BagTrack: Smart Backpack for Students & Professionals

Presented by:
Nehal Garg 2022UCS1686
MD Arman Mansoori 2022UCS1672
Anjali 2022UCS1679

TEACHER INCHARGE:
Prof. Gaurav Singhal

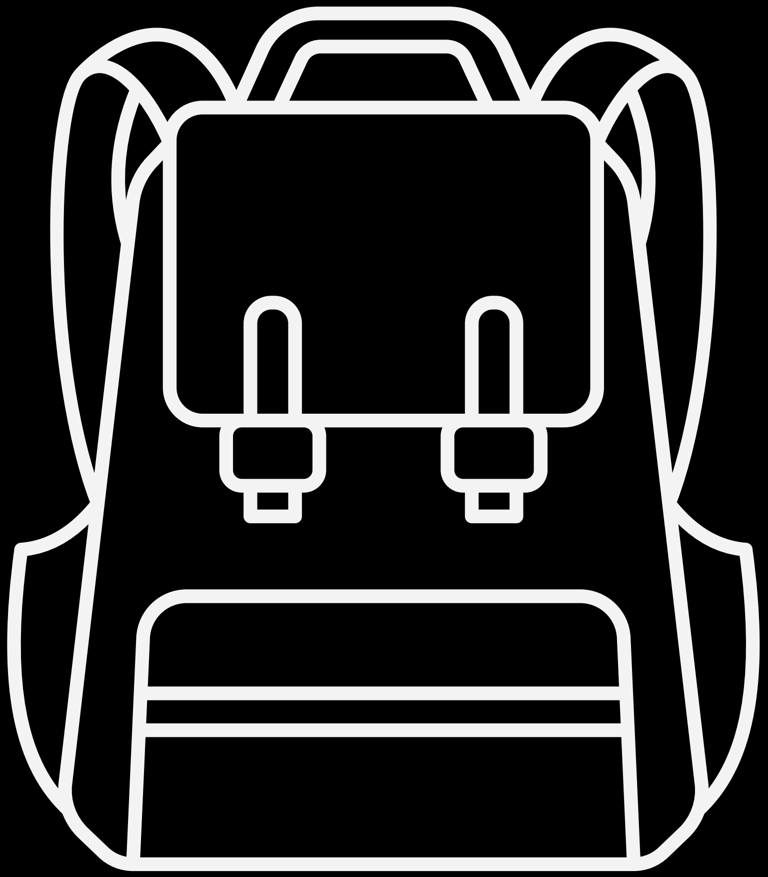
TABLE OF CONTENT

- Introduction
- Motivation
- Problem Statement
- Contributions, Novelty
- Literature Work, Summary, Research Gap
- Methodology, Algorithms, Approach
- Hardware, Software
- Results & Observations
- Prototype Link
- Conclusion and Future Work

introduction

What is BagTrack?
BagTrack is a smart IoT-enabled backpack designed specifically for students and professionals who need a secure and organized way to manage their belongings. With embedded smart components, the backpack goes beyond just carrying items—it helps you track, manage, and protect what matters most.

Why Do We Need It?
In today's fast-paced lifestyle, it’s easy to misplace or lose important belongings such as laptops, notes, books, or gadgets. Existing backpacks don’t offer real-time tracking or theft prevention. That’s where BagTrack steps in—offering technology-powered convenience, organization, and security.

