SMART GOODS TRANSFER USING IOT BOX

A Project Report

Submitted By

Rahul Bamaniya (210303105161)

Aryan Hinglajia (21030105178)

Abhishek Lodhari (210303105189)

Vraj Jiyani (210303105182)

in Partial Fulfilment For the Award of

the Degree of

BACHELOR OF TECHNOLOGY

COMPUTER SCIENCE & ENGINEERING

Under the Guidance of

Prof. Gaurav Varshney

Assistant Professor

VADODARA

April - 2024

PARUL UNIVERSITY

CERTIFICATE

This is to Certify that Project - 1 (203105499) of 6th Semester entitled “SMART GOODS TRANSFER USING IOT BOX” of Group No. G-350 has been successfully completed by

- RAHUL BAMANIYA- 210303105161
- ARYAN HINGLAJIA- 210303105178
- ABHISHEK LODHARI- 210303105189
- VRAJ JIYANI- 210303105182

under my guidance in partial fulfillment of the Bachelor of Technology (B.Tech) in Computer Science & Engineering of Parul University in Academic Year 2023- 2024.

Date of Submission :_________________

Prof. Gaurav Varshney,

Project Guide

Dr. Amit Barve,

Head of Department,

CSE, PIET,

Parul University.

Project Coordinator:- **Dr. Kruti Sutaria** or

Prof. Yatin Shukla

Acknowledgements

"The single greatest cause of happiness is gratitude."

-Auliq-Ice

We extend our sincere appreciation to all those who have contributed to the progress of our ongoing project, "Smart Goods Transfer utilizing an IoT Box."

First and foremost, we would like to express our gratitude to our project supervisor Mr. Gaurav Varshney for their continuous guidance, valuable insights, and unwavering support throughout the initial phase of this project. Their expertise has been instrumental in steering us in the right direction and providing clarity on our objectives.

We also extend our thanks to the faculty members of Computer Science and Engineering for their encouragement and feedback during the initial stages of our project. Their constructive criticism has helped shape our approach and refine our ideas.

Furthermore, we appreciate the efforts of our team members who have diligently contributed to the project's progress thus far. Each member's dedication and collaboration have been crucial in achieving our milestones and overcoming challenges.

Additionally, we would like to acknowledge the support of our families and friends for their understanding and encouragement as we navigate through the complexities of this project. Their unwavering belief in our abilities has been a source of motivation.

Finally, we express our gratitude to all individuals, organizations, and resources that have provided assistance, guidance, and inspiration throughout the course of our project.

Thank you to everyone involved in making the initial phase of this project possible. Your contributions are deeply appreciated, and we look forward to continuing our efforts to complete the project successfully.

Team 350

CSE, PIET

Parul University,

Vadodara

Abstract

The Smart Goods Transfer utilizing an IoT Box is a pioneering solution designed to revolutionize the delivery landscape by integrating cutting-edge IoT technology with logistics operations. This project aims to provide a secure, efficient, and user-friendly platform for transferring goods between senders and recipients in a transparent and streamlined manner.

At the core of the system is an IoT Box equipped with sensors, microcontrollers, communication modules, and GPS technology. These components work in synergy to enable secure item transfer, real-time tracking, and seamless communication between users throughout the delivery process.

The process begins with the sender reserving a designated IoT Box through a web application and securely placing the desired item inside. Upon detection of the item, the IoT Box generates a unique QR code, locking the box for added security. Both sender and recipient receive timely notifications via email and phone, allowing them to monitor the package's real-time location through the integrated GPS module.

Upon arrival, the recipient presents the QR code to unlock the box, facilitating secure retrieval of the item. The box can then be returned to the sender, directed to a collection point, or retrieved by delivery personnel, completing the delivery cycle seamlessly.

In summary, the Smart Goods Transfer utilizing an IoT Box offers a comprehensive solution that combines advanced technology with intuitive design to deliver a secure, efficient, and dependable delivery experience for all stakeholders involved. This project represents a significant advancement in the field of logistics, promising to redefine the way goods are transferred and delivered in the digital age.

Table of Contents

Acknowledgements iii

Abstract iv

1 Introduction 1

1.1 Purpose 1

1.2 Intended Audience 1

1.3 Reading Suggestions 2

1.4 Product Scope 2

2 Literature Survey 3

2.1 Paper 1: Internet of Things (IoT): A vision, architectural elements, and future directions 3

2.1.1 Author: 3

2.1.2 Abstract: 3

2.2 Paper 2: Internet of Things (IoT) in logistics: A comprehensive literature review . . 3

2.2.1 Author: 3

2.2.2 Abstract: 4

2.3 Paper 3: A systematic review of Internet of Things (IoT) applications in supply chain management (SCM) 4

2.3.1 Author: 4

2.3.2 Abstract: 4

2.4 Paper 4: Internet of Things for smart logistics: Technologies and applications . . . 4

2.4.1 Author: 4

2.4.2 Abstract: 5

2.5 Paper 5: Research on Internet of Things architecture for logistics monitoring 5

2.5.1 Author: 5

2.5.2 Abstract: 5

2.6 Paper 6: Internet of Things for smart cities 6

2.6.1 Author: 6

2.6.2 Abstract: 6

2.7 Paper 7: Smart logistics for industry 4.0: A review 6

2.7.1 Author: 6

2.7.2 Abstract: 6

2.8 Paper 8: Smart logistics in the context of Industry 4.0: A review 6

2.8.1 Author: 6

2.8.2 Abstract: 6

2.9 Paper 9: Smart logistics for perishable products based on Internet of Things 7

2.9.1 Author: 7

2.9.2 Abstract: 7

2.10 Paper 10: A review on the application of Internet of Things in logistics 7

2.10.1 Author: 7

2.10.2 Abstract: 7

2.11 Paper 11: The internet of things: a survey 7

2.11.1 Author: 7

2.11.2 Abstract: 7

2.12 Paper 12: Online Structural Health Monitoring and Parameter Estimation for Vibrating Active Cantilever Beams Using Low-Priced Microcontrollers 8

2.12.1 Author: 8

2.12.2 Abstract: 8

2.13 Paper 13: Arduino core for ESP32 WiFi chip 8

2.13.1 Author: 8

2.13.2 Abstract: 8
2.14 Paper 14: Espressif IoT Development Framework 9
2.14.1 Author: 9
2.14.2 Abstract: 9
2.15 Paper 15: The Internet of Things (IoT): Applications, investments, and challenges for enterprises 9
2.15.1 Author: 9
2.15.2 Abstract: 9
2.16 Paper 16: Internet of Things (IoT): A vision, architectural elements, and future directions 9
2.16.1 Author: 9
2.16.2 Abstract: 10
2.17 Paper 17: A review of logistics Internet-of-Things: Current trends and scope for future research 10
2.17.1 Author: 10
2.17.2 Abstract: 10
2.18 Paper 18: 5G Internet of Things: a survey 10
2.18.1 Author: 10
2.18.2 Abstract: 10
2.19 Paper 19: Internet of Things (IoT): opportunities, issues and challenges towards a smart and sustainable future 11
2.19.1 Author: 11
2.19.2 Abstract: 11
2.20 Paper 20: Research on security of information sharing in internet of things based on key algorithm 11
2.20.1 Author: 11
2.20.2 Abstract: 11

3 Analysis / Software Requirements Specification (SRS) 12
3.1 User Interfaces 12

3.2 Hardware Interfaces . . . 12
3.3 Software Interfaces . . . 15
3.4 Communications Interfaces . . . 16

4 System Design **17**
4.1 System Overview . . . 17
4.2 Functional Requirements . . . 18
4.3 Non-Functional Requirements . . . 19
4.4 External Interfaces . . . 20
4.5 Limitations . . . 21

5 Methodology **23**
5.1 System Design . . . 23
5.2 Development . . . 23
5.3 Testing . . . 24
5.4 Deployment . . . 24
5.5 Evaluation . . . 25

6 Implementation **26**
6.1 Frontend Development . . . 26
6.2 Backend Development: . . . 28
6.3 IoT Device Configuration: . . . 29
6.4 Database Setup: . . . 31
6.5 Integration of External Services: . . . 32

7 Testing and Quality Assurance **33**
7.1 Unit Testing: . . . 33
7.2 Integration Testing: . . . 35
7.3 System Testing: . . . 36
7.4 User Acceptance Testing (UAT): . . . 36

8 Conclusion and Future Work **37**

8.1 Conclusion: 37

8.2 Future Work 37

9 References **39**

Chapter 1

Introduction

1.1 Purpose

The ”Smart Goods Transfer Using an IoT Box” project is designed to meet the increasing need for a secure, efficient, and user-friendly delivery system in today’s digital landscape. As online shopping and e-commerce continue to dominate, traditional delivery systems often fail to meet the high standards expected by both customers and businesses. By integrating cutting-edge Internet of Things (IoT) technology, this project seeks to transform the goods transfer process. It offers a seamless and reliable platform that enhances the overall delivery experience while ensuring the protection and safety of shipments.

This project also aims to connect technological advancements with real-world applications in the logistics industry. By developing an IoT Box featuring state-of-the-art sensors, micro-controllers, communication modules, and GPS capabilities, we showcase how IoT can reshape traditional delivery practices. We hope this initiative will inspire further exploration of IoT solutions in logistics, promoting more efficient, sustainable, and customer-focused delivery methods moving forward.

1.2 Intended Audience

This project is designed for a wide range of stakeholders, including:

- E-commerce businesses looking to enhance their delivery efficiency.
- Consumers who prioritize secure, fast, and reliable delivery options.
- Logistics companies interested in adopting cutting-edge technologies for goods transportation.
- Innovators and researchers exploring the role of IoT in the logistics sector.
- Supply chain experts focused on streamlining delivery operations through technological advancements.
- Tech enthusiasts curious about the integration of IoT in logistics and supply chain solutions.

