KEMENTERIAN PENGAJIAN TINGGI

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

DIPLOMA IN INFORMATION TECHNOLOGY

(DIGITAL TECHNOLOGY)

DFT50114: INTEGRATED PROJECT

HYDRO CONTROLLER

GROUP MEMBERS:

MUHAMMAD HADI BIN SUPARMAN	17DDT21F1037
ANIQ SHAQIF BIN MOHD ZEN	17DDT21F1064
MUHAMMAD AMIR FARHAN BIN SHAROL AZLAN	17DDT21F1077

SUPERVISOR:

PUAN NURUL AIN BT AMIR ADLI

SESSION: SHORT SEMESTER 2023

Declaration

We, the members of our project group, hereby declare that the project titled "Hydro Controller", represents our group's original work created under the direction and supervision of Madam Nurul Ain Bin Amir Adli. All of the material and data in this report are the result of our collaborative study and analysis. External sources have been properly acknowledged and credited.

We confirm that this report has not been submitted for academic credit or review elsewhere.

1.Signature : 

Name : MUHAMMAD HADI BIN SUPARMAN

Registration No : 17DDT21F1037

Date : 4/8/2023

2.Signature : 

Name : ANIQ SHAQIF BIN MOHD ZEN

Registration No: 17DDT21F1064

Date: 4/8/2023

3.Signature : 

Name : MUHAMMAD AMIR FARHAN BIN SHAROL AZLAN

Registration No: 17DDT21F1077

Date: 4/8/2023

Table of Content

1.0 Abstract

Projek ini bertujuan membangunkan sistem pintar pengesanan tekanan air dan kawalan pam automatik untuk hartanah komersial. Sistem ini memantau tekanan air secara masa nyata menggunakan pengesan aras air dan mengawal pam air berdasarkan aras yang ditetapkan. Integrasi notifikasi Telegram memberi makluman segera kepada pengguna, memastikan pengurusan air yang cekap dan bekalan air yang boleh diandalkan. Pam air, pengesan aras air dan pegesan tekanan air berfungsi akan dibangunkan menggunakan Model Prototaip, melibatkan kitaran berulang prototaip perisian, ujian, dan penambahbaikan. Sistem ini dilengkapi dengan mekanisme kawalan pam automatik yang mengaktifkan pam apabila aras air jatuh di bawah ambang had yang ditetapkan dan mematikannya untuk mencegah limpahan air dan menjimatkan tenaga. Dengan menggabungkan pemantauan tekanan air secara masa nyata dan kawalan pam automatik, sistem ini mengoptimumkan pengurusan air di hartanah komersial, meminimumkan kawalan secara manual dari pengguna. Notifikasi Telegram memberi notifikasi atau makluman status kepada pengurus hartanah untuk menangani isu berkaitan air dengan pantas. Sistem ini menyokong penggunaan air yang mampan, menjimatkan sumber dan mengamalkan amalan alam sekitar yang bertanggungjawab dalam persekitaran komersial.

This project aims to develop a smart water pressure detection and automatic pump control system for commercial properties. The system monitors water pressure in real-time using a water level sensor and controls the water pump based on the pre-set level. Telegram notifications provide instant alerts to property managers, ensuring efficient water management and a reliable water supply. The workable prototype will be developed using the Prototype Model, involving iterative cycles of software prototyping, testing, and improvement. The system includes an automated water pump control mechanism, activating the pump when the water level falls below the predefined threshold and deactivating it to prevent overflows and conserve energy. By integrating real-time monitoring and automatic pump control, the system optimises water management in commercial properties, minimising wastage and manual interventions. Telegram notifications empower property managers to address water-related issues promptly. The system promotes sustainable water utilisation, conserving resources and adopting environmentally responsible practices in commercial settings.

2.0 Project Plan

a. Introduction

Projects involving water pressure detection and automated water pump control system specifically suited for commercial properties is the main goal of this project. The major goal is to develop a dependable and effective system that continually tracks water pressure levels and automatically turns on or off the water pump in response to a predetermined set level that is received up by a water level sensor. The device will also include a Telegram app notification feature to instantly alert customers about any changes in water pressure or pump status. The execution of this project is especially appropriate for commercial buildings since it seeks to improve water management, avoid water waste, and provide a reliable and sustainable water supply.

In commercial properties, a consistent and reliable water supply is essential for various operations, including building maintenance, occupant usage, and sanitation. The water pressure detection system will play a pivotal role in real-time monitoring, providing property managers with accurate data on water pressure levels. By having access to this information, timely actions can be taken to address any irregularities, such as pressure fluctuations or leaks, contributing to efficient water management and minimising potential disruptions in daily activities.

The automated water pump control system is the second component of this project. The system will detect the water level and turn on the water pump when it drops below the specified threshold by incorporating a water level sensor. As a result, as soon as the water reaches the desired level, the system will quickly turn off the pump, reducing the chance of water spills and using less energy. This automatic feature minimises the need for manual intervention, simplifies water supply maintenance, and makes sure that commercial premises have a dependable water distribution system. Furthermore, the project will utilise the Telegram app to deliver real-time notifications to designated users. Whenever the water pressure deviates from the set level or when the pump's status changes, the system will generate instant alerts.

Finally, the goal of this project is to develop a system for commercial properties that detects water pressure and automatically controls the water pump. A responsible and sustainable approach to water use in business settings will be made possible by the integration of real-time monitoring, automatic pump control, and quick notifications via Telegram.

b. Problem Statements

It is difficult to regulate water pressure effectively and operate pumps at their best when they are manually controlled. These difficulties include: -

I. A problem to effective water management is the lack of sight about water pressure levels at commercial properties. The absence of real-time data on water pressure can lead to wastage, higher utility bills, and even costly repairs due to undetected issues.

II. **The user is unable to obtain current water pressure and water level sensor information:** An issue for users is the absence of real-time data on water pressure levels. Users are unable to monitor and resolve problems connected to water pressure without access to this information.

III. **Inefficient water usage:** commercial properties frequently ignore leaks, high water pressure, or inefficient consumption habits when they are not properly monitored and controlled. This raises water costs, has an impact on the environment, and puts pressure on water supplies.

c. Objective

The main objective of this project is to develop a water pressure detector and an automatic pump system. The specific objectives are as follows: -

I. **Monitor water pressure :** To monitor the current water pressure levels by establishing a water pressure monitoring system

II. **Send the user notifications:** To send notifications to users of any issues in water pressure using Telegram applications.

III. **Automatic on/off water pump :** To shut off the water supply automatically if the water level reaches the preset value.

d. Project Scope

User Scope:

I. **Monitor Water Level:**

Users can monitor the water level in the water tank in real-time through the system's interface. This allows them to stay informed about the available water quantity and plan water usage accordingly.

II. **Check Water Pressure:**

Users can check the water pressure in the water supply line using the system's interface. Monitoring water pressure helps them ensure an adequate and consistent flow of water.

III. Receive Notifications:

Users can receive real-time notifications via Telegram about critical events related to the water supply system. This includes notifications about low water levels, low water pressure, and status of pumps.

System Scope:

The system scope defines the boundaries and functionality of the automatic water pump and water level sensor system. It outlines what the system is capable of doing and the key features it offers. Here's the system scope for your project:

1. **Water Level Monitoring:**
    - The system will continuously monitor the water level in the tank or reservoir using the water level sensor.
    - It will measure the water level accurately and provide real-time data to the control unit.
2. **Automated Pump Control:**
    - Based on the water level readings, the system will automatically control the water pump's operation.
    - When the water level is low or falls below a predefined threshold, the pump will automatically turn on to fill the tank.
3. **Water Pump Shutdown:**
    - The system will stop the water pump automatically when the tank reaches a certain predefined water level (full capacity).
    - This prevents overfilling and unnecessary water wastage.
4. **Notification System:**
    - The system will be equipped with a notification feature to alert the user about the water level status.
    - Notifications will be sent via a chosen communication channel, such as WhatsApp or other messaging services.
    - Users will receive alerts when the tank is full or when the water level is low, along with relevant information.

e. Literature Reviews

After conducting thorough research, we have explored various pump models available in the market.

