CSE396 Computer Engineering Project

Final Report

Development of a Autonom Crane

PortFlow

Prepared by Group 13

31.05.2024

Contents

1. Group Members

Student ID	Name	Modules		
210104004204	ABDULKADİR RAMANLI	Embedded System	Desktop (Database)	Hardware
210104004142	ALPER TAVŞANOĞLU	Mobile Application	Embedded System	Hardware
200104004025	EMİR İNCE	Desktop (Database)	Mobile Application	Hardware
210104004270	FURKAN AYAN	Desktop (Simulation)	Embedded System	Hardware
200104004101	HÜSEYİN KOÇAK	Desktop (Simulation)	Embedded System	Mobile Application
200104004122	METE AHMET YAKAR	Desktop (Simulation)	Desktop (Database)	Mobile Application
210104004042	RECEP FURKAN AKIN	Desktop (Simulation)	Desktop (Database)	Mobile Application

2. Project Overview

PortFlow is designed to be a modular and scalable model that mimics the container handling systems used in global port logistics. Utilizing a combination of Raspberry Pi technology and custom-built hardware, the system integrates real-time data processing and IoT to manage and simulate container movements effectively. This project aims to address the challenges associated with the conventional handling of logistics at ports, which are often characterized by inefficiencies such as delays, high operational costs, and underutilization of resources.

2.1 Project Scope

Hardware Integration: Implementing an advanced setup using Raspberry Pi units and various electromechanical components to emulate real-world crane operations and container movements within a simulated port environment.

Software Development: Crafting robust software solutions including a mobile application for operational management, a MySQL database for real-time data tracking, and a dynamic Unreal Engine simulation for operational visualization.

2.2 System Functionality

Real-Time Tracking and Management: The project utilizes IoT technologies to track containers in real time, processing all data through a MySQL database to manage logistics dynamically and efficiently.

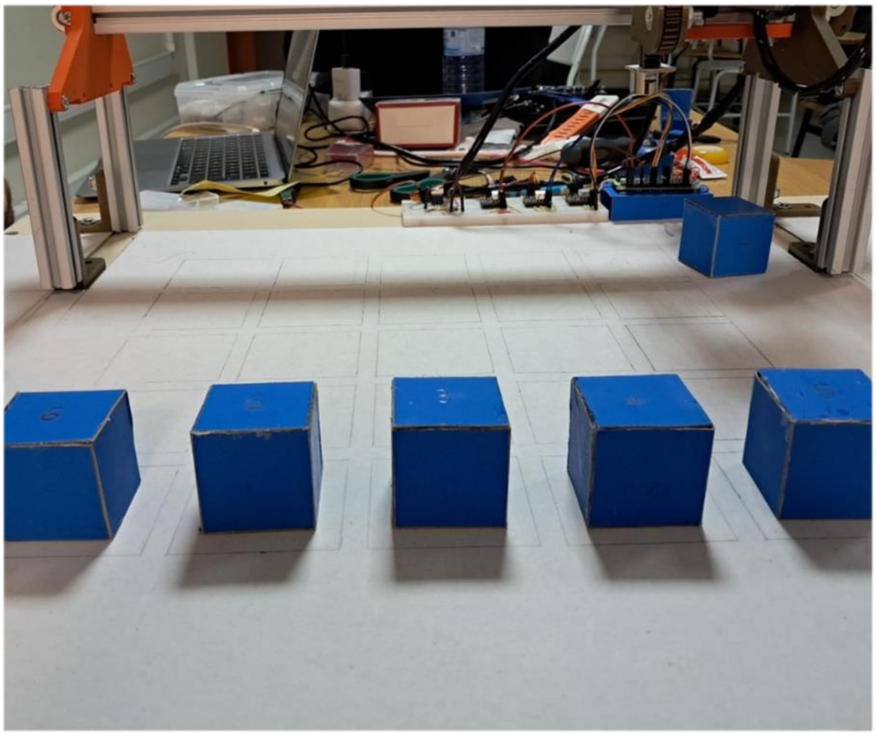
Mobile Application: Developed in Flutter, this application allows operational managers and logistics coordinators to remotely manage container logistics, including functionalities like adding, deleting, and efficiently sorting containers by number using a merge sort algorithm.

Simulation: Leveraging Unreal Engine, PortFlow provides a real-time 3D visualization of port operations, enabling operators to visualize the immediate effects of logistical decisions and manage operations with enhanced foresight and precision.