1.3 Reading Suggestions

To gain a deeper understanding of the key aspects of our project and its impact, we suggest the following readings:

- ”The Internet of Things: Key Concepts and Applications” by Samuel Greengard – This book offers valuable insights into the basics of IoT technology and its wide-ranging uses across various industries.
- ”Logistics and Supply Chain Management” by Martin Christopher – A comprehensive guide to logistics, highlighting how technological advancements are transforming modern supply chain operations.
- ”Designing Connected Products: UX for the Consumer Internet of Things” by Claire Rowland, Elizabeth Goodman, Martin Charlier, and Ann Light – This book dives into the design aspects of IoT devices, with a focus on creating intuitive user experiences, which is relevant to our project’s interface design.
- ”Security in the Internet of Things: Implementing the CIA Triad” by Sean Tyson – A detailed examination of security considerations in IoT systems, addressing critical issues like confidentiality, integrity, and availability to ensure the security of our solution.

1.4 Product Scope

Our project will cover the following areas:

- Designing and creating a functional prototype of the IoT Box, equipped with advanced sensors, microcontrollers, communication modules, and GPS capabilities.
- Building a user-friendly web application that allows customers to book, monitor, and manage their shipments securely and efficiently.
- Implementing real-time updates through email and SMS notifications, keeping users informed at every stage of the delivery process.
- Incorporating strong security features like QR code-based authentication and GPS tracking to ensure the safety of goods during transportation.
- Offering a convenient recipient verification system, enabling smooth and secure completion of the delivery cycle.

Chapter 2

Literature Survey

2.1 Paper 1: Internet of Things (IoT): A vision, architectural elements, and future directions

2.1.1 Author:

Jayavardhana Gubbi, et al.

2.1.2 Abstract:

- The paper presents a comprehensive vision for the Internet of Things (IoT), outlining its architectural elements and future directions.
- IoT is described as an integrated network of physical devices, vehicles, buildings, and other objects embedded with electronics, software, sensors, and connectivity.
- The authors highlight the significance of IoT in enhancing the efficiency of various systems and improving quality of life.
- The architecture of IoT is divided into three layers: perception, network, and application layers, each serving distinct roles in the ecosystem.
- The paper discusses challenges such as security, privacy, and interoperability that must be addressed for successful IoT implementation.
- Future directions include the need for scalable solutions, standardization of protocols, and the integration of IoT with emerging technologies like cloud computing and big data analytics.

2.2 Paper 2: Internet of Things (IoT) in logistics: A comprehensive literature review

2.2.1 Author:

Anupam Mishra, et al.

2.2.2 Abstract:

- This paper reviews the existing literature on the applications of IoT in logistics.
- It identifies key trends, challenges, and future research directions within the logistics domain.
- The authors highlight how IoT technologies enhance supply chain visibility, efficiency, and decision-making.
- The review categorizes IoT applications in logistics into tracking, monitoring, and management systems.
- Challenges include data security, integration with existing systems, and the need for standardized protocols.
- The paper concludes with suggestions for future research to address identified gaps and enhance IoT implementation in logistics.

2.3 Paper 3: A systematic review of Internet of Things (IoT) applications in supply chain management (SCM)

2.3.1 Author:

Ali Rezaei, Mohammad Mohebbi.

2.3.2 Abstract:

- This paper systematically reviews the applications of IoT in supply chain management (SCM).
- It provides a structured overview of IoT technologies and their impact on SCM processes.
- The authors categorize IoT applications into areas such as inventory management, transportation, and warehouse management.
- The paper discusses the benefits of IoT, including improved operational efficiency, enhanced data accuracy, and real-time decision-making capabilities.
- Key challenges identified include the high cost of implementation, data privacy concerns, and the need for interoperability among systems.
- The authors propose a research agenda to address these challenges and explore future IoT innovations in SCM.

2.4 Paper 4: Internet of Things for smart logistics: Technologies and applications

2.4.1 Author:

Zhonghua Sun, et al.

2.4.2 Abstract:

- This paper explores the role of IoT in smart logistics, emphasizing its technologies and applications.
- It discusses how IoT enhances logistics operations through automation, data analytics, and real-time tracking.
- The authors outline various IoT technologies, including RFID, GPS, and cloud computing, and their integration into logistics systems.
- Applications discussed include fleet management, asset tracking, and predictive maintenance.
- The paper highlights challenges such as system integration, data security, and technology adoption barriers.
- It concludes with future trends in smart logistics driven by IoT advancements and potential areas for further research.

2.5 Paper 5: Research on Internet of Things architecture for logistics monitoring

2.5.1 Author:

Yawen Yao, et al.

2.5.2 Abstract:

- This paper presents a novel IoT architecture for logistics monitoring, aiming to enhance operational efficiency.
- The proposed architecture integrates various technologies such as sensors, cloud computing, and big data analytics.
- The authors detail how this architecture can improve visibility, tracking, and management of logistics operations.
- The paper discusses the design and implementation of the architecture, highlighting its advantages over traditional systems.
- Challenges such as scalability, data management, and integration with existing logistics frameworks are addressed.
- The authors provide recommendations for future research to optimize IoT architectures in logistics.

2.6 Paper 6: Internet of Things for smart cities

2.6.1 Author:

Andrea Zanella, et al.

2.6.2 Abstract:

[left=0pt]This paper explores the integration of IoT technologies within smart cities, emphasizing the benefits and challenges. It outlines the architectural components and key technologies that enable smart city applications. The authors discuss various use cases, including smart transportation, energy management, and urban planning. The paper highlights the importance of data analytics and communication technologies in realizing smart city initiatives. Security and privacy concerns related to IoT in smart cities are critically examined. Future directions for research in smart city IoT applications are suggested.

2.7 Paper 7: Smart logistics for industry 4.0: A review

2.7.1 Author:

Xue Li, et al.

2.7.2 Abstract:

[left=0pt]This review paper discusses the role of IoT in transforming logistics within the context of Industry 4.0. The authors summarize key technologies that facilitate smart logistics, including IoT, big data, and artificial intelligence. Various applications in supply chain management are highlighted, showcasing improvements in efficiency and visibility. The paper addresses challenges such as interoperability, standardization, and data security. Recommendations for implementing smart logistics solutions are provided. The authors call for further research to explore the potential of IoT in logistics automation.

2.8 Paper 8: Smart logistics in the context of Industry 4.0: A review

2.8.1 Author:

Bin Jiang, et al.

2.8.2 Abstract:

[left=0pt]This paper reviews the evolution of logistics in the context of Industry 4.0, focusing on smart technologies. The authors analyze the impact of IoT on logistics operations and supply chain management. The paper categorizes various IoT applications in logistics, including tracking, monitoring, and predictive analytics. Challenges related to technology adoption and integration within existing logistics systems are discussed. The authors propose a framework for implementing smart logistics solutions leveraging IoT technologies. Future research directions are outlined to enhance the effectiveness of smart logistics.

2.9 Paper 9: Smart logistics for perishable products based on Internet of Things

2.9.1 Author:

Fang Jia, et al.

2.9.2 Abstract:

[left=0pt]This paper investigates the use of IoT for managing logistics of perishable products, focusing on real-time monitoring. The authors propose a system architecture that integrates IoT sensors for tracking temperature and humidity. Case studies illustrate the effectiveness of the proposed system in maintaining product quality during transportation. Challenges related to data collection and analysis in perishable product logistics are identified. The authors emphasize the importance of collaboration between stakeholders in the supply chain. Suggestions for further research to improve IoT applications in perishable product logistics are provided.

2.10 Paper 10: A review on the application of Internet of Things in logistics

2.10.1 Author:

Marco Machado, et al.

2.10.2 Abstract:

[left=0pt]This review paper provides a comprehensive overview of IoT applications in logistics and supply chain management. The authors categorize IoT applications into areas such as inventory management, tracking, and transportation. Key benefits of implementing IoT in logistics, including cost reduction and efficiency improvement, are discussed. The paper highlights challenges such as data privacy, integration issues, and the need for standardization. Recommendations for future research in the field of IoT logistics are presented. The authors call for collaborative efforts between academia and industry to advance IoT applications in logistics.

2.11 Paper 11: The internet of things: a survey

2.11.1 Author:

Shancang Li, Li Da Xu, Shanshan Zhao

2.11.2 Abstract:

- This paper provides a comprehensive survey on the Internet of Things (IoT), covering its definitions, technologies, and applications.

- The authors discuss the challenges and opportunities presented by IoT in various sectors.
- The paper highlights the importance of IoT architecture and protocols in enabling efficient data communication.
- Various case studies are presented to illustrate successful IoT implementations.
- Future trends in IoT technology and its potential impact on society are examined.

2.12 Paper 12: Online Structural Health Monitoring and Parameter Estimation for Vibrating Active Cantilever Beams Using Low-Priced Microcontrollers

2.12.1 Author:

G. Takacs, J. Vachálek, B. Rohal'-Ilkiv

2.12.2 Abstract:

- This paper discusses the development of a structural health monitoring system using low-cost microcontrollers.
- The authors present a method for real-time monitoring and parameter estimation of vibrating cantilever beams.
- The effectiveness of the proposed system is demonstrated through experimental results.
- The paper explores the potential applications of the monitoring system in civil engineering and infrastructure management.
- Limitations and areas for future research are identified.

2.13 Paper 13: Arduino core for ESP32 WiFi chip

2.13.1 Author:

GitHub

2.13.2 Abstract:

- This resource provides comprehensive information about the Arduino core for the ESP32 WiFi chip.
- The repository includes libraries and examples for building IoT applications using ESP32.
- Users can find detailed documentation on installation and usage.
- The resource is widely used by developers for creating various IoT projects.
- Updates and community contributions enhance the core's functionality and performance.

