**TITLE**

**INTELLIGENT IOT SMART TROLLEY WITH AUTOMATED INVENTORY AND CHECKOUT**

**1. ABSTRACT**

The Intelligent IoT Smart Trolley is designed to revolutionize the shopping experience by integrating RFID-based automated checkout, real-time inventory tracking, and robotic movement. This system aims to minimize billing queues, enhance customer convenience, and provide seamless inventory management for shop owners. The trolley follows customers using RFID-based identification and employs ultrasonic sensors to navigate obstacles. The billing system updates in real-time using Adafruit IO cloud integration, ensuring efficient inventory tracking. This report details the project’s methodology, system design, implementation, coding, and future enhancements

**2. INTRODUCTION**

In today’s fast-paced world, shopping has become an essential part of daily life. However, one of the biggest challenges faced by customers in supermarkets and shopping malls is long queues at billing counters. Traditional checkout processes require customers to wait for their turn while cashiers manually scan and process each item, causing delays and frustration. Additionally, managing a shopping trolley manually can be inconvenient, especially for physically challenged individuals and the elderly.

With advancements in Internet of Things (IoT) and automation, smart solutions are being developed to enhance shopping experiences. Our Intelligent IoT Smart Trolley aims to automate billing, assist in movement, and integrate real-time inventory management. By incorporating RFID/barcode scanning, an OLED display, robotic movement, and cloud-based inventory tracking, this trolley eliminates the need for manual billing and ensures a seamless shopping process.

**2.1 Problem Statement**

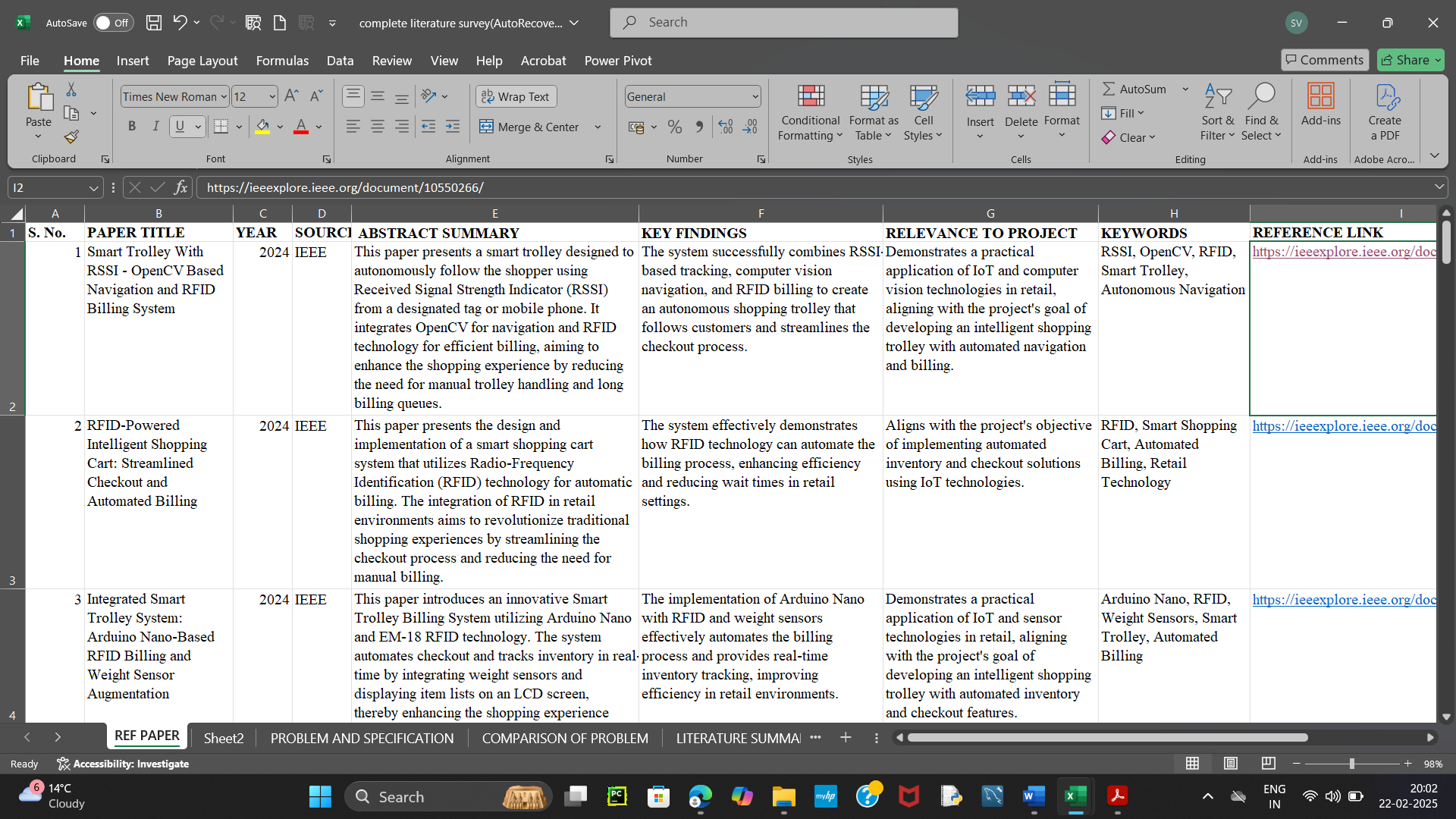
* Customers face long queues at billing counters.
* Manually pushing trolleys is inconvenient, especially for elderly or disabled individuals.
* Shop owners struggle with real-time inventory tracking and stock management.

