress and scope.
- Client (Rakusens) – To validate system functionalities.
- End Users (Operators, Admins) – To understand system capabilities.

## 1.4 Product Scope

The Interactive Real-Time Visualisation Dashboard will:

- Display real-time sensor data using visualisations (graphs, charts, heatmaps).
- Alert users to anomalies based on predefined thresholds.
- Store and retrieve historical data for trend analysis.

- Provide different access levels (Operator, Admin).
- Integrate with AI/ML models for anomaly detection.

## 1.5 Rationale

This project was selected to enhance real-time monitoring and data-driven decision-making at Rakusens. It will help improve product consistency, prevent overheating/cooling errors, and allow for energy-efficient baking.

## 1.6 Team Expertise

Our team is proficient in:

- Front-end Development: JavaScript, HTML, CSS
- Back-end Development: Node.js, Express.js, Python (for AI/ML integration)
- Database Management: PostgreSQL
- APIs & Data Handling: JSON, SQL

## 1.7 References

- Rakusens Client Specifications
- PostgreSQL Documentation
- Express.js API Development Guidelines

# 2. Overall Description

## 2.1 Product Perspective

This dashboard is a new implementation designed to replace or complement existing manual temperature monitoring methods at Rakusens. It will be deployed on internal company servers or cloud infrastructure.

## 2.2 Product Functions

- Real-Time Data Display: Temperature updates from sensors displayed graphically.
- Threshold-Based Alerts: Colour-coded alerts (Green, Amber, Red) when temperature exceeds limits.
- Historical Data Retrieval: Ability to filter and view past temperature trends.
- User Authentication & Access Control: Different permissions for Operators, Managers, and Admins.
- AI/ML Integration: Machine Learning model to detect temperature anomalies.

## 2.3 User Classes and Characteristics

- Operators: Can monitor real-time data but cannot modify system settings.
- Admins: Can approve/reject user registrations and generate reports.

## 2.4 Operating Environment

- Web-Based: Accessible via modern browsers (Chrome, Firefox, Edge).
- Compatible with Internal & Cloud Hosting.
- Optimised for Desktop and Tablet Viewing.

## 2.5 Design and Implementation Constraints

- Data Security: Must comply with company regulations for data protection.
- Performance: The system must handle real-time updates with minimal latency.
- Scalability: The system should support additional sensors in the future.

## 2.6 User Documentation

- User Manuals for Operators and Admins.
- Technical Documentation for Developers.

## 2.7 Assumptions and Dependencies

- Sensors are correctly calibrated and connected to the system.
- Internet access is available for cloud-based functionalities.

# 3. External Interface Requirements

## 3.1 User Interfaces

- Login & Dashboard
- Graphical Data Display (Line Charts, Heatmaps, Alerts)
- Admin Panel for User Management

## 3.2 Hardware Interfaces

- Sensors transmitting data via internal servers/cloud.

## 3.3 Software Interfaces


- PostgreSQL Database for storing sensor data.
- Express.js API for data retrieval and processing.
- Machine Learning API (Python-based) for anomaly detection.

### 3.4 Communication Interfaces

- RESTful API for fetching and sending sensor data.
- Email/SMS Notifications for critical alerts.

### 3.5 User Interface Design

#### Main Dashboard



Home   Historical Data   Alerts   Settings   Sign Out

### Main Dashboard Area

Total Sensors  
Active  
[Live Count]

Alerts Triggered  
[Active Alerts]


Real-Time Temperature Graph  
(Line graph - Sensor data updating in real-time)

Sensor Table (Sortable and Searchable)

Sensor ID	Location	Current Temp	Status
001	Line 4	120°C	Green
002	Line 5	180°C	Red
003	Line 4	150°C	Amber

© 2025 Rakusen's - Real-Time Dashboard

#### Historical Data Page



Home   Historical Data   Alerts   Settings   Sign Out

### historical data page

Date range selector and filters  
[Start date: ] [End date: ]  
[Sensor ID: ] [Line Number: ]  
[Status: Green Amber Red]  
[Apply Filters] [Reset Filters]

Historical Temperature Graph  
(Line graph - temperature trends over time)

Sensor Table (Sortable and Searchable)

Sensor ID	Location	Current Temp	Status	Timestamp
001	Line 4	120°C	Green	2025-02-10
002	Line 5	180°C	Red	2025-02-11
003	Line 4	150°C	Amber	2025-02-12

© 2025 Rakusen's - Real-Time Dashboard



Alert Page

Alert Filter and Search Options

[Sensor ID: \_] [Line Number: \_]

[Alert Level: \_Low\_Medium\_High]

