DEPARTMENT OF MECHANICAL ENGINEERING

RV COLLEGE OF ENGINEERING®

(Autonomous Institution affiliated to VTU, Belagavi)

PROJECT REPORT

On

Smart Rain Shed Controller

Submitted in partial fulfilment of First year IDEA Lab

Submitted by

1. **BHAVYA CHAWAT [1RV24IS025]**
2. **CHETAN ANAND N [1RV24028]**
3. **AARADHYA SAXENA [1RV24IS003]**
4. **ANIKETA SUBUDHI [1RV24IS014]**

Under the guidance of

Mr. Keshava Murthy Y C
Assistant Professor

Dept. of Mechanical Engineering

2024 – 2025

RV COLLEGE OF ENGINEERING®, BENGALURU–560059
(*Autonomous Institution Affiliated to VTU, Belagavi*)
DEPARTMENT OF MECHANICAL ENGINEERING

CERTIFICATE

This is to certify that the project work titled "Smart Rain Shed Controller" carried out by [Bhavya Chawat (1RV24IS025)], [Chetan Anand N (1RV24IS028)], [Aaradhya Saxena (1RV24IS003)], [Aniketa Subudhi (1RV24IS014)], in partial fulfilment of the IDEA Lab (ME121DL) course introduced in the first year for all branches, during the academic year 2024-2025. It is certified that all necessary corrections/suggestions indicated during internal assessment have been duly incorporated in the final report. The project work has been approved as it satisfies the academic requirements in respect of the IDEA Lab Experiential Learning Project as prescribed by the institution.

Mr. Keshava Murthy Y C	Dr. Jagannatha Gupta V L	Dr. Shanmukha Nagaraj
Assistant Professor	Assistant Professor	Prof. & Head
Department of Mechanical Engineering	Department of Mechanical Engineering	Department of Mechanical Engineering

External Examiners

Name of Examiners	**Signature with Date**
1. ____________________	_________________________
2. ____________________	_________________________

ABSTRACT

Title: *Smart Rain Shed Controller*

This project presents a fully automated and remotely controllable IoT-based Smart Clothes Protection System designed to safeguard drying clothes from unexpected rainfall. The system utilizes both a digital rain sensor for rain presence detection and an analog rain sensor to measure rain intensity (0–100%), enabling smarter decisions based on actual weather conditions.

An LDR sensor is integrated to detect daylight, ensuring the shed only opens when it is bright outside—preventing unnecessary exposure at night. The shed operates in two modes: Automatic Mode, where the system autonomously opens or closes the shed based on rain, light, time window, and weather API data; and Manual Mode, where users can control the shed via the Blynk IoT app or a physical override switch installed on-site.

Live weather data is fetched every 10 minutes using the Tomorrow.io API, allowing preemptive closure of the shed during predicted rainfall. A user-defined time window (e.g., 6 AM to 8 PM) set through the app ensures that automation only occurs during appropriate hours.

A 16×2 I2C LCD provides real-time feedback about the current system state, including rain status, light condition, rain intensity, mode, and connectivity. Key actions and environmental conditions are continuously logged to Google Sheets via a Wi-Fi connection for remote monitoring and analysis. A debounce mechanism ensures false rain triggers are ignored unless rain persists for 5 seconds, improving stability.

The hardware includes an ESP32 microcontroller, L298N motor driver, DC motor, MT3608 booster, Li-ion battery pack, LM2596 Buck Converter, Solar Panel, TP4056 charging Module and other custom mechanical elements.

TABLE OF CONTENTS

Chapter 1 – Introduction

1.1 Background

In recent years, smart home automation has evolved rapidly with the integration of Internet of Things (IoT) technologies. Weather-responsive systems are becoming increasingly popular for ensuring safety, convenience, and energy efficiency in residential environments. One such application is automated protection of open terraces during rain. Traditional methods rely on human intervention to manually cover or retrieve items placed in open spaces, which is often inconvenient or delayed.

1.2 Problem Definition

People often hang clothes or use terraces for relaxation. However, during sudden weather changes like rainfall or low light, especially when unattended, these areas become exposed and vulnerable. Manually deploying a protective shed or cover is inefficient and impractical, particularly for residents living in high-rise apartments. There is a need for an automated and remotely accessible system that can operate a terrace cover intelligently based on real-time weather conditions.

1.3 Motivation

The motivation behind this project stems from the inconvenience faced in daily life due to unexpected rain, especially when one is not present at home. With advancements in microcontrollers and wireless communication, a cost-effective and scalable solution can be built using ESP32, sensors, and mobile control to create a smart protective system.

1.4 Objectives

- To design and prototype an IoT-based system that detects rain and low-light conditions on a terrace.
- To automate the deployment of a protective shed using a DC motor and motor driver.
- To integrate a manual override and power ON/OFF switch for user control.
- To implement false trigger prevention for reliable system response.
- To enable real-time status display using an LCD and remote monitoring/control via the Blynk IoT platform.
- To fetch weather data using an online weather API and enable proactive actuation.

1.5 Scope

The project focuses on developing a compact, single-node ESP32-based system installed on the terrace to automatically control a protective shed. It detects environmental conditions such as rain and ambient light using onboard sensors and responds by activating a DC motor through a motor driver. The system supports manual control through a physical override switch and remote access using the Blynk IoT platform. A weather API is integrated to enable proactive operation based on forecasted rain. An LCD display provides real-time status updates. The scope of the project includes both electronic hardware and software development, while the full mechanical shed deployment structure is considered beyond the current prototype phase.

Chapter 2 – Literature Review

The field of home automation has seen significant progress with the integration of IoT, enabling intelligent control, remote accessibility, and real-time environment monitoring. Recent innovations focus on automating everyday tasks in households—especially weather-responsive systems—by integrating sensors, microcontrollers, and mobile applications. This project builds on these foundations by exploring how IoT technologies, sensor data, and cloud connectivity can automate the deployment of a terrace shed.

2.1 Smart Weather Detection Systems

Rain and light sensors are commonly used in smart irrigation, greenhouse automation, and rooftop protection systems. Studies have shown that combining **rain sensors** with **light- dependent resistors (LDRs)** provides reliable environmental sensing. These low-cost and simple sensors are well-suited for microcontroller-based automation where responsiveness and affordability are priorities.

2.2 IoT-Based Control Using Blynk

The **Blynk IoT platform** is widely used in home automation projects for remote device control and status monitoring. Blynk offers mobile app integration, virtual switches, and real-time feedback, enabling users to control systems such as fans, sprinklers, or blinds from anywhere. Its compatibility with ESP32 and ease of use make it ideal for DIY and academic automation projects.

2.3 Weather Forecast Integration

Integrating **weather APIs** allows IoT systems to act proactively based on forecasted weather conditions. Platforms like **Tomorrow.io** provide structured weather data (e.g., rain probability, humidity, temperature) via REST APIs. When paired with ESP32 and Wi-Fi connectivity, this data can enhance the system's ability to respond to conditions not yet sensed locally.

2.4 Motorized Automation Using ESP32 and L298N

Motorized systems are commonly used in smart home projects like curtain openers, drying racks, and door automation. The **L298N motor driver**, combined with a **DC motor**, enables controlled movement based on environmental triggers or user input. In this project, such a setupis used to automate the rolling of a protective terrace shed.

