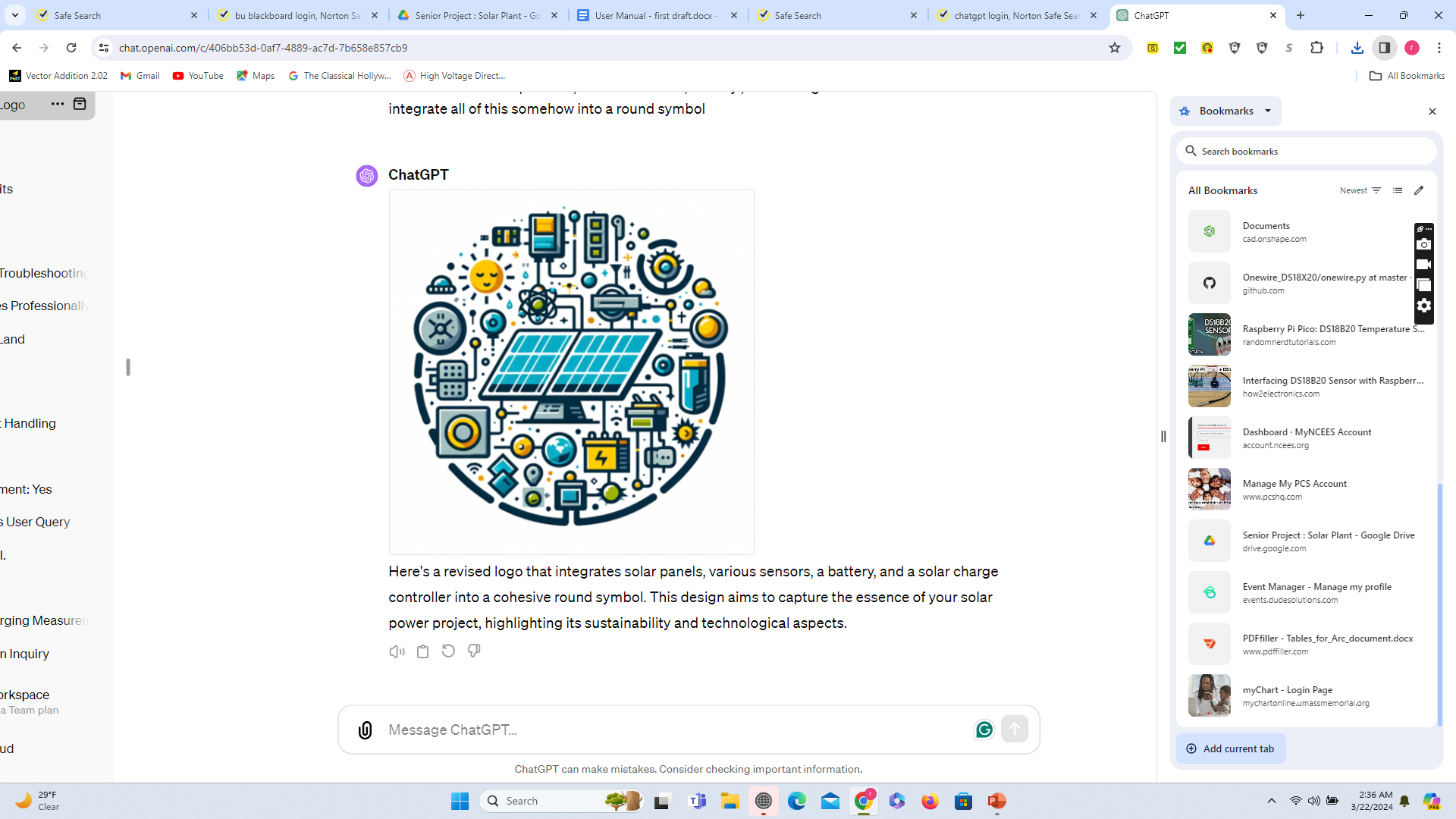
**Boston University**

**Electrical & Computer Engineering**

**EC463 Senior Design Project**

User Manual

Sol: Small Scale Stand Alone Solar Power Plant with Remote Monitoring & Control Application



Submitted to

Prof. Malay Mazumder

Boston University

8 St. Mary’s St Boston, 02215

(617) 353-0162

[mazumder@bu.edu](mailto:mazumder@bu.edu)

by

Team (15)

SOL

Team Members

Majid Al-Muhaideb [majidmuh@bu.edu](mailto:majidmuh@bu.edu)

Rahat Mahmood [rahat73@bu.edu](mailto:rahat73@bu.edu)

Alexa Wiencek [awiencek@bu.edu](mailto:awiencek@bu.edu)

Oluwaseun Angelo Soyannwo [seun@bu.edu](mailto:seun@bu.edu)

Steven Cheng [stcheng@bu.edu](mailto:stcheng@bu.edu)

Submitted: 15th April, 2024

#### Table of Contents

[Executive Summary 2](#_heading=h.30j0zll)

[1.0 Introduction](#_heading=h.1fob9te) 3

[2.0 System Overview and Installation](#_heading=h.3znysh7) 4

[3.0 Operation of the Project](#_heading=h.p62g92dyuvf)9

[4.0 Technical Background](#_heading=h.9ykws0h2npni) 11

[5.0 Relevant Engineering Standards](#_heading=h.m1kd2uyqcfm5) 16

[6.0 Cost Breakdown](#_heading=h.4d34og8) 18

7.0 Appendices 20

# Executive Summary

Off-grid solar plants, solar power plants that work independently from the grid, are among the most promising solutions to rising alternative energy demand. Our system integrates bi-facial solar modules with moving reflectors, which capture sunlight on both sides for enhanced efficiency, requiring a specialized management approach to optimize positioning and energy capture. Furthermore, these systems require an energy-storage element, usually a LiPo or SLA battery, and need additional safety considerations. Our client contracted us to develop a solar plant remote-monitoring and control system that keeps track of key metrics in various parts of the plant while ensuring precise and robust control of the bi-facial module system for optimized off-grid solar power performance. Our remote monitoring solution incorporates a web application that streamlines all of our plant monitoring features. Our overall technical approach involves the iteration of monitoring and control systems on a small-scale solar plant model, to provide our client with a robust, flexible system that can be easily adapted to full-scale configurations.

# Introduction

Our client, Professor Malay Mazumder, is working on a proposal to build a solar power plant on the rooftop of BU buildings. This proposal is in its early stages, and we have been contracted to assist with modeling the solar plant and creating a management system that monitors and controls the plant.

Off-grid solar plants require additional considerations to operate safely. Batteries are required to power loads during sun-absence periods, such as on cloudy days or during the evening, and must be constantly monitored to detect anomalous behavior that may cause damage. Furthermore, the condition of the solar panels must be constantly monitored to prevent reduced power or blackouts resulting from contamination or deterioration of the panels. These considerations in both safety and reliability introduce the need for a flexible and robust monitoring and control system that ensures the constant operation of the plant. Our client requested that we integrate remote monitoring considerations into our design, which we aim to achieve via a web application that displays key plant metrics.

The web application we have launched currently monitors our prototype model, displaying real-time data from the system to users upon login. Beyond this, it has been designed such that as our client and his team continue their research, additional solar arrays can easily be added to the web application to be monitored, and the data from these arrays will be stored securely in the cloud. Our physical system consists of two reflective panels and sliding connector points, with closed-loop control capabilities enabling automatic adjustment of the reflectors to maximize incident light on the bifacial solar panels. While the reflectors are idle, the stepper motors are disabled to maximize energy efficiency for 24-hour operation. All of our onsite programming is handled by two Raspberry Pi Pico W’s for a simple, compact, and robust design. With these features, we aim to increase the efficiency and productivity of our model solar power plant all while minimizing the necessity for user maintenance.

Our project has been to deliver a robust, flexible management system that can be adapted for larger configurations of off-grid solar plants and can be remotely monitored, reducing the need for physical presence near the plant. Our solution will allow the client to develop his proposal and assist in ensuring the safety and reliability of the final product and provide him with insight into what pain points could cause delays in his proposal.

# System Overview and Installation

## Overview block diagram

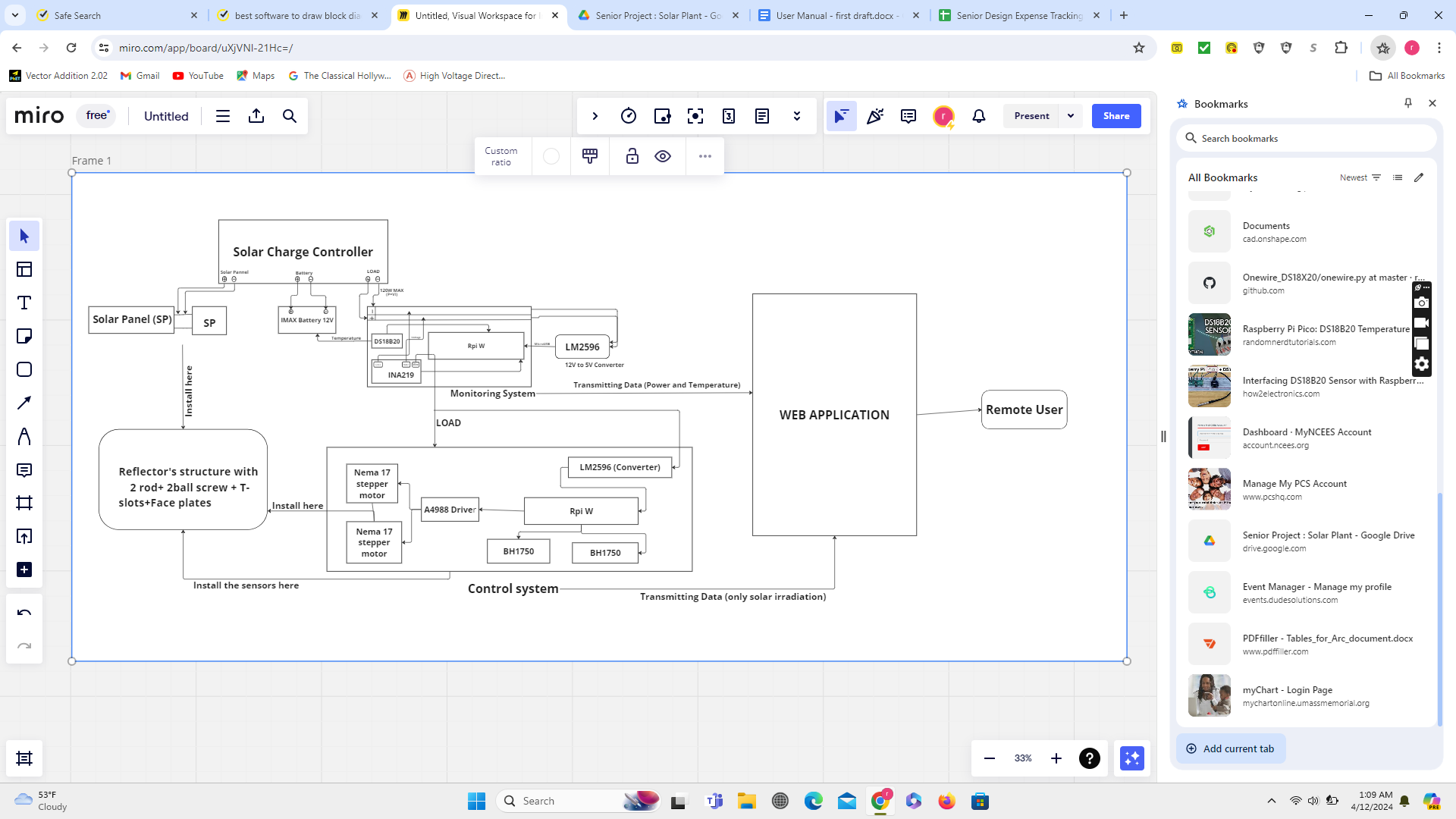
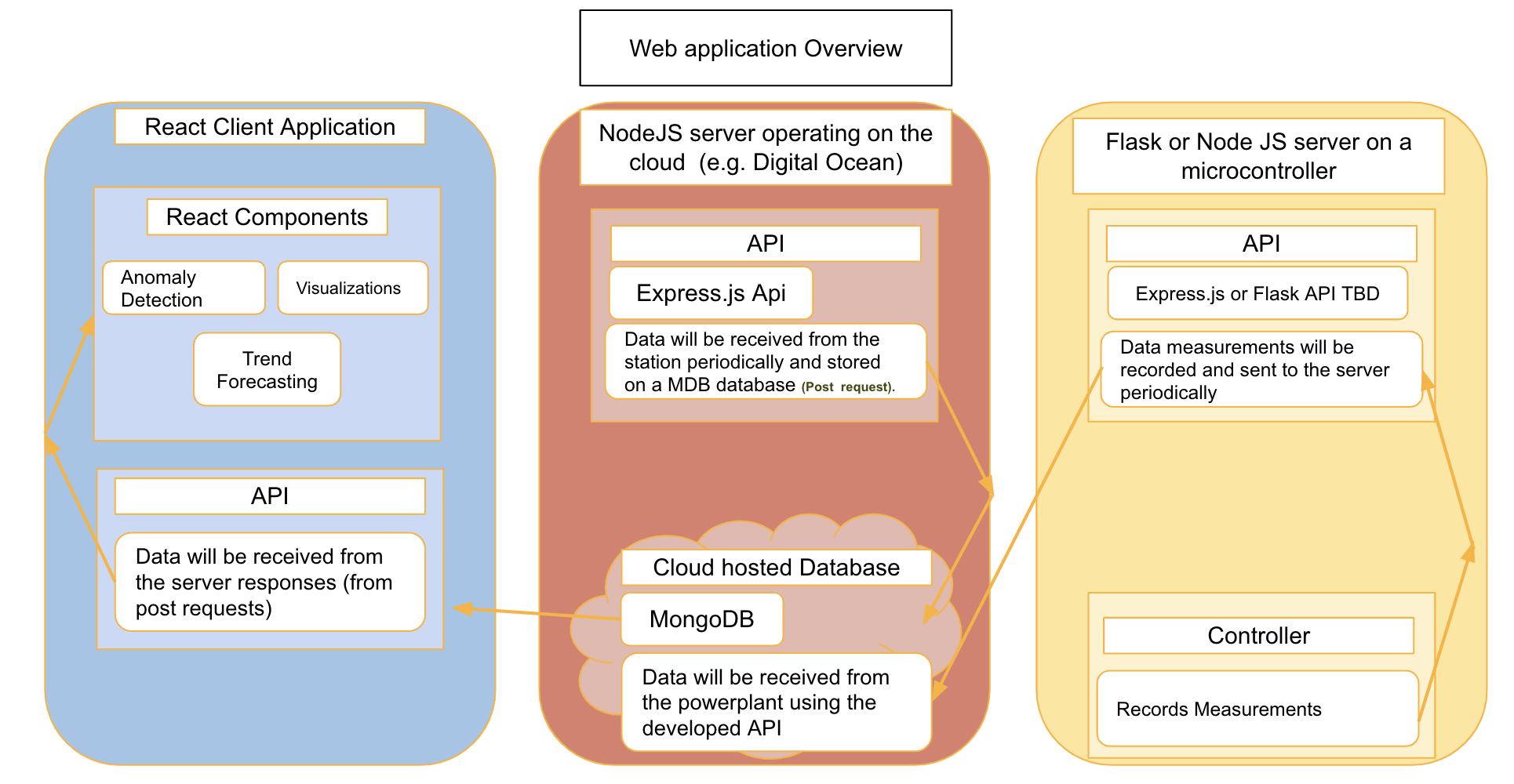
****

