Colour Sorting Project

by

Group B38

A Project Report

NSBM Green University

March 2024

Acknowledgements

.

First and foremost, we would like to express our sincere gratitude to each member of our team for their commitment, diligence, and spirit of cooperation during the project. The project has been enhanced and successfully completed because of the diverse abilities and viewpoints of each team member.

Our sincere gratitude goes out to our project supervisor Mr. Chamindra Attanayake for all their great advice, help, and mentoring. Their knowledge, support, and helpful criticism have been crucial in guiding the project in the proper path and overcoming obstacles along the road.

We also recognize the support that our friends, family, and classmates have provided along the way, including their understanding, encouragement, and help.

We appreciate your participation in this fulfilling and enriching experience, everyone.

Abstract

The goal of the color sorting project was to increase sorting process efficiency using an Arduino IDE, an ESP32 board, OpenCV Python, Arduino Cloud, Internet of Things components, a website, and a MySQL database. The requirement to expedite sorting procedures while maintaining accuracy and speed was the issue that needed to be resolved. Creating a thorough sorting procedure with all required features and streamlining the system to increase productivity were among the goals.

The techniques involved integrating hardware and software components. These included configuring the ESP32 board for data communication, using the Arduino Cloud and Internet of Things components for remote monitoring and control, programming the Arduino IDE for actuator control, implementing OpenCV Python for color detection and image processing, and setting up a MySQL database for data storage and analysis. The webcam was used to collect the data, which was then instantly examined using image processing methods.

The successful deployment of a comprehensive sorting process, with effective color detection and classification, flawless conveyor belt operation, and real-time monitoring via the internet interface, were among the main outcomes. Throughput increased, mistakes were decreased, and sorting tasks were sorted more efficiently by the system.

By creating a color sorting system that is fully operational and capable of improving industrial process efficiency, the project met its goals. Restrictions such as the requirement for additional optimization and scalability were noted, and suggestions for upcoming improvements and uses were made.

Table of contents

Chapter 1

Introduction

Requiring accurate and productive sorting procedures is essential in today's fast-paced industrial environment. To create a complete color sorting system, our project makes use of contemporary technologies including the ESP32 board, Arduino IDE, OpenCV Python, Arduino Cloud, IoT components, a website, and a MySQL database. With features for data storage, real-time monitoring, and user interaction, the system seeks to improve sorting process efficiency.

Research on sorting systems today emphasizes how crucial it is to automate processes and integrate different technologies to maximize sorting efficiency. By utilizing an ESP32 board for data communication, an Arduino IDE for control logic, OpenCV Python for color detection, Arduino Cloud for remote monitoring, Internet of Things components for connectivity, a website for user interface, and a MySQL database for data storage, our project expands on this body of knowledge.

Our project's goals are to provide a full sorting procedure with all required features, such as data storage, user interaction, real-time monitoring, and remote access. Additionally, by simplifying procedures and lowering errors, we want to improve sorting efficiency.

We will go over the use of our color sorting system in this report, outlining the techniques employed, important outcomes attained, and conclusions reached. We will pay particular attention to the way the ESP32 board uploads sorted details to the MySQL database, the features offered by the Arduino Cloud, the website's user interaction features, and the procedure for publishing sorted details and adding comments. We will now give a summary of the report's remaining sections.

Chapter 2

Literature review

Considerable attention has been paid to the literature on color sorting procedures, especially when it comes to industrial automation and efficiency enhancement. Numerous topics have been the subject of studies, including data management, image processing methods, hardware integration, and user interfaces.

Numerous studies have investigated the integration of hardware control in sorting systems using microcontrollers such as the Arduino IDE and ESP32 board. The significance of real-time data processing and communication for effective operation has been emphasized by this research. Furthermore, it has been noted that using IoT components and cloud platforms like Arduino Cloud makes it possible to have remote monitoring and control capabilities.

OpenCV Python has been a well-liked option when it comes to image processing methods for color analysis and detection in sorting applications. Research has indicated that it is a useful tool for precisely recognizing and categorizing items according to their color attributes.

For the purposes of data storage, retrieval, and visualization, the integration of MySQL databases with web-based interfaces has been thoroughly investigated. These systems facilitate user engagement for data editing and analysis, and they also allow for the smooth handling of sorted data.

Even while earlier studies have significantly advanced the discipline, there are still certain issues and gaps that require attention. Many of the systems now in use concentrate on some aspects of the sorting process rather than offering a complete solution with all features. Furthermore, current systems do not prioritize user engagement or feedback mechanisms.

Theoretical Structure:

The principles of data management, image processing, industrial automation, and human-computer interaction are all included in the theoretical framework that directs this study.

Real-time data processing, database management, color detection algorithms, hardware-software integration, and user interface design are among the fundamental ideas. The MVC (Model-View-Controller) pattern and client-server architecture are two theoretical frameworks that are used to organize the system and promote effective communication between its components.

This project integrates many technologies and features into a single system to fulfill the requirement for a comprehensive approach to color sorting procedures. This project intends to contribute to the growth of industrial automation and efficiency improvement in sorting processes by addressing a gap in the literature and developing a comprehensive sorting solution with improved user interface and feedback mechanisms.

Chapter 3

Methodology

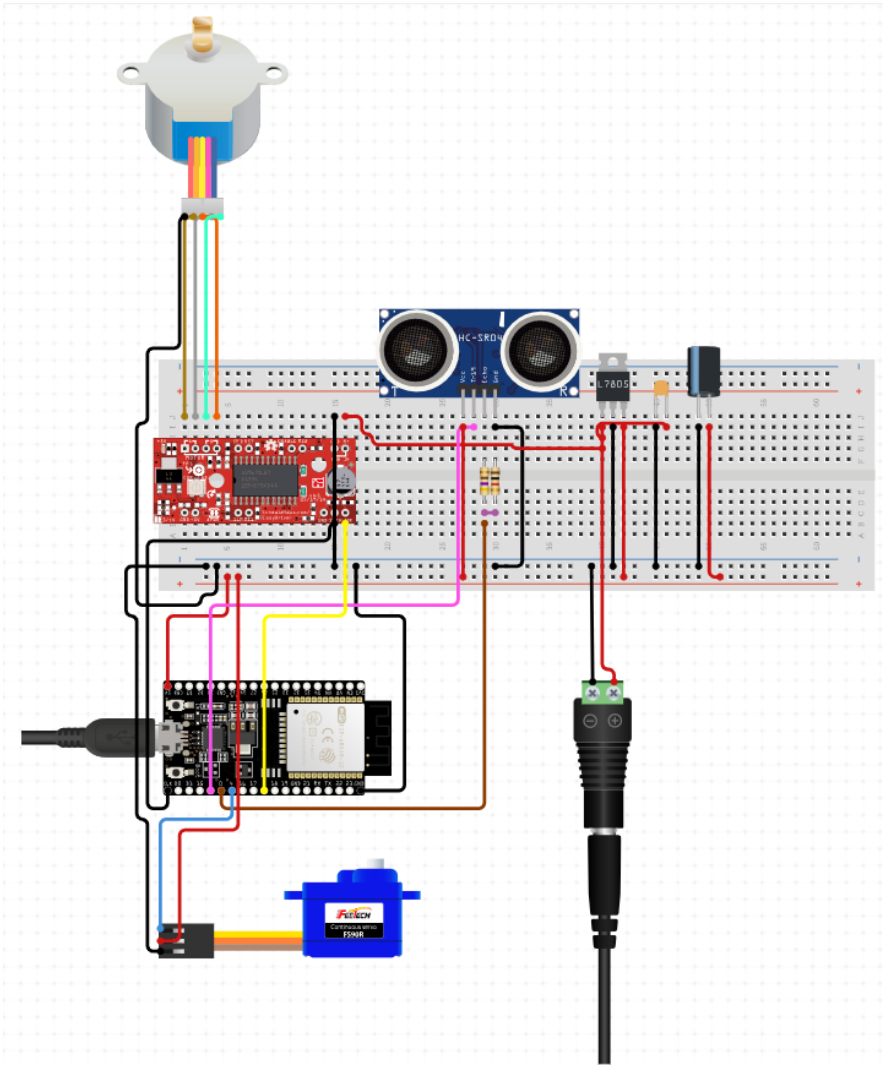
Techniques:

Using an ESP32 board, the Arduino IDE, OpenCV Python, Arduino Cloud, IoT components, a website, a MySQL database, and other technologies, the color sorting project's technique was centered on combining these technologies to create a thorough sorting system.

General Method:

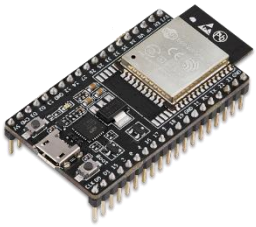
1.Hardware Configuration:

The MySQL database, actuators, and sensors could all be reached by the ESP32 board. The system included Internet of Things (IoT) components like actuators for sorting and sensors for color detection.



IOT components

1) ESP 32 Board



The ESP32 is a powerful microcontroller board known for its versatility and connectivity capabilities. It features a dual-core processor, Wi-Fi, Bluetooth, and ample GPIO pins, making it suitable for a wide range of IoT projects. It's popular for its ease of use and compatibility with various development environments.

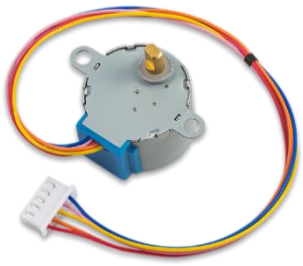
2) Ultrasonic Sensor

A device that measures distance by emitting ultrasonic sound waves and calculating the time it takes for the waves to bounce back after hitting an object. It's commonly used in robotics, industrial automation, and proximity sensing applications due to its accuracy and non-contact operation.

3) Servo Motor

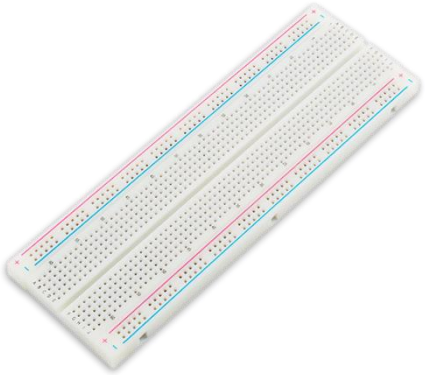
A rotary actuator that allows for precise control of angular position. It consists of a motor coupled with a feedback sensor and a control circuit. Servo motors are widely used in robotics, RC vehicles, and automation systems where accurate control of position, speed, and torque is required.

4) Stepper Motor



A stepper motor is a brushless DC motor that divides a full rotation into several equal steps. Unlike servo motors, stepper motors move in discrete steps rather than continuously rotating. They are commonly used in applications requiring precise position control, such as 3D printers, CNC machines, and robotic arms.

5) Breadboard



A breadboard is a prototyping tool used to create temporary circuits without the need for soldering. It consists of a grid of holes connected by metal strips beneath the surface. Components such as resistors, LEDs, and wires can be inserted into the holes to quickly build and test electronic circuits. Breadboards are commonly used by hobbyists, students, and engineers during the early stages of circuit design and experimentation.

6) Jumper wires

Jumper wires are flexible wires with connectors at each end, typically used to make temporary electrical connections between components on a breadboard or between various components in an electronic circuit.

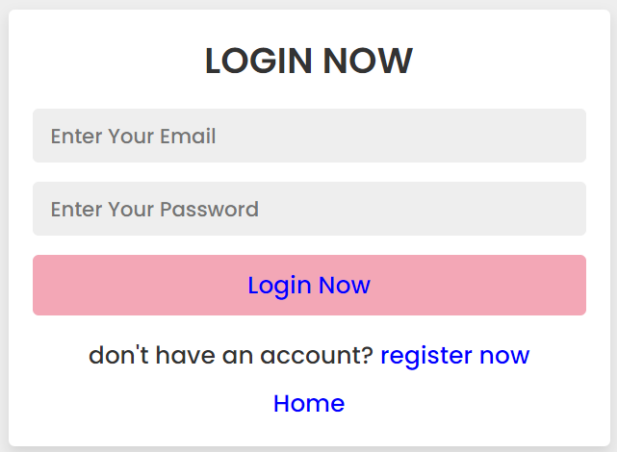
2.Software Engineering:

The ESP32 board was programmed for data collection, processing, and communication with other components using the Arduino IDE. The OpenCV Python library was utilized for image processing and color identification. Real-time sorting process monitoring and a display of the current counts of red, blue, green, and total sorted objects were made possible using Arduino Cloud.