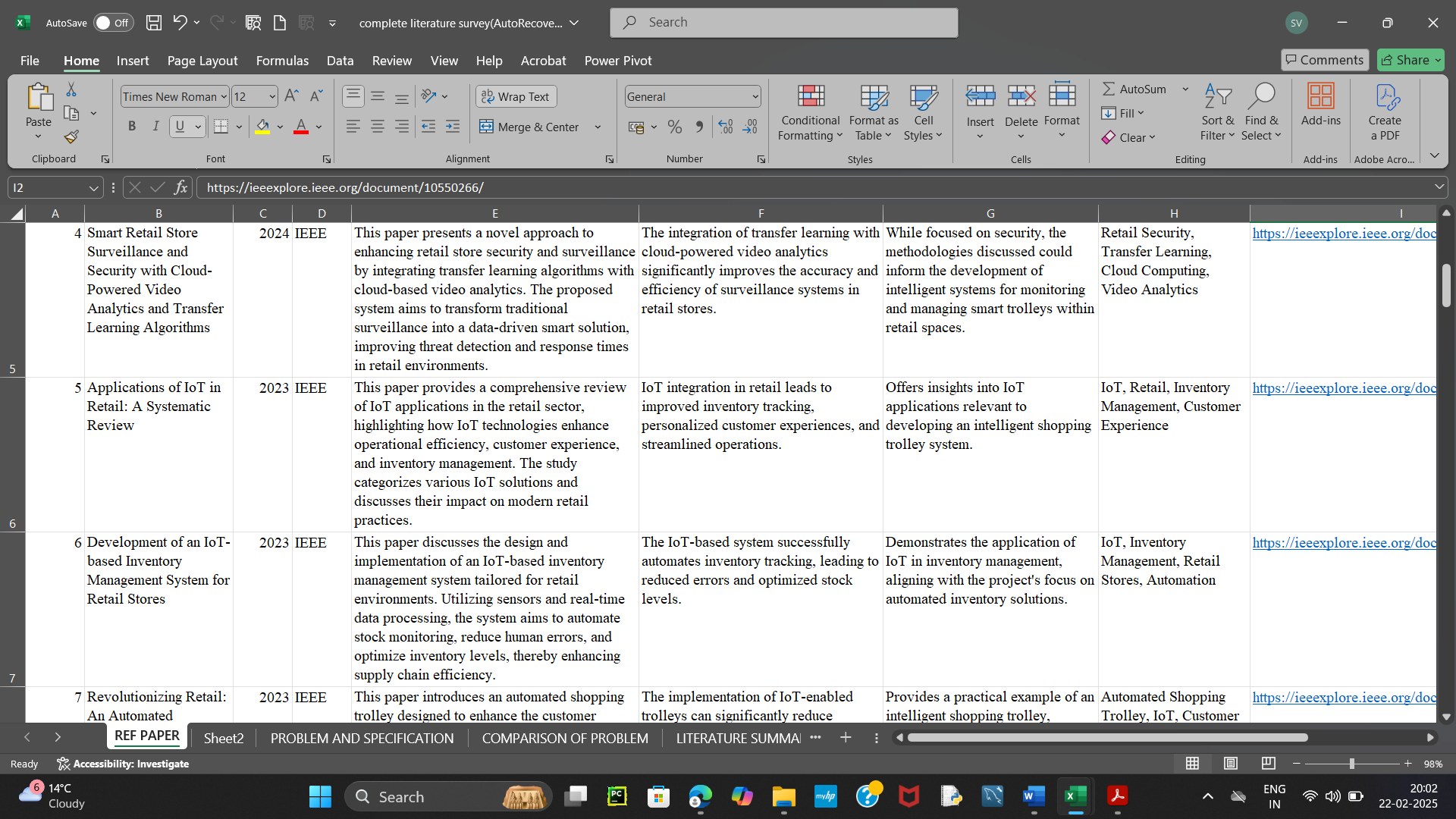
**2.2 Proposed Solution**

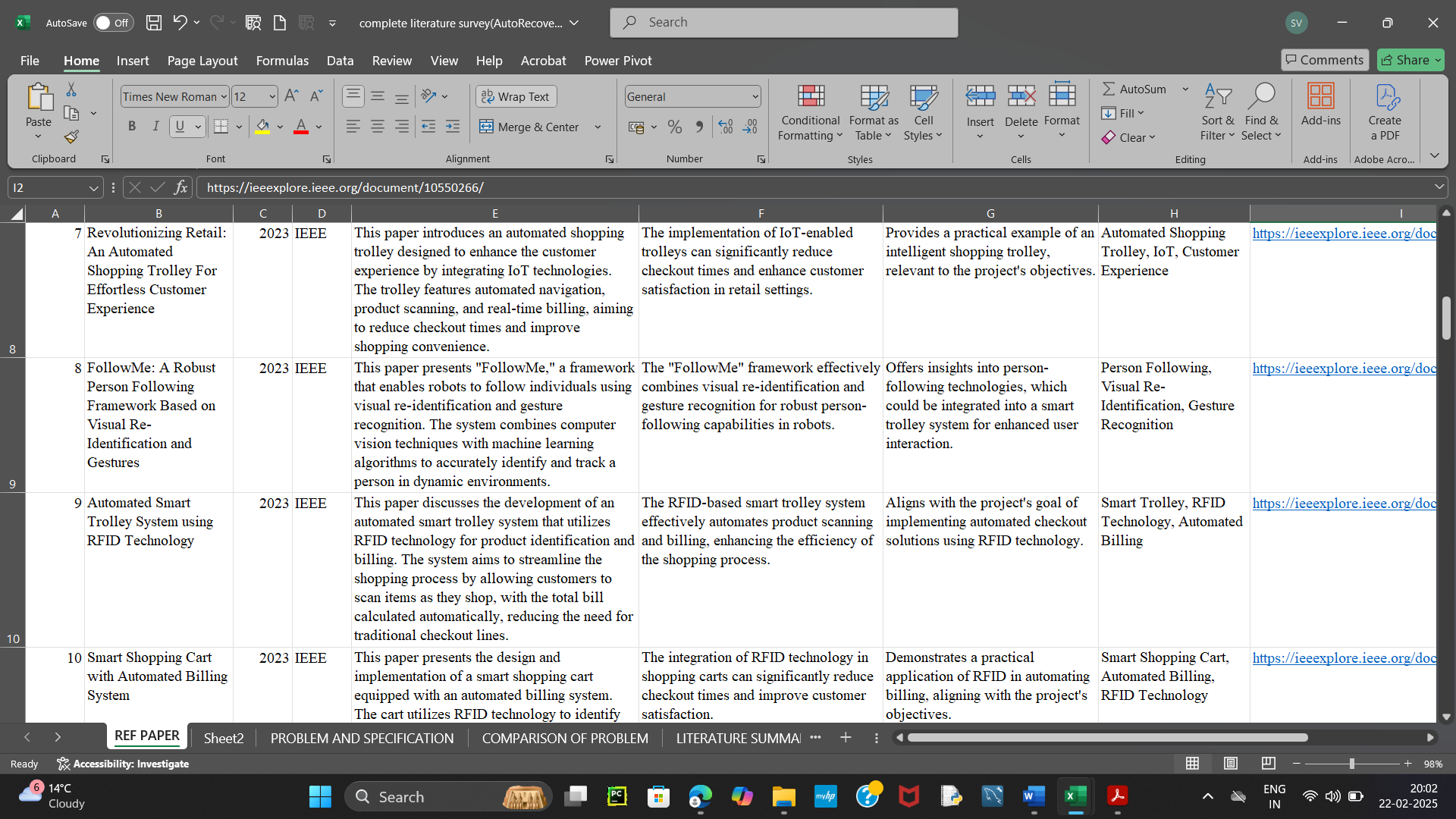
* A smart trolley that follows customers using RFID-based tracking.
* Automated billing system with an RFID scanner and real-time price calculation.
* Cloud-based inventory management for shop owners via Adafruit IO.
* Obstacle detection and smart navigation using ultrasonic sensors.

