**IOT Product and Design**

**Smart Pill Dispensor – Project Report**

**Submitted to**

**AMITY UNIVERSITY UTTAR PARDESH**

**A blue and yellow logo

AI-generated content may be incorrect.**

**In partial fulfilment of the requirements for the award of the degree of Bachelor of technology**

**In**

**Information Technology**

**By**

**Shagun Yadav**

**(A2305322007)**

**Submitted to:**

**Dr Ashish Mani**

**DEPARTMENT OF AMITY SCHOOL OF ENGINEERING**

**AMITY UNIVERSITY UTTAR PARDESH**

**NOIDA (U.P.)**

**1. Abstract**

The Smart Pill Dispenser is an IoT-based automation system designed to dispense medication based on user-defined schedules through a web interface. It connects to a local WiFi network, hosts a web server, and controls an MG995 servo motor to release pills at specific times. The goal is to assist users in managing their medication without the need for manual intervention. The interface allows users to set the days and times of pill dispensing, providing reliability for patients with time-sensitive medical routines. This report outlines the design, implementation, testing, and future scope of the system.

**2. Introduction**

Modern lifestyles and age-related health issues often result in people forgetting or delaying their medication. This poses a risk especially to elderly individuals, patients with chronic illnesses, or those requiring regular doses at specific times.

To address this, we propose the Smart Pill Dispenser — a compact, IoT-enabled device using the ESP8266 NodeMCU and a high-torque MG995 servo motor. Unlike traditional dispensers, this system allows scheduling pill dispensing at precise times on specific days through a web-based interface. The interface is accessible from any device connected to the same local WiFi, removing the need for mobile apps or cloud platforms.

By offering time-based automation, local control, and web configuration, this system bridges convenience with health reliability, ensuring users never miss a dose. The project is scalable and serves as a strong foundation for future features like alerts, logs, or remote monitoring

A box with wires and wires

AI-generated content may be incorrect.

**Figure 3**: Hardware Setup (ESP8266 + Servo Motor)

**3. Literature Review**

Existing Solutions:

* Manual Dispensers: These rely entirely on the user to remember their medication schedule, which is prone to human error.
* Raspberry Pi & Arduino RTC-Based Solutions: Some solutions use external modules for scheduling and cloud-based platforms, increasing cost and complexity.
* Commercial Smart Dispensers: These may come with apps and subscription services, making them expensive and less accessible.

Challenges in Current Systems:

* Complex UI / Overengineering
* Dependency on cloud services
* High cost
* Lack of simple scheduling mechanisms

Our Approach:

* NodeMCU ESP8266 for inbuilt WiFi
* MG995 servo motor for precise dispensing
* Web interface to schedule pill timings
* No cloud/app dependency
* Time-based automation using software scheduling logic

**4. Methodology**

Hardware Used

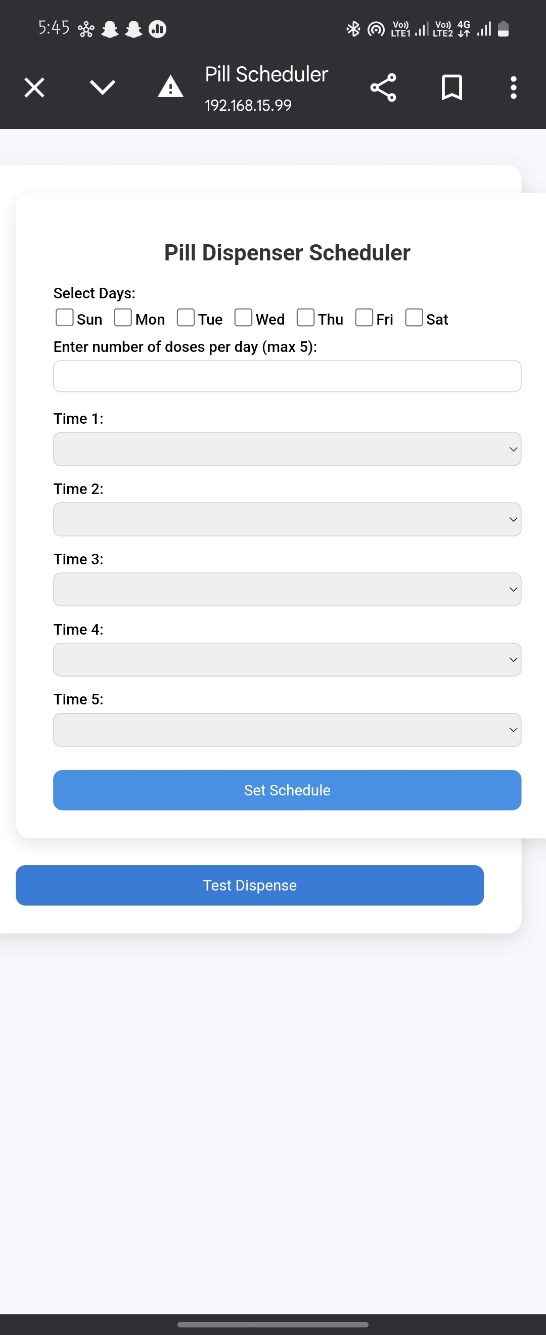
* ESP8266 NodeMCU – Serves as the WiFi host and main controller
* MG995 Servo Motor – Rotates a dispensing mechanism to release pills
* Power Supply – 5V power from USB/power bank
* Container with Pill Slots – Rotated by servo to align pills with the outlet