3. Modules

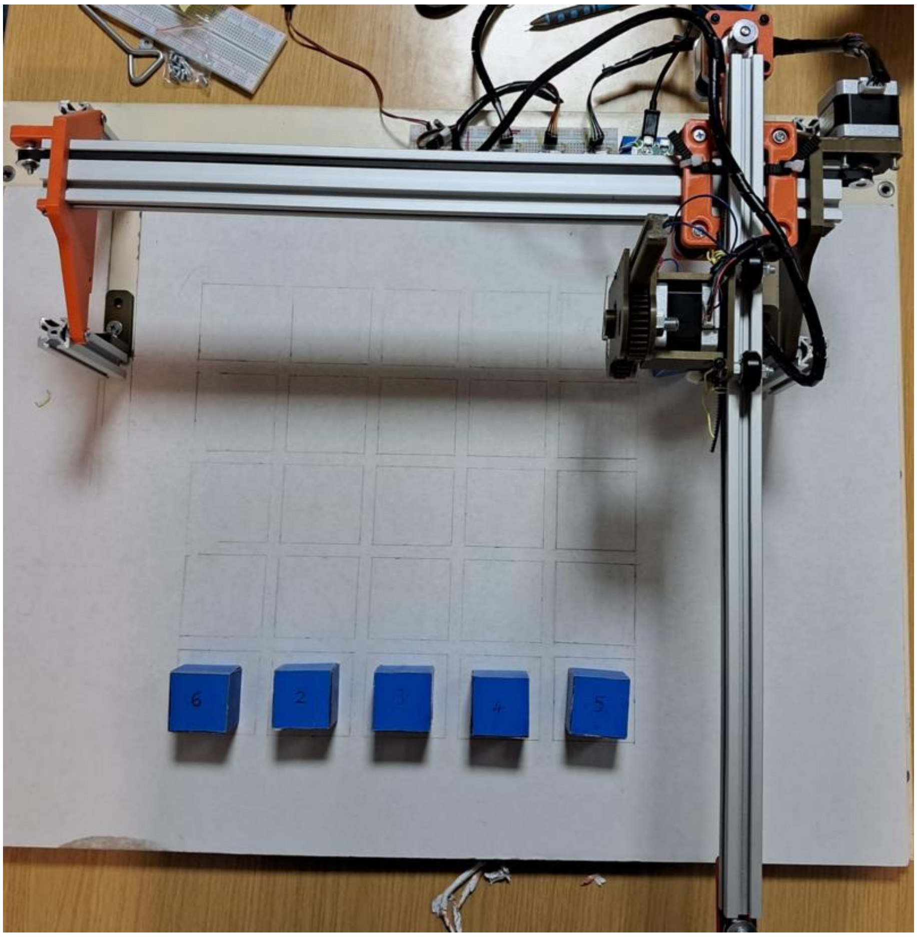
3.1 Hardware

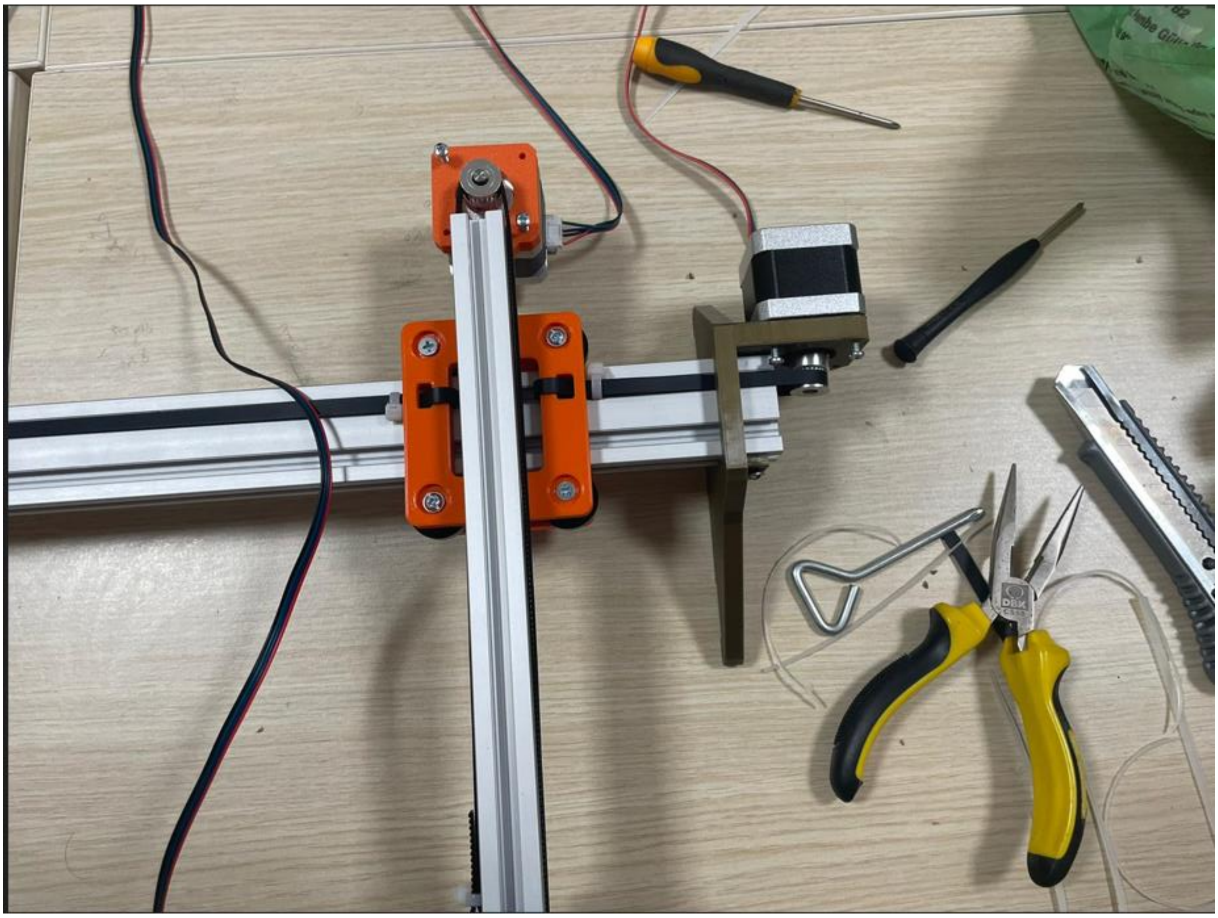
The hardware module is responsible for the assembly and integration of various components, forming the backbone of the system. It comprises aluminum sigma profiles, Raspberry Pi, stepper motors, electromagnets, and 3D printed parts. This module ensures the structural integrity and functionality of the system.



The system's physical structure is built using aluminum sigma profiles with integrated stepper motors and electromagnets for moving and handling miniature containers.