2.14 Paper 14: Espressif IoT Development Framework

2.14.1 Author:

GitHub

2.14.2 Abstract:

- This resource presents the official Espressif IoT Development Framework for IoT applications.
- The framework supports development for ESP32 and ESP8266 chips, facilitating rapid application development.
- The repository includes examples, libraries, and APIs for building robust IoT solutions.
- The framework aims to simplify the development process for developers and enthusiasts.
- Ongoing updates ensure compatibility with the latest IoT technologies.

2.15 Paper 15: The Internet of Things (IoT): Applications, investments, and challenges for enterprises

2.15.1 Author:

Lee, J.

2.15.2 Abstract:

- This paper explores various applications of IoT across different industries and sectors.
- The author discusses the investments required for implementing IoT solutions and their potential return on investment.
- Challenges related to IoT adoption, including security and interoperability, are highlighted.
- The paper provides insights into future trends and the evolving landscape of IoT technology.
- Recommendations for enterprises looking to leverage IoT are offered.

2.16 Paper 16: Internet of Things (IoT): A vision, architectural elements, and future directions

2.16.1 Author:

Gubbi, J., et al.

2.16.2 Abstract:

- This paper presents a vision for the Internet of Things (IoT) and its architectural components.
- The authors discuss the fundamental elements required for building an effective IoT ecosystem.
- Key challenges and future directions in IoT research are outlined.
- The paper emphasizes the importance of interoperability and standardization in IoT deployments.
- A case study is provided to illustrate practical IoT applications in smart cities.

2.17 Paper 17: A review of logistics Internet-of-Things: Current trends and scope for future research

2.17.1 Author:

Héris Golpîra

2.17.2 Abstract:

- This review paper examines current trends in logistics IoT and identifies gaps in the existing literature.
- The author explores various applications of IoT in supply chain management and logistics.
- Challenges related to technology adoption, data security, and system integration are discussed.
- Future research directions are proposed to enhance the efficiency and effectiveness of logistics IoT.
- The paper highlights the need for collaborative research across multiple disciplines.

2.18 Paper 18: 5G Internet of Things: a survey

2.18.1 Author:

S. Li, et al.

2.18.2 Abstract:

- This survey investigates the impact of 5G technology on the Internet of Things (IoT).
- The authors discuss the key features of 5G that enable enhanced connectivity for IoT devices.
- Various application scenarios in smart cities, healthcare, and industrial IoT are explored.
- The paper highlights challenges related to security, scalability, and deployment of 5G IoT solutions.
- Future research opportunities in the intersection of 5G and IoT are identified.

2.19 Paper 19: Internet of Things (IoT): opportunities, issues and challenges towards a smart and sustainable future

2.19.1 Author:

S. Nižetić, et al.

2.19.2 Abstract:

- This paper discusses the opportunities and challenges presented by IoT in achieving sustainability.
- The authors analyze the potential of IoT technologies in various sectors, including energy and transportation.
- Key issues such as data privacy, security, and ethical considerations are examined.
- The paper emphasizes the need for robust policies and frameworks to support sustainable IoT development.
- Future directions for research are suggested to address the identified challenges.

2.20 Paper 20: Research on security of information sharing in internet of things based on key algorithm

2.20.1 Author:

P. Wei, et al.

2.20.2 Abstract:

- This paper investigates security challenges related to information sharing in IoT environments.
- The authors propose a key algorithm to enhance data security during transmission.
- The effectiveness of the proposed approach is validated through experiments and simulations.
- The paper highlights the importance of secure data sharing for the success of IoT applications.
- Recommendations for future research on IoT security are provided.

Chapter 3

Analysis / Software Requirements Specification (SRS)

3.1 User Interfaces

The ”Smart Goods Transfer using IoT Box” project features several user interfaces designed for ease of use and efficient interaction:

1. **Web Application:** Users engage with a user-friendly web application to reserve IoT boxes, track their package deliveries, and receive real-time notifications. This platform includes straightforward booking forms, interactive maps for tracking shipment progress, and timely alerts to keep users informed about their packages.

2. **IoT Box Interface:** The IoT box is equipped with an intuitive interface for recipients to easily retrieve their packages. This interface may feature a camera for scanning QR codes, ensuring a seamless and secure package collection process.

3.2 Hardware Interfaces

The IoT Box is equipped with various hardware interfaces designed to enhance functionality and efficiency:

1. **Microcontroller:** Utilizing an ESP32 or Arduino to manage operations and processing within the IoT Box.
   - The ESP32 is available in various modules and development boards, making it easy to integrate into your projects.
   - It is supported by a large and active community, providing extensive resources and support.
   - The ESP32 is a great choice for IoT applications that require wireless connectivity, low power consumption, and a wide range of peripherals.
   - It is also used for communication between the IoT box and the server, facilitating data transfer and remote control.

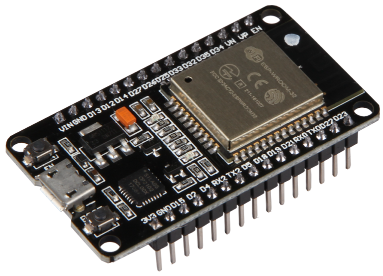


Figure 3.1: Microcontroller used in the IoT Box

2. **GPS Module:** Equipped with GPS modules like the NEO-6M or SIM808 to enable real-time tracking of the IoT Box's location.

    - It is a cost-effective option compared to other GPS modules, making it accessible for a wide range of projects.
    - The module is relatively simple to integrate into various systems, requiring minimal setup and configuration.
    - It provides reliable GPS positioning information, making it suitable for applications that require accurate location data.
    - The NEO-6M can be used in a variety of applications, including GPS tracking, navigation, location-based services, and geospatial data collection.
    - Its low power consumption makes it ideal for battery-powered devices, such as GPS trackers and wearable devices.

Figure 3.2: GPS module for location tracking

3. **QR Code Scanner:** Integrating an ESP32-CAM module for efficient scanning of QR codes, facilitating seamless package retrieval and verification.

   - The ESP32-CAM is a compact and cost-effective module that combines an ESP32 chip with a camera, making it ideal for IoT applications.
   - It supports a wide range of resolutions, with the camera capable of capturing images up to 2 megapixels (1600 x 1200 pixels).
   - The module features built-in Wi-Fi and Bluetooth capabilities, enabling wireless communication and remote access.
   - It has a microSD card slot for storing images and videos, allowing for expanded memory options.
   - The ESP32-CAM includes a variety of GPIO pins, facilitating connection to other sensors and actuators for diverse applications.
   - It supports various camera configurations and is compatible with libraries such as ESP32 Camera and ESP-IDF for easy integration and programming.
   - The module operates at a voltage range of 5V, making it compatible with standard power supplies for development projects.
   - It includes a built-in flash LED for better illumination in low-light conditions during image capture.

Figure 3.3: ESP32 Cam used for QR Code Scanning

4. **Servo Motor:** Implementing a servo motor to control the opening and locking mechanism of the IoT Box, enhancing security and user convenience.

   - The SG90 is small and lightweight, typically measuring about 22.5 x 11.5 x 31 mm and weighing around 9 grams. This makes it suitable for projects with limited space.
   - It provides a torque range of approximately 1.2 kg·cm at 4.8V and up to 1.5 kg·cm at 6V, making it capable of handling light loads effectively.
   - The operating voltage ranges from 4.8V to 6V, allowing for compatibility with a variety of power sources.
   - The servo motor has an approximate rotation range of 180 degrees, enabling versatile movement in various applications.

- It utilizes PWM (Pulse Width Modulation) for control, making it easy to interface with microcontrollers.
- The SG90 features a built-in feedback mechanism that ensures accurate positioning and movement.

Figure 3.4: Servo motor controlling the IoT Box mechanism

5. **Power Supply:** He system includes a voltage regulator to ensure a stable voltage supply to the ESP32 and other components, protecting them from power fluctuations.
   - The setup allows for easy recharging of the 1000mAh battery using a USB connection, making it convenient to maintain power during long-term use.
   - An integrated battery level monitoring system provides real-time feedback on the battery status, ensuring users are informed about the remaining power and can recharge accordingly.
   - The ESP32 is designed for low power consumption, allowing the system to operate efficiently and extend battery life, especially during idle periods.

3.3 Software Interfaces

1. **Backend Server:**
   - **Programming Languages:** Utilizes JavaScript (Node.js) and Python for robust server-side operations.
   - **Frameworks:** Built on Express.js to streamline web application development.
   - **Database Management System:** Employs MongoDB for efficient data storage and retrieval.
   - **Communication Protocols:** Utilizes HTTP for seamless communication between the server and client.
2. **IoT Box Firmware:**
   - **Programming Languages:** Developed in C/C++ for efficient hardware control.
   - **Integrated Development Environment (IDE):** Programmed using the Arduino IDE for intuitive coding and debugging.
   - **Communication Protocols:** Implements HTTP to enable connectivity with the backend server.
   - **Libraries:** Leverages Arduino libraries to facilitate sensor interfacing and communication with various modules.

3.4 Communications Interfaces

Communication interfaces play a crucial role in facilitating data exchange among various system components:

1. **Web-Based Application and Backend Server:** Data transfer occurs through HTTP requests, employing JSON or XML formats to ensure seamless communication.
2. **IoT Boxes and Backend Server:** Communication is established via HTTP protocols, where payloads include vital sensor readings and updates on package statuses.
3. **Integration with E-commerce Platforms:** RESTful APIs are utilized for smooth integration, allowing for the transmission of JSON or XML payloads over HTTP to maintain data consistency.

Chapter 4

System Design

4.1 System Overview

The Smart Goods Transfer system represents a cutting-edge web-based platform designed to optimize the transfer of goods between online shoppers and merchants. By harnessing the power of Internet of Things (IoT) technology, this system creates an efficient and user-friendly experience for users to reserve IoT-enabled boxes, securely deposit items, and track their deliveries in real time.