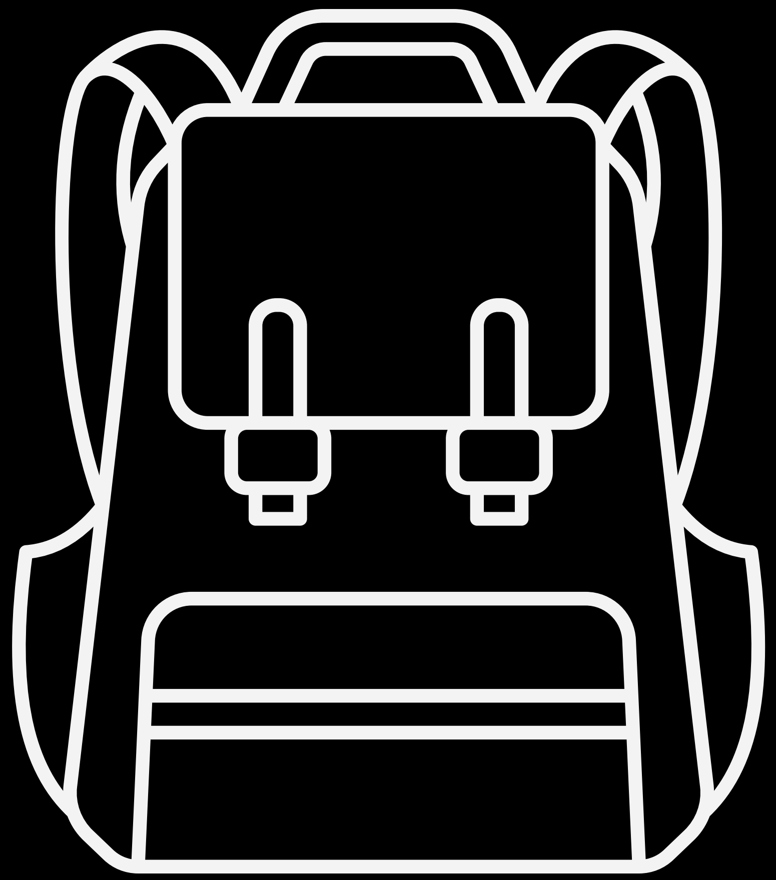


Motivation

- Lost an expensive laptop charger, highlighting the need for better item management.
- Frequent "Lost and Found" notifications on nsut mail id for essentials like wallets, chargers, keys, and ID cards.
- Daily struggle to remember and carry all important belongings.
- Lack of smart tracking or security features in regular backpacks.

Problem Statement

- Frequent misplacement of essential items like laptops, books, chargers, and gadgets.
- Lack of real-time tracking and efficient inventory management systems.
- Increased risk of theft and unauthorized access to valuable belongings.

