SmartPool: An Automated CPS-Based System for Real-Time Water Quality Management

André Ávila[1], André C. Costa[1], André Soares[1], Daniel Rodrigues[1], Eduardo T. Ramos[1],
João Alves[1], João C. Matos[1], Lourenço Gonçalves[1], Luís Cabral[1], Marco Rocha[1]
Rui Pinto[1,2], Gil Gonçalves[1,2]
Dept. of Informatics Engineering, Faculty of Engineering, University of Porto, Porto, Portugal[1]
SYSTEC, ARISE, Faculty of Engineering, University of Porto, Porto, Portugal[2]
Email: {up202006767, up201905916, up202004161, up202006562, up201906732, up202007614, up202006280, up202004816, up202006464, up202004891}@up.pt
{rpinto, gil}@fe.up.pt

***Abstract*—Maintaining optimal water quality in swimming pools is critical for ensuring safety, hygiene, and user comfort. Traditional methods often rely on manual and regular adjustments, which are time-consuming, prone to errors, and operationally inefficient. External factors can lead to excessive maintenance and unnecessary water waste, while undermaintenance could result in cloudy, discoloured, and dirty water. This research presents a SmartPool Internet of Things (IoT)-based solution that automates the maintenance and management of water quality using integrated sensors and actuators. The pool system measures parameters, such as pH, chloride, temperature, and water levels to preserve and control aesthetic and health parameters. The solution provides real-time data visualization and user interaction through a user-friendly dashboard and supports automated decisions. The developed prototype solution consists of a minimum-value product version of the presented architecture centred on a Raspberry Pi that incorporates camera and recognition algorithms to enhance pool safety, physical assets, a web interface, and a middleware open-source solution. This work demonstrates how Internet of Things and Cyber-Physical Systems technologies can automate pool parameter management by ensuring sustainability and improving efficient operations.**

***Keywords*-Pool System; Pool Maintenance; Automation; Control; Asset Administration Shell.**

I. Introduction

Swimming pools have become an essential amenity in modern society, serving diverse purposes ranging from recreation and exercise to hospitality and tourism. As the number of pool users increases, the demand for efficient and safe pool management systems in private and public pools has also increased, due to growing health awareness, technological advancements, and cost efficiency [1]. Poor water conditions can lead to health hazards, such as skin irritations, eye discomfort, and the spread of waterborne diseases (Angdresey et al. [2]), underscoring the critical importance of effective pool maintenance.

Traditional pool maintenance practices typically involve manual and routine adjustments to address water quality imbalances. These methods are labor-intensive, time-consuming, and susceptible to human error, making them less than optimal. Additionally, they depend heavily on field operators, which can result in significant costs—posing challenges for pool owners with limited budgets [1].

In today's smart technology era, the Internet of Things (IoT), Cyber-Physical Systems (CPS), and Digital Twins (DTs), as industrial-technological enablers, present an opportunity to transform traditional pool management into a more automated and data-driven process [3]. By integrating advanced sensors and actuators with a central control and processing unit leveraging IoT Machine-to-Machine (M2M) connectivity, the system can continuously collect and process insights from the environment, as described in Kaur, Mishra, and Maheshwari [4]. Leveraging the implementation of DTs, asset's digital counterparts, the solution presented in this study enables advanced monitoring, analysis, and control throughout the lifecycle of systems or products [5].

This paper presents the development and implementation of a five-level architecture CPS-enabled solution for the SmartPool system, aimed at addressing the challenges of traditional pool maintenance. The primary objective of this work is to enhance the efficiency, sustainability, and safety of pool management systems. By integrating automation and real-time monitoring, the SmartPool system reduces reliance on manual interventions, minimizes operational costs, and improves water quality control. However, as this study represents a prototype, it presents certain limitations related to security, performance, and hardware, further discussed in Sections V and VI.