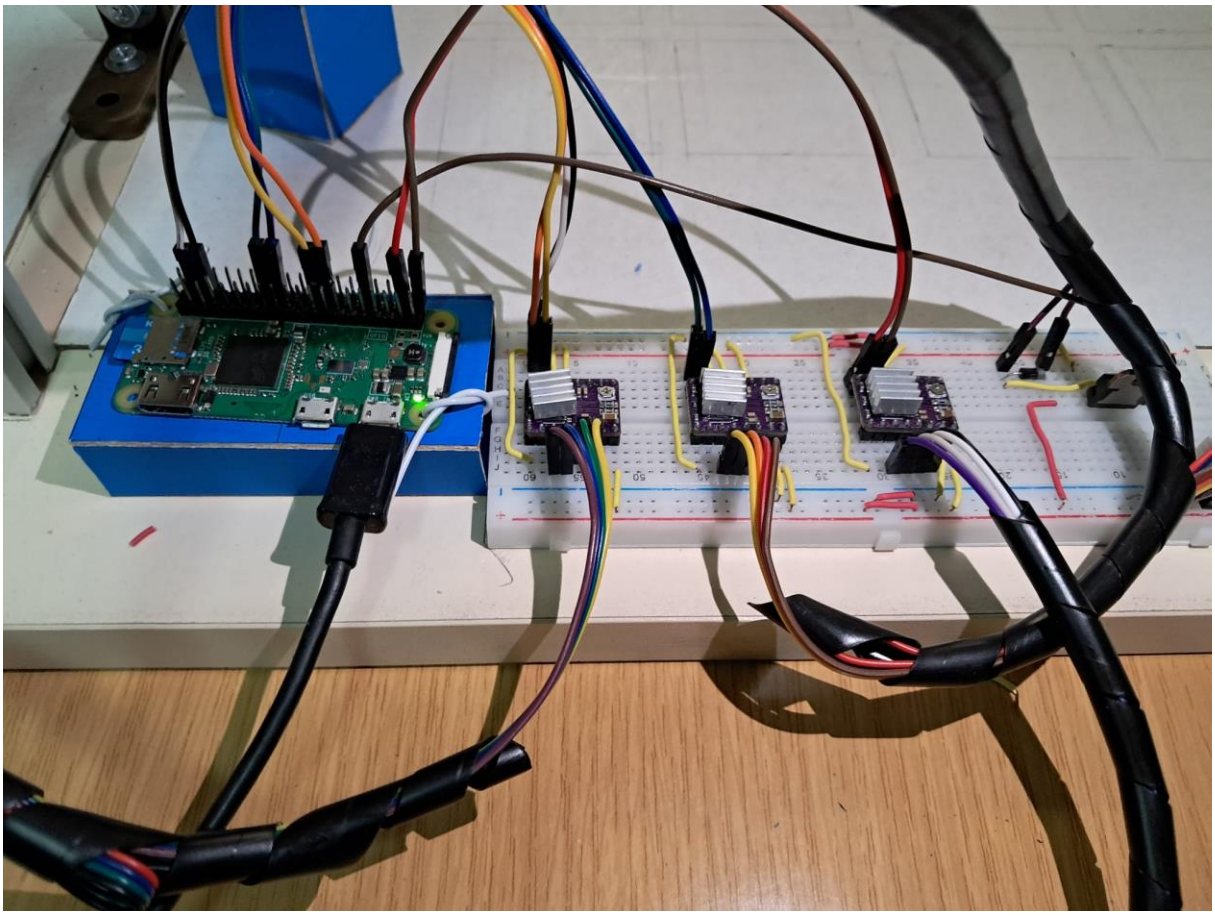
The assembly of the physical components has been successfully completed. We have achieved functional integration of the electromagnets which are now capable of picking up and moving metal containers as per commands from the control system.


6
2
3
4
5



3.2 Embedded System

This module develops an autonomous crane system using a Raspberry Pi Zero. It aims for balanced movement, autonomous operation, seamless control system integration, command execution, and hardware-software coordination.



Utilizes pigpio for GPIO pin management and motor control, ensuring precise movement and operation consistency. Integration with the mobile and desktop applications via a local network ensures real-time operations and data consistency.

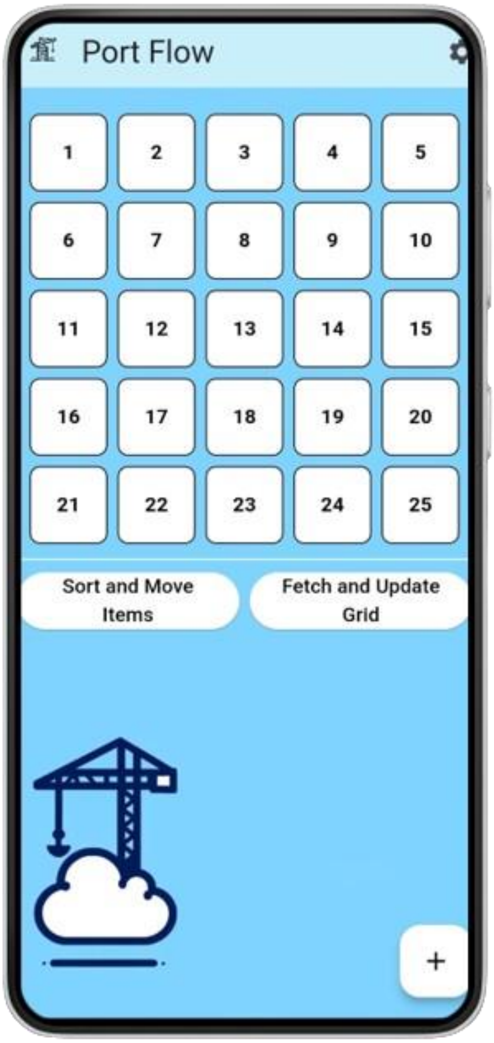
Successful implementation of motor controls and preliminary testing of command execution have been carried out, showing promising precision and reliability in operations.

3.3 Mobile Application

The mobile application for the PortFlow project serves as a key interface for users to interact with the container management system directly from their smartphones. It provides functionalities that allow users to manage, track, and sort containers efficiently while on the move.

Developed using Flutter, this framework was chosen for its ability to deliver high-performance, natively compiled applications for mobile from a single codebase. Flutter facilitates rapid UI development with a rich set of widgets and is ideal for implementing reactive apps with streamlined user experiences.

Leveraging the system's database, the app updates the real-time status and location of containers, providing push notifications for changes or required actions. A key feature in the app is the implementation of a sorting algorithm that organizes containers by their numbers. This function helps in quick retrieval and organization of container data, enhancing operational efficiency.