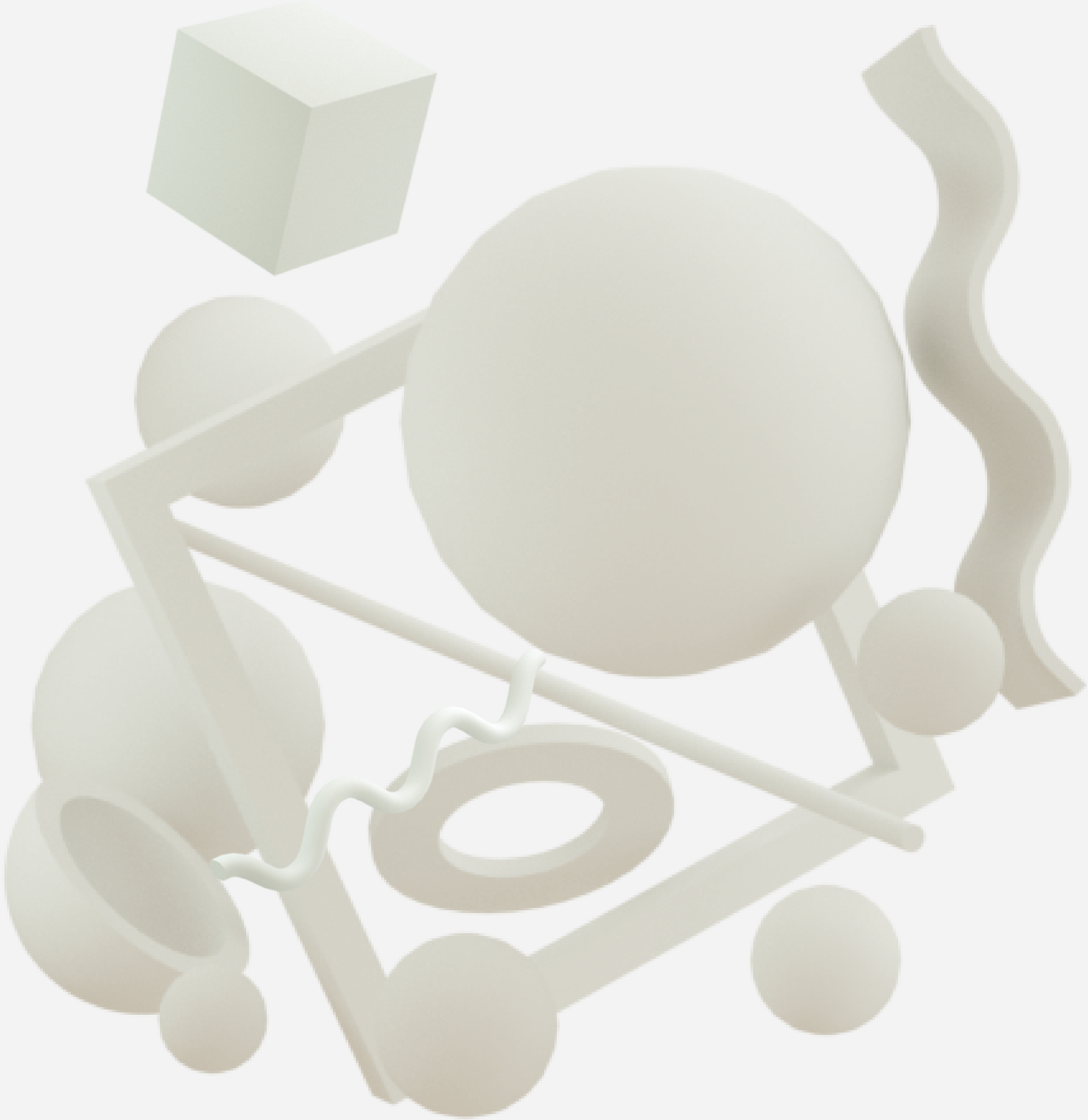


Contributions and Novelty

- Integration of GPS, Arduino, and mobile app for comprehensive IoT functionality
- Item-level monitoring using GPS and RF REceiver and transmittor tags for precise tracking
- Affordable smart solution tailored for students and professionals
- Real-time notifications for forgotten items and unauthorized movement
- User-centric design based on everyday item-loss scenarios

