HydroSense

**Table of contents**

[Introduction](#_topic_Introduction) [3](#_topic_Introduction)

[Software Overview](#_topic_SoftwareOverview) [3](#_topic_SoftwareOverview)

[Software Architecture](#_topic_SoftwareArchitecture) [3](#_topic_SoftwareArchitecture)

[Key Terminology](#_topic_KeyTerminology) [5](#_topic_KeyTerminology)

[Getting Started](#_topic_GettingStarted) [7](#_topic_GettingStarted)

[System Requirements](#_topic_SystemRequirements) [8](#_topic_SystemRequirements)

[Installation Guide](#_topic_InstallationGuide) [8](#_topic_InstallationGuide)

[First Run](#_topic_FirstRun) [9](#_topic_FirstRun)

[Features and Functionalities](#_topic_FeaturesandFunctionalities) [9](#_topic_FeaturesandFunctionalities)

[Core Features](#_topic_CoreFeatures) [10](#_topic_CoreFeatures)

[User Interface (UI) Guide](#_topic_UserInterfaceUIGuide) [12](#_topic_UserInterfaceUIGuide)

[Dashboard Overview](#_topic_DashboardOverview) [12](#_topic_DashboardOverview)

[Charts Overview](#_topic_ChartsOverview) [13](#_topic_ChartsOverview)

[Hardware Overview](#_topic_HardwareOverview) [15](#_topic_HardwareOverview)

[Distribution Overview](#_topic_DistributionOverview) [16](#_topic_DistributionOverview)

[AI Overview](#_topic_AIOverview) [17](#_topic_AIOverview)

[Cloud Overview](#_topic_CloudOverview1) [18](#_topic_CloudOverview1)

[Reports Overview](#_topic_ReportsOverview) [20](#_topic_ReportsOverview)

[Profile Overview](#_topic_ProfileOverview) [21](#_topic_ProfileOverview)

[Settings Overview](#_topic_SettingsOverview) [22](#_topic_SettingsOverview)

[Feedback Overview](#_topic_FeedbackOverview) [24](#_topic_FeedbackOverview)

[API Documentation](#_topic_APIDocumentation) [25](#_topic_APIDocumentation)

[API Overview](#_topic_APIOverview) [25](#_topic_APIOverview)

[Endpoint Descriptions](#_topic_EndpointDescriptions) [25](#_topic_EndpointDescriptions)

[Troubleshooting](#_topic_Troubleshooting) [26](#_topic_Troubleshooting)

[Glossary](#_topic_Glossary) [28](#_topic_Glossary)

[Best Practices](#_topic_BestPractices) [30](#_topic_BestPractices)

[Appendix](#_topic_Appendix) [32](#_topic_Appendix)

[Technical Specifications](#_topic_TechnicalSpecifications) [32](#_topic_TechnicalSpecifications)

[SDKs and Libraries](#_topic_SDKsandLibraries) [34](#_topic_SDKsandLibraries)

**Introduction**

**Software Overview**

Hydrosense is a real-time water quality monitoring system powered by IoT and AI technologies. The software module plays a central role in acquiring data from sensors, processing it using machine learning models, and delivering insights through a user-friendly web interface. Designed with modularity, scalability, and usability in mind, the software seamlessly bridges the hardware components and the end-users via a robust backend, intuitive frontend, and well-documented API services.

The software enables:

* Real-time monitoring and anomaly detection of water quality parameters such as turbidity, TDS, and temperature.
* Data visualization and interactive dashboard for both administrators and end-users.
* Integration with cloud storage (Firebase) for persistent and secure data logging.
* Secure user authentication with support for Google/GitHub sign-ins.
* AI-driven chatbot support using Hugging Face APIs for interpreting data trends and providing recommendations.
* Offline testing mode for simulation-based validation without the need for connected hardware.

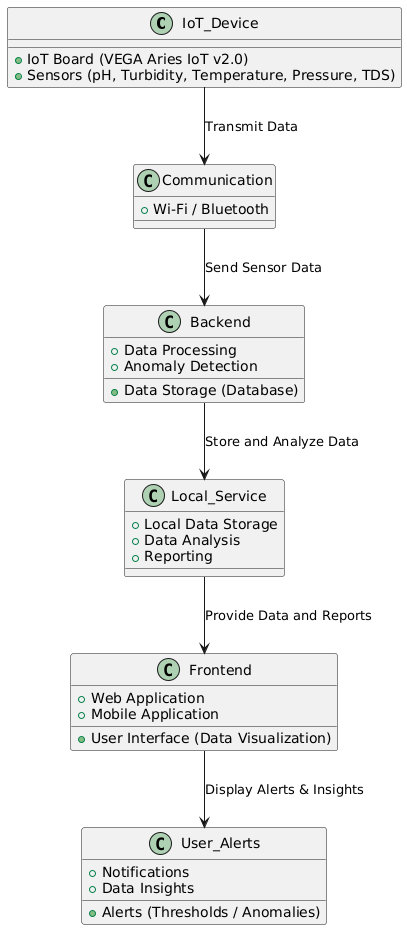
The software is structured to be easily extendable, allowing developers and researchers to integrate new sensors, improve AI models, or adapt it for different environmental monitoring applications.

**Software Architecture**

# Hydrosense Software Architecture

Welcome to Hydrosense – your real-time, AI-powered water quality assistant. This guide walks you through how the software works behind the scenes so you can use it confidently, whether you're a developer, tester, or just tech-curious.

## 1. High-Level Architecture



## 2. Components Breakdown

### A. ****Hardware (Vega Aries Board)****

* Reads sensor data: pH, turbidity, TDS, temperature.
* Hosts a mini web server (/, /buzzer) for real-time access.
* Sends JSON-formatted sensor data every 5 seconds.

### B. ****React.js Frontend****

This is what you see in your browser or mobile:

* Built with React + Redux.
* Pulls data from the Vega board or simulation server.
* Pages include:
  + Dashboard (Live graphs, metrics)
  + Reports (PDF, CSV, XLS)
  + AI Chatbot (Talk to your data!)
  + Feedback & Help
  + Authentication & Profile management

### C. ****AI Engine (FastAPI + HuggingFace API)****

* Runs in the backend to power the /ask endpoint.
* Takes in sensor data + questions → returns smart answers.
* Communicates securely with HuggingFace Mistral 7B model.

### D. ****Firebase (Auth + Firestore DB)****

* Auth:
  + Email/password
  + Google or GitHub
  + hCaptcha + Email verification
* Firestore DB:
  + Stores user-specific sensor snapshots.
  + Fully sandboxed: no cross-user access.
  + Load, delete, and backup data anytime.

## 3. Data Flow (Live Mode)

1. Vega board sends sensor data every 5 seconds.
2. Web app fetches this data (via API call to http://<device-ip>/).
3. Data is parsed, stored in **localStorage**, and rendered in:
   * Charts
   * Stats Cards
   * AI Assistant
4. If values exceed safe thresholds:
   * Visual alerts are shown.
   * Optional buzzer activation on hardware.

## Data Flow (Test Mode)

For devs or non-hardware users:

1. Enable **Test Mode** in Settings.
2. Web app fetches simulated data from:

https://hydrosense.pythonanywhere.com/

1. Same UI experience, powered by random safe/unsafe values.

## 5. Key API Endpoints

|  |  |
| --- | --- |
| **Endpoint** | **Purpose** |
| / | Get latest sensor data |
| /buzzer | Activate buzzer on Vega board |
| /ask | Ask questions about sensor data |

## 6. State Management (Redux)

The app uses Redux for:

* Global state sharing (sensor data, user auth, charts)
* Smooth real-time updates
* Consistent UI no matter which page you're on

## 7. File & Folder Snapshot

Software/

├── src/

│ ├── components/ # Reusable UI blocks

│ ├── pages/ # Dashboard, Reports, Settings, etc.

│ ├── redux/ # Actions, reducers, store setup

│ └── App.js # App entrypoint and router

├── public/ # Static files

├── assets/ # Images, styles

├── .env.local # Firebase & API config

## 8. Security + Privacy

* Authentication protected with hCaptcha + email verification.
* Data access is scoped to user ID.
* LocalStorage data never leaves unless you hit “Backup to Cloud.”

**Key Terminology**

# HydroSense:

The full system you're using. It’s a smart platform that monitors water quality using sensors + AI, and shows everything in a clean dashboard.

### ****Vega Aries IoT V2.0:****

The brains of the hardware setup. It’s a microcontroller that reads sensor values and shares them with the web app over WiFi. Think of it like a mini computer that talks to the cloud.

### ****Sensor:****

A device that measures stuff in the water:

* + **pH Sensor** – Measures how acidic or basic the water is.
  + **Turbidity Sensor** – Checks how clear or cloudy the water is.
  + **TDS Sensor** – Measures the Total Dissolved Solids (salts, minerals, etc.).
  + **Temperature Sensor** – Obvious one. Just tracks water temperature.

### ****Dashboard:****

The first screen you see in the app. Shows sensor readings, trends, and water quality summary in real time. It updates every few seconds automatically.

### ****Authentication:****

Logging into the app using Email, Google, or GitHub. It makes sure only you see your data. Includes extra security like email verification and hCaptcha to keep bots away.

### ****Cloud Backup:****

Store your data safely in Firebase. If you switch devices or clear cookies, you can still get your old readings back by loading from the cloud.

### ****AI Assistant (Talk with Data):****

A built-in chatbot that understands your sensor data. You can ask stuff like: “How's my water today?” or “Is the TDS too high?” Powered by HuggingFace Mistral 7B model and FastAPI.

### ****Threshold Alert:****

Each sensor has a safe range. If values go out of bounds, the app flashes alerts. If hardware is connected, the buzzer also goes off to let you know.

### ****Test Mode:****

Don’t have the real sensors? No worries. Enable test mode and get simulated data streamed to the app. Perfect for demos, development, and debugging.

### ****AI Prediction:****

Uses trained ML models (like SVM) to tell whether your water is **Good**, **Average**, **Bad**, or **Worst** based on the latest sensor data.

### ****Redux:****

A behind-the-scenes tool that keeps your app state consistent. For example, if you’re on one page and switch to another, your data stays synced without bugs.

**Getting Started**

Welcome to Hydrosense! Whether you're checking water quality from your laptop or phone, or testing the system without any hardware, getting up and running is simple. Here’s how to dive in:

## Option 1: Use the Web App

You can access Hydrosense **instantly** from your browser—no install needed.

🔗 **Visit Now:** <https://hydrosense-app.vercel.app/>

### What You Can Do:

* View real-time sensor data (if your hardware is connected)
* Test the system with simulated data (Test Mode)
* Generate reports (PDF/Excel)
* Chat with the AI Assistant
* Backup and load data from Firebase
* Manage your profile and settings

## Option 2: Windows EXE App

* Prefer using a native desktop app?
* **Hydrosense for Windows** (Download link to be provided by you)
* Just double-click the .exe to launch Hydrosense like any other Windows app. No browser needed!