Concerning the structure, the remainder of this document is organized as follows: Section II provides a review of the related work in industrial solutions for pool management and a discussion of the outcomes, methodologies used, and gaps for innovation. Section III describes the methods and architecture of the proposed system. Section IV presents the results of the experiments and evaluations. Section V discusses the implications of this solution, along with limitations and potential improvements. Finally, Section VI concludes the paper and highlights directions for future research.

II. Related Work

The evolution of IoT technologies has transformed swimming pool management by creating intelligent systems that go beyond traditional maintenance approaches. Modern smart pool systems integrate advanced sensors, communication technologies, and data analytics to provide comprehensive monitoring, safety, and efficiency solutions.

Recent research has demonstrated significant advances in real-time water quality monitoring. Hamid et al. [6] proposed a Smart Water Quality Monitoring System (SWQMS) that can automatically track critical parameters, such as pH levels and water temperature. The study revealed that while the time of day influences water temperature, it does not significantly impact pH levels, highlighting the importance of continuous monitoring. The work developed by Lakshmikantha et al. [7] further expanded this concept by creating a prototype that can measure multiple parameters, including temperature, pH, turbidity, water level, and water flow. Their system successfully demonstrated the ability to detect issues across different water sources, from swimming pools to industrial wastewater.

Safety remains a critical concern in pool management. The Smart Swimming Pool Management System (SSPMS) proposed by Sangeetha et al. [8] integrates ultrasonic and Passive Infrared (PIR) sensors to detect potential drowning incidents. The system can automatically activate an alarm and initiate water drainage procedures, representing a significant leap in proactive safety mechanisms. Building on this approach, Raj et al. [9] presented an enhanced system that not only detects drowning individuals but also monitors additional parameters, such as water temperature, water level, and the presence of intoxicated individuals. Their solution includes an ESP32-CAM module for image analysis and can send emergency notifications to rescue teams.

Underwater communication presents unique technological challenges. A recent study investigated IoT-LoRa technology for underwater applications, revealing that water type and turbidity significantly impact signal transmission. This research showed that swimming pool water provides the best signal characteristics, while seawater presents the most challenging environment for communication [10].

The work of Gloria, Cercas and Souto [11] demonstrated the potential of IoT gateway systems in creating smart environments. Their approach utilized a Raspberry Pi as an aggregation node and an Arduino as a sensor node, communicating through the MQTT protocol. This architecture provides a flexible framework for remote monitoring and control of pool environments. These Edge hardware can be combined with the Cloud for extra storage and computational analysis.

Swimming pool-related studies presented in the literature often shift the focus to the drowning problem and their solution into prevention and monitoring systems [9], and forget about the tons of diseases that poor water quality brings to pool users every year. The reviewed studies highlight key challenges, potential areas for improvement, and solutions aimed at enhancing automation and improving the quality of life for swimming pool users [12]. Building on these insights, this study introduces an innovative, state-of-the-art solution that leverages the implementation of Asset Administration Shells (AAS) using the open-source Eclipse BaSyx middleware [13].

III. SmartPool Solution

The proposed solution incorporates an open-source middleware platform that utilizes an Asset Administration Shell [14], a standardized representation of a DT with uniform interfaces. This approach enables seamless control of the physical twin through bidirectional communication, facilitating real-time monitoring and adjustments. The implementation ensures horizontal scalability, allowing for the easy integration of additional devices. It also offers adaptability for deployment at the Edge [15] or within Fog computing [16] levels, along with robust interoperability, supporting a wide range of communication and data protocols.

The system includes an intuitive web-based User Interface (UI) that displays over-time metrics through a dashboard, giving users the power to monitor water conditions at a glance and to manually adjust settings, such as the desired water temperature to which the system dynamically responds, recalibrating itself to meet the updated preferences, promoting high adaptability through user-friendly control.

The methodology used in the development of the SmartPool system followed the design, implementation, and evaluation of an automated system for real-time water quality management and pool safety. Figure 1 showcases the system architecture in a high-level design of the flow of information from field devices into Protocol Adapters, the Platform's core, and external applications for visualization and decision-making. The following section explains in detail the system components, data flow, and implementation details.

