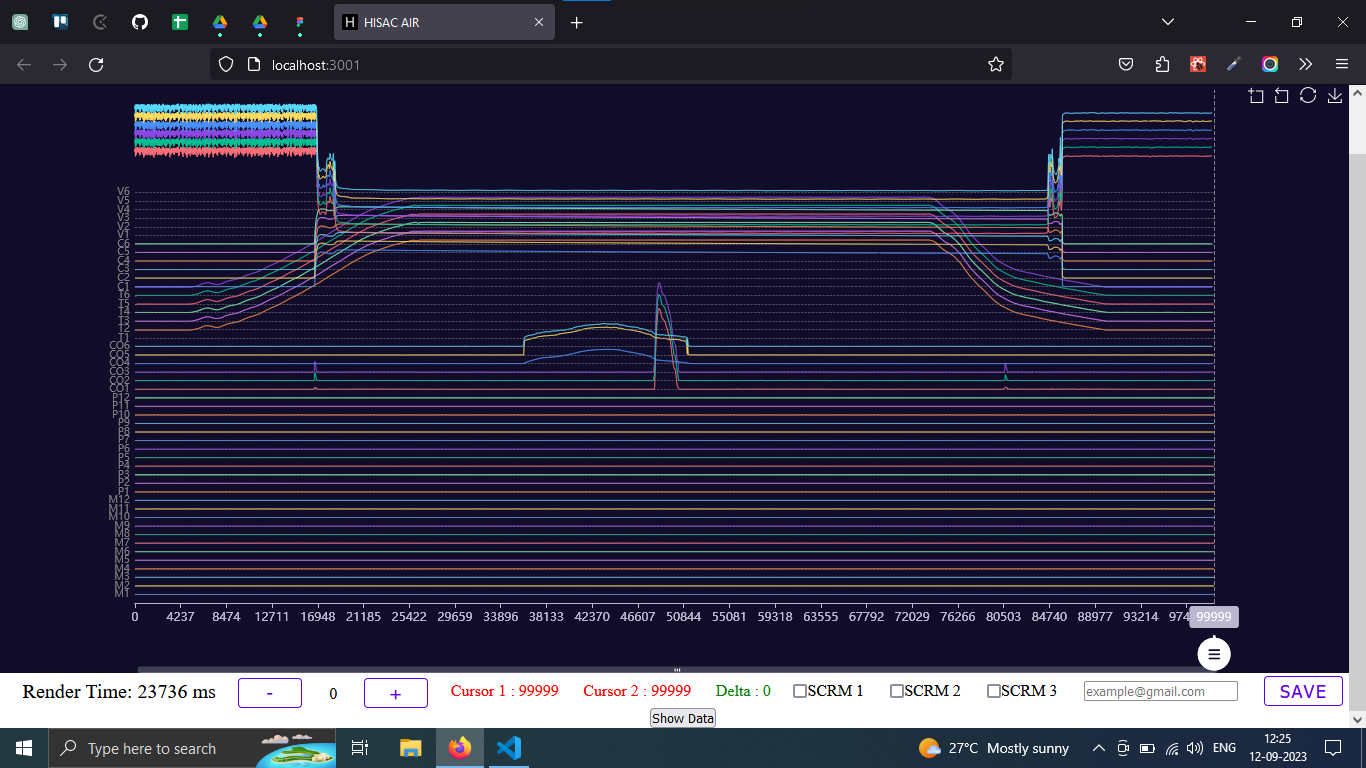
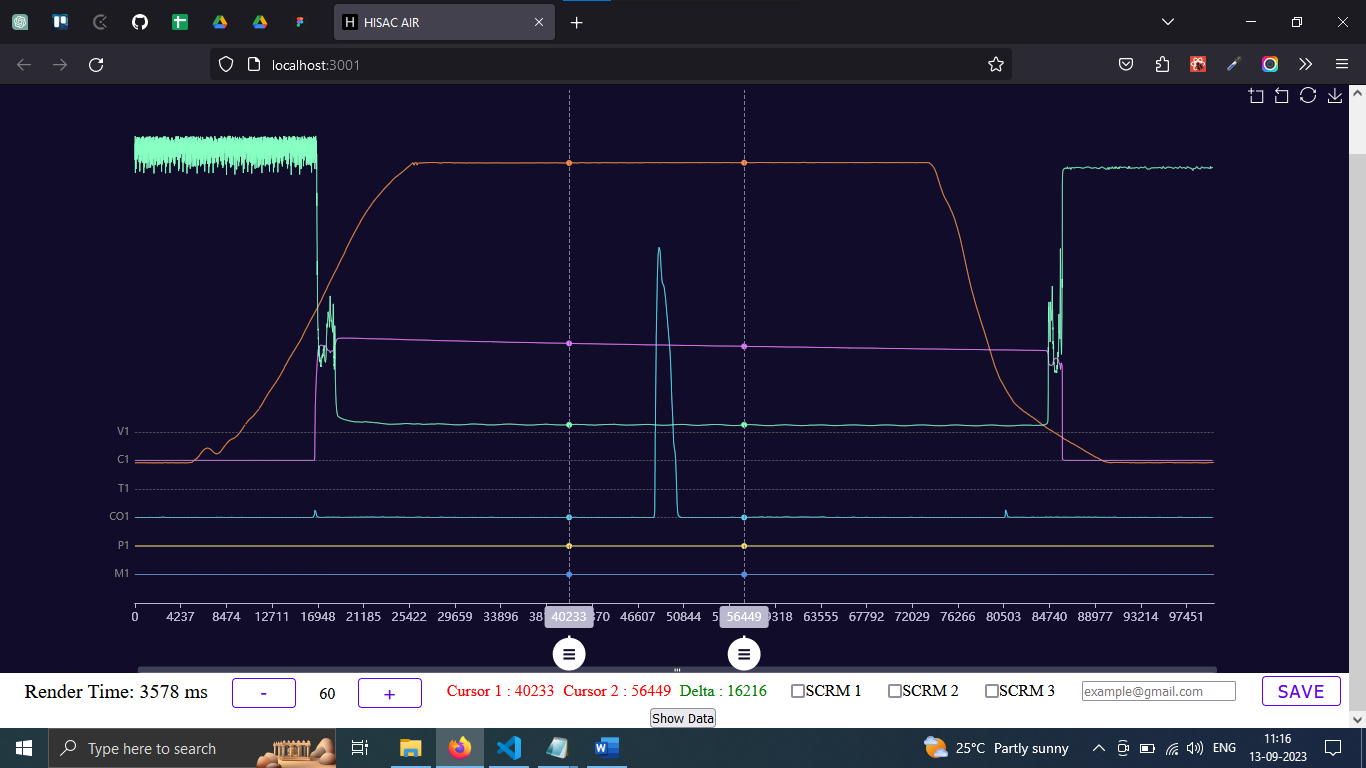
Project Title :- **Hisac Air**Aim :-  
The aim of this project is to develop a data visualization app capable of handling large datasets, optimizing API performance, and ensuring a seamless user experience.  
  