Central to the system are several integral components: IoT boxes equipped with advanced sensors and microcontrollers, a robust web application for managing reservations and shipment tracking, and sophisticated communication modules that facilitate notifications for both senders and recipients. This cohesive architecture automates key delivery processes, enhancing transparency, security, and dependability throughout the logistics chain.

Key functionalities of the Smart Goods Transfer system encompass:

- **Reservation Management:** Users can easily book specific IoT boxes through the web interface, providing essential delivery details and personal preferences.
- **Item Detection and Authentication:** Each IoT box is equipped with advanced sensors that detect deposited items and create unique QR codes, ensuring secure locking mechanisms.
- **Real-time Tracking:** With integrated GPS modules, users can monitor the precise location of their packages in real-time, allowing senders to conveniently track deliveries through the web application.
- **Automated Notifications:** The system proactively sends out automated notifications to both senders and recipients during critical phases of the delivery process, ensuring transparency and timely updates.
- **Access Control:** Recipients can easily access their items by scanning the generated QR code using the built-in camera of the IoT box, facilitating a smooth retrieval process.

4.2 Functional Requirements

1. **Reservation Management:**
    - The system must allow users to reserve IoT boxes through a user-friendly web application.
    - Senders will be able to provide essential delivery details, including recipient information and shipping address.
    - The system will dynamically display only available IoT boxes based on real-time inventory data, ensuring accurate reservations.
2. **Item Placement by Sender**
    - The sender presents the generated QR code to the camera integrated within the IoT box.
    - Upon successful recognition of the QR code, the IoT box unlocks.
    - The sender places the item intended for delivery inside the box.
    - After the item is securely placed, the IoT box automatically closes and locks itself after a 30-second countdown.
3. **Real-time Tracking:**
    - GPS modules integrated into the IoT boxes will enable the system to offer real-time tracking of packages throughout the entire delivery process.
    - The system will offer an intuitive web interface for senders and recipients to easily track the location and status of their shipments.
4. **Automated Notifications:**
    - Notifications will be automatically sent to both senders and recipients via email or SMS, keeping them updated at all times.
    - Notifications will be triggered based on significant events such as reservation confirmations, package pickup, transit, and final delivery.
    - Users will have the option to personalize their notification settings, controlling both the frequency and delivery method of these updates.
5. **Access Control:**
    - Recipients will scan the unique QR code generated by the system using the camera on the IoT box to access their packages.
    - The system will validate the QR code, granting access if authenticated successfully.
    - In the case of expired or invalid QR codes, alternative options, such as customer support contact or initiating a return, will be made available.

6. **Return Process:**

   - The system will allow the return of IoT boxes after delivery completion, providing a seamless process for senders and recipients.
   - Recipients may choose to return IoT boxes to designated drop-off points or request a collection service by delivery personnel.
   - The return process will be fully trackable through the web application, with notifications sent upon successful returns.

7. **Administration and Management:**

   - The system will offer robust administrative tools for managing the IoT box inventory, handling user accounts, and configuring delivery parameters.
   - Administrators will have access to real-time reservation data, shipment tracking, and performance reports to improve operational efficiency.
   - User accounts will be protected using authentication mechanisms like secure passwords or multi-factor authentication to ensure unauthorized access is prevented.

4.3 Non-Functional Requirements

1. **Performance:**

   - The system must ensure fast response times for user interactions to enhance the overall user experience.
   - It should efficiently manage multiple user requests simultaneously without sacrificing performance, even during high-traffic periods.
   - The architecture should support scalability to adapt to future increases in both user numbers and transaction loads.

2. **Security:**

   - All data exchanges between the web application and IoT devices must be encrypted using secure protocols (e.g., HTTPS) to safeguard against unauthorized access or data manipulation.
   - Robust user authentication and access control measures must be in place to restrict sensitive information access and privileged actions to authorized individuals only.
   - IoT devices should be fortified against physical tampering and unauthorized access to prevent theft or alterations during transit.

3. **Reliability:**

   - The system must exhibit high reliability, minimizing downtime to ensure uninterrupted service availability.
   - Routine backups of essential data should be conducted to protect against data loss due to system failures or unexpected incidents.

- Redundancy and failover strategies should be integrated at both hardware and software levels to mitigate the effects of potential disruptions.

4. **Usability:**

- The user interface of the web application must be designed to be intuitive and user-friendly, featuring clear navigation and informative feedback to assist users.
- Accessibility features must be incorporated to enable users with disabilities to navigate and use the system effectively.
- The system should offer multi-language support to cater to a diverse user base from various linguistic backgrounds.

5. **Scalability:**

- The system architecture must allow for both horizontal and vertical scaling to accommodate growth in user traffic and data management.
- Leveraging cloud infrastructure and containerization tools (e.g., Docker) will facilitate easy scalability and optimal resource allocation.

6. **Maintainability:**

- The system should be structured with modular, well-documented code to simplify maintenance and enable future enhancements.
- Automated testing and deployment processes should be implemented to expedite the development and release cycles, allowing for quick iterations and updates.

4.4 External Interfaces

1. **Email Integration:**

- The system shall be designed to connect with email service providers to automatically dispatch notifications to users at key phases of the delivery process, such as confirming reservations, providing shipment tracking updates, and notifying delivery completion.
- Utilizing email APIs (e.g., SMTP) will facilitate smooth communication between the system and the users' email accounts.

2. **GPS Tracking Services:**

- The system will utilize GPS services to monitor the real-time location of IoT-enabled delivery boxes throughout the entire delivery process.
- By integrating with GPS APIs (e.g., Google Maps API, Mapbox API), the system will be able to retrieve and visually display location information on the web application interface.

3. **Payment Processing Interfaces:**

   - The system will connect with payment gateway services to enable secure online transactions for reservation fees and delivery charges.
   - Integration with payment gateway APIs (e.g., PayPal, Stripe) will provide users with various payment options, including credit/debit cards and digital wallets.

4. **IoT Device Communication Protocols:**

   - The system shall establish communication with IoT devices, including sensors and microcontrollers embedded within delivery boxes, to facilitate item detection, QR code generation, and access management.
   - Integrating IoT communication protocols and standards (e.g.HTTP, MQTT, CoAP) will enable bi-directional data exchange between the system and the IoT devices, ensuring efficient data flow and operational control.

4.5 Limitations

1. **Hardware Constraints:**

   - The capabilities of the system may be limited by the hardware specifications of the IoT devices utilized in the delivery boxes. Factors such as processing capabilities, memory capacity, and battery longevity could potentially affect overall system performance and functionality.

2. **Dependence on Network Connectivity:**

   - Reliable internet connectivity is crucial for the operation of the system, facilitating communication among the web application, IoT devices, and third-party services. In regions with poor network coverage or inconsistent connectivity, the performance of the system may suffer, resulting in delays or interruptions in service.

3. **Impact of Environmental Factors:**

   - The performance of the system could be influenced by environmental conditions, including temperature fluctuations, humidity levels, and physical barriers. Extreme weather or adverse environmental conditions might impair the functionality and longevity of the IoT devices, leading to possible operational failures.

4. **Compatibility Challenges:**

   - The system may face compatibility issues with older or legacy systems and outdated hardware and software. Interoperability between the Smart Goods Transfer system and external devices or platforms may pose challenges, potentially complicating integration and data sharing, and necessitating additional development efforts to resolve.

5. **Scalability Constraints:**

   - The ability of the system to scale may be restricted by limitations in its infrastructure resources, including server capacity, available storage, and bandwidth. A sudden increase in user demand or data traffic could surpass the system's scalability thresholds, resulting in performance decline or interruptions in service delivery.

Chapter 5

Methodology

5.1 System Design

- The initial phase focuses on establishing a comprehensive framework for the system. This includes defining the overall system architecture, functionalities, and specific requirements that the system must fulfill. Key design decisions encompass the integration of IoT devices, selection of communication protocols, development of user interfaces, and strategies for data management. A collaborative approach is employed to ensure that all stakeholders contribute to a well-rounded design that addresses both technical and user-centric perspectives.
- Detailed analysis is performed to assess the scalability and flexibility of the system. This ensures that the architecture is capable of adapting to future requirements and integrating new technologies without significant rework. Provisions for system security and data privacy are also incorporated into the design to safeguard against potential vulnerabilities.
- Consideration is given to the system's energy efficiency and cost-effectiveness. Design choices are made to optimize the power consumption of IoT devices and other components, which is crucial for long-term sustainability, especially in battery-powered applications.
- During the design phase, system interfaces are clearly defined to ensure smooth interaction between hardware and software components. Additionally, the physical layout and design of IoT hardware are considered to optimize signal transmission and device placement. Redundancy and failover mechanisms are also introduced to improve system resilience, ensuring minimal downtime in case of component failures.

5.2 Development

- This phase is characterized by a detailed design process that translates requirements into tangible system architecture, components, and interfaces. Various techniques are employed to enhance clarity and understanding, including architectural modeling to define system structure, use case diagrams to capture user interactions, and sequence diagrams to illustrate system behavior over time. This stage emphasizes coding best practices, version control, and collaborative development to foster an environment conducive to quality output.

- Different development methodologies such as Agile or Scrum are considered to ensure flexibility in project management. Iterative development allows for continuous refinement of features based on feedback from stakeholders and team members.
- Emphasis is placed on modular design, where each component of the system is developed and tested independently before integration. This modularity enhances maintainability and simplifies future updates or expansions of the system.
- The development phase also includes the implementation of comprehensive error handling mechanisms and logging systems to track system activities and failures. Continuous integration (CI) tools are set up to automatically run tests and deploy changes, ensuring that the development process is smooth and efficient. Developers collaborate closely with designers to ensure that the system's user interface aligns with functional goals.