A. System Architecture

The architecture of the SmartPool system follows a five-level structure, ensuring a clear division of responsibilities and functionalities. The *Perception Level* includes the physical sensors and actuators responsible for data collection and environment control. Key components include water quality sensors, which measure temperature; environmental sensors, which monitor light intensity and detect water levels; a camera, which captures real-time visual data of the pool and its surroundings; LEDs, used for signaling critical parameter levels to alert users and for enhancing overall illumination; and an alarm, which provides alerts to notify users of potential drowning incidents involving animals or children.

The *Network Level* facilitates communication between the perception level and higher levels. The system uses Wi-Fi [17] to transmit data from sensors and cameras to the Raspberry Pi [18] via the MQTT protocol [19]. As this protocol uses a publish-subscribe pattern, several topics were defined, both for the sensors and the actuators, with the strings "sensor/#" and "actuator/#", where "#" can be represented by the strings "luminosity", "distance", and "temperature" for the sensors, and "rgb/1", "rgb/2", "buzz", and "buzz/silence" for the actuators. An important consideration was a fault tolerance mechanism for Wi-Fi failures. The Arduinos [20] keep trying to establish Wi-Fi connections, publish data, and collect actionable insights.

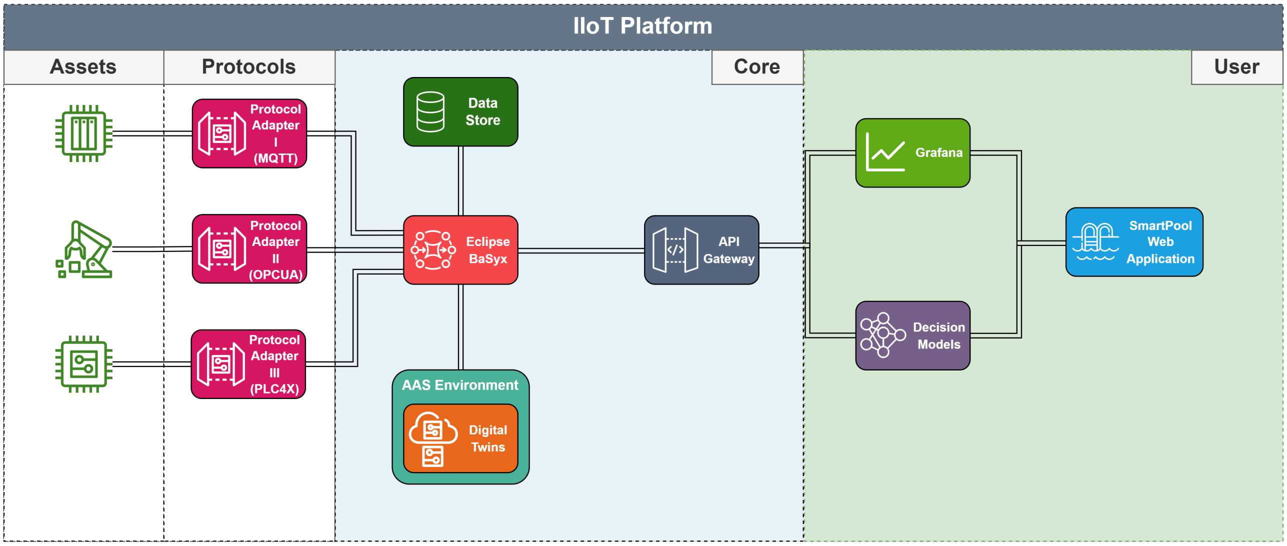

Figure 1. Architecture of the system.