[Date Range: \_ - \_]

[Apply Filters] [Reset Filters]

Recent Alerts & notifications panel

[high] sensor 002 - 195°C (exceed threshold)

[MEDIUM] Sensor 005 - 175°C (Close to limit)

[LOW] Sensor 007 - 140°C (Stable)

Alert Table (Sortable and Searchable)					
Sensor ID	Location	Current Alert ID	Status	Timestamp	Level
001	Line 4	1001	Read	2025-02-10	HIGH
002	Line 5	1002	Read	2025-02-11	MEDIUM
003	Line 6	1003	Unread	2025-02-12	LOW



# Alert Page

**Alert Filter and Search Options**  
[Sensor ID: ] [Line Number: ]  
[Alert Level: \_Low \_Medium \_High]  
[Date Range; \_ - \_]  
[Apply Filters] [Reset Filters]

**Recent Alerts & notifications panel**  
[high] sensor 002 - 195°c (exceed threshold)  
[MEDIUM] Sensor 005 - 175°c (Close to limit)  
[LOW] Sensor 007 - 140°c (Stable)

## Alert Table (Sortable and Searchable)

Sensor ID	Location	Current Alert ID	Status	Timestamp	Level
001	Line 4	1001	Read	2025-02-10	HIGH
002	Line 5	1002	Read	2025-02-11	MEDIUM
003	Line 6	1003	Unread	2025-02-12	LOW

Item Dropdown





[Home](#)

[Historical Data](#)

[Alerts](#)

[Settings](#)

[Sign Out](#)

Alert Table (Sortable and Searchable)					
Sensor ID	Location	Current Alert ID	Status	Timestamp	Level
001	Line 4	1001	Read	2025-02-10	HIGH
002	Line 5	1002	Read	2025-02-11	MEDIUM
003	Line 6	1003	Unread	2025-02-12	LOW

4. System Features

4.1 Real-Time Dashboard

4.1.1 Description and Priority

The real-time dashboard provides continuous monitoring of temperature readings from multiple sensors. It includes graphical visualisations such as line charts, heatmaps, and traffic-light alerts to

indicate normal, warning, or critical states. The dashboard updates automatically to reflect the latest readings. Priority: High

#### 4.1.2 Stimulus/Response Sequences

1. The user accesses the dashboard.
2. The system retrieves real-time sensor data from the database.
3. The system updates the displayed readings and highlights anomalies.
4. The user observes the data and, if necessary, takes action.

#### 4.1.3 Functional Requirements

- REQ-1: The system must display real-time sensor readings from Line 4 and Line 5.
- REQ-2: The dashboard must refresh data automatically every 5 seconds.
- REQ-3: The system must provide a colour-coded alert system for abnormal readings.
- REQ-4: The system must support interactive graphs and charts.

### 4.2 Historical Data View

#### 4.2.1 Description and Priority

Users can view past temperature trends by selecting specific date ranges. This feature enables operators and managers to analyse sensor behaviour over time and detect patterns. Priority: High

#### 4.2.2 Stimulus/Response Sequences

1. The user selects a date range in the historical data interface.
2. The system retrieves relevant data from the database.
3. The system displays temperature trends for the selected period.
4. The user can analyse the data using graphs and filters.

#### 4.2.3 Functional Requirements

- REQ-1: Users must be able to filter historical data by date range.
- REQ-2: The system must retrieve and display historical sensor data efficiently.
- REQ-3: Users must be able to export historical data in CSV, PDF, and Excel formats.
- REQ-4: The system must support zooming and panning in historical charts.

### 4.3 Alerts & Notifications

#### 4.3.1 Description and Priority

The system must provide instant alerts when temperatures exceed predefined thresholds. Notifications ensure that operators and managers can respond quickly to critical conditions. Priority: High

### 4.3.2 Stimulus/Response Sequences

1. The system continuously monitors temperature data.
2. If a threshold is exceeded, the system generates an alert.
3. The system displays a notification on the dashboard.
4. If critical, the system sends an email or SMS alert to assigned personnel.

### 4.3.3 Functional Requirements

- REQ-1: The system must classify alerts using a traffic-light system (Green = Normal, Amber = Warning, Red = Critical).
- REQ-2: Alerts must be displayed in real-time on the dashboard.
- REQ-3: Critical alerts must trigger email and SMS notifications.
- REQ-4: Users must be able to acknowledge and track alerts.

## 4.4 User Authentication & Role Management

### 4.4.1 Description and Priority

The system supports secure login and role-based access control (RBAC) to manage permissions for different user roles (Operators, Admins, and General Users). Priority: High

### 4.4.2 Stimulus/Response Sequences

1. The user navigates to the login page.
2. The system prompts for authentication (username & password).
3. If valid, the system assigns the correct user role.
4. The user is directed to their appropriate dashboard.