## Option 3: Android APK

* Want to run Hydrosense on your phone?
* **Hydrosense APK** (Download link to be provided by you)
* Install the APK manually by enabling **Install from Unknown Sources** in your Android settings.

## No Hardware? No Problem

You can still test everything using **Test Mode**:

1. Open the app.
2. Go to **Settings**.
3. Click **Enable Test Mode**.

You'll get fake—but realistic—sensor data streamed from our Python backend. Great for exploring features and demos.

## Tip: First-Time Login

To get the best experience, sign in with:

* Google
* GitHub
* Or Email & Password (email verification + hCaptcha for safety)

Your data is stored securely and synced with Firebase.

**System Requirements**

## Requirements

* Modern browser (Chrome, Firefox, Edge)
* Node.js (for dev mode)
* For EXE: Windows 10 or 11
* For APK: Android 8.0 or above

**Installation Guide**

# Installation Guide

You can use Hydrosense across multiple platforms: **Windows**, **Android**, and **Web**. Choose what works best for you.

## Web-Based Dashboard

**Recommended for quick access — no installation needed.**

**URL**: <https://hydrosense-app.vercel.app/>

### Features:

* Full Dashboard Access (Charts, Reports, AI Assistant, etc.)
* Cloud Backup & Restore
* Test Mode available
* Works on any modern browser (Chrome, Firefox, Edge)

## Windows Application

### Steps to Install:

1. Visit the **Hydrosense website** and download the latest **Windows EXE** file.
2. Double-click the downloaded .exe file to start the installation.
3. Follow the installation wizard to complete setup.

### Important Note:

* When installing, Windows may show a **“Unknown Publisher” warning**. This happens because the installer isn't digitally signed by Microsoft.
* **Don't worry — the software is safe.** Click **“More Info” → “Run Anyway”** to proceed.

## Android Application (APK)

### Steps to Install:

1. Go to the **Hydrosense website** and download the Android **APK** file.
2. On your Android phone, open the APK file.
3. If prompted, enable **Install from Unknown Sources** in your settings.
4. Proceed with installation.

### Once Installed:

* Open Hydrosense from your app drawer.
* Log in with Google, GitHub, or Email.
* Start monitoring or use **Test Mode** if no hardware is connected.

## For Developers: Local Setup

If you’re a developer and want to run the frontend or backend locally :

### Frontend (React.js)

cd Software

npm install

npm run start

### Backend (Python + FastAPI)

cd ai/bot

uvicorn app:app --reload

Make sure Python 3.9+ and Node.js are installed.

**Features and Functionalities**

Hydrosense is a smart, portable water contamination detection system designed for **real-time environmental monitoring**. It combines **custom hardware**, **AI-driven analytics**, and a slick **React-based interface** to deliver clear, actionable insights about water quality.

### Core Functionalities

#### Real-Time Sensor Monitoring

* Continuously reads **pH, turbidity, TDS, and temperature** values.
* Displays sensor readings with live updates on the **Dashboard**.
* Pulls additional metadata like **battery level**, **device name**, **test count**, and **last test timestamp**.

#### Data Visualization & Trends

* **Dashboard View**: Shows the last 10 test results in a compact chart.
* **Full Chart Page**:
  + Detailed graphs of the last 1000 readings.
  + Interactive features: **Zoom**, **Highlight Danger Limits**, **Export as PDF/XLSX**, and **Restore View**.
  + Helps identify **anomalies** or **long-term trends**.

#### AI Assistant - “Talk with Data”

* Built-in **AI chatbot** trained on historical sensor data.
* Ask natural language questions like:
  + "When was the last time pH was too low?"
  + "How does turbidity correlate with temperature?"
* Uses NLP + your collected data for **smart querying**.

#### Hardware Diagnostic Tools

* **Vega Aries Status Page**:
  + Displays data from the board’s **internal sensors**.
  + Live testing of **buzzer** and **LED** via button triggers.
* **Device Health**: Tracks uptime, communication status, and sensor error flags.

#### Web API (Hardware-Software Communication)

* **Wi-Fi Access Point Mode** with endpoints:
  + / → Returns all sensor data in JSON.
  + /buzzer → Triggers buzzer for 2 seconds.
* Enables smooth **data sync** between Vega board and frontend.

#### Sensor Simulation (Software Side)

* Simulated sensor data for **testing UI/UX** and **AI models** without real hardware.
* Useful for debugging or working remotely.

#### Firebase Integration

* **Authentication** with Firebase Auth.
* Cloud backup of all data to **Firestore Database**.
* Secure login and easy cloud access to historical reports.

#### Reports & Downloads

* Generate downloadable **PDF reports** using jsPDF and jsPDF-AutoTable.
* Export full charts and sensor logs to **Excel** (.xlsx) using xlsx library.

**Core Features**

### Real-Time Water Quality Monitoring

* Live tracking of **pH**, **TDS**, **Turbidity**, and **Temperature**.
* Dashboard auto-refreshes with latest values from hardware or simulations.

### Advanced Data Visualization

* Mini-dashboard: Last 10 test values chart.
* Full chart view: 1000+ historical entries with **zoom**, **limit markers**, and **download** options.

### AI-Driven Insights

* Built-in AI bot to answer natural-language questions based on collected sensor data.
* Understand trends, spot anomalies, and interpret environmental shifts.

### Vega Board Integration (IoT)

* Connects with **Vega Aries IoT V2.0** via Wi-Fi.
* Exposes REST endpoints for data fetch and hardware controls (e.g., /buzzer).

### Simulated Data Mode

* Toggle to simulated sensor input when hardware is not connected.
* Ideal for UI testing, development, and training.

### Hardware Diagnostics & Control

* Check hardware health, device stats, and internal readings.
* Trigger **buzzer** and **LED** tests remotely from the interface.

### Secure Cloud Sync with Firebase

* User authentication and secure access.
* Auto backup of test results to the cloud for long-term storage.

### Auto Report Generation

* Generate and download test reports as **PDFs** or **Excel files**.
* Automatically includes charts, timestamp, and device info.

**User Interface (UI) Guide**

**Dashboard Overview**

## 

## ****Overview****

HydroSense helps you monitor water quality and distribution in real time using sensor data, AI-based predictions, and intuitive controls. This dashboard gives you live updates on contamination levels, system health, and sensor status.

## ****Dashboard Layout Breakdown****

### 1. ****Header Section****

* **Start Monitoring:** Initiates real-time data collection from connected sensors.
* **Stop Monitoring:** Pauses data collection.
* **Load Data:** Load the locally stored data into the application.
* **Save Data:** Stores current sensor readings locally.
* **Status Indicator:** Displays Stopped or Active based on monitoring state.
* **Timer:** Shows elapsed time (in seconds) since monitoring started.
* **Endpoint Connected:** To hardware or test endpoint

### 2. ****Sensor Readings Panel****

Each sensor tile shows:

* **Value**: Real-time measurement (e.g., Turbidity, TDS, Temp).
* **Limit**: Threshold beyond which water is considered unsafe.
* If a sensor is offline, you’ll see a placeholder or warning badge.

Tip: If all values show 0, check your sensor connections or wait for data to sync.

### 3. ****Sliders/Indicators****

These show a visual representation of each sensor’s reading using sliders or progress bars. Color-coded to indicate safety:

* Green - Safe
* Yellow - Borderline
* Red - Unsafe

### 4. ****Water Test Details Block****

Shows:

* **Total Test Conducted: Number of tests performed.**
* **Last Tested: The last tested timestamp.**
* **Test Location: Geographic location of the device.**
* **Water Status**: Good / Average / Bad

### 5. ****Alerts Section****

If the system detects poor water quality or hardware issues:

* Alerts appear here in red.
* Includes timestamps and short descriptions.

If you see “No alerts at the moment,” it means everything is good!

### 6. ****Announcements Section****

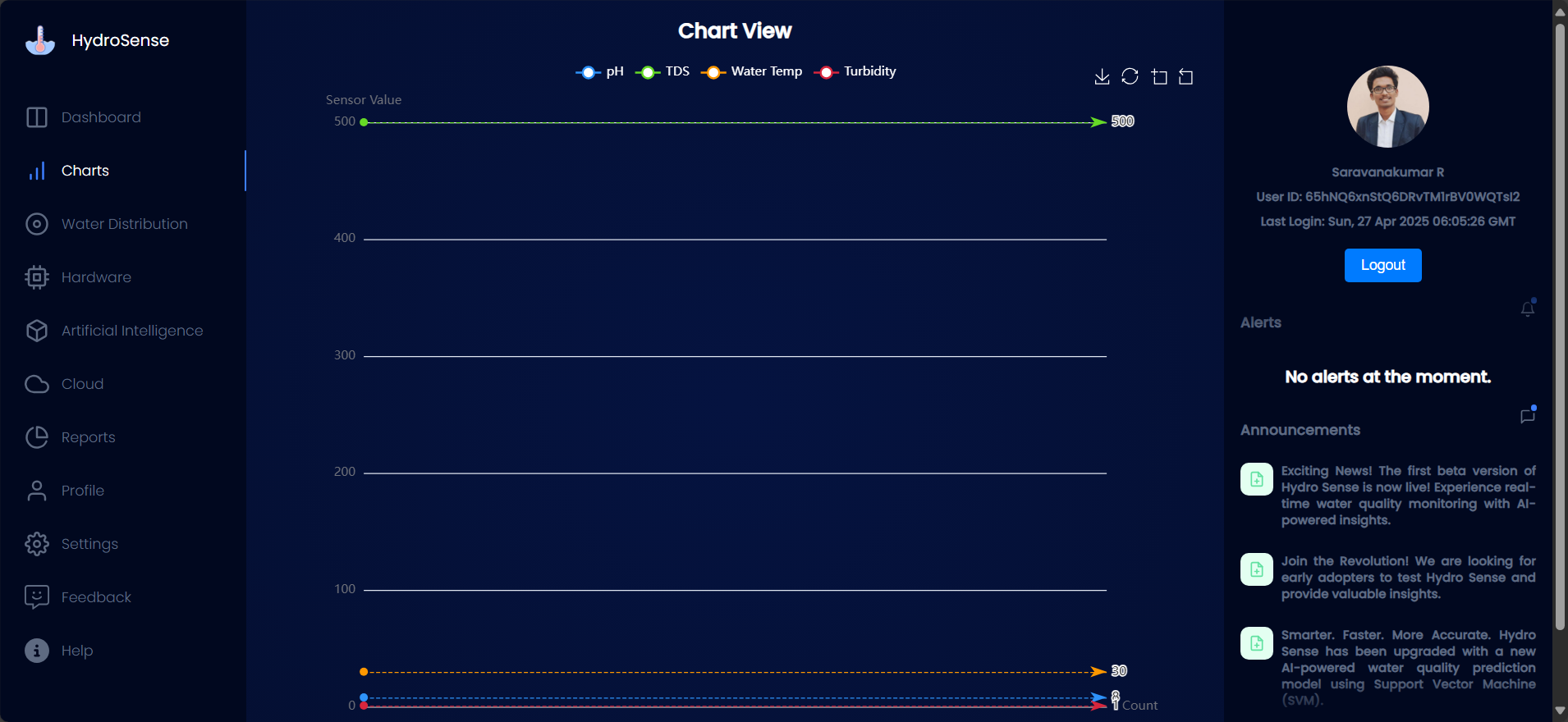
Stay updated with the latest app changes or maintenance schedules:

* Each card contains a title, date, and details.
* These are pushed by the development/admin team.