Subsequently, in the *Middleware Level*, the Raspberry Pi serves as the primary central hub, handling data processing from sensors, executing decision-making algorithms, and managing communication with the AAS structure. To address potential computational power limitations, a Fog node—represented by a dedicated computer—was integrated into the architecture. The Fog node offloads some of the computational responsibilities from the Raspberry Pi, enabling the system to scale and accommodate additional central hub devices, such as those for future clients' pools. This node also hosts the AAS structure, using the open-source framework Eclipse BaSyx implementation [21]. This incorporation enables control of physical assets through their digital replicas, a standardized representation of the data, and bi-directional communication. Components of Eclipse BaSyx were deployed as Docker containers [22], ensuring efficient AAS registration and MQTT-based communication. The middleware level is responsible for storing sensor data in an InfluxDB database [23], with Telegraf [24] acting as a bridge between the MQTT broker and the database. The incorporation of Eclipse BaSyx.

Application Level is mainly composed by a web application providing visualizations of the collected data. As shown in Figure 2, it presents water and environment parameters, as well as the *live feed* of the pool and historical visuals created and displayed using Grafana.

Finally, the *Business Level*, which focuses on the broader operational and strategic goals of the system. This level addresses data analytics and notifications to optimize pool maintenance and enhance user safety and comfort by integrating advanced monitoring technologies.

B. Hardware and Software Implementation

The SmartPool system incorporates a combination of hardware and software components to ensure robust functionality across all levels of the architecture. The hardware setup includes a Raspberry Pi, which acts as the primary processing and communication node, equipped with the Raspberry Pi Camera Module 2 for visual monitoring. Arduinos are utilized as intermediaries for data collection from sensors, transmitting measurements to the Raspberry Pi. Sensors and actuators are integrated to measure water quality parameters and execute corrective actions. The sensors are connected to the Arduinos, which preprocess and relay the data to the Raspberry Pi. The Raspberry Pi then processes this data and interfaces with actuators for implementing corrective measures.

The software system was developed using a combination of tools and frameworks to ensure seamless operation. An MQTT Broker was deployed on the Raspberry Pi to facilitate message processing between edge-level functionality and the AAS, enabling efficient real-time data exchange. For object detection, YOLOv8 integrated with OpenCV [25] is used to identify entities near the pool captured in the camera feed. Detection results trigger alerts and are communicated to the dashboard via the middleware. The AAS functionality is implemented using the Docker-based version of Eclipse BaSyx, which includes essential components, such as the AAS Server for hosting the digital twin representation of the pool's assets, the AAS Registry for managing and organizing multiple AAS instances, and the Databridge Component for communication with external protocols like MQTT to ensure data flow between the physical and digital twins.

A user-friendly web interface, represented in Figure 2, was developed using React [26] with Vite [27]. This interface serves as the SmartPool dashboard, providing real-time monitoring, alert notifications, and control capabilities to ensure an intuitive user experience. The interface design prioritizes a balance between usability and system capacity, as performing essential actions is equally important to overall system usability in the final product.