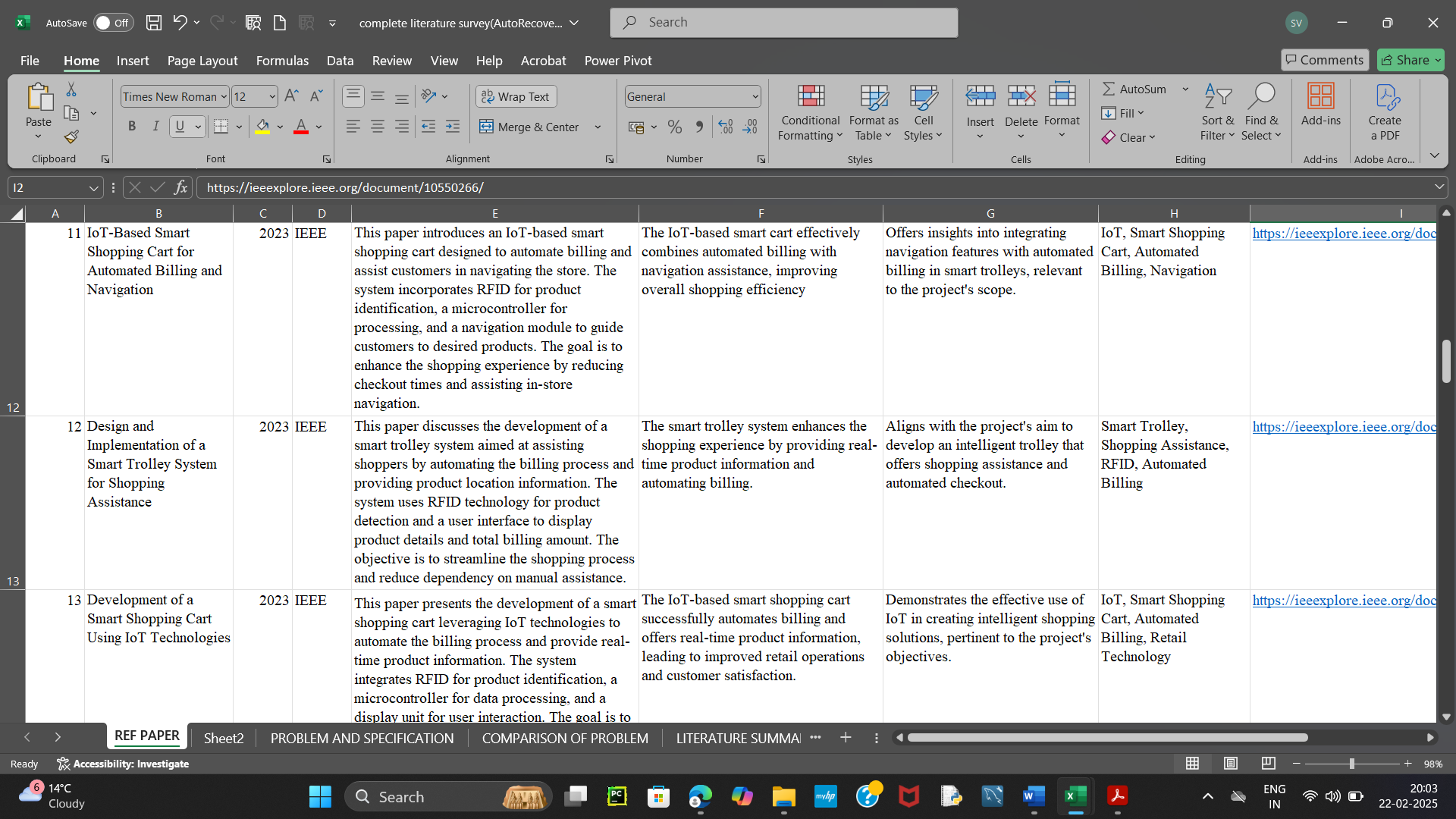
**3. LITERATURE REVIEW**

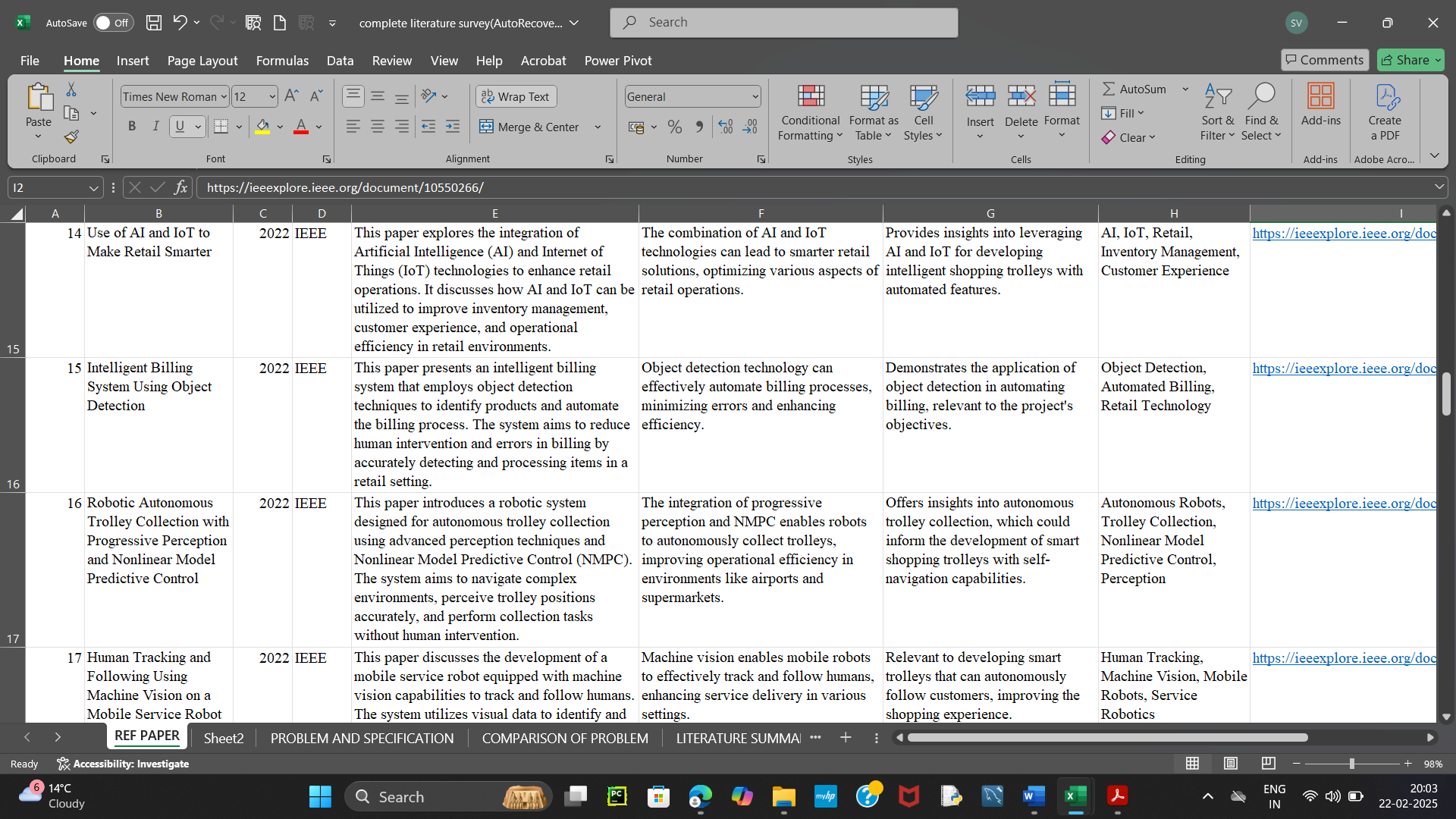
This project is inspired by research on smart shopping systems, RFID-based automated billing, and IoT-driven inventory management. Prior studies highlight challenges like signal interference, scanning accuracy, and real-time data processing.

