PROJECT TITLE

Rise 'n Riddle: Puzzle Your Morning

STUDENT/TEAM INFORMATION

Team Name if any: **Team # on Canvas you have self-signed-up for:**	**Lonely Riddler**
Team member 1 (Team Lead) (Lastname, Firstname; SDSU email; picture):	Pielmaier, Julian – jpielmaier9966@sdsu.edu 

ABSTRACT (15 points)

(Summarize your project (motivation, goals, system design and results). Max 300 words).

The project delivers an interactive alarm system designed to reduce sleep inertia by requiring users to solve a four-LED puzzle before silencing the alarm. Performance metrics—reaction time and attempt count—are recorded to quantify morning alertness. Environmental conditions (temperature and humidity) are measured via a DHT sensor, while current weather data is retrieved server-side through a third-party API. All device events and user sleep ratings are packaged as JSON and transmitted over Wi-Fi to a cloud-hosted server, which exposes RESTful endpoints for data ingestion and hosts a web dashboard. The dashboard offers interactive visualizations of environmental trends, sleep-quality ratings, puzzle performance, and live weather, as well as remote configuration options. An adaptive difficulty algorithm adjusts puzzle length and display speed based on past performance to maintain an optimal challenge. End-to-end testing confirmed stable connectivity, accurate data logging, responsive analytics, and seamless user interaction. The resulting system provides a holistic solution for engaging cognitive function upon waking and delivering actionable insights into how environmental factors influence sleep quality and daily productivity.

INTRODUCTION (15 pts)

Motivation/Background (3 pts)

(Describe the problem you want to solve and why it is important. Max 300 words).

Standard alarms do little beyond making noise; they leave users to overcome early-morning sluggishness unaided. This project embeds an active task into that moment. When the alarm sounds, a random pattern appears on a compact four-LED display; the tone stops only after the same sequence is entered on a matching four-button keypad. This brief puzzle delivers an immediate mental and motor stimulus, helping the user reach full alertness faster than a tap-to-dismiss alarm.

Each alarm event is recorded in detail. The device logs reaction time, error count, and number of attempts, creating a daily measure of morning alertness. Ambient temperature and humidity are captured simultaneously, and current weather data are fetched, linking performance to environmental conditions.

Difficulty adapts automatically: recent performance raises or lowers sequence length and timing, keeping the challenge engaging without becoming frustrating. All settings and historical logs are available through a lightweight web dashboard, enabling remote configuration and long-term trend analysis.

By combining an interactive dismissal task, contextual data logging, and adaptive difficulty, Rise ’n Riddle turns an ordinary alarm into a concise tool for sharper mornings and informed sleep-habit improvement.

Project Goals (6 pts)

(Describe the project general goals. Max 200 words).

The project seeks to deliver an interactive alarm system that boosts morning alertness and supports better sleep management. The core objective is to guarantee full wakefulness by requiring users to reproduce a randomly generated pattern on a four-LED, four-button interface before the alarm can be silenced, engaging cognitive and motor skills immediately after waking.

Each alarm event is logged with key performance metrics—reaction time, error count, and number of attempts—alongside a user-entered sleep-quality rating. A DHT sensor continuously records bedroom temperature and humidity, adding environmental context to the performance data. Logs are transmitted via Wi-Fi to an AWS server, where they are combined with current outdoor weather information retrieved server-side.

To keep the challenge effective over time, an adaptive algorithm adjusts sequence length and timing according to recent results, maintaining a balance between difficulty and user motivation. All records and trend visualisations are accessible through a companion web dashboard, enabling remote configuration, sleep rating and long-term review.
Taken together, these goals create a self-contained, user-friendly system that not only secures an alert start to the day but also supplies actionable insights for ongoing improvements in sleep quality and daily productivity.

Assumptions (3 pts)

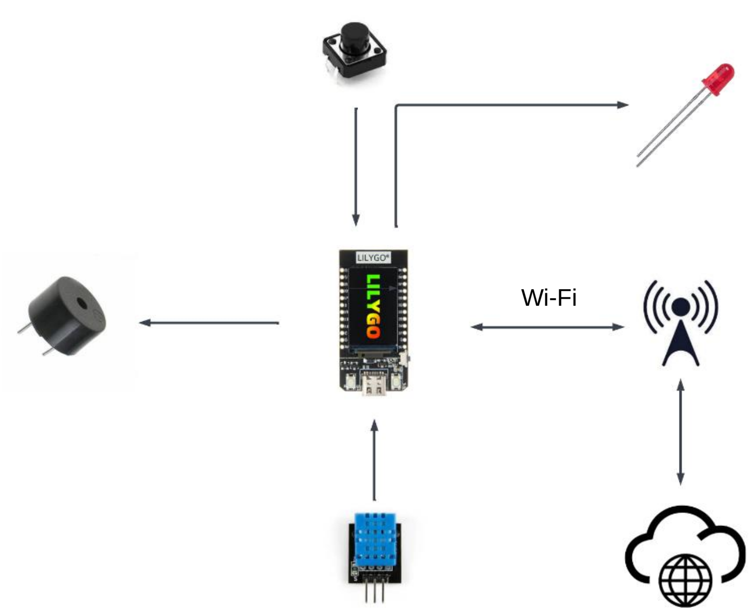
(Describe the assumptions (if any) you are making to solve the problem. Max 180 words).

- **Single-user operation** The device records and interprets data for one primary user; profile switching and simultaneous multi-user support are outside this scope
- **Stable home Wi-Fi** A reliable connection is available at the bedside for log uploads and dashboard access; intermittent outages are not handled beyond local buffering
- **Consistent placement** The unit remains in the same bedroom location, so temperature-and-humidity readings accurately reflect the sleeping environment
- **Mains power** The prototype is powered by USB; battery-only operation and energy optimisation are not addressed

SYSTEM ARCHITECTURE (20 pts)

(Describe the final architecture you have implemented listing sensors, communication protocols (Wi-Fi, BLE, ...), cloud services and user interfaces. Include a block diagram of the system. Max 300 words).