Software and Libraries

* Arduino IDE – For development and code upload
* Libraries:
  + ESP8266WiFi.h – Network connectivity
  + ESP8266WebServer.h – Hosts the web server
  + Servo.h – Controls MG995 motor
  + time.h – Manages time-based operations

Web Interface

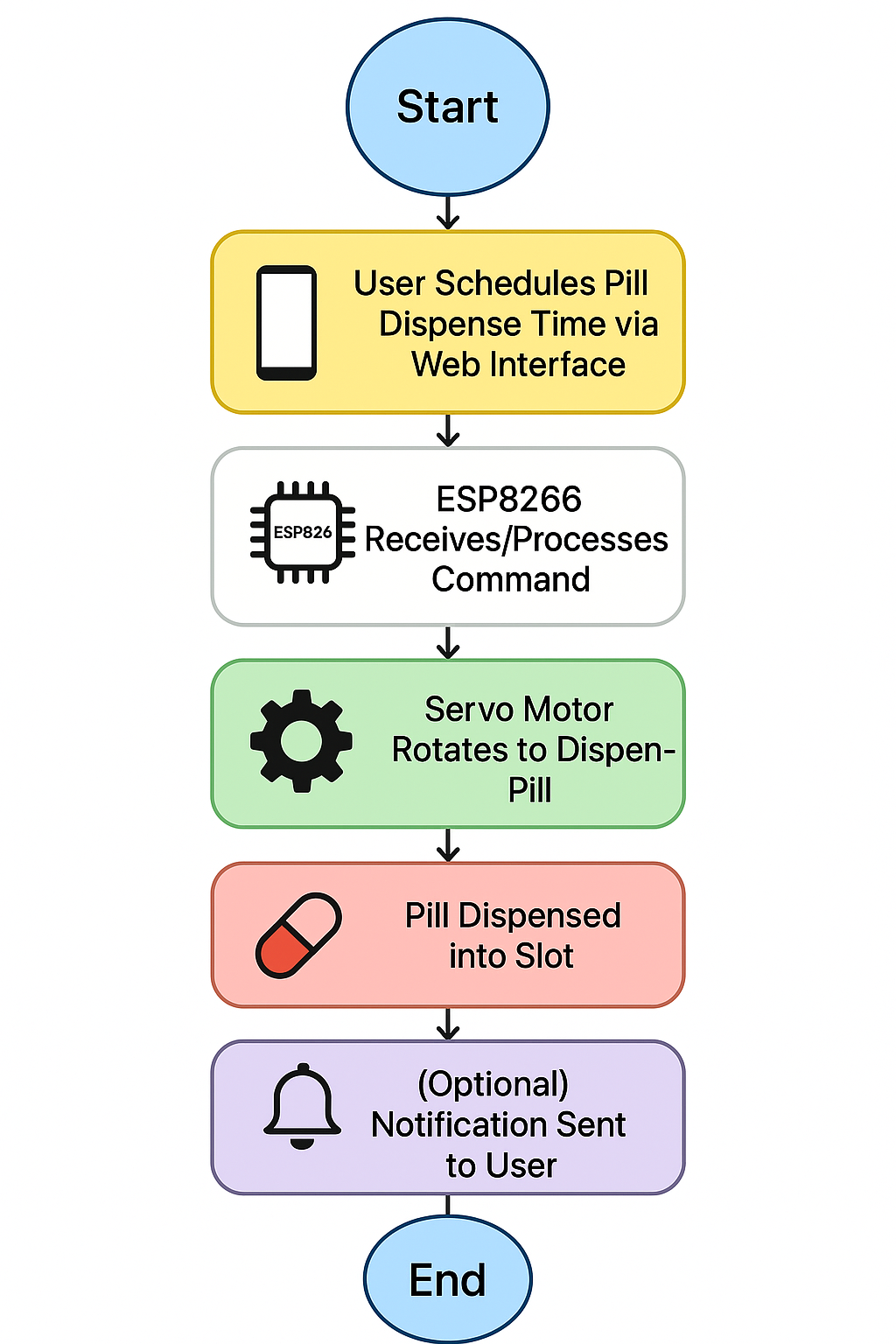
* HTML + CSS – Simple interface to set time & days for pill dispensing
* Stores user-defined schedule in variables or EEPROM



