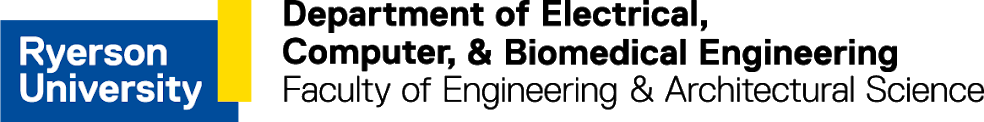
****

**COE/ELE 70AB Milestones Compliance Report (MCR)**

| **Project Title** | AA05 - Automated Parking Management System |
| --- | --- |
| **MCR Number** | III (Weeks 8-9) |
| **Project Manager for the MCR period** | Kisoban Rajendran |
| **Team Players for the MCR period** | Yadu Krishnan Madhu, Zohraan Badar, Taskin Abdur-Rahman |
| **Faculty Supervisor** | Alagan Anpalagan |

# **Tasks Outlined for the Reporting Period (e.g. MCR I – Weeks 4 & 5)** (Provide detailed information on the tasks to be completed for the reporting period as per the milestone submitted to your FLC in Week3)

**Group:**

1. Design software management setup (Version control, documentation, workflow tools and setup).
2. Initialize Frontend App.
3. Establish a working ESP32 Web Server for demonstration purposes.
4. Create Figma Designs for Frontend implementation.
5. Design Schematic planning.

**Student A (Yadu):** Contributed to initializing the React App for frontend, setting up Project Tables, Roadmaps and Todo Lists and establishing a working ESP32 Web Server.

**Student B (Zohraan):** Created frontend web app design for initial set of pages to start development. Other pages still need to be defined by group consensus (data analysis, etc.) as project progresses.

**Student C (Taskin):** Initialized github organization with repositories and projects tables for team workflow setup. Researched into optimal object detection strategy.

**Student D (Kisoban):** Created schematic for microcontroller board wiring setup.

# **Progress Made in Reporting Period (e.g. MCR I – Weeks 4 & 5) -** (Provide detailed information on the progress that you (as a group and individual) made during the reporting period. You can include figures, datasheets, flowcharts etc. and additional information as requested by your FLC. You should use your progress to justify compliance to the tasks outlined for the reporting period as per the milestones submitted to your FLC in Week3)

**Group:**

**Student A (Yadu):**

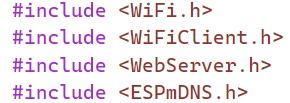
Implemented the ESP32 Web Server for communication purposes. This server will act as the potential hub where all the sensor and video camera captures aggregates are processed. Here is an image of the ESP32 used for the demonstration:

This particular model is the ESP32-DEVKITC-32D and it has Wifi, Bluetooth capabilities to boot.

The web server is implemented mainly using the ESP32 libraries and packages provided by Arduino, hence, the Arduino IDE is required for development.

The main language used for development in the Arduino IDE is **C++ (C conventions are used however)**.

Here is the Code implementation of the web server:

****

These are the header files needed for the Web Server Implementation



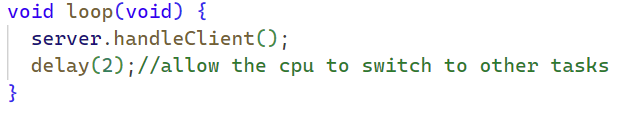
These are some of the custom functions and variables declared in the program.

* The SSID and Password variables store the Wifi SSID and Password as strings.
* The Webserver is declared as the “server” variable and it will have its port set to 80 for accepting incoming connections.
* The Handle root is a simple function that simply sends a success message after connecting with the server.



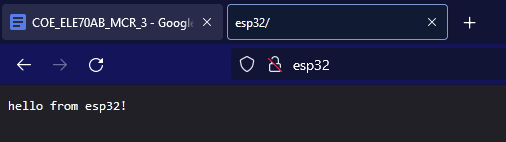
This is the main Setup loop that will **run only once** the program has been flashed into the ESP32. Any variables or functions that need to live during the program lifetime are declared here.

* The Serial.begin() function sets the communication BAUD rate which determines how fast data is transmitted from my PC to the ESP32. Here I have declared it to be 115200 bits per second.
* The 2 lines below initialize the Wifi connection.
* MDNS protocol makes it possible for resolving the hostname to its ip address. So instead of entering the Ip address, I can simply type “esp32” to reach the server.
* The server.on() function declares valid “URI”s so clients can connect and access various API’s
* Finally, the server is set to open to accept connections after the “server.begin()” function.