Literature Work

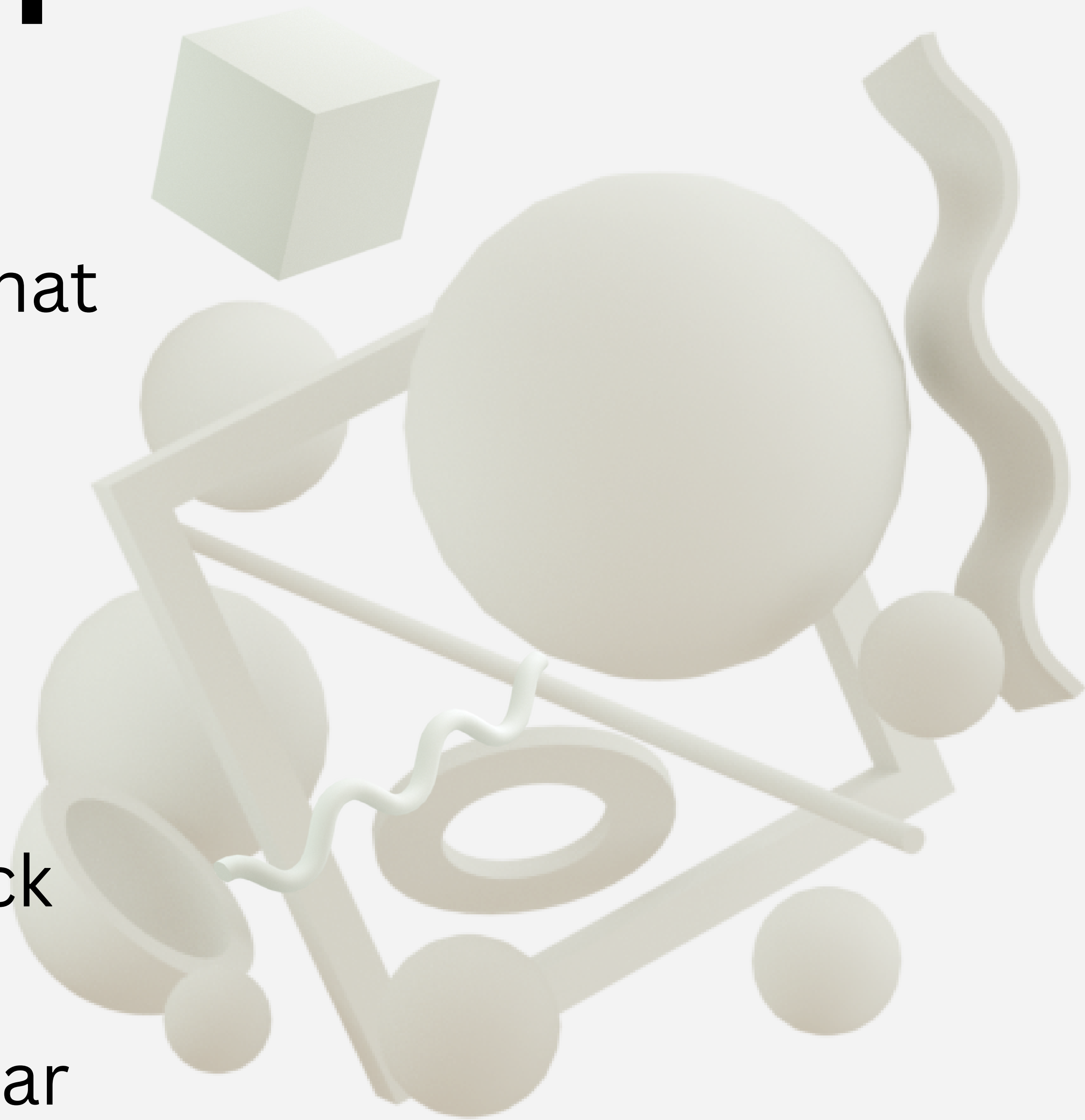
- Real-Time GPS Tracking: Continuous, precise location monitoring of the backpack.
- RF based receiver and transmitor: tags that are attached to individual items with real-time alerts for missing or misplaced items are receivers with bag acting as transmittor.
- Anti-Theft Features: Immediate notifications upon unauthorized access or tampering.



Summary

- The system uses an ESP8266 module connected to a GPS sensor and a Telegram bot.
- It connects to Wi-Fi and waits for either:
- A physical trigger (button press), or
- A Telegram command (/status).
- Upon activation, it reads GPS coordinates.
- If a valid location is found, it sends a Google Maps link via Telegram.
- If GPS fails, it uses default fallback coordinates.
- The system checks for new Telegram messages every 1 second.

Research Gap



Research Gap:

- Lack of a low-cost, all-in-one smart backpack that combines:
  - GPS-based bag tracking
  - RF_based item tracking
  - Real-time alert system
  - User-friendly mobile integration
- Existing systems are either too expensive or lack multi-item tracking features.
- limitations of gps inside a buliding to a particular place

Methodology

1. System Design:
    - Identified the problem of misplaced items and the need for real-time location tracking.
    - Proposed an IoT-based smart backpack with GPS, Bluetooth tracking, and Telegram notifications.
2. Hardware Integration:
    - Used ESP8266 for Wi-Fi connectivity and processing.
    - Integrated NEO-6M GPS for location tracking and RF receiver for physical trigger events.
    - Powered by a Li-Po battery for portability.
3. Software Development:
    - Programmed using Arduino IDE to handle GPS, Wi-Fi, and Telegram communication.
    - Developed a Telegram bot to send location updates based on commands.
4. Testing:
    - Tested GPS accuracy and Telegram communication.
    - Verified physical trigger functionality for location notifications.

Algorithms

- Wi-Fi Connection:

Connects to a pre-configured Wi-Fi network (SSID, password).

- GPS Data Acquisition:

Reads and decodes GPS data using the TinyGPSPlus library.
Sends location data to Telegram if valid GPS is available.

