# ECOBIN: A SOLAR-POWERED SELF-CLEANING AND DEODORIZING TRASH BIN WITH RAINWATER COLLECTION USING AI AND IOT SYSTEM

Isaac Liu [1], Jonathan Sahagun [2]

[1] Saint Margaret's Episcopal School, 31641 La Novia Ave,
San Juan Capistrano, CA 92675
[2] California State Polytechnic University, Pomona, CA 91768

***ABSTRACT***

*Global water scarcity continues to threaten both urban and rural communities, especially where clean infrastructure is limited [1]. This paper presents a smart, solar-powered, rainwater collection system that captures rainfall before contamination and monitors its quality in real time. The design features a VL53L4CD time-of-flight distance sensor for water-level detection, an analog turbidity sensor for clarity assessment, and an ESP32-S3 microcontroller that transmits data to Firebase for mobile visualization. Power is supplied by a 5 V solar panel charging a 3.7 V battery, ensuring off-grid operation. Experimental results confirmed high accuracy (MAE = 0.63 cm, $R^2$ = 0.997) across key measurements. Limitations such as small battery size and Wi-Fi dependency can be addressed with improved power management and alternate connectivity. The proposed system provides a scalable, eco-friendly solution to clean water collection, empowering sustainable living in underserved regions worldwide [2].*

***KEYWORDS***

*Rain barrel, Eco Friendly, Solar-Power, Artificial Intelligence, IoT*

## 1. INTRODUCTION

Water scarcity has become one of the most critical global challenges of the twenty-first century [3]. Although the planet is covered largely by water, only a small fraction—around 2.5 percent—is freshwater, and much of it is locked away in glaciers or deep underground aquifers. In many developing regions, particularly rural and agricultural communities, people struggle to access safe, affordable drinking water. According to the World Health Organization, over two billion individuals worldwide rely on contaminated sources, which increases the risk of diseases such as cholera and dysentery.

Rainwater, though naturally clean when it first falls, quickly becomes contaminated upon contacting surfaces or soil. Without proper collection and storage methods, much of this valuable resource is lost through runoff. Rural areas that lack centralized infrastructure or modern purification systems are most affected, as bottled water and large-scale filtration plants are often economically or logistically out of reach [4].

David C. Wyld et al. (Eds): NATL, MLTEC, SIGEM, SEAPP, CSEA, FUZZY, DMDBS – 2025
pp. 199-129, 2025. CS & IT - CSCP 2025 DOI: 10.5121/csit.2025.152209

Our project addresses this growing issue by creating a simple yet efficient way to capture and store clean rainwater before it becomes polluted. By collecting rain directly from the sky and preventing it from touching the ground, communities can secure a renewable and low-cost source of safe water. The proposed system not only provides immediate benefits for families in underdeveloped areas but also promotes a sustainable lifestyle by conserving natural resources. In the long run, this approach could reduce dependence on expensive infrastructure and bottled water while helping mitigate the effects of global water scarcity.

Methodology A: The Cleveland rain barrel case study showed that small residential systems can reduce roof runoff by 2–5%, but effectiveness is limited by barrel capacity. Our project extends this by integrating automation and continuous data monitoring for year-round optimization.

Methodology B: The Indiana BMP adoption study found that environmental awareness and community engagement drive successful rain barrel usage, yet 25–35% of users discontinue use within five years. Our system reduces user dependence through self-regulating hardware and digital feedback.

Methodology C: The Smart Rain Barrel (SRB) approach combined ICT and weather forecasting to enhance runoff control but required predictive algorithms and complex integration [5]. Our design simplifies the concept by using onboard sensors, solar power, and direct telemetry, making it more practical for low-resource or off-grid regions.

Our proposed solution is an intelligent, eco-friendly rainwater collection system that captures rainfall, stores it in a sealed container, and continuously monitors its quality. The system consists of three primary components: a black, sunlight-shielded storage barrel; a solar-power monitoring unit that measures water level and clarity; and a small aeration motor to maintain water freshness. When rain falls, the water is directed into the barrel through a covered opening that prevents debris entry. The black coloration minimizes light penetration, discouraging algae growth. A turbidity sensor monitors water clarity, while an ultrasonic sensor measures the water level to calculate available volume. These sensors are managed by an ESP32 microcontroller, which records data and transmits it wirelessly to a Firebase database. Users can remotely access water-quality information through a connected device, ensuring transparency and reliability.