3. Web Design:

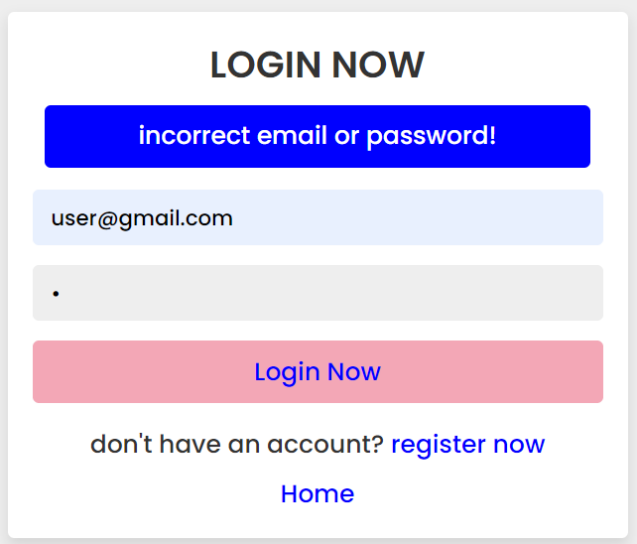
To house the MySQL database and offer a user interface for managing data sorting, a website was created.
Users can now change sorted data and add comments thanks to the implementation of user login and authorization methods.

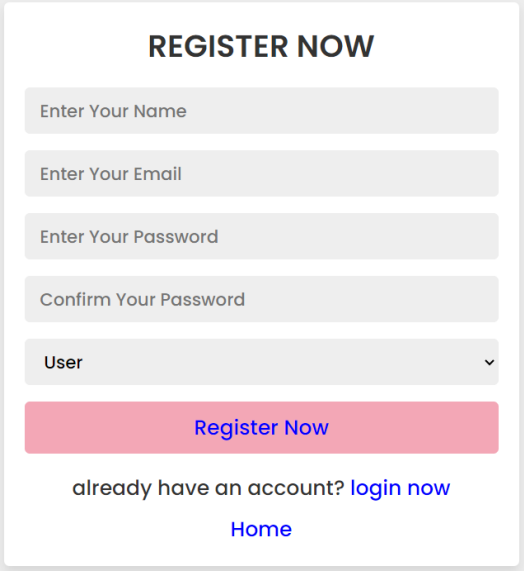
This is the index page of our official website. When the user clicks on “Login Here” it goes to Login Page.



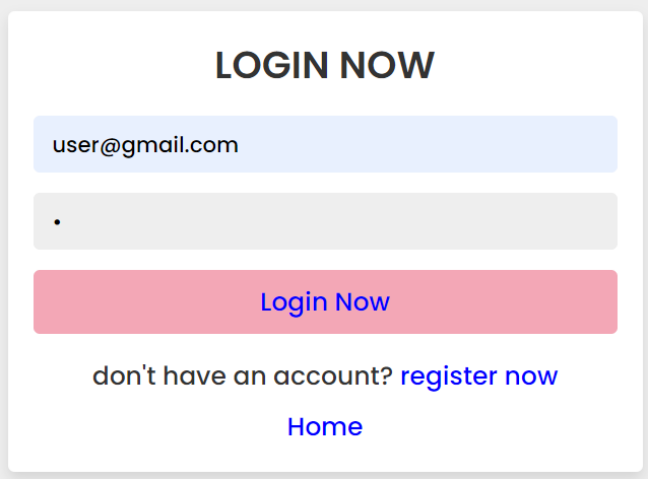
In the login page get email and password then it corrects it go to home page. In here check what is the user type if user type “user” goes to user home page else if user type “admin” goes to admin home page then finally check if user type “driver” got driver page. It depends on the user type that given in register.

When the user, admin or driver giving a wrong email or password then it shows an error message,



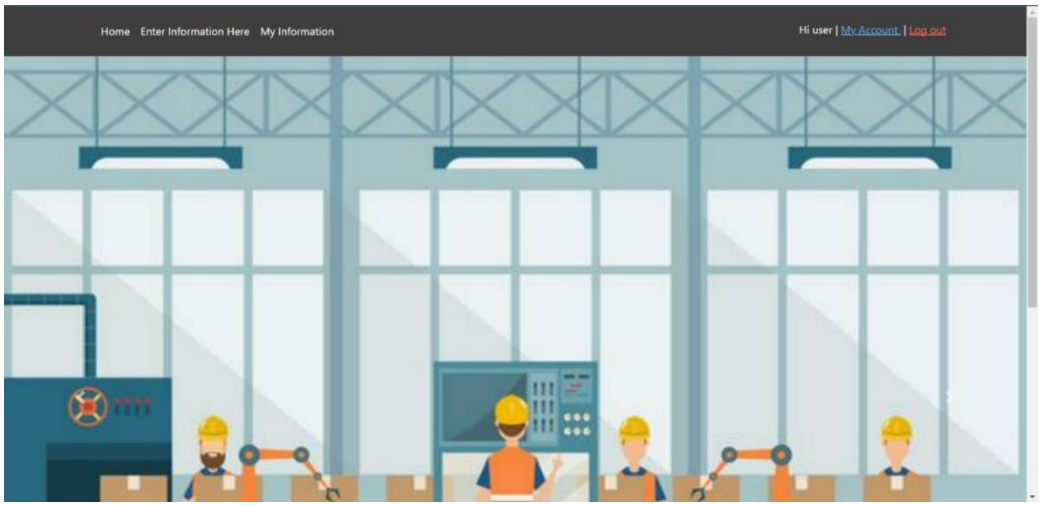


This is Register Page. Can register given that information. In user type can select one user type and depend on that goes to home pages.

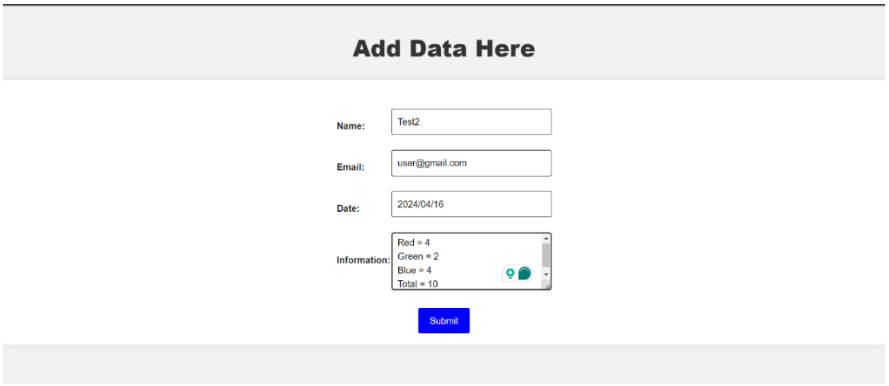


If the user type is “user” user can login given email and password then goes to user home page in user home page user shows two pages “Enter information here” and “My information” and in the right side in the navbar it shows greeting with username, user account and logout.

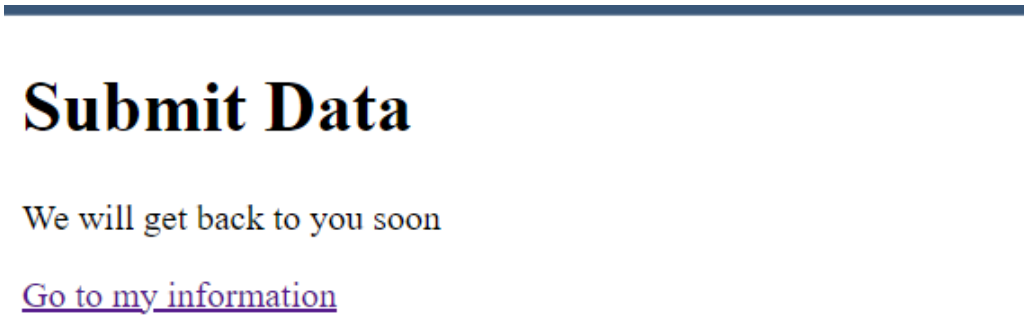
User home page



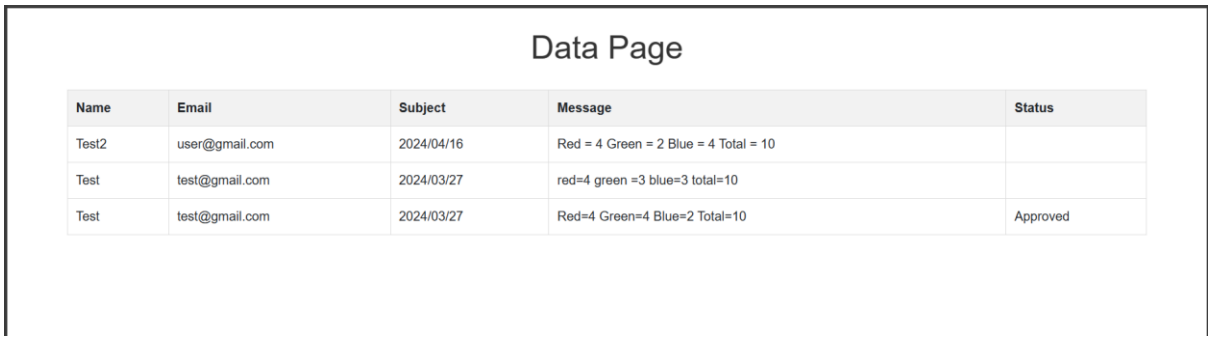
In “Enter information here” page, users can enter the data that get from manually. then click on submit it goes to submit page and show data is submitting correctly. This image an example data that sent to database.



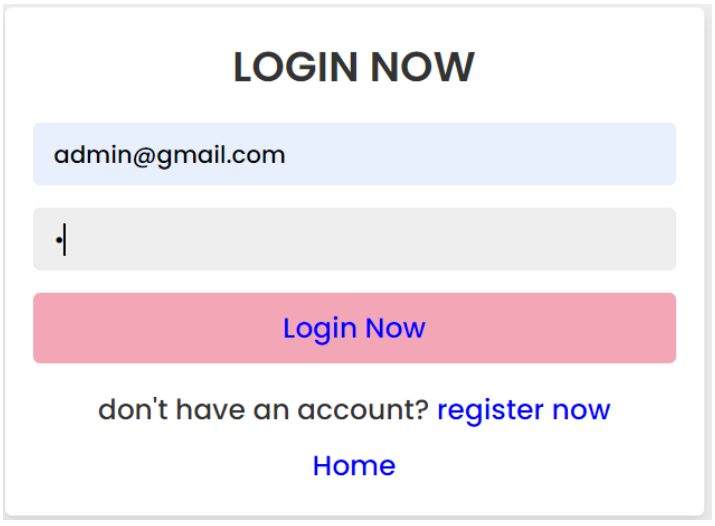
After submitting the data, user gets a confirmation.