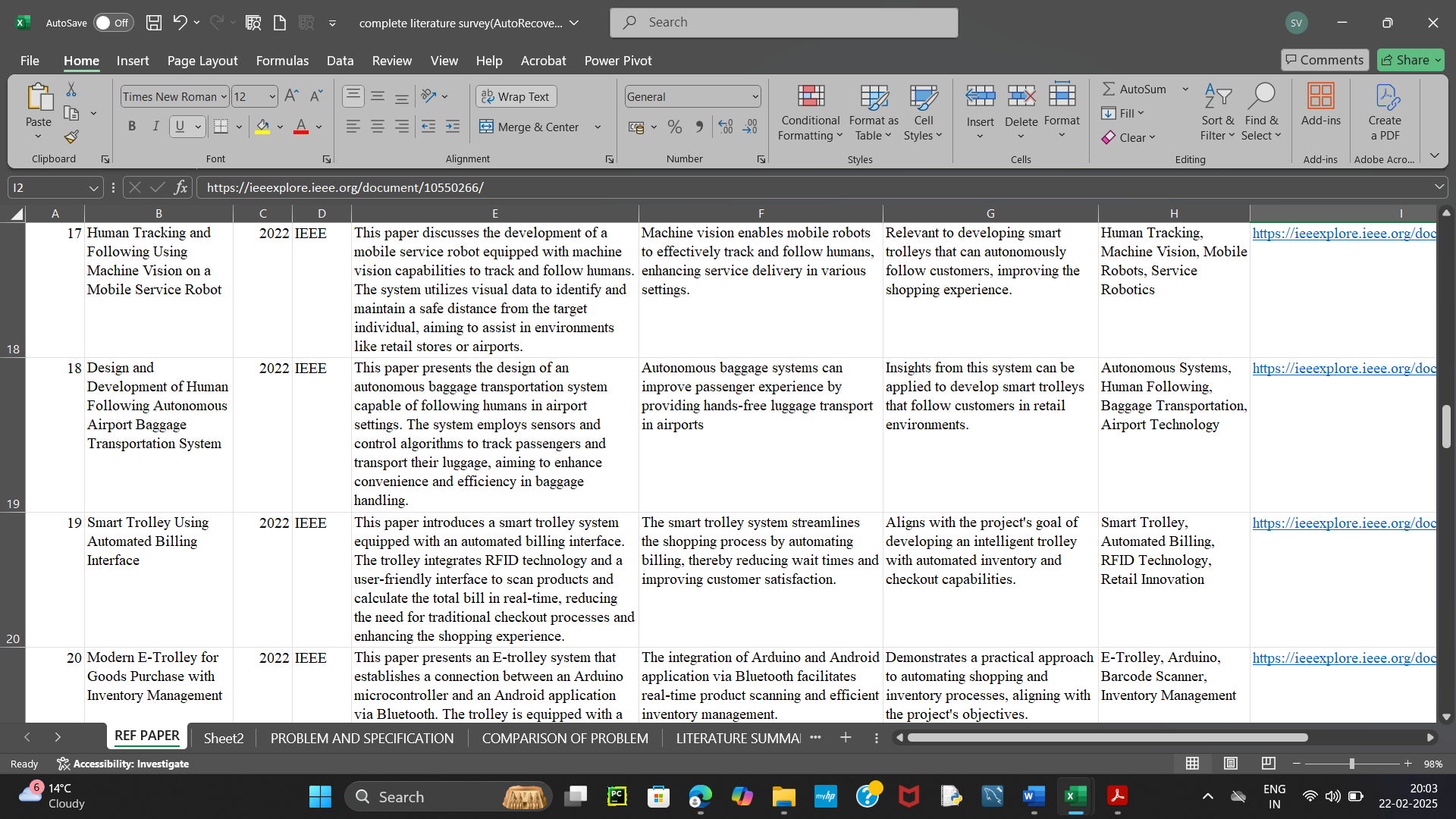


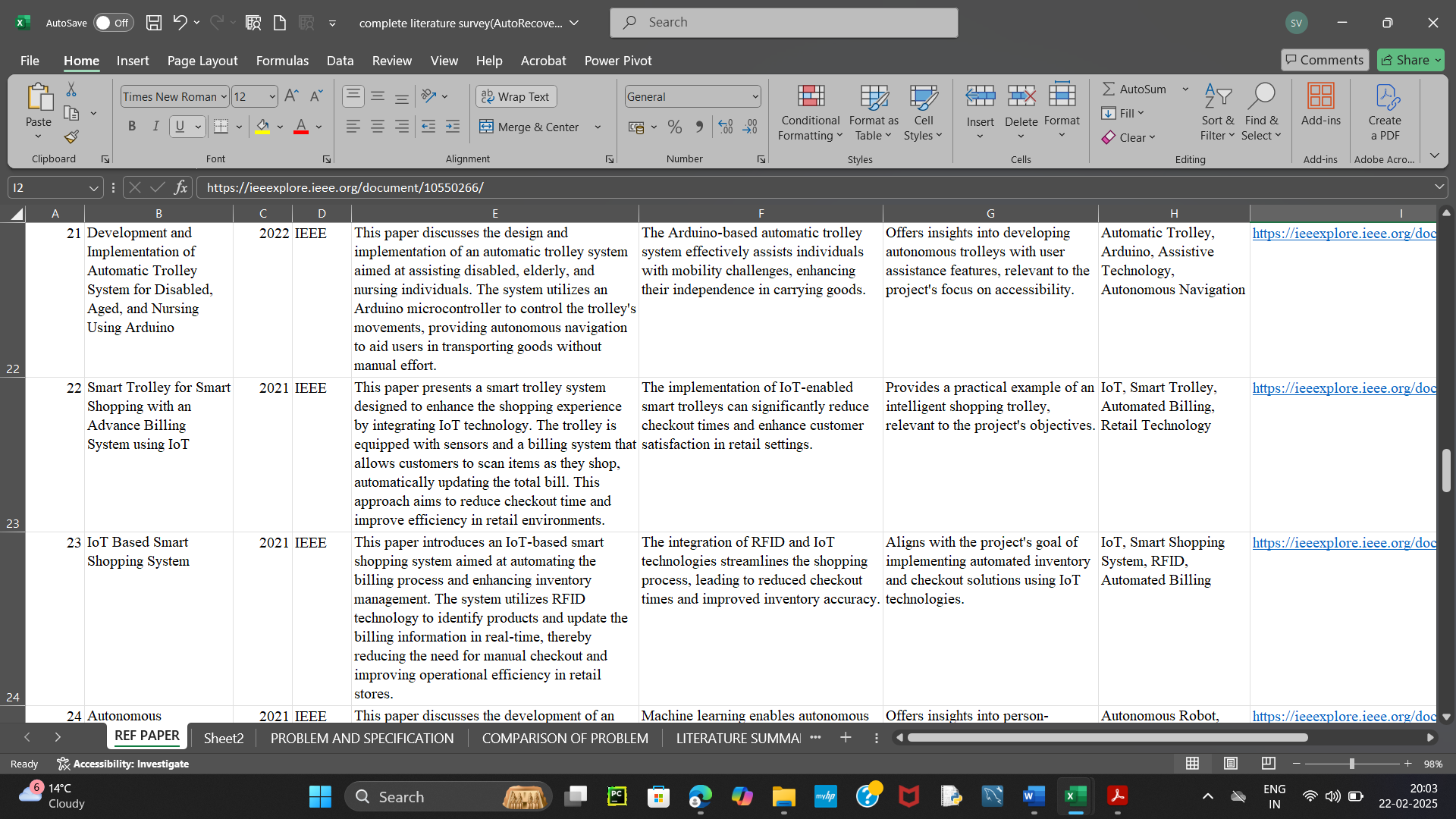
****

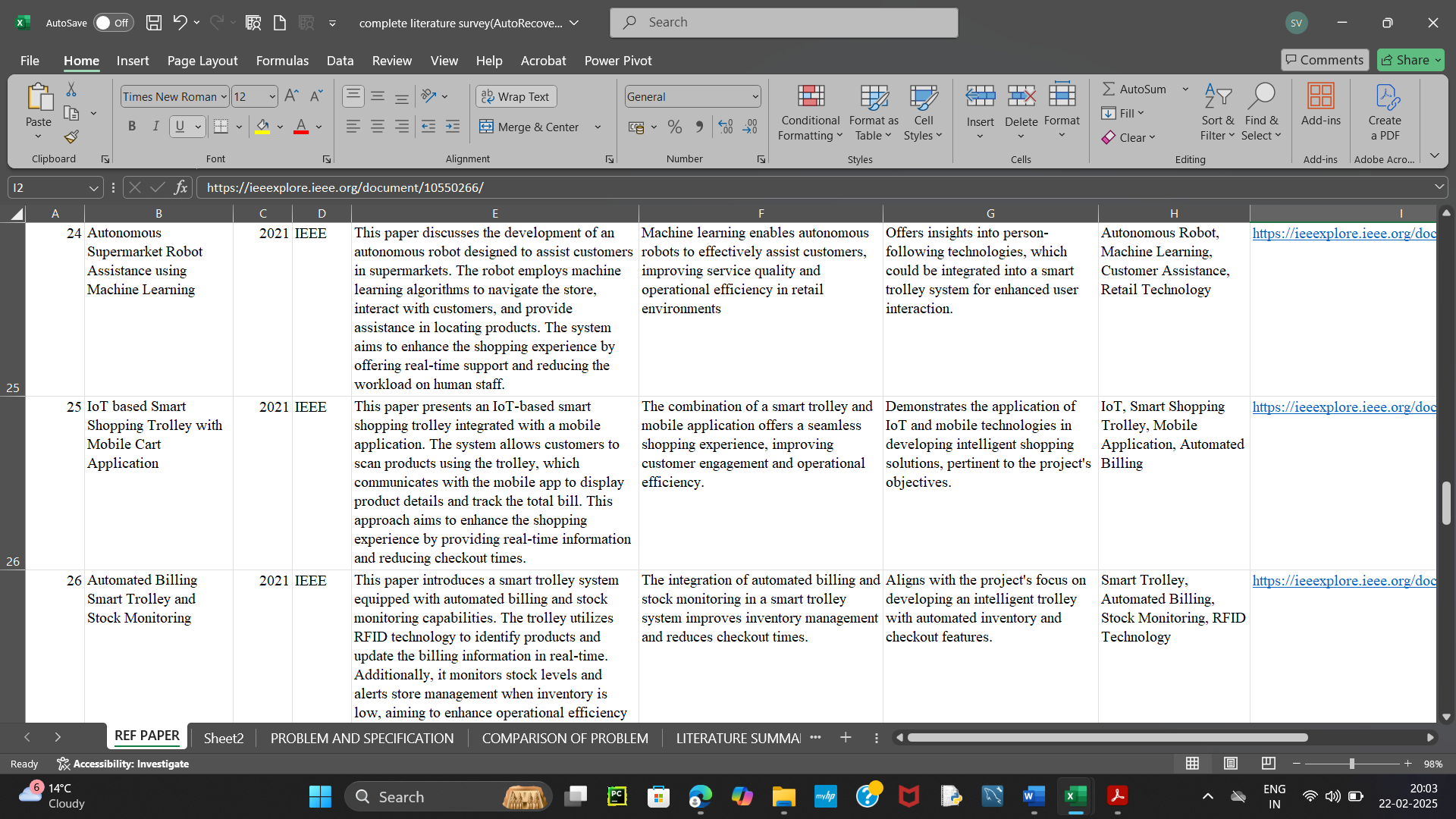
****

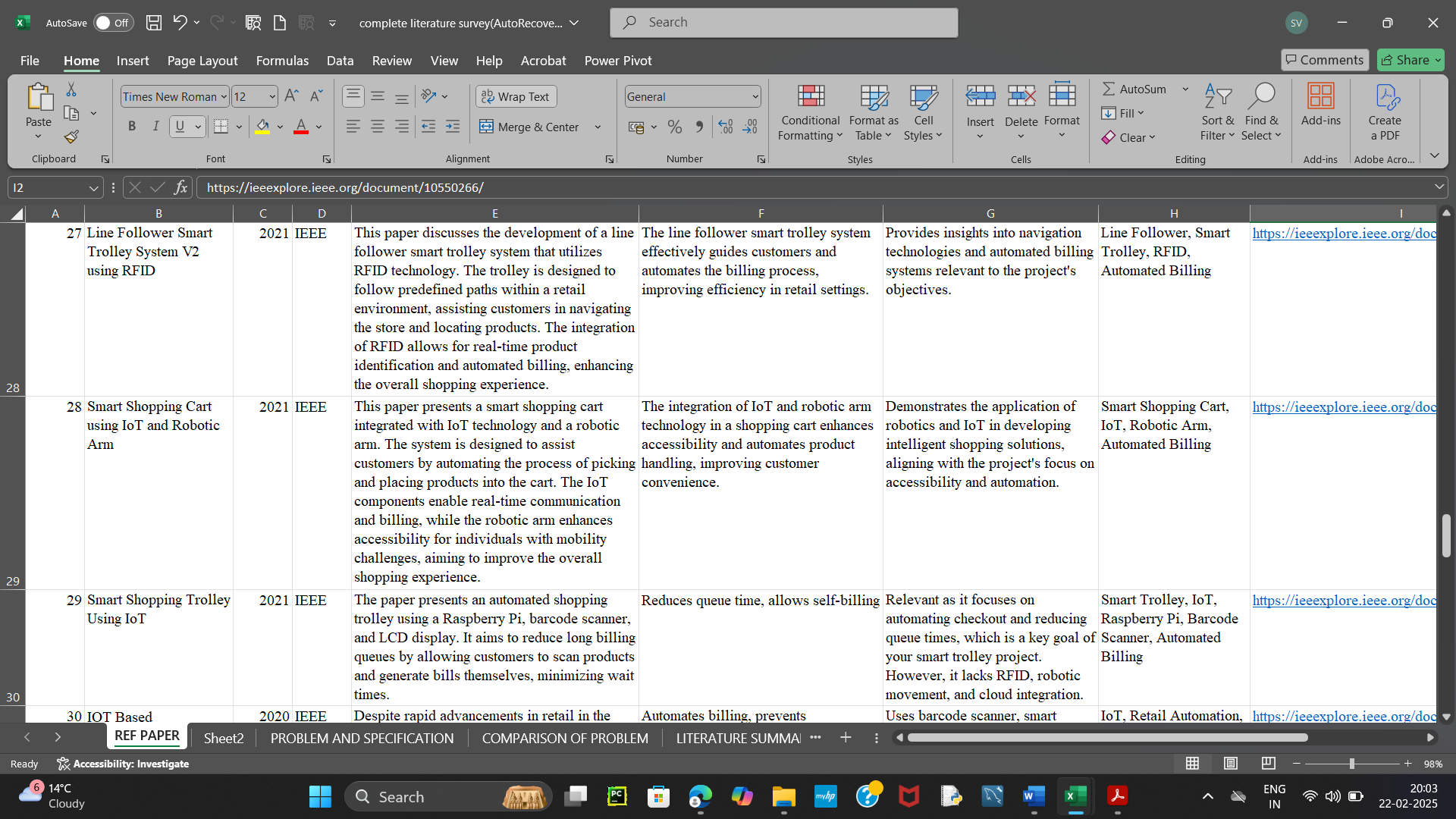
****

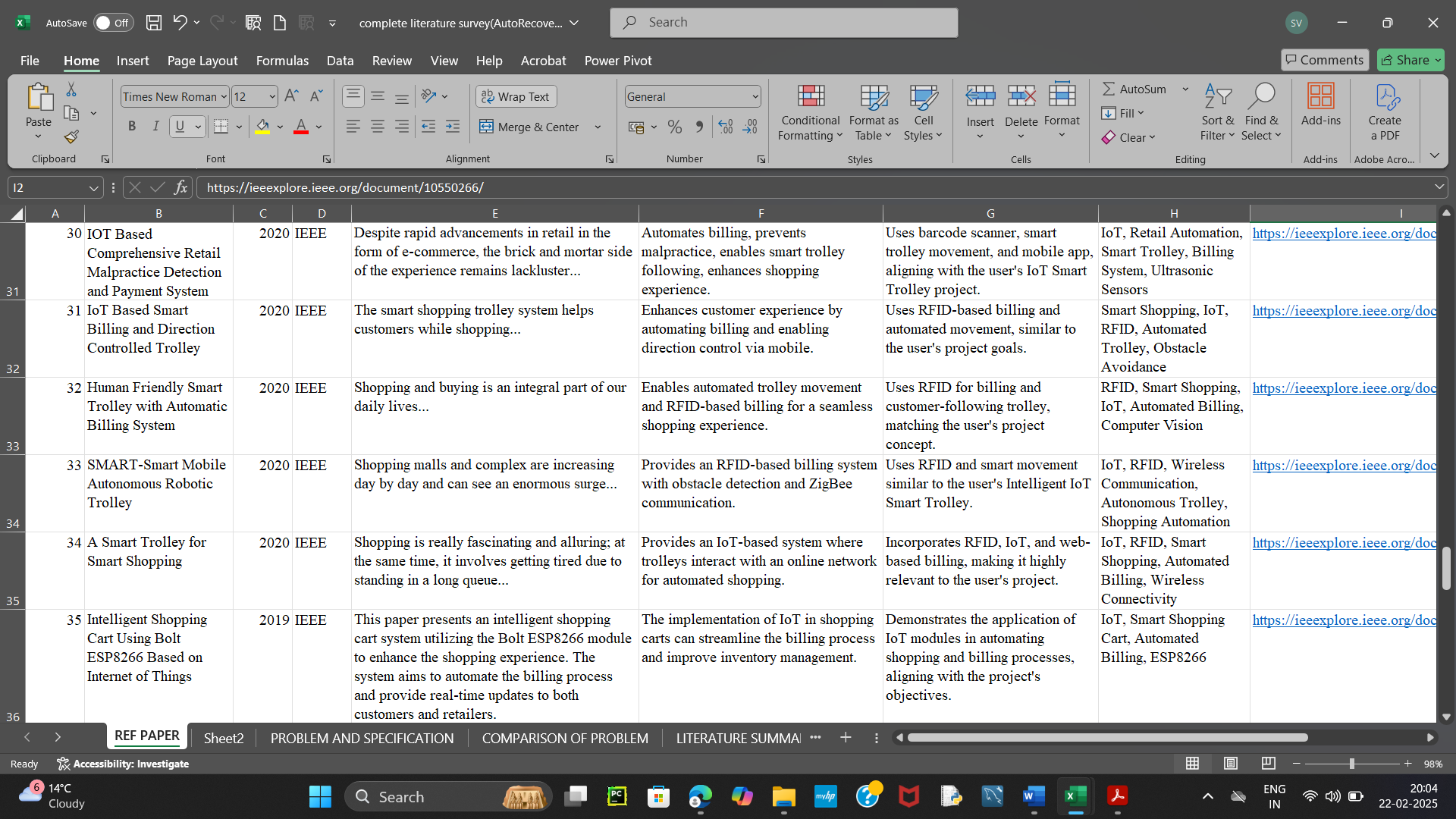
****

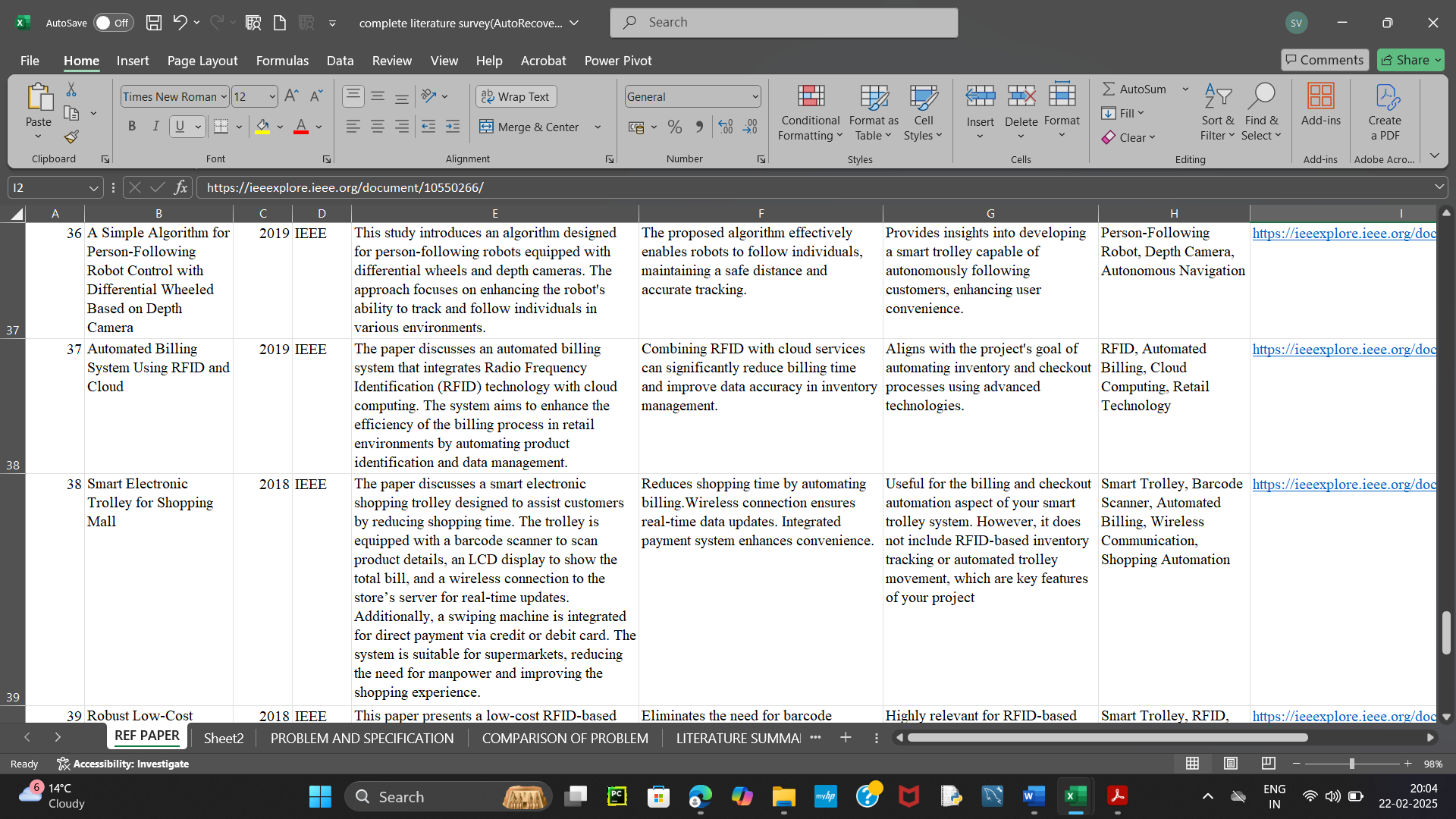
****

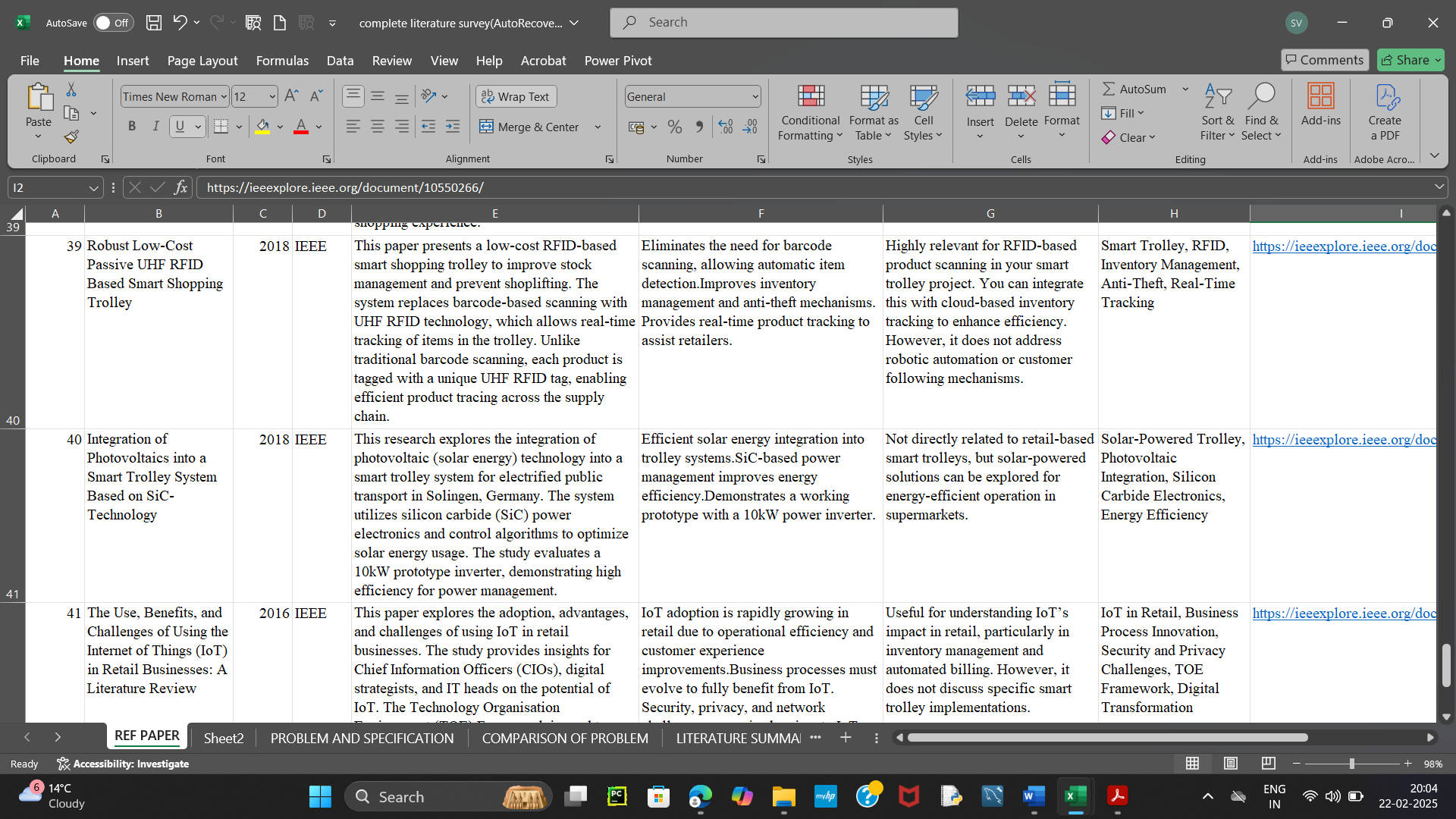
****

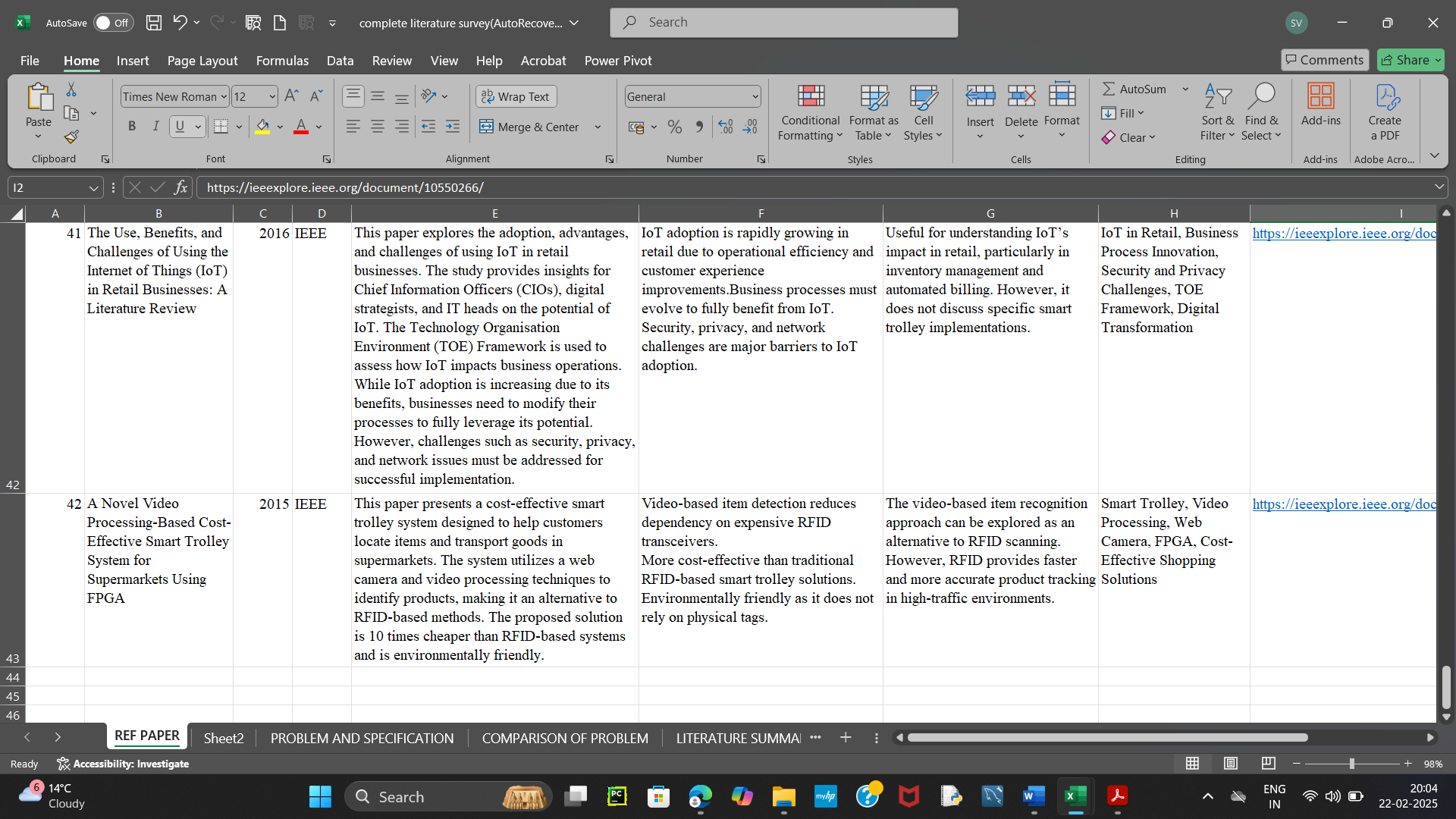
****

****

****

****

****

****

**3.1 COMPARISON OF PROBLEM**

|  |  |  |
| --- | --- | --- |
| **PROBLEM** | **SOLUTION** | **SPECIFICATIONS** |
| **1. Manual Billing Causes Delays in Supermarkets** | **Automated Billing System:** Scanner reads product details and calculates total amount for quick checkout. **Payment Integration:** NFC or other methods for seamless payment. | 1. Scanner to read product details. 2. Payment processing system for automatic billing. 3. System written in Arduino C. |