### 4.4.3 Functional Requirements

- REQ-1: Users must log in with unique credentials.
- REQ-2: The system must implement role-based access control (RBAC).
- REQ-3: Admins must approve new user registrations.
- REQ-4: Users must be able to reset their passwords securely.

## 4.5 AI/ML Anomaly Detection

### 4.5.1 Description and Priority

The system integrates machine learning (ML) models to detect abnormal temperature trends based on past data. The model flags potential issues before they escalate. Priority: Medium

### 4.5.2 Stimulus/Response Sequences

1. The ML model analyses incoming temperature data.

2. If a value is outside expected bounds, it flags an anomaly.
3. The system displays the flagged reading with an anomaly indicator.
4. Alerts are triggered if necessary.

#### 4.5.3 Functional Requirements

- REQ-1: The system must use a pre-trained ML model for anomaly detection.
- REQ-2: The model must compare real-time values against historical trends.
- REQ-3: The system must indicate anomalies visually on the dashboard.
- REQ-4: Users must be able to fine-tune the anomaly detection thresholds.

### 4.6 User Experience & Accessibility

#### 4.6.1 Description and Priority

The system must be user-friendly and accessible to all users, including those with disabilities.

Priority: Medium

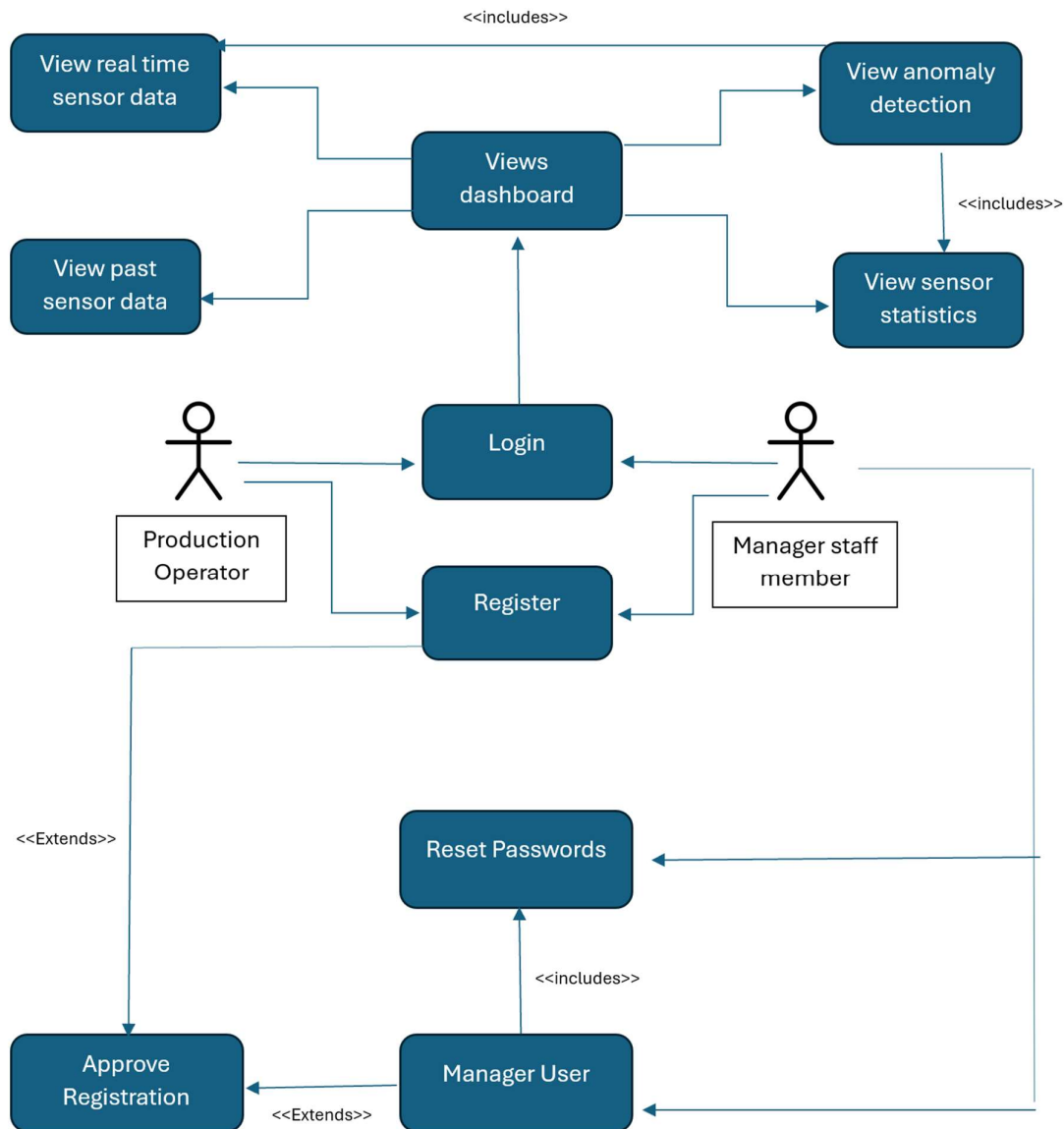
#### 4.6.2 Stimulus/Response Sequences

1. The user accesses the settings menu.
2. The system displays customisation options (e.g., theme, font size).
3. The user enables or customises accessibility features.
4. The system updates the interface accordingly.

#### 4.6.3 Functional Requirements

- REQ-1: The system must support light and dark modes.
- REQ-2: The system must provide accessibility options (dyslexic font, high contrast mode).
- REQ-3: Users must be able to customise alert sounds.
- REQ-4: The system must have mobile-friendly responsiveness

## 4.7 Use Case Diagram



#### 4.7.1 User

The Operator is responsible for monitoring real-time sensor data, acknowledging alerts, and analysing historical data to ensure smooth production operations. They have access to relevant dashboards but cannot modify system settings.

- View real-time sensor data from assigned production lines.
- Monitor historical trends to identify potential issues.
- Receive and acknowledge alerts based on the traffic-light system.
- Add comments or annotations to sensor data for documentation.
- Generate and download reports containing temperature and alert history.

- Update their login credentials for security purposes.
- Log in securely to access their assigned data and functionalities.

Operators interact with the dashboard to monitor data, acknowledge alerts, and review historical records. They are limited to viewing assigned sensors and cannot modify system configurations.

#### 4.7.2 Admin

The Admin holds the highest level of system access. They manage user accounts, system settings, and ensure the platform functions correctly. Admins also configure thresholds for anomaly detection and oversee security measures.