Then click on “Go to my information” it comes to home page. then user can review his/her data in the My information page .it show like this

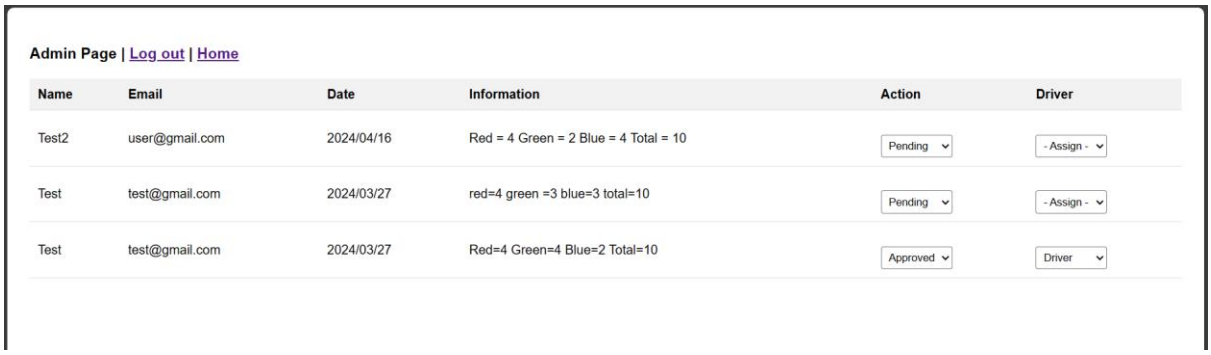


The example that put in it shows in the first in the table. its status is null because the admin didn’t show it. in here user submit a data then it goes to database and to admin. when the admin select it “Approved”, “Rejected” it show in null in here and the database to.



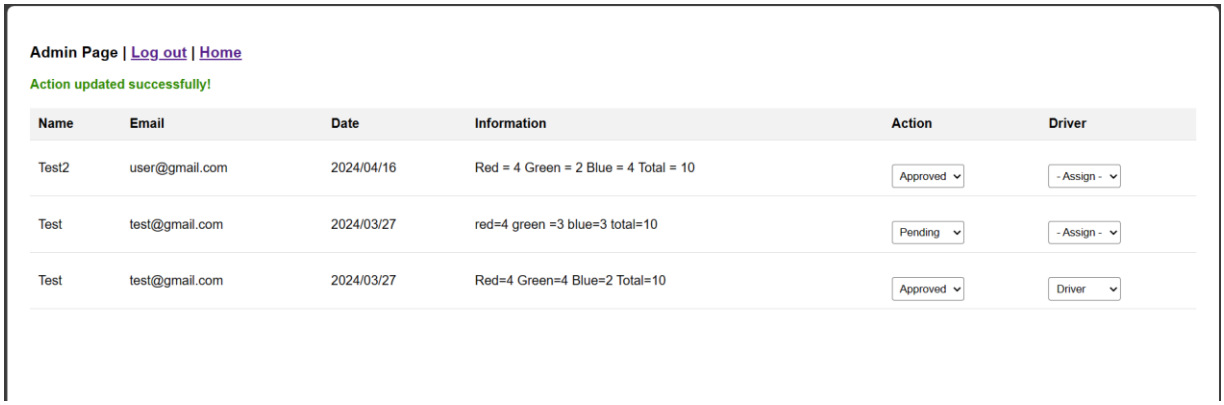
Then admin enter by using same login page. then admin goes to admin home page.it show like this,

This is the admin home page in here page call “Information”. And the right side is same in user home page. When admin click on that page it shows like this,

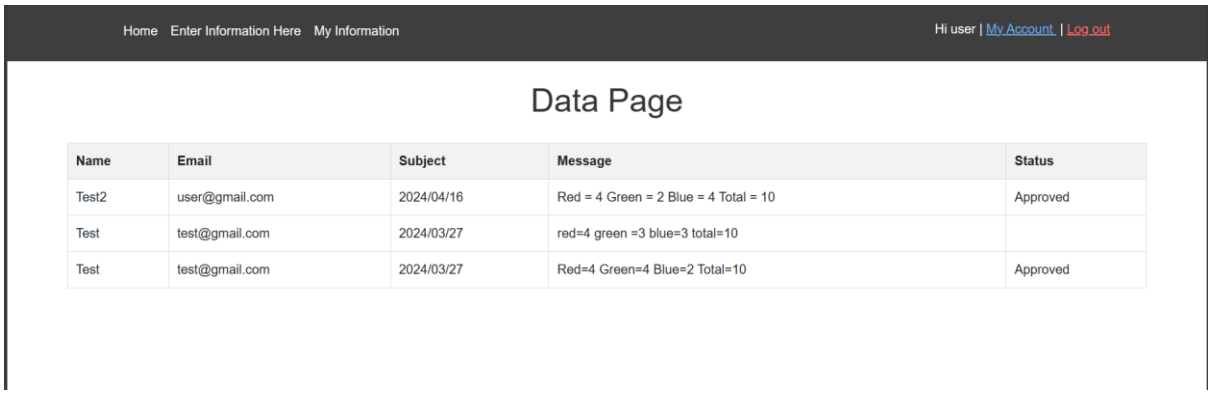


Then admin can check the values that are given by sorting system. The details in here and given from that system is correct admin can give “Approved” and assign a driver. Not can give “Rejected”. and inform the user that submits the data.

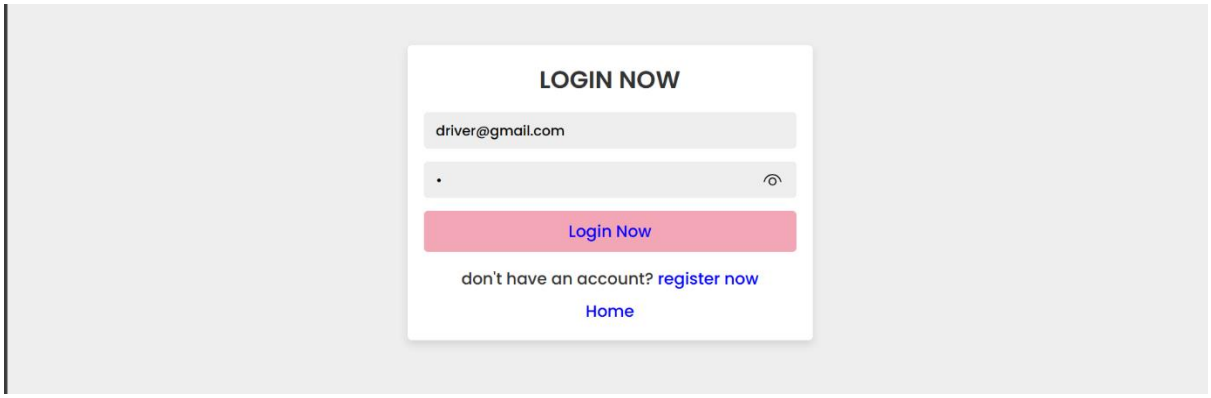
In this image show that,



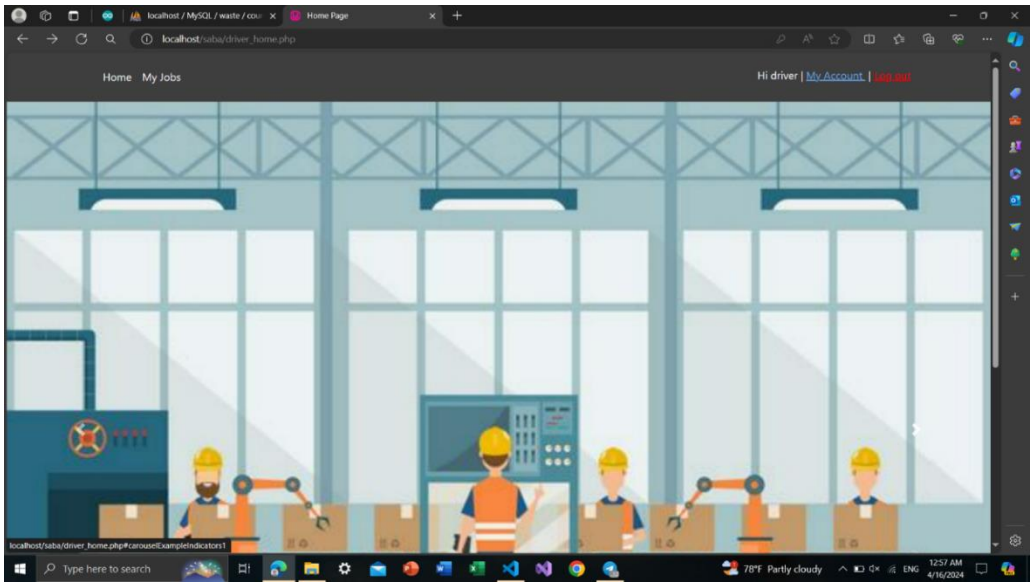
Then in user “My information” page show this this,



The admin assigns a driver can show it in his/her page

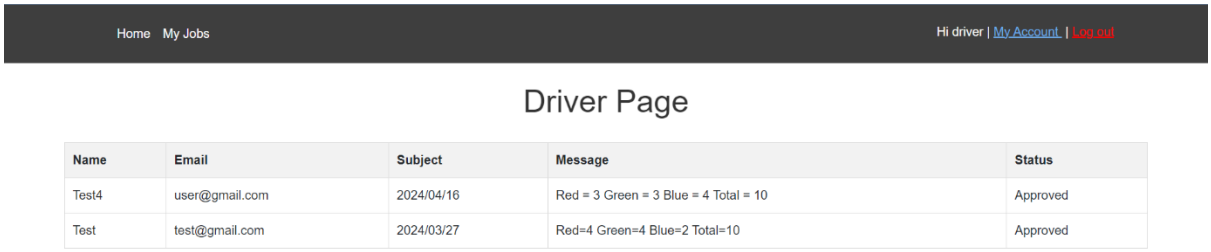


Driver can also login in same login page. Then user go to driver home page. It shows like this,



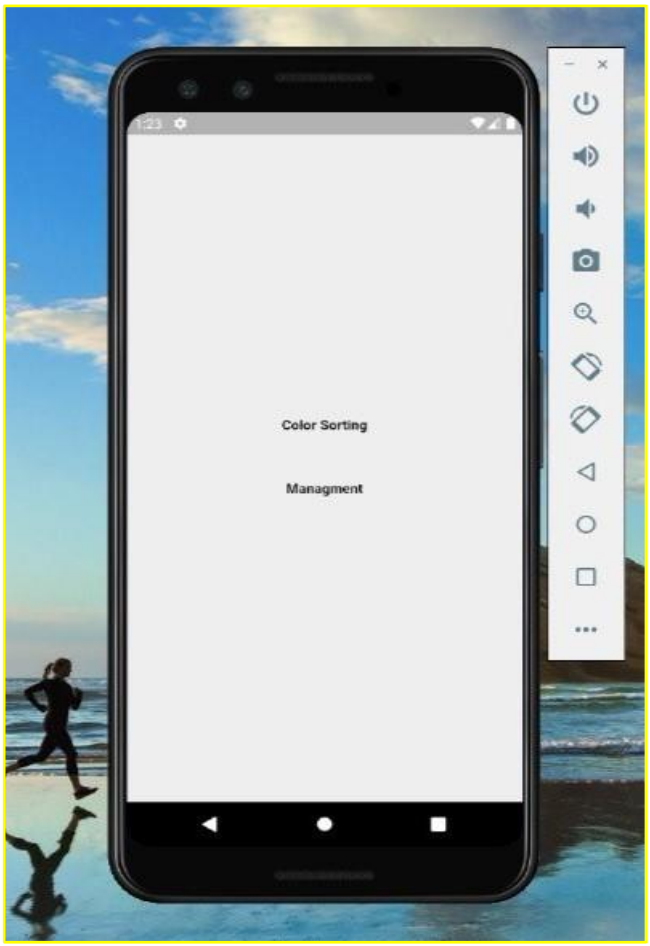
In driver home page there is a page call “My Jobs”. In my jobs page show the driver what job to assign and he/she can go there and get things and distribute.