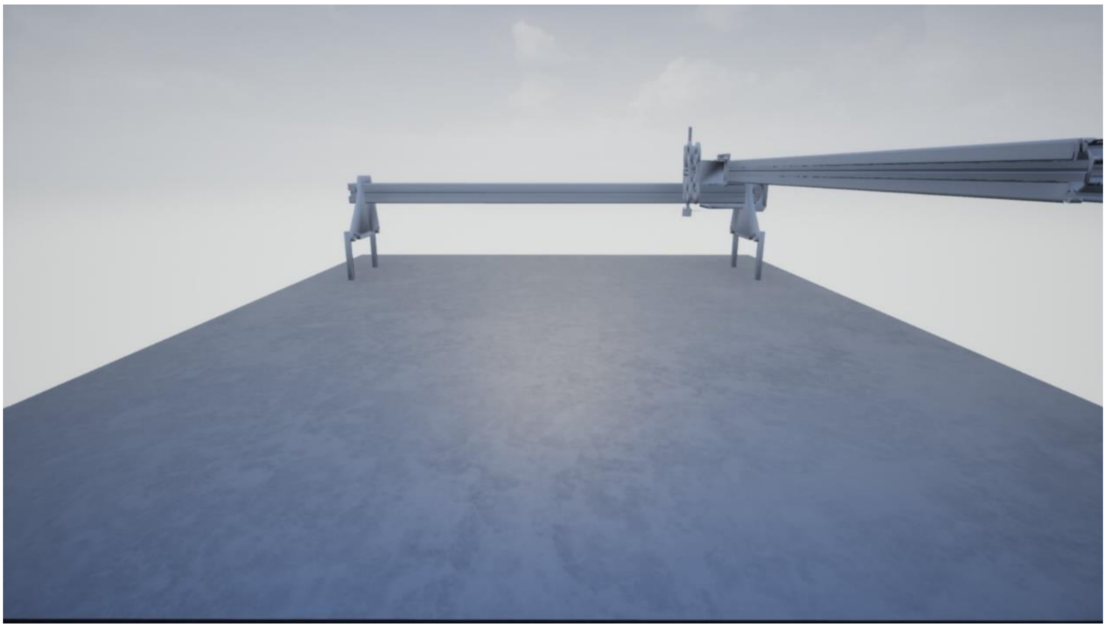


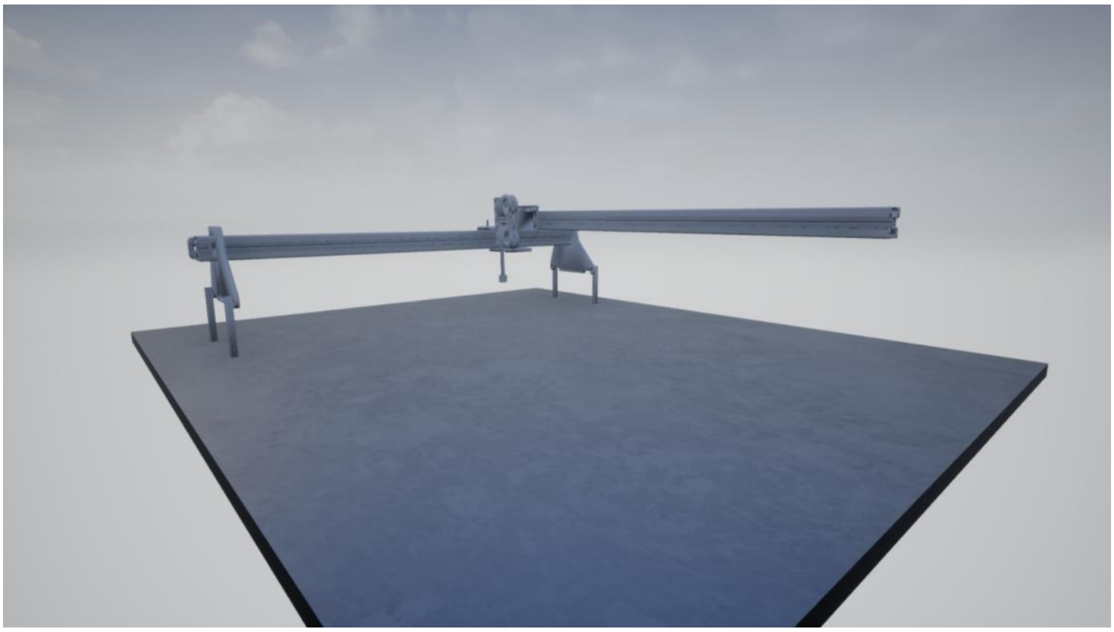
3.4 Desktop (Simulation)

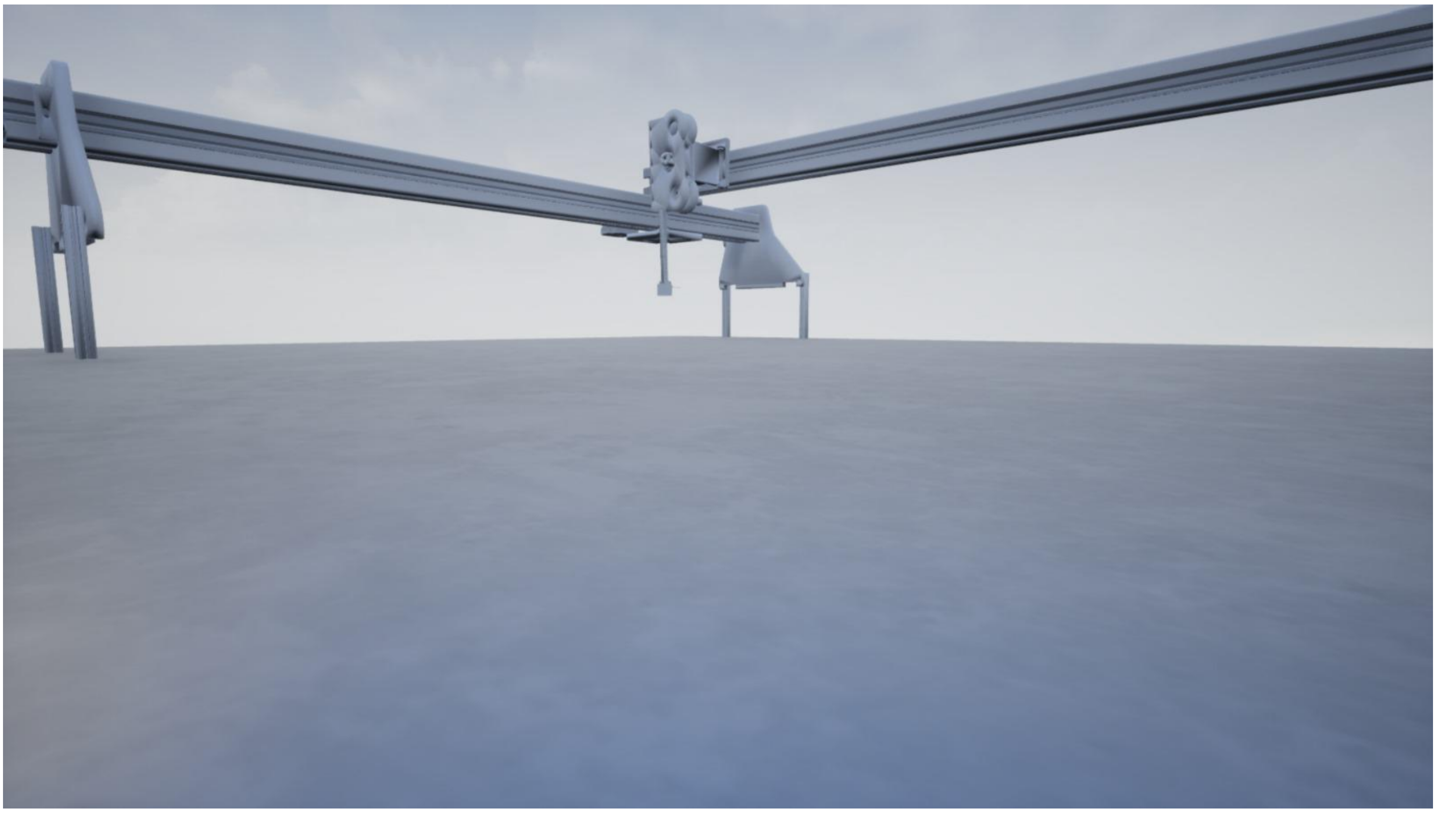
The simulation module of PortFlow is designed to provide an immersive, real-time visualization of the miniature port's operations. Using Unreal Engine, the module offers high-fidelity graphics and physics simulations that closely mimic real-world behaviors of cranes, containers, and other port machinery.