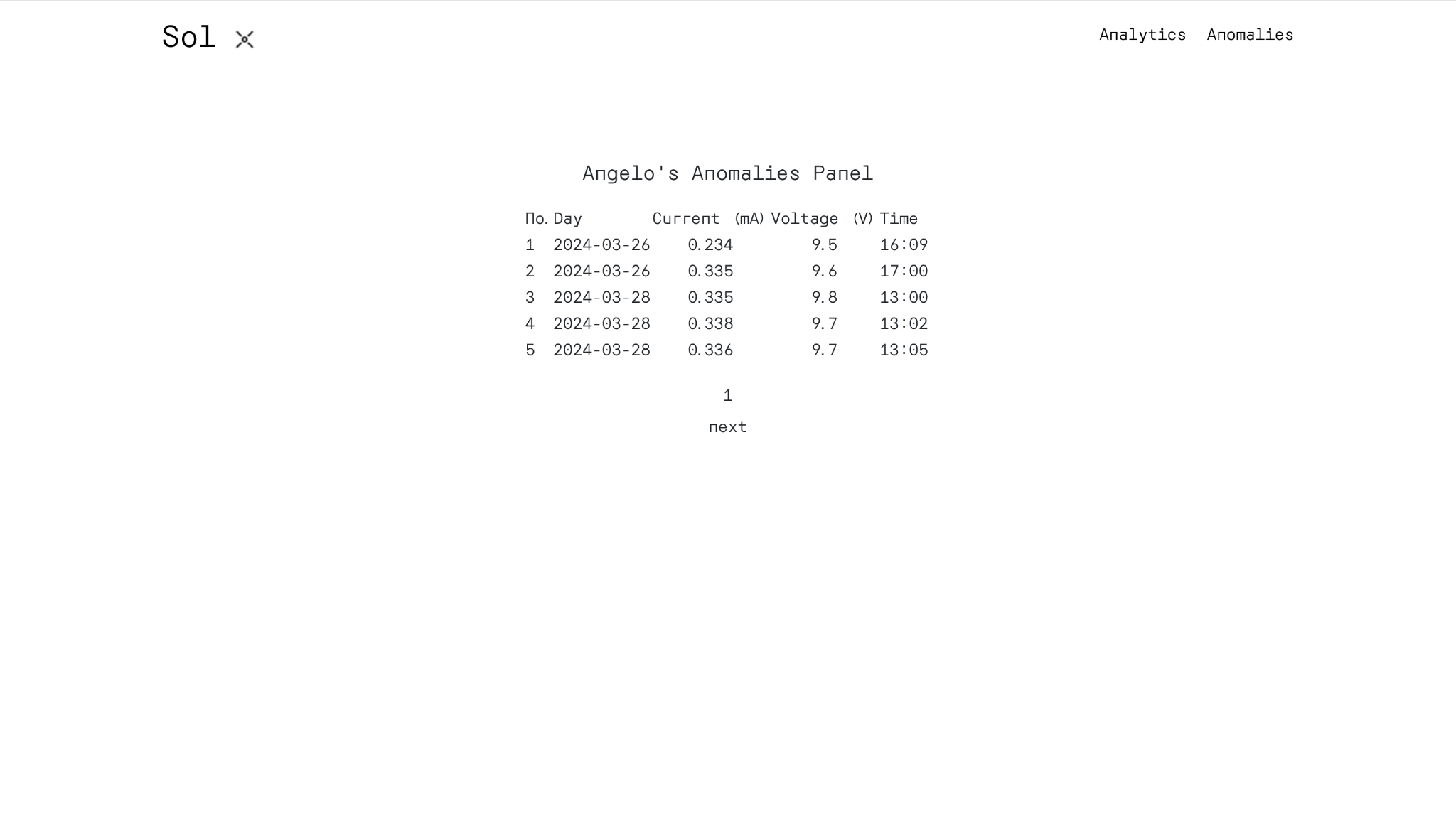
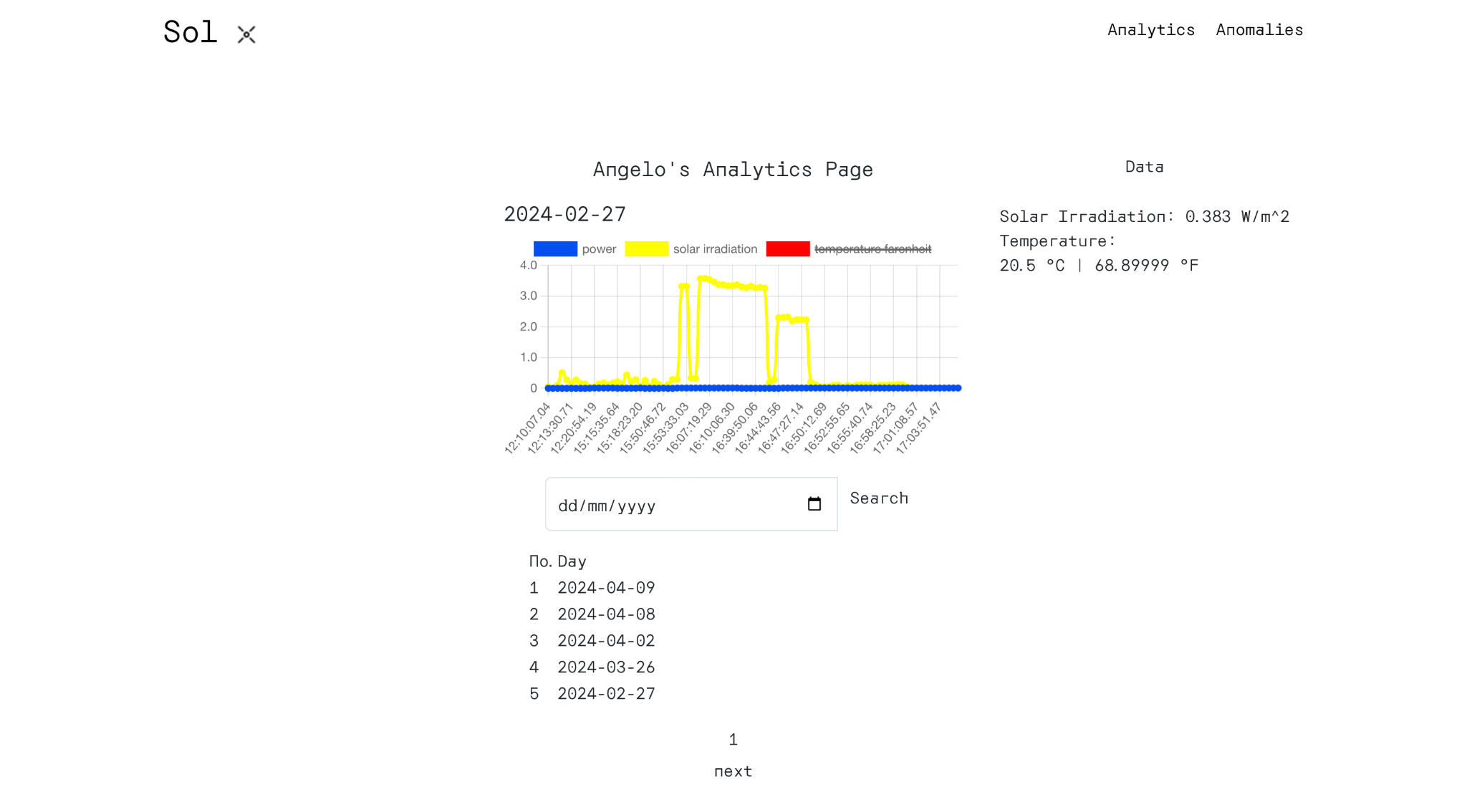
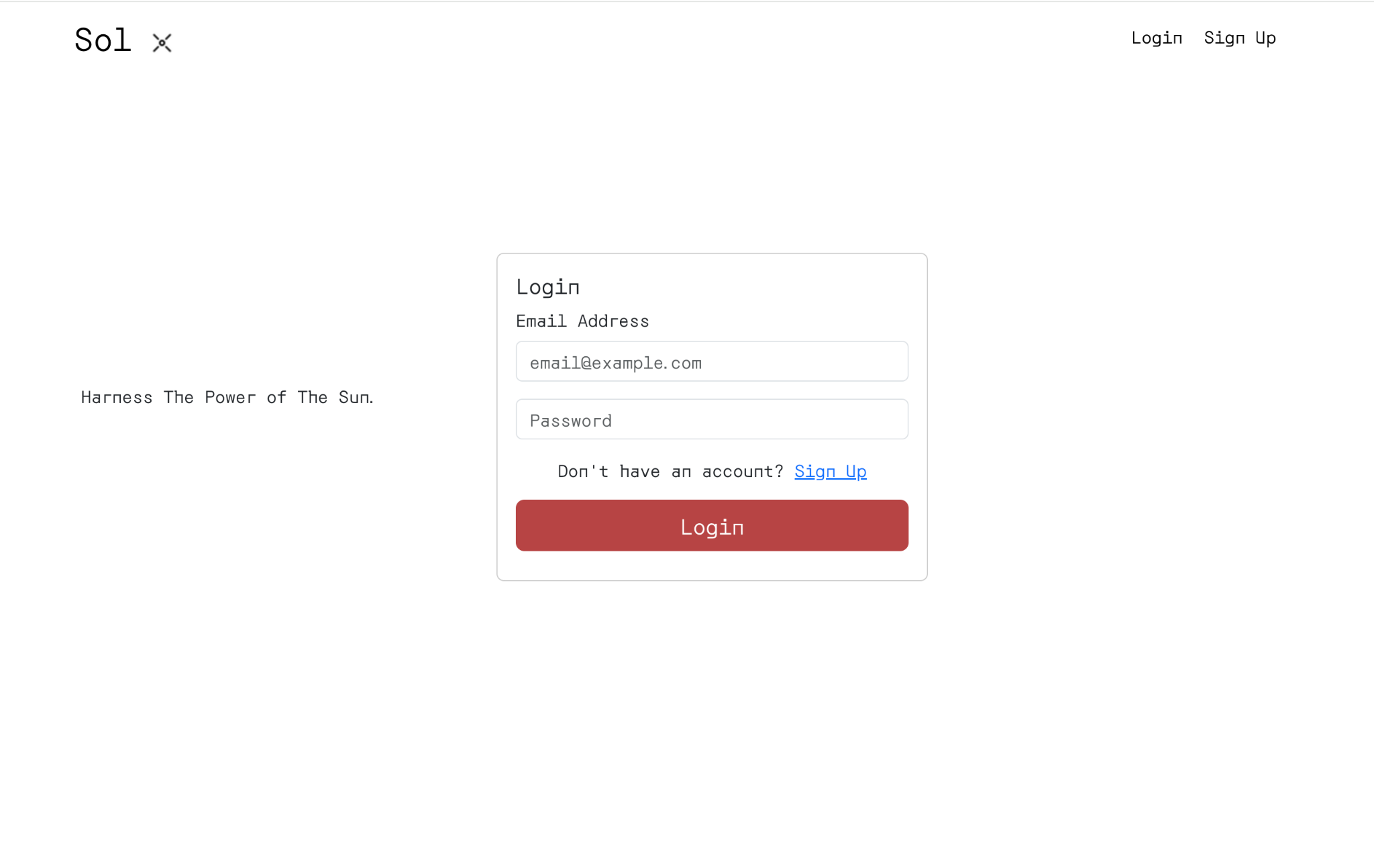
Figure (1): Diagram of the system overview