2.5 Manual Override and LCD Feedback

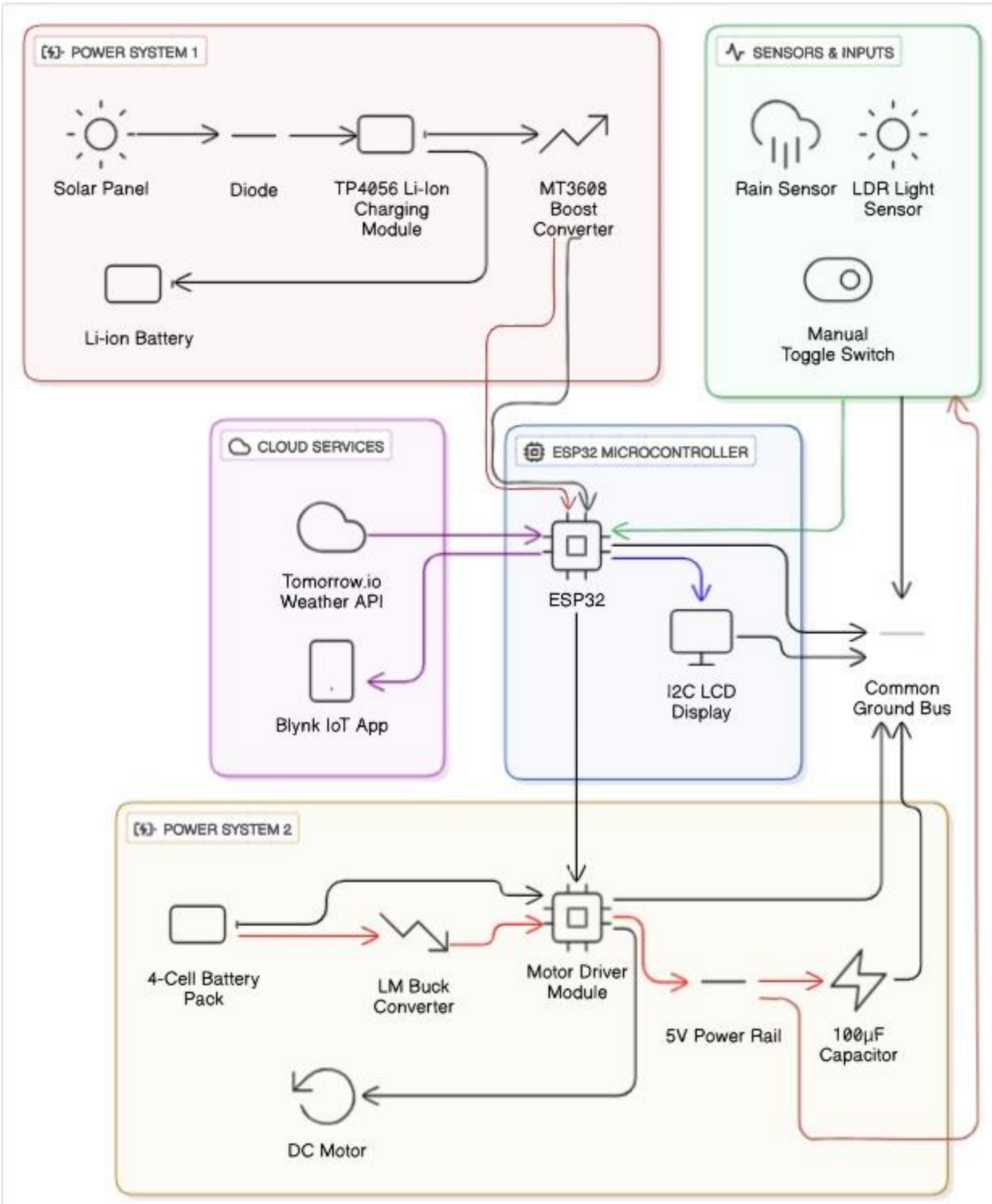
Manual override mechanisms are essential for user safety and control in automated systems. Simple tactile switches allow users to bypass automatic logic when needed. In addition, **LCD displays** (especially I2C-based 16x2 modules) are frequently used to provide users with real- time sensor readings and system status, making the system more user-friendly and informative.

Conclusion

The reviewed literature validates the feasibility of combining **sensor-based automation**, **motor control**, **weather forecast integration**, and **IoT platforms like Blynk** to build intelligent home automation systems. This project adapts these approaches into a **practical, single-ESP32-based terrace shed protection system**, offering responsive, remote- controllable, and cost-effective automation for real-world residential needs.

Chapter 3 – System Design and Architecture

3.1 Block Diagram



3.2 Components Selection

Component	Quantity	Specification	Purpose
ESP32 Dev Module	1	Wi-Fi, GPIO	Main microcontroller for sensing, control, and Blynk
Rain Sensor Module	1	LM393 Comparator, digital output	Detects rainfall on the terrace
LDR Module	1	Photodiode, digital output	Detects ambient light for day/night detection
DC Motor (6V)	1	Brushed DC Motor	Operates the shed mechanism
Li-ion Battery	4	3.7V, 2200–3000mAh	Powers the motor via regulated output
L298N Motor Driver	1	Dual H-Bridge, 5V logic, 6V motor support	Drives the DC motor
Capacitor	1	100µF or higher	Stabilizes voltage during motor startup
MT3608 Booster	2	Step-up to 5V or 12V and 3.7V to 5V	Power ESP32/sensors (terrace), 12V boost (flat)
70×70 Solar Panel	1	~5.5V, 1W	Terrace power input
LM2596 Buck Converter	1	Step-down voltage regulator, adjustable output	Steps down 12V or higher input to 5V for ESP32 or other modules
LCD Display (I2C)	1	16×2, I2C interface	Displays status and weather information
Manual Override Switch	1	Tactile push-button	Enables manual override of the system
ON/OFF Switch	1	SPST or Rocker Switch	Turns system power on or off
1N5819 Diode	1	Schottky Diode, reverse protection	Protects booster or charging circuit
Breadboards, Wires	—	Jumper wires, solderless breadboards	Circuit assembly

3.3 Working Principle

The IoT-based smart clothes drying protection system is designed to automatically and manually operate a motorized protective shed installed on a terrace or balcony to prevent clothes from getting wet due to rain. At the heart of the system lies the **ESP32 microcontroller**, which serves as the central processing unit, continuously monitoring environmental parameters, executing control logic, and handling both manual and remote user interactions.

Power Supply

The system is powered using **rechargeable 18650 Li-ion batteries**, supported by a **solar panel** as the primary energy source. Power is distributed across two distinct paths:

- **ESP32 and Control Circuitry:** A **solar panel** charges a single 18650 cell through a **1N5819 diode** and a **TP4056 charging module**. The battery output (~3.7V) is fed to a **DC-DC boost converter (MT3608)**, which steps up the voltage to **5V–6V** and powers the **ESP32 via its VIN pin**.
- **Motors and Peripherals:** A separate pack of **4x 18650 Li-ion cells** supplies higher voltage (up to ~16.8V) to a **DC-DC buck converter (LM2596)**, which steps the voltage down to **5V**. This output is routed through a **manual switch** to power the **L298N motor driver**. The **5V output pin of the L298N** is used to power **peripherals** such as sensors (rain, LDR), LCD, and buttons.

To ensure stability, especially during motor startup, a **100 µF electrolytic capacitor** is connected across the **5V output of the motor driver**, preventing voltage dips that could reset the ESP32 or sensors.

A **manual ON/OFF switch** is provided between the LM2596 and motor driver to cut power to the entire actuator side, offering safety and energy efficiency during idle or maintenance periods.

Sensor Inputs

The ESP32 receives real-time inputs from multiple sensors:

- **Rain Sensor**: A combined digital and analog rain sensor detects the presence of rainfall and provides an analog signal proportional to rain intensity. The digital pin helps in immediate detection (e.g., rain started), while the analog pin provides a percentage (0%–100%) indicating the amount of rainfall.
- **LDR (Light Dependent Resistor)**: This sensor is used to measure ambient brightness. It ensures the system does not open the shed during nighttime or in dark conditions.
- **Weather API (Tomorrow.io)**: Every 10 minutes, the ESP32 fetches weather data from the cloud using HTTP requests. This API provides real-time updates like precipitation probability, weather condition, and rain forecasts. This allows the system to take predictive actions before rain starts.

Decision Logic

The system operates in two modes: **Auto Mode** and **Manual Mode**

Auto Mode

This is the default smart mode where the shed is controlled based on environmental inputs. Auto Mode is only active during a user-defined time window (for example, between 6:00 AM and 8:00 PM), which prevents unnecessary operation during sleep hours.