## ****Known Limitations****

* Real-time data works only when connected to Vega Board WiFi.
* Some sensors may take 5–10 minutes to warm up.
* Water quality result is based only on available sensor inputs. Heavy metal detection not yet supported.

**Charts Overview**



The **Charts View** is your visual hub to monitor real-time water quality sensor readings. It's designed to be clean, intuitive, and data-rich.

### ****Key Elements****

#### Chart Title

* **"Chart View"** — the main title indicating you're viewing live or historical sensor data.

#### ****Legend****

Color-coded indicators for each parameter:

* **pH** – Blue
* **TDS (Total Dissolved Solids)** – Yellow
* **Water Temp** – Red
* **Turbidity** – White

These help you visually differentiate sensor values on the chart.

### ****Graph Area****

* **Y-axis (Vertical)**: Sensor Value scale (from 0 to 500 in this view).
* **X-axis (Horizontal)**: Time or count-based data points (context-dependent).
* Sensor values are plotted as lines with animated arrows showing direction and live value.

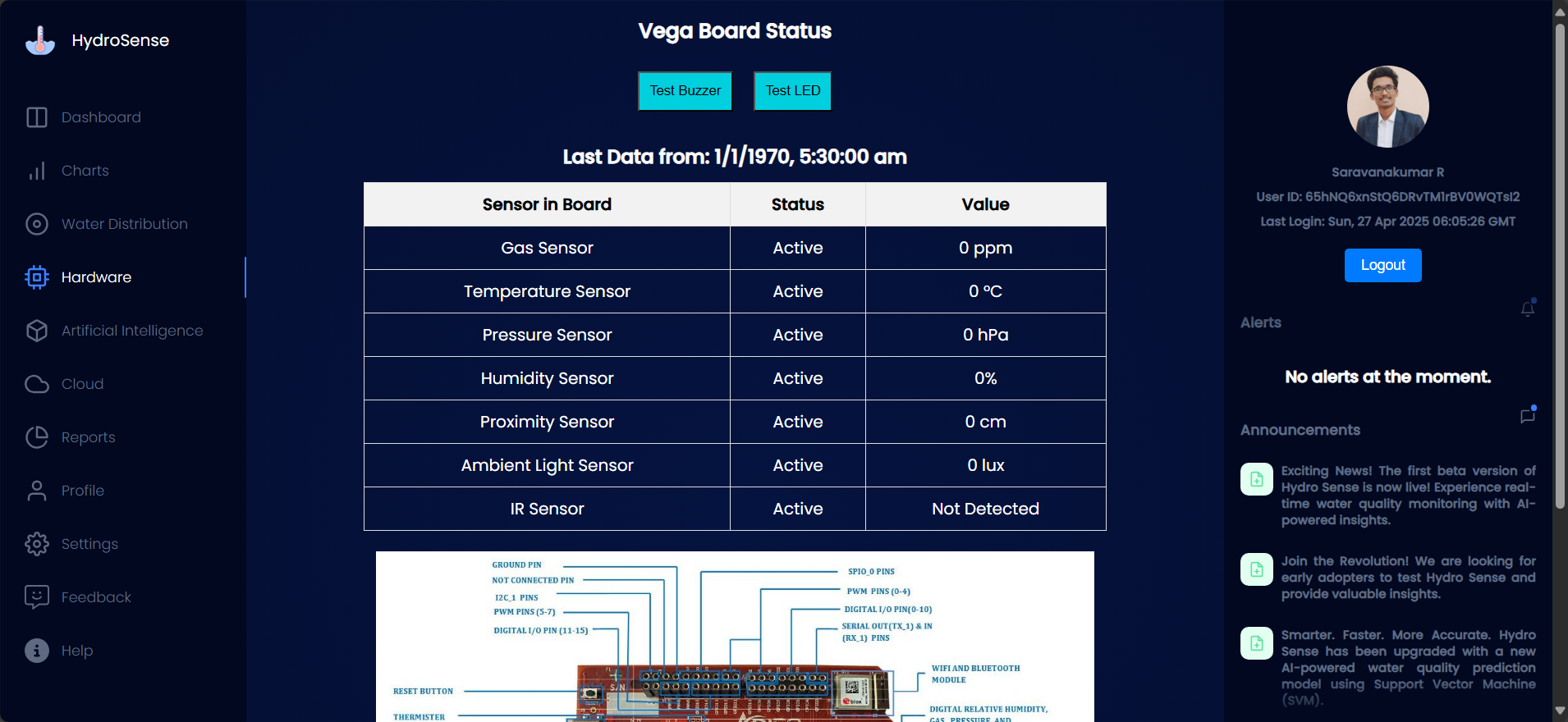
### **Chart Controls** (Top-right icons)

* **Refresh**: Pull the latest sensor data.
* **Download**: Export chart data (CSV/Excel format).
* **Zoom**: Focus on a specific range.
* **Copy**: Copy chart snapshot or data.

### Highlighted Limits:

* **TDS is 500** (highlighted as the **green** **line**),
* **pH is 8** (highlighted as the **blue** **line**),
* **Turbidity is 1** (highlighted as the **red** **line**),
* **Temperature is 30** (highlighted as the **orange** **line**),

**Hardware Overview**



Welcome to the HydroSense hardware interface! This dashboard gives you real-time control and monitoring of your village water quality and sensor network.

### 1. Hardware Status Table

This table shows the live status of all connected sensors and devices in the HydroSense system.

|  |  |
| --- | --- |
| **Field** | **Description** |
| **Sensor in Board** | The name of the sensor/device installed (e.g., Temperature, Turbidity, Buzzer). |
| **Status** | Shows if the sensor is currently connected and functional. If it says “Active,” everything’s good. |
| **Value** | The latest reading from that sensor. |

#### Note:

* If you see 0 or Not Detected, the sensor might not be active or is facing an issue.
* If the "Last Data From" shows 1/1/1970, it means no recent data has been received yet.

### 2. Control Panel

* **Test Buzzer:** Click to make the board buzzer beep. Use this to test alert systems.
* **Test LED:** Click to flash the LED. Useful to confirm the board is live and responding.

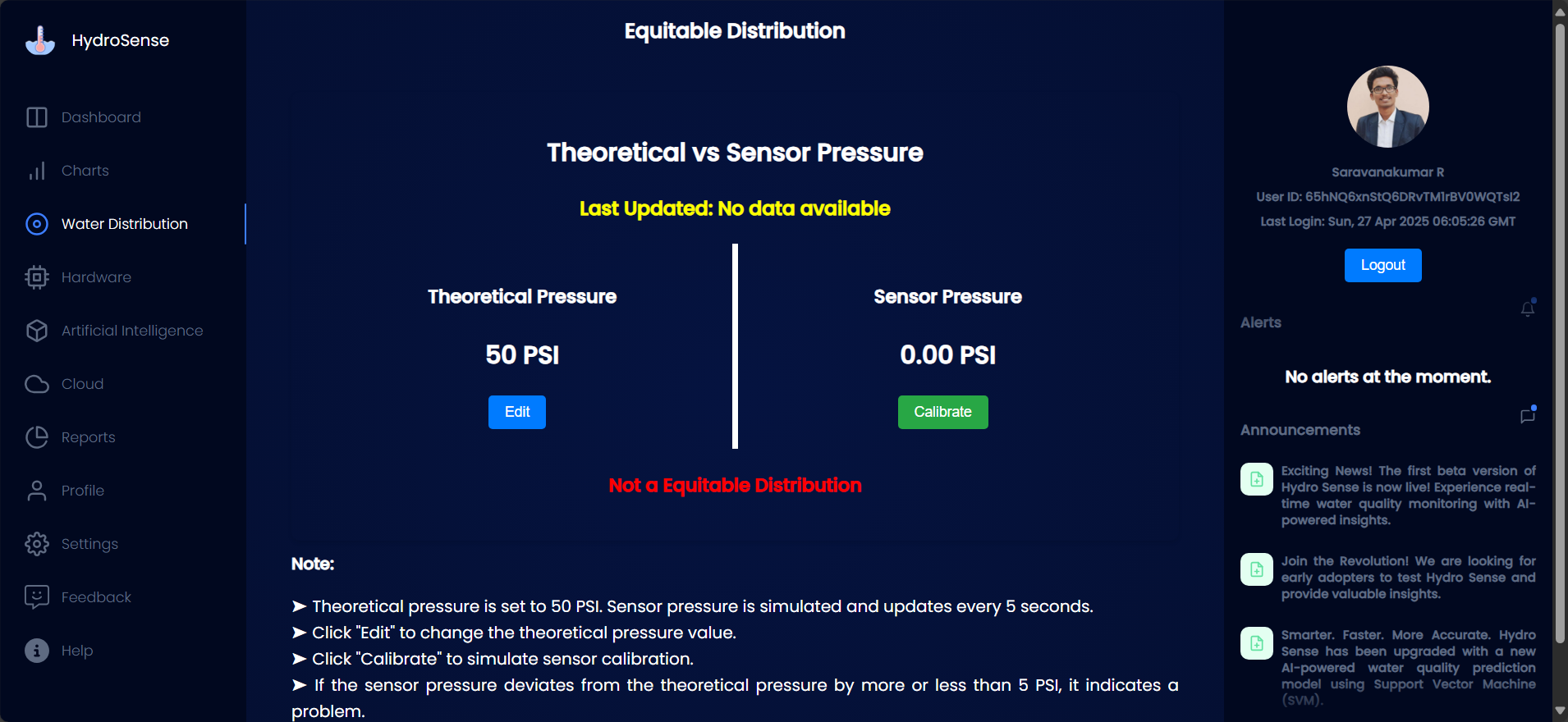
Use these features to verify board functionality — especially during installation or troubleshooting.

### 3. System Info & Notifications

#### Last Data Timestamp

Shows when the dashboard last received data from the board. If it shows No recent data available or an odd date like 1970, the board might be offline or just booting up.