The system proposed for implementation comprises a solar power plant, a monitoring system, a control system, and a web application. The proposed solar power plant will include a parallel connection of two bifacial solar panels, an energy storage system such as a 12V lead acid battery, and a charge controller. Each bifacial panel has a power output rating of 40W and is equipped with a reflector module that is positioned next to it. The reflector module rotates throughout the day to optimize the amount of solar irradiation received. The solar charge controller functions as a digital intermediary relay between the solar panel (SP) and the battery, as well as between the battery and the DC load which, in our situation, serves as a monitoring and control system. The monitoring system comprises the INA219 and DS18B20 sensors, which are utilized to assess several battery parameters such as voltage, discharge rate, and temperature. Additionally, it incorporates BH1750 to measure solar irradiation. The control system comprises 2 Nema 17 stepper motors, an A4988 motor driver, and 2 BH1750 light sensors. These components are utilized to autonomously monitor the solar irradiation and provide exact movement of the reflector to increase the solar panels' efficiency. The structure is equipped with the BH1750, motors along with the reflectors and face plates, while the monitoring and control system are controlled by a microcontroller such as the Raspberry Pi Pico W.



Figure(2): Diagram detailing the setup/integration of our sensors/reflector controls and the web server

The web application consists of three general parts: front end, back end, and microcontroller. The front end consists of a react.js application which has login and signup functionality, in addition to the ability to render data from our cloud-hosted database in the form of charts and tables. The React application also has an API based on the React Axios library that connects to the backend of the application. The backend consists of an express.js web server that handles requests from the front end and interfaces with our MongoDB, a cloud-hosted, database to fetch the necessary data. This database stores the application data sent to it by the Raspberry Pi Pico W microcontroller.

**2.2 User Interface** 

Figure(3): Screenshot of a web server application and user interface

Users can create an account and then log in to the application. They can then enter the solar array ID of the power module into the application to view the data and anomalies of the module. Once they have added the array to their account, their email address will be added to the solar array email list. This means that when anomalous behavior occurs they will be sent an email. All anomalies for the module are displayed in a paginated table shown above. The application also can display sensor data in graph format for the user. Users can see solar irradiation, power output, and temperature data for the power module throughout each day. Past days can be seen using the search function. The homepage displays the last measured values received by the monitoring board.

## 2.3 Physical description



Figure (4): Overall structure of the small-scale stand-alone solar plant

The main three parts of our project are the web server, the circuitry, and the reflector setup. While the web-based application does not exist in a physical sense, it receives its data from our physical setup. We have created a plant model to securely mount the solar panels with our light sensors, as well as the moving reflector panels. This setup consists of an aluminum frame, as well as several sliding faceplates powered by a stepper motor attached to a ball screw. The solar panel setup can be repeated and placed in parallel with each other to achieve an effect akin to bifacial solar panels.

The solar panels are connected to a solar charge controller which also connects the battery and monitoring and control circuit. This configuration ensures that the system is operating well within safe parameters, as well as letting users check in on energy collected by the solar array from afar. We also have a circuit designed to control the motion of the motor and ball screw, which is housed with our control circuitry.

All circuits, including the solar charge controller, will be housed within a storage box, with each circuit having its dedicated housing. The housing will be equipped with a compact lock for safety and will be impervious to water.

## Installation, setup, and Support

First, place the unit in an open area with direct access to sunlight and access to Wi-Fi, such as a flat rooftop. If using the array(s) of units, place units in parallel, with solar panels of consecutive units back-to-back. Each Solar Charge Controller (SCC) is equipped with three pairs of ports. The first two pairs serve as inputs for the solar panel and battery, while the last pair functions as an output port for the load. The solar panels must be linked to the solar input of the SCC in a manner that ensures positive connections to positive and negative connections to the ground. In the same manner, the battery will be linked to the battery input, and subsequently, our load, which in this instance pertains to the monitoring and control system, will be connected to the load port. The load port has a maximum output of 120W and receives its input directly from the battery. If the user disconnects everything from the SCC for any troubleshooting purpose, make sure you follow this order: Connection to battery → Solar Panel→ Load, otherwise, it will damage the SCC in the long term.

The INA219, BH1750, and DS18B20 components of our monitoring system and control system will establish communication with the Pico W through their respective VCC, 3V3, GND, SCL, and SDA pins. The control system, similar to the connection principle of the monitoring system; the A4988 driver functions as an intermediary device between the Pico W and the motor. It executes the motor's operations following the instructions provided by Pico W. Initially, the load's positive output from SCC will be linked to the Vin+ terminal of the INA219 sensor, while the negative terminal will be connected to GND. Subsequently, Vin-, with a 12V output contingent upon the battery input voltage, will be utilized to supply power to the remaining circuits, encompassing the control system. A detailed connection can be seen from the individual circuit diagram in Figures 5 and 6. To power up the Pico W, the user will be using a 12V to 5V converter and the user must upload the correct library and main.py file to the Pico W from the GitHub repository to make the overall system work.

For the housing of the circuits, a finger joint box will be used inside a storage box. The user should install each PCB to its corresponding housing, and ensure clear routing paths. The reflector system should be installed to the mounting frame first, installing the horizontal and vertical slider plates, then be attached to the ballscrew(s). Following this, the reflectors themselves can be installed with bolts to the L-brackets that are on the faceplates. The user should ensure that the L-brackets can rotate smoothly about their axis.

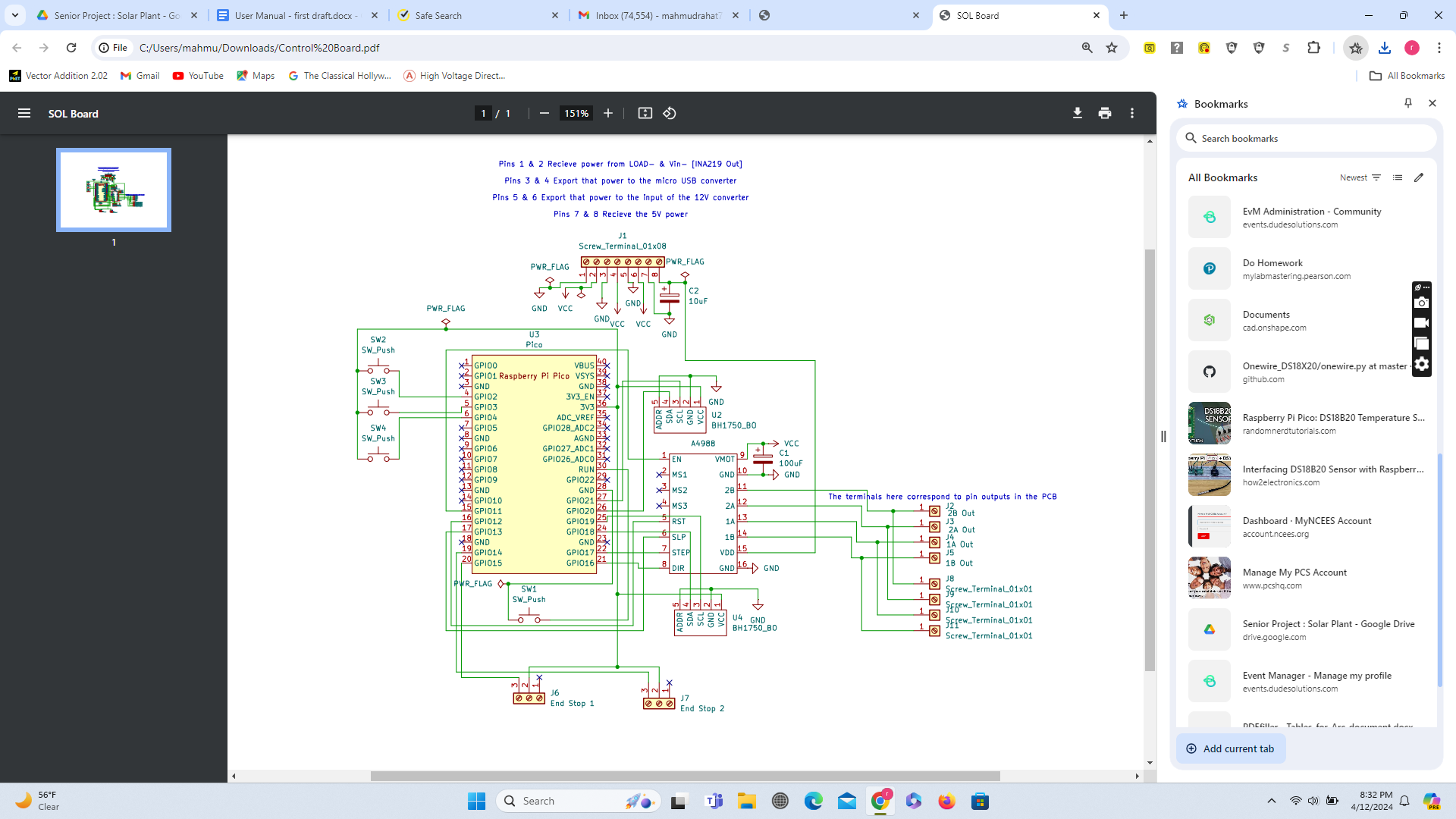
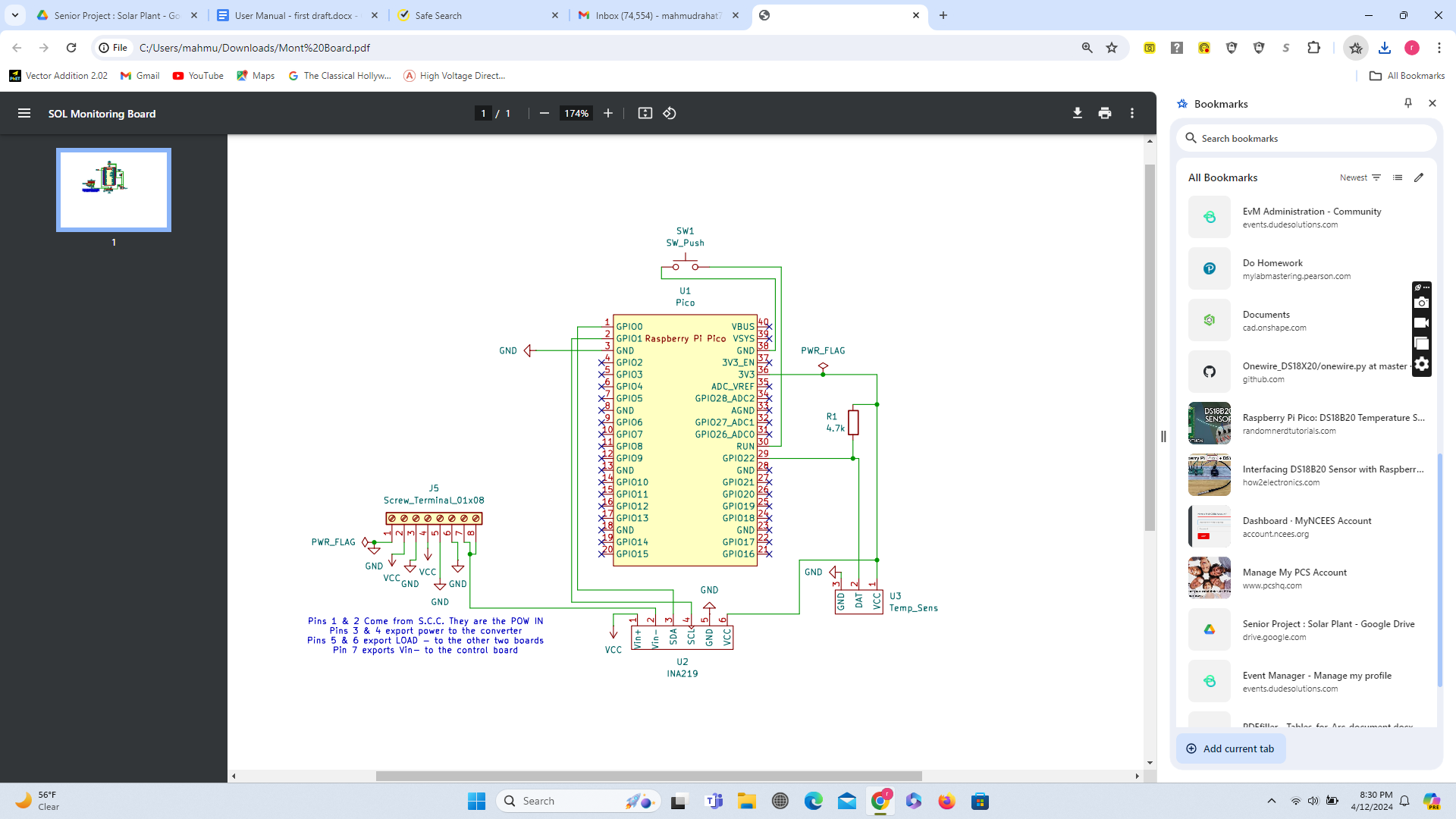