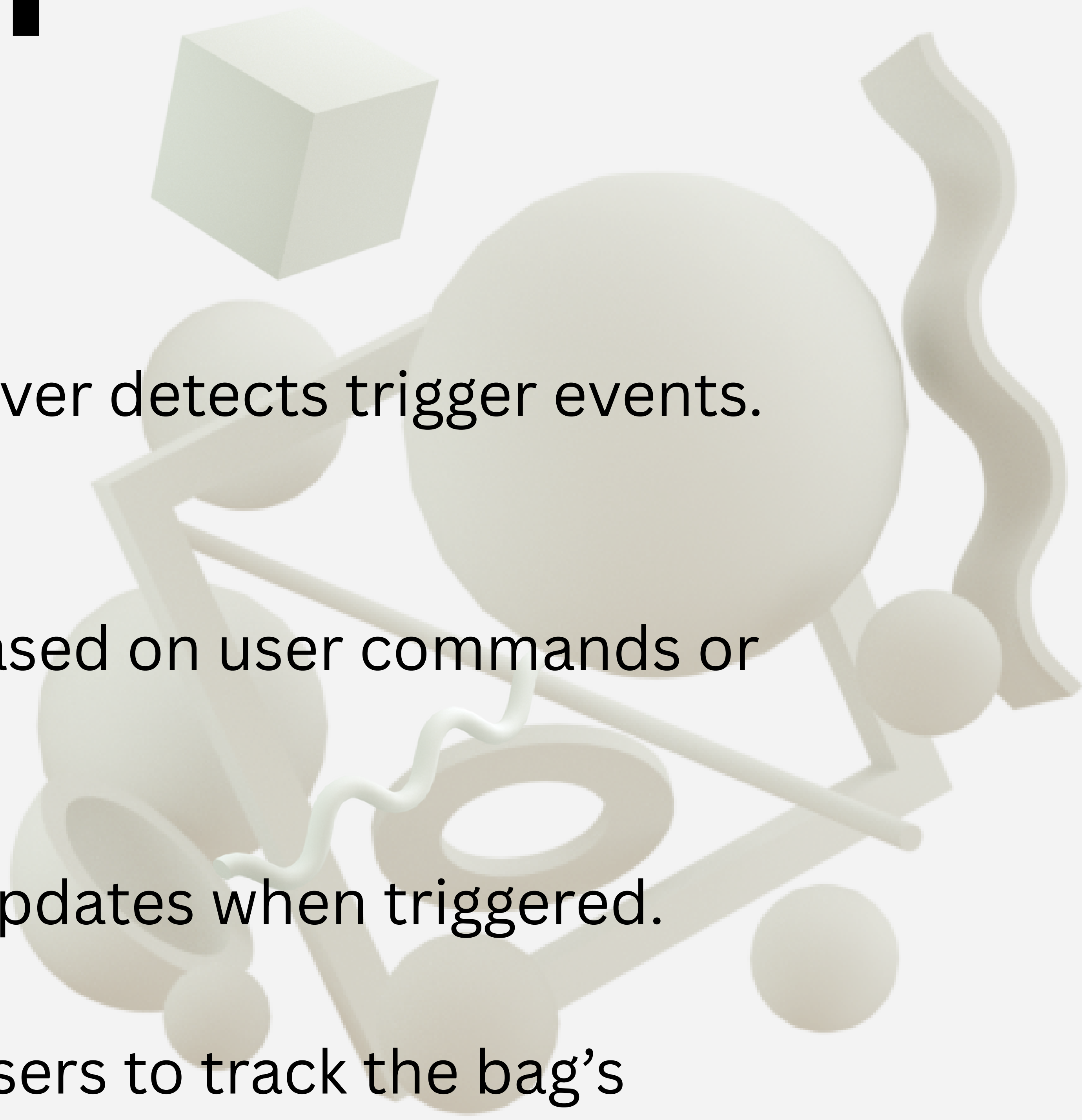
- Telegram Interaction:

The bot handles commands like /status to send current location.
Sends location updates when physical trigger is activated.

- Trigger Detection:

Monitors physical interactions (e.g., opening the bag) and sends location updates if the trigger is pressed.

Approach



Hardware:

- ESP8266 processes GPS data and sends it via Wi-Fi.
- NEO-6M GPS provides location data, and the RF receiver detects trigger events.