5.3 Testing

- Rigorous testing procedures are implemented to validate the functionality, performance, and reliability of the system. This includes conducting unit testing to ensure individual components perform as expected, integration testing to verify that components work together seamlessly, and system testing to assess the overall behavior of the entire system. Test cases are meticulously designed to cover a wide range of scenarios, enabling the identification and resolution of defects or inconsistencies in system behavior. Continuous feedback loops are established to enhance the testing process.
- Load and stress testing are conducted to evaluate how the system behaves under peak demand or extreme conditions. This helps identify potential performance bottlenecks and ensures the system can handle real-world usage at scale.
- Automated testing tools are utilized to streamline repetitive test cases, ensuring that any modifications in the codebase do not introduce regressions or unexpected issues.
- In addition to functional tests, security testing is conducted to ensure that the system is protected against unauthorized access or malicious attacks. Penetration testing and vulnerability scanning are carried out to identify potential security loopholes. Performance monitoring tools are also integrated to measure resource usage, including memory and processing time, during testing, ensuring that the system remains efficient.

5.4 Deployment

- After successful testing, the system is deployed into a production environment. This critical phase involves multiple steps, including setting up servers, configuring network connections, installing necessary software dependencies, and ensuring compatibility with external services. A phased deployment strategy may be employed to mitigate risks and allow for incremental validation of system performance in real-world scenarios. Documentation and user training sessions are also part of this stage to prepare end-users for effective system utilization.

- A comprehensive backup and rollback plan is established before deployment to ensure that any issues arising during deployment can be quickly addressed without disrupting system functionality.
- Post-deployment monitoring tools are set up to continuously track system performance and identify potential issues early. This enables quick remediation and enhances system reliability.
- During deployment, attention is also given to load balancing and scalability to handle variable demand efficiently. Cloud services may be employed to enable auto-scaling based on traffic. Additionally, deployment scripts and tools like Docker or Kubernetes are used to streamline the process and ensure consistency across different environments.

5.5 Evaluation

- The final phase is dedicated to evaluating the system's performance and gathering valuable feedback from users. Key metrics, such as delivery times, user satisfaction ratings, and system reliability, are assessed to measure the success and effectiveness of the project. User feedback is collected through surveys and interviews, providing insights into areas for improvement. Based on the evaluation results, necessary adjustments and enhancements are implemented to optimize system performance and elevate user experience, ensuring that the system evolves in line with user needs and technological advancements.
- An in-depth post-implementation review is conducted to assess the project's return on investment (ROI) and the extent to which the system's objectives have been met. Any lessons learned are documented for future reference.
- Future maintenance plans are developed to ensure ongoing support, including scheduled system updates, security patches, and any necessary hardware replacements.
- Continuous evaluation is performed through real-time analytics and monitoring of system KPIs (Key Performance Indicators). This data-driven approach helps identify areas where system improvements can be made. Additionally, user satisfaction surveys are periodically distributed to track changes in user perception over time, ensuring that the system continues to meet evolving user needs.

Chapter 6

Implementation

The implementation stage centered on transforming the design specifications into a fully operational system by writing code, configuring settings, and integrating multiple components. This phase emphasized the hands-on development process, ensuring that the system was constructed to align with the established requirements and design blueprint.

6.1 Frontend Development

- The frontend of the web application was built using modern technologies such as HTML, CSS, and JavaScript frameworks like React, ensuring a robust and maintainable codebase. The application prioritizes a clean, intuitive user interface (UI) that enhances user experience (UX) while incorporating several key features:
  - **Reservation Management:** Users can seamlessly make and manage reservations, with real-time updates and confirmations. This feature is designed to minimize friction in the user journey, ensuring a streamlined booking process.
  - **Package Tracking:** An integrated tracking system allows users to monitor their packages with real-time updates, enhancing transparency and trust. This feature provides detailed status updates from dispatch to delivery, contributing to an overall positive experience.
  - **User Authentication and Profile Management:** Secure authentication systems were implemented using industry best practices, allowing users to create, edit, and manage their profiles efficiently. Users can easily update their personal information, track their order history, and maintain control over their data.
- For administrators, a dedicated **Admin Dashboard** has been developed. This powerful tool allows admins to manage users, products, and orders through an organized interface. Admins can monitor key metrics, manage inventories, and resolve customer issues, contributing to the smooth operation of the platform.
- The **Delivery Dashboard**, designed for delivery personnel, provides an interface where they can view assigned orders, mark deliveries as completed, and update order statuses. This feature streamlines the workflow for delivery personnel and ensures timely updates for customers and administrators alike.

- Although the user interface was designed with attention to detail, it is important to note that full responsiveness and accessibility across all devices have not yet been fully achieved. Currently, the application functions well on larger screens such as desktops. However, optimizations are still needed to ensure smooth functionality and visually appealing layouts on smaller screens, including tablets and smartphones.
- To improve this, upcoming development phases will focus on enhancing the application's responsiveness to provide a seamless and consistent experience across different screen sizes and orientations. The goal is to ensure that users can interact with the platform without any difficulty, regardless of the device they are using.
- Moreover, performance has been a key focus in the development process. Interactive components such as forms, modals, and navigation menus have been optimized for fast load times and smooth transitions. This ensures that users can interact with the platform without experiencing delays, contributing to an overall seamless and responsive experience, even before full mobile optimization is achieved.
- Despite these improvements, the team recognizes the need to implement accessibility features in the next development cycle. Ensuring the platform is usable by individuals with varying abilities, including screen reader compatibility and keyboard navigation support, will be a high priority moving forward. These enhancements will help broaden the platform's reach and ensure that it is compliant with accessibility standards, contributing to a more inclusive user experience.

Figure 6.1: Landing Page

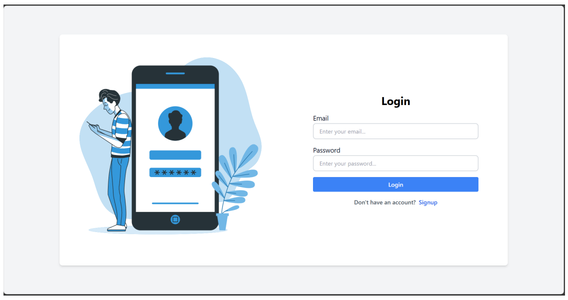


Figure 6.2: Login Page

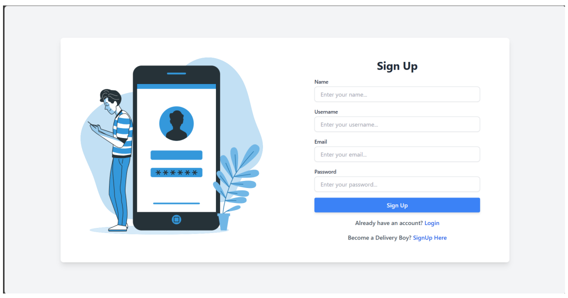


Figure 6.3: Sign Up Page (Version 1)