Figure (5): Monitoring Circuit Figure(6): Control Circuit

# Operation of the Project

## Operating Mode 1: Normal Operation

1. Once the Pico W is powered up by the converter, the system will automatically initialize. The stepper motors will first move the reflectors to their bottom-most position until they activate the bottom limit switch.
2. After initialization, the reflectors will try to find the optimal position for maximizing incident light on the solar panels, every 20 minutes. The system determines this by moving the reflectors up or down a small distance, and continuing in the direction that results in a greater light sensor reading, until the change in light intensity sensed due to a change in reflector positioning no longer exceeds a set threshold.
3. Enter your solar arrays in your account on the web application, if not completed previously.

Continue to allow the system to run automatically, checking the status and analytics of solar irradiation, discharge rates, and battery status on the remote application as desired.

Web Application

The web application will be usable by creating an account and logging in to the website. The solar array data can be accessed by adding the solar array ID to the application.

## Operating Mode 2: Abnormal Operations

Manual Reflector Control Mode

For setup, testing, demonstration, or other purposes, users can activate the system’s manual reflector control mode at any time.

1. To enable manual control mode, locate the PCB with four buttons within the housing. The four buttons are reset, mode selects, and directional input during manual mode.
2. Press the mode select button to activate the manual control mode.
3. Users can then press the two directional buttons to move the reflectors up and down.
4. Once finished with manual mode, users can push the mode select button again to continue the automatic light maximizing mode.

Control/Monitoring System Reset

If either the control system or monitoring system is behaving abnormally, users can press the reset button on the respective PCB in the housing. This will reboot the system of which the reset button has been pressed.

Other Abnormalities

If the monitoring or control system ceases to respond, such as when the user does not receive periodic updates from the web server, it is advisable to investigate the presence of any loose connections to the solar charge controller. If all connections are functioning properly, then it is most likely that the microcontroller is defective. To proceed, the user only needs to substitute the defective microcontroller and upload the suitable code and library from our GitHub repository to the new Pico W, which is provided in the appendix. For any other issues, users should contact the team at the emails listed in the appendix.

## Safety Issues

The most prevalent safety concern in any power generation project is battery safety. Always ensure the solar charge controller and fuse are in good condition and replace them when required. Additionally, while our wiring is achieved with PCBs and the circuits have been sealed in the housing, do not touch or handle the circuits or components while in use. Don’t use a battery of more than 25V as our monitoring system can only handle up to 26V and 3.2Amp

# 

# Technical Background

* 1. **Monitoring System**

The main goal of our project is to make the solar array for our client more functional. As such, our monitoring sensors will ideally be powered by the energy generated by the solar panels, with some left over to be stored in a battery. This influenced our decision to connect the monitoring system to the solar charge controller for the 5V required to power the circuit. We are also using a 12V to 5V converter to achieve this. The sensors we have selected are a BH1750 to measure incoming light, an INA219 for current and voltage, and a DS18B20 for temperature. Our sensors send their data through two Raspberry Pi Pico W microcontrollers, which then wirelessly transmit the data to be displayed on the web server.

The BH1750 device cannot directly quantify solar irradiation. Solar irradiation refers to the quantity of sunshine that is received by a solar panel per unit area. The measurement of solar irradiation can be conducted indirectly through the utilization of multiple variables. These variables include lux, which represents the intensity of light, clearness index, which indicates the level of brightness or cloudiness of the day based on lux, and angle of incidence adjustment, which is contingent upon the geographical location. In our specific case, we are employing the average angle for Boston, thereby utilizing the BH1750 instrument to facilitate the measurement of solar irradiation. The underlying laws that govern our monitoring system are Ohm’s law(INA219), the principle of photoconductivity (BH1750) which belongs to the control system, and the concept that the voltage across the terminals of a diode varies in response to changes in temperature. The accurate measurement of temperature by the DS18B20 is facilitated by the relationship represented by the diode equation and finally, the Pico W relies on the principle of digital electronics.

Several Python libraries and programs have been developed in micropython so that the sensors and microcontroller can communicate with each other and transmit the data to the server.

The decision to use MongoDB to host our web server was influenced by the need for a reliable/stable way to store large amounts of data. We have designed this remote monitoring system so that it is not only capable of maintaining records on and detecting anomalies in one solar array but can show and store data for numerous combinations of solar energy collecting systems for different user accounts. As such, having a large storage capacity was important. Additionally, this allowed us to implement our user interface on a cloud-based database that can be accessed from a standard browser tab; making the final webserver design easily accessible across devices.

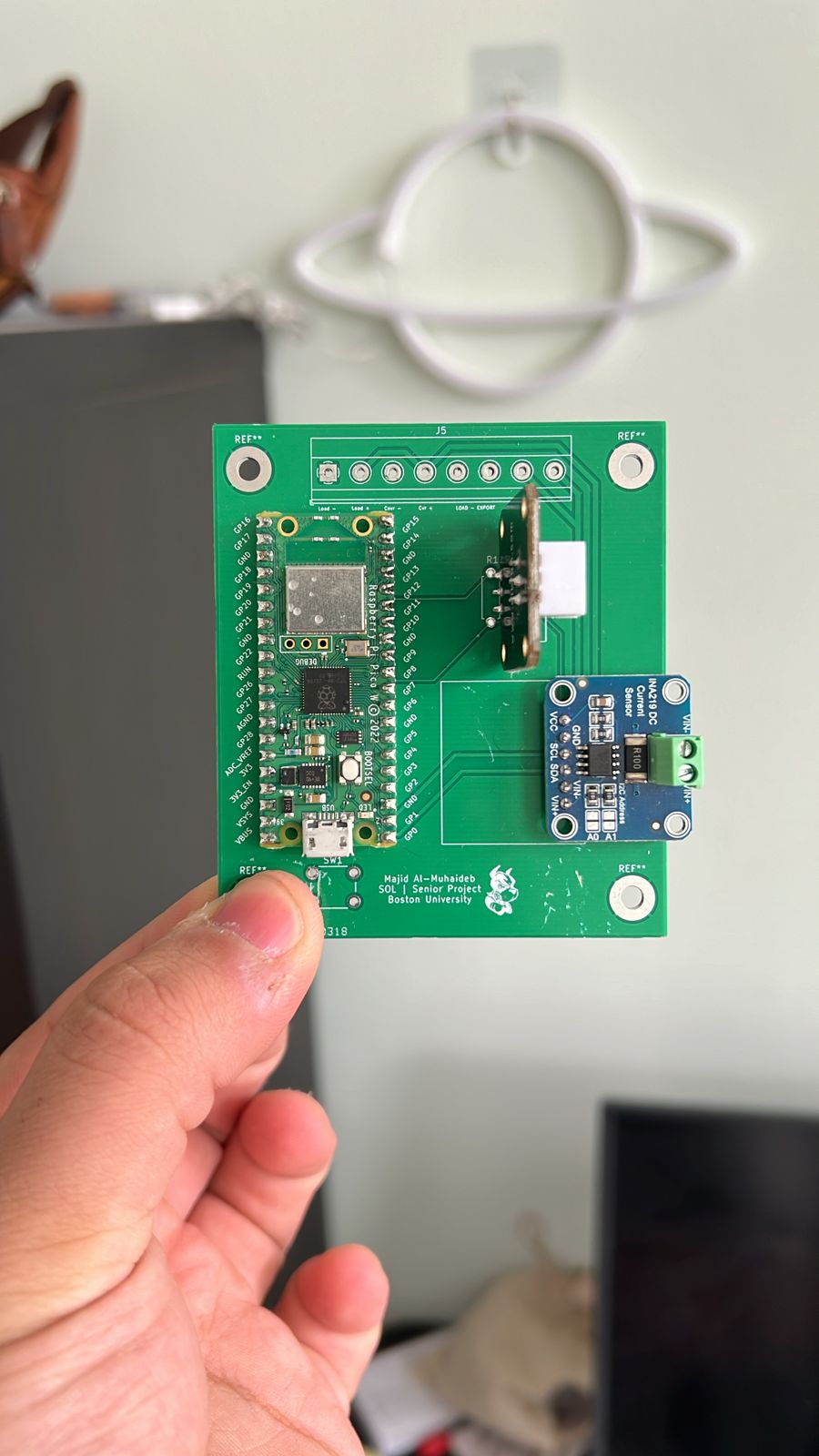


Figure (7): Monitoring system PCB

**4.2 Reflector System**

One of the aims of our project is to explore an alternative setup configuration for solar panels. Instead of tracking solar panels, a popular setup scheme in non-equatorial countries, our setup opts to have moving reflectors that shine light on stationary solar panels. The benefits of such a setup include possible space conservation, cheaper repair costs, and robust operation due to the reduced weight of the moving parts.