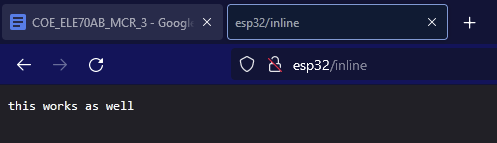


This is the main loop that is run indefinitely. A single line of code can handle accepting clients during runtime with “server.handleClient()”.

Here is the response from the connecting to the web server:

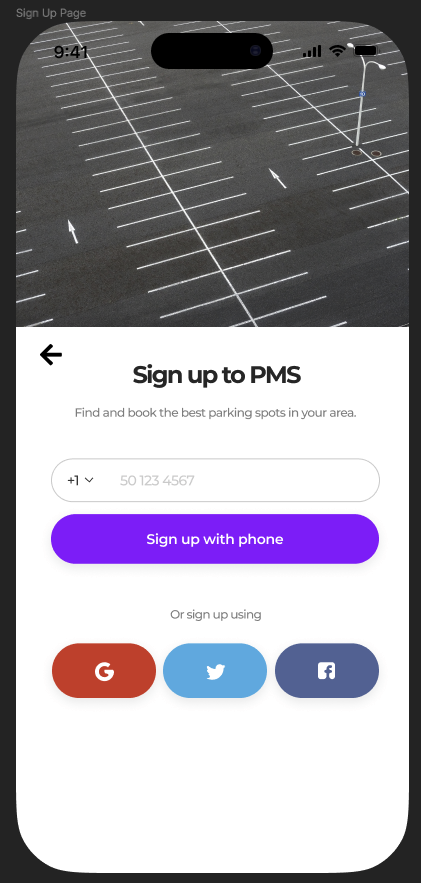
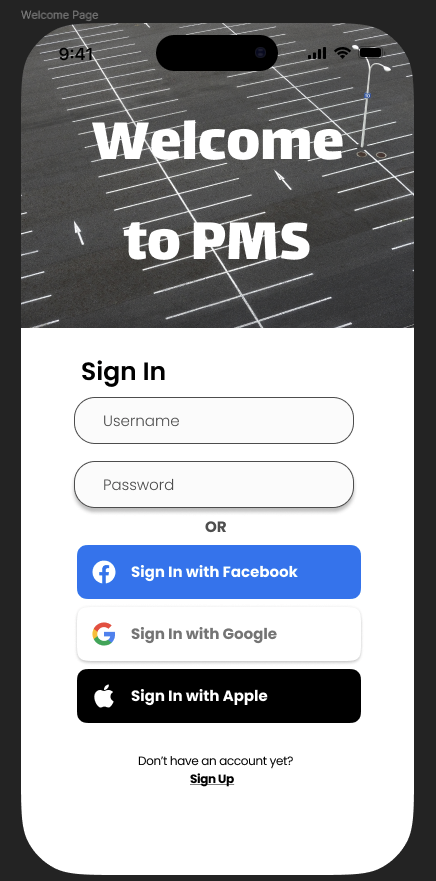


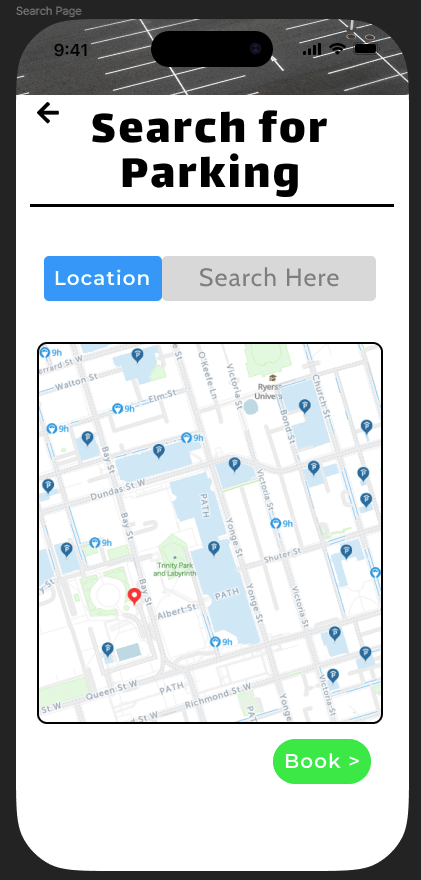
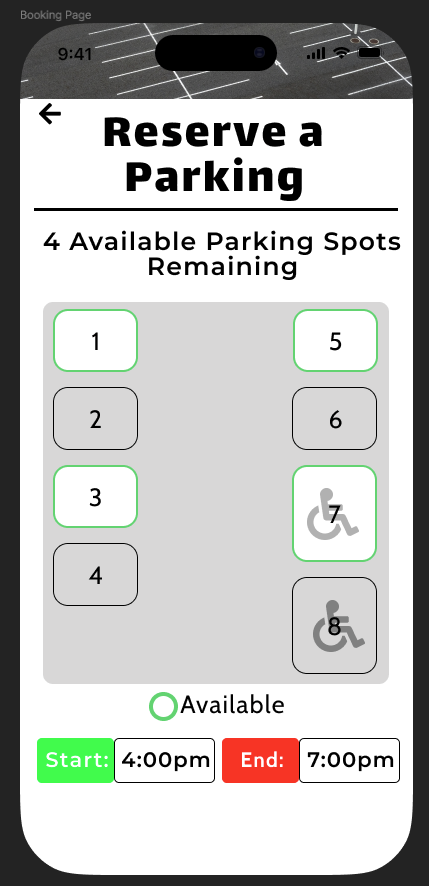
Here is a response from accessing a custom uri(inline):