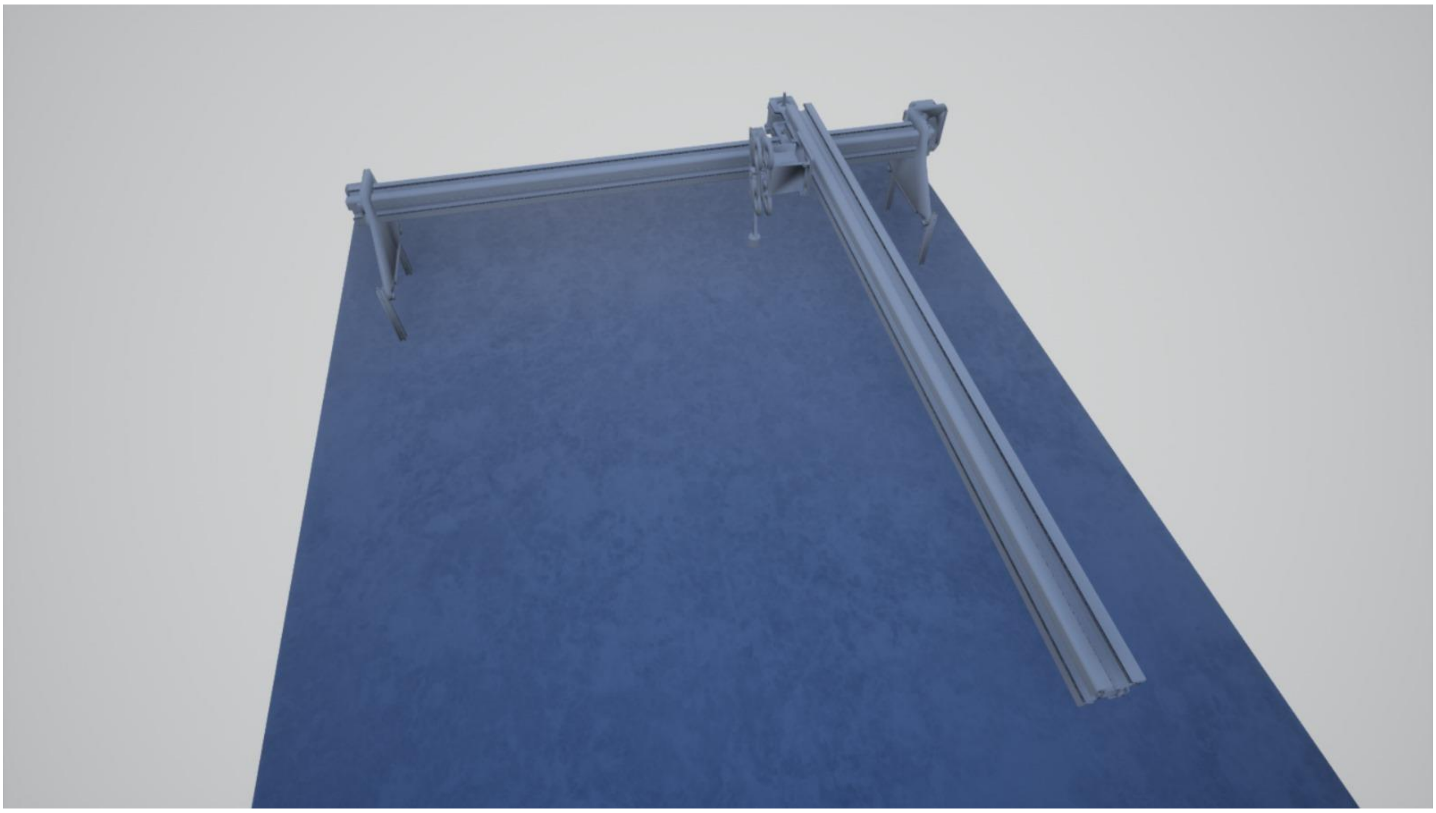
Unreal Engine is selected for its advanced rendering capabilities, comprehensive physics engine, and broad support for real-time interactions and simulations, which are essential for accurately depicting complex industrial environments. The environment is a scaled model of a typical port, including dock areas, storage zones, and transportation lanes. Each element within the environment is interactive and responds realistically to user commands and system events.

Users can interact with the simulation in real-time, initiating movements of containers and equipment through the desktop and mobile interfaces. The simulation reflects changes immediately, showing the real-time position and status of containers as they are logged and moved within the system. Besides operational planning and testing, the simulation serves as an educational tool, demonstrating port logistics and container management to users.