The mechanisms of our reflectors operate similarly to the movement seen in 3D Printers, where bearings move across linear rails while being driven by ball screws connected to stepper motors. Our system consists of 3D Printed face-plates, as well as mounting attachments for the ballscrew, the motor, and the limit switches. Following installation, the system runs through a limit test, which can be read in further detail in the controls section.

The reflectors, which are attached to the faceplates, can rotate about an axis via attached ball bearings. This rotation, coupled with the movement of the faceplates on the rails, facilitates movements of the reflectors along a radial path.

The system itself consists mostly of 3D-printed parts. The parts are printed in PLA, which affords the greatest flexibility in printing setup, and is a sensible choice since the parts do not undergo constant stress. However, using PETG or another strong filament such as ABS can assist in reliability, particularly on the rod-faceplate ring.

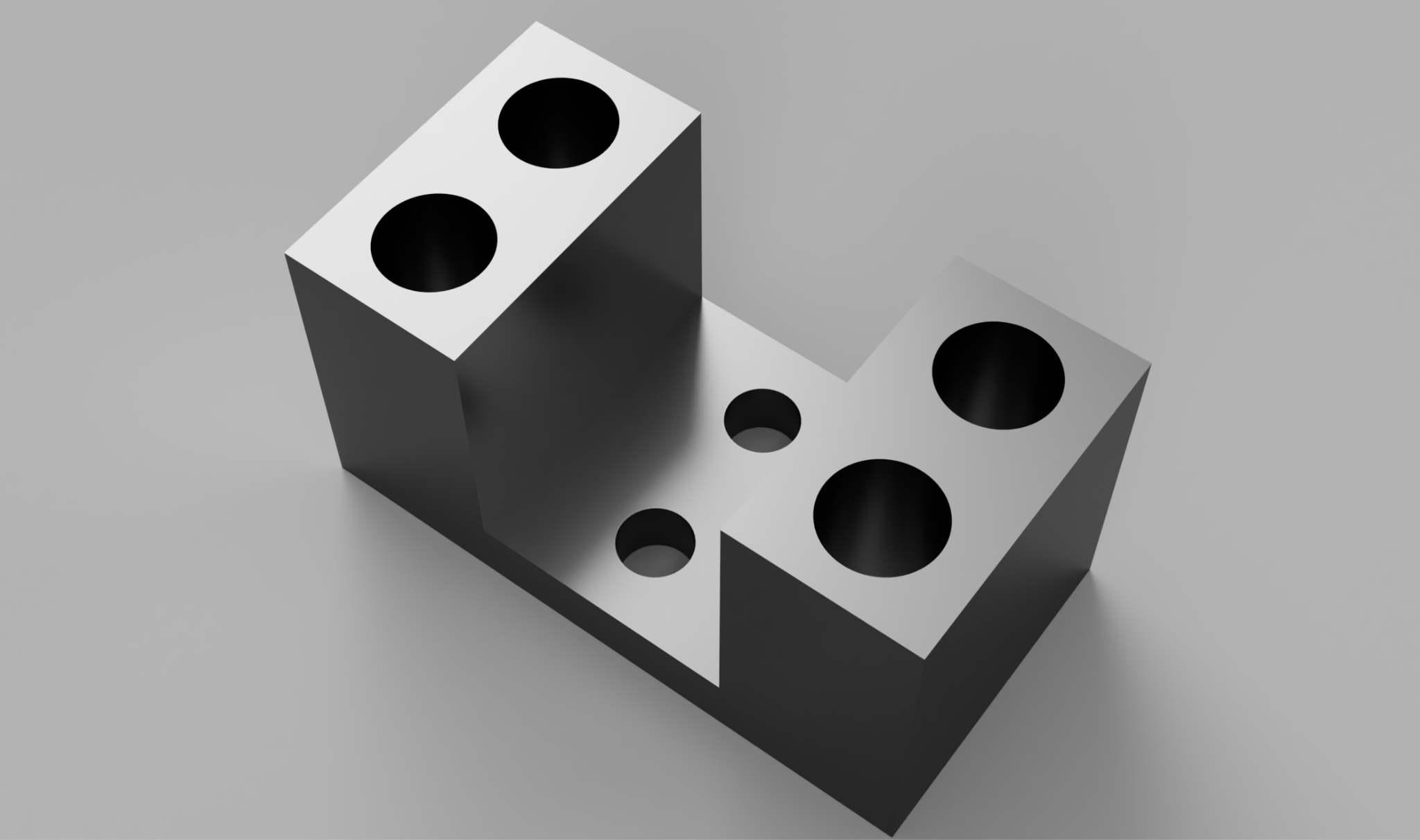
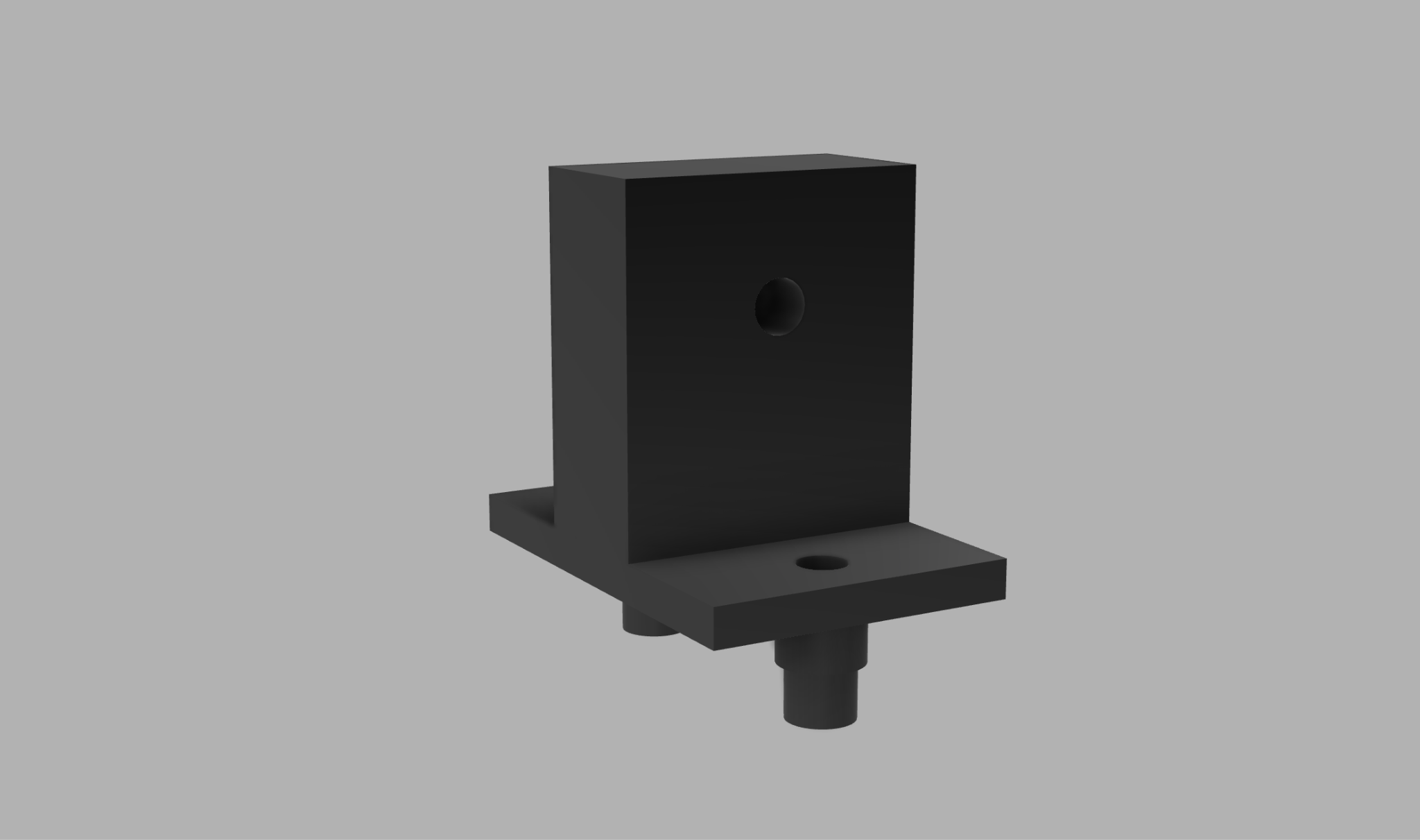
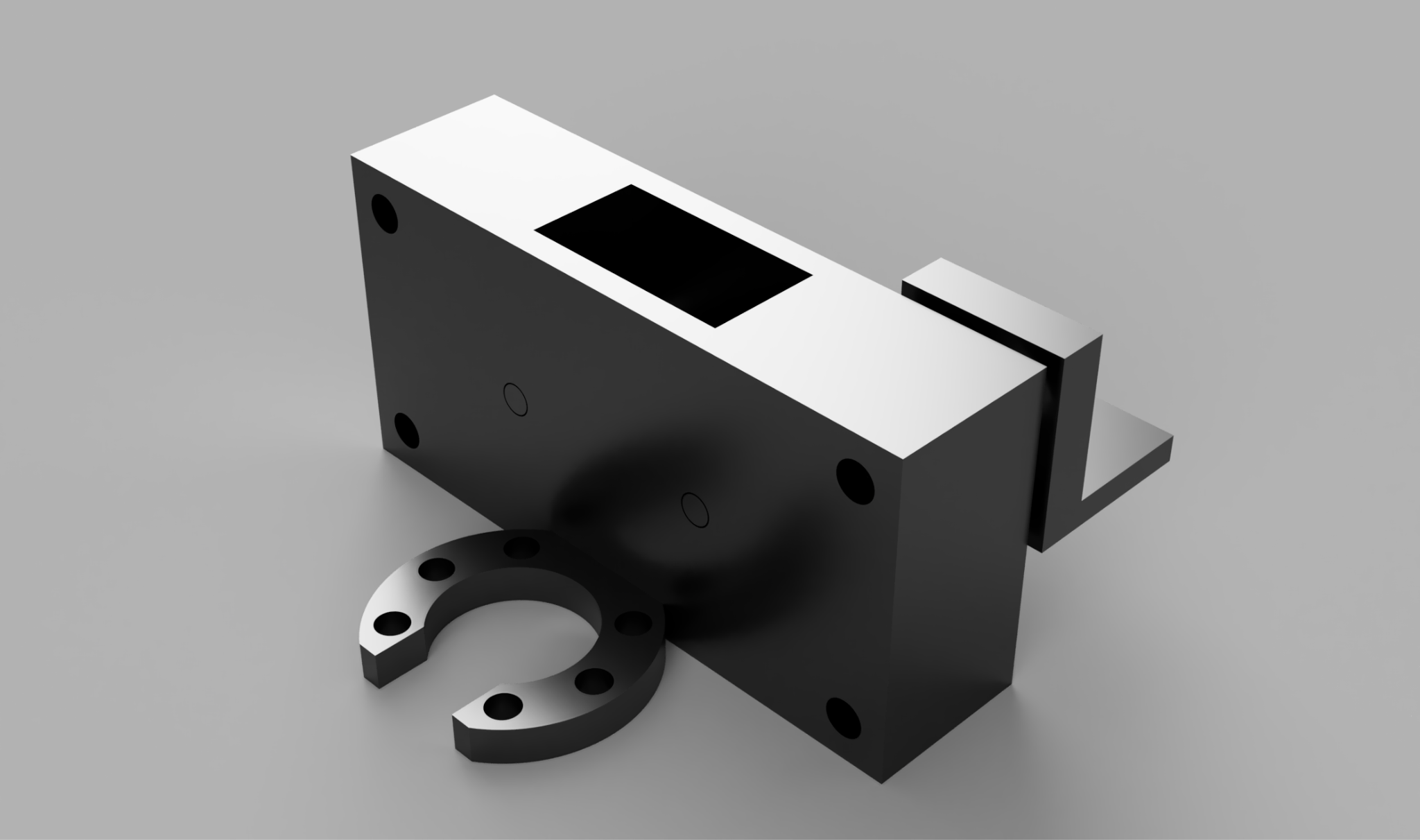
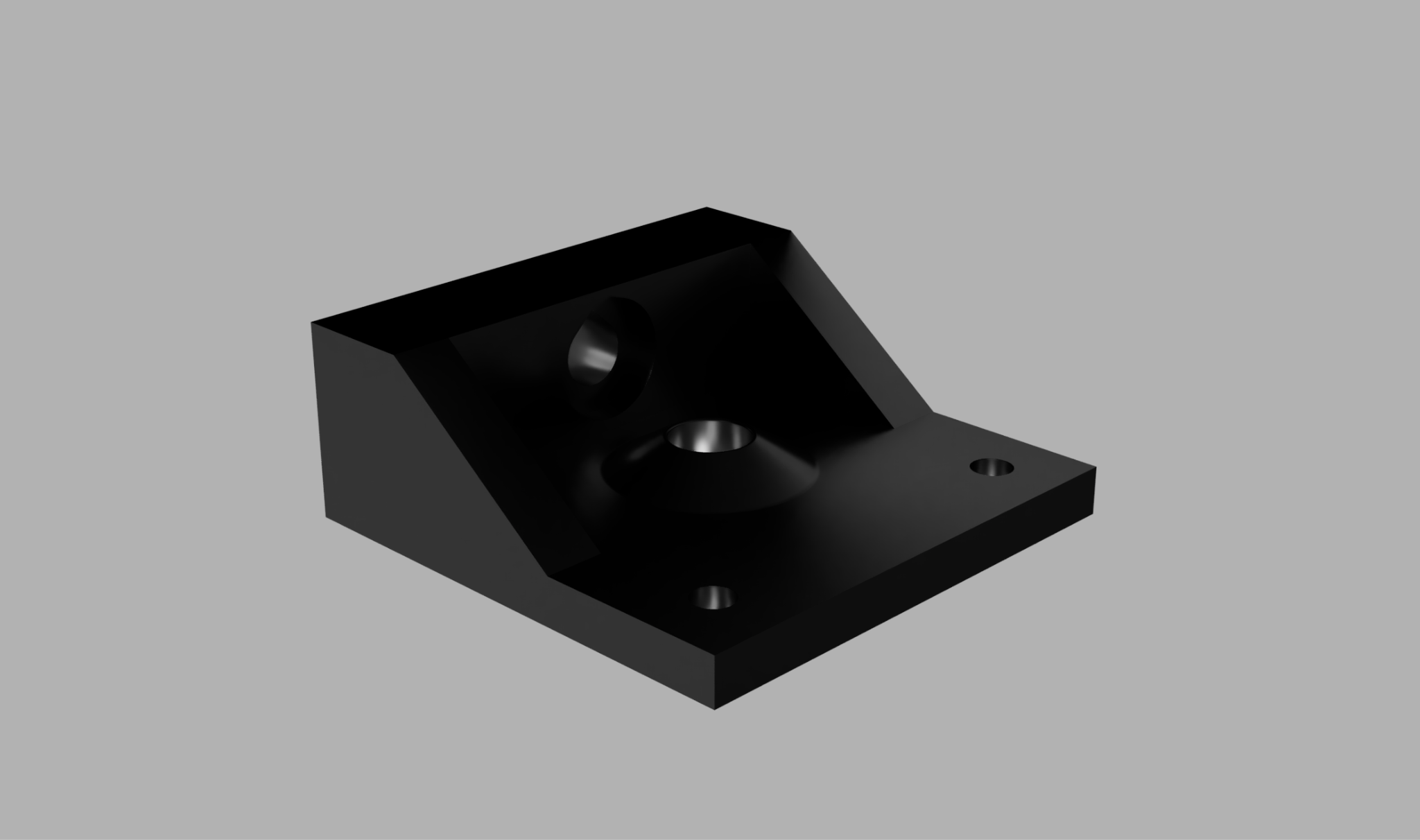