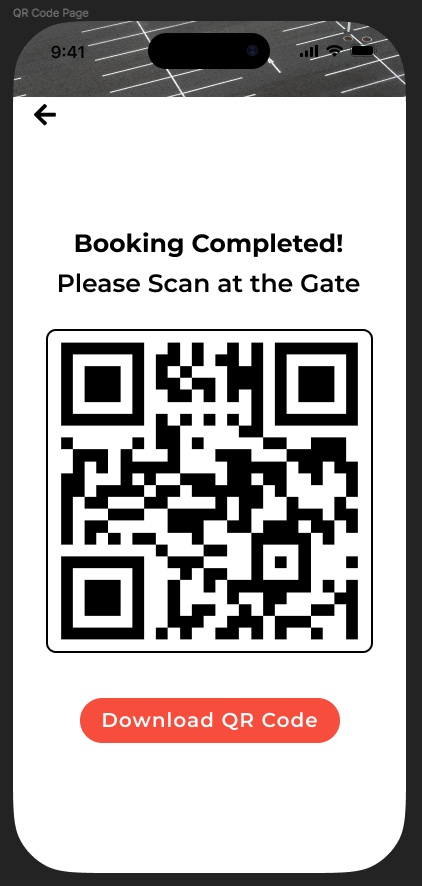
**Student B (Zohraan):** Created the mobile user interface design that will be used in the design of the mobile app.

Here are design figures for each page that must be implemented within the React App.