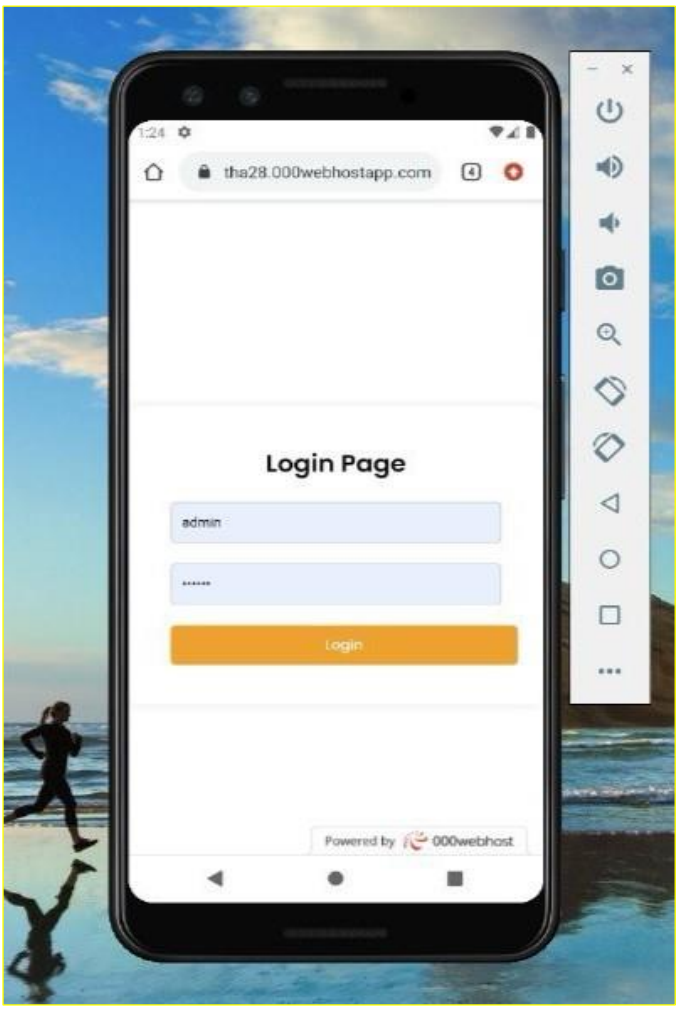
My Jobs page show like this

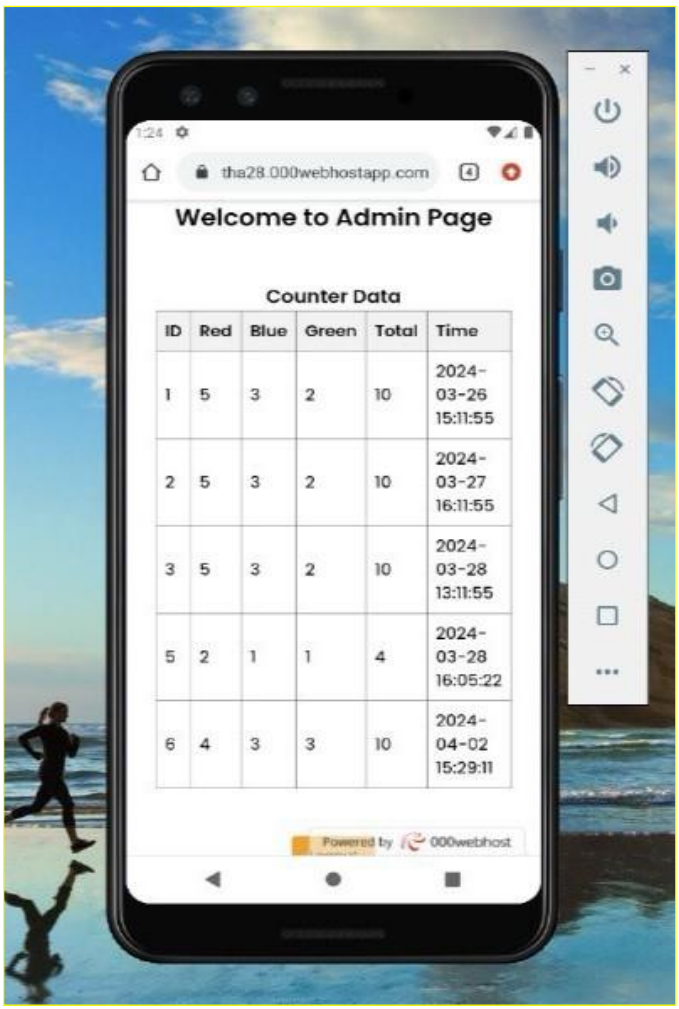


4. Mobile Application

The mobile application, developed using the Flutter framework, is a central gateway to our system. It provides quick and direct access to essential functionalities, making it easier for users to interact with the system. The application features two clickable links that simplify navigation. These links take users to the respective URLs with seamless ease, maximizing efficiency. With such an intuitive design, not only does the application simplify the user experience, but it also ensures accessibility to vital system features, optimizing usability and enhancing overall user satisfaction.







The website has been designed with a focus on user-friendliness. It features a smooth login process that enhances security and accessibility. The interface is clear and intuitive, allowing users to easily navigate through data displayed in a visually comprehensible table view. The platform leverages real-time tracking capabilities, ensuring dynamic monitoring of color sorting processes, which enhances efficiency and accuracy. The homepage seamlessly retrieves data from the database, facilitating streamlined access to relevant information. After completing their tasks, users can easily logout, ensuring a secure and user-centric experience throughout their interaction with the platform.

```
void sendDataToServer() {

String postData = "Red=" + String(Red) + "&Green=" + String(Green)
+"&Blue=" + String(Blue) + "&Total=" + String(Total);
```

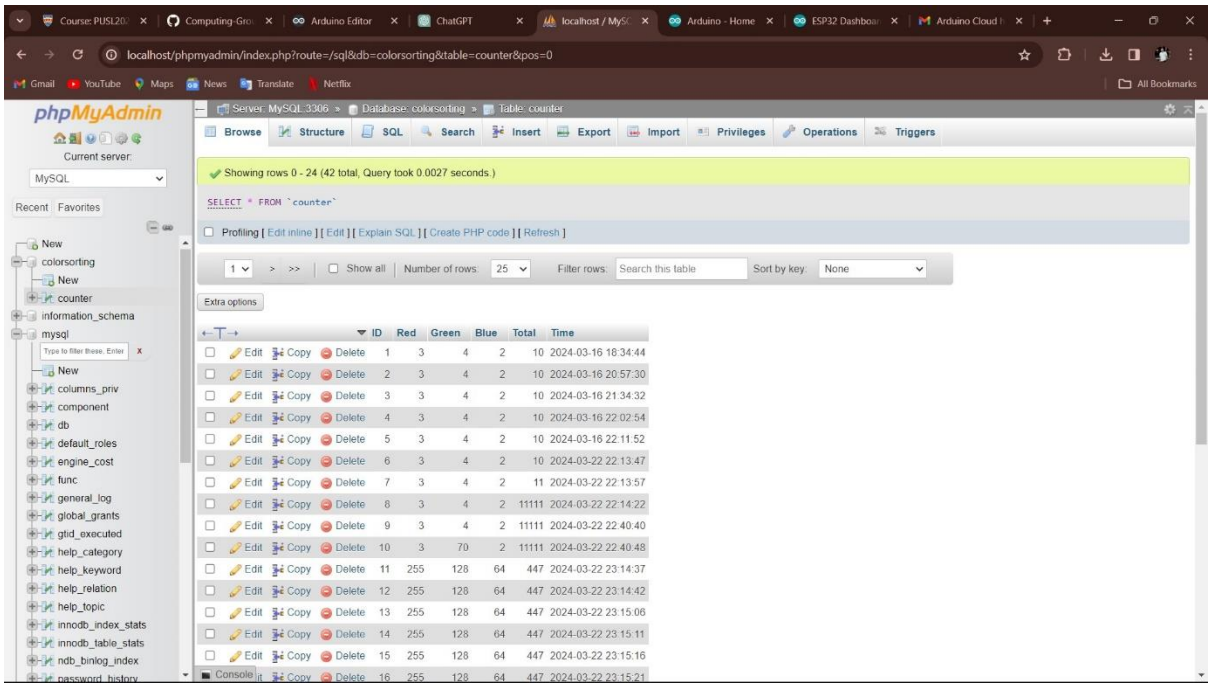
Send data to the server specified by the URL using an HTTP POST request. This data includes the values for Red, Green, and Blue, each converted to a string format, along with the total value.

```
<?php


$hostname = "localhost";

$username = "root";

$password = "";

$database = "colorsorting";


$conn = mysqli_connect($hostname, $username, $password, $database);


if (!$conn) {

	die("Connection failed: " . mysqli_connect_error());

}


echo "Database connection is OK<br>";


if(isset($_POST["Red"]) && isset($_POST["Green"])&&isset($_POST["Blue"])&&isset($_POST["Total"])) {


	$R = $_POST["Red"];

	$G = $_POST["Green"];

  $B = $_POST["Blue"];

  $T = $_POST["Total"];


	$sql = "INSERT INTO counter (Red, Green, Blue, Total) VALUES ('.$R.', '.$G.', '.$B.', '.$T.')";


	if (mysqli_query($conn, $sql)) {

		echo "\nNew record created successfully";

	} else {

		echo "Error: " . $sql . "<br>" . mysqli_error($conn);

	}

}


?>
```

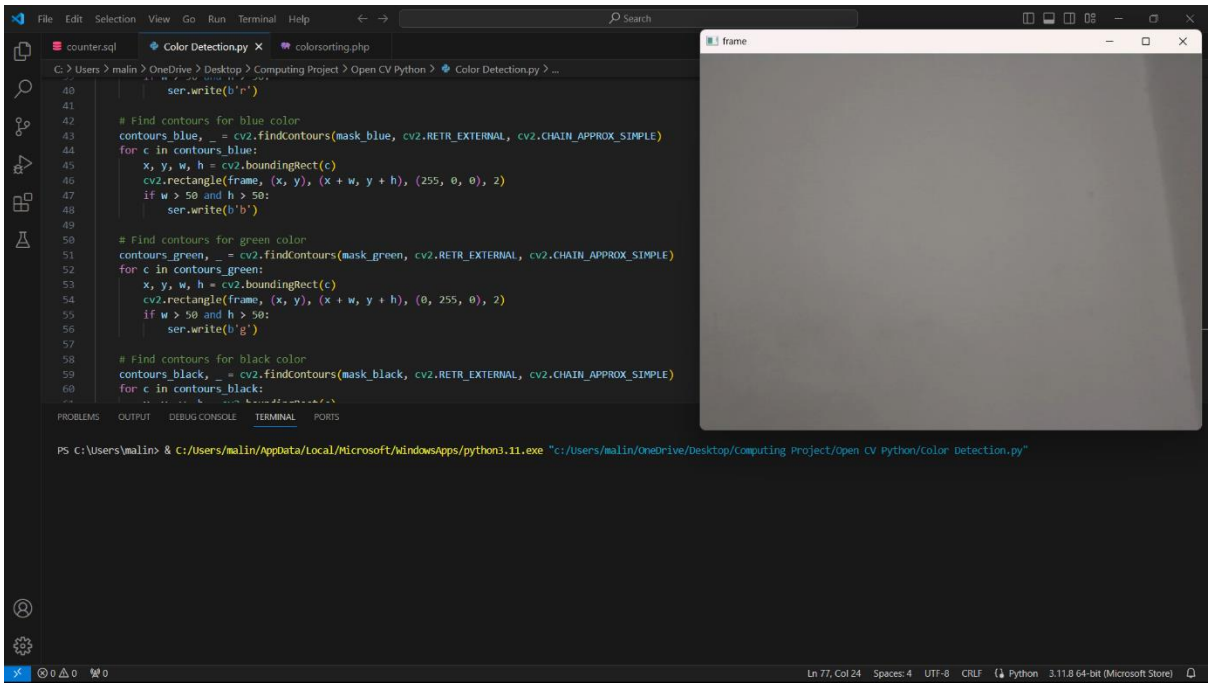
This PHP script serves as the go-between for the user interface and the MySQL database, making it easy to insert color sorting data. It receives POST requests with values for red, green, blue, and total, then securely connects to the database and constructs SQL queries to insert the data. The script has error handling mechanisms in place to provide useful feedback and ensure the insertion process runs smoothly. All in all, the script is essential to the color sorting system's functionality and reliability by managing data transmission and storage efficiently.