Figure (8): Some of the 3D-printed parts of the structure

**4.3 Control system**

The control system is written in MicroPython and consists of an automatic module and a manual module. Upon startup, the automatic module calibrates the reflector positioning by setting the reflectors down until they activate the bottom limit switch. From here, the system can keep track of the reflector positioning using the steps rotated by the stepper motor.

After calibration, the automatic control moves the reflectors up and down to find the optimal position for incident light on the solar panels, using feedback from the light sensors. It does so every 20 minutes by first moving the reflectors a set amount up or down based on the previous cycle and whether they are at the limits, and determining whether the change in reading from the light sensors exceeds a certain threshold. If it does, the reflectors continue moving in the same direction or change direction, depending on whether the initial movement resulted in a higher or lower incident light, respectively. This continues until the change in incident light no longer exceeds the threshold, in which case the system will have found the optimal position. After this position is found, the motors turn off to save energy during inactivity.

The manual module enables the user to adjust the reflector positioning manually, for setup, testing, demonstration, etc. This is done by using an interrupt to activate a flag when the mode selection button is pressed and having the main module check for this flag iteratively throughout the program. If the flag is found true, it calls the manual motor function and stays within the loop of this function until the mode selection button is pressed again, setting the flag back to false and returning to the main function.

Similarly, the limit switches utilize interrupts that immediately disable the motor upon activation, and set a flag representing either the top or bottom limit switch depending on which is hit. In the main program, the motors may then only be enabled if the direction of reflector movement is away from the limit switch it activated.

It should be noted that the solar irradiation component of the monitoring system runs in the control system’s main module on the same Pico W. In this way, as the control system utilizes the light sensor readings for reflector control, these readings are also sent to the database as part of the monitoring system.

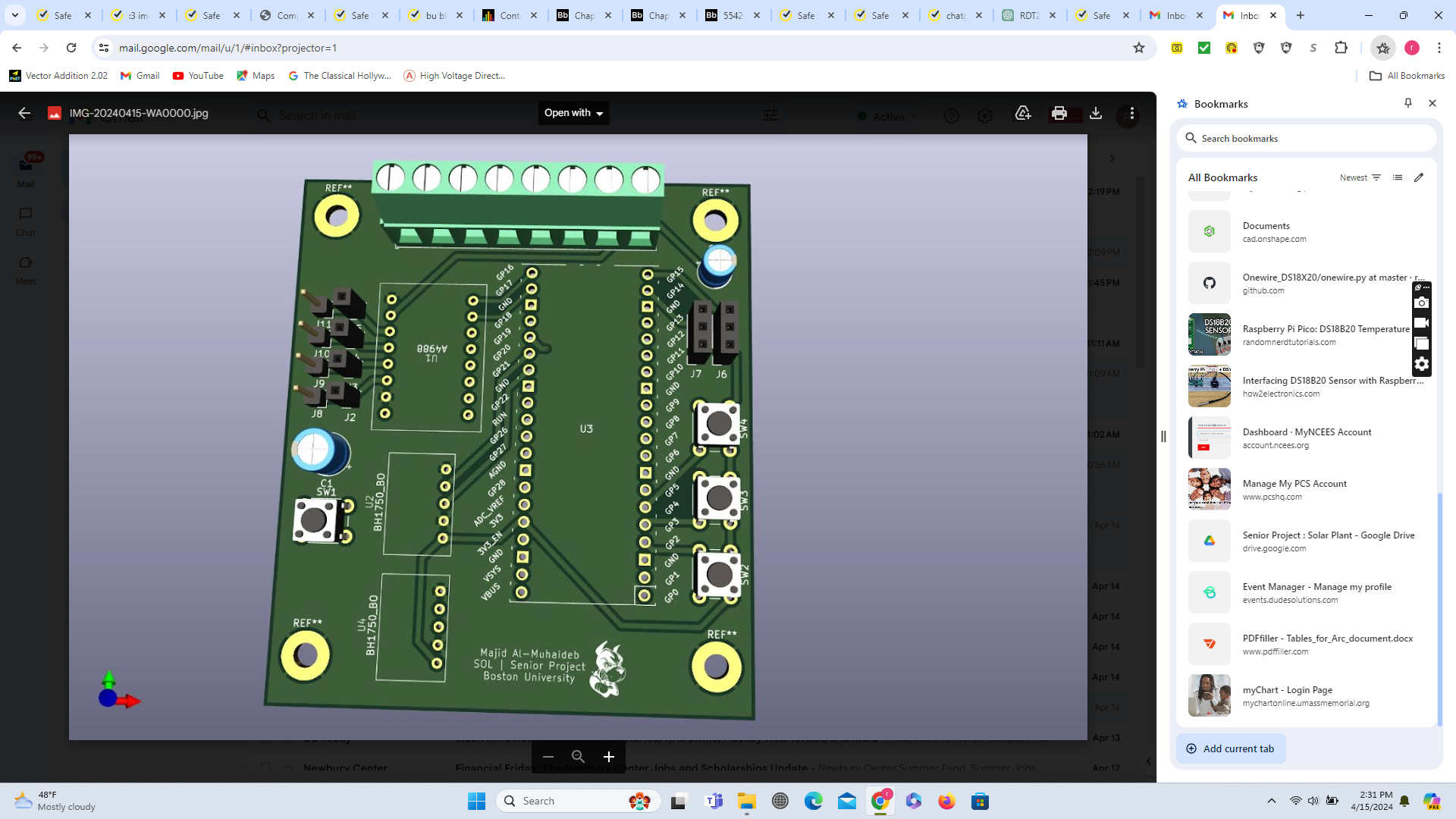
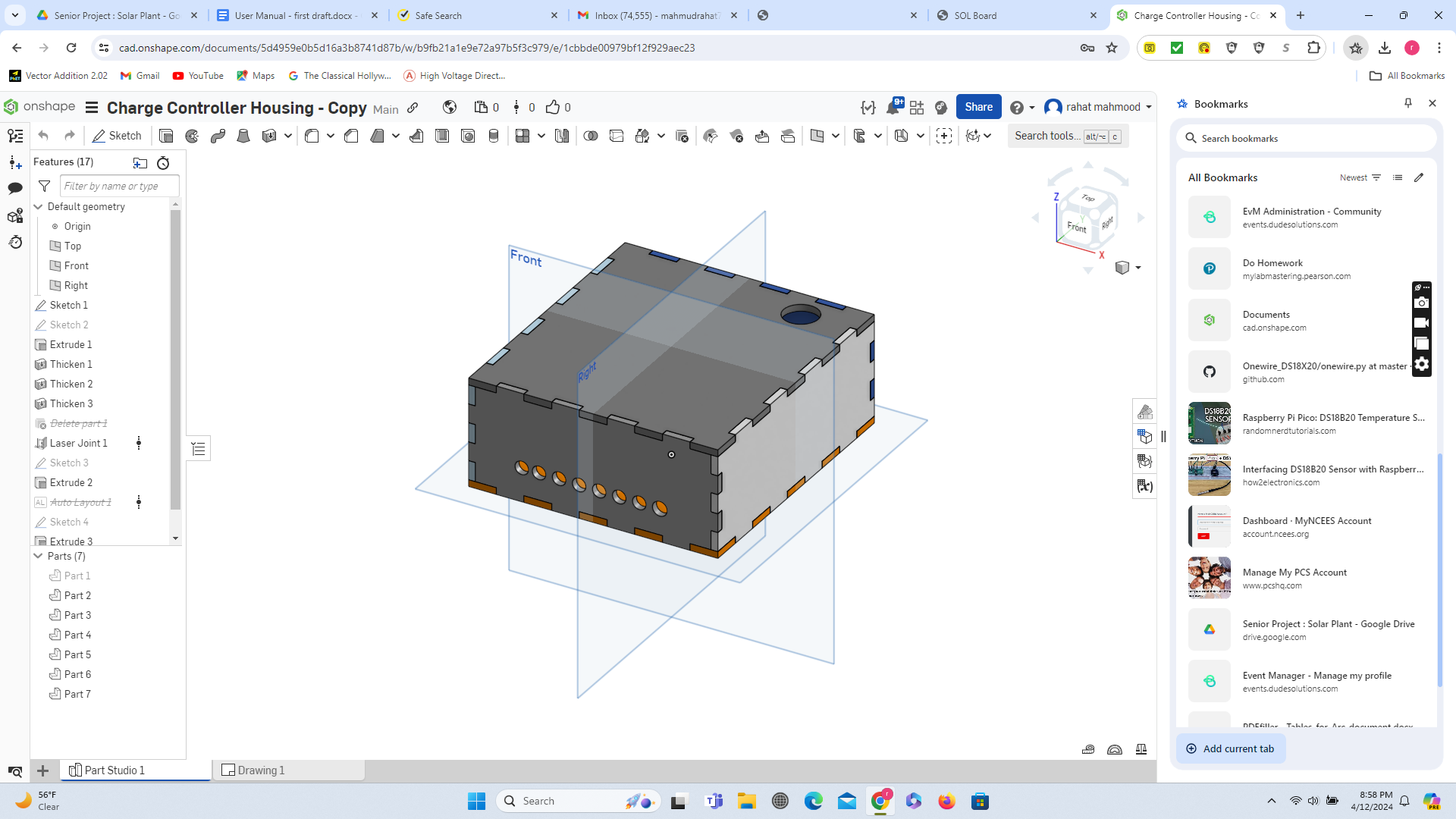
 

Figure (9): Limit Switches Figure (10): Control System PCB

**4.4 Housing**

The housing has been made using black acrylic sheets (12\*24, 1/8" thick), which have undergone laser cutting and subsequent bonding specifically designed for acrylics. The box utilized in this study is referred to as a finger joint box, and many videos sourced from YouTube have been used as references. To accommodate all the circuits and solar charge controllers, a 13Qt storage box has been employed. This box features several apertures on its cover, facilitating the placement of the motor and BH1750 external components. Housing has been designed based on the size of the parts and the law of mechanical stress and strain.