| **2. Customers Face Difficulty Locating Items in the Store** | **Product Information Display:** Trolley displays product details on the screen. **Voice Assistance:** Provides directions or product info. | 1. LCD/OLED display to show product details. 2. Voice assistant to guide customers. |
| **3. Inventory Management is Time-Consuming for Shop Owners** | **Cloud Integration:** Real-time inventory management via Adafruit IO. **Automated Stock Update:** Product details automatically uploaded to the cloud when scanned. | 1. Cloud storage via Adafruit IO. 2. Real-time data sync. 3. Automated inventory updates. |
| **4. Lack of Accessibility for Physically Challenged Individuals** | **Automated Movement**: Robotic system with ultrasonic sensors to make the trolley move autonomously. **Voice Assistance & Visual Aids**: Helps individuals find products. | 1. Motors and sensors for autonomous navigation. 2. Voice assistance and visual aids (LCD) |
| **5. In Crowded Environments, Identifying and Managing a Customer-Specific Trolley is Challenging** | **RFID Tracking**: Each customer gets an RFID tag, and the trolley follows the customer using the RFID system. **Customer Identification**: Tracks specific trolley-customer pair. | 1. RFID tag (keychain) for customers. 2. RFID reader on the trolley for tracking. 3. Trolley follows the customer in crowded areas. |

**3.2 Research Gap Identified**

* 1. Barcode-based systems require customers to manually scan products, leading to inefficiencies.
  2. RFID-based billing trolleys exist but lack an automated tracking feature to follow users.
  3. Smart trolleys with ZigBee or Bluetooth-based tracking require mobile apps, adding complexity for users.
  4. Obstacle detection mechanisms in autonomous trolleys exist but are not optimized for crowded shopping areas.