**Distribution Overview**



This module helps monitor the balance between expected (theoretical) water pressure and actual pressure detected by sensors in a water distribution system. It’s designed to detect distribution issues, calibrate sensor readings, and maintain optimal flow conditions.

|  |  |
| --- | --- |
| **Component** | **Description** |
| **Theoretical Pressure** | Manually set value representing the expected pressure in PSI. Default: 50 PSI. |
| **Sensor Pressure** | Live or simulated value received from the pressure sensor. Updates every 5 seconds. |
| **Edit Button** | Allows you to modify the theoretical pressure value. |
| **Calibrate Button** | Simulates or triggers sensor calibration to refresh the sensor reading. |
| **Status Indicator** | Displays if the distribution is **Equitable** or **Not Equitable**, based on a ±5 PSI threshold. |
| **Alerts Panel** | Displays real-time alerts (if any discrepancies or critical issues are detected). |
| **Announcements Panel** | Shows important updates, feature releases, and AI model improvements. |



## How It Works

* **Start** by reviewing the default theoretical pressure (e.g., 50 PSI).
* Click **Edit** if you want to change the expected value based on real-time needs or testing conditions.
* Click **Calibrate** to simulate sensor pressure input. The value auto-updates every 5 seconds.
* If the sensor reading deviates by more than ±5 PSI, a warning is displayed:

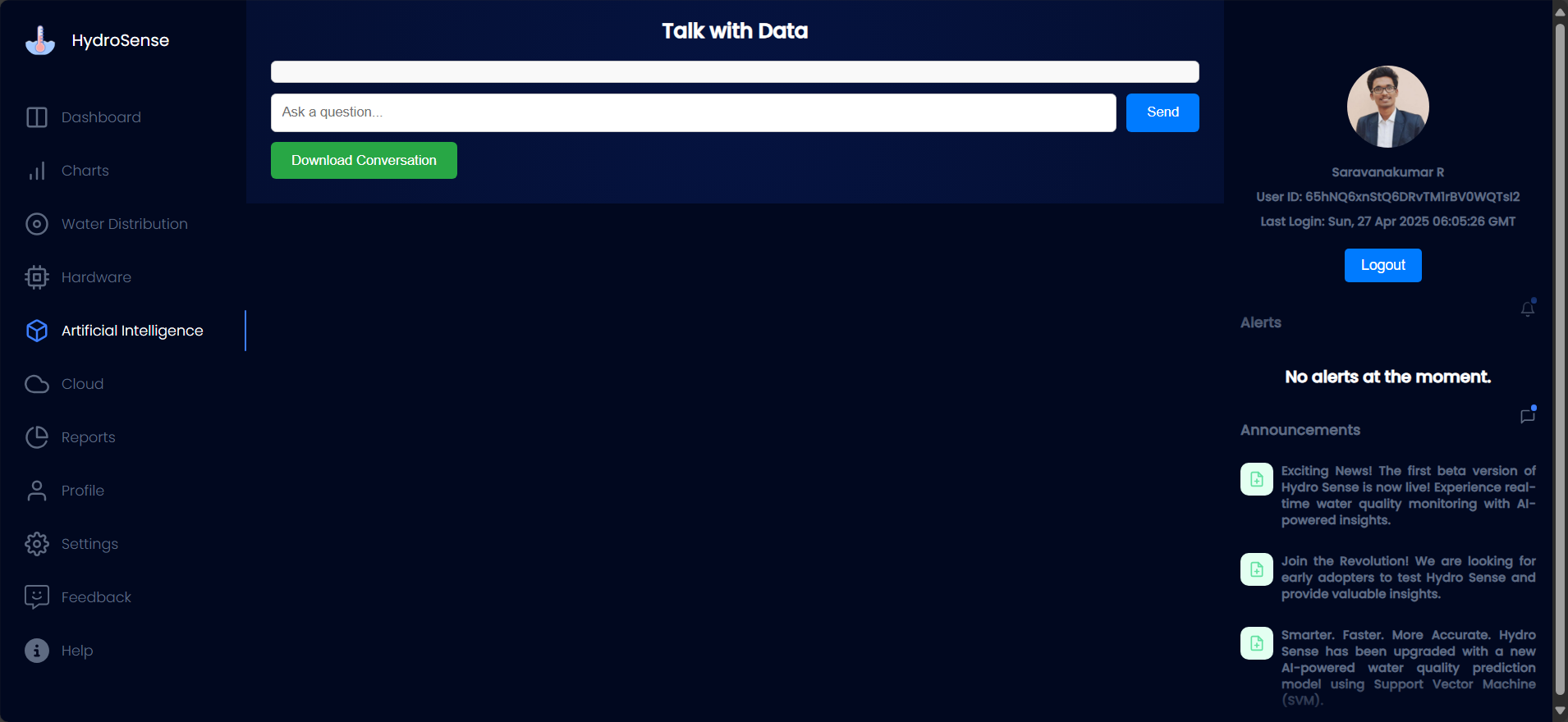
**"Not a Equitable Distribution"** (in red).

* Use this discrepancy as a sign to check the distribution system for faults or recalibrate the sensors again.

## Notes

* The sensor pressure may show **0.00 PSI** initially if it hasn’t received data or calibration hasn’t occurred yet.
* The data is simulated for demo/testing purposes in the beta version.
* This module does **not** control actual valves or pressure regulators — it’s for monitoring & analysis only.

**AI Overview**



The **Artificial Intelligence** section of HydroSense allows users to interact directly with the water quality dataset using natural language. It's your personal water scientist—just ask a question and get immediate insights. This feature is powered by AI and trained on real-time sensor data and historic trends.

## Key Features

### Talk with Data Panel

* **Text Input**: Ask any water-quality-related question (e.g., "What was the pH average last week?" or "When did turbidity last spike?").
* **Download Conversation**: Use this green button to export your question-answer session as a **PDF** for documentation or report submission.

### AI Response Engine

* Uses natural language processing to interpret your queries.
* Runs on live & stored HydroSense data.
* Supports **trend extraction**, **anomaly detection**, and **regulation comparison**.

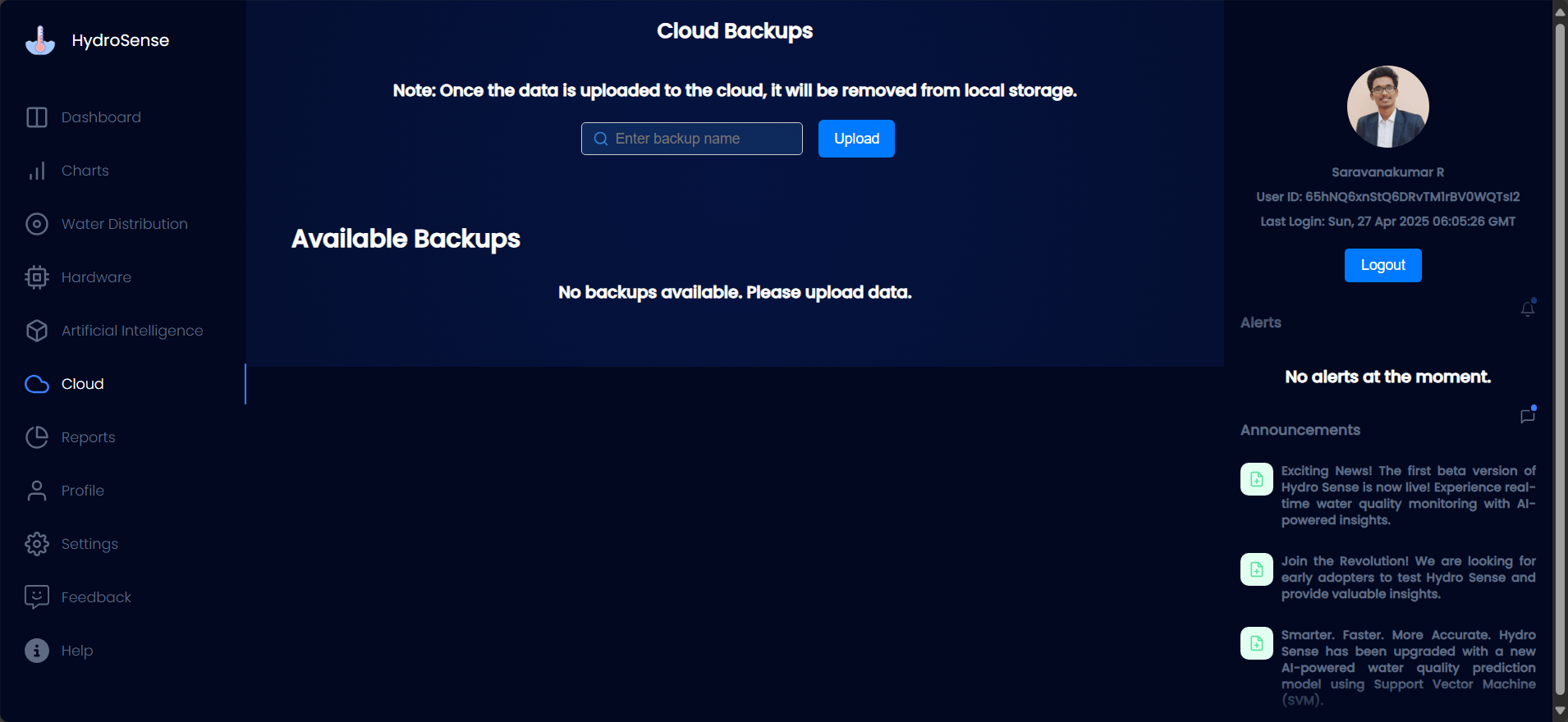
## How to Use This Module

* Type your question in plain English (e.g., "Show me the days with unsafe pH levels.")
* Press **Send**
* View the response directly below.
* Click **Download Conversation** if needed.

## Pro Tips

* Use terms like "average", "compare", "peak", "lowest", "safe", "unsafe", etc., for more accurate responses.
* Try asking follow-up questions to dig deeper into previous results.
* Works best when recent sensor data is available and the backend is running.

**Cloud Overview**



The **Cloud Backups** module allows you to safely store water quality data in the cloud and retrieve it whenever needed. This ensures that your local storage stays clean and optimized, while your historical data remains secure and easily accessible.

## How It Works

* **Upload to Cloud**: Once you upload a backup to the cloud, the system automatically **removes it from local storage** to save space.
* **Load From Cloud**: You can restore any uploaded backup by clicking **Load This Data**.
* **Delete Backup**: Remove unused or outdated backups with the **Delete** button.

## Features

### Create a New Backup

* Enter a unique name in the **"Enter backup name"** field.
* Click **Upload** to back up the current dataset to the cloud.