Table 1: Literature Reviews: Automated Water Pump Control Products

No.	Pump Model	Flow Rate (L/min)	Power Consumption (W)	Connectivity	Control Options	Additional Features
1.	Grundfos MQ3-45	40	900	Wi-Fi	Mobile App, Timer	Self-priming, Dry-run protection
2.	Ebara PRA Booster	50	800	Bluetooth	Mobile App, Pressure Switch	Energy-efficient, Thermal protection
3.	DAB E.SYBOX	60	1000	Wi-Fi	Mobile App, Pressure Sensor	Variable Speed, Noise reduction

4.	Pentax JET 100M	45	1200	Wi-Fi	Mobile App, Pressure Switch	Stainless steel construction
5.	Pedrollo PKm 60	50	950	Bluetooth	Mobile App, Timer	Compact design, Easy installation

We have identified certain weaknesses associated with each of the mentioned pump products.Here the following comparison of disadvantages:

Table 2: Literature Reviews: Disadvantages Automated Water Pump Control Products

No	Name	Disadvantages
1.	**Grundfos MQ3-45**  **Diagram 1.0**	I. Relatively high-power consumption of 900 W. II. Limited connectivity option with only Wi-Fi available. III. May not be suitable for applications requiring higher flow rates.
2.	**Ebara PRA Booster**  **Diagram 1.1**	I. Limited connectivity option with only Bluetooth available. II. May not provide as high flow rate as some other models on the list. III. App availability and compatibility may vary.

3.	**DAB E.SYBOX**  **Diagram 1.2**	I. Higher power consumption compared to some other models at 1000 W. II. May have a higher initial cost due to advanced features. III. Wi-Fi connectivity may require a stable and reliable internet connection.
4.	**Pentax JET 100M**  **Diagram 1.3**	I. Relatively higher power consumption of 1200 W. II. Limited additional features compared to some other models. III. May not provide as high flow rate as the DAB E.SYBOX.

5.	**Pedrollo PKm 60**  **Diagram 1.4**	I. Relatively higher power consumption of 1200 W. II. Limited additional features compared to some other models. III. May not provide as high flow rate as the DAB E.SYBOX.

So based on our Summary Review and analysis, we have elaborated and suggested that our product will contain this feature:

I. **Enhanced Connectivity**: Opt for a pump with Wi-Fi or Bluetooth for advanced connectivity. Because of this, IoT systems can be seamlessly integrated, enabling remote monitoring and control via mobile apps or other devices.

II. Provide users with a **range of control options** so they can customise the system to their preferences. Include functions that enable users to change settings in accordance with their individual needs, such as timers, pressure switches, and pressure sensors.

III. **Features to Increase Safety and Protection**: Add new features to improve safety and protection. Examples include mechanisms for noise reduction, thermal protection, dry-run protection, and self-priming capability. These characteristics guarantee the safe and effective operation of the pump, increasing product longevity.

IV. Prioritise **energy-efficient** pumps in order to minimise power consumption and lower operating expenses. Choose pumps with variable speed options so they can change their speed to meet demand and use less energy.

V. Consider pumps with a sturdy construction, such as those made of stainless steel, to ensure **longevity and corrosion resistance**. A strong design helps ensure dependable performance and lowers the frequency of maintenance or replacement.

VI. **User-Friendly Design**: Place a focus on an intuitive user interface and a simple installation procedure. Compact design, simple mobile app controls, and detailed installation instructions should all be present on the pump. This improves user satisfaction and comfort.

f. Methodology

A workable prototype should be created at an early stage of the development process, according to the Prototype Model, a technique for developing software. It entails repeated cycles of software prototyping, testing, and improvement to satisfy the required criteria. The Prototype Model technique is described here in general terms:

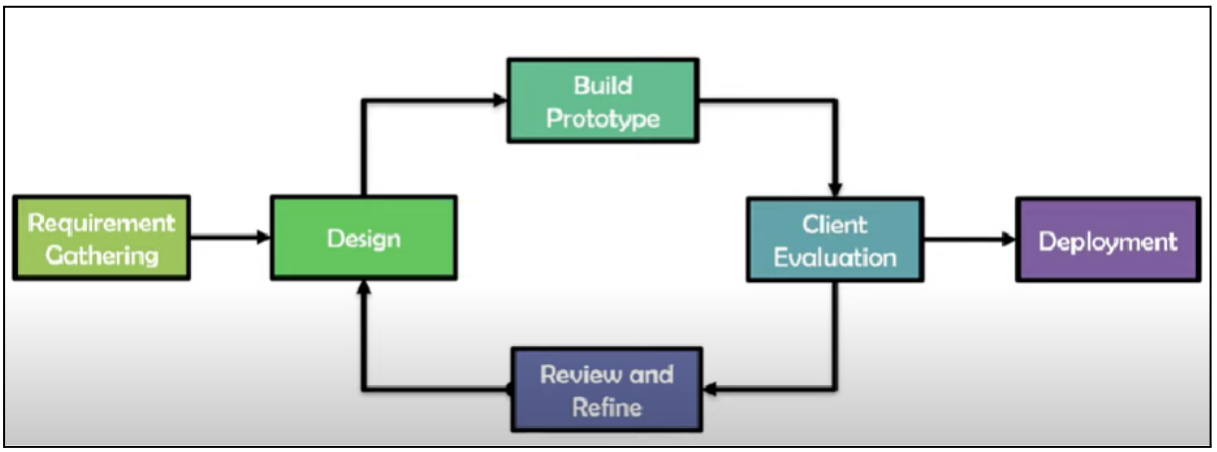

Diagram 2.0: Prototype Model

https://www.geeksforgeeks.org/advantages-and-disadvantages-of-prototype-model/

The project involves a systematic eight-phase approach, this is the following phase of prototype model and description for project development:-

Table 2: The Prototype Model for Project Development.

No	Phase	Description
1.	**Requirements Gathering**	During the requirements gathering phase, we have interacted with stakeholders, who are mechanical engineers, to collect precise requirements and expectations for the water pressure detection and automatic water pump control system to be implemented in commercial properties. This involves conducting interviews via Whatsapp between client and our team to understand the users' needs, preferred functionalities, and any limitations they might have.
2.	**Prototype Design**	Once all the necessary requirements have been gathered, we move to the prototype design phase. We carefully plan and create the system layout, taking into account the selection of appropriate hardware components such as water level sensors, ESP8266, relays and other necessary sensors or actuators. Additionally, we have defined software functions, user interfaces and communication protocols essential to enable real-time water pressure monitoring and automatic pump control. This phase ensures that the prototype is well designed to meet the specific needs and objectives of a smart water pressure management system for commercial properties.
3.	**Prototype Development**	During the prototype development phase, all team members began building early versions of the water pressure detection and automatic water pump control system. We integrate carefully selected hardware components and begin writing the initial software code necessary to enable the system's core functionality. The main focus of the team is to create the basic work of the system to demonstrate its feasibility and functionality. This phase allows the supervisor, team and client to test and verify the basic features of the system and ensure that it is aligned with the intended design and project requirements.
4.	**Prototype Testing**	The proposed water pressure detection and automated water pump control system is thoroughly evaluated throughout the prototype testing stage to determine its effectiveness, reliability, and dependability. To mimic various water pressure levels and pump control operations, the project team runs a variety of test scenarios. This thorough testing procedure seeks to find and highlight any potential problems, faults, or functional flaws in the prototype. Before moving on to the project's subsequent phases, the system must fulfil the intended performance requirements as determined by the input received during testing.