Description :-  
This project involves the POC of a data visualization app that can efficiently plot 42 channels of data, each with a large number of data points(1,00,000), optimize API code for hardware communication, and troubleshoot performance issues.  
  
Overview :-

The project timeline spans from July 03, 2023, to September 12, 2023. It consists of multiple tasks, each contributing to the overall functionality and performance of the application. Here's a summary of key accomplishments and developments.  
  
Frontend :-

1. Initial Plotting (12-09-23) :- 42 channels of data with 100,000 data points plotted using SP Line, Apache Chart.  


2. Cursor Functionality (11-07-23) :- Implemented crosshair functionality and aligned cursors.  


3. Pan-X (13-07-23) :- Implemented Pan-X for data exploration.

4. 2 Cursor (13-07-23) :- Tested various features, including graph, Zoom, Pan-X, and 2 Cursor functionality.

5. Baseline (08-08-23) :- Implemented baseline functionality.  
A screenshot of a computer

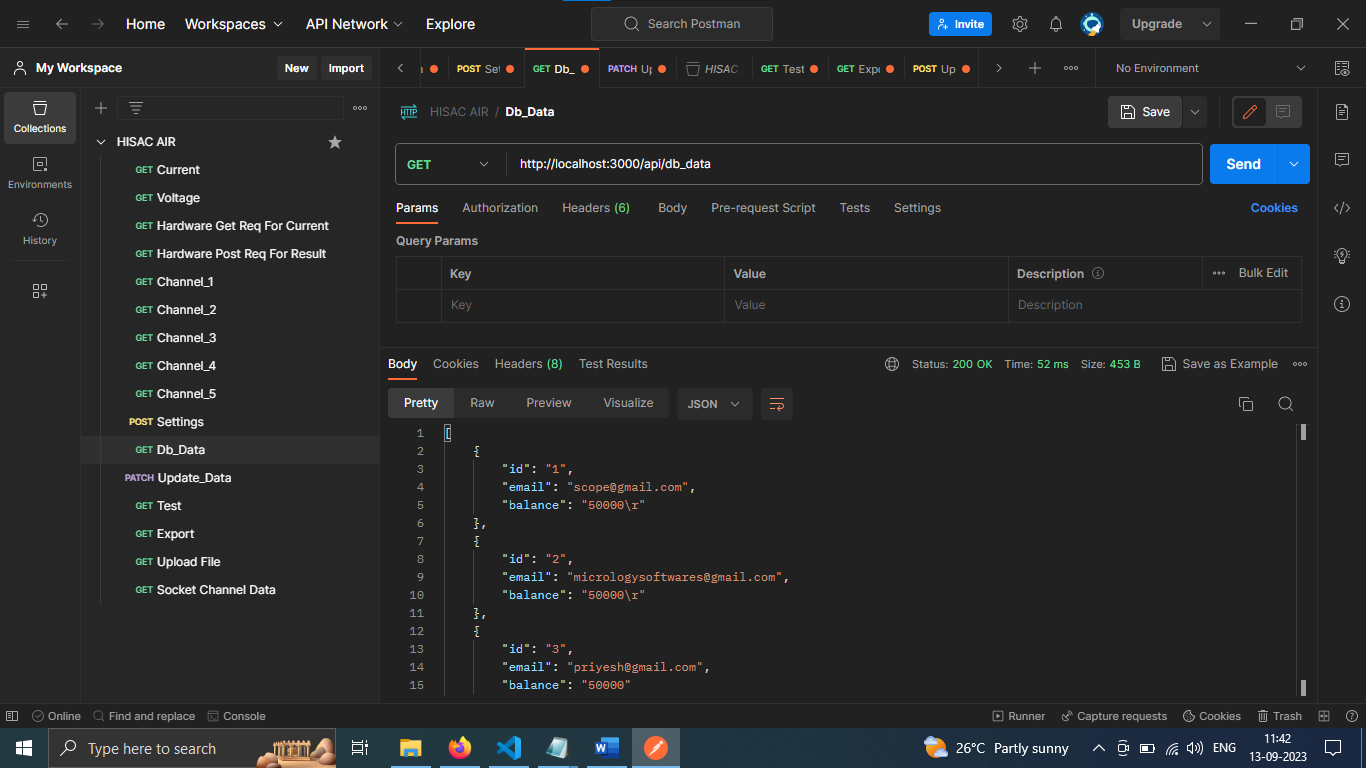
Description automatically generated

6. Delta Calculations (17-07-23) :- Calculated delta values, displayed them in the UI, and managed application state.  
  
  
7. Slider Implementation (31-07-23) :- Introduced a slider functionality to improve initial data loading performance.

A screenshot of a computer

Description automatically generated

Backend :-

1. API Creation (19-07-23) :- Created backend APIs (Channel, Database - CRUD).  


2. Backend-Frontend Integration (20-07-23) :- Connected the backend to the frontend, made Axios calls to fetch data, and displayed it using Apache charts.  
A screenshot of a computer