- Access all system functionalities, including user and sensor data management.
- Create, update, and delete user accounts, assigning roles as needed.
- Configure alert thresholds and system-wide parameters.
- Reset or modify passwords for operators when required.
- Oversee database maintenance, including archiving outdated sensor data.
- Ensure system security, compliance, and performance monitoring.

Admins interact with the user management interface, system settings, and alert configurations to oversee operations and maintain system integrity. They have full control over the system's parameters and user access rights.

## 5. Non-functional Requirements

### 5.1 Performance Requirements

- Dashboard updates every 5 seconds.
- API response time should be <1 second.

### 5.2 Security Requirements

- Role-based access control (RBAC)
- Encrypted user credentials

### 5.3 Scalability & Maintainability

- Designed to accommodate more sensors in the future.
- Modular development to support new features.



## 6. Legal, Social, Ethical, and Professional Issues

### 6.1 Legal

#### 6.1.1 Data Protection

The system must comply with the UK General Data Protection Regulation (UK GDPR) and Data Protection Act 2018 as it involves processing sensor data and user credentials. This includes:

- Lawful Data Processing – The system must only collect and process sensor and user data for legitimate purposes, such as temperature monitoring and alert notifications.
- User Consent and Transparency – Users must be informed about how their data is used and stored.
- Data Retention Policies – There must be clear guidelines on how long sensor logs and user data are stored before deletion to prevent unnecessary data retention.

#### 6.1.2 Industry and Regulatory Compliance

Since Rakusens operates within the food manufacturing industry, the system must comply with relevant regulations, including:

- Food Safety Standards – Ensuring proper temperature monitoring to maintain food safety standards.
- Cybersecurity Standards – Following the UK National Cyber Security Centre (NCSC) guidelines for data encryption and network security.
- Regular Audits – The system should support audit logs and compliance checks to ensure data integrity and accountability.

#### 6.1.3 Contractual Agreements and Intellectual Property

- Software Ownership – Clear definitions of ownership rights regarding the source code, ensuring no disputes over system control.
- Third-Party Integration – Any external services used (e.g., machine learning models, cloud storage providers) must comply with licensing agreements and security policies.

### 6.2 Social

#### 6.2.1 Fair Workload Distribution

The system should ensure an even distribution of monitoring responsibilities across operators, preventing overburdening specific employees.

#### 6.2.2 Work-Life Balance

- The system should avoid excessive notifications or monitoring, ensuring operators are not overwhelmed by alert fatigue.