5. Database Administration:

Sorted data, including counts of red, blue, and green items as well as the total number of sorted items, were stored in a MySQL database. Users were able to upload sorted details, provide comments, and have the material reviewed and approved by administrators through integration with the website.



6. Open CV program:



This Python code utilizes OpenCV (cv2) and serial communication to detect and process colors in real-time using a webcam feed. It detects strong red, green, blue, and black colors. When a color is detected, it sends a corresponding signal ('r', 'g', 'b', 'k') to an Arduino board via serial communication. The Arduino can then perform actions based on the received signal, like controlling LEDs or motors. The code continuously captures frames from the webcam, applies color detection algorithms in HSV color space, identifies contours for each color, and sends signals to the Arduino accordingly. The program ends when the 'q' key is pressed.

Evaluation of Methods:

Obstacles: During the research, the following obstacles were faced:

smooth integration of various hardware and software components.
ensuring that the MySQL database, Arduino Cloud, website, and ESP32 board are all in real-time communication. putting in place reliable error-handling procedures to deal with unanticipated circumstances that arise during the sorting process.

Overcoming Obstacles:

To find and fix integration problems, testing and debugging were done often.
Reliability and system architecture were enhanced through prototyping and iterative development. The timely identification and correction of problems was made possible by the system's ongoing monitoring and logging of actions.

Effectiveness of Approach:

The whole strategy worked well to accomplish the project's goals of creating a comprehensive sorting procedure with increased effectiveness. The solution effectively expedited the sorting process and enabled data management by utilizing Internet of Things (IoT) technology, real-time monitoring capabilities, and an intuitive online interface.

Justification for Methods:

Utilizing an ESP32 board, the Arduino IDE, OpenCV Python, Arduino Cloud, Internet of Things components, a website, and a MySQL database offered a flexible and expandable framework for the sorting system's implementation. These technologies were selected because they met the project requirements and were flexible, compatible, and appropriate. Their integration made it possible for effective data processing, smooth user interface, and communication, all of which helped the project succeed.

❖ User requirement

Identification of users

As this protype will be used in industrial level machines, the business owners can integrate this machine into the industry to increase the effectiveness and accuracy of the sorting process. Also, for the business wise by implementing this technology, owners can reduce the employment salary expenses as machines can be used to do the work done by the employees. The profit margins can be increased due to the effectiveness added to the industry.

For the employees, the workload is reduced as the process is automated using the machines. A user can observe the process. In an emergency, the user can pause the sorting process. Also, if there are miss separations in the sorting process, the user can edit the count of the object. Mainly this project can be used in logistic companies, fruit exports, and airports.

User interviews

For the fact gathering process, we mainly used internet-based information. Forget the ideas and expectations about the project, we gathered video demonstrations about the sorting process. By referencing the information, we noted the main user requirements that should be fulfilled by the project. The colour detection and sorting should be the main objective of the project as the counting and detail viewing will be secondary options.

Persona Development

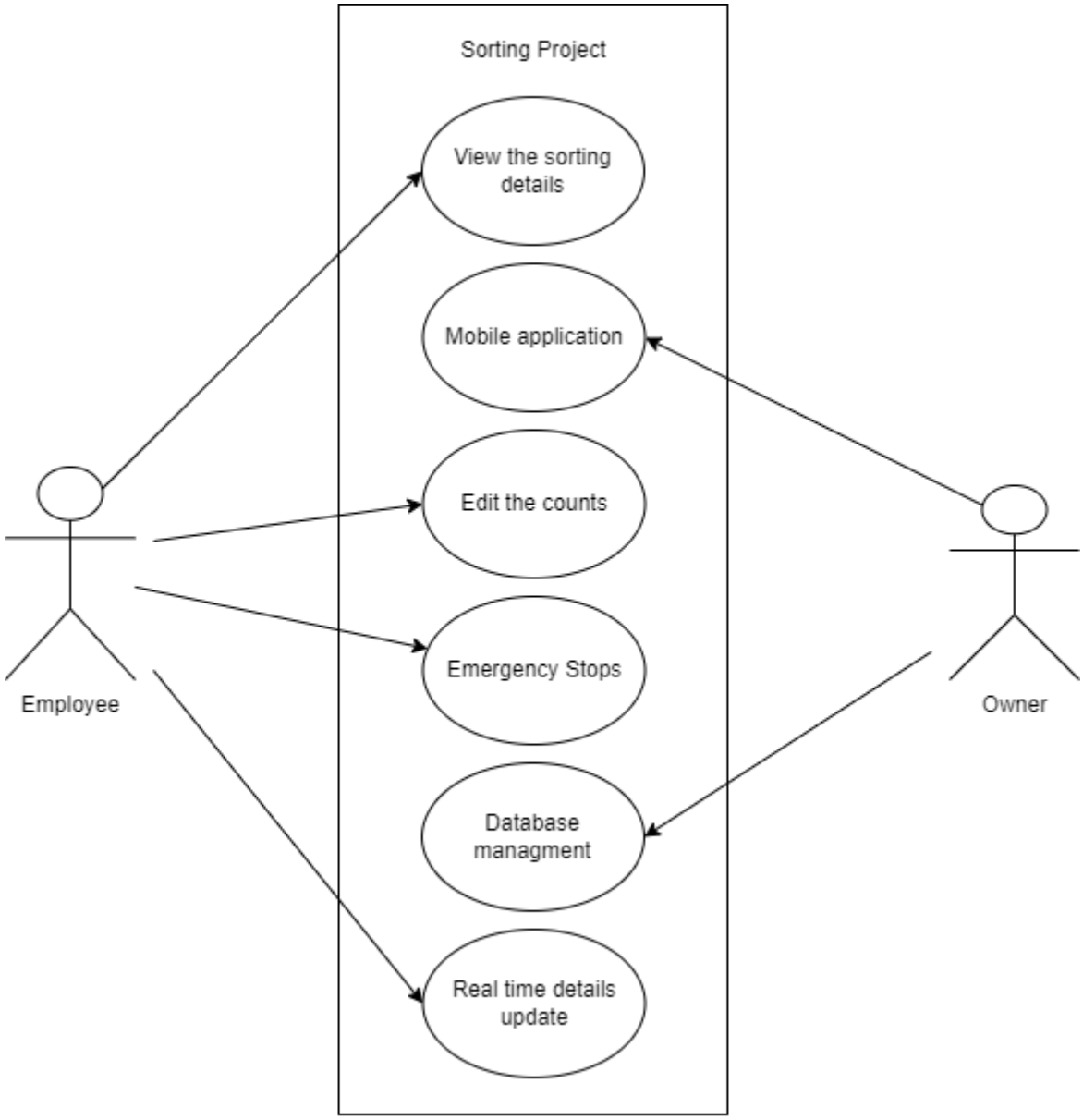
The main personnel that can be attracted to this project can be business owners with industrial level companies. To gain the attraction of the owners, we have come up with an idea of introducing the project to the owners by presenting the project as gathering their requirements and adapting and making changes to the project.

- Maximize Output and Optimize Production Efficiency: Main objective is to maximize output and uphold high standards of quality.
- Introduce Innovative Solutions: To increase sorting accuracy and expedite the manufacturing process, it is keen to introduce innovative solutions like the IoT color sorting project.
- Maintain Quality Control: dedicating to making sure that all products satisfy legal and quality standards, and she views the IoT colour sorting project to improve quality control procedures.

Challenges:

- Employee Training: understanding how crucial it is to give staff members who will be using and maintaining the IoT system proper training.
- Budget Restrictions: Been aware of the financial limitations and will have to convince higher management of the value of the IoT color sorting project.
- User-Friendly Interface: User requires an interface for the system that is simple to use so that they can monitor, control, and modify the sorting parameters.

Use case analysis.



Requirements Prioritizations

- Accurate color identification is the main need for efficiently classifying objects according to their color characteristics. To minimize sorting errors and meet quality standards, this criterion is essential.
- Real-Time Sorting: As items travel along the conveyor belt, the system should be able to sort them in real-time. Maintaining manufacturing efficiency and throughput requires real-time sorting.
- Scalability: To handle changing production quantities and potential future growth, the system must be scalable. Scalability guarantees that the system can accommodate growing demand without compromising dependability or performance.
- connection with Current Systems: Streamlining operations and data flow throughout the company depends on a seamless connection with current manufacturing systems, such as production control and inventory management systems.
- User Interface Design: To monitor sorting processes, modify parameters, and resolve problems, operators and administrators need an intuitive interface. User adoption and productivity are increased by intuitive design and simplicity of usage.
- Stability and Reliability: There should be little downtime or production interruptions in the system. In addition to avoiding expensive delays or interruptions in the manufacturing process, reliability guarantees consistent sorting performance.
- Flexibility in Handling Various Object Sizes and Shapes: The system must have the capacity to handle objects with a range of dimensions, forms, and compositions. Sorting criteria that are flexible can handle a wide range of products or components and provide production adaptability.
- Cost-Effectiveness: The system should be economical in terms of original investment, maintenance costs, and operating expenses while guaranteeing high-quality performance. Cost-effectiveness fits within budgetary restrictions and optimizes return on investment.
- Training and assistance: For effective implementation and user adoption, operators and maintenance staff must receive thorough training as well as continuous assistance. Training guarantees that users can optimize the system's capabilities, troubleshoot problems, and run it successfully.

Functional requirements

- Color Identification:
  The sorting procedure requires the system to precisely identify and distinguish between various colors on items.
  It should be able to accurately discriminate between a large variety of hues and tones.
- Sorting in real time:
  As items go along the sorting line or conveyor belt, the system should be able to sort them in real time.
  For production throughput to be maintained, sorting decisions need to be made fast and effectively.
  Configuration of Sorting Criteria:
- Sorting criteria ought to be configurable by operators according to certain color properties and sorting rules.
  It should be possible to modify and customize parameters like acceptable color ranges, thresholds, and sorting destinations.
- Flexibility Regarding Object Properties:
  In order to retain correct color recognition, the system should be able to adjust to changes in object sizes, shapes, and surface textures. It must be able to handle items with varying finishes and materials without sacrificing the precision of its sorting.
- Mechanisms for Feedback Control:
  In order to dynamically modify sorting parameters in response to system feedback and real-time sensor data, the system should have feedback control mechanisms.
  This guarantees that the judgments made during sorting are optimal and adaptable to modifications in the surroundings or attributes of the objects.
- Combining Actuation Systems Integration:
  To physically separate sorted objects, the sorting system must work in unison with actuation systems like diverters, robotic arms, or conveyor belts.
  Precise control over item routing and support for many actuation mechanism types should be provided by integration.
- Exception Management and Error Handling:
  Errors or exceptions that arise during sorting, such as incorrectly classified objects or sorting failures, should be detected, and handled by the system.
  It should have systems in place for notifying operators of problems, recording mistakes, and taking immediate corrective action.
- Remote Management and Entry:
  Via a secure network connection, operators should be able to view and manage the sorting system remotely.
  Remote access increases operational flexibility and efficiency by making it easier to monitor and troubleshoot the system from any place.
- Expandability & Scalability:
  In order to support future growth or changes, such adding more sorting stations or connecting with more sensors, the system architecture should be scalable.
  Modular parts and adjustable configurations should be supported so that it can adjust to changing manufacturing needs.

Non-functional requirements

- Performance:
  At least 50 objects should be able to be sorted by the system per minute.
- Dependability: At least 10 hours should elapse between system failures.
  In fewer than one minute, it should be able to automatically recover from malfunctions or faults without the need for human involvement.
- Availability: During regular business hours, the system needs to maintain an uptime of at least 90%.
  Less than 2 hours of maintenance and planned downtime should occur each month.
- Scalability
  To accommodate a minimum 30% increase in production volume without experiencing a decrease in performance, the system ought to scale horizontally. It ought to allow for the addition of more sensors or sorting stations without requiring a large amount of modification.
- Usability:
  Operators should need little training to operate the user interface efficiently since it should be simple to use and intuitive. System's alerts and error messages should be easy to read and comprehend to facilitate troubleshooting and problem solving.
- Sustainability:
  Modularity and encapsulation in the system design will make maintenance and future upgrades easier.
  The provision of diagnostic tools and logging capabilities is necessary to help maintenance staff locate and fix problems.
- Compatibility:
  A large variety of object types, materials, and sizes that are frequently encountered in the manufacturing process should be compatible with the system.
  It ought to facilitate integration with current manufacturing systems, including Manufacturing Execution Systems (MES) and PLCs (Programmable Logic Controllers).
- Environmental Factors to Be Considered
  The system's components ought to be built to function in a manufacturing environment's typical temperature, humidity, and other environmental constraints.
  Energy-saving techniques ought to be used to reduce electricity usage and its negative effects on the environment.