**3.3 OUR PROJECT FILLS THE GAP:**

* RFID-based automatic user tracking eliminates the need for mobile apps.
* Real-time inventory management via cloud integration helps shop owners keep track of stock.
* Ultrasonic sensors for obstacle detection allow smooth navigation in crowded spaces.
* Touchscreen LCD enhance accessibility for all users, including those with disabilities.

**3.4 REFERENCES I SELECTED FOR LITERATURE REVIEW**

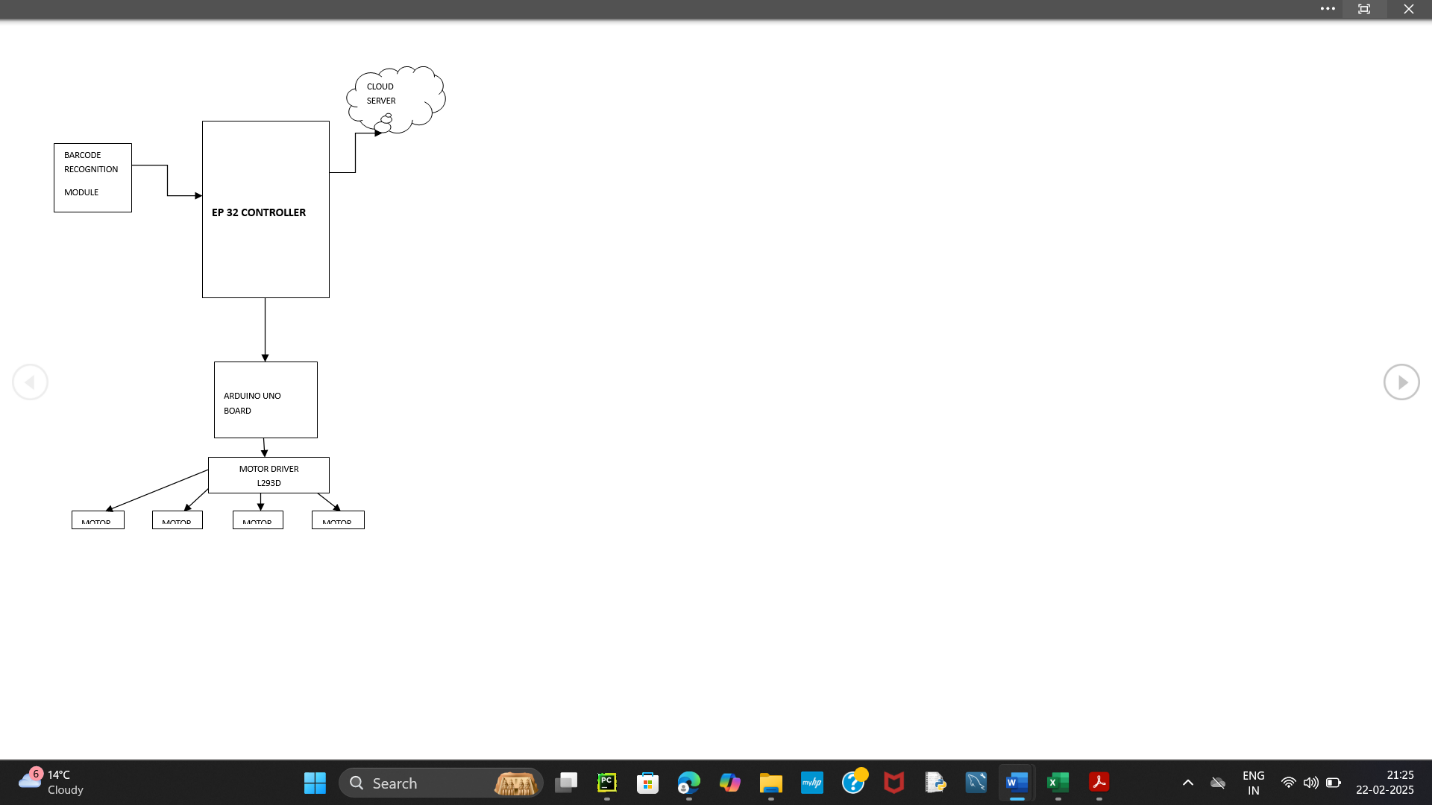
* **SMART: Smart Mobile Autonomous Robotic Trolley**
* Uses RFID for smart movement.
* Implements wireless communication for real-time tracking.
* Similar autonomous trolley concept as your project.
* **Automated Billing System using RFID and Cloud**
* Focuses on RFID-based automated billing.
* Uses cloud storage for real-time updates.
* Eliminates manual billing queues for a seamless shopping experience.
* **Robust Low-Cost Passive UHF RFID Based Smart Shopping Trolley**
* Proposes a cost-effective UHF RFID system for smart trolleys.
* Minimizes errors in scanning using passive RFID tags.
* Focuses on enhancing trolley efficiency with low power consumption.

**4. SYSTEM DESIGN**

**4.1 Block Diagram**

The system consists of the following components:

* Arduino Nano – Microcontroller for system control.
* RFID Reader & Tags – Identifies the user and scans product details.
* Ultrasonic Sensors – Detects obstacles and guides movement.
* Motor Driver & Motors – Enables trolley automation.
* LCD/OLED Display – Shows scanned product details and total bill.
* Wi-Fi Module (ESP8266) – Connects to Adafruit IO for cloud inventory updates.

****

**DESCRIPTION**

1. The block diagram represents a Location Tracking Smart Robot System integrated with a barcode recognition module.

2. An EP 32 Controller is the core processing unit, managing input, output, and communication with other components.

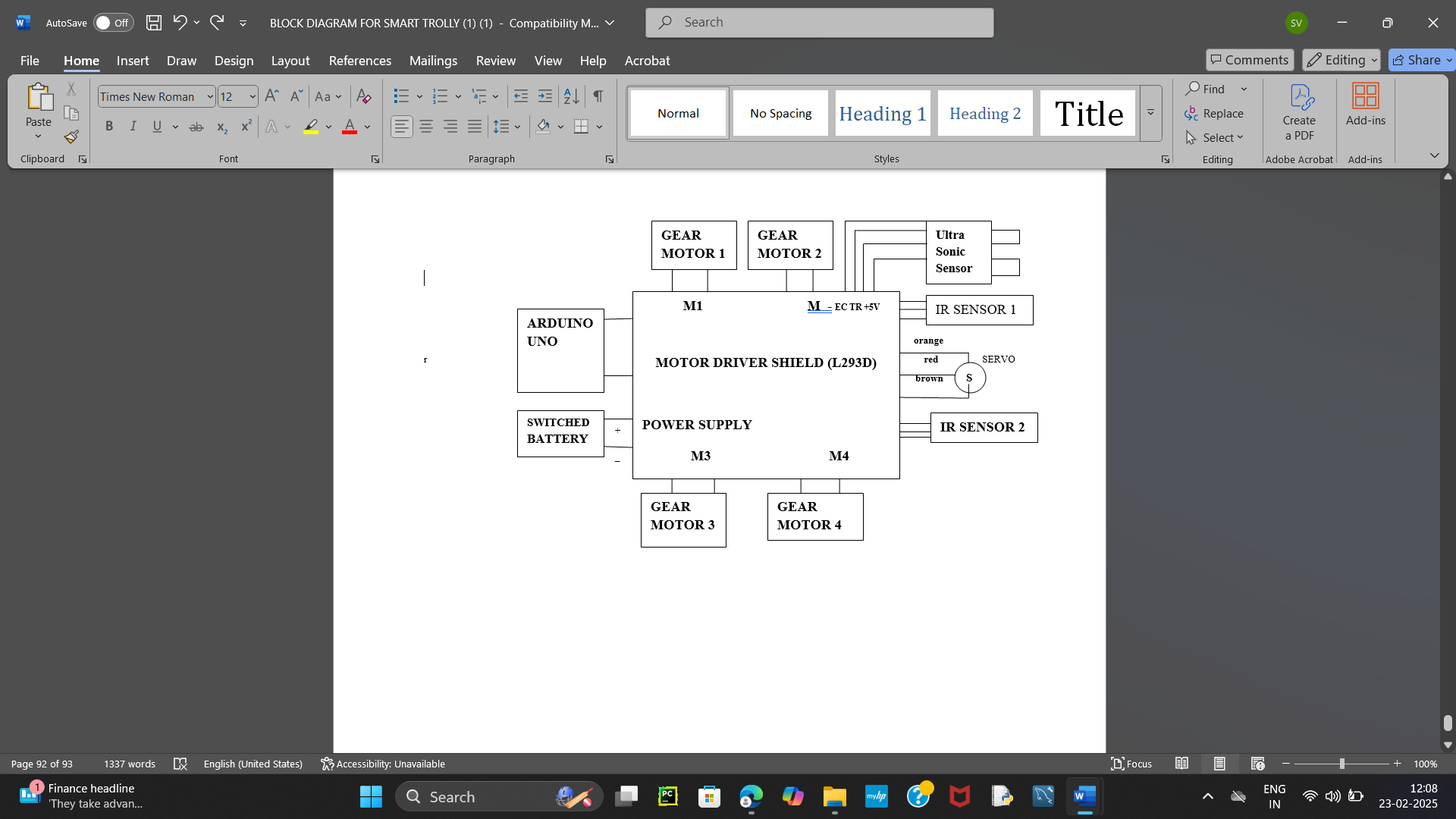
3. A Barcode Recognition Module scans and identifies products, allowing the system to track items in real time.