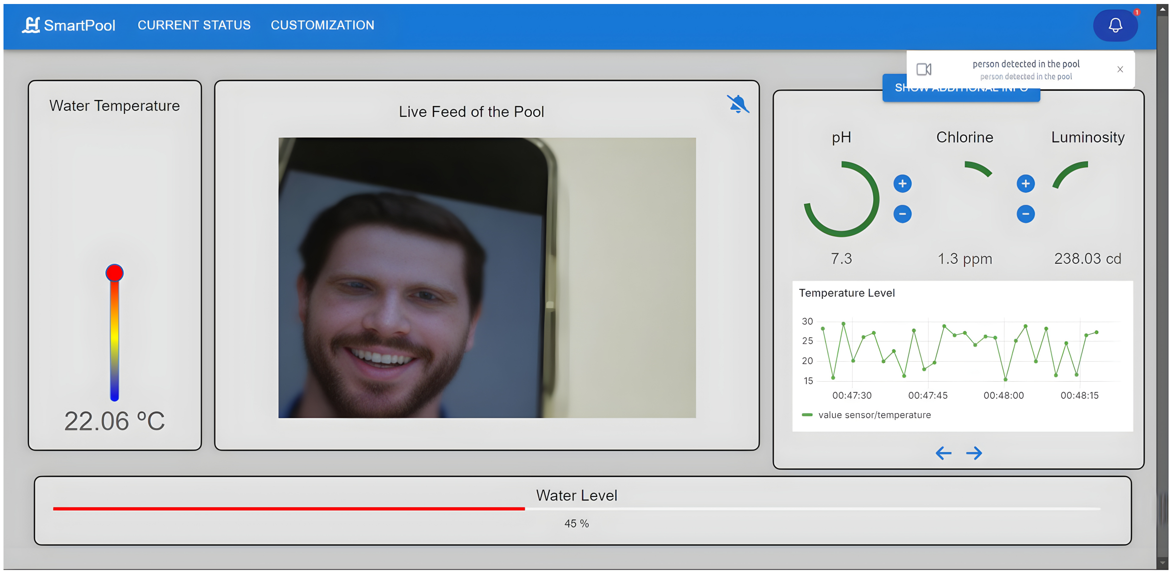

Figure 2. SmartPool UI.

IV. Validation and Result Analysis

To comprehend the tangible impact and benefits of the SmartPool System, it is important to analyze the results in the different fields, namely: Continuous Visual Monitoring Performance; System Responsiveness; and Parameters Monitoring.

A. Continuous Visual Monitoring

The system integrates intelligent recognition algorithms and real-time alert mechanisms to enhance pool safety by reducing the risk of drowning accidents. This is accomplished by a live camera feed that continuously monitors the pool and its immediate surroundings, notifying the user in the web application when any recognized entities of interest are detected (e.g., a dog near the pool). Users receive rapid alerts via the interface and can also view the live feed, ensuring they are always informed and able to respond when necessary.

A *Raspberry Pi* was used to obtain the live feed through the *Pi camera module 2*. This camera uses a Sony IMX219 8-megapixel sensor and can obtain somewhat high-definition images and video. However, due to the high strain placed on the Raspberry Pi by the various programs being run on it, the video quality and frame rate cap out at very low values. One such program is intelligent recognition, which leverages the Python library cv2 and the YoloV8 model to get the entity recognition algorithms used in the live feed. Since video capture is handled by the *Raspberry Pi*, the computationally costly frame processing, powered by the YoloV8 model, is only performed at six-second intervals. This study briefly considered offloading the object detection algorithm to a more powerful device, fully utilizing a distributed IoT architecture, but later refrained from doing so because object detection should be independent of device connectivity and achieve low latency for critical and urgent situations, such as when a child or a pet falls into the water. The final implemented setup optimizes performance and energy consumption by reducing the image processing frequency while still ensuring timely detection of any noteworthy events.

This study utilizes the Raspberry Pi as the central processing unit within the pool environment. Video frames captured by the Pi camera are transmitted as file properties of an AAS via POST requests. These files can then be retrieved and processed at the application level, enabling the delivery of high-quality images to the front end and providing a live feed of the pool.

The real-time visual monitoring system demonstrated strong capabilities in detecting and identifying entities near the pool, such as animals or people. Using the YOLOv8 model integrated with the Raspberry Pi, in a testing phase, the system, after performing the entity recognition process, was able to send alerts with less than a second delay every time with light or partial obstruction having little to no effect on the result.

To ensure near-certainty in the entity recognition, the confidence threshold defined was 0.7, meaning that the user would only receive an entity recognition notification if the model was 70% certain of its decision. Although the camera quality was high, with video live streams maintaining a stable resolution of 1080p, due to computational strain on the Raspberry Pi, the entity recognition was only performed once every 6 seconds.

B. System Responsiveness

During testing, the web-based User Interface (UI), represented in Figure 2, consistently displayed updated pool conditions with delays of less than 5 seconds. System notifications, including alert information and corresponding alarm sounds, were delivered within 1 second of completing the Event Response Process, even under both normal and high-stress environments. The dashboard demonstrated seamless responsiveness on both mobile and desktop platforms, maintaining its functionality and design integrity across devices.