### Available Backups

**A list of previously uploaded backups is displayed, showing:**

* + **Name** (e.g., Test 3, IOAD, Timer -10+)
  + **Uploaded Time** (Timestamp of when the backup was made)

**Each backup entry has:**

* + **Load This Data**: Restores that specific backup from the cloud.
  + **Delete**: Permanently removes the backup from cloud storage.

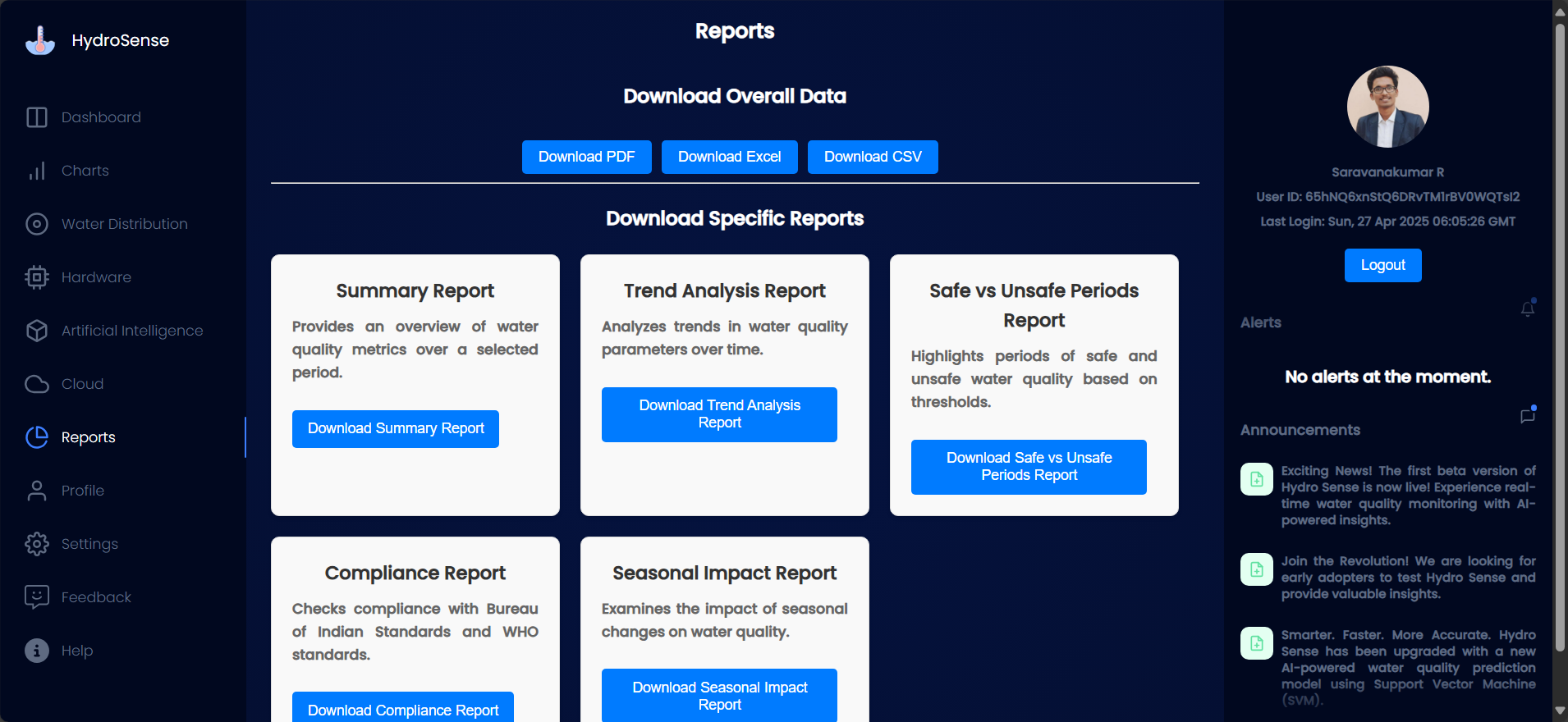
## Important Notes

* Once uploaded, **the data is deleted from your local storage**. Be sure you want to move it before uploading.
* Backup names should be **distinct and meaningful** to avoid confusion during restoration.
* Deleted backups **cannot be recovered**.

## Best Practices

* Name backups with the **project/date/purpose** (e.g., March\_Report\_2025, Sensor\_Tuning\_Test).
* Regularly **clean up old backups** to avoid clutter.
* Before deleting, always **double-check** that the backup is no longer needed.

**Reports Overview**



The **Reports** page provides access to downloadable data insights from HydroSense, helping users analyze water quality trends, compliance, and system health over time. You can export all data or specific reports based on your needs.

## Types of Reports Available

### ****Download Overall Data****

Use the top panel to export the entire dataset collected from HydroSense in your preferred format:

* **PDF**: For printable reports
* **Excel**: For advanced data filtering or pivoting
* **CSV**: For integration with other tools or analysis platforms

### ****Download Specific Reports****

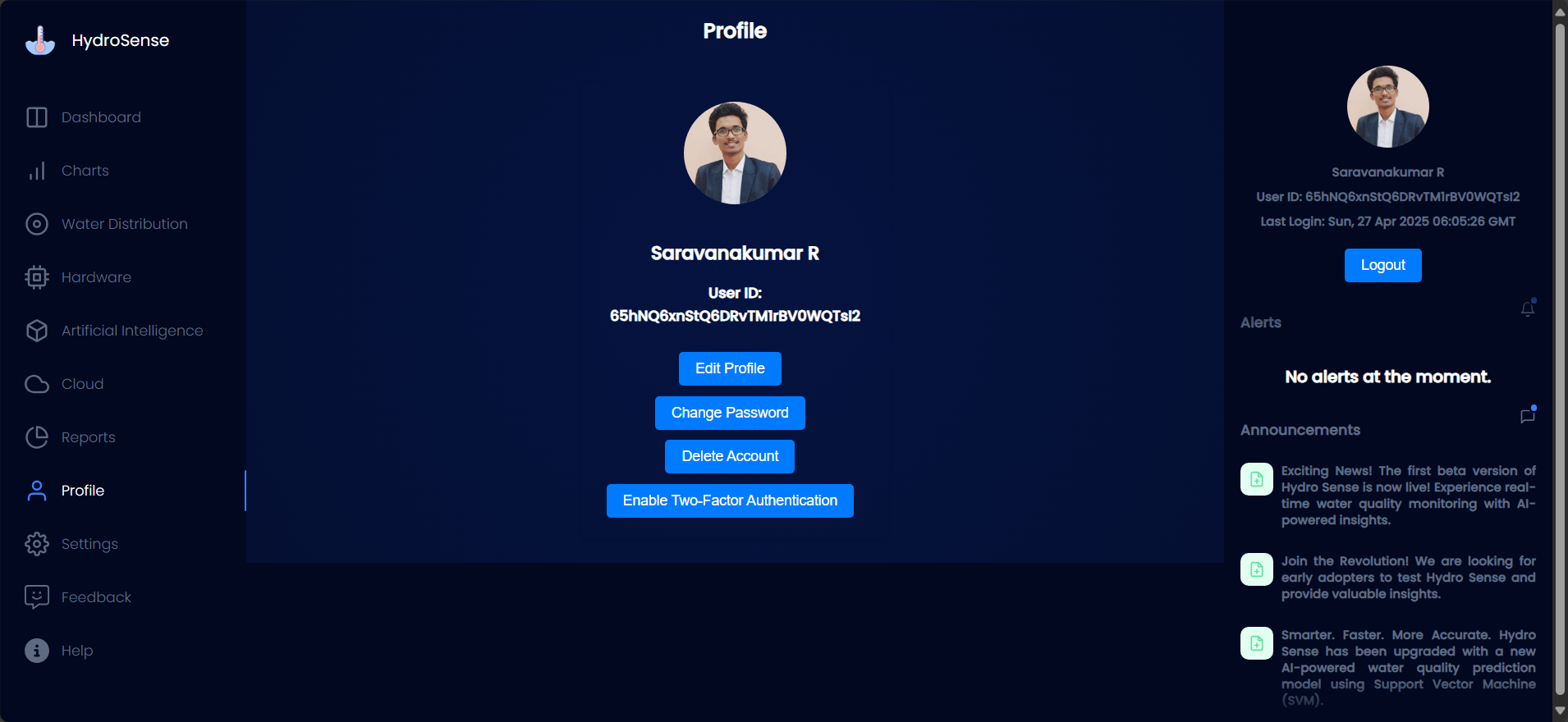
Choose from 5 targeted reports for focused insights:

|  |  |
| --- | --- |
| **Report** | **Description** |
| **Summary Report** | Provides an overview of water quality metrics (like pH, turbidity, TDS) for a selected time period. |
| **Trend Analysis Report** | Highlights long-term changes and anomalies in water quality parameters using time-series analysis. |
| **Safe vs Unsafe Periods Report** | Identifies periods where water quality was within or outside acceptable limits. Useful for alert tracing. |
| **Compliance Report** | Compares collected data against BIS and WHO standards to assess regulatory compliance. |
| **Seasonal Impact Report** | Analyzes how water quality varies across different seasons — especially helpful for agricultural or municipal planning. |



Each card has a **Download** button that will generate and download the corresponding report.

**Profile Overview**



The **Profile** section is your personal space on HydroSense. It allows you to manage your account, update credentials, and secure your data with additional layers of protection.

## Profile Information

**This section displays your:**

* **Full Name**: Saravanakumar R
* **User ID**: 65hNQ6xnStQ6DRvTM1rBVOWQTs12

Your User ID is unique and used internally to identify your account. Keep it safe and **do not share it** publicly.