Description automatically generated

3. Data Retrieval from Hardware (20-07-23) :- Implemented data retrieval from hardware using the FS module.  


4. Binary Data Handling (21-07-23) :- Processed binary data from hardware, created APIs for each channel, and send the response in a integer data ranges.  
A screenshot of a computer

Description automatically generated

Issue Resolved :-   
  
1. Lag Reduction Logic (01-08-23) :- Implemented logic to reduce lag during initial data plotting.

2. Cursor Value Issue (02-08-23) :- Resolved cursor lag.

3. Slider Functionality (03-08-23) :- Disabled Zoom functionality within the slider, ensuring a smoother user experience.

4. Data Reading Logic (03-08-23) :- Updated data reading logic to improve performance.

5. Real Data Integration (04-08-23) :- Processed real data from files, modified API logic, and tested APIs for multiple channels and data ranges.

UI Development :-

1. Sign-In and Sign-Up Pages (17-08-23) :- Developed sign-in and sign-up pages, set up routing, and explored theme color changes.  
A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

2. Page Development (18-08-23) :- Created 404 page, modified loading icon, implemented dashboard header, vertical navbar, and forgot password page.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

Graph Functionality :-

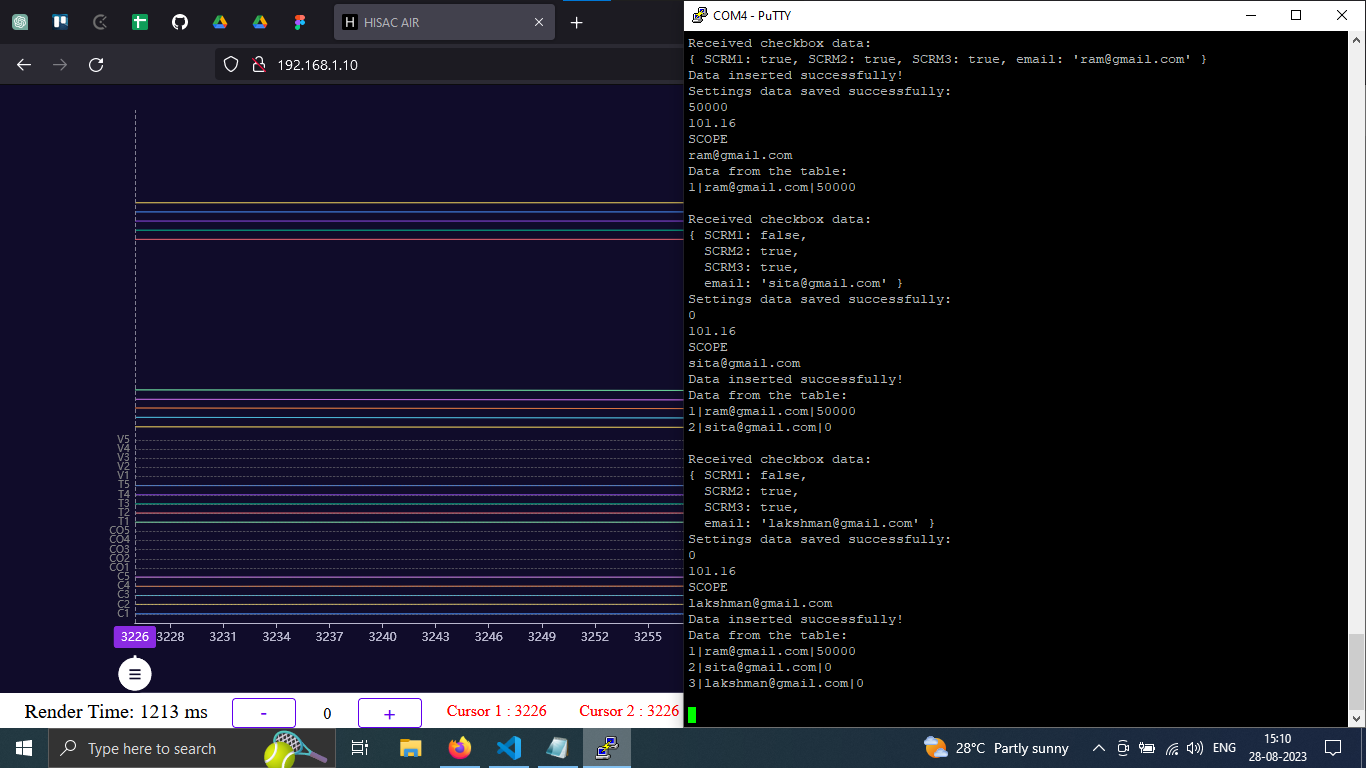
1. Channel Focus (21-08-23) :- Implemented the ability to focus on specific channels and blur others.  
A screenshot of a computer