User-triggered actions, such as toggling lights or muting/unmuting alarms, were executed within 2 to 3 seconds

 ISBN: 978-1-68558-236-4

under normal conditions. In hardware-stressed scenarios, delays increased slightly, ranging between 6 and 7 seconds.

C. Parameters Monitoring

The system, represented in Figure 3, effectively monitored critical water quality parameters, including temperature, light intensity, and water level, with updated values consistently displayed on the interface. Readings were recorded with minimal delays and achieved near-real-time accuracy, providing reliable data on pool conditions.

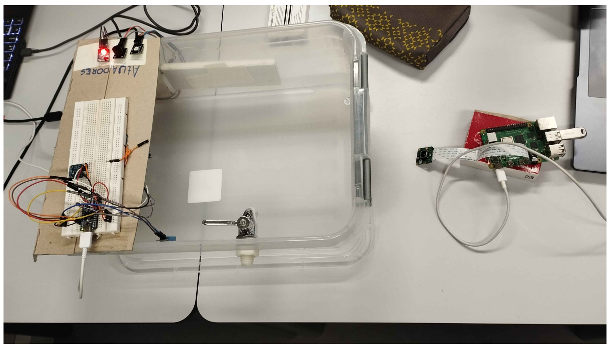

Figure 3. SmartPool Prototype.

The water level monitoring, while generally satisfactory, experienced occasional inaccuracies due to the transparent nature of water affecting the distance sensor. However, these discrepancies were minimized after stabilization.

All actuators within the system performed as expected during testing, ensuring the successful execution of automated responses and user-triggered actions.

V. DISCUSSION

Having successfully achieved its primary objectives as a prototype, the SmartPool system shows significant promise in automating pool maintenance and improving safety. Real-time visual monitoring with YOLOv8 delivered accurate alerts despite the hardware limitations caused by Raspberry Pi's processing constraints, which led to some slightly longer response times. Improved hardware or more efficient detection models specific to such devices could address this issue. The system's sustainability can also be improved by integrating energy-efficient components or renewable energies. In outdoor pool areas, solar power could be a viable eco-friendly energy source. However, an increase in energy usage may be the result of a compromise to increase the system responsiveness and the safety of the users of the pool.

Water quality monitoring for temperature and light intensity was effective, although the water level sensor showed occasional inaccuracies. Sensor calibration or even alternative technologies could help tackle this problem.

Another feature lacking in this implementation is security. MQTT protocol and POST requests to transmit information between the various physical assets of the pool to the middleware. These packets, however, are not encrypted and do not have any integrity checks. An attacker can not only see the private information in the packets, including sensor information and the images captured by the camera, but they can also change the packets at will. The attacker can change parameters meant for the actuators during transmission, which can cause unwanted behaviors. Another problem is that the nodes used in the system are not authenticated, and that leaves room for spoofing attacks, where an attacker can trick the nodes and middleware into sending the data to them instead of the actual nodes and components. Even though these issues present a big concern to a normal system, this study does not address them, as this implementation serves as a prototype not meant to be fully deployable, as is, in the real world.

While user feedback was positive regarding the dashboard's intuitive and functional design, some improvements have been suggested. The interface is simple, and while it works adequately in its current state, enhancements to customization could be made to better support more complex systems.

VI. CONCLUSION & FUTURE WORK

The SmartPool system demonstrates how sensors, actuators, and data-driven strategies can enhance pool maintenance, safety, and user experience. By automating the monitoring and adjustment of critical water quality parameters, the system reduces manual intervention while improving safety, operational efficiency, and sustainability. Real-time visual surveillance, powered by advanced recognition algorithms, further enhances safety by detecting and responding to potential risks, such as a child falling into the pool.