Software:

- Arduino IDE is used to program the system.
- The Telegram bot sends real-time location updates based on user commands or trigger events.

Real-Time Monitoring:

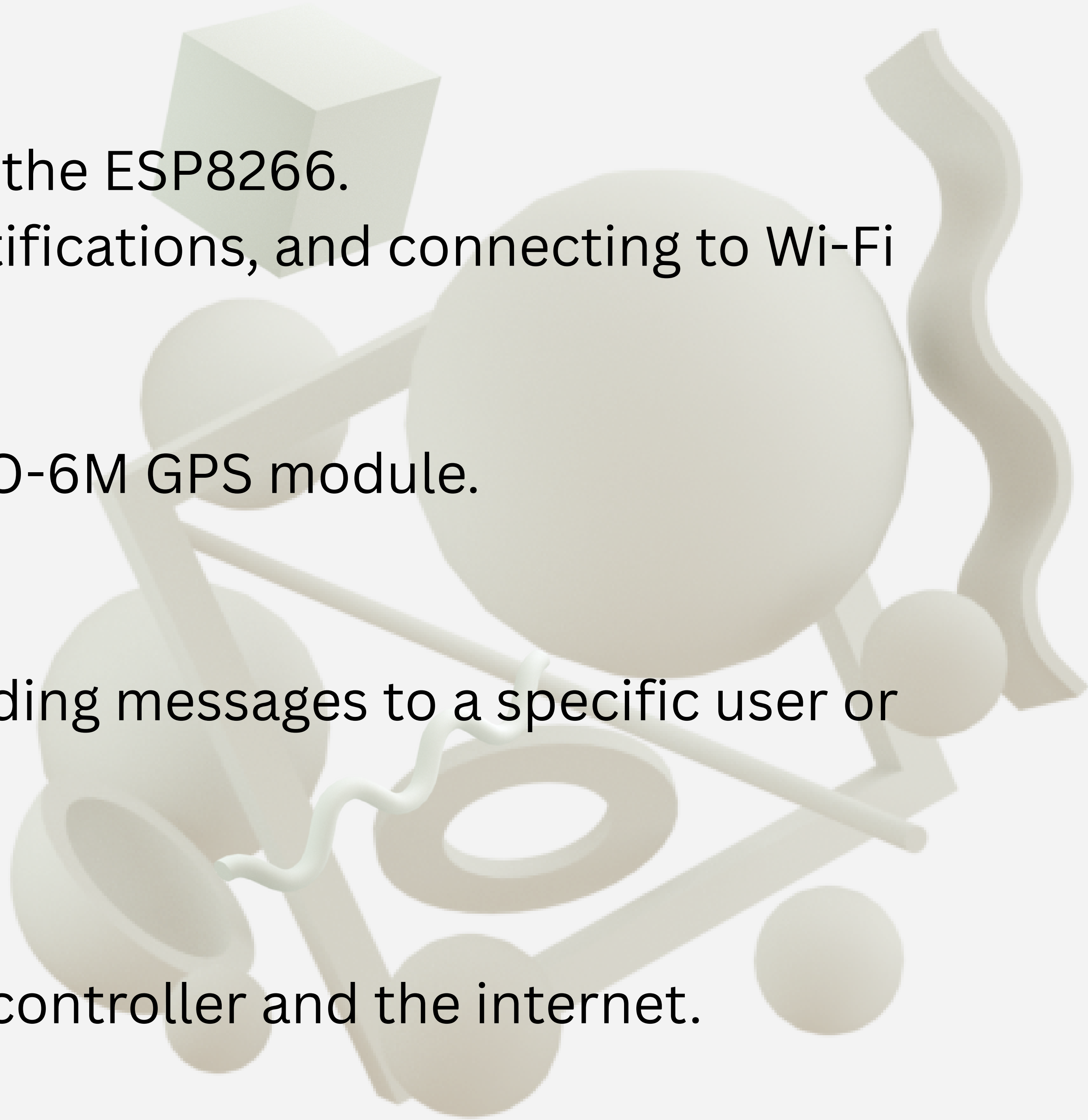
- The system continuously reads GPS data and sends updates when triggered.
- User Interface:
- Simple interaction through the Telegram bot allows users to track the bag’s location easily.