Description automatically generated

2. Cursor Swapping Functionality (22-08-23) :- Implemented cursor swapping functionality.

3. Configuration Data Storage (23-08-23) :- Implemented a POST API to store configuration settings data in shared memory.  
A screen shot of a computer

Description automatically generated

4. Data Storage Testing (24-08-23) :- Created a table in SQLite3 for data storage, tested on hardware, and verified data storage functionality.  


Node.js Compatibility :-

1. Node Version Adjustments (25-08-23) :- Adjusted Node.js versions to ensure compatibility with hardware, resolved network errors, and ensured smooth server operation.

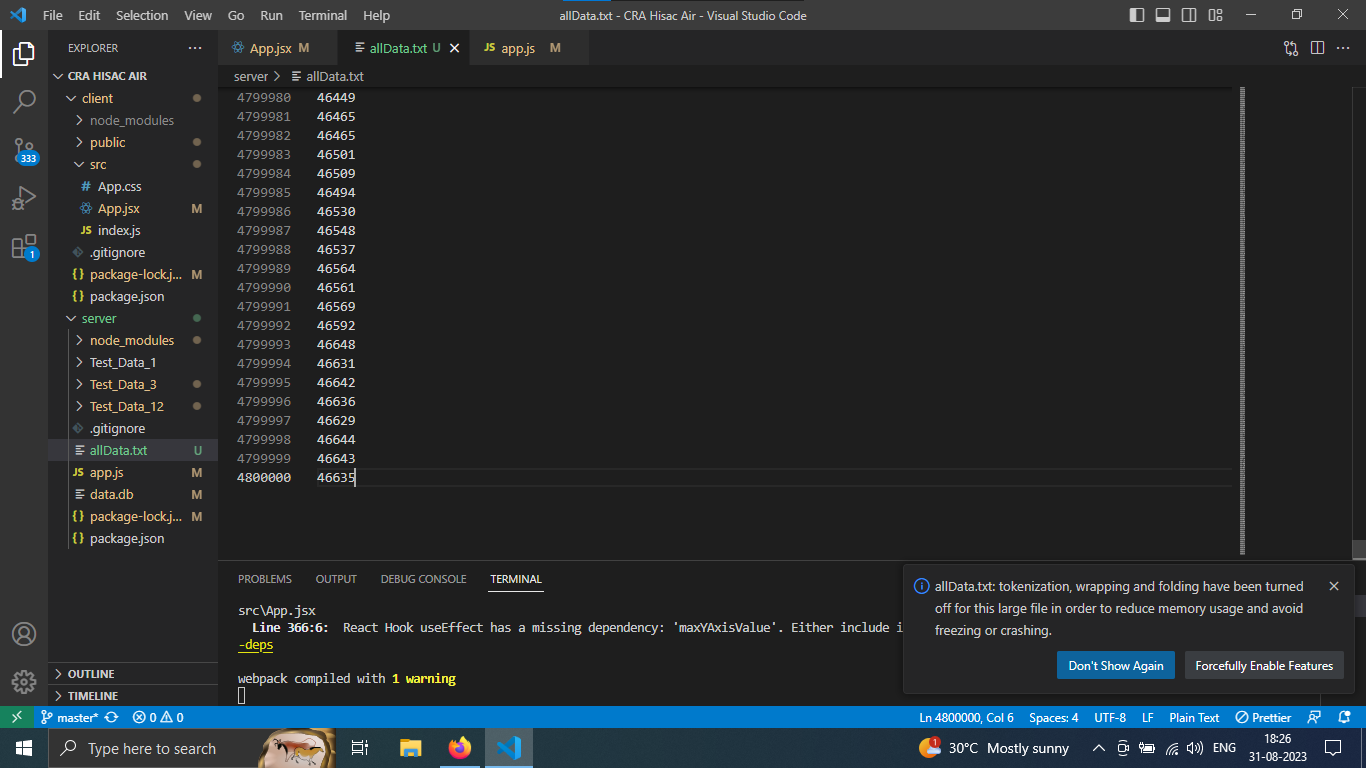
API Development :-

1. Data Fetching and Updating APIs (29-08-23) :- Created GET and PATCH request APIs for fetching and updating data.

2. Data Aggregation (31-08-23) :- Developed logic to read data from multiple files across 42 channels, aggregate it into a single file for efficient storage.

Binary Data Handling :-

1. Binary Data Conversion (01-09-23) :- Converted integers to binary, stored them in a single file, created an API, and tested response times.