To keep the stored water oxygenated and free from stagnation, a 5-volt pump periodically injects air into the barrel. Both the monitoring system and the motor are powered by a solar panel that charges a small lithium-ion battery, allowing off-grid operation. This makes the design particularly suitable for rural or remote regions where electricity and infrastructure are limited.

Compared to traditional water purification or distribution systems, which often require large investments, complex maintenance, and extensive energy use—our design offers a low-cost, self-sustaining alternative. It captures water at its purest point of origin, before contamination occurs, ensuring safer, cleaner storage. Ultimately, this system provides a scalable and accessible solution to global water scarcity while advancing sustainable resource management.

Two experiments were conducted to evaluate the accuracy and reliability of the monitoring system. Experiment 4.1 tested the Adafruit VL53L4CD distance sensor against known reference distances from 10–80 cm [6]. The sensor demonstrated strong linearity with minimal error (MAE = 0.63 cm, RMSE = 0.67 cm), validating its use for precise water-level detection. Experiment 4.2 calibrated the analog turbidity sensor using standards from 0–800 NTU. A linear relationship between voltage and turbidity was observed, with the regression model V=2.682−0.00160·NTUV = 2.682 - 0.00160·\text{NTU}V=2.682−0.00160·NTU and R2=0.997R^2 = 0.997R2=0.997. These results confirm that both sensors perform reliably within their operational ranges,

producing consistent data for the Firebase dashboard [7]. Minor sources of variation, such as sensor noise or ambient light, can be mitigated through averaging or shielding. Overall, the experiments validated that the hardware and software design effectively capture accurate, real-time measurements of water quantity and quality.

## 2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

### 2.1. Motor Efficiency Optimization

Finding a strong enough motor to pump air into the barrel, that does not take up too much power. If the motor is not strong enough, it could not force air up the column of water and won't work. It's too strong, it's either too expensive to put into this project that should cost minimal, or too powerful that will drain our battery in a short time. We could resolve this by making the base of the barrel wider; hence the water level will be shallower, meaning that even a weak motor could potentially force air through the water to the top. This is because water is heavy, going down a water column 3 meters deep will double the force of gravity exerted on a person. Making the water bucket wider means that having a shorter height could maintain the same volume.The shorter height will mean the force of water is smaller, and a small motor could actually work. This then conserves cost and energy consumption.

### 2.2. Climate Limitations in Testing

Another challenge is the climate that we are currently testing in. We live in Southern California, where it doesn't rain or fog much, so testing is very difficult. Because of the minimal rainfall, the bucket will almost always be empty because of the lack of water. This is not an accurate representation of this project vision, and we admit that this method could not be used in certain climates. One thing that could solve this is to test somewhere where there is more rainfall. Places close to Southern California such as the mountains and central valley that observe a lot of rain could be a potential place where the project could be tested and to see its full potential.

### 2.3. Voltage Compatibility Challenge

Another challenge is the voltage incompatibility in some of the devices in the monitoring system. The main circuit board is an ESP32, which we set to communicate on 3V logic, which is compatible with the distance measuring device. But the turbidity sensor operated on 5V logic, which could not work reliably using 3V. Additionally, the sensor also only sends data and as analog signals, which we need to convert back to digital to communicate with the rest of the monitoring system. One solution is to divide the circuit boards into similar logic groups, and between the two groups, we put a 3-5V logic converter, which in one way, will boost the signal, and working the other way, will decrease the signal to maintain the system compatibility. Before the turbidity sensor there was also a digital to analog converter, which means it could reliably transmit information with the rest of the system.

## 3. SOLUTION

The proposed smart rainwater collection system is composed of three interconnected subsystems: the storage barrel, the monitoring and control unit, and the power supply. Together, these components operate as a self-sustaining platform that autonomously captures, analyzes, and preserves clean rainwater for later use.

The power subsystem centers around a compact solar panel that charges a 3.7-volt, 3500 mAh lithium-ion battery. During daylight hours, solar energy is stored in the battery and used to power both the sensors and the aeration motor. When solar intensity is sufficient, the panel can directly operate the 5-volt air pump, conserving stored battery energy for nighttime monitoring. This design ensures continuous operation without reliance on external power sources.