HARDWARE

1. ESP8266 (NodeMCU)
Function: This microcontroller handles Wi-Fi connectivity and communication with other hardware components.
Cost: ₹300
Role: Sends location data to Telegram and processes commands from the user.
2. NEO-6M GPS Module
Function: Provides location tracking with real-time latitude and longitude data.
Cost: ₹600
Role: Gathers location data to be sent via Telegram.
3. RF Receiver and Transmitter
Function: Detects physical interactions, such as opening the bag, and triggers the sending of location updates.
Cost: ₹200
Role: Enables interaction with the bag to send notifications via Telegram.
4. Li-Po Battery and Power Management
Function: Powers the entire system.
Cost: ₹400
Role: Ensures portability of the smart backpack.
5. Wires and Connectors
Cost: ₹100
Role: Establishes the connections between the various hardware components.

Total Hardware Cost: ₹1,600

SOFTWARE



1. Arduino IDE
Function: The primary development environment used to program the ESP8266.
Role: Allows the coding of logic for reading GPS data, triggering notifications, and connecting to Wi-Fi and Telegram.
2. TinyGPSPlus Library
Function: A library for parsing and decoding GPS data from the NEO-6M GPS module.
Role: Provides an easy interface to read latitude and longitude.
3. UniversalTelegramBot Library
Function: Interface for communicating with Telegram APIs and sending messages to a specific user or group.
Role: Allows the system to send location updates via Telegram bot.
4. Wi-Fi and Telegram APIs
Function: Provides communication capabilities between the microcontroller and the internet.
Role: Facilitates the sending and receiving of data over Wi-Fi.
5. Tools Used:
GitHub: Repository to store and share the project code.

Results and Observations

GPS Accuracy and Location Tracking:

- Observation: The GPS module (NEO-6M) provided reliable location data within a 5-meter radius, confirmed by Google Maps.
- Result: The GPS data was accurate enough for real-time tracking, with fallback to default coordinates when data was unavailable.

Trigger Response Time:

- Observation: The trigger pin response time was consistent, with notifications sent within 2-3 seconds.
- Result: The system responded efficiently to trigger presses and sent notifications promptly.

Wi-Fi Connectivity:

- Observation: The ESP8266 connected to the Wi-Fi network within 2-3 seconds after boot.
- Result: Wi-Fi connectivity was stable and reliable, with no connection issues.

Telegram Bot Communication:

- Observation: The bot successfully processed commands like "/status" and responded with location data.
- Result: Telegram integration worked as expected, delivering real-time location updates.

Battery Performance:

- Observation: The Li-Po battery lasted approximately 4-5 hours with moderate use.
- Result: The battery performance met expectations, providing enough power for daily usage.

BLE Tag Integration (Testing in Progress):

- Observation: BLE tags were detected with reasonable accuracy during early testing.
- Result: Further testing is needed to optimize BLE tag tracking and alert generation.

System Integration and Reliability:

- Observation: The system demonstrated reliable performance with minimal failures in location tracking and message delivery.
- Result: The integration of all components was successful, providing consistent and accurate results.