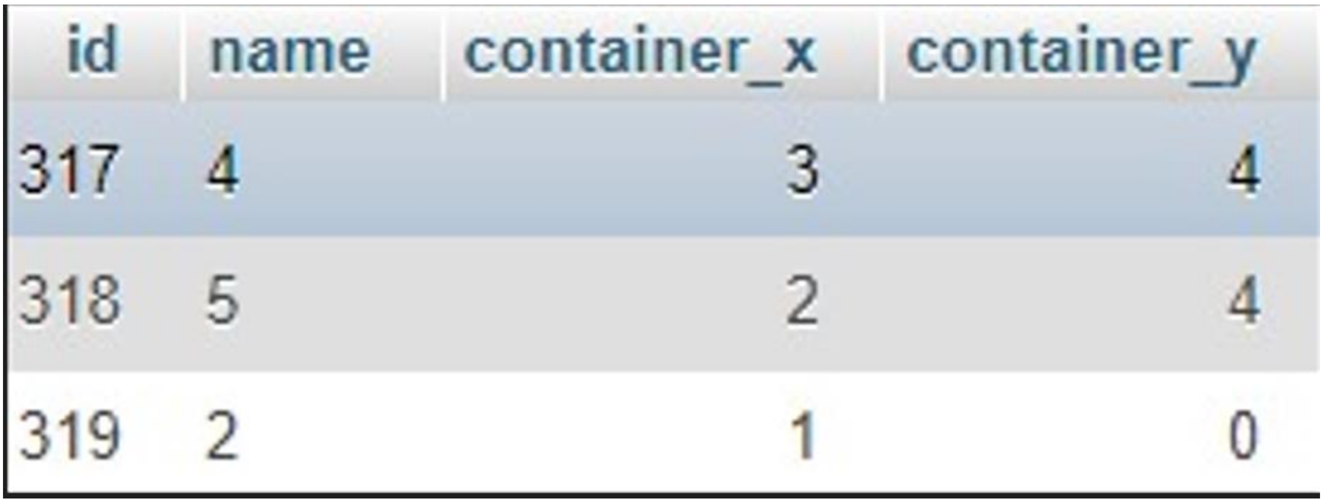




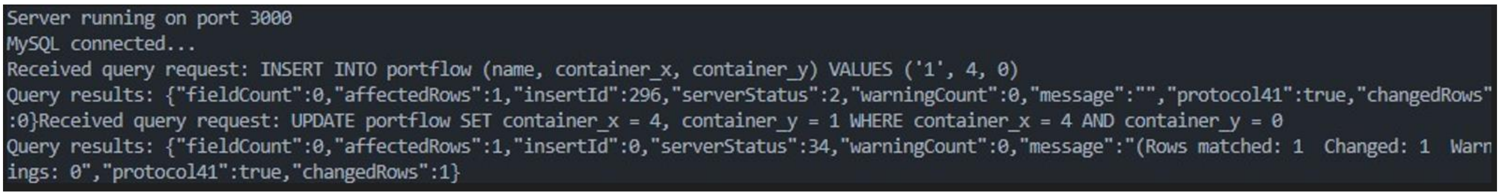
3.5 Desktop (Database)

The database module serves as the backbone for the entire PortFlow system, handling all data related to container tracking, logistics operations, and user interactions. It stores detailed records of container specifications, locations, and movement logs, ensuring that data is consistently available and reliable for both real-time operations and historical analysis.

The schema includes tables for containers, operations logs, user actions, and system settings. Relationships are carefully defined to optimize query performance and data integrity.



id	name	container_x	container_y
317	4	3	4
318	5	2	4
319	2	1	0



```
Server running on port 3000
MySQL connected...
Received query request: INSERT INTO portflow (name, container_x, container_y) VALUES ('1', 4, 0)
Query results: {"fieldCount":0,"affectedRows":1,"insertId":296,"serverStatus":2,"warningCount":0,"message":"","protocol41":true,"changedRows"
:0}Received query request: UPDATE portflow SET container_x = 4, container_y = 1 WHERE container_x = 4 AND container_y = 0
Query results: {"fieldCount":0,"affectedRows":1,"insertId":0,"serverStatus":34,"warningCount":0,"message":"(Rows matched: 1  Changed: 1  Warn
ings: 0","protocol41":true,"changedRows":1}
```

4. Project Scenario

1. Transport ships will ferry containers to designated sections of the port.

2. Upon arrival, containers will be logged into the system via a mobile application, detailing their precise locations and contents.

3. Simultaneously, this data will be stored on the desktop interface for comprehensive tracking.

4. Users will initiate container requests seamlessly through the mobile app.

5. Cranes will then navigate to the specified container locations, efficiently lifting and relocating them to their designated destinations.

6. Sorting Operation

By Numbers: The mobile application activates a sort algorithm, which efficiently sorts the containers by their numbers in ascending order. This is particularly useful for inventory checks, audits, and when planning container movements or shipments.

By Object Names: When sorting by object names, the algorithm adjusts to sort alphabetically, allowing for easy access to containers based on the type of goods they carry. This is crucial for fulfilling specific container handling operations that may prioritize certain types of goods.

7. Upon receiving a request to load onto a truck via the mobile app, the crane will precisely maneuver the designated cargo to the awaiting loading vehicle, ensuring seamless transfer and efficient logistics coordination.

8. This entire process will be dynamically visualized in real-time through an immersive 3D simulation on the desktop interface, offering a vivid depiction of port operations.