Figure (11): PCB Housing 

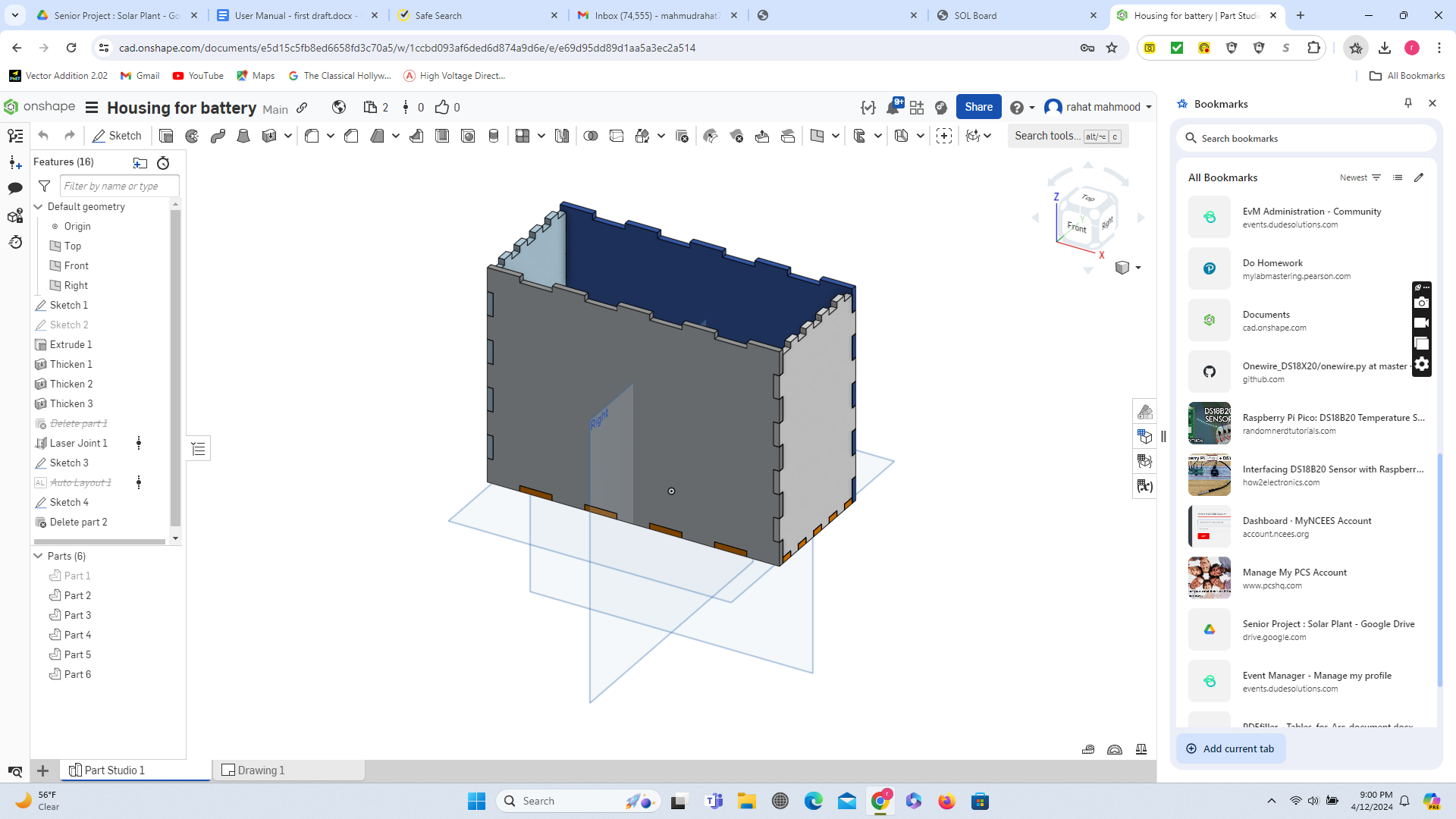
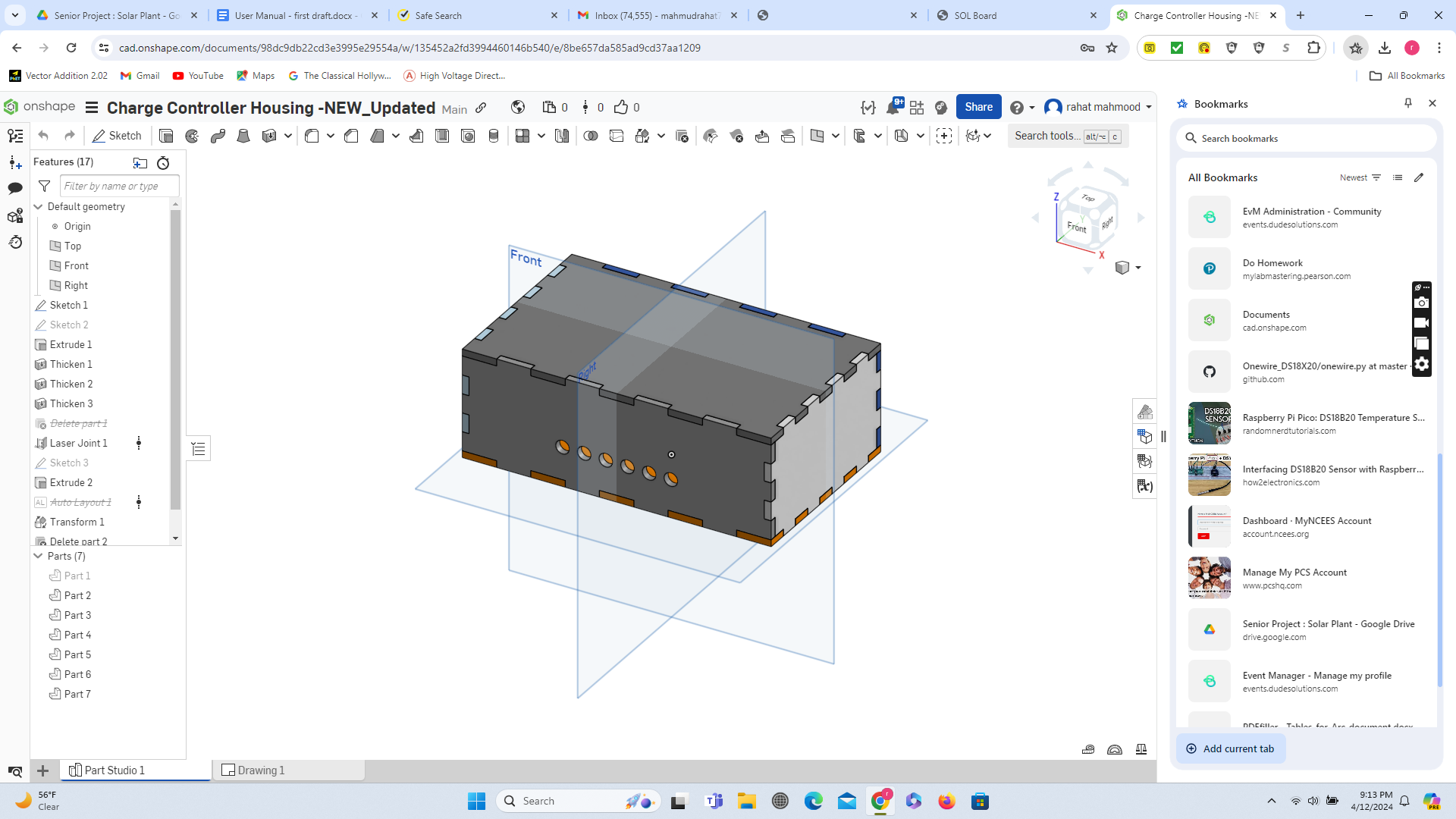
 

Figure (12): Battery Housing Figure (13): SCC Housing

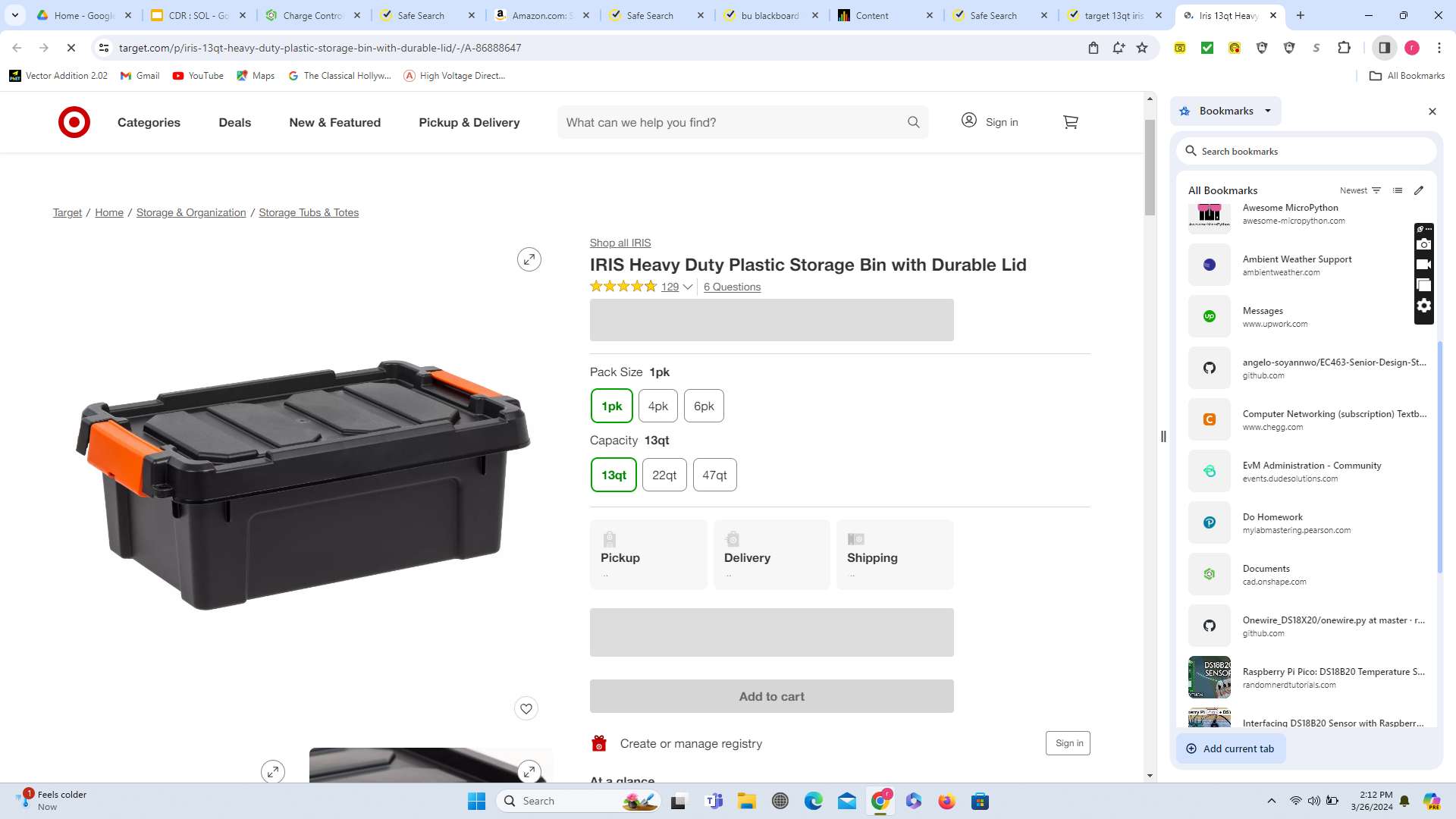


Figure (14): 13Qt storage box

**4.5 Web Server**

The Raspberry Pi Pico W utilizes various Python networking libraries, such as the network and urequest libraries, to establish a connection with the Internet and transmit sensor data to our database located on the MongoDB Cloud Atlas hosting service. This has been accomplished through the utilization of the service’s data API. The voltage, current, temperature, and solar irradiation are measured using an array of sensors. The GPIO pins on the Pico W and Python libraries, such as machine and sensors, are utilized to extract the values from these sensors.