5.	**Prototype Refinement**	After analyzing the feedback and test results obtained from the previous phase, we continued by improving our prototype. A group of us have dealt with issues that occurred or identified problems for the improvement process to ensure that the system can function optimally and well. Through an iterative process of adjustment the prototype evolves into a better and more functional version. This continuous filtration cycle allows us to examine the system, ensuring it aligns with the needs and objectives of a smart water pressure management system for the desired commercial property.
6.	**Iterative Development**	In this iterative development phase of the water pressure detection and automatic water pump control system project, we continue to improve the prototype system based on feedback from stakeholders and test results we have obtained. With each iteration, we make improvements to the functionality and system. This is to ensure that it is in line with the desired needs and objectives. As the project progressed, we found the system to be more refined and efficient, leading to a reliable and effective solution for managing water pressure and pump control in commercial properties.
7.	**Full-Scale Development**	We have started the full-scale development phase after carefully evaluating and perfecting the prototype water pressure sensing and autonomous water pump control system. The system is improved during this phase to incorporate all the features and user interfaces necessary to satisfy the unique requirements of commercial real estate. To offer a complete and user-friendly solution, additional crucial elements, including the Telegram notification system, are incorporated. For effective water management in commercial buildings, the full-scale development intends to establish a fully functioning system that can manage water pressure and pump control in real-time while giving consumers useful information and alerts via Telegram.
8.	**Testing and Deployment**	After completing the development of the water pressure detection and automatic water pump control system, we have thoroughly tested the system to ensure its stability, efficiency and responsiveness. We conduct user acceptance tests to verify that the system meets specific requirements and works smoothly when used in real life and situations. Once the testing phase proves successful, the system is ready for use in commercial properties. During deployment, the system is continuously monitored and maintained to ensure long-term effectiveness, enabling efficient water management and timely notifications via Telegram for property owners or managers.

g. Gantt Chart

A Gantt chart will be created to visualise the project timeline, tasks, and milestones. This chart will aid in planning and monitoring the progress of the project, ensuring timely completion of deliverables.

Diagram 3.0 : Gantt Chart for Project Development

3.0 Requirement Specification

a. Functional Requirement

I. **Real-time Water Pressure Monitoring:** The system must continuously monitor the water pressure using the water level sensor to obtain real-time data

II. **Automatic Pump Control:** The system should automatically control the water pump based on the pre-set water level threshold. When the water level falls below the threshold, the pump should activate, and when it reaches the set level, the pump should deactivate.

III. **Telegram Notifications:** The system needs to integrate with Telegram to send instant notifications to property managers when the water level changes or the pump status alters.

IV. **Reliable and Accurate Measurements:** The water pressure detection and pump control mechanisms should be accurate and reliable to ensure proper water management.

V. **Energy Efficiency:** The system should be designed to minimise energy consumption, particularly by turning off the pump when it is not required.

VI. **User Authorization:** Only authorized users, such as property managers, should have access to the Telegram notifications and configuration settings.

VII. **Responsive User Interface:** If applicable, the system's user interface should be easy to navigate and responsive.

VIII. **Error Handling:** The system should be equipped with error-handling mechanisms to detect and report any issues in real-time.

IX. **Compatibility:** The system should be compatible with the selected hardware components, including the water level sensor, ESP8266, and relay.

X. **Scalability:** The system should be designed with scalability in mind to accommodate future expansion or integration with additional features.

XI. **Robust Connectivity:** The system should ensure a stable and reliable connection to the Wi-Fi network and Telegram servers for seamless operation.

XII. **Data Logging:** The system may include a data logging feature to record water pressure and pump status over time for analysis and troubleshooting.

XIII. **Low Maintenance:** The system should require minimal maintenance to ensure continuous and long-term operation.

XIV. **Compliance with Regulations:** The system must comply with relevant regulations and standards for water management and automation.

XV. **Easy Installation:** The system should be designed for easy installation and setup in commercial properties**.**

XVI. **Cost-effectiveness:** The system's components and overall design should be cost-effective and affordable.

XVII. **Compatibility with Telegram API:** The system should integrate seamlessly with the Telegram API to facilitate message delivery and user interaction.

XVIII. **Real-time Monitoring Alerts:** Telegram notifications should be sent in real-time to provide users with immediate updates on water pressure and pump status changes.

b. Non-Functional Requirement

I. **Performance:** The system should have real-time monitoring with a response time of less than one second for water pressure readings and pump control.

II. **Reliability:** The system should operate with a 98% uptime, ensuring continuous and reliable water pressure monitoring and pump control.

III. **Availability:** The system should be available 98% of the time, allowing property managers to access real-time water pressure data and pump status.

IV. **Maintainability:** If maintenance is required for the system or any components, it should be planned during off-peak hours to minimize disruption. Maintenance periods should not exceed three hours.

V. **Recoverability:** In the event of any unplanned system downtime, the system should recover and be fully operational within one working day.

VI. **Capacity:** The system should have the capacity to handle up to 500 simultaneous requests for water pressure data and pump control. It should also be able to store up to 1,000,000 data records.

VII. **Serviceability:** The system's automated emails for notifications and alerts should be easily editable and replaceable by uploading an XML file, eliminating the need for code recompilation.

VIII. **Security:** Access to sensitive data, such as verified phone numbers of property managers, should be restricted to users with the role of "site admin" to maintain data confidentiality.

IX. **Manageability:** Code changes for applicants' profile pages should not affect the overall system's performance, allowing seamless updates and edits.

X. **Environmental:** The system's operation for water pressure detection and pump control should be available during regular business hours (Monday through Friday from 9 AM to 6 PM).

XI. **Data Integrity:** The system should ensure data integrity by regularly backing up all updates to the database for every water pressure reading and pump control action.

XII. **Interoperability:** The system should follow a service-oriented architecture, allowing easy integration with other systems and services if required.

XIII. **Usability:** The system's user interface should be intuitive, user-friendly, and easy to navigate for property managers and other users.

c. Hardware and Software Requirement

I. Hardware Specification

The table of Hardware Specifications provides a concise summary of the essential details and functionalities of the hardware components used in the water pressure detection and automatic water pump control system for commercial properties.

Table 1.0 : Hardware Specifications

No.	Hardware	Description
1.	ESP8266 NodeMCU	The ESP8266 NodeMCU is a versatile microcontroller board with built-in Wi-Fi capabilities, making it an excellent choice for IoT projects like the smart water pressure detection and automatic pump control system. It will be the brain of the system, responsible for processing data from the water level sensor and controlling the water pump and other components.
2.	Donut Board (Breadboard)	The donut board, also known as a breadboard, is a prototyping tool used to create temporary electrical connections between various electronic components. It provides a convenient platform for testing and assembling the system before soldering the components onto a permanent PCB.
3.	Jumper Connector (20 pieces)	Jumper connectors, also known as jumper wires, are essential for creating temporary electrical connections on the donut board. They provide a flexible and easy-to-use means of connecting various components during the prototyping phase.
4.	Silicone Tube	The silicone tube is used to create a conduit for water between the water pump and the water level sensor. It ensures a smooth flow of water and avoids leakages.
5.	Water Pump	The water pump is a crucial component responsible for pumping water from a water source to a designated location, such as a storage tank. It will be controlled by the ESP8266 NodeMCU through a relay.

6.	Water Level Sensor	The water level sensor is used to measure the water level in real-time. It provides data to the ESP8266 NodeMCU, which will then make decisions regarding pump control based on the water level readings.
7.	Battery	The batteries are the energy source for the system, supplying power to the ESP8266 NodeMCU and other components when the system is not connected to a mains power supply.
8.	Solenoid Valve	The solenoid valve is an electromechanical valve used to control the flow of water. It can shut off the water supply when necessary, as directed by the ESP8266 NodeMCU.
9.	The Double Channel 5V Relay Breakout Board	an electronic component used to control high-power devices, such as water pumps, using a low-power microcontroller like the ESP8266 NodeMCU. It consists of two separate relays on a single board, each capable of switching high voltage and current circuits on and off.
10.	Voltage Regulator	The voltage regulator is an electronic component used to maintain a stable voltage level within the system. It ensures that the components receive the correct and consistent power supply, preventing damage from fluctuations.
11.	Step-Down Module (Buck Converter)	Step-Down Module is an electronic device used to efficiently convert a higher voltage to a lower voltage level. It is commonly used in power supply circuits to step down the voltage and provide a stable and regulated output for various electronic components and devices. The Buck Converter operates by switching the input voltage to achieve the desired output voltage, making it an essential component in many electrical and electronic applications.
12.	Terminal Block	A terminal block is a modular electrical connector used to safely and securely connect and terminate wires or conductors. It provides a convenient and organized way to join multiple wires together, making it easier to connect them to a power source, a circuit, or other electrical components

II. Software Requirement

The table of Software Specifications provides a concise summary of the essential details and functionalities of the hardware components used in the water pressure detection and automatic water pump control system for commercial properties.