Data Export and Import :-

1. File Export (05-09-23) :- Enabled file export from hardware memory to the user's computer.  
A screenshot of a computer screen

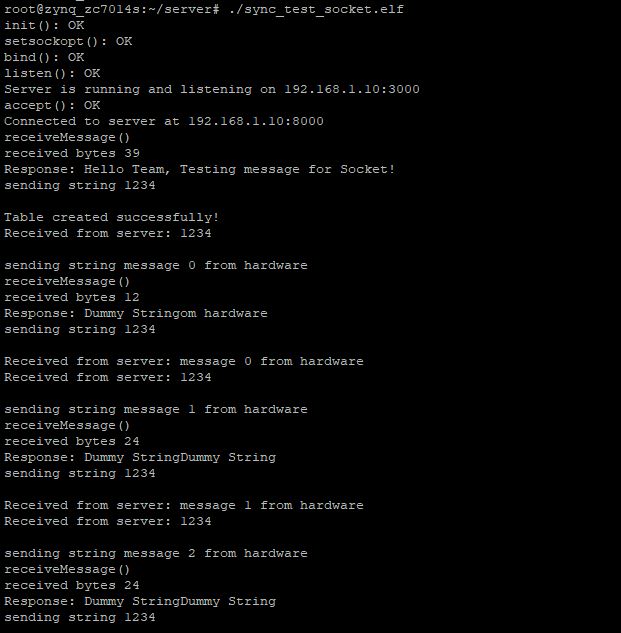
Description automatically generated

2. File Import (06-09-23) :- Implemented file import from the user's laptop and data storage in hardware memory for subsequent graph plotting.

A screenshot of a computer

Description automatically generated

Socket Integration :-

1. Socket Communication (07-09-23) :- Established socket connections, sent/received messages from a hardware port socket, and tested for handling multiple messages.  


Conclusion :-

This project has successfully addressed various challenges related to data visualization, API optimization, and hardware compatibility. The completed tasks have contributed to a robust and efficient data visualization app capable of handling large datasets and delivering a smooth user experience. Further development and testing are ongoing to refine the application and ensure its reliability in various scenarios.