## Profile Options

### Edit Profile

* Update your personal details such as name, email, or profile picture.
* **How to use:**
  + Click **Edit Profile**
  + Modify your details (Name and Photo can be changed)
  + Save the changes

### Change Password

* Secure your account by changing your password regularly.
* **How to use:**
  + Click **Change Password**
  + A password change email will be sent to your registered Email ID

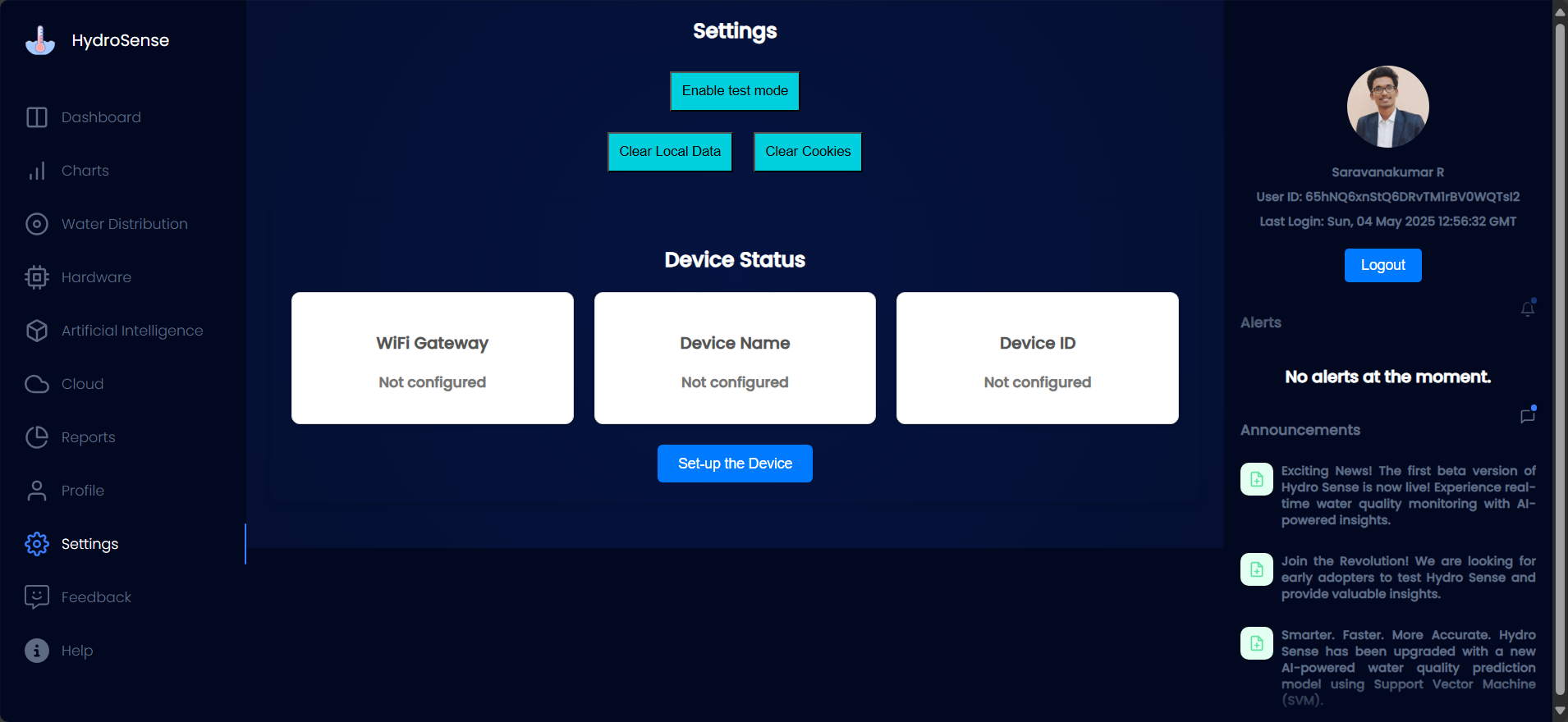
### Enable Two-Factor Authentication (2FA)

* Add an extra layer of security using 2FA. This requires both your password and a second verification method (like an OTP or authenticator app).
* **Why enable?**
  + It protects your account even if your password is compromised.
  + **Pro Tip:** Use **Google Authenticator** or **Authy** for the best experience.

### Delete Account

* Permanently removes your HydroSense account and all related data.
* **This action is irreversible.**
* Only use if you're absolutely sure you want to leave the platform.

**Settings Overview**



#### **1. Enable Test Mode**

* Test Mode allows you to simulate sensor data and test system functionalities without actual sensor connections. This is useful for troubleshooting or testing features before deployment.
* **How to use:**
  + Toggle the "Enable Test Mode" setting to "ON" to start using simulated data.
  + Use "Set-up Device" button to connect hardware.

#### **2. Clear Local Data**

* Clearing local data removes any cached or stored data on your device, ensuring a fresh start. This will not delete data from the cloud but will clear any temporary files saved on your device for better performance.
* **How to use:**
  + Click the “Clear Local Data” button.
  + Confirm the action when prompted.

#### **3. Clear Cookies**

* Cookies are small files stored on your device that help with session management and app preferences. Clearing cookies will log you out of the app and reset any user-specific settings.
* **How to use:**
  + Click the “Clear Cookies” button.
  + Confirm when prompted.

#### **4. Set-Up the Device**

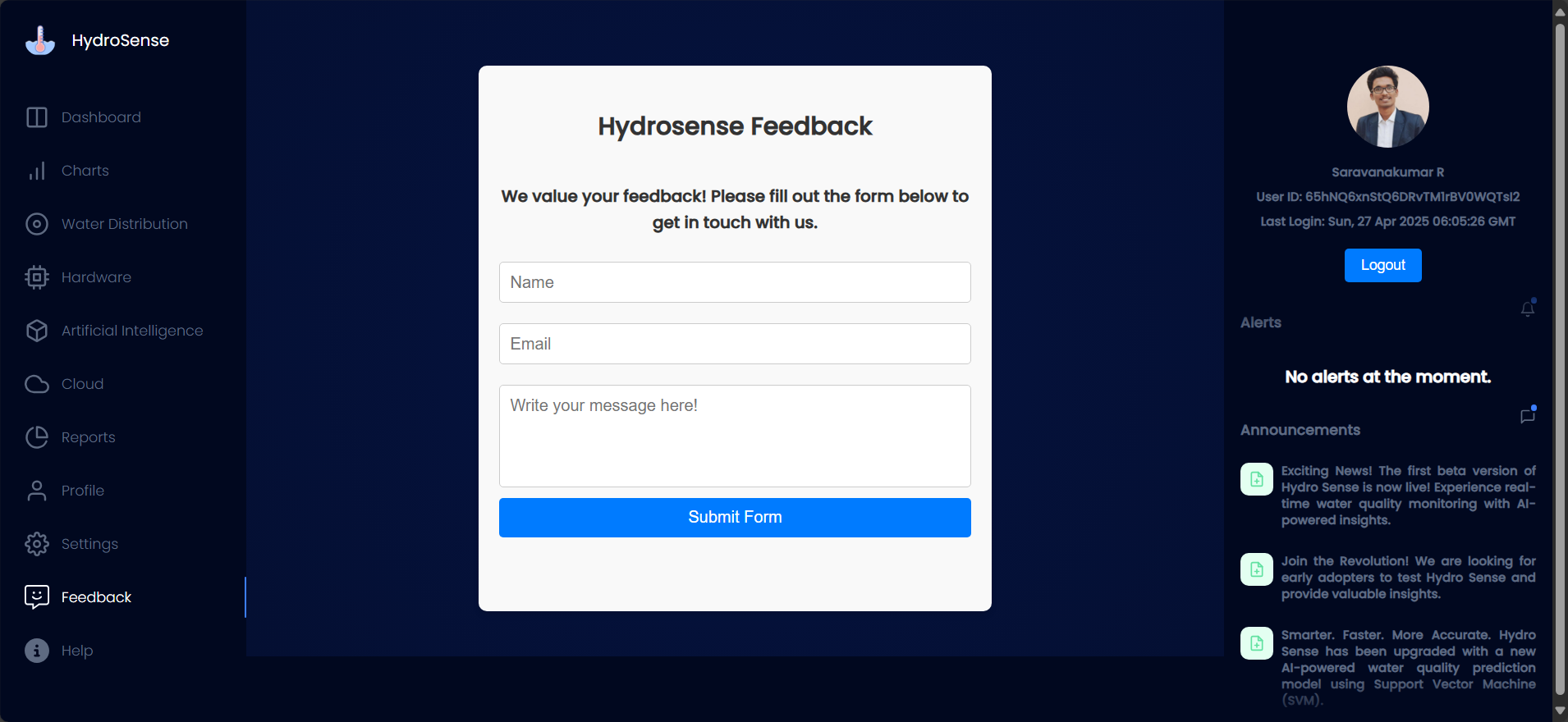
* If your device isn’t configured, you can set it up through this section. Ensure you follow the steps to properly configure the device and begin collecting water quality data.
* **How to set-up:**
  + Tap “Set-up the Device” and follow the on-screen instructions for device setup.
  + Make sure your Wi-Fi gateway is connected and your device name is entered.

#### **5. Device Status**

* This section provides a real-time overview of your HydroSense device's current status.
  + **WiFi Gateway:**
    - **Not configured**: Your device is not connected to Wi-Fi.
    - **Configured**: Your device is connected to a Wi-Fi network.
  + **Device Name**:
    - The name of your device (e.g., “HydroSense Station 1”).
    - **Not configured**: The device name has not been set yet.
  + **Device ID**:
    - A unique identifier for your device.
    - **Not configured**: The device ID has not been set yet.

**Note:** If you experience any issues with device setup, Wi-Fi configuration, or data synchronization, please refer to the troubleshooting section or contact support for assistance.

**Feedback Overview**



The **Feedback** section is your direct line to the HydroSense team. We genuinely want to hear from you—whether it's a bug, a feature request, or just something cool you want to share.

## How to Use the Feedback Form

### ****Name****

Type in your full name so we know who’s giving us feedback. This helps us respond to you personally (if needed).

### ****Email****

Enter a valid email address. We’ll use it to follow up with you regarding your feedback or queries.

### ****Write your message here!****

* **This is your space to:**
  + Report issues or bugs
  + Suggest new features
  + Share your experience using HydroSense
  + Give general comments or shoutouts
* **Tip:** Be as specific and clear as possible so we can help you better.

### ****Submit Form****

Click this button once all fields are filled. Your message will be sent to the HydroSense support team instantly.

## What Happens Next?

* You’ll receive a confirmation email with 24 hours (if a valid address is provided).
* Our team will review your message and get back to you if a follow-up is needed.

**API Documentation**

**API Overview**

Hydrosense provides a simple, RESTful API interface to interact with the hardware (Vega Aries IoT V2.0 board) in real-time. This allows developers and testers to fetch sensor data or trigger hardware actions directly through HTTP endpoints.

### Base URL

* When connected to the Vega Aries board via its Wi-Fi hotspot (Access Point mode):

http://192.168.4.1/

* You must be connected to the device’s Wi-Fi to access this API.

### Available Endpoints

|  |  |  |  |
| --- | --- | --- | --- |
| **Endpoint** | **Method** | **Description** | **Example Response** |
| / | GET | Returns live sensor data in JSON format | { "ph": 6.8, "tds": 320, "temp": 28 } |
| /buzzer | GET | Triggers buzzer for 2 seconds (used for hardware testing) | "Buzzer ON for 2s" |



### Sample Use (Python)

import requests

res = requests.get("http://192.168.4.1/")

data = res.json()

print("pH:", data['ph'], "TDS:", data['tds'])

### Notes:

* This API is lightweight and designed for local, real-time hardware testing.
* No authentication is needed since it’s used over a local, closed Wi-Fi network.
* Extendable for future endpoints like /led, /reset, or /status

**Endpoint Descriptions**

These endpoints are served by the **Vega Aries IoT V2.0** board when it’s running in **Access Point (AP) mode**. You can interact with them by connecting to the device’s Wi-Fi and sending HTTP requests to http://192.168.4.1.