Table 2.0 : Software Specifications

No.	Software	Description
1.	Telegram	Telegram is a cloud-based instant messaging app that will be used as a communication platform in the smart water pressure detection and automatic pump control system. It enables the system to send instant notifications and alerts to property managers regarding water levels and pump status.
2.	Arduino IDE	The Arduino Integrated Development Environment (IDE) is an open-source software used for programming the ESP8266 NodeMCU microcontroller. It provides a user-friendly interface for writing, compiling, and uploading the code to the microcontroller.

d. System Configuration

1. Hardware Components:

The technical report discusses the hardware components used in the automatic water pump system. The report emphasises the integration and functionality of these hardware components to create an automated and efficient water supply system.

Table 1.0: Hardware Components

No	Components	Description
1.	ESP8266 NodeMCU	The central microcontroller responsible for system control and interfacing with other components.
2.	Water Level Sensor	Monitors water pressure in real-time and provides data to the NodeMCU.
3.	Double Channel 5V Relay Breakout Board	Controls the water pump based on the water level readings.
4.	Water Pump	The pump that supplies water to the commercial property and is automatically controlled by the system.
5.	Donut Board	A breadboard or prototype board used for circuit connections.
6.	Voltage Regulator	Ensures stable power supply to the NodeMCU and other components.
7.	Jumper Connectors (20 pieces)	Establish electrical connections on the breadboard.
8.	Silicone Tube	Connects the water level sensor to the water source.
9.	Battery	The power source for the system when using batteries.
10.	Solenoid Valve	Optional component for controlling water flow, if needed.

2. Software Components:

A software component is a piece of computer code or program that performs a specific function or task within a larger software system.

I. Telegram: Messaging platform used to send instant alerts and notifications to property managers.

II. Arduino IDE: Integrated development environment for programming the ESP8266 NodeMCU.

3. System Architecture:

System architecture is the high-level design and structure of a system, outlining how its components interact and work together to achieve its goals.

I. The ESP8266 NodeMCU serves as the central processing unit, receiving data from the water level sensor and controlling the water pump through the relay breakout board.

II. The water level sensor continuously monitors water pressure and sends the data to the NodeMCU.

III. Based on the water level readings, the NodeMCU activates or deactivates the relay to turn the water pump on or off accordingly.

IV. Telegram integration allows the system to send real-time alerts and notifications to property managers.

4. Power Source:

I. The system can be powered by either an external power supply or a set of 3x18650 batteries when a power source is not available.

II. The NodeMCU manages power usage efficiently, especially when running on battery power.

5. Communication Protocol:

I. The ESP8266 NodeMCU communicates with the water level sensor to receive water pressure data.

II. The NodeMCU controls the relay breakout board through digital output signals to manage the water pump.

III. Telegram Bot API is used to send instant alerts and notifications to property managers.

6. User Interface:

I. The user interface is provided through Telegram, allowing property managers to interact with the system and receive real-time notifications on their devices.

7. Environmental Considerations:

I. The system should be designed to withstand outdoor conditions if deployed in commercial properties with outdoor water sources.

II. Components may need waterproofing to ensure reliability and durability in wet environments.

8. Data Management:

I. The system may store relevant data locally on the NodeMCU for monitoring and control purposes.

II. External cloud services could be leveraged for data storage and analysis if required.

9. Implementation Approach:

I. The Prototype Model will be used for development, involving iterative cycles of prototyping, testing, and improvement to achieve the desired functionalities and performance.