5. Components and Materials

- Raspberry Pi Zero W
- 20X20 Sigma Profile 50cm
- 20X40 Sigma Profile 50cm
- 3x 17HS4401 Nema 17 Stepper Motor
- 3x DRV8825 Stepper Motor Driver
- GT2 Belt - 6mm
- 7x V Slot Wheel
- 2x GT2 Pulley
- 2x Idler Pulley
- 100 µf Capacitor
- 3D Printed Materials
- Electromagnet
- Diode
- Transistor
- Jumper Cables
- 12V/1.5A Power Supply

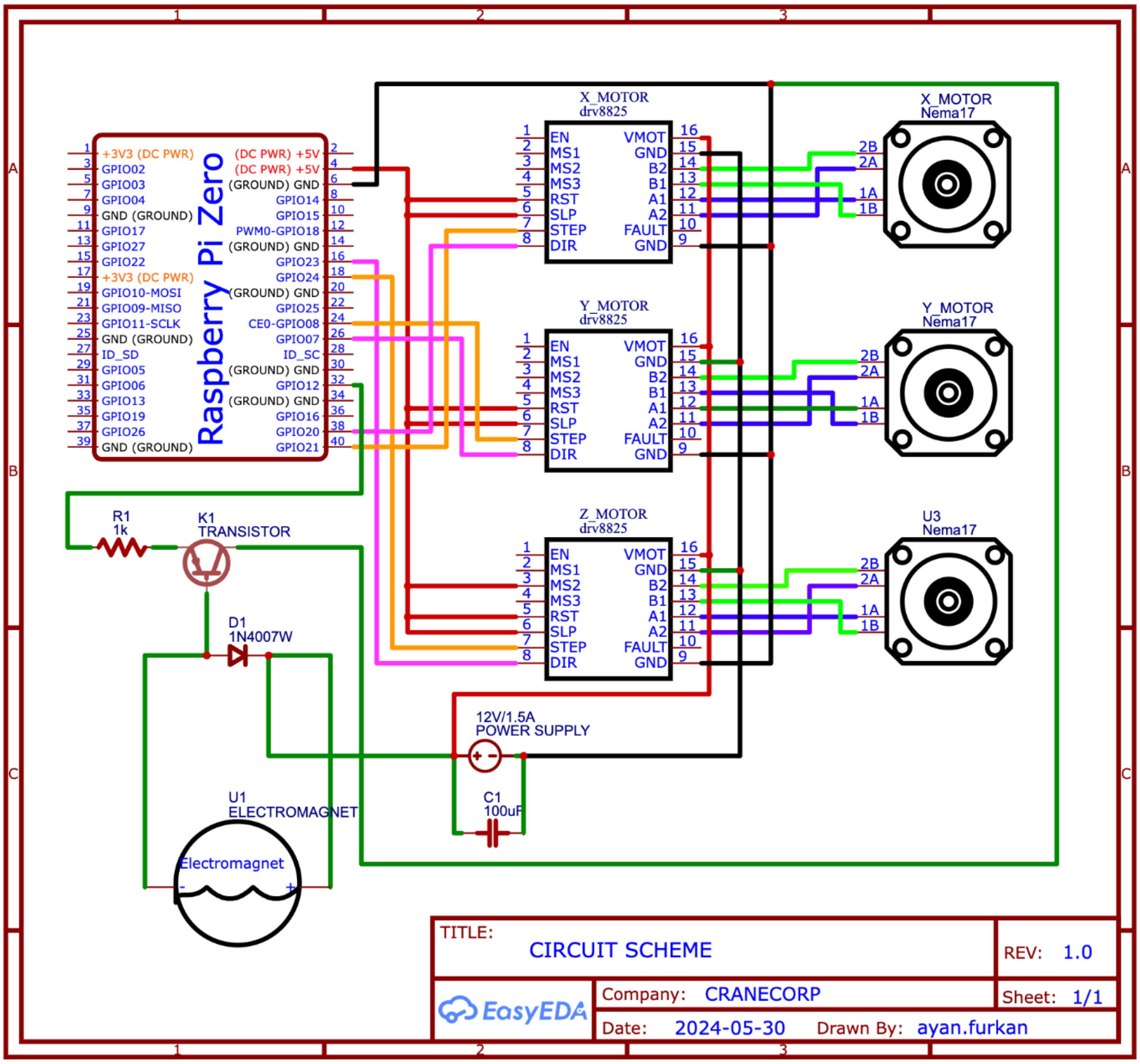
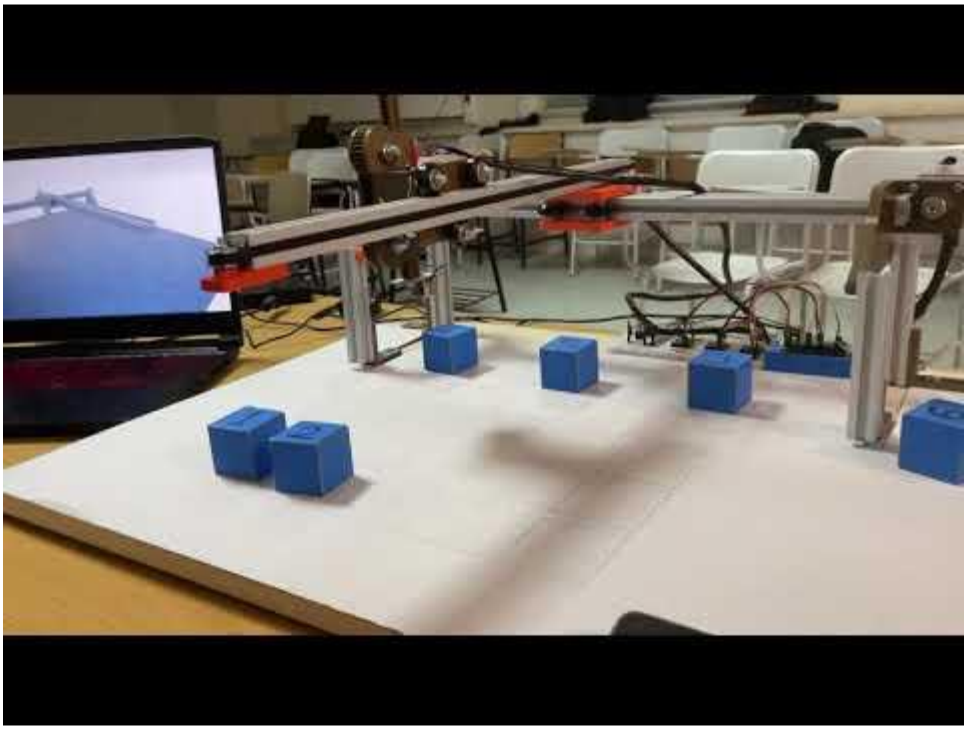