### 1. GET /

#### Description:

Fetches real-time data from all connected water quality sensors.

#### Response:

Returns a JSON object with the latest readings from the sensors.

#### Sample Output:

{

"ph": 7.12,

"tds": 335,

"turbidity": 3.1,

"temperature": 26.4,

"battery": 91,

"last\_test\_time": "2025-05-04T14:10:23Z"

}

#### Notes:

* This is the **main data endpoint**.
* Values are updated with each reading cycle on the microcontroller.
* Used for visualizing data on the dashboard and full charts.

### 2. GET /buzzer

#### Description:

Triggers the onboard buzzer for 2 seconds to test hardware alert systems.

#### Response:

A simple confirmation buzzer sound.

#### Sample Output:

Buzzer activates for 2 seconds

#### Notes:

* Primarily used in the **Hardware Status** section of the dashboard.
* Helps ensure the buzzer is functional during setup or maintenance.
* Safe to call multiple times.

**Troubleshooting**

Hydrosense is a robust system, but like any tech solution, it can face issues during setup or operation. Below is a general troubleshooting guide to help resolve some of the most common problems.

### General Troubleshooting Steps

* **Check Hardware Connections**:
  + Ensure that all hardware components are properly connected.
  + For Vega Aries IoT board:
    - Double-check sensor wiring (I2C/SPI connections).
    - Ensure that Wi-Fi module (NINA W102) is correctly connected and powered.
* **Reboot Devices**:
  + Power cycle the **Vega Aries board** and any connected sensors. This helps reset communication.
  + Restart the **web app** or **dashboard** for UI-related issues.
* **Verify Wi-Fi Connection**:
  + Ensure that the **Wi-Fi** is stable and correctly configured in the Vega Aries board.
  + If the board cannot connect to the network, check if the Wi-Fi credentials are correct and re-upload the code to the board.
  + Check if the device is within the Wi-Fi range.
* **Check Power Supply**:
  + If using external power (e.g., power adapters), make sure that the voltage and current meet the requirements for the **Vega Aries IoT board** and any sensors.

### Common Issues and Fixes

#### 1. ****Sensor Data Not Reading / Incorrect Values****

* **Check Wiring**: Ensure all sensors are correctly connected to the correct pins (e.g., I2C or SPI pins for the respective sensor).
* **Verify Sensor Calibration**: Some sensors may need to be calibrated before use. For example, the **pH sensor** requires calibration with a standard buffer solution.
* **Verify Sensor Code**: Ensure you are using the correct library for your sensor. For example, the **BME680** sensor requires specific initialization.

#### 2. ****Dashboard Not Loading / Connection Issues****

* **Clear Cache and Cookies**: Sometimes, issues with loading the dashboard can be caused by stale cached data.
* **Check Browser Compatibility**: Ensure you're using a modern browser (Chrome/Firefox). Some older browsers might not support all features of the app.
* **Server Down**: If the app isn’t loading, check the **server** status or restart the **Flask server** running the backend.
* **Check API Endpoints**: Ensure that the API URL endpoints are correctly configured and that they are accessible.

#### 3. ****AI Prediction Results Are Inaccurate****

* **Training Data**: Ensure that your model is trained with accurate and representative data. If necessary, retrain the model with new, more diverse data.
* **Overfitting/Underfitting**: Consider tuning model parameters or using a larger dataset if the model is overfitting or underfitting.
* **Model Update**: Periodically, the AI model might need updates to handle new types of water quality data. Reassess training and update your model.

#### 4. ****PDF/Excel Export Not Working****

* **Check Library Dependencies**: Ensure that the **jspdf** and **xlsx** libraries are properly installed and up-to-date.
* **Export Data Format**: Ensure the data passed to the export function is correctly formatted and doesn’t contain unsupported characters or data types.

### Debugging Tips

1. **Use Logs for Debugging**:
   * If you're encountering issues, always check the **serial monitor** or the **console logs** for error messages or warnings.
   * For web-related issues, **open the browser developer tools** (press F12) and check the **Network** tab for failing API calls or missing assets.
2. **Check API Response**:
   * Use **Postman** or **cURL** to manually test the API endpoints.
   * If the data response is wrong or missing, check the backend code or ensure the hardware is correctly transmitting data.

**Glossary**

This glossary provides definitions for key terms and acronyms used throughout the documentation to help users better understand the concepts, components, and processes involved in the **Hydrosense** project.

### **A**

* **API (Application Programming Interface)**: A set of rules and protocols that allow different software applications to communicate with each other. In **Hydrosense**, it is used to interact with the backend and exchange data between the hardware and web dashboard.
* **AI (Artificial Intelligence)**: A technology used in the **Hydrosense** system to analyze water quality data and make predictions, such as whether the water is potable or not.
* **Anomaly Detection**: A method used in **Hydrosense** to identify unusual patterns or data points in the water quality, which could indicate contamination or sensor malfunction.

### **B**

* **BME680**: A sensor used in the **Hydrosense** system for measuring temperature, humidity, pressure, and air quality (gas sensors).
* **Backend**: The part of the **Hydrosense** system responsible for processing data, handling business logic, and storing information. It interacts with the frontend (web dashboard) and the hardware components (sensors).

### **C**

* **Calibration**: The process of adjusting the sensors in the **Hydrosense** system to ensure accurate readings, especially for sensors like **pH** and **turbidity**.
* **CSV (Comma Separated Values)**: A file format used for storing tabular data, such as sensor readings or exported data from the **Hydrosense** system.
* **Cloud Storage**: The online storage service used to store data generated by the **Hydrosense** system, such as sensor data and AI model results.

### **D**

* **Dashboard**: The user interface of the **Hydrosense** web application, where users can view real-time water quality data, receive alerts, and generate reports.
* **Data Logging**: The process of continuously recording and storing sensor data over time in the **Hydrosense** system for later analysis.

### **E**

* **Endpoint**: A specific URL or address in the **Hydrosense** API that allows interaction with different resources, such as retrieving sensor data or uploading user information.
* **ESP32**: A low-cost, low-power microcontroller used in **Hydrosense** for connecting sensors to the internet via Wi-Fi and Bluetooth.
* **Event Handler**: A function in the **Hydrosense** web application that responds to specific user actions, such as button clicks or form submissions.

### **F**

* **Flask**: A micro web framework used in the backend of **Hydrosense** to create and handle web requests, including data submission from the frontend to the backend.
* **Frontend**: The part of the **Hydrosense** system responsible for the user interface, where users interact with the application (e.g., web dashboard).

### **G**

* **GPS (Global Positioning System)**: The technology used to track the geographical location of the **Hydrosense** device, which can help in mapping water quality across different regions.
* **Google Sheets API**: An API used in **Hydrosense** to fetch and update data from Google Sheets, which can be integrated with the backend for dynamic data input/output.

### **I**

* **I2C (Inter-Integrated Circuit)**: A communication protocol used by some sensors in the **Hydrosense** system, such as the **BME680** sensor, to exchange data with the microcontroller (e.g., ESP32).
* **ID Card Generation**: A feature in **Hydrosense** that allows users to generate a unique ID card with personal details, event participation, and a QR code for attendance verification.
* **IoT (Internet of Things)**: The technology that connects sensors, devices, and systems (like **Hydrosense**) to the internet for real-time data collection, monitoring, and control.

### **J**

* **JSON (JavaScript Object Notation)**: A lightweight data interchange format used in **Hydrosense** to send and receive structured data between the frontend and backend.

### **M**

* **Model Training**: The process of training the AI algorithm (like **SVM** for water quality analysis) with historical data to predict water quality conditions such as contamination.
* **Monitoring**: The real-time tracking of water quality data, including parameters like turbidity, pH, and TDS (Total Dissolved Solids), in the **Hydrosense** system.

### **P**

* **pH Sensor**: A sensor used in **Hydrosense** to measure the acidity or alkalinity of water, which is crucial for assessing water potability.
* **Postman**: A tool used to test and debug API endpoints in the **Hydrosense** system by simulating API requests and analyzing the responses.
* **Preprocessing**: The process of cleaning and preparing raw data (e.g., sensor readings) before it's used for AI model training or analysis.

### **Q**

* **QR Code**: A type of barcode used in **Hydrosense** for generating unique IDs for participants, which can be scanned for event registration and attendance tracking.

### **S**

* **SDK (Software Development Kit)**: A collection of tools, libraries, and documentation provided for developers to interact with **Hydrosense** components, such as the API or hardware.
* **Sensor Fusion**: The process of combining data from multiple sensors in **Hydrosense** to produce more accurate or reliable readings of water quality.
* **SVM (Support Vector Machine)**: An AI model used in **Hydrosense** for predicting water quality by analyzing data from sensors like pH, turbidity, and TDS.

### **T**

* **Turbidity Sensor**: A sensor used in **Hydrosense** to measure the cloudiness or haziness of water, which is an indicator of suspended particles and contamination.
* **Tokenization**: The process of converting sensitive data (e.g., user personal information) into tokens to protect privacy in **Hydrosense**.

### **W**

* **Wi-Fi Module (NINA-W102)**: A wireless communication module used in the **Hydrosense** system to connect the **Vega Aries IoT board** to the internet for data transmission and remote monitoring.
* **Webhooks**: A way for the **Hydrosense** system to send real-time data updates or alerts to other systems (like a mobile app or external server).

### **Z**

* **Zero Calibration**: The process of calibrating certain sensors (e.g., turbidity sensor) by exposing them to a known reference solution (e.g., clean water) to establish a baseline.

**Best Practices**

To get the most out of your **Hydrosense** system, follow these best practices for optimal performance, reliability, and user experience. These tips are designed to help you maintain the system, ensure accurate readings, and troubleshoot common issues efficiently.

### 1. ****Proper Sensor Calibration****

* **Why it matters**: Accurate sensor calibration is crucial for ensuring reliable water quality readings.
* **Best Practice**: Regularly calibrate your sensors, especially the **pH** and **turbidity** sensors, using certified calibration solutions. Refer to the calibration guide in the hardware manual for step-by-step instructions.
* **Tip**: Perform calibration after sensor installation and periodically (e.g., every month) to maintain precision.

### 2. ****Optimal Sensor Placement****

* **Why it matters**: The placement of your sensors can significantly impact data accuracy.
* **Best Practice**: Ensure that sensors are placed in locations that provide consistent and representative water samples, such as at the inlet, midpoint, and outlet of the water source.
* **Tip**: Avoid placing sensors in areas where they could be affected by turbulence or debris that might skew the readings.