Even though the prototype shows that automated pool administration is feasible, there are still a number of areas that might be improved:

- Advanced Machine Learning: Incorporating more sophisticated machine learning models could improve decision-making and anomaly detection. Enhanced hardware performance would support faster processing and more frequent data transmission, reducing the current three-second interval and enabling more comprehensive real-time monitoring.
- Improved Security: Strengthening system security is a critical priority. Future implementations will include a Certificate Authority to issue certificates for Fog nodes and middleware components, enabling robust identification and authentication. By employing Transport Layer Security (TLS) on top of the MQTT protocol, all transmissions will be encrypted and safeguarded against tampering. These measures aim to resolve the security vulnerabilities present in the current system.
- Sustainability Enhancements: The system's environmental footprint can be minimized by adopting energy-efficient hardware and integrating renewable energy sources, such as solar panels, to power its components.
- Scalability: Expanding the platform's scalability will enable it to support larger facilities or more complex aquatic environments, broadening its applicability.

This research highlights the transformative potential of IoT and CPS in pool management, showcasing innovations that promote convenience, safety, and environmental responsibility. By addressing the identified limitations—such as enhancing machine learning capabilities, improving security, and ensuring sustainability—future developments can cater to the diverse needs of pool operators and users, paving the way for smarter, more sustainable aquatic solutions.

Acknowledgment

This work is financially supported by national funds through the FCT/MCTES (PIDDAC), under the Associate Laboratory Advanced Production and Intelligent Systems – ARISE LA/P/0112/2020 (DOI 10.54499/LA/P/0112/2020) and the Base Funding (UIDB/00147/2020) and Programmatic Funding (UIDP/00147/2020) of the R&D Unit Center for Systems and Technologies – SYSTEC.

References

[1] A. Jemat, S. Yussof, S. S. Sameon, and N. A. Alya Rosnizam, "IoT-Based System for Real-Time Swimming Pool Water Quality Monitoring," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13051 LNCS, pp. 332–341, 2021, Cited by: 3. DOI: 10.1007/978-3-030-90235-3_29.

[2] A. Angdresey, L. Sitanayah, and V. J. A. Sampul, "Monitoring and Predicting Water Quality in Swimming Pools," *EPI International Journal of Engineering*, vol. 8, no. 2, pp. 119–125, 2020, Accessed from https://doi.org/10.25042/epi-ije.082020.05. DOI: 10.25042/epi-ije.082020.05.

[3] S. Zeb *et al.*, "Industrial digital twins at the nexus of NextG wireless networks and computational intelligence: A survey," *Journal of Network and Computer Applications*, vol. 200, p. 103 309, 2022, ISSN: 1084-8045. DOI: https://doi.org/10.1016/j.jnca.2021.103309.

[4] M. Kaur, V. Mishra, and P. Maheshwari, "The Convergence of Digital Twin, IoT, and Machine Learning: Transforming Data into Action," in *Internet of Things*, Springer, Cham, 2020, pp. 3–17. DOI: 10.1007/978-3-030-18732-3_1.

[5] A. Redelinghuys, A. Basson, and K. Kruger, "A six-layer architecture for the digital twin: a manufacturing case study implementation," *Journal of Intelligent Manufacturing*, vol. 31, no. 6, pp. 1383–1402, 2020. DOI: 10.1007/s10845-019-01516-6.

[6] S. A. Hamid *et al.*, "IoT based Water Quality Monitoring System and Evaluation," in *2020 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, 2020, pp. 102–106. DOI: 10.1109/ICCSCE50387.2020.9204931.

[7] V. Lakshmikantha *et al.*, "IoT based smart water quality monitoring system," *Global Transitions Proceedings*, vol. 2, no. 2, pp. 181–186, 2021, International Conference on Computing System and its Applications (ICCSA- 2021), ISSN: 2666-285X. DOI: https://doi.org/10.1016/j.gltp.2021.08.062.

[8] A. Sangeetha *et al.*, "Smart Swimming Pool Management System (SSPMS) using IoT," in *2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA)*, 2023, pp. 840–846. DOI: 10.1109/ICIDCA56705.2023.10099729.

[9] K. J. S. Raj *et al.*, "Enhancing Pool Safety and Efficiency with an IoT Supported Monitoring System," in *2023 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN)*, 2023, pp. 1232–1237. DOI: 10.1109/ICPCSN58827.2023.00208.