The shed closes if:

- Rain is detected by the rain sensor

The shed opens if:

- Rain is no longer detected
- It is **bright enough** (checked using the LDR)

This ensures that the shed opens **only during the day** and **only when the weather is clear**.

False Trigger Prevention

To avoid unnecessary operation due to sensor error, the system includes a delay mechanism. If the **rain sensor alone** detects rain but the **weather API does not predict rain**, the system waits for **5 seconds** before taking action. If rain continues after this delay, the shed is closed. If the signal was false or glitchy, the system avoids acting, improving reliability.

User Interface & Control

Users can interact with the system using two main interfaces:

Physical Switch (Manual Override)

This is a tactile switch installed near the shed. When pressed, it **immediately closes the shed** and bypasses the automatic logic. This is useful in emergency situations or if the user wants instant manual control.

Blynk App

The Blynk IoT mobile application provides the following features:

- **Toggle between Auto and Manual mode**
- **Remotely open or close** the shed
- **Monitor** current weather, rain intensity, LDR status, and system mode
- **Set the active automation hours** (time window during which Auto Mode works)

LCD Display (16x2 I2C)

The system includes a **16x2 character LCD display with I2C module**, which shows live system status. The display updates once per second and provides the following information:

- **Current weather condition** (from API)
- **Shed status** (Open/Closed)
- **System mode** (Auto or Manual)
- **Rain presence and intensity** (0–100%)
- **Error messages** (e.g., “WiFi Lost”, “Blynk Lost”) in case of connectivity issues

Cloud Logging & Reliability

Google Sheets Integration

All major system events (e.g., shed closed/opened, rain started, manual override used) are **logged to Google Sheets** with a timestamp. This allows users to review the system’s historical behavior, identify patterns, or troubleshoot issues.

Chapter 4 – Hardware and Software Implementation

4.1 Circuit Details

1. Power Supply Circuit : The system uses two separate power paths: one for the ESP32 and control logic, and another for the motor driver and sensors.

Power Supply for ESP32 (Control Unit)

Component	Connection	Purpose / Notes
Solar Panel (5V)	→ 1N5819 Diode → TP4056 IN+	Charges 1x 18650 Li-ion battery safely
TP4056 OUT+	→ MT3608 Boost Converter IN+	Supplies ~3.7V from battery to booster
MT3608 OUT+	→ ESP32 VIN	Boosts to ~5V–6V for ESP32 operation
ESP32 GND	← Common GND	Connected to booster GND and L298N GND

Purpose: Ensures ESP32 is solar-powered with battery backup and stable boosted voltage.

B. Power Supply for Motor & Sensors

Component	Connection	Purpose / Notes
4x 18650 Battery Pack	→ LM2596 Step-Down IN	Provides higher voltage (~14.8V) for motor operations
LM2596 OUT (5V)	→ Power Switch → L298N Motor Driver VCC	Stepped down to 5V for L298N input and switched control
L298N GND	Shared with ESP32 and LM2596	Common ground reference
L298N 5V OUT	→ Power Rail	Powers sensors (Rain, LDR), LCD, buttons, etc.
Capacitor (100µF)	Across L298N 5V OUT	Prevents voltage drops during motor startup

Purpose: Isolates motor load from ESP32, supplies stable 5V to all peripherals, and allows manual switching.

Sensor Connections to ESP32

Sensor/Component	ESP32 Pin	Type	Description
Rain Sensor (D0 pin)	GPIO 34	Digital	Detects rain presence (0 = rain)
Rain Sensor (A0 pin)	GPIO 32	Analog	Measures rain intensity (mapped to 0–100%)
LDR Sensor (D0 pin)	GPIO 35	Digital	Detects brightness (0 = bright)

Manual Override Switch	GPIO 25	Digital	Used to manually close the shed on button press
LCD Display (I2C)	SDA: GPIO 21 SCL: GPIO 22	I2C	Shows weather, shed state, rain/light status, etc.

Motor Control via L298N

Component	ESP32 Pins	L298N Pins	Purpose
DC Motor	GPIO 26 & 27	IN1 & IN2	Controls direction for shed open/close

Software Implementation

The software for the Smart Rain Shed Controller is developed using the Arduino IDE for the ESP32 microcontroller, integrating sensor input, cloud communication, real-time automation, and a user-friendly interface via the Blynk IoT platform.

Development Platform

- **Microcontroller: ESP32 (WiFi-enabled)**
- **Programming Environment: Arduino IDE**
- **Programming Language: Embedded C++**

Library	**Purpose**
WiFi.h	**Connect ESP32 to local Wi-Fi network**
BlynkSimpleEsp32.h	**Interface with Blynk IoT platform**
HTTPClient.h	**Fetch HTTP data from Tomorrow.io weather API**
ArduinoJson.h	**Parse JSON responses from the API**
Wire.h	**Communicate with I2C devices (LCD)**
LiquidCrystal_I2C.h	**Drive the 16x2 LCD module**
time.h	**Get system time for scheduling operations**

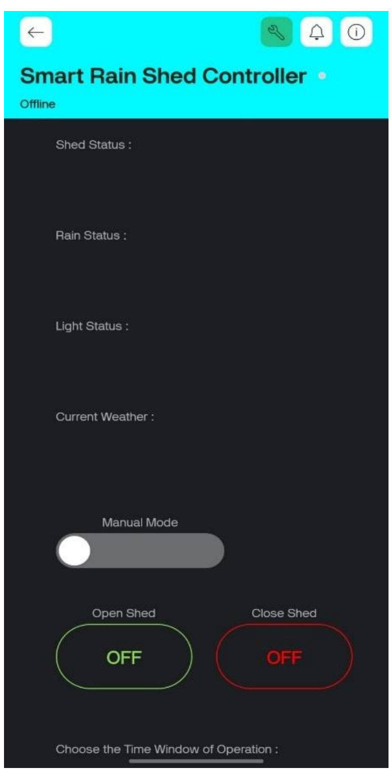
Core Features

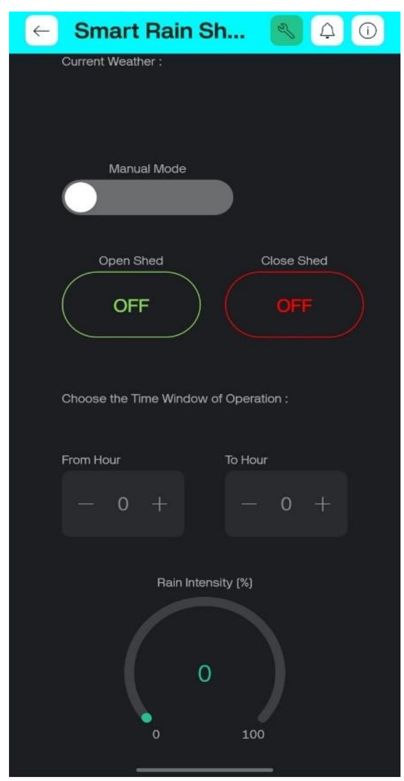
1. Mode Handling Logic
   - Auto Mode: Fully autonomous shed operation using sensor and weather data.
   - Manual Mode: User overrides automatic behavior via Blynk app or physical button.
2. Sensor Data Acquisition
   - Rain Sensor (Analog + Digital): Measures rainfall presence and intensity.
   - LDR (Analog): Detects ambient brightness to support time-based decision making.