4.0 Final Design

A. Logical Design

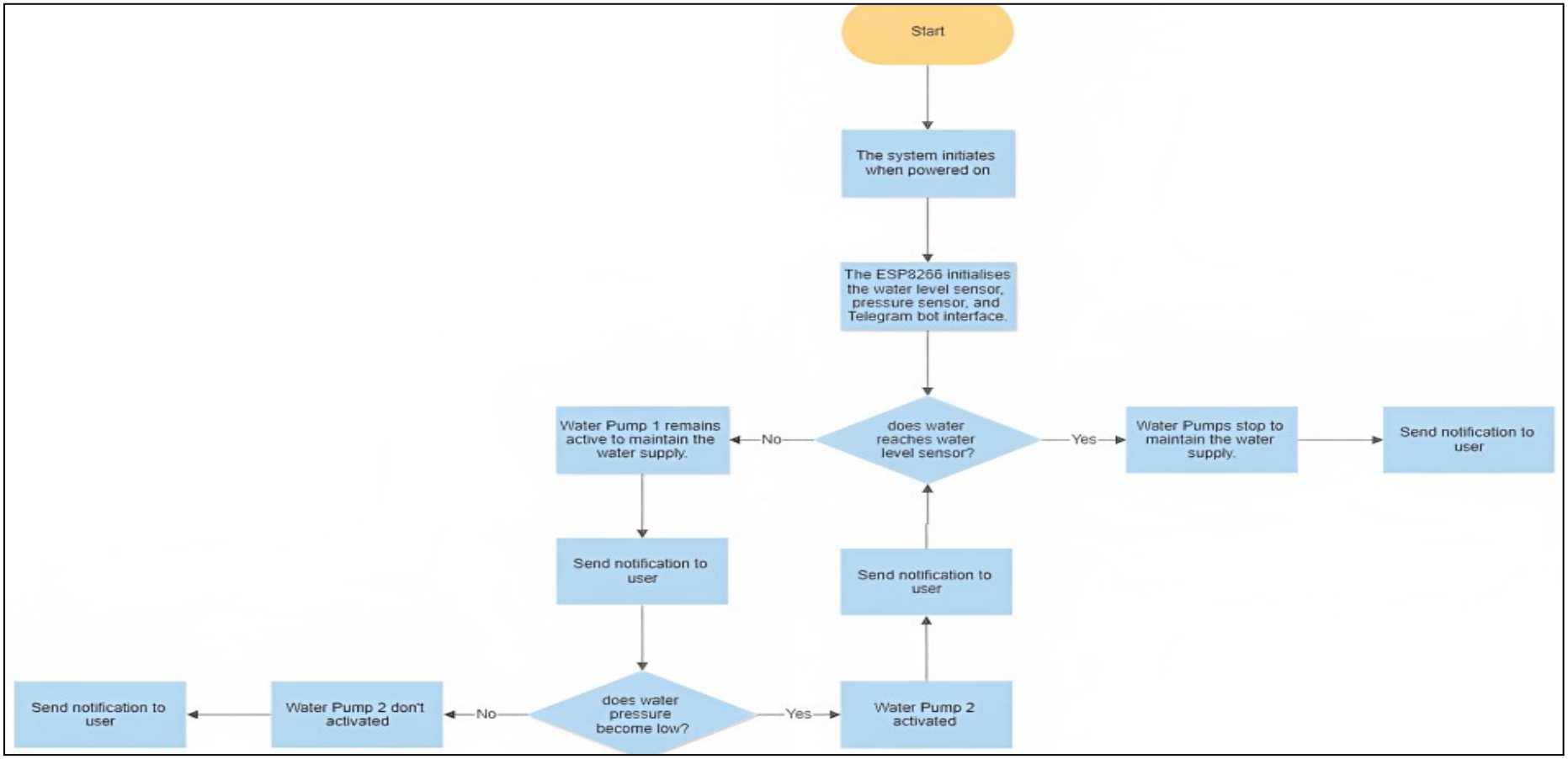

Diagram 1.0: Flowchart of how the product works

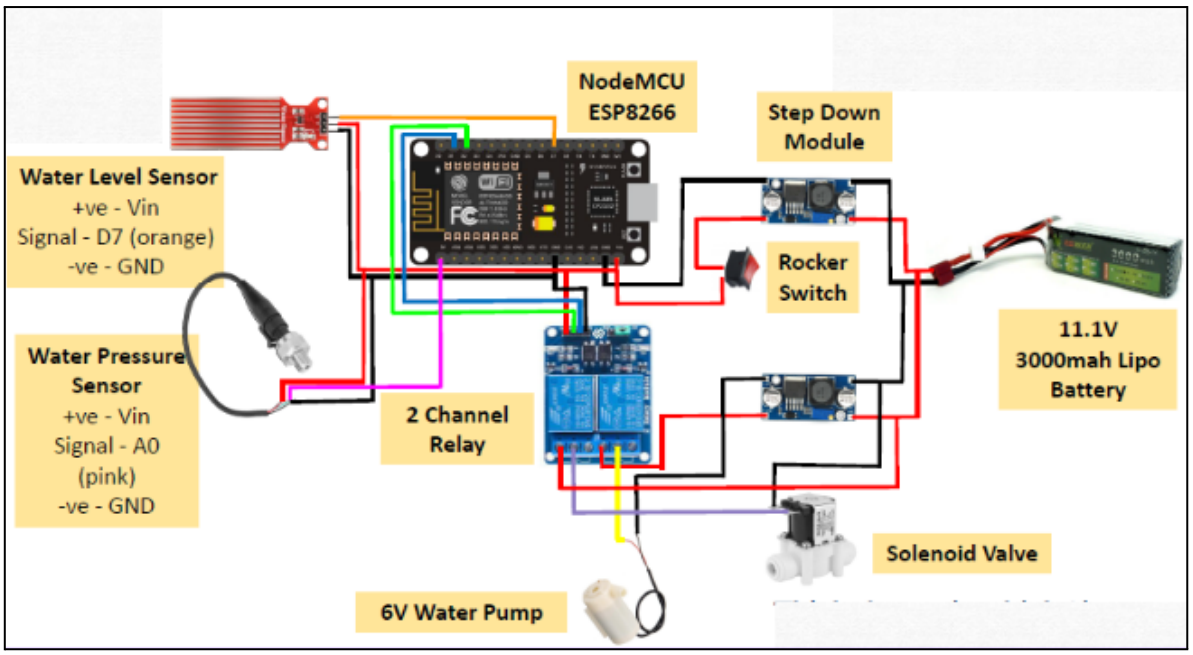

Diagram 2.0: Project schematic diagram

1. Initialization:

I. The ESP8266 powers up and initialises the system.

II. Calibration: Calibrate the water level sensor and pressure sensor for accurate readings.

2. Monitoring Water Level and Pressure:

I. The water level sensor continuously measures the water level in the tank.

II. The pressure sensor constantly monitors the water pressure in the system.

3. Telegram Notification Setup:

I. The system is configured with a Telegram bot, allowing it to send notifications to a specific user or group.

II. The user's Telegram credentials (e.g., API token, chat ID) are securely stored in the ESP8266 for message sending.

4. Normal Operation:

I. If the water level is above a pre-defined "high" level, and the water pressure is within an acceptable range, the system operates in normal mode.

II. Water Pump 1 remains active, providing a steady water supply to the water lines.

5. Low Water Pressure Detection:

I. If the pressure sensor detects low water pressure (below a pre-set threshold), it indicates that Water Pump 1 is struggling to maintain adequate pressure.

II. The system enters the low water pressure mode.

6. Low Water Pressure Mode:

I. In this mode, the system verifies if Water Pump 2 is available and operational.

II. If Water Pump 2 is functional:

III. The ESP8266 activates Water Pump 2 to augment the water supply to increase the pressure.

IV. The solenoid valve is opened to allow the water from Water Pump 2 to flow into the main water line.

7. Water Level Monitoring in Low Water Pressure Mode:

I. While operating in low water pressure mode, the system continues to monitor the water level.

8. Return to Normal Operation:

I. Once the water pressure reaches an acceptable level (as detected by the pressure sensor) in low water pressure mode, the system returns to normal operation.

II. Water Pump 2 is deactivated, and the solenoid valve is closed.

III. The system sends a notification via Telegram to inform the user about the return to normal operation.

9. Periodic Status Updates:

I. The system sends periodic status updates via Telegram, providing information about the current water level and water pressure.
II. The user can use the Telegram interface to check the status of the system at any time.

B. Physical Design

The physical design of the prototype for the Automatic Water Pump Controller is depicted in the figure below:

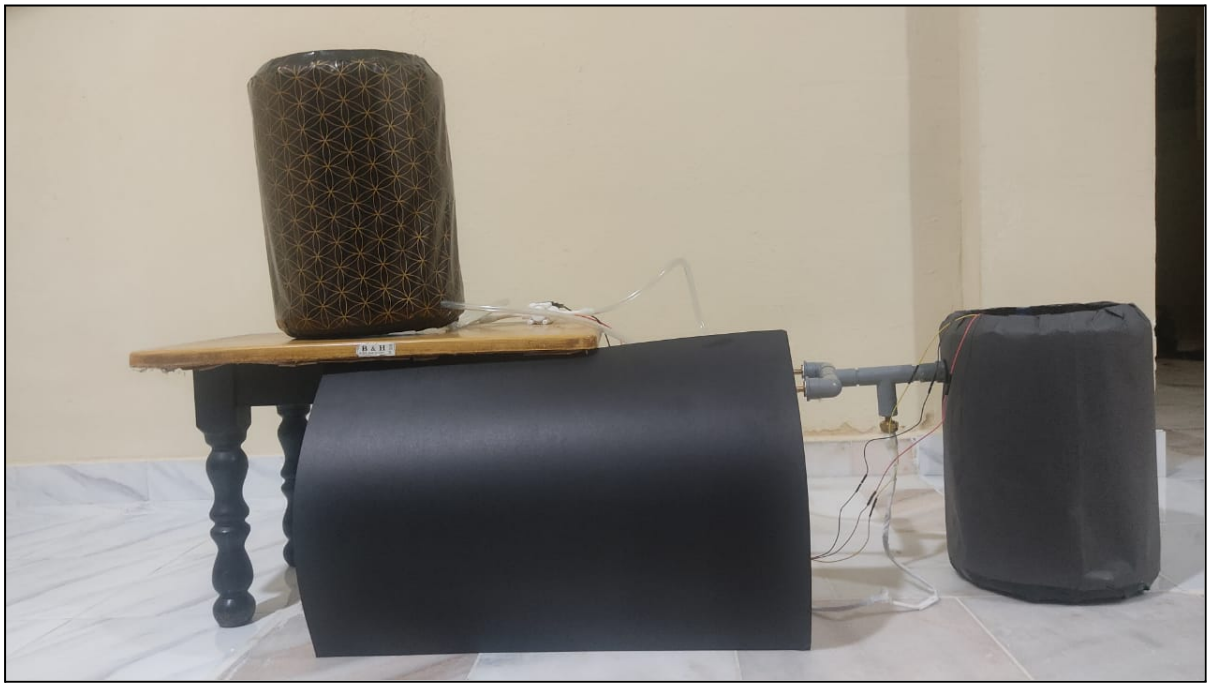

Diagram 1.0: Physical Design of the Prototype

Table 1.0 Physical Component for the Prototype.

The table below provides an overview of the physical components used in the Automatic Water Pump Control Project:

No	Name	Description
1.	Water Level Sensor  Diagram 1.0	I. The water level sensor is installed inside the water tank. II. It is positioned to measure the water level accurately and reliably. III. The sensor is connected to the ESP8266 using appropriate wiring and connectors.
2.	Pressure Sensor  Diagram 2.0	I. The pressure sensor is placed along the water supply line. II. It is positioned to measure the water pressure effectively. III. The sensor is connected to the ESP8266 using appropriate wiring and connectors.

3.	Water Pump  Diagram 3.0	I. A Water Pump is the primary water pump responsible for maintaining the water supply. II. It is positioned near the water source or storage tank and connected to the water supply line. III. The pump is powered by a suitable power source and controlled by the ESP8266.
4.	Solenoid Valve  Diagram 4.0	I. The solenoid valve is placed in the water supply line Water Pump 2. II. It controls the flow of water from Water Pump 2 to the main water line when activated. III. The solenoid valve is connected to the ESP8266.

5.	ESP8266  Diagram 5.0	I. The ESP8266 is the central control unit of the system. II. It processes the input from the water level sensor and pressure sensor. III. It controls the activation and deactivation of Water Pump 1, Water Pump 2, and the solenoid valve based on sensor readings and system logic. IV. The ESP8266 also handles the communication with the Telegram bot for sending notifications.
6.	Telegram Bot Interface  Diagram 6.0	I. The Telegram bot interface is integrated into the ESP8266. II. It allows the system to send notifications and receive commands via Telegram. III. User credentials, such as API token and chat ID, are securely stored in the ESP8266.