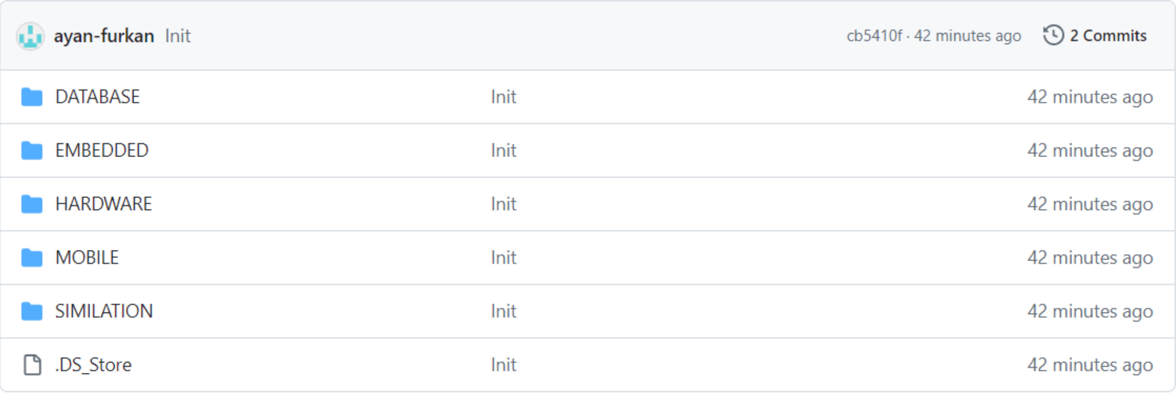
Figure 1. Circuit Scheme

6. Project Video



7. GitHub Page

https://github.com/ayan-furkan/CRANECORP

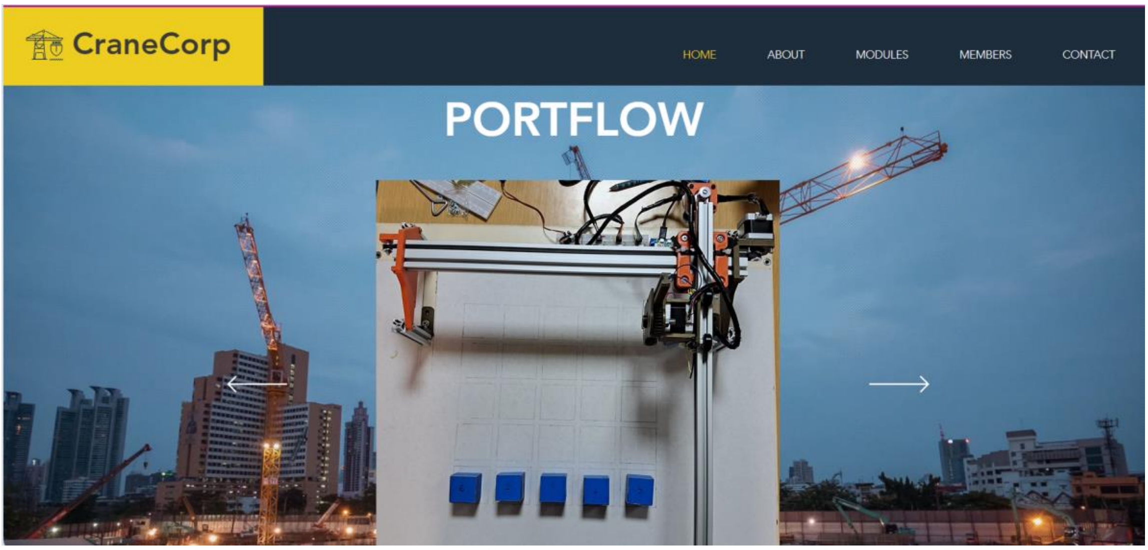


ayan-furkan Init   cb5410f · 42 minutes ago   2 Commits

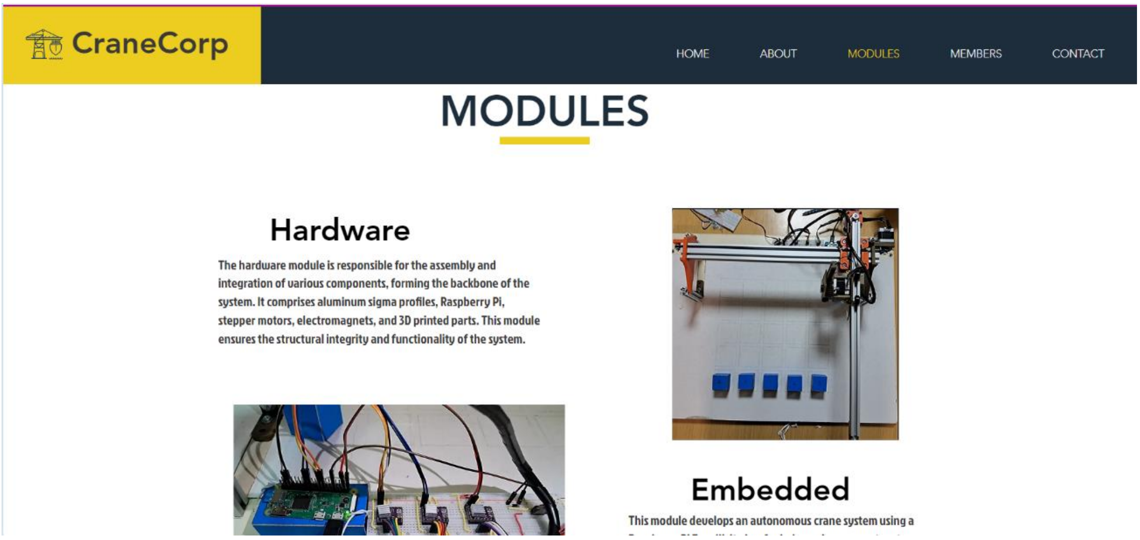
DATABASE	Init	42 minutes ago
EMBEDDED	Init	42 minutes ago
HARDWARE	Init	42 minutes ago
MOBILE	Init	42 minutes ago
SIMILATION	Init	42 minutes ago
.DS_Store	Init	42 minutes ago

8. Web site

https://cranecorpcse.wixsite.com/cranecorp



PORTFLOW



MODULES

Hardware

The hardware module is responsible for the assembly and integration of various components, forming the backbone of the system. It comprises aluminum sigma profiles, Raspberry Pi, stepper motors, electromagnets, and 3D printed parts. This module ensures the structural integrity and functionality of the system.

Embedded

This module develops an autonomous crane system using a