3. Actuator Control
    - Motor direction is controlled via two GPIO pins connected to an L298N motor driver.
    - Logic ensures precise one-time shed opening or closing actions, preventing jitter or repeated toggling.
4. LCD Display Output
    - A 16x2 I2C LCD continuously shows:
        - Weather condition
        - Rain status
        - Light condition
        - Shed state (Open/Closed)
5. Weather API Integration
    - Uses Tomorrow.io API to fetch real-time weather codes every 10 minutes.
    - Weather codes like “Clear”, “Light Rain”, or “Thunderstorm” are interpreted to validate rain sensor readings and reduce false positives.
6. Cloud Logging with Google Sheets
    - Webhook requests log events such as:
        - Rain detection
        - Shed state changes
        - Light condition
        - Control mode
    - Data is stored in real-time for debugging and performance tracking.
7. Time-Window Scheduling
    - Auto mode operates only between startHour (V7) and endHour (V8).
    - Prevents shed movement at night or during inactive hours.
8. Debounce and Delay Handling
    - Adds a 5-second validation period before acting on rain input.
    - Ensures decision stability and avoids unnecessary motor activity.
9. Offline Resilience
    - Even during Wi-Fi or cloud failure, the system retains:
        - Manual button functionality
        - Sensor-driven shed control
        - LCD status output

Blynk Virtual Pin Configuration

Virtual Pin	Purpose	Explanation
V0	Manual/Auto Mode Toggle	Switch between **Auto** (sensor-based) and **Manual** (user-controlled) modes.
V1	Shed Status (Open/Closed)	Displays the current position of the shed in **real-time**.
V2	Rain Status	Shows if rain is detected by the **digital rain sensor**.
V3	Light Status	Indicates whether it's **Bright** or **Dark**, based on LDR input.
V4	Manual Open Button	Opens the shed **only in Manual mode**.
V5	Manual Close Button	Closes the shed **only in Manual mode**.
V6	Weather API Status	Displays real-time weather from **Tomorrow.io** (e.g., Clear, Rain).
V7	Auto Mode Start Hour (0–23)	User-defined **start time** for automatic control.
V8	Auto Mode End Hour (0–23)	User-defined **end time** for automatic control.
V9	Rain Intensity (0–100%)	Live analog rain sensor reading, used to measure **rainfall strength**.





Chapter 5 – Experimental Setup

5.1 Testing Procedure

Power Circuit Testing

The system used a combination of a buck converter, the L298N motor driver's onboard voltage regulator, and a boost converter to power various subsystems.

Buck Converter + L298N Motor Driver

- The buck converter was configured to output 12V, which powered the L298N motor driver's VCC (motor supply).
- The L298N board's onboard 5V regulator provided a 5V output on its 5V pin, used to power auxiliary components.
- A 100µF capacitor was added across the 5V output from the motor driver to stabilize voltage during motor operations, preventing dips due to inrush current.
- Observations with a multimeter:
    - 12V line remained stable under motor load.
    - 5V line showed improved stability with capacitor — voltage dips during motor switching were significantly reduced.

MT3608 Boost Converter (ESP32 Power)

- A separate MT3608 boost converter took input from a single 3.7V Li-ion cell, boosting it to 5V.
- This output was fed to the ESP32's VIN pin.
- No brownouts or restarts occurred during WiFi operation, LCD updates, or sensor polling.

ESP32 Microcontroller

Powered via USB connection, the ESP32 was tested for GPIO pin control of the motor via the L298N driver, sensor input reading from the rain and LDR sensors, I2C communication with the LCD display, Wi-Fi connectivity for Blynk, and HTTP API fetch from Tomorrow.io. All these functions worked as intended during bench testing.

Rain Sensor

The digital output from GPIO 34 was used for basic rain detection, while the analog output on GPIO 36 captured rain intensity scaled from 0 to 100 percent. The sensor was manually exposed to water droplets to validate both digital threshold switching and analog values.

LDR Sensor

The LDR module was connected to GPIO 35. It reliably detected low-light conditions by going digital LOW when dark. Bright light returned the signal to HIGH, validating its use in daylight-dependent automation logic.

Manual Override Switch

Connected to GPIO 25, this physical push-button switch was pressed during various system states to ensure it correctly bypassed automatic logic and forced the shed to close. The switch was recognized immediately and worked as expected.

DC Motor & L298N Driver

Motor direction was controlled via IN1 and IN2 pins connected to GPIO 26 and GPIO 27, respectively. The ENA pin of the motor driver was connected directly to 5V, keeping it constantly enabled. The motor rotated smoothly in both directions during testing, with proper response to GPIO signals.

LCD Display (I2C)

The LCD was wired to SDA (GPIO 21) and SCL (GPIO 22) and successfully displayed system information including current operating mode (Auto or Manual), shed status (Open or Closed), weather data from the API, rain and light conditions, and connection status (e.g., Blynk or Wi-Fi disconnected). Text updates were clear and timely.

Subsystem Testing

Sensor Integration:

Rain sensor and LDR were tested under various combinations of environmental input. For example, when it was raining and bright, the system closed the shed. In bright, dry conditions, the shed was opened. When raining at night, the shed remained closed and did not reopen until conditions improved. These cases validated the logical combination of sensor inputs.

Motor Control Logic

Motor direction was confirmed by controlling GPIO 26 and GPIO 27. Setting GPIO 26 HIGH and GPIO 27 LOW rolled the shed down. Reversing the signals (GPIO 26 LOW and GPIO 27 HIGH) rolled the shed up. The motor stopped when both signals were LOW. This verified directional control and safe stop behavior.

Manual Override

The physical switch reliably overrode automatic logic. Upon activation, the LCD updated to show “MANUAL” mode with a “Manual Close Activated” message. This test confirmed that the manual control had immediate and consistent priority over auto logic.

Weather API

Tomorrow.io weather API was polled every 10 minutes. When weather conditions like “Rain” or “Light Rain” appeared in the API response, the shed closed in advance, even before physical rain was detected. This proactive behavior was verified through serial monitor logs and physical motor response.

Communication Testing

Blynk Integration (via Wi-Fi)

The ESP32 was connected to the Blynk IoT platform and all virtual pins (V0 to V9) were tested. These included mode toggle (V0), manual shed open/close controls (V4 and V5), shed status (V1), rain detection (V2), light detection (V3), API status (V6), auto mode start and stop hours (V7 and V8), and rain intensity percentage (V9). Every control and display component responded accurately.

Google Sheets Logging

System events like “Shed Opened”, “Shed Closed”, or “Rain Detected” were logged into Google Sheets every 10 minutes. These entries were checked remotely through the web and confirmed to be timestamped and accurate.

Wi-Fi Reliability

The ESP32 auto-reconnected after deliberate Wi-Fi disconnections, and the weather API resumed functioning without requiring manual reset. API data was parsed successfully using HTTPClient and ArduinoJson libraries.

Full System Integration

Once all modules were validated individually, the complete system was assembled and run under normal operating conditions. The ESP32 successfully monitored rain, light, time, and forecast to roll the shed up or down automatically. Manual override using the switch worked without interfering with future auto operations. LCD status was consistently accurate and up to date. Mobile app control through Blynk remained synced with local state. All sensor readings, shed states, and API statuses were logged to Google Sheets, and the system showed no delays, errors, or false triggers during repeated testing sessions

Prototype Power Testing

The ESP32 microcontroller was powered through a USB connection from a laptop or adapter, ensuring a stable logic voltage for sensor reading, Wi-Fi, and control logic. The 12V DC motor was powered separately using 4 3.7V Li-ion batteries from buck converter.

A DC-DC boost converter (MT3608) was used to step up the battery voltage to approximately 5V, which was then supplied to the ESP32 .

To maintain steady voltage during motor activation, a 100µF capacitor was connected. This helped smooth out any fluctuations caused by sudden current demands.

Voltage levels were monitored using a multimeter during testing and were confirmed to be within safe and functional ranges for all components.

Testing Version

To ensure safe and consistent testing without relying on unpredictable real-world weather conditions, a dedicated **Testing Mode** was implemented in the firmware. When enabled via the code (#define TEST_MODE true), the ESP32 bypassed sensor inputs and allowed **manual simulation through the Serial Monitor**.

- **Commands** such as "rain", "ldr bright", "ldr dark", and "intensity 75" could be typed into the Serial Monitor to emulate sensor conditions.
- This allowed us to verify how the system responded to various combinations of rain, light, and intensity levels—without requiring actual environmental changes.
- Testing Mode also displayed all sensor statuses, motor actions, shed state, and mode on the LCD and Blynk app in real-time, allowing full verification of **automatic logic**, **manual override**, and **cloud interactions**.