❖ Functional specification

- Architecture of the System:
    - Describe the necessary hardware, such as the color sensors, controllers, actuators (such as conveyor belts), and communication modules.
    - Describe the software's user interface, control algorithms, data processing modules, and integration with cloud or IoT systems.
    - High Importance level.

- Color Identification:
    - Specify the color detecting technology that will be utilized.
    - Provide specific calibration steps to guarantee precise color recognition in a variety of lighting scenarios.
    - Explain the color space representation and the feature extraction and color analysis techniques.
    - High Importance level.

- Mechanism for Sorting Objects:
    - Describe the actuation mechanism that sorts things according to color characteristics.
    - Define feedback control techniques to dynamically modify sorting parameters according to current sensor data.
    - Indicate the destinations and sorting criteria for various item colors or categories.
    - Medium importance level.

- Sorting in Real Time:
    - Indicate what is needed to sort items as they travel over the conveyor belt in real time.
    - Specify the latency needed to recognize colors and make sorting judgments.
    - High Importance level.

- Designing User Interfaces:
    - Describe the operator and administrator interface that allows them to examine system status, change parameters, and monitor sorting activities.
    - Describe the user interface's design concepts, such as responsiveness, clarity, and ease of use.
    - Low importance level.

- Combining with Current Systems:
    - Describe the prerequisites for a smooth integration with the production control and inventory management systems that are currently in place in the manufacturing industry.
    - Indicate the APIs for integration, communication protocols, and data formats.

- Instruction and Assistance:
    - Describe the conditions under which operators and maintenance staff must be trained in system operation, troubleshooting, and maintenance techniques. To handle any problems or downtime promptly and effectively, specify support and maintenance services.

- Adherence to Regulations:

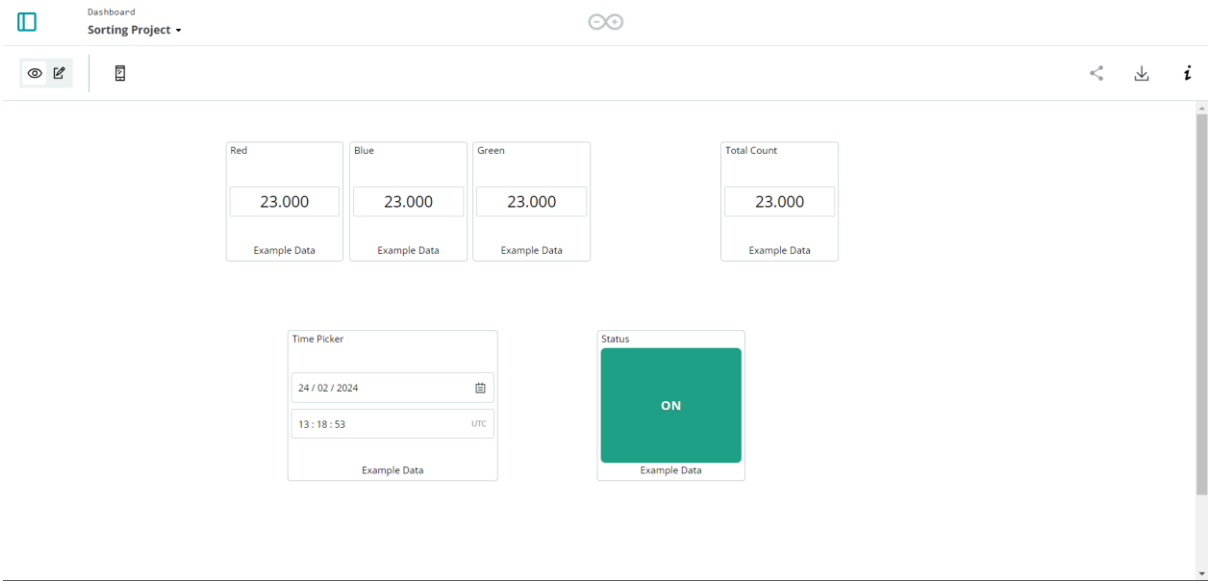
    - Indicate what must be done to ensure adherence to industry norms and laws, such as those pertaining to data protection, the environment, and safety.
    - Specify certification and testing processes to guarantee adherence to relevant standards.
    - Low importance level.

❖ Technical Specification

We are developing a color sorting Arduino project designed to sort the object according to the color. The main technology that is used is Arduino and Open CV. The user can view the number of sorted objects as well as edit the misplaced objects. Users can also have a mobile application to view the details of the object's quantity after the sorting process.

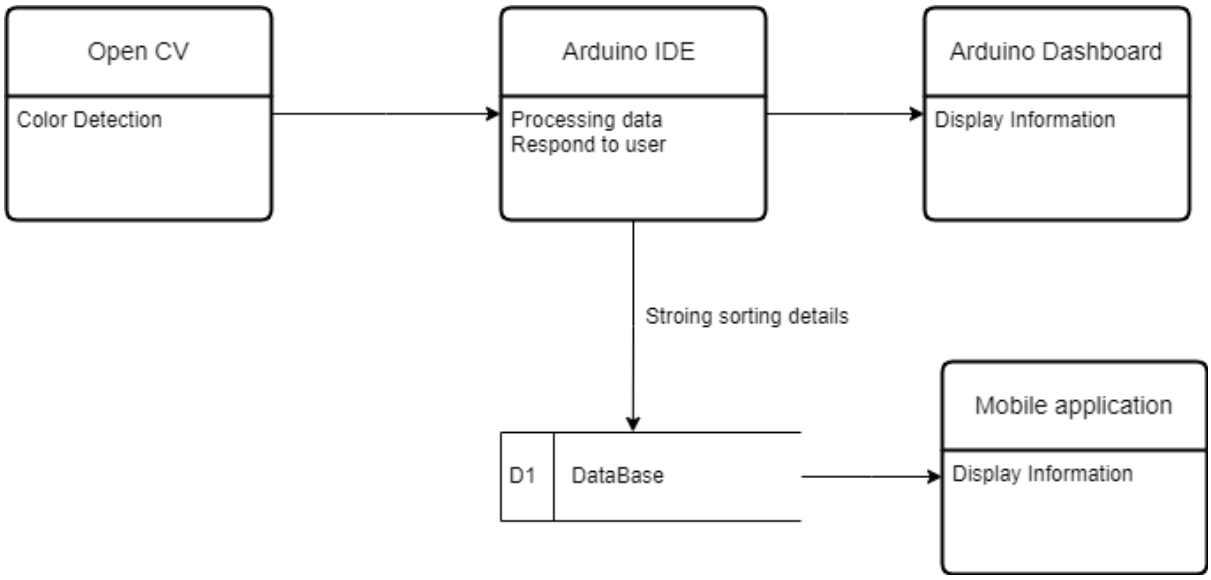
User Interface Design

The main user interface that the user will interact with is Arduino dashboard. This dashboard will display the number of objects sorted in the process.

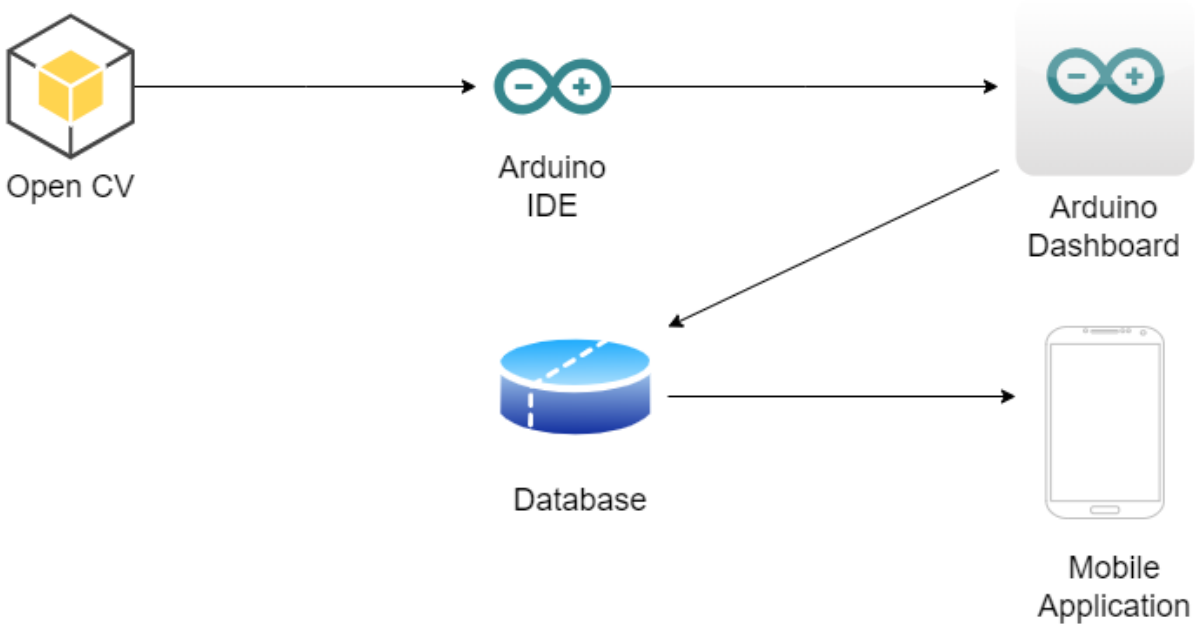


The color detection will be done by the Open CV software. Mainly red, green, and blue colors will be detected and sorted through the software. Then the output will be sent to the Arduino IDE. The Arduino code will be triggered for the respective color of the object. The past sorting details will be saved in a database that can be retrieved in a mobile application.