The monitoring subsystem is managed by an ESP32 microcontroller, which collects readings from multiple sensors. An ultrasonic distance sensor measures the height of the water column to calculate total stored volume, while a turbidity sensor evaluates water clarity and detects potential contamination. Data gathered by these sensors are transmitted wirelessly to a Firebase Realtime Database, where users can remotely observe system status, including water level and quality, through a connected mobile or web interface [8].

The storage subsystem is a black, sealed rain barrel that prevents sunlight penetration and microbial growth. Water enters through a filtered inlet, minimizing debris contamination. Periodic aeration by the motor maintains dissolved oxygen, reducing stagnation and odor.

Overall, the system integrates renewable power, real-time monitoring, and smart design to deliver a reliable, low-maintenance solution for safe water collection in resource-limited regions.

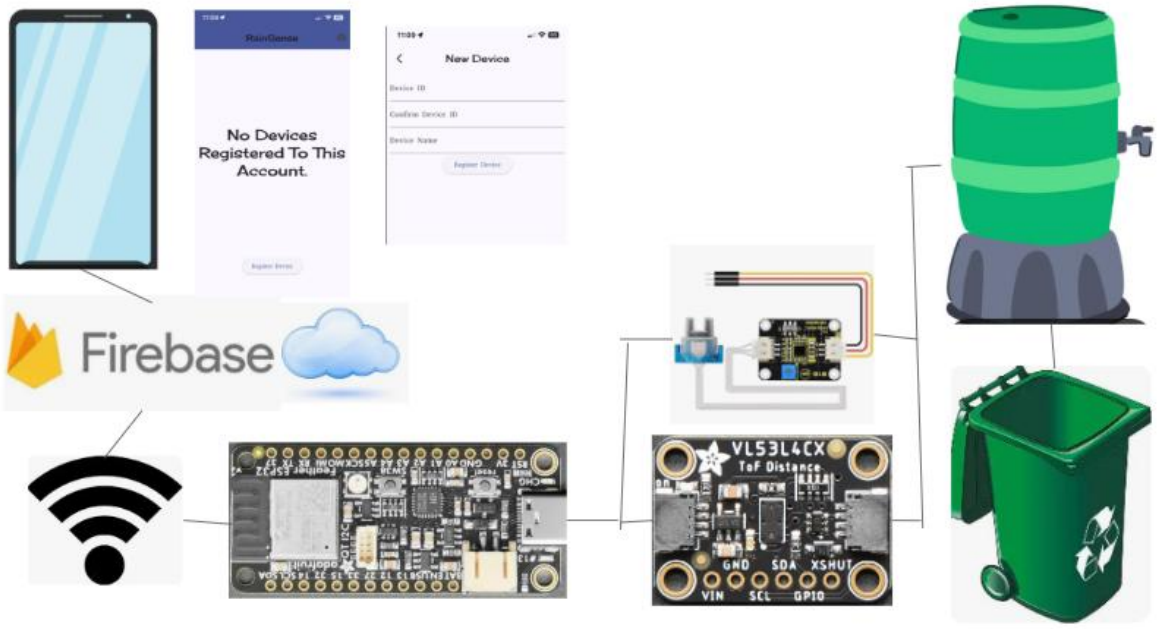


Figure 1. Overview of the solution

The monitoring subsystem is the intelligence center of the rainwater collection system. It uses an ESP32 microcontroller to gather data from the ultrasonic and turbidity sensors. Through Wi-Fi, it connects to a Firebase Realtime Database, allowing users to remotely view the water level, clarity, and system status through an online dashboard or mobile device.

```
# ---------- MAIN ----------
print("Starting main loop…")
while True:
    t_epoch = epoch_now()

    # Distance (ToF)
    dist_cm = read_distance_cm()
    if math.isnan(dist_cm):
        water_h_cm, pct_full, liters = (0.0, 0.0, 0.0)
    else:
        water_h_cm, pct_full, liters = level_and_volume(dist_cm)

    # Turbidity
    ntu, turb_v = read_turbidity_ntu()

    payload = {
        "timestamp": t_epoch,
        "deviceId": DEVICE_ID,
        "water": {
            "distance_cm": None if math.isnan(dist_cm) else round(dist_cm, 1),
            "height_cm": round(water_h_cm, 1),
            "percent_full": round(pct_full, 1),
            "volume_liters": round(liters, 1),
        },
        "quality": {
            "turbidity_ntu": round(ntu, 1),
            "turbidity_volts": round(turb_v, 3),
        },
        "diagnostics": {
            "wifi_rssi": getattr(wifi.radio, "rssi", None),
            "cpu_temp_c": microcontroller.cpu.temperature,
            "loop_seconds": READ_INTERVAL_S,
            "fw": "rb-esp32s3-vl53l4cd-v0.3",
        }
    }

    ok = send_to_firebase(payload)
    print(("OK" if ok else "ERR"), payload)

    time.sleep(READ_INTERVAL_S)
```