This feature significantly improved **development speed**, **debugging efficiency**, and **demonstration capabilities**, especially in controlled indoor environments.

Output Image

```
17:01:10.796 -> 🌧 Smart Rain Shed Controller [REAL SENSOR]
17:01:10.796 -> ==============================
17:01:10.796 -> Boot Reason: 1 (POWERON_RESET)
17:01:10.796 -> ----------------------------------
17:01:14.083 -> [BOOT] Shed state = OPEN
17:01:14.177 -> [WIFI] Connecting .... CONNECTED!
17:01:16.886 -> [BLYNK] Initialization complete.
17:01:16.886 -> [TIME] Syncing time ✅
17:01:19.110 ->
17:01:19.110 -> === [WEATHER API] FETCH ===
17:01:20.979 -> [WEATHER] Code: 1001 | Condition: Cloudy
17:01:20.979 -> [BLYNK] Weather Update: Cloudy
17:01:20.979 -> === SETUP DONE ===
17:01:20.979 ->
17:01:21.289 -> [LCD] Wx:Cloudy S:Op AUTO R:NoRn Int:0% L:Br
17:01:21.289 ->
17:01:21.289 -> ---- [STATUS SNAPSHOT] ----
17:01:21.289 -> Rain: No
17:01:21.289 -> Light: Bright
17:01:21.289 -> Mode: AUTO
17:01:21.289 -> Shed: OPEN
```

```
17:02:11.415 -> ---- [STATUS SNAPSHOT] ----
17:02:11.415 -> Rain: YES
17:02:11.415 -> Light: Bright
17:02:11.415 -> Mode: AUTO
17:02:11.415 -> Shed: OPEN
17:02:11.415 -> Rain %: 66
17:02:11.415 -> ----------------------------
17:02:11.415 ->
17:02:13.199 -> [LCD] Wx:Cloudy S:Op AUTO R:NoRn Int:37% L:Br
17:02:14.173 -> [LCD] Wx:Cloudy S:Op AUTO R:Rain Int:45% L:Br
17:02:15.045 -> [ACTION] 🚪 Shed Closing...
17:02:15.045 -> [FALSE_TRIG] Sensor steady 5s - Closing shed.
17:02:22.221 -> [LCD] Wx:Cloudy S:Cl AUTO R:Rain Int:45% L:Br
17:02:22.221 -> [BLYNK] Shed Closed via Auto
17:02:26.056 -> [ACTION] 🏠 Shed CLOSED!
17:02:26.056 ->
17:02:26.056 -> ---- [STATUS SNAPSHOT] ----
17:02:26.056 -> Rain: YES
17:02:26.096 -> Light: Bright
17:02:26.096 -> Mode: AUTO
17:02:26.096 -> Shed: CLOSED
17:02:26.096 -> Rain %: 43
17:02:26.096 -> ----------------------------
```

```
17:04:12.976 -> ---- [STATUS SNAPSHOT] ----
17:04:12.976 -> Rain: YES
17:04:12.976 -> Light: Dark
17:04:12.976 -> Mode: AUTO
17:04:13.071 -> Shed: CLOSED
17:04:13.071 -> Rain %: 73
17:04:13.071 -> ----------------------------
17:04:13.071 ->
17:04:22.168 -> [WDT] Loop slow: 3035 ms
17:04:25.104 ->
17:04:25.104 -> ---- [STATUS SNAPSHOT] ----
17:04:25.104 -> Rain: No
17:04:25.104 -> Light: Dark
17:04:25.104 -> Mode: AUTO
17:04:25.147 -> Shed: CLOSED
17:04:25.147 -> Rain %: 38
17:04:25.147 -> ----------------------------
17:04:25.147 ->
17:04:25.175 -> [LCD] Wx:Cloudy S:Cl AUTO R:NoRn Int:33% L:Dr
17:04:29.934 ->
17:04:29.934 -> ---- [STATUS SNAPSHOT] ----
17:04:29.934 -> Rain: No
17:04:29.934 -> Light: Bright
17:04:29.934 -> Mode: AUTO
```

```
17:04:25.175 -> [LCD] Wx:Cloudy S:Cl AUTO R:NoRn Int:33% L:Dr
17:04:29.934 ->
17:04:29.934 -> ---- [STATUS SNAPSHOT] ----
17:04:29.934 -> Rain: No
17:04:29.934 -> Light: Bright
17:04:29.934 -> Mode: AUTO
17:04:29.934 -> Shed: CLOSED
17:04:29.934 -> Rain %: 0
17:04:29.934 -> ----------------------------
17:04:29.934 ->
17:04:30.215 -> [ACTION] 🚪 Shed Opening...
17:04:30.215 -> [AUTO] ✅ Bright/No Rain, Shed Opening!
17:04:35.241 -> [LCD] Wx:Cloudy S:Op AUTO R:NoRn Int:0% L:Br
17:04:35.241 -> [BLYNK] Shed Opened via Auto
17:04:38.134 -> [ACTION] 🏠 Shed OPENED!
17:04:38.134 ->
17:04:38.134 -> ---- [STATUS SNAPSHOT] ----
17:04:38.134 -> Rain: No
17:04:38.180 -> Light: Bright
17:04:38.180 -> Mode: AUTO
17:04:38.180 -> Shed: OPEN
17:04:38.180 -> Rain %: 0
17:04:38.180 -> ----------------------------
```

```
17:04:38.134 -> ---- [STATUS SNAPSHOT] ----
17:04:38.134 -> Rain: No
17:04:38.180 -> Light: Bright
17:04:38.180 -> Mode: AUTO
17:04:38.180 -> Shed: OPEN
17:04:38.180 -> Rain %: 0
17:04:38.180 -> ----------------------------
17:04:38.180 ->
17:04:38.180 -> [WDT] Loop slow: 2962 ms
17:04:38.269 -> [LCD] Wx:Cloudy S:Op AUTO R:NoRn Int:0% L:Br
17:04:47.421 -> [ACTION] 🚪 Shed Closing...
17:04:47.421 -> [BLYNK] Switch closed shed
17:04:50.515 -> [SWITCH] Shed Closed (by switch)
17:04:50.515 -> [WDT] Loop slow: 3053 ms
17:04:53.779 -> [LCD] Wx:Cloudy S:Cl AUTO R:NoRn Int:0% L:Br
17:04:53.779 -> [BLYNK] Shed Closed via Auto
17:04:58.610 -> [ACTION] 🏠 Shed CLOSED!
17:04:58.610 ->
17:04:58.610 -> ---- [STATUS SNAPSHOT] ----
17:04:58.610 -> Rain: No
17:04:58.610 -> Light: Bright
17:04:58.610 -> Mode: AUTO
17:04:58.610 -> Shed: CLOSED
17:04:58.610 -> Rain %: 0
```

```
17:04:58.610 -> ---- [STATUS SNAPSHOT] ----
17:04:58.610 -> Rain: No
17:04:58.610 -> Light: Bright
17:04:58.610 -> Mode: AUTO
17:04:58.610 -> Shed: CLOSED
17:04:58.610 -> Rain %: 0
17:04:58.610 -> ----------------------------
17:04:58.653 ->
17:04:58.653 -> [WDT] Loop slow: 8140 ms
17:04:58.739 -> [LCD] Wx:Cloudy S:Cl AUTO R:NoRn Int:0% L:Br
17:05:07.433 -> [ACTION] 🚪 Shed Opening...
17:05:07.433 -> [AUTO] ✅ Bright/No Rain, Shed Opening!
17:05:12.448 -> [LCD] Wx:Cloudy S:Op AUTO R:NoRn Int:0% L:Br
17:05:12.449 -> [BLYNK] Shed Opened via Auto
17:05:15.051 -> [ACTION] 🏠 Shed OPENED!
17:05:15.051 ->
17:05:15.051 -> ---- [STATUS SNAPSHOT] ----
17:05:15.051 -> Rain: No
17:05:15.051 -> Light: Bright
17:05:15.051 -> Mode: AUTO
17:05:15.051 -> Shed: OPEN
17:05:15.051 -> Rain %: 0
17:05:15.051 -> ----------------------------
```

```
17:05:29.540 -> [BLYNK] Manual Mode toggled ON
17:05:29.540 ->
17:05:29.540 -> ---- [STATUS SNAPSHOT] ----
17:05:29.540 -> Rain: No
17:05:29.540 -> Light: Bright
17:05:29.540 -> Mode: MANUAL
17:05:29.540 -> Shed: OPEN
17:05:29.540 -> Rain %: 0
17:05:29.540 -> ---------------------------
17:05:29.615 ->
17:05:30.219 -> [LCD] Wx:Cloudy S:Op MAN  R:NoRn Int:0% L:Br
17:05:37.532 -> [BLYNK] Manual Close button pressed
17:05:37.532 -> [ACTION] 🚪 Shed Closing...
17:05:42.614 -> [LCD] Wx:Cloudy S:Cl MAN  R:NoRn Int:0% L:Br
17:05:42.614 -> [BLYNK] Shed Closed via Manual
17:05:45.543 -> [ACTION] 🏠 Shed CLOSED!
17:05:45.543 ->
17:05:45.543 -> ---- [STATUS SNAPSHOT] ----
17:05:45.543 -> Rain: No
17:05:45.543 -> Light: Bright
17:05:45.543 -> Mode: MANUAL
17:05:45.543 -> Shed: CLOSED
17:05:45.543 -> Rain %: 0
17:05:45.543 -> ---------------------------
```

```
17:05:29.615 ->
17:05:30.219 -> [LCD] Wx:Cloudy S:Op MAN  R:NoRn Int:0% L:Br
17:05:37.532 -> [BLYNK] Manual Close button pressed
17:05:37.532 -> [ACTION] 🚪 Shed Closing...
17:05:42.614 -> [LCD] Wx:Cloudy S:Cl MAN  R:NoRn Int:0% L:Br
17:05:42.614 -> [BLYNK] Shed Closed via Manual
17:05:45.543 -> [ACTION] 🏠 Shed CLOSED!
17:05:45.543 ->
17:05:45.543 -> ---- [STATUS SNAPSHOT] ----
17:05:45.543 -> Rain: No
17:05:45.543 -> Light: Bright
17:05:45.543 -> Mode: MANUAL
17:05:45.543 -> Shed: CLOSED
17:05:45.543 -> Rain %: 0
17:05:45.543 -> ---------------------------
17:05:45.543 ->
17:05:45.543 -> [WDT] Loop slow: 2987 ms
17:05:45.683 -> [LCD] Wx:Cloudy S:Cl MAN  R:NoRn Int:0% L:Br
17:05:53.636 -> [WDT] Loop slow: 4493 ms
17:05:58.750 -> [BLYNK] End Hour set to: 16
17:05:59.429 -> [BLYNK] Start Hour set to: 8
17:06:00.477 -> [BLYNK] End Hour set to: 17
17:06:00.847 -> [BLYNK] End Hour set to: 18
17:06:01.014 -> [BLYNK] End Hour set to: 19
```

```
17:06:15.636 -> [BLYNK] Manual Mode toggled OFF
17:06:15.636 -> [ACTION] 🚪 Shed Opening...
17:06:15.636 -> [AUTO] ✅ Bright/No Rain, Shed Opening!
17:06:15.636 ->
17:06:15.636 -> ---- [STATUS SNAPSHOT] ----
17:06:15.636 -> Rain: No
17:06:15.636 -> Light: Bright
17:06:15.636 -> Mode: AUTO
17:06:15.636 -> Shed: CLOSED
17:06:15.636 -> Rain %: 0
17:06:15.636 -> ---------------------------
17:06:15.636 ->
17:06:16.175 -> [LCD] Wx:Cloudy S:Cl AUTO R:NoRn Int:0% L:Br
17:06:22.092 -> [LCD] Wx:Cloudy S:Op AUTO R:NoRn Int:0% L:Br
17:06:22.092 -> [BLYNK] Shed Opened via Auto
17:06:25.428 -> [ACTION] 🏠 Shed OPENED!
17:06:25.428 ->
17:06:25.428 -> ---- [STATUS SNAPSHOT] ----
17:06:25.428 -> Rain: No
17:06:25.428 -> Light: Bright
17:06:25.428 -> Mode: AUTO
17:06:25.428 -> Shed: OPEN
17:06:25.428 -> Rain %: 0
17:06:25.428 -> ---------------------------
```