Welcome/Getting Started Page Sign Up Page 

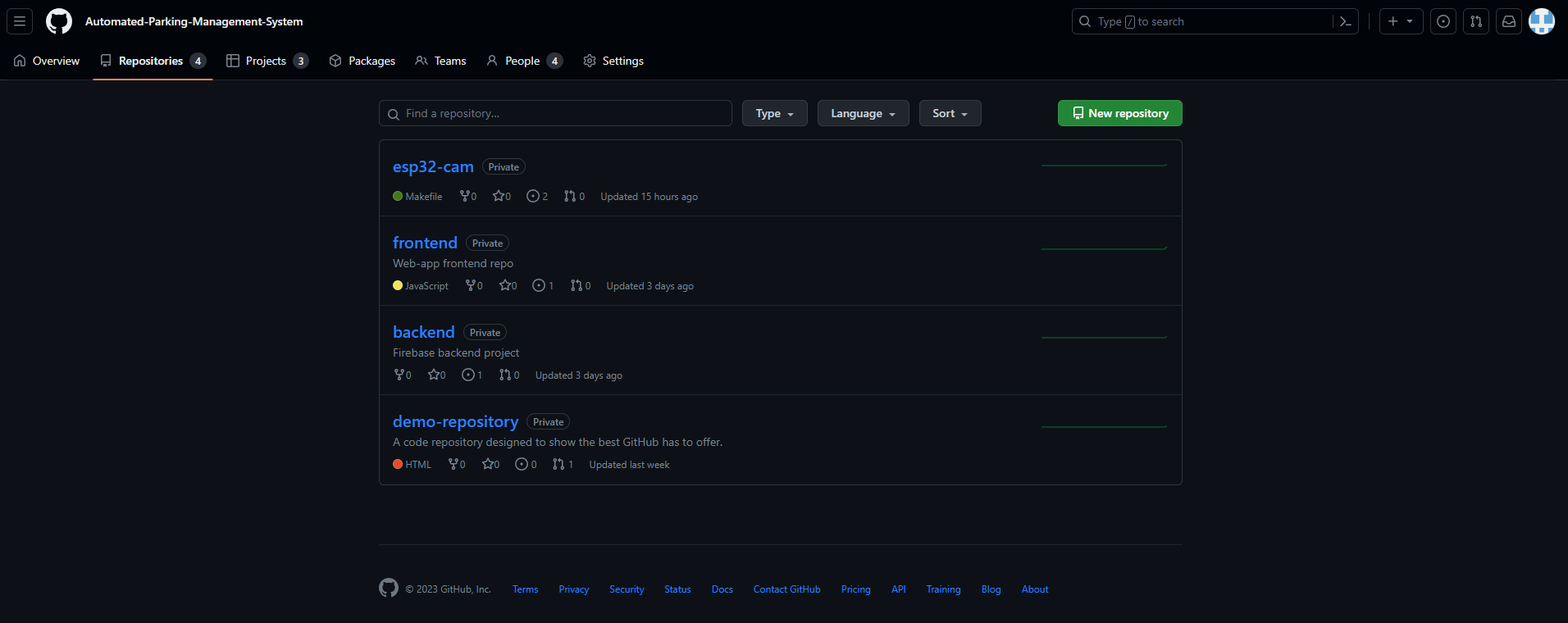
Search Page Booking Page

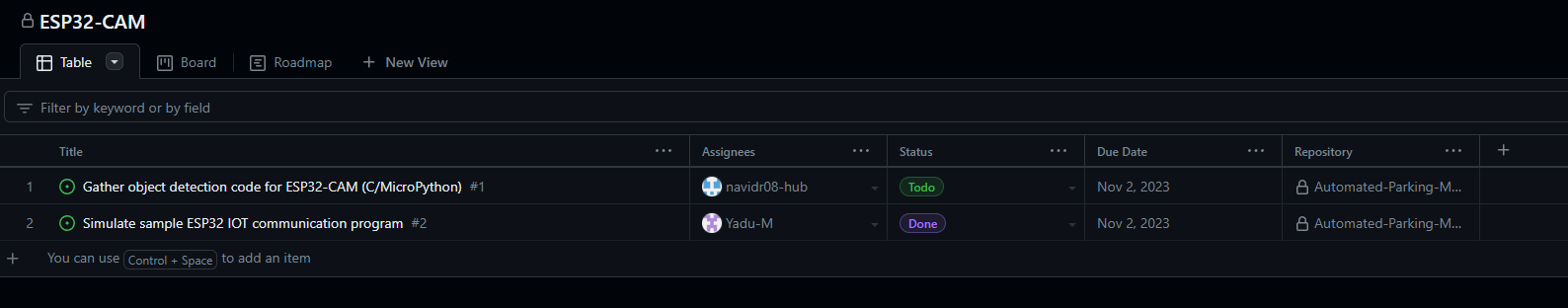
And Finally the Booking Page

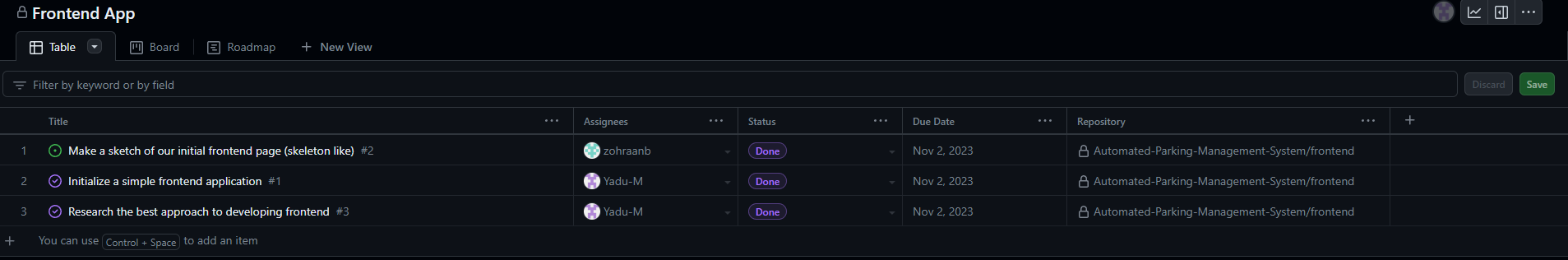
[](https://github.com/Automated-Parking-Management-System)

**Student C (Taskin):** Initialized github organization with repositories and projects tables for team workflow setup. Researched into optimal object detection strategy.