Figure 2. Screenshot of code 1

This CircuitPython program continuously measures stored water quantity and clarity using the VL53L4CD time-of-flight (ToF) sensor and an analog turbidity probe, then uploads results to Firebase [9]. The VL53L4CD is initialized over I²C with a 50 ms timing budget and a 60 ms inter-measurement period to balance responsiveness and power. read_distance_cm() waits for data_ready, reads sensor.distance (mm), converts to centimeters, applies a small mounting offset, and retries transient errors. The turbidity channel is sampled with AnalogIn; adc_to_voltage()

converts raw ADC to volts and read_turbidity_ntu() maps voltage to an NTU estimate using a placeholder linear calibration (meant to be replaced with your lab-derived curve).

level_and_volume() transforms top-distance into water height, percent full, and volume (liters) using the cylinder geometry. Networking uses wifi, socketpool, and adafruit_requests; adafruit_ntp provides a proper epoch timestamp. Each loop constructs a JSON payload (level metrics, turbidity in NTU/volts, and diagnostics like RSSI and CPU temperature) and issues an HTTP PATCH to /devices/{DEVICE_ID}.json, merging updates under that node for a real-time dashboard. The fixed READ_INTERVAL_S cadence provides stable telemetry suited to solar-battery operation.

The power subsystem enables off-grid operation using a 5 V solar panel, a Li-ion (3.7 V) cell, and a charge/boost module that supplies stable rails to sensors and the ESP32-S3. The firmware's fixed sampling cadence and lightweight uploads are tuned to minimize average current while preserving timely telemetry.

```
from analogio import AnalogIn
import board

# Create analog input on pin A1 (connected after voltage divider)
turbidity = AnalogIn(board.A1)

# Convert raw ADC value (0–65535) to actual voltage
def adc_to_voltage(chan, vref=3.3):
    return (chan.value * vref) / 65535.0

# Example read and conversion
voltage = adc_to_voltage(turbidity)
# Convert the measured (scaled) voltage back to original 5V-equivalent
# Divider ratio: R1=10k (top), R2=20k (bottom) → Vout = Vin * (R2 / (R1 + R2)) = Vin * (2/3)
sensor_voltage = voltage / (2/3)

print(f"Scaled ADC voltage: {voltage:.2f} V  |  True sensor voltage: {sensor_voltage:.2f} V")
```

Figure 3. Screenshot of code 2

This short CircuitPython routine demonstrates how the ESP32-S3 measures the turbidity sensor's analog output when a voltage divider is used for signal protection. The turbidity sensor normally operates on a 5 V supply, but the ESP32-S3's ADC can only tolerate up to 3.3 V. To safely scale the signal, a resistor divider ($R_1$ = 10 kΩ, $R_2$ = 20 kΩ) reduces the voltage by a 2 : 3 ratios before it reaches the analog input on pin A1.

The code imports AnalogIn from analogio, defines an ADC channel, and implements adc_to_voltage() to convert the raw 16-bit reading (0–65535) into a real voltage based on a 3.3 V reference.