Chapter 6 – Results & Discussion

6.1 Interpretation of results

The implemented smart rain shed controller consistently responded to environmental conditions as expected. Rain sensors and LDR accurately detected changes, with system actions aligning correctly under various scenarios. The integration of the Tomorrow.io Weather API significantly improved reliability by validating sensor readings, especially in borderline conditions, thereby minimizing false triggers.

During testing, the manual override mechanism—both via physical switch and Blynk app—worked without lag or error, confirming robust fallback control. The LCD interface updated status information in real-time, improving usability and transparency.

The solar-powered setup, consisting of TP4056, MT3608/LM2596, and Li-ion batteries, provided stable power throughout prolonged operations. Minor voltage dips were observed only at low battery levels, which did not affect core functionality. Furthermore, Google Sheets logging successfully recorded sensor values, shed state, and timestamps, confirming accurate data capture and cloud integration.

The system's time-based automation prevented unnecessary motor activity during inactive hours, thereby enhancing energy efficiency. The motor driver performed with no overheating or jittering due to the implemented cooldown logic, contributing to hardware longevity.

Overall, the results indicate that the system is functionally sound, energy-efficient, and user-friendly, validating the design choices made and confirming its feasibility for real-world urban residential use.

6.2 Key Features

1. **Automatic Rain Shed Control**

   Auto Mode: The shed opens or closes automatically based on rain, light, and time conditions, along with weather forecast data from Tomorrow.io.

   Manual Override: The user can control the shed manually using a physical button or the Blynk IoT app.

2. **Dual Rain Detection**

   Digital Rain Sensor: Detects simple rain presence (ON/OFF).

   Analog Rain Sensor: Measures rain intensity (0–100%) for precise, context-aware responses.

3. **Light (LDR) Integration**

   The system uses an LDR sensor to check ambient brightness. The shed does not open automatically at night, even if it’s not raining.

4. **Time-Window Auto Control**

   Users can define the auto-mode active period using Blynk app sliders (V7 and V8). This prevents unwanted operation outside preferred hours.

5. **Blynk IoT Remote App**

   A complete interface for remote monitoring and control. The app displays shed status, rain/light conditions, live weather code, and allows manual/automatic control toggling.

6. **LCD User Interface**

   A 16x2 I2C LCD continuously displays real-time data such as rain intensity, light level, shed state, mode, Wi-Fi status, and weather conditions in simple language.

7. **Robust Motor Control via State Machine**

   Prevents redundant toggles and double activations.

   Ensures motor runs only for a defined timeout, avoiding overrun.

   Physical, auto, and app commands all go through a safe centralized logic.

8. **Rain Sensor False-Trigger Immunity**

   Non-blocking debounce logic ensures that the shed only responds to sustained rain (minimum 5 seconds), filtering out false spikes or sensor noise.

9. **Cloud Weather Integration**

   The system fetches live weather forecast codes every 10 minutes from Tomorrow.io, enabling proactive control and enhancing decision-making.