- Reasonable thresholds should be set to prevent unnecessary alerts that could contribute to stress.

### 6.2.3 User Experience and Accessibility

- Stress-Free Interface – The dashboard should be intuitive, with a clear layout that avoids frustration.
- Accessibility Compliance – The system must include features such as high-contrast mode and screen reader compatibility to ensure usability for employees with disabilities.

## 6.3 Ethical

### 6.3.1 Data Ethics and Accountability

- Audit Logs – All system activities, including sensor adjustments and anomaly detections, should be logged for transparency and accountability.
- Data Minimisation – The system should only collect essential information required for monitoring and reporting, avoiding unnecessary personal data collection.
- Data Ownership – Users should have the right to access, redact, or delete their personal data within legal and organisational guidelines.

### 6.3.2 Employee Wellbeing and Communication

- Balanced Monitoring – While tracking sensor data is critical, excessive surveillance of employees should be avoided to ensure trust and a positive work environment.
- Respectful Communication – Alerts and notifications should be neutral and professional, avoiding language that could be perceived as punitive.

### 6.3.3 Ethical Compliance

The system must align with corporate ethics and employment laws, including:

- UK Government Act 1996 – Ensuring fair working conditions.
- UK Equality Act 2010 – Preventing discrimination based on role, age, disability, or any other factor.
- Regular Ethical Audits – Conducting periodic ethics reviews and gathering employee feedback on system usability and impact.
- System and Policy Changes – Employees must be informed of any system updates that affect their workflow, privacy, or data security.

## 6.4 Professional

### 6.4.1 Clear Responsibilities and Assignments

- Defined Roles – Operators and Admins should have clearly defined responsibilities within the system.

- Professional Communication – All system notifications and reports should use concise and professional language to avoid misunderstandings.

#### 6.4.2 Administrative Responsibilities

- Fair and Professional Management – Admins must ensure that the system is operated without bias or conflicts of interest.
- System Oversight – Regular performance reviews should be conducted to maintain system efficiency and reliability.

### 7. Gantt Chart

Planning	
Design	
Development	
Assessment	

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Assigned To
Team Preference	21/01/2025							Whole Team
Project Selection		28/01/2025	Project Allocation					Whole Team
Client Interview			6/02/2025					Whole Team
Role Allocation and Requirements Confirmation			7/02/2025	11/02/2025				Whole Team
Database								
Database for Login and Registration Design								Areesha
Database for API/Sensor								Kian

Data Design								
Create Tables using SQL								Areesha , Kian
Login Page								
Page design using HTML and CSS								Mahad
Code with JS to connect to SQL and gather input details								Lina
Main Dashboard Page								
Page Design using HTML and CSS								Ehtasham, Mahad, Lina
Drop down menu to navigate								Ehtasham, Mahad
Historical Data Page								
Page Design using HTML and CSS								Ehtasham, Mahad
Alerts Page								
Page Design using HTML and CSS								Ehtasham, Mahad
API								
Created database connection file using Node JS								Kian
Created Node JS file to								Kian

import data into database								
Created History data JS file								Ehtasham
Created Anomaly JS file								Tayyib
Created Alerts JS file								Adam
Started on machine learning for anomaly detection								Tayyib, Adam
Peer Review							6/03/2025	Everyone
Meeting Minutes								Kian

## 8. Peer Review & Contributions

Each team member's contribution will be evaluated based on assigned tasks, achievements, and effort, using a grading scale of 1 to 10 (where 1 = minimal contribution, 10 = outstanding contribution).

Ehtasham Afzal – worked as the team leader, Ehtasham made sure that each member of the team was given an opportunity to include ideas and contribute to the project. He specialised on the front end of the project whilst contributing ideas to API. 10/10

Kian Biswas – as secretary Kian was an important role to the team as he voluntarily took charge of monitoring the teams meetings and observing all the roles in the team. He specialised in API working as the backbone of the API. 10/10

Areesha Qazi – was given the role to work solely on the SQL side of the project. This was vital to ensure that the data that was given to run with the project. 10/10

Mahad Nazar – worked on the front end of the project. Mahad specialised in the login ensuring security and worked with SQL. 10/10

Lina Khalid – as the team speaker, Lina's included her ideas of the project during teams meeting. She specialised mainly on the front end of the project. 10/10

Adam Rasheed – Adam solely worked on the machine learning of the project to ensure that that it would run with the rest of the project. He contributed to teams meetings. 10/10

Tayyib Khan – Tayyib was able to observe the whole team and give key ideas of how the project should be laid out. His main role was to help with the Anomaly detection/ML . 10/10