7.	Power Supply  Diagram 7.0	I. The entire system requires a stable and reliable power supply. II. Depending on the power requirements of the components, an appropriate power source is chosen.

C. Experimental Concept

a. Experimental Design:

The experimental concept aims to demonstrate the functionality and feasibility of an automatic water pump system with water level and pressure sensors. The system will automatically switch to a secondary water pump and open a solenoid valve when low water pressure is detected. Additionally, the system will send notifications via Telegram to monitor water level and water pressure in real-time.

b. Experimental Procedure:

1. Calibration and Initialization:

I. The water level sensor and pressure sensor are calibrated for accurate readings.
II. The ESP8266 is initialized, and the Telegram bot interface is set up with user credentials.

2. Normal Operation:

I. Water Pump 1 is activated to maintain the water supply.
II. The system continuously monitors the water level and pressure through the sensors.

3. Low Water Pressure Detection:

I. If the pressure sensor detects low water pressure (below a pre-set threshold), the system enters the low water pressure mode.

4. Low Water Pressure Mode:

I. The ESP8266 checks if Water Pump 2 is operational.
II. If Water Pump 2 is available, it is activated, and the solenoid valve is opened to allow the water from Water Pump 2 to flow into the main water line.
III. A notification is sent via Telegram to inform the user about the switch to Water Pump 2 and the activation of the solenoid valve.

5. Water Level Monitoring in Low Water Pressure Mode:

I. The system continuously monitors the water level during low water pressure mode.

6. Return to Normal Operation:

I. Once the water pressure reaches an acceptable level (as detected by the pressure sensor) in low water pressure mode, the system returns to normal operation.
II. Water Pump 2 is deactivated, and the solenoid valve is closed.
III. A notification is sent via Telegram to inform the user about the return to normal operation.

7. Real-time Monitoring:

I. The system continuously sends real-time notifications via Telegram about the water level and water pressure status.

The experimental concept will be evaluated based on the following criteria:

I. Efficiency in maintaining water pressure during normal operation.
II. Effectiveness in switching to Water Pump 2 during low pressure.
III. Accuracy in water level measurement and responsiveness to critical low water levels.
IV. Reliability of Telegram notifications for real-time monitoring.
V. Performance in handling multiple scenarios and variations in water pressure.

The results and findings will help determine the viability and potential improvements for a real-world implementation of the automatic water pump system.

5.0 Test Description and Results

The testing phase of the project involves two main types of testing: unit testing and integration testing. Unit testing focuses on testing individual components or units of the system in isolation, ensuring that each unit works as intended. Integration testing, on the other hand, involves testing the interactions between different components to verify the seamless integration of the entire system.

Unit Testing Plan

Table 1.0: Unit Testing Plan for Automatic Water Pump System (Water Level Sensor):

No.	Test Case Name	Test Procedure	Pre-condition	Expected Result	Tester	Result (Pass/Failure)
1.	Water Level Sensor Maximum Limit Test	Fill the water tank to a level exceeding the maximum limit that the water level sensor can detect, activate the system, and record the displayed water level reading	None	The water level sensor should accurately report the water level reading, indicating that it has reached the maximum limit. The system should handle this situation appropriately, possibly triggering an alert or taking necessary actions to prevent any overflow or damage.	Sir Fahmi	Pass
2.	Water Level High Detection and Pump Stop Test	Fill the water tank to a level where the water level sensor detects a high water level, activate the system, and observe if it triggers the water pump to stop in response to the high water level.	None	The water level sensor should accurately detect the high water level, and as a result, the system should trigger the water pump to stop automatically.	Sir Fahmi	Pass

Table 2.0: Unit Testing Plan for Automatic Water Pump System (Pressure Sensor):

No.	Test Case Name	Test Procedure	Pre-condition	Expected Result	Tester	Result (Pass/Failure)
1.	Water Pressure Sensor Accuracy Test	Connect the water pressure sensor to the prototype system, apply a known and controlled water pressure, initiate the measurement process, and record the displayed water pressure reading to verify it.	None	The water pressure sensor's accuracy in measuring water pressure is confirmed by ensuring that it accurately measures high or low pressure readings, which should align with the known and controlled pressure applied during the test.	Sir Fahmi	Pass

Table 3.0: Unit Testing Plan for Automatic Water Pump System (Water Pump):

No.	Test Case Name	Test Procedure	Pre-condition	Expected Result	Tester	Result (Pass/Failure)
1.	Water Pump Functionality Test	Verify that the water pump activates when the water level does not reach the sensor and deactivates when the water level reaches the sensor, ensuring it correctly operates as intended through the prototype system.	None	Confirm that the water pump, correctly connected to the prototype system and installed in a water supply setup, operates to turn on when the water level does not reach the sensor and turn off when the water level reaches the sensor.	Sir Fahmi	Pass
2.	Water Pump Activation and Deactivation Test	Verify that the water pump correctly activates and deactivates based on control signals sent through the prototype system, ensuring it starts and stops water supply as intended.	None	When the control signal to activate the water pump is sent through the prototype system, the water pump should promptly start operating and supply water as intended. Similarly, when the control signal to deactivate the water pump is sent, the water pump should cease operation and stop the water supply as intended.	Sir Fahmi	Pass

Table 4.0: Unit Testing Plan for Automatic Water Pump System (Solenoid Valve):

No.	Test Case Name	Test Procedure	Pre-condition	Expected Result	Tester	Result (Pass/Failure)
1.	Solenoid Valve Functionality Test	Verify that the solenoid valve correctly opens and allows water flow when activated through the prototype system and closes to stop water flow when deactivated. The solenoid valve will open when pressure low and the solenoid valve will close and deactivate when pressure high.	None	When the solenoid valve activation process is initiated through the prototype system, the valve should open, enabling water flow through the system. Similarly, when the solenoid valve deactivation process is initiated, the valve should close, stopping water flow through the system, confirming its proper functionality in controlling water flow as intended.	Sir Fahmi	Pass

Table 5.0:Unit Testing Plan for Automatic Water Pump System(Telegram Notification):

No.	Test Case Name	Test Procedure	Pre-condition	Expected Result	Tester	Result (Pass/Failure)
1.	Telegram Water Level Notification and Water Pump Control Test	Ensure the water level sensor is correctly connected, Telegram integration is set up for message sending, verify Telegram accessibility, simulate low water level to trigger a notification and water pump activation, then simulate high water level to trigger another notification and water pump deactivation, confirming their correct operation.	None	When the water level does not reach the water level sensor, the system should send a notification through Telegram indicating low water level, and the water pump should be activated to pump the water. Conversely, when the water level reaches the water level sensor, the system should send a notification through Telegram indicating high water level, and the water pump should be deactivated and stop pumping water. This ensures that the Telegram notifications and water pump control function as expected based on the water level readings from the water level sensor.	Sir Fahmi	Pass

Integration Testing Plan

Table 1.0: Integration Testing Plan for Automatic Water Pump System (Water Level Sensor and ESP8266):

No.	Test Case Name	Test Procedure	Pre-condition	Expected Result	Tester	Result (Pass/Failure)
1.	Connect and Verify Data Transmission	Connect the water level sensor to the ESP8266 according to the provided hardware integration specifications.	None	The ESP8266 successfully establishes a connection with the water level sensor, and data transmission occurs smoothly. The ESP8266 accurately receives the test data from the water level sensor, and there are no transmission errors or data loss during the process.	Sir Fahmi	Pass
2.	Data Accuracy and Processing	Send predefined test data with known water levels from the water level sensor to the ESP8266.	None	The ESP8266 precisely reads and processes the test data sent by the water level sensor. The data received by the ESP8266 matches the known water levels, confirming the accurate functionality of the integration.	Sir Fahmi	Pass

Table 2.0: Integration Testing Plan for Automatic Water Pump System (Pressure Sensor and ESP8266):