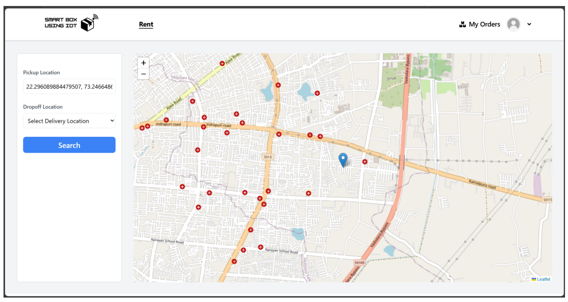


Figure 6.4: User Dashboard

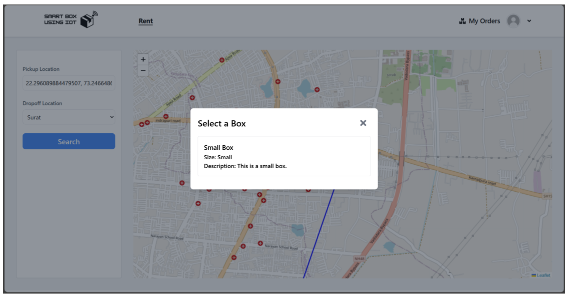


Figure 6.5: Box Selection

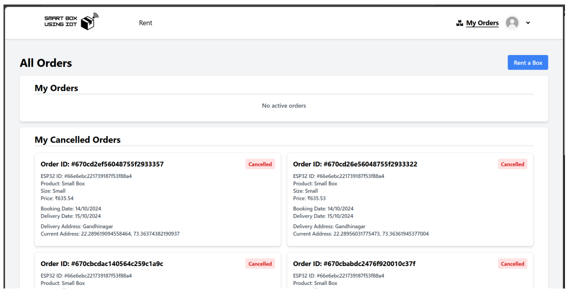


Figure 6.6: Orders Page

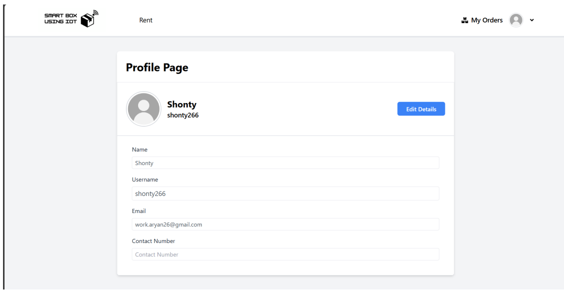


Figure 6.7: User Profile

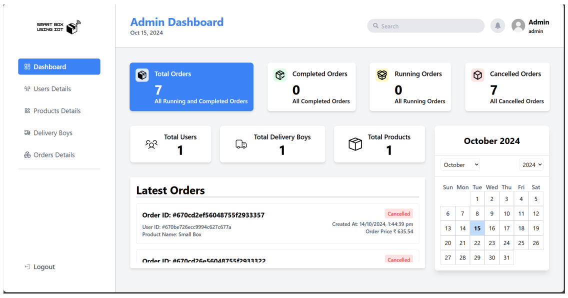


Figure 6.8: Admin Dashboard

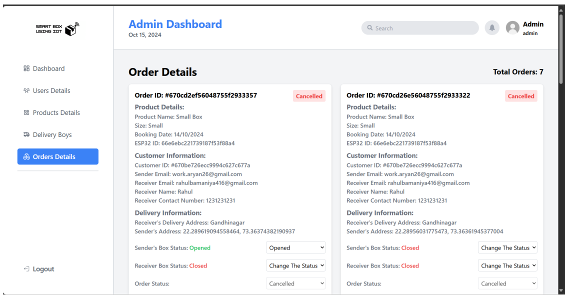


Figure 6.9: Admin Orders Panel

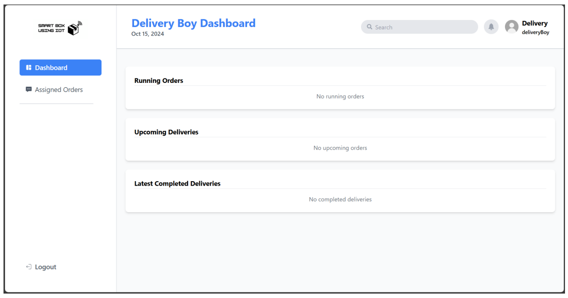


Figure 6.10: Delivery Personnel Dashboard

6.2 Backend Development:

- The backend of the web application was implemented using **Node.js** with the **Express** framework. This choice was made due to Node.js's non-blocking architecture, which is well-suited for handling multiple simultaneous requests.
- It was responsible for handling:
  - All server-side logic, ensuring that business rules are consistently applied across the application.
  - Database interactions, utilizing an efficient database management system to store and retrieve user data securely.
  - API integrations, allowing seamless communication with external services and third-party APIs for enhanced functionality.
- This robust and scalable infrastructure efficiently manages:

  - User requests, providing quick responses even under heavy load.
  - Data processing, including validation and transformation to maintain data integrity.
  - Secure transactions between the frontend and the database, implementing authentication and authorization protocols to protect sensitive information.

- Express provided:
  - Streamlined routing, which simplified the management of different endpoints and improved the organization of the codebase.
  - Efficient middleware management, enabling the use of reusable components to handle requests and responses systematically.

- These features enhanced the overall system performance and maintainability, allowing for easier debugging and updates to the application as requirements evolve.

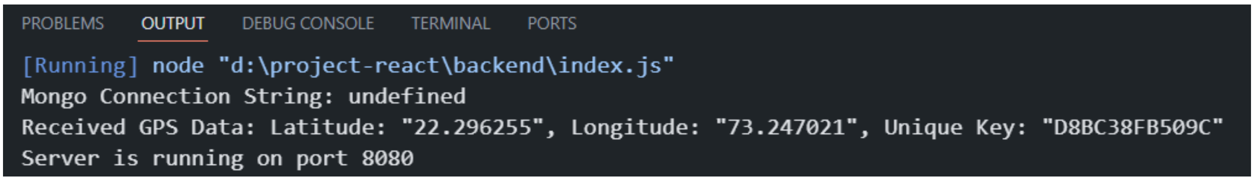


Figure 6.11: Backend Server

6.3 IoT Device Configuration:

- The IoT devices, including sensors and microcontrollers embedded in delivery boxes, were configured to perform various tasks such as:
  - **Operating the ESP32, ESP32-CAM, and Servo motor:** These components are essential for enabling automated actions based on real-time data, such as opening and closing delivery boxes upon user request, ensuring a seamless delivery process.
  - **Communicating with the web application:** This constant communication ensures the system remains updated with the latest device status, allowing for quick responses to any changes in conditions or user actions, thereby enhancing user experience and operational efficiency.

- These devices communicated using protocols like:
  - **HTTP**: This protocol allows real-time data exchange between the IoT hardware and the system. By utilizing HTTP, the devices can send immediate updates regarding their status, facilitating prompt actions based on user interactions and ensuring a responsive and dynamic system.

- The configuration ensured that the devices operated accurately, sending timely updates on:
  - **Package status:** This feature provides users with real-time tracking capabilities, allowing them to monitor their deliveries closely and stay informed about their package's journey from dispatch to arrival.

  - **Delivery information:** This aspect improves the efficiency of logistics by providing timely data on expected delivery times, delays, or issues, thereby enhancing customer satisfaction and trust in the service.

- This real-time communication enhanced the efficiency of the overall system by reducing delays in information processing. By ensuring that all components work in harmony, the system is able to function smoothly, minimizing the risk of errors and improving the overall user experience.

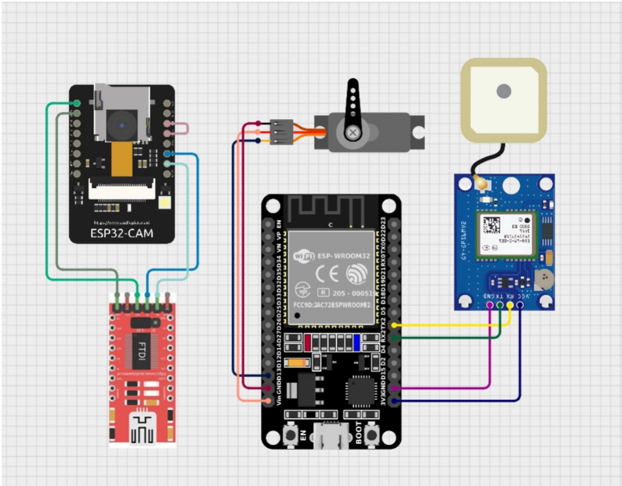


Figure 6.12: Circuit Diagram

Figure 6.13: IOT Box Opened

Figure 6.14: IOT Box Closed

6.4 Database Setup:

- **MongoDB** was implemented as the database system to store all essential data related to:
  - **Reservations:** This includes details about booked packages, user preferences, and timestamps for reservation changes, enabling easy tracking and management.
  - **User accounts:** User profiles contain vital information such as personal details, order history, and preferences, facilitating a personalized experience and enhancing user engagement.
  - **Package tracking:** Real-time updates on package status, location, and delivery timelines are stored to provide users with accurate information and improve transparency.
  - **System configurations:** Settings and configurations related to the application and its functionalities are stored to allow for flexible adjustments and optimizations based on user feedback and usage patterns.
- The database schema was carefully designed to ensure:
  - **Efficient data storage:** The use of document-based storage allows for the organization of complex data structures, minimizing redundancy and optimizing storage space.
  - **Fast retrieval:** Indexing strategies were implemented to enhance query performance, ensuring that data is accessible quickly and efficiently, even as the dataset grows.
  - **Simplified management:** The schema is designed to be intuitive and flexible, making it easier for developers to modify and expand the database structure as new features are introduced.
- Relationships between various data entities were optimized for quick access, enabling:
  - **Real-time data updates:** The database allows for immediate updates and synchronization, which is crucial for maintaining the accuracy of package tracking and user interactions.
  - **Seamless performance without compromising system efficiency:** The efficient design minimizes latency in data access, ensuring that user requests are processed swiftly, which is essential for a smooth user experience.
- The MongoDB setup played a critical role in maintaining:
  - **Data integrity:** Ensuring that all data entered into the system adheres to defined rules, which prevents corruption and maintains the quality of information.
  - **Reliability:**The database's inherent replication features safeguard against data loss, ensuring that the system can recover quickly from any failures.
  - **Scalability as the system grew:** MongoDB's ability to handle large volumes of data and traffic makes it suitable for growing applications, allowing for easy scaling horizontally to accommodate increasing user demands.

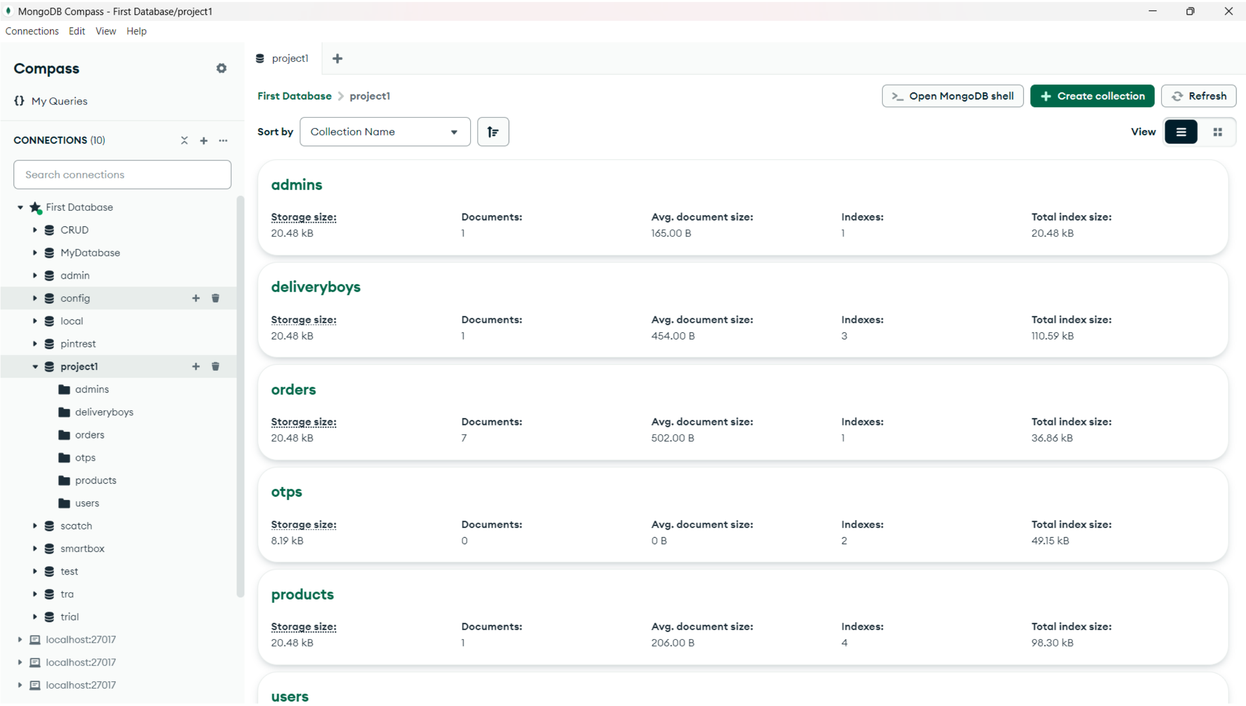


Figure 6.15: Database

6.5 Integration of External Services:

- These services were crucial for:
  - Sending automated notifications to users regarding order status and updates,
  - Enabling real-time package tracking to enhance user experience,
  - Processing secure payments to ensure financial safety.
- APIs provided by these external services were utilized to ensure:
  - Seamless integration within the existing architecture,
  - Enhanced communication capabilities with external platforms for better service delivery.
- This integration added essential features to the system while maintaining:
  - Smooth interoperability between different services, ensuring a cohesive user experience.