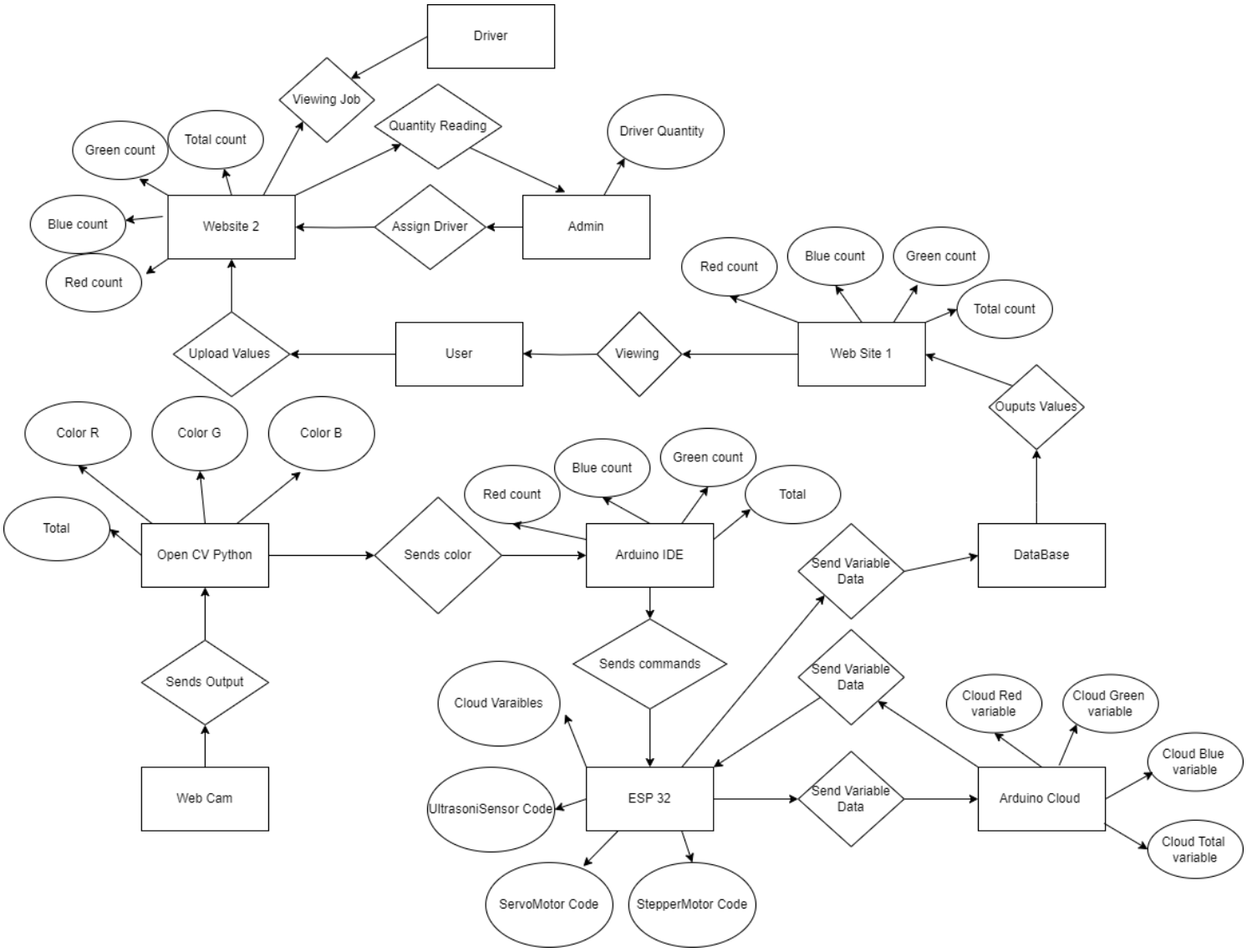
Data Model



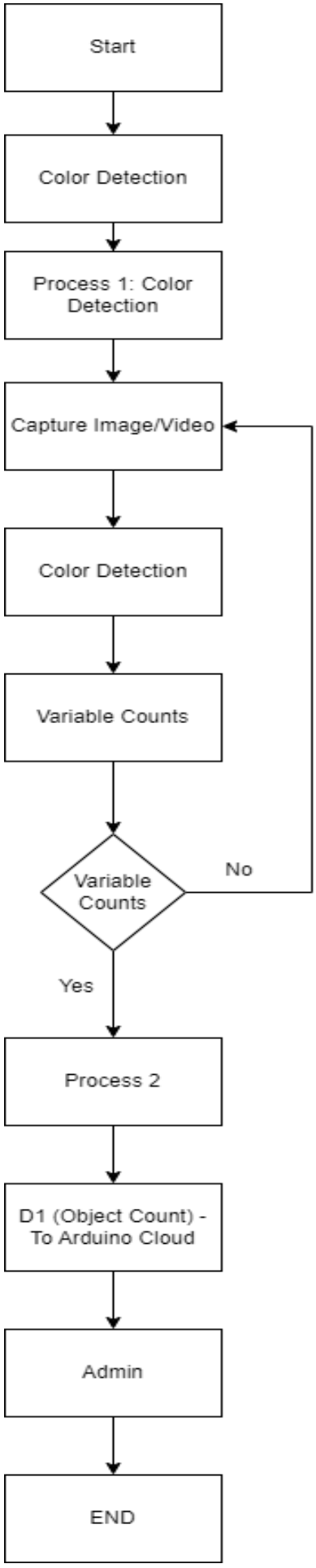
System Architecture



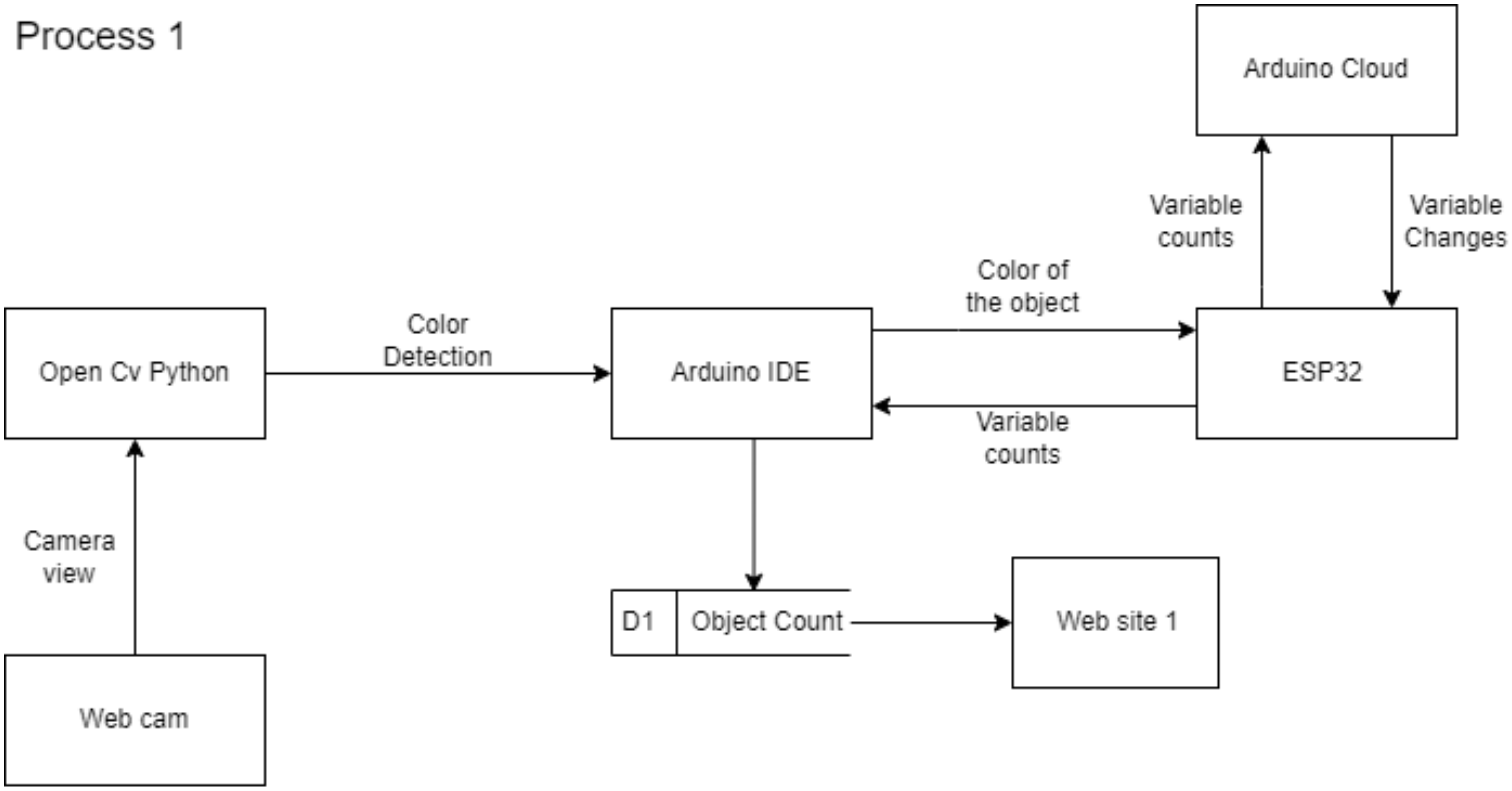
ER Diagram

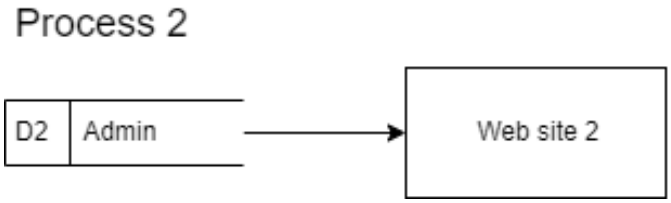

Driver
Viewing Job
Quantity Reading
Driver Quantity
Green count
Total count
Blue count
Website 2
Assign Driver
Admin
Red count
Red count
Blue count
Green count
Total count
Upload Values
User
Viewing
Web Site 1
Ouputs Values
Color R
Color G
Color B
Total
Open CV Python
Sends color
Red count
Blue count
Green count
Total
Arduino IDE
Send Variable Data
DataBase
Sends Output
Sends commands
Send Variable Data
Cloud Varaibles
Cloud Red variable
Cloud Green variable
Cloud Blue variable
Web Cam
UltrasoniSensor Code
ESP 32
Send Variable Data
Arduino Cloud
Cloud Total variable
ServoMotor Code
StepperMotor Code

Flow chart Diagram.



Data Flow Diagram





Deployment and Infrastructure

All the components will be deployed on an Arduino uno board and a breadboard. The Arduino board will be connected to a device to display the current sorting details to the user. The sorting will be done on a conveyor belt, while the objects are moved through the belt. The camera will detect the color and the servo motor will direct the object to the respective color bucket. After the sorting, the sorting details will be saved on to the database. User can retrieve the sorting data through the mobile application.

Testing Strategy

The sorting process can be tested by a sorting round. If the object is placed correctly according to the color, the accuracy of the project can be determined. Users can stop the conveyor belt in an emergency and start again. The current sorting details can be read on the Arduino dashboard. We can test the accuracy of the reading by checking the details of the sorted object with the actual sorted quantity of the objects.

Dependencies

For the application, Arduino IDE is used with respective libraries called stepper, Servo, and ultrasonic. For the color detection operation Open CV software will be used to detect the color of the object. The output will be received by Arduino IDE to sort the object into the correct color basket. Arduino dashboard will be used to display the current sorting process. The sorting details will be saved to a database. The mobile application will display the past sorted details by retrieving data from the database.

Main Operating systems will be used in the project will be windows and android. The Arduino IDE and Open CV software will be run on the windows operating system and mobile application will be run on android operating system.

❖ Work Breakdown Project Timeline:

Project Phases and Deliverables

Phase	Deliverables
Identify the Problem	Problem statement, initial requirements.
Research	Detailed project plan, research findings.
Planning	Project plan, resource allocation, timeline.
Information Gathering	Complete requirements document.
Build the Project	Prototype of the sorting machine with color detection.
Testing 01	Initial testing report.
Discuss the Issues and Developing	Updated prototype.
Final Testing	Final testing report, updated prototype.
Quality Assurance	QA report, final prototype.
Submit the Project	Final documentation, submitting the project.