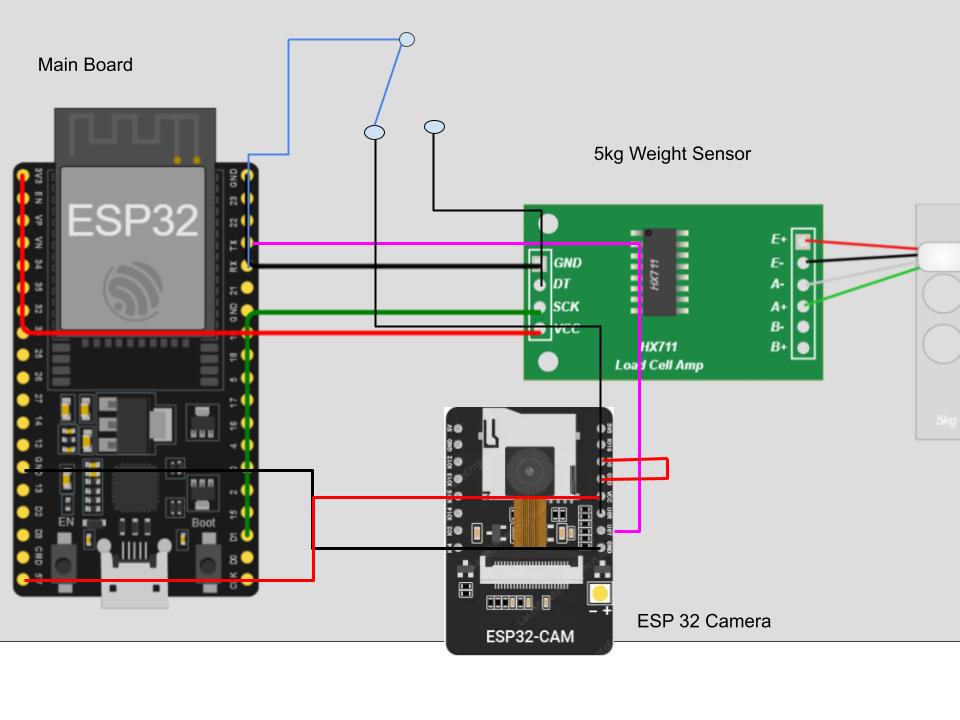
* <https://github.com/Automated-Parking-Management-System>







**Student D (Kisoban):** Was tasked as Project Manager for this period. Created a schematic of the overall circuit design for the entrance, illustrating the different parts and sensors that will be used in our design and how they will be connected to the board.



# **Difficulties Encountered in Reporting Period** (Provide detailed information on the difficulties and issues that you encountered during the reporting period and how you plan to address this in the following periods)

**Group:**

**Student A (Yadu):** Encountered difficulties while trying to flash the ESP32. Different ESP32 boards have various ways for flashing programs, hence, it was not easy to figure out the download mode. Additionally, to connect to any microcontrollers such as ESP32 which has built in UART, it is essential that the USB cable supports Data transfer along with Power, or else, the device will not be able to flash, from not knowing this fact, a lot of time was spent trying to diagnose the board.

**Student B (Zohraan):** had very little difficulties in developing the mobile UI that is to be used for the mobile app. Spent some time figuring out the UI and functionalities of FIGMA before creating the design pages.

**Student C (Taskin):** After looking at all the different object detection strategies it was clear that more thought and consideration is required before selecting any particular strategy. So demo ready code will not be included in this submission.

**Student D (Kisoban):** With the design, it may be hard to be cost-efficient since 1 ‘main board’ may not be enough to support 3 cameras as our original parking lot design intended. So different parts may need to share pins on the main board through the use of a switch or a mux or we can reduce the number of cameras in the overall design. Schematic will have to be modified so that it does not rely heavily on the main board.

# **Tasks to Be Completed in the Next Reporting Period** (Outline the tasks to be completed in the next reporting period. Please note this should match with your milestones submitted to your FLC in Week3, however in consultation with (and approval of) your FLC, you can modify this to accommodate incomplete tasks from previous period. Here you should also identify the Project Manager for the next period)

**Group:**

# Design review for parking spot tracking subsystem

1. Design review for GUI application.
2. Design review for database solution.
3. Design review for parking lot model.
4. Design review for testing process/ simulation.
5. Review integration between subsystems.

**Student A (Yadu):** Contributed to 1, 2 & 3.

**Student B (Taskin):** Contributes to 2, 3 & 4.

**Student C (Zohraan):** Contributes to 3, 4 & 6.

**Student D (Kisoban):** Contributes to 4, 5 & 6.