The system is built around an ESP32-based development board that reads environmental conditions via a DHT sensor, captures user input through buttons to control a puzzle interface with LEDs, and issues feedback via a buzzer. Sensor measurements and user interactions—including button presses, LED states, and buzzer alerts—are packaged as JSON and transmitted over Wi-Fi using HTTP requests to a cloud-hosted web server. The server exposes RESTful endpoints to receive device data and to serve the Web UI, persisting all incoming records in a database for historical analysis. Separately, it periodically queries an external weather API to enrich stored records with current meteorological conditions. The Web UI—accessible in any modern browser—offers a simple form for users to rate their sleep quality and displays interactive charts of environmental trends, sleep ratings, puzzle-performance metrics, and real-time weather information. User submissions from the dashboard are sent back to the server, stored, and immediately reflected in the analytics views. In this way, the ESP32 module, cloud services (including API integration and data storage), and Web interface work together to deliver continuous monitoring, user feedback, and actionable insights.



FINAL LIST OF HARDWARE COMPONENTS (5 pts)

(Write the final list and quantity of the components you have included in your system)

Component/part	Quantity
Breadboard	1
LILYGO-TTGO	1
Humidity and Temperature Sensor	1

USB Type-C Cable	1
Buzzer	1
Button	4
LED	4
Resistor	8
Jumper Wires	21

PROJECT IMPLEMENTATION (30 PTS)

Tasks/Milestones Completed (15 pts)

(Describe the main tasks that you have completed in this project. Max 250 words).

Task Completed	Team Member
All planned milestones have been completed, resulting in a fully functional interactive alarm system: • **Hardware Assembly** The TTGO (ESP32) board, LED line, button array, DHT sensor, and buzzer were mounted and wired on a custom breadboard layout. Incremental validation ensured reliable electrical connections and component operation. • **Core Firmware Development** Alarm scheduling, random sequence generation for the four-LED puzzle, and button-press input handling were implemented. The buzzer alerts and puzzle-success logic now correctly silence the alarm and advance difficulty. • **Environmental Sensing & Connectivity** Onboard temperature and humidity readings were integrated and Wi-Fi connectivity was established. Server-side RESTful endpoints now receive sensor data and trigger periodic weather API requests to enrich the dataset. • **Remote Data Logging & Dashboard** A cloud-hosted web server with database storage was deployed. Device events and user sleep ratings are logged remotely, and a Web UI was developed to visualize environmental trends, puzzle performance, and real-time weather. User inputs from the dashboard update immediately. • **Adaptive Difficulty Algorithm** Performance metrics (reaction time, attempt count) feed an algorithm that dynamically adjusts puzzle length and display speed, maintaining an appropriate challenge level. • **Testing & Calibration** End-to-end testing verified stable Wi-Fi reconnection, accurate data serialization, dashboard access, and responsive analytics. Iterative bug fixes and layout refinements have produced a robust, user-friendly system.	Julian

Challenges/Roadblocks (5 pts)

(Describe the challenges that you have faced and how you solved them if that is the case. Max 300 words).

One major hardware constraint was the limited GPIO availability on the ESP32, which forced a shift from a 3×3 LED matrix to a single line of four LEDs paired with four buttons. This change required redesigning the interaction flow to maintain the project's core functionality. Spatial limitations on the breadboard also posed challenges: optimal wiring between the TTGO board, LEDs, buttons, buzzer, and DHT sensor took several layout iterations. By planning the component arrangement in advance and conducting step-by-step connectivity tests, reliable hardware performance was achieved. On the software side, adapting the puzzle generator to the revised button layout introduced integration difficulties. Refining the input-capture logic and iteratively debugging the comparison routine ensured accurate recognition of user responses. Throughout development, an iterative cycle of testing, layout refinement, and targeted debugging addressed each obstacle, resulting in smooth operation of both hardware and software subsystems.

Tasks Not Completed (5 pts)

(Describe the tasks that you originally planned to complete but were not completed. If all tasks were completed, state so. Max 250 words).

Task	Reason
ESP32 Weather Retrieval	All originally planned tasks have been completed. The only adjustment involved relocating weather data retrieval from the ESP32 to the server-side dashboard. While the proposal specified that the device would fetch current meteorological data, the server now handles API calls and integrates weather information into the analytics views. This change was made to simplify device firmware and centralize external API management. Aside from this rational design decision, every milestone—hardware integration, core firmware development, sensor and Wi-Fi integration, remote logging, adaptive difficulty implementation, and end-to-end testing—has been fully implemented as outlined in the proposal.

WEAK POINTS / FUTURE WORK (15 pts)

(Mention at least two points of your project that have room for improvement. These points can be additions to the existing project setup (components) or improvement of the current implementation. Max 200 words).

While the current system meets its objectives, several enhancements lie beyond the original scope. First, the reduction from a 3×3 LED matrix to four LEDs has simplified the puzzle's complexity; reinstating a larger array or adopting a more intricate interaction pattern would better challenge users. Second, the Web UI—though fully functional—would benefit from a visual polish, including cleaner layouts, smoother transitions, and full mobile responsiveness to improve overall usability. Finally, the ESP32's onboard display remains under-utilized; repurposing it as a bedside clock and real-time status panel would add value without additional hardware.

SOURCE CODE (25 pts)

Please include a link to the source code of your project. A link to a repository (like GitHub) is preferred.

AWS Server Code: https://github.com/Jaypee2109/iot_server.git
Project Code: https://github.com/Jaypee2109/iot_project.git