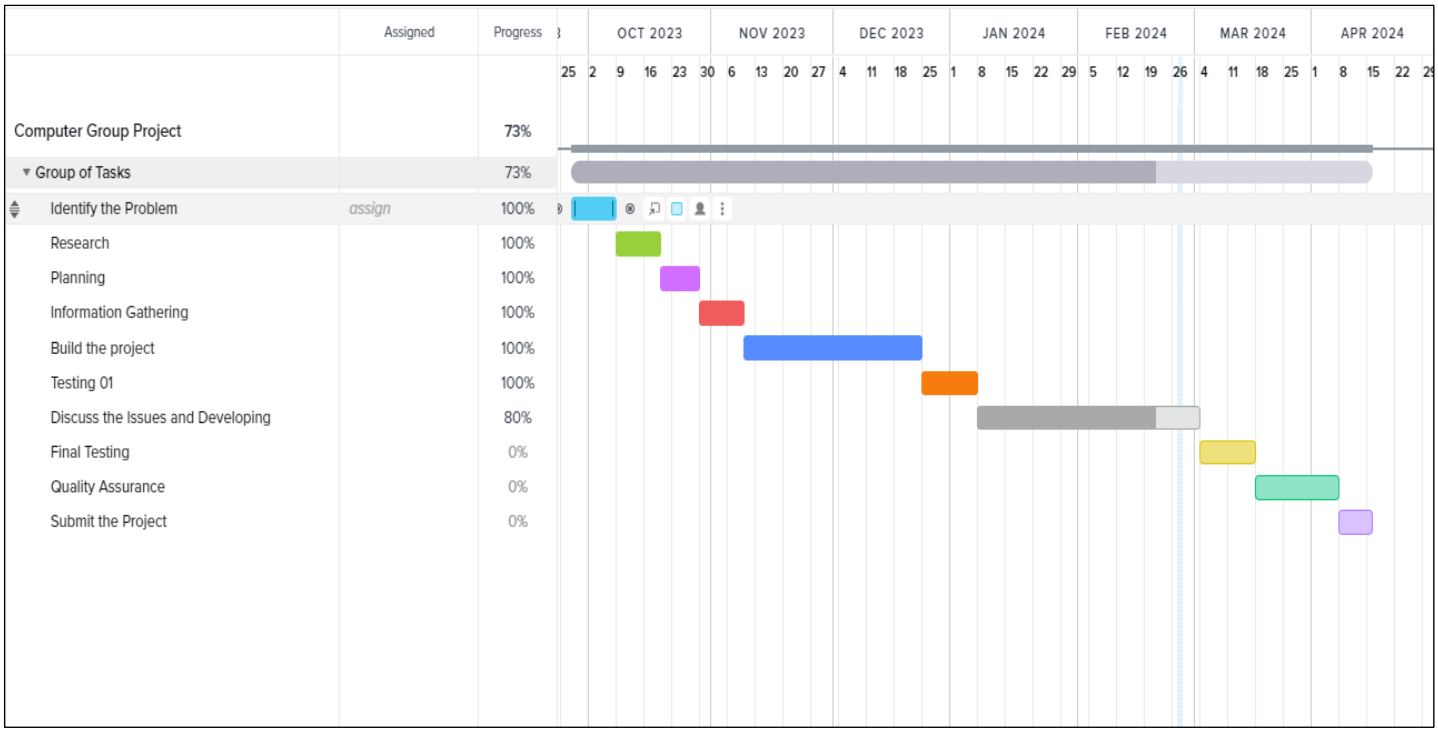
Tasks Breakdown at Each Phase:

Phase	Tasks
Identify the Problem	Forming the project team, conducting initial research, defining the project scope.
Research	Conducting in-depth research on sorting machines, color detection, and conveyer belts.
Planning	Assigning roles, setting up communication channels, creating a detailed project plan.
Information Gathering	Gathering detailed requirements from stakeholders, finalizing the design.
Build the Project	Developing the hardware, coding the software, assembling the machine.
Testing 01	Conducting initial tests on the prototype, identifying issues.
Discuss the Issues and Developing	Addressing issues found in testing, making necessary improvements.
Final Testing	Conducting final tests, ensuring all requirements are met.
Quality Assurance	Ensuring the quality of the prototype, making any final adjustments.
Submit the Project	Preparing final documentation, submitting the project.

Project Timeline:

Phase	Duration
Identify the Problem	10 days
Research	10 days
Planning	1 week
Information Gathering	10 days
Build the Project	6 weeks
Testing 01	2 weeks
Discuss the Issues and Developing	7 weeks
Final Testing	2 weeks
Quality Assurance	3 weeks
Submit the Project	1 week

Gantt Chart:


Assigned
Progress
OCT 2023
NOV 2023
DEC 2023
JAN 2024
FEB 2024
MAR 2024
APR 2024
25 2 9 16 23 30 6 13 20 27 4 11 18 25 1 8 15 22 29 5 12 19 26 4 11 18 25 1 8 15 22
Computer Group Project 73%
Group of Tasks 73%
Identify the Problem assign 100%
Research 100%
Planning 100%
Information Gathering 100%
Build the project 100%
Testing 01 100%
Discuss the Issues and Developing 80%
Final Testing 0%
Quality Assurance 0%
Submit the Project 0%

Milestones Aligned with Deliverables:

Project Proposal (End of Identify the problem, Research, Planning Phases)
Project timeline (End of Information gathering phase)
Prototype Development (End of building the project phase)
System Assembly (End of testing 01 phase)
Final Prototype (End of discuss issues and development, Final testing phases)
Final Project Documentation (End of all phases)

Chapter 4

Results & Discussion

Findings and Discussion:

In order to create a thorough sorting process with increased efficiency, the color sorting project successfully combined a few technologies, including the ESP32 board, Arduino IDE, OpenCV Python, Arduino Cloud, IoT components, website, and MySQL database. The system's goals were to boost sorting operations' throughput while offering real-time monitoring and user-interaction capabilities.

Actuators and sensors used in the sorting process were easier to control when the ESP32 board was programmed using the Arduino IDE. Accurate color-based item classification was made possible by using OpenCV Python for image processing and color detection. Real-time counts of the red, blue, green, and total sorted objects were shown on the Arduino Cloud platform, which allowed for remote monitoring of the sorting process.

The system's interaction with the website's MySQL database was one of its key advantages. The MySQL database received sorted data from the ESP32 board, including counts of red, blue, green, and total items. This made it possible to store, analyze, and retrieve data via the web interface. After sorting, users were given the option to change the sorted data, which increased accuracy and flexibility.

To provide security and control while in use, the system was designed with two buttons: one for emergencies and the other to initiate and stop the sorting process. Through a different website interface, people could also input sorted details and write comments. Administrators could then examine and approve the information, choosing drivers according to the data sorted. The website allowed drivers to view task specifics, which streamlined communication and processing.

A comprehensive, effective color sorting system with improved functionality was produced by integrating these technologies. Results from additional analysis and interpretation would shed light on the functionality, efficiency, and possible areas for development of the system.

Chapter 5

Conclusion:

The task of creating a thorough sorting procedure using an ESP32 board, the Arduino IDE, OpenCV Python, Arduino Cloud, Internet of Things components, a website, and a MySQL database was effectively tackled by the color sorting project. Our goal in integrating these technologies was to improve sorting efficiency, which would lead to increased accuracy and productivity.

The effective implementation of real-time data transfer from the ESP32 board to the MySQL database, which enables smooth monitoring and tracking of sorting operations, is one of the study's key findings. Users were able to observe the counts of red, blue, green, and total objects being sorted by using the Arduino Cloud interface, which offered visibility into the ongoing sorting process.

Furthermore, the system's adaptability and security were improved with the addition of emergency buttons and user-editable data, which gave users the ability to step in and make changes as needed.

Nonetheless, a few drawbacks were discovered throughout the project, such as the requirement for further user interface and sorting algorithm optimization. In addition, to support greater sorting activities, the system's scalability might need to be addressed.

Future Works:

A few of these issues deserve more investigation and development in future research projects. First off, the efficiency of the system would be improved by accelerating processing and improving the sorting algorithms to handle more complicated sorting tasks.

Furthermore, investigating the use of machine learning methods may enhance the precision of color identification and categorization, particularly in demanding settings.

Additionally, usability and user experience would be enhanced by improving the user interfaces of the website for posting sorted details and adding comments, as well as the sorting interface.

Additionally, looking into the possible integration of new sensors or technologies to enhance the sorting system's capabilities—such as adding weight sensors to sort items according to weight or putting RFID technology in place to track specific items—could open new functionalities and optimization opportunities.

All things considered, the project establishes the groundwork for future study and innovation around automated sorting systems, which has the potential to transform industrial operations and boost productivity and efficiency.

Reference list

1. TechPacs.com (2022) *Color based ball sorting machine using Arduino*. https://www.youtube.com/watch?v=Yb-LUUzE1ak.

2. Minh Hoàng Lê (2019) *SORTING TOMATOES BASED ON SIZE AND COLOR USING TENSORFLOW AND ARDUINO*. https://www.youtube.com/watch?v=h0-NS3z-EXo.

3. *Counter for color sorting project problem* (2024). https://forum.arduino.cc/t/counter-for-color-sorting-project-problem/1236310.

4. Dejan (2022) *Arduino Color Sorter Project*. https://howtomechatronics.com/projects/arduino-color-sorter-project/.

5. *DIY Arduino Based Color Sorter Machine using TCS3200 Color Sensor* (no date). https://circuitdigest.com/microcontroller-projects/arduino-color-sorter-machine-using-tcs3200-color-sensor.

6. *Code for bottle cap color sorting machine* (2022). https://forum.arduino.cc/t/code-for-bottle-cap-color-sorting-machine/1043513.

7. *Need help joining colour sorting code and servos code. Help truly appreciated* (2014). https://forum.arduino.cc/t/need-help-joining-colour-sorting-code-and-servos-code-help-truly-appreciated/213202.

8. *Couple questions about color sorting* (2014). https://forum.arduino.cc/t/couple-questions-about-color-sorting/223268.

9. Ijraset (no date) *A review on Arduino based color sorting machine*. https://www.ijraset.com/research-paper/arduino-based-color-sorting-machine.

10. IJRASET Publication (2022) 'A review on Arduino based color sorting machine,' *www.academia.edu* [Preprint]. https://www.academia.edu/70695751/A_Review_on_Arduino_Based_Color_Sorting_Machine.

Contribution

Name	ID Number	Contributions
Bathala Wicramasinhe	10899497	• Assigning the course work to group members. • Monitoring progress keeping in track the objectives to finish to the time limit. • Technology selection and Component selections for the Project. • Creating Arduino accounts and forms to gather information. • Updating the GitHub account and committing updates to the platform • Developing the Arduino code for the ESP32 board • Gathering research materials and project tutorials about the Arduino coding and libraries for data communications with Esp 32 board and the Arduino IDE • Adding necessary variables and functions to the code to refrain from getting false outputs from the open CV python code. • Building the IOT components together using jumper wires and bread boards • Testing the functionality of the components. • Adding the communications with the python color detection outputs • Dealing the serial monitor communication. • User testing and gathering 3rd party overviews and delivering to the team and discussing them. • Building the design of the project • Testing the code and solving faults and confusion in the Arduino code. • Checking the maximus weight that can be carried by the conveyor belt and adjusting the correct speed of the belt. • Testing the minimum distance of the object that should be triggered. • Forming the connection between the Esp32 board and the Arduino cloud • Setting the database connection with ESP 32.

Gallage Weerarathne	10899433	• Open CV python coding • Color detection code writing. • Dealing the serial monitor communication. • Designing the Project architecture. • Adjusting the code for detect the colour red, green, and black. • Sending outputs to the Arduino IDE • Adding a frame to the object to detect the object's color and isolate the object to easy the detection process. • Adjusting the frame of the camera input • Building an AI model for detect the color of the object. • Developing the code to only detect strong shades of color red, green and blue to minimise the defect of detecting false colors. • Updating the GitHub account and committing updates to the platform
Galahitiyawe Senevirathne	10898910	• Researching about the Arduino cloud platform • Selecting proper icons, and widgets for the dashboard. • Exploring the capabilities with the Arduino cloud editor. • Linking variables with the corresponding variables. • Setting the correct parameters for the widgets • Testing the connection with the board and the Arduino cloud. • Gathering research workloads. • Designing the architecture of the project. • Updating the GitHub account and committing updates to the platform • Gathering nonfunctional requirements.

Hewa Seelawansha	10899426	• Exploring application development with flatter software. • Developing the mobile application • Adding proper functionalities to the app by adding buttons and frame works. • Developing the website to retrieve the past sorted data. • MySQL database development. • Setting the parameter to connect database and the Arduino IDE. • Hosting website1. • Developing the interfaces, login, data tables and backend of the website. • Adding the website to the mobile application. • Setting the PHP coding for the website. • Setting the interface and buttons to navigate through the application. • Developing the front end of the website. • Gathering User requirements. • Updating the GitHub account and committing updates to the platform • Gathering References
Hingurala Wijesingha	10899435	• Website2 Developing • Database with user inputs. • Adding login interfaces to the user, admin, and drivers. • Developing the communication between the three entities (user, admin, and drivers). • MySQL database handling for the website 2. • Developing interface for corresponding entities with the respective website handling powers. • Testing the website and database connection. • Testing the communication between the three entities. • Updating the GitHub account and committing updates to the platform • PHP file handling for Website2 • Hosting the website 2 • Developing the UI for the website 2. • User testing