The implementation of these external services significantly improved the system's overall efficiency and user satisfaction. By leveraging robust APIs, we ensured that users receive timely updates and secure transactions, which are vital in today's digital landscape. Furthermore, this integration laid the foundation for future enhancements, allowing for scalability and adaptability to evolving user needs and technological advancements. As we continue to refine and expand our services, the reliable support of these external services will remain a cornerstone of our operational strategy.

Chapter 7

Testing and Quality Assurance

The testing phase ensured that the entire system functioned according to the defined specifications and was free of critical defects. Various stages of testing, including unit testing, integration testing, and system-wide validation, were performed to ensure seamless performance across all components.

7.1 Unit Testing:

- **Frontend Testing:**
  - We tested all UI components, such as buttons, forms, and navigation menus, to ensure they functioned properly and displayed correctly across different scenarios.
  - Form validation was thoroughly assessed, covering package tracking IDs, reservation details, and user authentication inputs, to guarantee error-free input handling.
  - We utilized testing tools like Jest to validate the rendering and functionality of frontend components in a React-based environment, ensuring that user interactions were seamless.

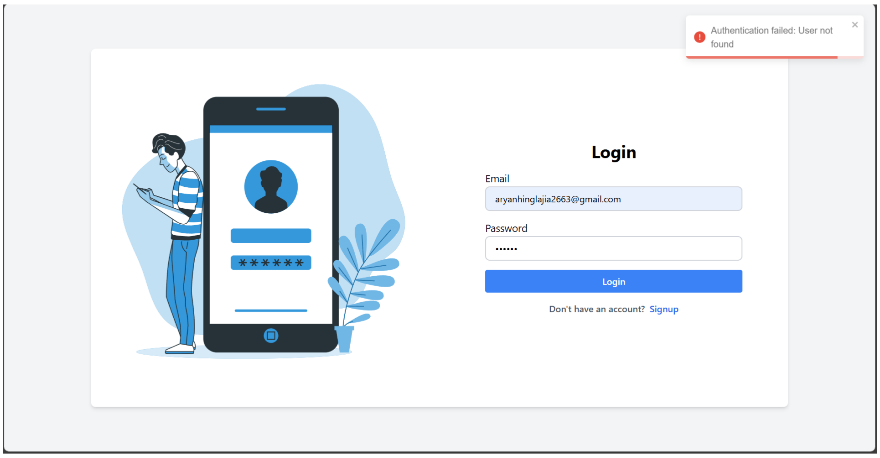


Figure 7.1: Wrong Email and Password

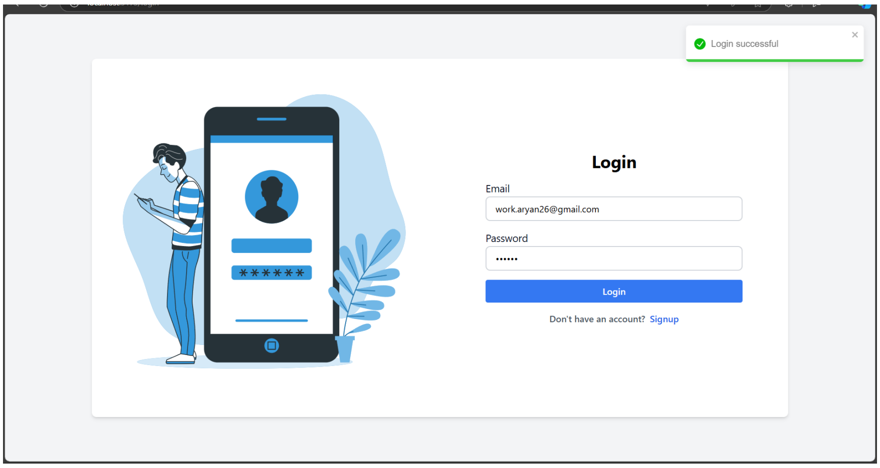
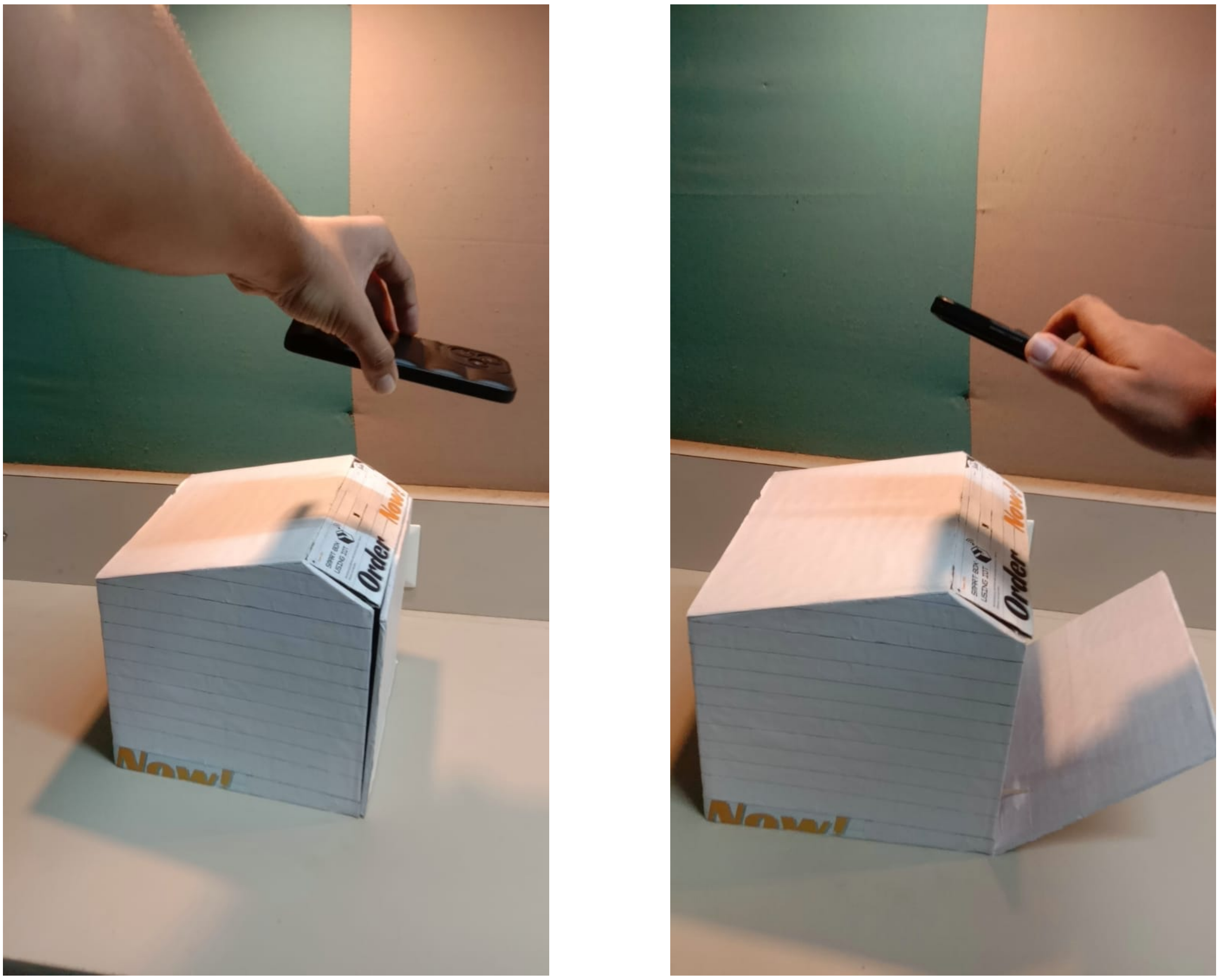


Figure 7.2: Correct Email and Password

- **Backend Testing:**
  - Each API endpoint was rigorously tested for response accuracy, including checking for proper JSON formatting and the correct handling of various HTTP status codes.
  - We performed tests on all database CRUD operations (Create, Read, Update, Delete) using Postman to confirm smooth interaction with the database and to validate data integrity.
  - Mocha was employed to run unit tests for backend functions, verifying that they operated correctly under various conditions and edge cases.
- **IoT Device Testing:**
  - Individual sensors, including item detection and QR code scanners embedded in the IoT device, were tested to ensure they accurately captured and transmitted data to the system.
  - The connectivity of the ESP32 module to the Wi-Fi and server was verified by monitoring real-time data transmission through the Arduino IDE's Serial Monitor.
  - We validated sensor data against expected outputs, ensuring the accuracy of real-time updates provided by the IoT devices.