10. **Google Sheets Logging**

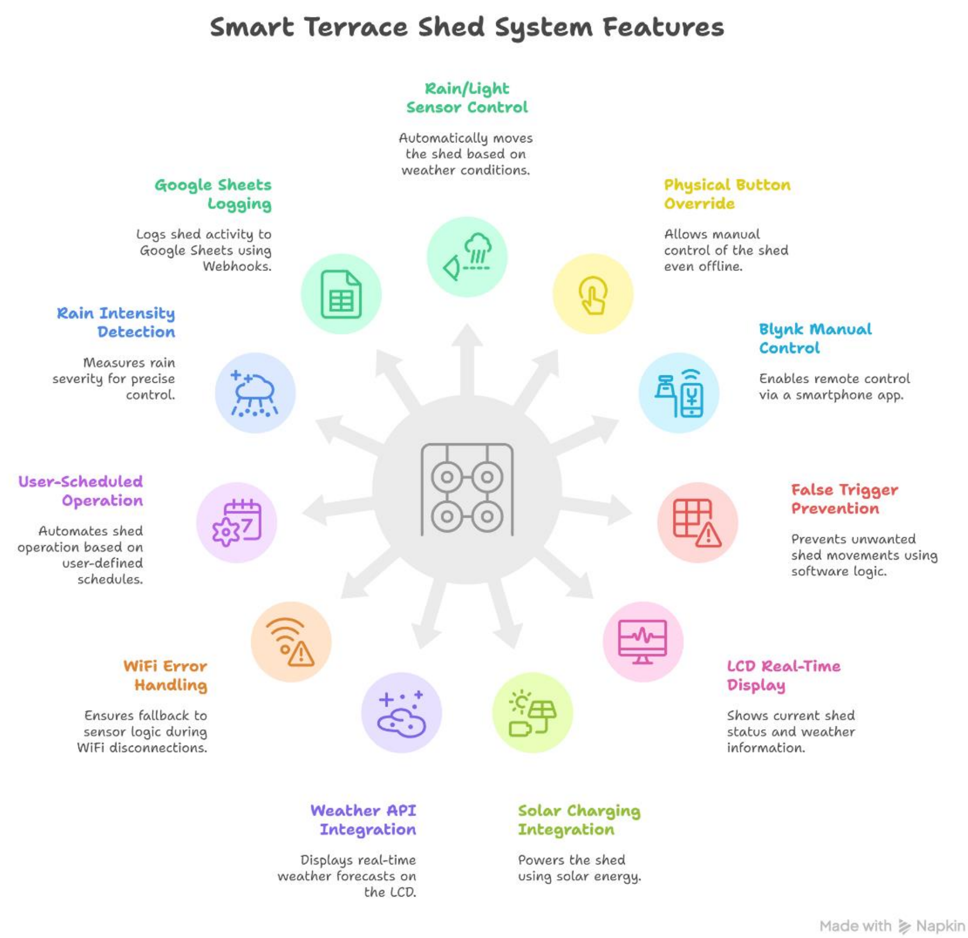
    Every event and sensor change is logged with a timestamp to Google Sheets for transparency, debugging, or long-term analysis.

11. **Reliability – Watchdog Timer**

    A hardware watchdog timer is enabled on the ESP32. If the system hangs or crashes, the ESP32 auto-reboots to restore functionality.

12. **Serial Debugging & Event Logging**

    All decisions, sensor readings, errors, and ignored conditions are printed to the serial monitor for local debugging and event tracking.



Chapter 7 – Conclusion

7.1 Conclusion

The IoT-Based Terrace Shed Protection System was successfully designed, developed, and implemented as a fully functional prototype, including both the mechanical andelectronic components. The system uses a single ESP32 microcontroller to monitor environmental conditions such as rainfall and ambient light through rain and LDR sensors. When rain is detected or forecasted via the integrated weather API, the system automatically activates a 6V motor via the L298N driver to roll down a mechanical shed, thereby protecting clothes and terrace space from getting wet.

Real-time status updates are displayed on an LCD, and users can also manually operate the system through a physical override switch. Additionally, Blynk IoT integration enables remote monitoring and control via a mobile app, enhancing convenience and accessibility.

All system components—including sensors, power modules, motor driver, LCD, and the complete physical shed structure—were successfully tested and validated under realistic operating conditions. The project demonstrates a practical, scalable, and cost-effective solution for automated terrace protection, aligning with smart home and weather-responsive automation trends.

This fully integrated system serves as a reliable smart automation solution for residential use and has the potential for further enhancements such as voice assistant support and battery monitoring.

Future Work:

The system can be enhanced by adding **over-the-air (OTA) updates**, **battery level monitoring**, and **smartphone notifications for rain alerts**. Future versions may also support **AI-based weather prediction**, a **dedicated mobile app**, and **multi-shed control** for larger installations. Integration with **home automation platforms** like Alexa or Google Home can further improve usability and accessibility.

7.2 Applications

1. Household Clothes Drying Automation

Use Case: Automates shed deployment to protect clothes in open spaces like terraces or balconies.

Problem Solved: Prevents clothes from getting wet during unexpected rain.

Benefits:

- Saves manual effort
- Ideal for working professionals, elderly, or large families

2. Smart Home Integration

Use Case: Can be linked with smart home platforms (e.g., Alexa, Google Home).

Benefits:

- Enables future voice control
- Syncs with weather dashboards and scheduling systems

3. Apartment Complexes & PG Accommodations

Use Case: Shared drying areas in hostels or apartments.

Benefits:

- Remote control by administrators
- Reduces disputes over wet clothes
- Enables user-level logging

4. Agricultural Mini-Shed Automation

Use Case: Protects drying crops or produce from rain (e.g., chillies, seeds).

Benefits:

- Prevents crop spoilage
- Works off-grid using solar power

5. Outdoor Gear / Tool Storage Protection

Use Case: Shields outdoor equipment like bikes or garden tools.

Benefits:

- Protects from rust or water damage
- Automatically retracts in dry conditions

6. Retail Shops and Street Vendors

Use Case: Covers goods of vendors from sudden rain.

Benefits:

- Reduces damage to merchandise
- Allows continued operation in rain
- Manual override option

7. Rain-Prone or Disaster-Prone Regions

Use Case: Ideal for locations with unpredictable or heavy rainfall.

Benefits:

- Minimizes reliance on human action
- Works efficiently with local weather APIs
-

8. Elderly / Differently-Abled Friendly Homes

Use Case: Designed for people with mobility issues.

Benefits:

- No need to manually operate the shed
- Fully automatic or app-based control
- Safer than accessing rooftops

9. Data Logging & Research Applications

Use Case: Real-time IoT data collection for labs or research.

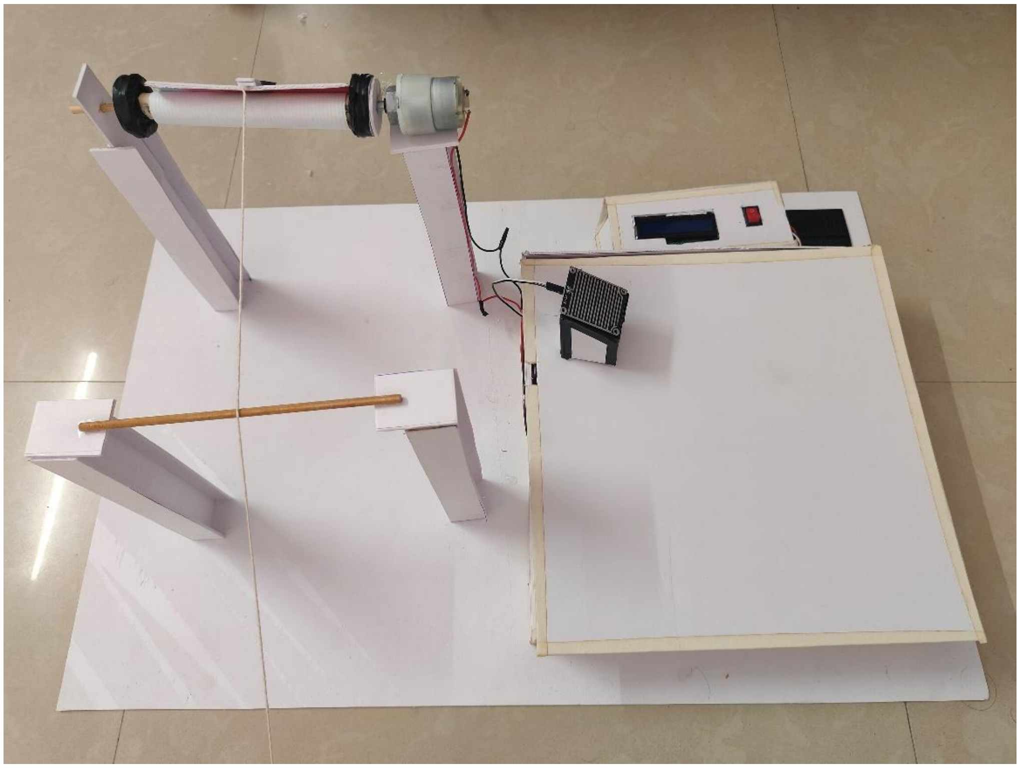
Benefits:

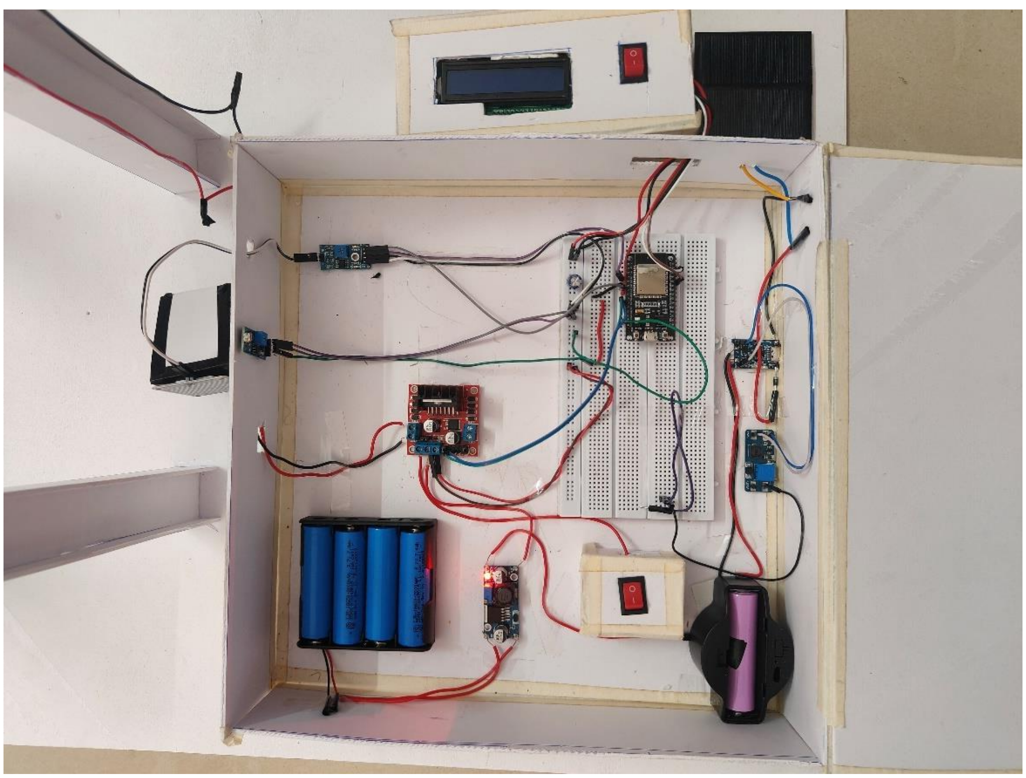
- Logs environmental changes for analysis
- Integrates with Google Sheets or Firebase
- Useful for testing weather-based logic

References

1. Rao, N., & Naik, A. (2020). *Automated window blind system using IoT*. *International Journal of Scientific Research in Engineering and Management, 4*(12), 1–5. https://doi.org/10.36282/IJSREM.2020.4212
2. Shinde, S., & Kadam, V. (2020). *IoT based smart home automation using Blynk*. *International Journal of Creative Research Thoughts, 8*(6), 3467–3471. https://doi.org/10.6084/m9.figshare.13023574
3. Kumar, A., & Verma, S. (2022). *Design and implementation of low-power communication using ESP-NOW protocol for smart agriculture*. *International Journal of Engineering Research & Technology, 11*(7), 231–234. https://doi.org/10.17577/IJERTV11IS070123
4. Singh, K., & Prasad, R. (2021). *Design and implementation of solar-powered IoT node for smart farming*. *International Journal of Engineering Research & Technology, 10*(6), 212–216. https://doi.org/10.17577/IJERTV10IS060134
5. Jayakrista, S., & Wowiling, R. G. (2024). *Prototype of IoT-based clothes drying control system using infrared thermometer MLX90614*. *Journal of Computer Engineering, Electronics and Information Technology (COELITE), 3*(1), 27–34. https://ejournal.upi.edu/index.php/COELITE/article/view/68360
6. Research on IoT-based smart clothes drying rack. (2024). *Proceedings of the ACM International Conference on Internet of Things*. https://doi.org/10.1145/3653081.3653127
7. Sari, N. P., & Sari, S. P. (2025). *The internet of things intelligent drying rack system*. *Indonesian Journal of Electrical Engineering and Computer Science, 38*(1), 99–106. https://doi.org/10.11591/ijeecs.v38.i1.pp99-106
8. Prasetyo, A., & Setiawan, D. (2024). *Monitoring and information system for automatic clothes drying based on IoT*. *PIKSEL: Penelitian Ilmu Komputer Sistem Embedded dan Logic, 12*(3), 9828–3331. https://jurnal.unismabekasi.ac.id/index.php/piksel/article/view/9828
9. Kusuma, D. S., & Prabowo, H. (2024). *Clothesline smart device design based on IoT device*. *Journal of Information Technology and Computer Science Education (JITCSE), 2*(1), 21–29. https://ysmk.org/ejournal/index.php/jitcse/article/view/21

Appendices





ESP32 Code :

https://github.com/Bhavya-Chawat/Smart-Rain-Shed-Controller/blob/main/code/Code.ino

Google Sheets data :

https://docs.google.com/spreadsheets/d/1R-LVrv1EYpHeQtYyeMxDi2P7ZqYr3wQ-Kpr4Besj38I/edit?gid=0#gid=0

Demonstration Video:

COST SHEET

Component	Qty	Specification	Purpose	Unit Cost (₹)	Total Cost (₹)
ESP32 Dev Module	1	Wi-Fi, BLE, GPIO	Main microcontroller for sensing, control, and Blynk	350	350
Rain Sensor Module	1	LM393 comparator, digital & analog output	Detects rainfall presence and intensity	70	70
LDR Module	1	Photodiode, digital output	Detects ambient brightness (day/night)	30	30
DC Motor (12V Geared)	1	Brushed DC motor	Drives the shed opening/closing mechanism	180	180
L298N Motor Driver	1	Dual H-Bridge, 5V logic	Drives DC motor based on ESP32 logic	140	140
18650 Li-ion Battery	4	3.7V, 2200–3000mAh	Primary power source for the system	200	800
Battery Holder	1	4-cell 18650 holder	Holds batteries securely	40	40
TP4056 Charging Module	1	1A Li-ion charger with protection	Safely charges 18650 batteries from solar	50	50
MT3608 Boost Converter	2	Step-up 3.7V to 5V / 12V	Powers ESP32 and drives motor voltage	60	120
LM2596 Buck Converter	1	Step-down 12V to 5V	Powers 5V components from motor rail	90	90
Solar Panel (70×70mm)	1	~5.5V, 1W	Harvests solar energy for charging	200	200
Capacitor	1	100µF or above	Stabilizes power during motor surge	10	10
LCD Display (I2C)	1	16×2 with I2C backpack	Displays shed status, rain/light condition, weather	150	150
Manual Override Button	1	SPST switch	Allows physical override of shed	10	10
Power ON/OFF Switch	1	SPST switch	Turns system ON or OFF	10	10
1N5819 Diode	1	Schottky diode	Prevents back current from solar to TP4056	5	5
Breadboard + Jumpers	1 set	Solderless board and jumper wires	For temporary circuit prototyping	275	275