**Figure 2: UI Screenshot with Toggle Butto**

**5. System Workflow**

1. Startup: ESP8266 connects to WiFi and hosts a local web server.
2. UI Access: User connects to the device IP (e.g., 192.168.1.15) and accesses the scheduling page.
3. Schedule Input: User selects time(s) and days for dispensing via the web form.
4. Time Check Loop: The code continuously compares current time with the schedule.
5. Servo Action: When time matches the schedule, the MG995 rotates to dispense a pill and resets after a delay.
6. Idle State: The device waits for the next scheduled interval.



**Figure 3**: Block Diagram of Smart Pill Dispenser

**6. Survey (Adapted)**

Survey of 30 individuals including senior citizens, caregivers, and patients revealed:

| Question | Key Findings |
| --- | --- |
| Do you forget medication occasionally? | 85% admitted to missing doses |
| Would a scheduled dispenser help? | 90% showed interest |
| Preferred interface type? | 75% preferred browser over apps |
| Would manual override be useful? | 65% agreed |

Conclusion: The majority favored scheduled automation with a minimal web interface.

7. Prototype

Hardware Setup:

* NodeMCU ESP8266 + MG995 mounted to a rotating pill container.
* USB power or portable bank.
* Mechanism allows pill drop via slot rotation.

Software/UI:

* Responsive HTML page with time/day selectors
* Simple buttons for setting, testing, or resetting schedule
* All files and logic hosted onboard the ESP8266

Prototype Features:

* Schedule-based control
* Local web access without login
* No third-party dependencies
* Expandable for alerts or remote logs

**8. Testing**

| Test Type | Result |
| --- | --- |
| Schedule Matching | Accurate with system time |
| Servo Activation | 100% consistent movement |
| Response Delay | <1 second |
| Web Access Uptime | >24 hours tested stable |
| Power Consumption | ~120-150 mA during activation |

**9. Discussion**

Strengths:

* Automated Scheduling: Reduces human error
* User-Friendly Interface: Accessible via browser
* Affordable and Scalable: Minimal hardware, high reliability
* No Cloud Dependency: Private and local operation

Limitations:

* No Backup Battery: Power outage resets
* No Real-Time Clock (RTC): Dependent on system time unless integrated
* No Alerts/Feedback: No confirmation or logs

Future Scope:

* RTC Module or NTP Integration
* Pill Dispense Logging (to EEPROM or Firebase)
* Telegram/Email Alerts for missed doses
* Integration with Health Apps or Wearables
* Sensor feedback for pill drop confirmation

**10. Conclusion**

The Smart Pill Dispenser demonstrates how embedded systems and IoT can solve real-world health challenges. By combining ESP8266 and MG995 with a browser-based schedule UI, the system ensures timely pill delivery without manual input.

Key Takeaways:

* Successfully implemented a scheduled, web-controlled pill dispenser
* Local server hosting for ease of access
* Reliable and accurate servo-based dispensing
* Opens doors for advanced smart health solutions