PROTOTYPE LINK

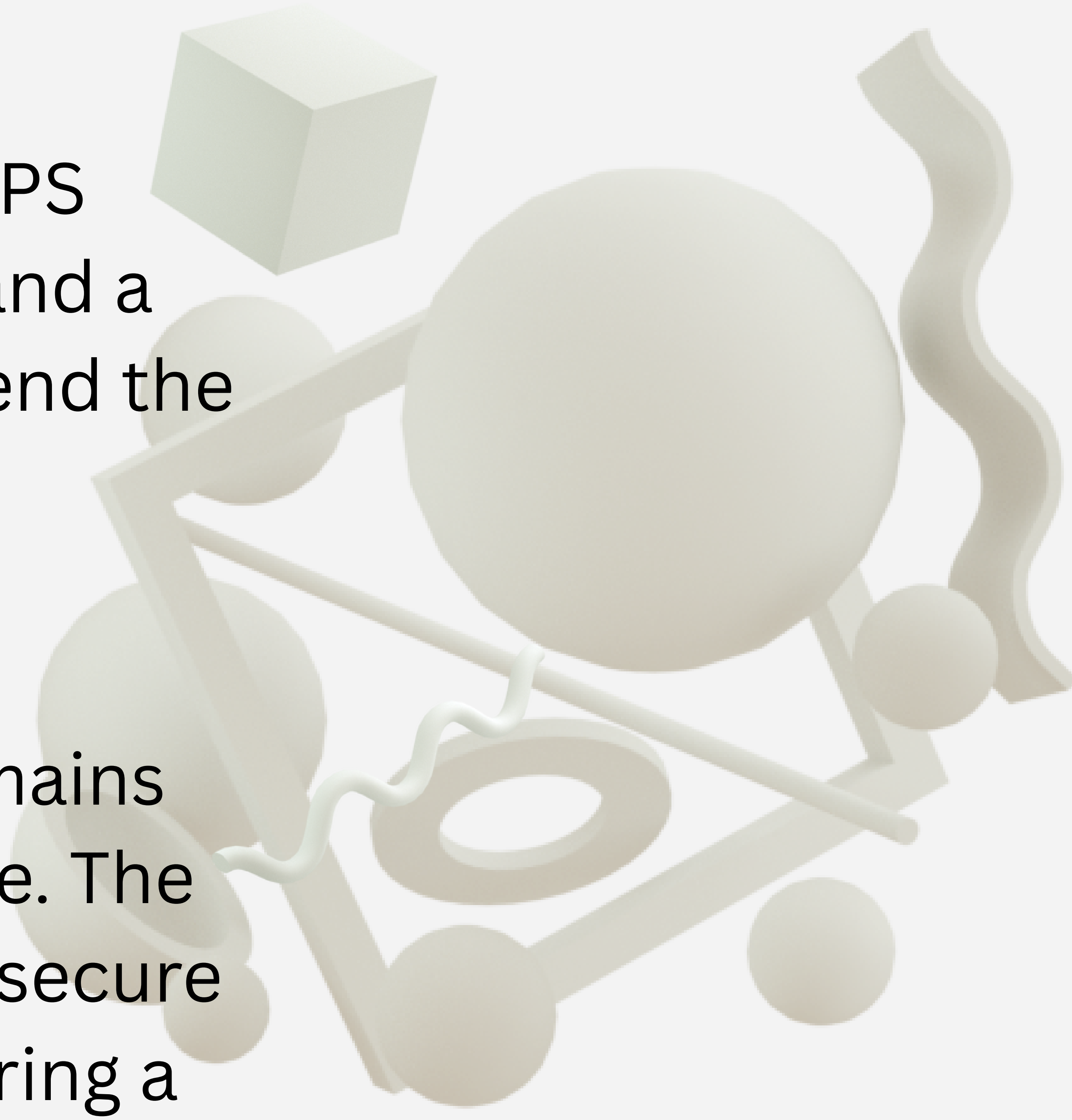
https://github.com/nehalgarg
-23/Smart-Bagtrack-

CONCLUSION

This project successfully demonstrates a real-time GPS tracking system using the ESP8266 microcontroller and a Telegram bot interface. The system can accurately send the current location to the user when:

- A physical trigger is activated, or
- A Telegram command (/status) is received.

The use of a fallback location ensures the system remains functional even if GPS data is temporarily unavailable. The integration of Telegram provides a user-friendly and secure method for receiving location updates without requiring a dedicated app or website.



Future Work

For future development, this project can be expanded into a fully-fledged GPS tracking solution suitable for applications like lost item tracking, personal safety devices, or smart navigation systems. By integrating features such as live location tracking, periodic updates, and geofencing, the system can automatically alert users when the device enters or exits predefined zones.

In addition, cloud integration could greatly enhance data management and accessibility. For example, storing location logs in Firebase, Google Sheets, or a custom backend would allow for historical route tracking and data analytics. Incorporating a small OLED display, buzzer, or LED indicators could give real-time feedback to the user on the device’s status.

THANK YOU