[10] J. P. Christopher, S. D. Damayanti, and M. Suryanegara, "Investigating IoT-LoRa Technology for The Underwater System Application," in *2023 IEEE 8th International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, 2023, pp. 1–4. DOI: 10.1109/ICRAIE59459.2023.10468069.

[11] A. Glória, F. Cercas, and N. Souto, "Design and implementation of an IoT gateway to create smart environments," *Procedia Computer Science*, vol. 109, pp. 568–575, 2017, 8th International Conference on Ambient Systems, Networks and Technologies, ANT-2017 and the 7th International Conference on Sustainable Energy Information Technology, SEIT 2017, 16-19 May 2017, Madeira, Portugal, ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2017.05.343.

[12] M. Elgorma *et al.*, "A review of methods for detecting and preventing drowning incorporating various techniques, devices and technologies," in *2nd International Conference on Electrical Engineering and Automatic Control*, 2024, pp. 1–6.

[13] S. Karthik, D. Priya E.L., G. Anand K.R., and A. Sharmila, "IoT Based Safety Enhanced Swimming Pool with Embedded Techniques to reduce drowning accidents," in *2020 International Conference on Smart Electronics and Communication (ICOSEC)*, 2020, pp. 843–847. DOI: 10.1109/ICOSEC49089.2020.9215247.

[14] N. Chilwant and M. S. Kulkarni, "Open Asset Administration Shell for Industrial Systems," *Manufacturing Letters*, vol. 20, pp. 15–21, 2019, ISSN: 2213-8463. DOI: https://doi.org/10.1016/j.mfglet.2019.02.002.

[15] B. Chen *et al.*, "Edge Computing in IoT-Based Manufacturing," *IEEE Communications Magazine*, vol. 56, pp. 103–109, 2018. DOI: 10.1109/MCOM.2018.1701231.

[16] F. Haouari, R. Faraj, and J. Jaam, "Fog Computing Potentials, Applications, and Challenges," *2018 International Conference on Computer and Applications (ICCA)*, pp. 399–406, 2018. DOI: 10.1109/COMAPP.2018.8460182.

[17] S. S. Salwe and K. K. Naik, "Heterogeneous Wireless Network for IoT Applications," *IETE Technical Review*, vol. 36, pp. 61–68, 2019. DOI: 10.1080/02564602.2017.1400412.

[18] E. Upton and G. Halfacree, *Raspberry Pi User Guide*, 3rd. Indianapolis, IN, USA: Wiley, 2014, ISBN: 978-1118921661.

[19] A. Banks and R. Gupta, "MQTT Version 3.1.1," 2014, [Online]. Available: https://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html (visited on 02/11/2025).

[20] M. Banzi and M. Shiloh, *Getting Started with Arduino*, 3rd. Sebastopol, CA, USA: Maker Media, Inc., 2014, ISBN: 978-1449363338.

[21] E. Foundation, "Eclipse BaSyx Documentation," 2023, [Online]. Available: https://www.eclipse.org/basyx/ (visited on 02/11/2025).

[22] I. Docker, "Docker Documentation," 2023, [Online]. Available: https://docs.docker.com/ (visited on 02/11/2025).

[23] InfluxData, "InfluxDB Documentation," 2025, [Online]. Available: https://docs.influxdata.com/influxdb/ (visited on 02/11/2025).

[24] InfluxData, "Telegraf Documentation," 2025, [Online]. Available: https://docs.influxdata.com/telegraf/ (visited on 02/11/2025).

[25] OpenCV Team, "OpenCV: Open Source Computer Vision Library," 2025, [Online]. Available: https://opencv.org/ (visited on 02/11/2025).

[26] Meta Platforms, Inc., "React: A JavaScript Library for Building User Interfaces," 2025, [Online]. Available: https://reactjs.org/ (visited on 02/11/2025).

[27] Evan You and Vite Contributors, "Vite: Next Generation Frontend Tooling," 2025, [Online]. Available: https://vitejs.dev/ (visited on 02/11/2025).