(a) Wrong QR Code (b) Correct QR Code

Figure 7.3: Comparison of QR Code Scans

7.2 Integration Testing:

- **Frontend and Backend Integration:**
  - We tested the communication between the frontend and backend by ensuring that form submissions, such as reservation creation and package tracking, triggered appropriate API calls and received correct responses.
  - We confirmed that error messages were accurately displayed on the frontend for scenarios like invalid credentials or missing tracking details, enhancing user experience.
- **Backend and Database Integration:**
  - Data insertion and retrieval between the backend and the database were successfully tested, ensuring that all reservation details, package tracking information, and user accounts were accurately stored and updated.
  - We verified data consistency between the database and frontend to ensure that any changes in data were properly reflected in real-time on the user interface, providing users with accurate information.

- **IoT Device Integration:**
  - We thoroughly tested the communication between IoT devices and the web application through MQTT/HTTP protocols, ensuring that data from IoT sensors, such as real-time item detection, was successfully transmitted to the backend and displayed in the application.

7.3 System Testing:

- We conducted an end-to-end workflow test of the system, starting from user interaction with the reservation system to real-time package tracking using the IoT devices. The entire process operated smoothly, with no significant disruptions observed.
- Performance testing was carried out to ensure that the system could handle multiple simultaneous users without experiencing significant degradation in performance, confirming its robustness under load.

7.4 User Acceptance Testing (UAT):

- Testers were engaged to interact with the system, simulating real-world usage scenarios. Their feedback was largely positive, confirming that the system met the expected usability standards.
- We addressed any minor usability concerns raised during UAT, implementing improvements to enhance the overall user experience based on tester feedback.

Chapter 8

Conclusion and Future Work

8.1 Conclusion:

- In conclusion, the ”Smart Goods Transfer using IoT Box” project represents a significant advancement in the field of package delivery, offering a comprehensive solution that combines cutting-edge technology with practical functionality. By leveraging IoT devices, secure authentication methods, and real-time tracking capabilities, this project addresses key challenges in the delivery process while providing tangible benefits for e-commerce businesses, logistics companies, and consumers alike.
- Through enhanced security measures, streamlined operations, and improved customer experiences, the project not only mitigates risks associated with package theft and tampering but also fosters trust, loyalty, and satisfaction among users. Furthermore, the project’s potential to expand market reach, drive operational efficiencies, and generate valuable data insights underscores its value proposition for businesses seeking to stay competitive in the evolving digital landscape.
- As the project moves forward, ongoing refinement, optimization, and adaptation to changing market dynamics will be crucial to its long-term success. By continually improving user interfaces, enhancing security protocols, and embracing emerging technologies, the project can remain at the forefront of innovation and deliver lasting value to stakeholders across the e-commerce and logistics ecosystem.

8.2 Future Work

- To ensure the integrity of temperature-sensitive medical goods, such as vaccines and pharmaceuticals, precise temperature control during shipping is essential. Smart boxes equipped with advanced environmental sensors are designed to maintain the ideal temperature throughout the entire journey.
- When transporting high-value items or sensitive documents, heightened security measures and accountability become crucial. Smart boxes incorporate tamper-detection mechanisms, providing an additional layer of protection against unauthorized access.

- Moreover, integrating a display screen on the smart box allows users to monitor the camera's viewpoint and view real-time footage of what the camera is capturing. This feature enhances transparency, offering users reassurance that their valuable items are being vigilantly overseen during transit.
- In addition, we can implement a sensor to detect items inside the smart box, allowing it to automatically close after a specified period. This automated closure mechanism not only bolsters security but also ensures the internal environment remains optimal for sensitive goods, safeguarding them throughout the shipping process.

Chapter 9

References

1. Gubbi, Jayavardhana, et al. ”Internet of Things (IoT): A vision, architectural elements, and future directions.” Future generation computer systems 29.7 (2013): 1645-1660.

2. Mishra, Anupam, et al. ”Internet of Things (IoT) in logistics: A comprehensive literature review.” Computers & Industrial Engineering 158 (2021): 107448.

3. Rezaei, Ali, and Mohammad Mohebbi. ”A systematic review of Internet of Things (IoT) applications in supply chain management (SCM).” Journal of Enterprise Information Management (2021).

4. Sun, Zhonghua, et al. ”Internet of Things for smart logistics: Technologies and applications.” IEEE Access 5 (2017): 26799-26810.

5. Yao, Yawen, et al. ”Research on Internet of Things architecture for logistics monitoring.” 2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER) (2015): 1567-1571.

6. Zanella, Andrea, et al. ”Internet of Things for smart cities.” IEEE Internet of Things journal 1.1 (2014): 22-32.

7. Li, Xue, et al. ”Smart logistics for industry 4.0: A review.” International Journal of Production Research 57.8 (2019): 2438-2460.

8. Jiang, Bin, et al. ”Smart logistics in the context of Industry 4.0: A review.” Computers in Industry 101 (2018): 103-114.

9. Jia, Fang, et al. ”Smart logistics for perishable products based on Internet of Things.” Computers Industrial Engineering 141 (2020): 106268.

10. Machado, Marco, et al. ”A review on the application of Internet of Things in logistics.” Computers in Industry 79 (2016): 75-81.

11. Shancang Li, Li Da Xu, Shanshan Zhao, ”The internet of things: a survey”, Published online: 26 April 2014, Springer Science+Business Media New York 2014.

12. G. Takacs, J. Vachálek, B. Rohal'-Ilkiv, "Online Structural Health Monitoring and Parameter Estimation for Vibrating Active Cantilever Beams Using Low-Priced Microcontrollers", Shock and Vibration, Volume 2015, DOI: 10.1155/2015/506430.

13. GitHub "Arduino core for ESP32 WiFi chip" [online], 2019, available at: https://github.com/espressif/arduino-esp32.

14. GitHub "Espressif IoT Development Framework" [online], 2019, available at: https://github.com/espressif/esp-idf/.

15. Lee, J. "The Internet of Things (IoT): Applications, investments, and challenges for enterprises" (2015).

16. Gubbi, J. et al. "Internet of Things (IoT): A vision, architectural elements, and future directions." Future Generation Computer Systems (2013).

17. Hêriş Golpîra. "A review of logistics Internet-of-Things: Current trends and scope for future research" (2021).

18. S. Li et al. "5G Internet of Things: a survey." J. Ind. Inf. Integr. (2018).

19. S. Nižetić et al. "Internet of Things (IoT): opportunities, issues and challenges towards a smart and sustainable future." J. Clean. Prod. (2020).

20. P. Wei et al. "Research on security of information sharing in internet of things based on key algorithm." Fut. Gener. Comput. Syst. (2018).

21. Brous, P., et al. "The Internet of Things in logistics: A literature review." *Logistics* 4.3 (2020): 15.

22. Hassall, M., et al. "Smart Logistics Systems: A Review of the State-of-the-Art and Future Research Directions." *Logistics* 4.4 (2020): 27.

23. Pahlavan, K., et al. "Smart logistics: An overview of the Internet of Things (IoT) and its impact on supply chain management." *Journal of Supply Chain Management* 56.1 (2020): 3-18.

24. Chen, M., et al. "The role of IoT in supply chain management: A review of current applications and future research directions." *International Journal of Production Economics* 224 (2020): 107548.

25. Adhikari, S., et al. "The role of Internet of Things in smart logistics: A survey." *International Journal of Production Research* 58.8 (2020): 2336-2360.

26. Kumar, R., et al. "IoT-based smart logistics systems: A survey." *IEEE Transactions on Industrial Informatics* 17.1 (2021): 654-664.

27. Nai, Y., et al. "Internet of Things and smart logistics: A review of applications and future directions." *Computers in Industry* 122 (2020): 103264.

28. Wang, H., et al. "Applications of Internet of Things in Logistics and Supply Chain Management." *Journal of Network and Computer Applications* 165 (2020): 102735.

29. Wang, Y., et al. ”A survey of the Internet of Things in logistics.” *Journal of Manufacturing Systems* 58 (2021): 235-247.

30. Zhao, Y., et al. ”IoT in Logistics: Opportunities and Challenges.” *Journal of Industrial Information Integration* 21 (2020): 100193.

31. Intuz. ”IoT-based Smart Box development for package tracking and tracing.” (2021). URL: iot box tracking ON intuz.com.

32. Machado, Marco, et al. ”A Review on the Application of Internet of Things in Logistics.” Computers in Industry 79 (2016): 75-81. URL: the internet of things a review on the application of internet of things in logistics ON ScienceDirect sciencedirect.com.

33. Jia, Fang, et al. ”Smart Logistics for Perishable Products Based on Internet of Things.” Computers Industrial Engineering 141 (2020): 106268. URL: smart logistics for perishable products based on internet of things ON ScienceDirect sciencedirect.com.

34. Yao, Yawen, et al. ”Research on Internet of Things architecture for logistics monitoring.” 2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER) (2015): 1567-1571. URL: research on internet of things architecture for logistics monitoring ON Institute of Electrical and Electronics Engineers ieeexplore.ieee.org.

35. Troindia. ”SMART PILL BOX USING IOT.” (2018). URL: smart medicine box using iot ON troindia.in.

36. IJARIIT. ”An IoT based smart medicine box.” International Journal of Advanced Research and Innovative Technologies (2019). URL: a n iot based smart medicine box ON ijariit.com.

37. ResearchGate. ”Using IoT technologies to develop a low-cost smart medicine box.” (2021). URL: using iot technologies to develop a low cost smart medicine box ON ResearchGate researchgate.net.

38. Prompt Dairy Tech. ”SmartBox - IOT Based Real Time Milk Monitoring System with Alert Generation and Data Analytics.” (2021). URL: iot milk monitoring system ON Prompt Dairy Tech promptdairytech.com.

39. Li, Xue, et al. ”The internet of things: a survey.” Springer (2014). URL: internet of things a survey ON Springer link.springer.com.

40. GitHub. ”Arduino core for ESP32 WiFi chip.” (2021). URL: github esp32 ON github.com.

41. GitHub. ”Espressif IoT Development Framework.” (2021). URL: espressif iot development framework ON GitHub github.com.