### 3. ****Maintain Cleanliness of Sensors****

* **Why it matters**: Dirt, algae, or mineral buildup can interfere with sensor readings, leading to inaccurate data.
* **Best Practice**: Clean your sensors regularly using appropriate cleaning solutions recommended by the manufacturer.
* **Tip**: If using a **turbidity sensor**, ensure it is free from algae or particulate matter buildup to maintain accurate measurements.

### 4. ****Network Connectivity****

* **Why it matters**: Reliable connectivity is essential for transmitting data from the sensors to the backend system or cloud for analysis.
* **Best Practice**: Ensure your **Wi-Fi** or **Bluetooth** connection is stable and has sufficient range. If using multiple devices, consider a dedicated network for **Hydrosense** to minimize interference.
* **Tip**: Use a **Wi-Fi extender** or **mesh system** if your device placement is far from the router.

### 5. ****Regular Data Monitoring****

* **Why it matters**: Monitoring data regularly allows for early detection of water quality issues or sensor malfunctions.
* **Best Practice**: Use the **Hydrosense** dashboard to check water quality data frequently. Set up automatic alerts for abnormal readings (e.g., high turbidity or pH changes).
* **Tip**: Enable email or SMS notifications for critical events, ensuring you can take immediate action when needed.

### 6. ****Data Backup and Export****

* **Why it matters**: Storing and backing up sensor data ensures you have a history for analysis and comparison.
* **Best Practice**: Regularly export your data to a secure storage platform (e.g., cloud storage, CSV files) to prevent data loss.
* **Tip**: Set up automatic backup schedules through your backend system or API to ensure that you always have the latest data saved.

### 7. ****AI Model Re-Training****

* **Why it matters**: Over time, the AI model’s predictions may drift as environmental conditions change.
* **Best Practice**: Retrain the AI model periodically using the latest data to maintain its accuracy.
* **Tip**: If your system collects data from new sources or locations, update your training dataset to improve model performance.

### 8. ****Test Webhooks and API Integration****

* **Why it matters**: Webhooks and API integrations allow for real-time updates and external system communication.
* **Best Practice**: Test webhooks and API integrations before deploying them to ensure they are working correctly.
* **Tip**: Use tools like **Postman** or **Insomnia** to simulate API requests and verify that data is being sent and received as expected.

### 9. ****System Security****

* **Why it matters**: Security is critical to protect sensitive water quality data and user information.
* **Best Practice**: Implement secure authentication methods for accessing the **Hydrosense** dashboard and API. Use encryption (HTTPS) for data transfer.
* **Tip**: Regularly update passwords and restrict access to the system to authorized users only.

### 10. ****Efficient Use of Resources****

* **Why it matters**: Optimizing power consumption and data usage can improve system efficiency.
* **Best Practice**: If using **battery-powered devices** or **solar-powered setups**, ensure energy-efficient configurations to prolong operation time.
* **Tip**: Consider using sleep modes for sensors when continuous monitoring is not necessary, or configure your system to send data at scheduled intervals.

### 11. ****Event and Alert Management****

* **Why it matters**: Being proactive in managing events and alerts can help prevent system malfunctions or water contamination.
* **Best Practice**: Set thresholds for water quality parameters such as **pH**, **Turbidity**, and **Temperature**. When these thresholds are crossed, the system will trigger an alert.
* **Tip**: Customize alert settings based on critical water quality metrics. For example, trigger an alert when **pH** falls below 6.5 or when **Turbidity** exceeds safe limits.

**Appendix**

**Technical Specifications**

### 1. ****System Overview****

The **Hydrosense** project is a water quality monitoring system designed to track various water quality parameters in real time, including **pH**, **turbidity**, **temperature**, **TDS**, and **water pressure**. The system includes both hardware (sensors and IoT devices) and software components (web and mobile app, backend services, and APIs).

### 2. ****Hardware Specifications****

#### 2.1 ****Microcontroller****

* **Board**: **VEGA ARIES IoT v2.0**
* **Processor**: **THEJAS32 SoC** with **VEGA ET1031 Microprocessor**
  + **Clock Speed**: 100 MHz
  + **RAM**: 256 KB SRAM
  + **Flash Memory**: 2 MB Flash
* **Wi-Fi Module**: **NINA-W102-01B** (Wi-Fi and Bluetooth connectivity)
* **GPIO Pins**: Multiple GPIO pins for sensor interfacing
* **Other Peripherals**: UART, SPI, I2C, PWM

#### 2.2 ****Sensors****

* **pH Sensor**:
  + **Type**: Analog pH sensor
  + **Measurement Range**: pH 0 - 14
  + **Accuracy**: ±0.1 pH
* **Turbidity Sensor**:
  + **Type**: Analog turbidity sensor
  + **Measurement Range**: 0-1000 NTU (Nephelometric Turbidity Units)
  + **Accuracy**: ±5%
* **Temperature Sensor**:
  + **Type**: **DS18B20** Digital Temperature Sensor
  + **Measurement Range**: -55°C to +125°C
  + **Accuracy**: ±0.5°C
* **TDS (Total Dissolved Solids) Sensor**:
  + **Type**: Analog TDS sensor (e.g., **Gravity TDS Sensor**)
  + **Measurement Range**: 0-1000 ppm
  + **Accuracy**: ±5%
* **Water Pressure Sensor**:
  + **Type**: **DF-Robot Pressure Sensor**
  + **Output**: mV/V
  + **Measurement Range**: 0-10 bar (adjustable)
  + **Accuracy**: ±1%

#### 2.3 ****Power Supply****

* **Voltage**: 5V via USB or 12V DC power supply
* **Power Consumption**: ~300mA during operation (varies by sensor load)

### 3. ****Software Specifications****

#### 3.1 ****System Architecture****

* **Frontend**:
  + **Web App**: Built with **ReactJS** and **Redux** for state management, integrated with **Chart.js** for data visualization.
  + **Mobile App**: Built with **React Native** for cross-platform mobile development.
* **Backend**:
  + **Language**: **Python 3.x** with **Flask** for RESTful API services.
  + **Database**: F**irebase** for storing sensor data and user profiles.
  + **Web Server**: **NGINX** for reverse proxying and load balancing.

#### 3.2 ****Communication Protocols****

* **Wi-Fi Communication**: Vega-Board communicates over **Wi-Fi (802.11b/g/n)** using the **built-in module** module.
* **HTTPS**: For secure communication between the frontend, backend, and IoT devices.

#### 3.3 ****Cloud Integration****

* **Storage**: **Firebase** for storing sensor calibration data and user-uploaded files (e.g., reports, photos).

#### 3.4 ****Data Security & Encryption****

* **Data Encryption**: All sensor data and user information are encrypted using **SSL/TLS** during transmission between devices, servers, and the cloud.
* **Authentication**:
  + **JWT (JSON Web Tokens)** for user authentication and authorization.
  + **OAuth 2.0** for third-party integrations.

#### 3.5 ****API****

* **REST API**: For external integration and data access. Built using **Flask** with routes for reading sensor data, managing users, and receiving alert notifications.
* **API Authentication**: API access is secured using **API keys** and **OAuth 2.0**.

#### 3.6 ****Data Formats****

* **Sensor Data**: JSON format (example):

json

CopyEdit

{

"timestamp": "2025-05-05T12:34:56Z",

"pH": 7.2,

"turbidity": 0.5,

"temperature": 22.5,

"TDS": 180,

"pressure": 2.5

}

* **Exported Data**: CSV/PDF formats for user download.

### 4. ****Maintenance and Calibration****

* **Sensor Calibration**:
  + **pH**: Manual calibration with standard buffer solutions (pH 4, 7, 10).
  + **Turbidity**: Calibrated using a standard solution and adjusted via software.
  + **TDS**: Calibration is done using a known TDS solution.
* **Maintenance**: Sensors require periodic cleaning and recalibration every 6 months.

**SDKs and Libraries**

Hydrosense leverages a mix of **hardware microcontroller libraries** and **modern web + AI software libraries** to deliver a robust real-time water quality monitoring experience.

### 1. Hardware (Arduino – Vega Aries IoT V2.0)

These libraries power the firmware running on the Vega Aries board:

|  |  |
| --- | --- |
| **Library** | **Purpose** |
| SPI.h | Enables **SPI (Serial Peripheral Interface)** communication for fast sensor or peripheral data transfer. |
| WiFiNINA.h | Enables Wi-Fi access for Vega Aries using **NINA W102 module**; used for setting up AP mode and HTTP server. |
| Wire.h | Powers **I2C communication**, connecting sensors like temperature or TDS with low latency. |
| Arduino\_APDS9960.h | Optional sensor library for **gesture, proximity, and light sensing** (if used). |



### 2. Software (Frontend – React + Firebase)

|  |  |
| --- | --- |
| **Library** | **Purpose** |
| react | Core library for building the component-based UI. |
| react-dom | DOM-specific methods for rendering React components in the browser. |
| react-router-dom | Enables **single-page routing** without full reloads. |
| redux | Manages global state across the app. |
| react-redux | Connects Redux with React components. |
| redux-thunk | Middleware for handling **asynchronous logic** inside Redux. |
| redux-logger | Logs dispatched actions and state changes (debugging aid). |
| react-icons | Loads icon packs like Material Icons, FontAwesome. |
| react-toastify | Displays customizable **toast notifications** for alerts or updates. |
| tailwindcss | Utility-first **CSS framework** for responsive, clean UI styling. |
| jspdf | Generates **PDF reports** from web data (e.g., sensor summaries). |
| jspdf-autotable | Creates well-formatted **tables in PDFs**, used in downloads. |
| xlsx | Reads/writes Excel files for **exporting sensor data** in .xlsx format. |
| web-vitals | Monitors real-world performance metrics like load speed, interactivity. |



### 3. AI & Python Backend

|  |  |
| --- | --- |
| **Library / Tool** | **Purpose** |
| scikit-learn | Trains and uses the **Support Vector Regression (SVR)** model for water quality prediction. |
| Flask | Lightweight Python web framework to serve APIs like the **AI Bot** and **summarization tool**. |
| numpy | Used for numerical computations during sensor simulations and model predictions. |
| pandas | Manages structured sensor data for training and AI analytics. |

**Conclusion**

HydroSense represents a leap forward in accessible, real-time water quality monitoring by seamlessly integrating IoT hardware, intelligent software, and data analytics into a single, powerful platform. Designed with portability, accuracy, and user experience in mind, the system empowers individuals, researchers, and communities to take control of their water safety with actionable insights.

From sensor calibration and anomaly detection to digital reports and mobile integration, HydroSense is not just a tool—it's a scalable solution tailored for the environmental challenges of today and tomorrow. With continued development, modular expansion, and open-source potential, HydroSense is poised to grow as both a research platform and a practical deployment-ready product.

In a world increasingly concerned with sustainability and health, HydroSense delivers not just data—but trust, transparency, and transformation.

**THANKYOU FOR USING**

**HYDROSENSE**