No.	Test Case Name	Test Procedure	Pre-condition	Expected Result	Tester	Result (Pass/Failure)
1.	Connect and Verify Data Transmission	Connect the pressure sensor to the ESP8266 according to the provided hardware integration specifications.	None	The ESP8266 successfully establishes a connection with the pressure sensor, and data transmission occurs smoothly. The ESP8266 accurately receives the test data from the pressure sensor, and there are no transmission errors or data loss during the process.	Sir Fahmi	Pass
2.	Data Accuracy and Processing	Send predefined test data with known water pressure levels from the pressure sensor to the ESP8266.	None	The ESP8266 precisely reads and processes the test data sent by the pressure sensor. The data received by the ESP8266 matches the known water pressure levels, confirming the accurate functionality of the integration.	Sir Fahmi	Pass

Table 3.0: Integration Testing Plan for Automatic Water Pump System (Water Pumps and ESP8266):

No.	Test Case Name	Test Procedure	Pre-condition	Expected Result	Tester	Result (Pass/Failure)
1.	Connect and Verify Control Signals	Connect Water Pump 1 and Water Pump 2 to the ESP8266 according to the provided hardware integration specifications.	None	The ESP8266 successfully establishes connections with Water Pump 1 and Water Pump 2. Both pumps respond to the test control signals, indicating that they are properly integrated with the ESP8266	Sir Fahmi	Pass
2.	Activation and Deactivation based on Sensor Readings	Simulate different sensor readings, such as low water pressure or low water levels, to trigger the activation and deactivation of the water pumps.	None	The ESP8266 accurately processes sensor readings and effectively controls the water pumps. It correctly activates Water Pump 2 when low water pressure is detected, ensuring sufficient water supply. Water Pump 2 deactivates when water pressure returns to normal, preventing unnecessary operation. Water Pump 1 remains active during normal operation and shuts down when low water levels are detected, safeguarding the pump from damage.	Sir Fahmi	Pass

Table 4.0: Integration Testing Plan for Automatic Water Pump System(Solenoid Valve and ESP8266):

No.	Test Case Name	Test Procedure	Pre-condition	Expected Result	Tester	Result (Pass/Failure)
1.	Connect and Verify Control Signals	Connect the solenoid valve to the ESP8266 according to the provided hardware integration specifications.	None	The ESP8266 successfully establishes a connection with the solenoid valve. The valve responds to the test control signals, indicating that it is properly integrated with the ESP8266.	Sir Fahmi	Pass
2.	Opening and Closing based on Sensor Readings	Simulate different sensor readings, such as low water pressure, to trigger the opening of the solenoid valve.	None	The ESP8266 accurately processes sensor readings and effectively controls the solenoid valve. It correctly opens the solenoid valve when low water pressure is detected, ensuring the flow of water from Water Pump 2 to maintain sufficient water pressure. The solenoid valve closes when normal water pressure is restored, preventing unnecessary water flow.	Sir Fahmi	Pass

Table 5.0: Integration Testing Plan for Automatic Water Pump System (Telegram Bot Interface and ESP8266):

No.	Test Case Name	Test Procedure	Pre-condition	Expected Result	Tester	Result (Pass/Failure)
1.	Connect and Verify Message Sending	Connect the Telegram bot interface to the ESP8266 according to the provided integration specifications.	None	The ESP8266 establishes a connection with the Telegram bot interface, and the test messages are sent without any errors. The successful transmission confirms that the Telegram bot interface is properly integrated with the ESP8266.	Sir Fahmi	Pass
2.	Notification Sending to Telegram User	Set up the Telegram bot interface to send notifications to a designated Telegram user or group.	None	The ESP8266 effectively sends real-time notifications to the designated Telegram user/group when specific events are detected. Users receive timely alerts about low water pressure or critical system conditions, ensuring prompt awareness and facilitating necessary actions	Sir Fahmi	Pass

Unit Testing Results: All unit tests have passed, indicating that individual components (water pump and water level sensor) are functioning correctly.

Integration Testing Results: The integration tests have also passed, indicating that the different components (pump, solenoid valve, water flow, and pressure) interact seamlessly and the system operates as intended.

Overall, the testing phase has been successful, with all unit and integration tests passing. This demonstrates that the water supply system is reliable, with accurate water level sensing, efficient pump activation, and seamless integration between components.

6.0 Major Findings and Discussion

The description of each content of the significant findings and discussions is defined as follows:

a) The Advantages of The Project

I. Improved Water Pressure: The installation of a water pump in the system ensures that users experience better water pressure throughout their homes, making it easier to perform tasks like showering, washing dishes, and doing laundry.

II. Consistent Water Supply: By using a larger tank as the water source and incorporating water level sensors, the system can maintain a steady water supply. Users won't face interruptions in their water usage due to low water levels in the tank.

III. Efficient Water Usage: The automation in the system, where the pump activates only when needed, helps in optimizing water usage. This can lead to reduced water wastage and lower water bills for users.

IV. Easy Monitoring: Sending updates and readings through Telegram allows users to monitor the system's status and be aware of any issues or maintenance requirements promptly.

V. User-Friendly: The system handles most operations automatically, making it easy for users to manage and control their water supply without manual intervention.

b) The Disadvantages of The Project

I. Initial Cost: The installation of water pumps, water level sensors, and other components might result in higher initial costs. Users might hesitate to adopt the system due to the upfront expenses.

II. Maintenance: As with any mechanical system, regular maintenance is essential to ensure the water supply remains efficient. Users will need to budget for ongoing maintenance and occasional repairs.

III. Dependency on Technology: The system's reliance on sensors, pumps, and automation means that if any of these components fail, it could disrupt the water supply until the issue is resolved.

IV. Energy Consumption: Running water pumps and other electrical components require energy. Depending on the efficiency of the system and the frequency of pump usage, this could increase energy consumption and impact the environment.

V. Limited Water Source: The project relies on a larger tank as the water source. In areas with limited water resources, this approach may not be suitable for extended periods or during droughts.

VI. Complexity: The integration of multiple components and automation might make the system complex to understand and troubleshoot for some users.

7.0 Recommendation and Conclusion

Conclusions about the Project:

The project has successfully demonstrated the implementation of a water supply system that addresses the issue of low water pressure and inconsistent water flow in users' homes. By integrating a water pump, water level sensor, and automation features, the system has improved the overall water delivery experience for users. It ensures a steady and sufficient water supply, enhances water pressure, and optimizes water usage, all of which are essential factors in providing a convenient and reliable water distribution system.

The use of two pipes, one with a water pump and the other with a solenoid valve, allows for a flexible and efficient approach to regulating water pressure. The automation feature, where the system automatically activates and deactivates the water pump based on water levels and pressure, offers user convenience and minimizes manual intervention. The reporting of system status through Telegram allows users to monitor the water supply remotely, ensuring transparency and prompt awareness of any issues.

Importance of the Outcome:

The outcome of the project holds significant importance on various levels. Firstly, it directly impacts the users by providing them with an improved and consistent water supply. This can lead to enhanced quality of life, better hygiene practices, and more efficient household chores. The increased water pressure can be particularly beneficial for multi-story buildings or areas with historically poor water pressure.

Additionally, the project contributes to sustainable water management practices. By optimizing water usage through automation and monitoring, it helps reduce water wastage, which is crucial in regions facing water scarcity or environmental challenges. The focus on responsible water consumption aligns with global efforts to conserve water resources and promote environmental sustainability.

The project's success also highlights the potential of technology and automation in solving real-world problems. It serves as a practical example of how IoT (Internet of Things) devices and smart systems can be integrated to improve infrastructure and utility services.

Recommendations for Future Works:

To enhance the project for future works, students can consider the following recommendations:

Energy Efficiency: Explore ways to increase the system's energy efficiency. This could involve optimizing the water pump's operation or investigating the use of energy-efficient pumps and components.

Leak Detection: Integrate leak detection sensors into the system. Identifying and addressing leaks promptly can further minimize water wastage and prevent potential damage to the infrastructure.

User Customization: Provide users with the ability to customize certain settings, such as water pressure thresholds or pump activation timings. Allowing users to tailor the system to their specific needs and preferences can improve user satisfaction.

Mobile App Interface: Develop a user-friendly mobile app interface for remote monitoring and control. A dedicated mobile app would offer a more intuitive and accessible way for users to interact with the system.

Water Treatment Options: Consider integrating water treatment options, such as filtration or purification systems, to ensure that the water supplied is of the highest quality and safe for consumption.