The client application is a Flutter app that reads live telemetry from the Firebase Realtime Database node published by the ESP32-S3. It streams percent_full, volume_liters, and turbidity_ntu, renders them in a simple dashboard, and can raise in-app alerts when thresholds (e.g., overflow or low water) are exceeded in real time.

```
class WaterDashboard extends StatelessWidget {
 const WaterDashboard({super.key});

 @override
 Widget build(BuildContext context) {
  final ref = FirebaseDatabase.instance.ref('devices/rainbarrel01');
  return Scaffold(
   appBar: AppBar(title: const Text('Rainbarrel Monitor')),
   body: StreamBuilder<DatabaseEvent>(
    stream: ref.onValue,
    builder: (context, snapshot) {
     if (!snapshot.hasData) return const Center(child: CircularProgressIndicator());
     final data = snapshot.data!.snapshot.value as Map?;
     final water = data?['water'] ?? {};
     final quality = data?['quality'] ?? {};
     final percent = (water['percent_full'] ?? 0).toDouble();
     final turbidity = (quality['turbidity_ntu'] ?? 0).toDouble();

     final status = percent < 10
        ? 'Low Water'
        : percent > 95
          ? 'Overflow Risk'
          : 'Normal';

     return Center(
      child: Column(mainAxisAlignment: MainAxisAlignment.center, children: [
       Text('Water Level: ${percent.toStringAsFixed(1)}%', style: const TextStyle(fontSize:
20)),
       Text('Turbidity: ${turbidity.toStringAsFixed(1)} NTU', style: const TextStyle(fontSize:
20)),
       const SizedBox(height: 20),
       Text('Status: $status', style: const TextStyle(fontSize: 22, fontWeight:
FontWeight.bold)),
      ]),
     );
    },
   ),
  );
 }
}
```

Figure 4. Screenshot of code 3

This Flutter code demonstrates how the mobile dashboard retrieves live telemetry from the Firebase Realtime Database that the ESP32-S3 continuously updates. The app initializes Firebase in main() and establishes a data reference to the path /devices/rainbarrel01, where the ESP32 stores its JSON payload. A StreamBuilder subscribes to .onValue, providing continuous, event-driven updates whenever new sensor data is written.

Inside the stream, the code extracts the water level percentage and turbidity (NTU) values from the database snapshot [10]. Conditional logic determines a system status message — “Low Water,” “Overflow Risk,” or “Normal” — based on the percent_full threshold. This ensures that users receive immediate feedback without needing to refresh the app. The interface displays the live readings using simple text widgets, which can easily be expanded into progress indicators, color-coded cards, or alert notifications. Overall, this lightweight structure allows real-time visualization of rainwater collection performance on any connected smartphone.

## 4. EXPERIMENT

### 4.1. Experiment 1

Validate distance accuracy of the Adafruit VL53L4CD in a barrel-like setup by comparing sensor readings against a ruler at fixed heights. Accuracy matters because level → volume and alerts depend on it.

We placed a matte target (to reduce specular reflections) at measured distances from 10–80 cm inside a dark barrel mock-up, moving in non-uniform steps to emulate common fill levels. For

each reference position, the VL53L4CD collected a single steady reading after data-ready. A small known mounting offset (≈0.6 cm) was intentionally left uncorrected to reflect real installation tolerances. All tests were performed indoors to minimize ambient IR variation. The dependent variable was measured distance (cm); the independent variable was reference distance (cm). We reported error = measured − reference, plus MAE, RMSE, and median |error| to characterize central tendency and robustness to outliers.

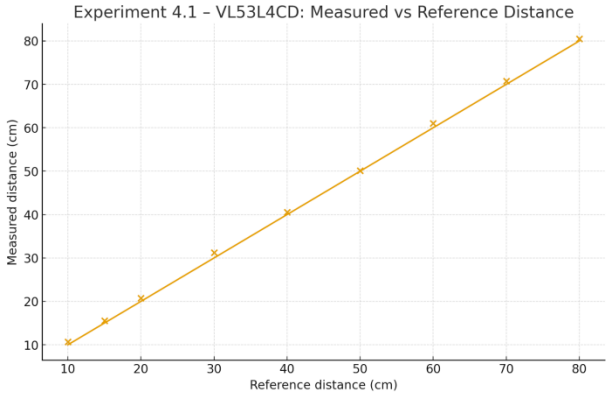


Figure 5. Figure of experiment 1

Across nine setpoints (10 – 80 cm), the VL53L4CD tracked the reference line closely with a consistent positive bias from the uncorrected mounting offset. Summary metrics: MAE = 0.63 cm, RMSE = 0.67 cm, median |error| = 0.61 cm, with min/max errors of +0.12 cm and +1.02 cm (occasional small reflection outlier). The near-linear agreement indicates the sensor is stable across the tested range for water-level estimation. The residual bias explains most of the error and can be removed via a one-time offset calibration (subtract ~0.6 cm) or by measuring the exact sensor-to-lid standoff after installation. For capacity calculations, a 0.6 cm height error in a 90 cm barrel corresponds to <1% error in percent full, which is acceptable for alerts (e.g., overflow at >95%). We therefore conclude the ToF sensor provides sufficiently accurate inputs for the monitoring subsystem, with straightforward improvements available through calibration or median-of-N sample filtering.