The web application consists of a MERN stack app, which includes MongoDB, express.js, react, and node.js. It will be hosted on a cloud hosting provider like digitalOcean or Heroku. The app utilizes a cluster on the MongoDB Cloud Atlas hosting platform to store the data from the Pico W. In this manner, the microcontroller is capable of transmitting data to the database directly from any IP address, as per the configuration chosen on the web platform. The web application utilizes an express.js server for its server side. This server manages all API calls made from the front end (React) side, including get, post, put, and head requests. The express.js server establishes communication with the database housed on our cloud platform. ExpressJS, a framework derived from Node.js, is an asynchronous runtime environment that facilitates the development of network applications. The client side of the system will comprise a client application developed using the react.js framework, which is structured as HTML. The react client will interact with the express.js web server by utilizing the Axios javascript requests module to initiate requests. Subsequently, the web server will transmit data to the client to render and present it to the user.

The utilization of sensor data from our monitoring system is crucial for doing trend analysis and detecting anomalies. To incorporate charts into our trend analysis page, we will utilize the chart embedding SDK provided by MongoDB. Anomalies are identified using a control loop implemented on the Pico W, which has predetermined criteria to ensure regular functioning. A notification will be dispatched to our users once certain levels are exceeded. To do this, we use the MicroPython umail package to send emails.

# Relevant Engineering Standards

The most relevant engineering standards for our project are:

**1. Electrical Design and Safety**

**IEEE 1547 (Standard for Interconnecting Distributed Resources with Electric Power Systems):** This standard is relevant for the integration of your solar panels with the electrical grid, ensuring safe and reliable connection.

**National Electrical Code (NEC)**: This code is particularly important for electrical wiring, safety, and installation practices. It covers the installation of electrical conductors, equipment, and raceways; signaling and communications conductors; and optical fiber cables.

**IEC 62109-1 (Safety of power converters for use in photovoltaic power systems - Part 1):** This is relevant for the solar charge controller, ensuring it meets safety requirements for photovoltaic systems.

**2. Electronic Component Standards**

**IEC 61000 (Electromagnetic compatibility (EMC):** It ensures that the system complies with EMC standards which is essential to prevent electromagnetic interference with other devices.

**3. Wi-Fi Communication**

The following standards are relevant for the wireless communication aspect of our project, especially since we are using the Pico W with Wi-Fi capabilities:

**IEEE 802.11:** This is the set of standards for wireless networking, commonly known as Wi-Fi. Ensuring our device's Wi-Fi module is compliant with the IEEE 802.11 standards is crucial for compatibility, security, and efficient communication. These standards cover various aspects, including data encryption, network bandwidth, and communication protocols.

**4. Python Coding Standards**

Since our project's code is written in Python, adhering to Python-specific standards is important for code maintainability, readability, and reliability.

**PEP 8:** This is the official Python style guide, providing conventions for writing readable and consistent Python code. It covers aspects such as naming conventions, indentation, line length, whitespace usage, and comments. Following PEP 8 ensures that the code written is Pythonic and maintainable, which is especially important for long-term projects and collaborative work.

**PEP 20 (The Zen of Python):** While not a strict standard, PEP 20 outlines a set of principles for writing computer programs in Python, emphasizing simplicity, readability, and the importance of writing explicit and beautiful code. It's a good set of guiding principles for Python developers.

**5. Internet Protocols**

For the data communication between our microcontroller and the MongoDB web server, understanding and implementing the correct internet protocols is crucial:

**HTTP/HTTPS:** Our system communicates over the Web, so understanding the Hypertext Transfer Protocol (Secure) is essential. HTTPS, the secure version of HTTP, is particularly important for encrypting data in transit, protecting against eavesdropping and tampering.

**MQTT:** For IoT devices like ours, MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol designed for limited bandwidth and device resources. It's efficient for device-to-server communication, making it a popular choice for IoT projects.

**6. Information Security Standards**

Given that our project involves network communication, ensuring data security is paramount:

**ISO/IEC 27001**: This is a widely recognized standard for managing information security. It provides a framework for setting up an Information Security Management System (ISMS), ensuring the confidentiality, integrity, and availability of data. This standard is particularly relevant when our system handles sensitive or personal data.

**OWASP Top 10**: The Open Web Application Security Project (OWASP) Top 10 is a regularly updated report outlining the most critical security risks to web applications. While not a standard, it's a valuable resource for understanding and mitigating common web application vulnerabilities.

# Cost Breakdown

| Project Costs for Production of Beta Version (Next Unit after Prototype) | | | | |
| --- | --- | --- | --- | --- |
| Item | Quantity | Description | Unit Cost ($) | Extended Cost ($) |
| 1 - stepper motor | 2 | provide the required torque to move the reflectors together | 11.99 | 23.98 |
| 2 - ball screw & coupler | 2 | provide the required support to move the face plates up and down | 42.38 | 84.76 |
| 3 - aluminum frame | 1 | held together with screws and brackets, composed of 20mm aluminum bars | 175 | 175 |
| 4 - monitoring system/control system/electrical components | 1 | includes printed PCBs with all sensors, microcontrollers, wiring/connections | 135 | 135 |
| 5 - limit switches | 4 | prevent motor burnout if the system becomes offset, is activated by the reflectors at the maximum and minimum angled positions | 1.50 | 6 |
| 9 - solar panels | 2 | generate power from collected light | 33.90 | 67.80 |
| 7 - reflector panels | 2 | reflect light onto the solar panels | 10 | 20 |
| 8 - reflector sliders | 8 | face plates and lower bar sliders consist of 3d printed components fitted with ball bearings that slide along the edges of the main aluminum frame | 3 | 24 |
| 9 - electrical housing | 1 | housing for electrical components to prevent exposure to the elements | 50 | 50 |
| 10.12V Pb Battery | 1 | Main power source and energy storage | 20 | 20 |
| 11. Solar Charge Controller | 1 | Digital relay between the battery, solar panels, and load. | 10 | 10 |
| 12. Fuse | 1 | battery safety | 3 | 3 |
| Beta Version-Total Cost | | | | $ 619.54 |

# Appendices

**Appendix 1.0 - Initial design of the control system**

The initial configuration of our control system from 2023 comprised an A4988 stepper motor driver, a Pico W microcontroller, an RTC DS3231 module, and a Nema 17 stepper motor. The control system was used to operate in a time-based manner to move the reflector gradually each hour as the sun moved from sunrise to sunset. After sunset, the system would return the reflector to its initial position. Finally, the system would cease operation from midnight until sunrise. The DS3231 had several advantages, such as a built-in button battery that enabled the module to autonomously keep time even when physically separated from the microcontroller. This eliminated the requirement for the user to reset the time each time power was lost to the module. The program for the initial system was written to allow users to independently adjust the speed and direction of the motor at various hours within a 24-hour time frame to eliminate the usage of any external device to detect the reflector’s position, but this required very precise testing and frequent changing of different variables. Another drawback of this technique was that if the user needed to make any changes in the future, they would need to connect the microcontroller to their laptop and directly modify the variables in the code. This could potentially cause inconvenience for the user. Users may continue to experiment with this or utilize it as a point of reference for any project involving the RTC DS3231 and Nema 17 stepper motor.

Later we also tried to improve the system using an LCD screen using a similar concept of RTC, but LCD wasn’t reliable to use in the long term. After much discussion and external advice from professors, the open-loop time-based infrastructure of the control system was replaced with the closed-loop light intensity maximizing system, described in the primary portion of this manual. All of these programs can be found in our GitHub in the updated control system code > old code folder, which is posted in the appendix with a link to our GitHub repository.

# Appendix 1.1 – Technical Specifications

| **Final deliverables** | **Performance** |
| --- | --- |
| Data storage | 100 MB of time series data storage |
| Power | Capable of measuring up to 83.2 W |
| Current | 0-3.2 Amps |
| Monitoring System | Capable of measuring up to 26V, 3.2A |
| Light Sensor | Measures up to 300 lux |
| Control System | Capable of moving reflectors through the desired range of angles. Provides 45 Ncm holding torque. |
| Reflectors | Reflect additional light onto the solar panels to increase energy absorption and overall efficiency by 7% |
| Number of concurrent users on the web application | 10-15 |
| Number of creatable Solar Array instances in our database | 100+ |
| Trend Analysis | Time series data like discharging power, temperature, and solar irradiation for the operation of the solar plant are present in the front-end application as a chart. |
| Notifications | The system notifies users through email of anomalous behavior. |
| Safety | The solar plant will make use of a Solar Charge Controller and fuse for the battery and overall system safety. The discharge stop is 11.5V, the discharge reconnects at 12.5V and the full voltage is 13.5V. The work mode is 24H and battery type b1. |

# Appendix 2 - Team Information

**Rahat Mahmood** primarily worked on monitoring, control systems, and housing. He previously worked with a consulting firm and utility for 5-20 MW solar plant design and high-level relay settings for power distribution. He will work at EVERSOURCE as a substation design and telecommunication engineer after graduation. He can be contacted at [rahat73@bu.edu](mailto:rahat73@bu.edu) or 857-389-5964

**Steven Cheng** is a senior Electrical Engineering student at Boston University. He has worked as a research assistant for Professor Joshua Semeter on the analysis of ionospheric data. He has also worked closely with Professor Min-Chang Lee in the development of the new Power Electronics for Energy Systems course. In March 2024, he completed his internship with Neural Dynamics Technologies as an embedded engineer. After graduation, Steven plans on moving to New London, Connecticut to work as a Systems Engineer for propulsion systems at General Dynamics Electric Boat. He can be contacted at [stcheng@bu.edu](mailto:stcheng@bu.edu) or 201-783-7390.

**Alexa Wiencek** is a senior undergraduate student majoring in Electrical Engineering at Boston University. They have worked as a Learning Assistant for Engineering Mechanics, including giving a guest lecture for the course, as well as having been a coordinator of the Artemis Project in the summer of 2022. They have worked for Boston University’s Electronics Design Facility as a student technician. Currently, they are the acting recording secretary for the Kappa Sigma chapter of IEEE-HKN. After completing their undergraduate degree, Alexa will continue their studies, working towards an MS in Electrical and Computer Engineering. They can be contacted at [awiencek@bu.edu](mailto:awiencek@bu.edu) or 646-531-2139.

**Angelo Soyannwo** is an undergraduate student at Boston University studying electrical and electronics engineering. He has previous experience working as an intern at James Cubitt Architects as well as First Class Group Limited. He has plans to incorporate his own company after graduation and is currently on track to complete his degree in the Fall of 2024. He can be contacted at [seun@bu.edu](mailto:seun@bu.edu) or 929-888-3930.

**Majid Almuhaideb** is a senior undergraduate student at Boston University, majoring in Electrical Engineering. He previously worked as a Teaching Assistant for Electric Circuits. He previously worked as an intern in management consulting firm McKinsey & Co. and plans to return after graduation. Majid is set to graduate in the Fall of 2024. He can be contacted at [majidmuh@bu.edu](mailto:majidmuh@bu.edu)

**References:**

* Github - <https://github.com/angelo-soyannwo/EC463-Senior-Design-Stand-Alone-Solar-Project-Github>