By implementing these recommendations, future iterations of the project can further refine and expand its capabilities. These enhancements can lead to a more energy-efficient, user-customizable, and environmentally conscious water supply system, benefitting both the users and the broader community in their efforts to conserve water and promote sustainable living.

8.0 References

Rujukan Internet

1. Santos, S., & Santos, S. (2020). Telegram: Control ESP32/ESP8266 Outputs with Arduino IDE | Random Nerd Tutorials. *Random Nerd Tutorials*. https://randomnerdtutorials.com/telegram-control-esp32-esp8266-nodemcu-outputs/

2. Hackster.io. (2021b, August 29). *Telegram Bot using NodeMCU | esp8266 Automation*. https://www.hackster.io/electronicprojects/telegram-bot-using-nodemcu-esp8266-automation-68377c

3. Hobby, S. (2023). WATER LEVEL SENSOR Tutorial| How to use WATER LEVEL SENSOR. *SriTu Hobby*. https://srituhobby.com/water-level-sensor-tutorial-how-to-use-water-level-sensor/

4. Hackster.io. (2021b, August 30). *Water level indicator using NodeMCU*. https://www.hackster.io/Techatronic/water-level-indicator-using-nodemcu-50e173

5. *Building an ESP8266 with multiple pressure sensors (using ADS1115 and ESPHome)*. (2020, March 5). Home Assistant Community. https://community.home-assistant.io/t/building-an-esp8266-with-multiple-pressure-sensors-using-ads1115-and-esphome/176518

Rujukan Video

1. Onkar Mendhapurkar. (2022, October 22). How To Make Home Security System | Using ESP8266 | Get notifications on Telegram! | Detailed Video. [Video]. YouTube. https://www.youtube.com/watch?v=_iyvJMkf_yY

2. Tech Trends Shameer. (2021, August 23). Control water pump from mobile | IOT Projects | Home Automation [Video]. YouTube. https://www.youtube.com/watch?v=Skgv3DF5qxw

3. Ovens Garage. (2020b, April 6). Pressure Sensor - arduino [Video]. YouTube. https://www.youtube.com/watch?v=UrqPxwsPWGk

4. Enjoy Mechatronics. (2022, April 22). Arduino Tutorial 37- Water Level Sensor (Set Up & Prgramming) [Video]. YouTube. https://www.youtube.com/watch?v=lEe-x7sXAVk

9.0 Entrepreneur Mind Set

To approach the development and implementation of an automatic water pump system with water level and pressure monitoring, and Telegram notifications, an entrepreneur mindset is crucial. Here are key aspects of the entrepreneur mindset that can drive the success of the project:

Vision and Purpose: As an entrepreneur, you need a clear vision of the project's purpose and the problem it aims to solve. Understanding the potential impact of the automatic water pump system on water supply efficiency and management will drive your motivation and passion for the project.

Innovation and Creativity: Embrace innovation and creativity to design a unique and efficient automatic water pump system. Seek novel approaches and solutions that set your system apart from existing solutions in the market.

Market Research: Conduct thorough market research to understand the demand for such a system, identify potential competitors, and grasp the needs and preferences of the target audience. This research will inform your system's features and functionalities.

Problem-Solving Attitude: As challenges arise during development, maintain a problem-solving attitude. Entrepreneurship often involves navigating obstacles and finding effective solutions.

Appendix

Coding :-

```
#ifdef ESP32
#include <WiFi.h>
#else
#include <ESP8266WiFi.h>
#endif
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>

// Replace with your network credentials
const char* ssid = "Redmi Note 9S";
const char* password = "11111112";

// Initialize Telegram BOT
#define BOTtoken "6310651934:AAFq5qME3R_eU5aXSBk6OhqOaw9DJLXT7tQ"
#define CHAT_ID "1431735849"

WiFiClientSecure client;
```

```
UniversalTelegramBot bot(BOTtoken, client);


#define waterLevelPin D7     // water level sensor pin (Analog input)
#define pressurePin A0       // water pressure sensor pin (Analog input)
#define pumpControlPin D2    // relay sensor pin IN1
#define relay2Pin D1         // relay sensor pin IN2


// Thresholds for water level (adjust according to your sensor)
const int waterLevelHighThreshold = 500; // Adjust this value for the high water level
const int waterLevelLowThreshold = 300;  // Adjust this value for the low water level


// Flag to indicate if the pump is currently on
bool pumpOn = false;


// Checks for new messages every 1 second.
int botRequestDelay = 1000;
unsigned long lastTimeBotRan;


// Function to send a notification message to Telegram
void sendNotification(const String& message) {
  bot.sendMessage(CHAT_ID, message, "");
```

```
}

void setup() {
  Serial.begin(115200);

#ifdef ESP8266
  // Load root certificate for api.telegram.org directly
  client.setInsecure(); // Use this line if your ESP8266 core version doesn't support loading certificates
#endif

  // Connect to Wi-Fi
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi..");
  }

  // Print ESP32/ESP8266 Local IP Address
  Serial.println(WiFi.localIP());
```

```
  // Set pump control pin and relay2 pin as OUTPUT
  pinMode(pumpControlPin, OUTPUT);
  pinMode(relay2Pin, OUTPUT);

  // Initially turn off the pump and relay2
  digitalWrite(pumpControlPin, HIGH);
  digitalWrite(relay2Pin, HIGH);

  // Send a notification on setup
  sendNotification("Device is online and ready.");
}

bool lastPumpState = false;
bool lastSolenoidState = false;
int lastWaterPressure = 0;

void loop() {
  if (millis() > lastTimeBotRan + botRequestDelay) {
    int numNewMessages = bot.getUpdates(bot.last_message_received + 1);
```

```
  while (numNewMessages) {
    Serial.println("Got response");
    numNewMessages = bot.getUpdates(bot.last_message_received + 1);
  }
  lastTimeBotRan = millis();
}

// Determine the state of the water pump
bool currentPumpState = (digitalRead(pumpControlPin) == LOW);

// Determine the state of the solenoid valve
bool currentSolenoidState = (digitalRead(relay2Pin) == LOW);

// Check for changes in the water pump state
if (currentPumpState != lastPumpState) {
  lastPumpState = currentPumpState;
  pumpOn = currentPumpState;

  // Send a message to Telegram indicating the water pump state change
  if (pumpOn) {
    sendNotification("Water pump is ON.");
```

```
  } else {
    sendNotification("Water pump is OFF.");
  }
}

// Check for changes in the solenoid valve state
if (currentSolenoidState != lastSolenoidState) {
  lastSolenoidState = currentSolenoidState;

  // Send a message to Telegram indicating the solenoid valve state change
  if (currentSolenoidState) {
    sendNotification("Solenoid valve is ON.");
  } else {
    sendNotification("Solenoid valve is OFF.");
  }
}

// Read the water level sensor value
int waterLevel = analogRead(waterLevelPin);

// Print the water level value to the Serial Monitor
```

```
Serial.print("Water Level: ");
Serial.println(waterLevel);

// Check if water level is below the threshold
if (waterLevel < waterLevelLowThreshold) {
  // Print the water level value to the Serial Monitor
  Serial.println("Water Level LOW");
  digitalWrite(pumpControlPin, LOW); // Turn ON the pump
  sendNotification("Water level is LOW. Pump is ON.");
} else if (waterLevel > waterLevelHighThreshold) {
  // Print the water level value to the Serial Monitor
  Serial.println("Water Level HIGH");
  digitalWrite(pumpControlPin, HIGH); // Turn OFF the pump
  sendNotification("Tank is FULL. Pump is OFF.");
}

// Read the water pressure sensor value
int waterPressure = analogRead(pressurePin);

// Check for changes in the water pressure value
if (waterPressure != lastWaterPressure) {
```

```
    lastWaterPressure = waterPressure;


    // Send a notification with the current water pressure value
    sendNotification("Current Water Pressure: " + String(waterPressure));
  }


  // Control relay2 based on water pressure
  if (waterPressure < 200) {
    digitalWrite(relay2Pin, LOW); // Turn ON relay2 (water pump)
  } else {
    digitalWrite(relay2Pin, HIGH); // Turn OFF relay2 (water pump)
  }
}
```