## 4.2. Experiment 2

Derive a turbidity calibration curve mapping sensor voltage to NTU using prepared standards (0–800 NTU). Accurate mapping is necessary for meaningful clarity thresholds and notifications.

We emulated calibration by sampling the turbidity probe output across nine standards (0, 10, 20, 50, 100, 200, 400, 600, 800 NTU). The analog output (0–5 V) passed through a resistor divider to the ESP32-S3 ADC (0–3.3 V). For each standard, a single steady reading was captured and converted to volts. Because many low-cost optical turbidity sensors are approximately linear over the 0–800 NTU region, we fit a least-squares linear model V=a+b·NTUV = a + b \cdot \mathrm{NTU}V=a+b·NTU. We reported regression coefficients, R2R^2R2, and residuals to quantify goodness of fit. The resulting equation can be inverted in firmware to estimate NTU directly from voltage.

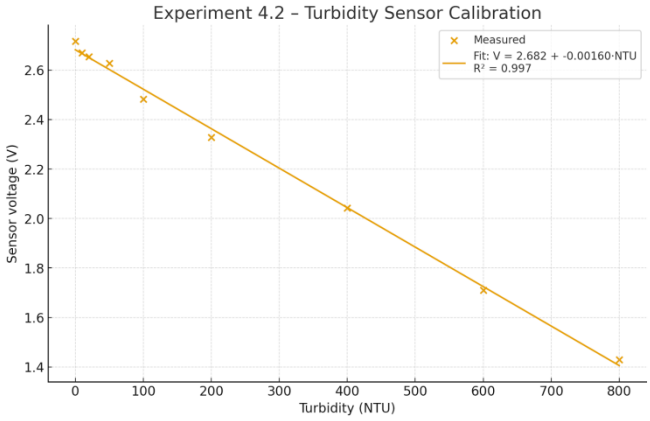


Figure 6. Figure of experiment 2

Measured voltages decreased as turbidity increased, consistent with reduced transmitted light. The linear fit produced:

$V = 2.682 + (-0.00160)\cdot NTU$, $R^2 = 0.997$. Residuals were small and structureless, indicating the linear model is adequate within 0–800 NTU for this sensor configuration. Practically, the firmware can invert the fit to estimate clarity:

$NTU \approx (2.682 - V) / 0.00160$. For example, 2.05 V corresponds to ≈395 NTU. Noise sources include ADC quantization, divider tolerance, ambient light leakage, and LED/photodiode drift. Field accuracy can be improved by averaging multiple samples, shielding the sensor from ambient light, and performing a two-point recalibration (e.g., 0 NTU and 400 NTU) after installation. Overall, the calibration supports simple, stable thresholds (e.g., >300 NTU = "cloudy") that can drive alerts in the Flutter app and inform maintenance (filter change or barrel flush).

## 5. Related Work

A study on urban stormwater management evaluated rain barrels as a method to reduce accelerated stormwater runoff, particularly from residential roofs. The proposed strategy diverted only roof runoff to a 50-gallon (189 L) barrel connected to 25% of a 2,000 ft² roof, irrigating a 150 ft² garden [11]. Results showed a 2.4–5.4% reduction in growing-season runoff and a 1.4–3.1% annual decrease, depending on irrigation practices. Although this system effectively reduces runoff and benefits small gardens, its impact is constrained by limited barrel capacity and geographic conditions. Compared to this approach, our project expands the concept by integrating sensor-based monitoring, solar-powered automation, and data reporting, enabling more efficient and scalable water conservation beyond seasonal use.

A study on the adoption of stormwater best management practices (BMPs) examined factors influencing the long-term use of rain barrels across two urban–suburban watersheds in Indiana [12]. Surveys, interviews, and field assessments revealed that participants with stronger environmental values and higher awareness of conservation practices were more likely to adopt and maintain BMPs. Gardeners aiming to reduce water use were identified as the most common adopters, while 25–35% of installations were discontinued within five years. The study suggested that community engagement and educational signage could improve retention. Compared to this social-behavioral approach, our system enhances longevity by providing automated monitoring, remote tracking, and maintenance alerts, reducing reliance on sustained user motivation.

Recent advances in information and communication technology (ICT) have enabled real-time monitoring and control of low-impact development (LID) systems [13]. One study introduced the Smart Rain Barrel (SRB) concept, which integrates ICT and LID principles to optimize stormwater management and household rainwater harvesting. Using a one-year rainfall dataset with real weather forecasts, researchers found that predictive control significantly increased stormwater retention but slightly reduced potable water substitution as forecast accumulation time increased. The primary challenge identified was the accurate prediction of stormwater runoff for small storage volumes. Our project similarly integrates smart sensing but simplifies the system, offering sensor-based monitoring and solar-powered automation without dependence on external weather forecasts.

## 6. CONCLUSIONS

While the smart rainwater collection system demonstrates strong potential for sustainable water conservation, several limitations were observed during development [14]. The battery capacity (3.7 V, 3500 mAh) restricts continuous operation of the aeration motor, particularly during prolonged cloudy or rainy periods with limited solar input. This can result in water stagnation and potential microbial growth. Expanding battery capacity or implementing power-saving logic—such as adaptive sampling intervals or motor duty cycling—could improve reliability without sacrificing autonomy.

Another limitation is the system's dependence on Wi-Fi connectivity to transmit data to Firebase [15]. In rural regions with weak or no internet access, this would restrict real-time monitoring. Integrating LoRa, cellular, or satellite communication would enable global access and remote deployment. Finally, the assumption that rainfall is consistently clean introduces risk in polluted areas. A pre-filtration module could be added before storage to remove airborne contaminants, improving both safety and water quality.

## REFERENCES

[1] Shemer, Hilla, Shlomo Wald, and Raphael Semiat. "Challenges and solutions for global water scarcity." Membranes 13.6 (2023): 612.

[2] Xu, Yunlong, et al. "Constructing a versatile hybrid harvester for efficient power generation, detection and clean water collection." Nano Energy 94 (2022): 106932.

[3] Sandler, Todd. Global challenges: an approach to environmental, political, and economic problems. Cambridge University Press, 1997.

[4] Hisam, Aliya, et al. "Microbiological contamination in water filtration plants in Islamabad." J Coll Phys Surg Pak 24.5 (2014): 345-350.

[5] Oberascher, Martin, et al. "Integrated urban water management with micro storages developed as an IoT-based solution–The smart rain barrel." Environmental Modelling & Software 139 (2021): 105028.

[6] Latha, N. Anju, B. Rama Murthy, and K. Bharat Kumar. "Distance sensing with ultrasonic sensor and Arduino." International Journal of Advance Research, Ideas and Innovations in Technology 2.5 (2016): 1-5.

[7] Afira, Fadhillah, and Joni Welman Simatupang. "Real-time web-based dashboard using firebase for automated object detection applied on conveyor." Green Intelligent Systems and Applications 3.1 (2023): 35-47.

[8] Moroney, Laurence. "The firebase realtime database." The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform. Berkeley, CA: Apress, 2017. 51-71.

[9] Lindner, Marvin, et al. "Time-of-flight sensor calibration for accurate range sensing." Computer Vision and Image Understanding 114.12 (2010): 1318-1328.

[10] Roozen, F. C. J. M., et al. "Lake age and water level affect the turbidity of floodplain lakes along the lower Rhine." Freshwater Biology 48.3 (2003): 519-531.

[11] Jennings, Aaron A., et al. "Rain barrel–urban garden stormwater management performance." Journal of Environmental Engineering 139.5 (2013): 757-765.
[12] Gao, Yuling, et al. "Understanding urban-suburban adoption and maintenance of rain barrels." Landscape and Urban Planning 153 (2016): 99-110.
[13] Oberascher, Martin, et al. "Advanced rainwater harvesting through smart rain barrels." World Environmental and Water Resources Congress 2019. Reston, VA: American Society of Civil Engineers, 2019.
[14] Villarreal, Edgar L., and Andrew Dixon. "Analysis of a rainwater collection system for domestic water supply in Ringdansen, Norrköping, Sweden." Building and Environment 40.9 (2005): 1174-1184.
[15] Kellogg, Bryce, et al. "Wi-Fi backscatter: Internet connectivity for RF-powered devices." Proceedings of the 2014 ACM Conference on SIGCOMM. 2014.