4. The system communicates with a Cloud Server for data storage, processing, and remote access, ensuring efficient data handling and retrieval.

5. The Arduino Uno Board acts as an intermediary, controlling the motors and executing movement instructions from the EP 32 Controller.

6. The Motor Driver L293D manages four motors (Motor 1, Motor 2, Motor 3, Motor 4) to enable the robot's movement, including forward, backward, and turning.

7. The overall system integrates hardware and software components to deliver a smart, autonomous robotic platform for product tracking and navigation.

****

**DESCRIPTION**

1. The block diagram illustrates a Smart Trolley System designed for automated movement and navigation.

2. A Switched Battery Box powers all components, ensuring uninterrupted operation.

3. Arduino Uno acts as the central controller, processing data from sensors and managing motor operations.

4. The Motor Driver Shield controls four Gear Motors (M1, M2, M3, M4)(L293D) to enable precise movements like forward, backward, and turning.

5. Ultrasonic Sensors detect obstacles in the trolley's path, ensuring smooth and collision-free navigation.

6. IR Sensors are used for line-following or path detection, guiding the trolley along predefined routes.

7. A Servo Motor performs specific tasks such as positional adjustments or mechanical actions.

8. The system integrates all components to enable autonomous operation and enhance usability in real-time applications like shopping malls.

**CLOUD SERVER**

**EP 32 CONTROLLER**

**RFID**

**READER**

**MODULE**

**TROLLEY**

**DISPLAY**

|  |
| --- |
| **TROLLEY ROBOT** |

**DESCRIPTION**

The block diagram represents the working architecture of the Intelligent IoT Smart Trolley with Automated Inventory and Checkout System. The system integrates various hardware components for seamless functionality.

**1. RFID-Based Item Scanning System**

* **RFID Reader Module**: Detects products tagged with RFID tags when placed inside the trolley.
* **ESP32 Controller**: Processes the scanned product data and sends it to the cloud server.
* **Trolley Display**: Shows product details, total price, and other relevant information.

**2. Cloud Integration**

* **Cloud Server**: Stores product details, inventory data, and transaction information.
* Ensures real-time updates for shop owners.

**3. Smart Trolley Movement Mechanism**

* **Arduino Uno**: Controls the movement of the trolley.
* **Motor Driver Shield (L293D)**: Interfaces with multiple gear motors for mobility.
* **Gear Motors (M1, M2, M3, M4)**: Enable the trolley to move in different directions.

**4. Sensor-Based Navigation & Obstacle Detection**

* **Ultrasonic Sensor**: Detects obstacles to prevent collisions.
* **IR Sensors (IR Sensor 1 & IR Sensor 2)**: Help in path detection and obstacle avoidance.
* **Servo Motor**: Assists in precise control and movement of the trolley.

**5. Power Management**

* **Switched Battery**: Supplies power to the Arduino and motor driver system.

**Functionality Summary**

* The trolley automatically scans products using RFID.
* It uploads product details to a cloud server for inventory management.
* Real-time display of scanned items and total price on the trolley's display.
* Automated movement using gear motors, IR sensors, and ultrasonic sensors to navigate the store.
* The Arduino Uno and Motor Driver Shield control the movement and navigation of the trolley.

This system enhances the shopping experience by reducing checkout time, automating inventory updates, and making shopping more convenient, especially for physically challenged individuals.

**4.2 Working Mechanism**

**1. Customer Authentication:** The trolley identifies the customer using their **RFID keychain**.  
**2. Smart Movement:** The trolley follows the customer using **ultrasonic sensors and motor control**.  
**3. Product Scanning:** RFID-based scanner detects **products placed in the trolley**.  
**4. Billing & Inventory Management:**

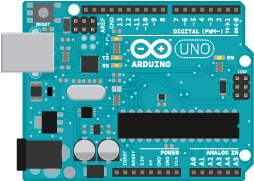
* The total bill is displayed on the **LCD screen**.
* Product details are uploaded to **Adafruit IO cloud**.

**5Checkout:** The bill is finalized, and payment is processed via **digital transactions or billing counters**.

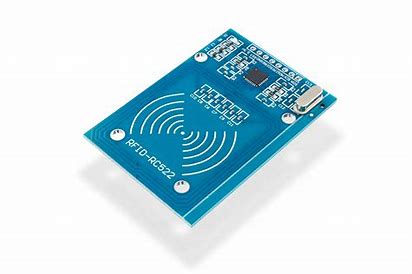
**5. IMPLEMENTATION**

**5.1 Hardware Components**

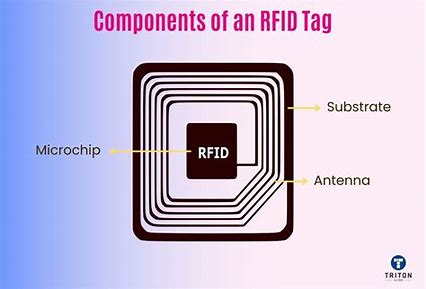
* **Microcontroller:** Arduino



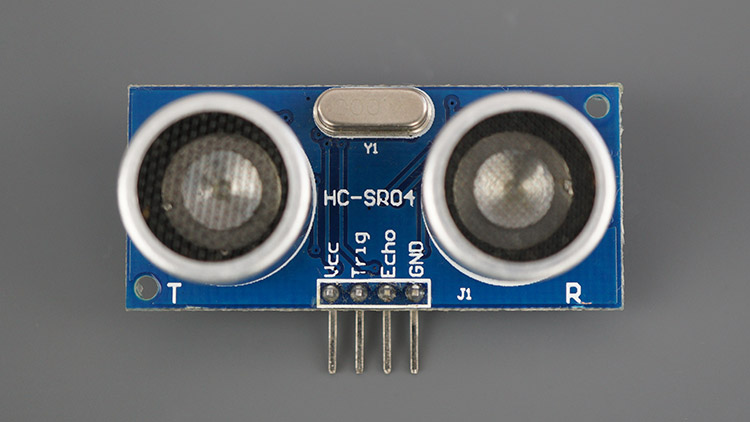
* **RFID Scanner:** EM-18 RFID Reader



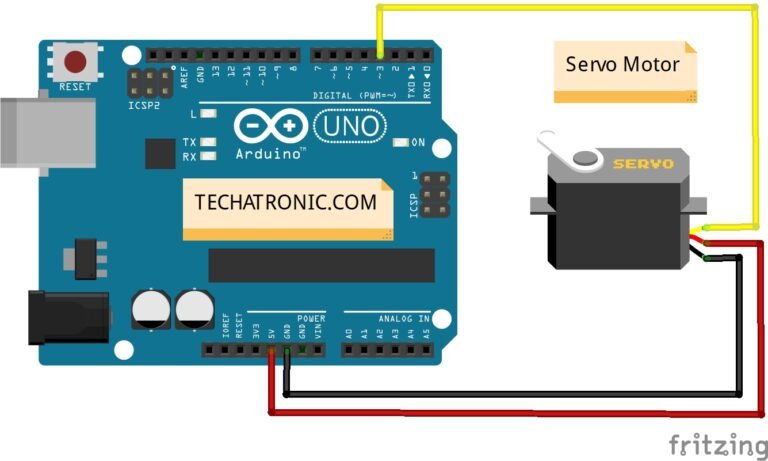
* **User Identification:** RFID Keychain Tags



* **Obstacle Detection:** Ultrasonic Sensors



* **Automated Movement:** Servo Motor & L293D Motor

* **Billing Display:** OLED/LCD screen
* **Cloud Integration:** Adafruit IO (IoT Dashboard)

**5.2 Software Development**

* **Arduino C** – Controls **RFID scanning, trolley movement, and sensor inputs**.
* **Adafruit IO Integration** – Cloud database for **product tracking and inventory updates**.
* **React-Based Web Dashboard** – Allows **shop owners to monitor stock levels**.

**6. CODING IMPLEMENTATION**

**Code for Trolley Movement**

#include <AFMotor.h> // Library for L293D Motor Shield

#define TRIG 9

#define ECHO 10

#define PIR 8

#define IR\_LEFT 4

#define IR\_RIGHT 5

AF\_DCMotor motor1(1); // Left Motor

AF\_DCMotor motor2(2); // Right Motor

void setup() {

pinMode(TRIG, OUTPUT);

pinMode(ECHO, INPUT);

pinMode(PIR, INPUT);

pinMode(IR\_LEFT, INPUT);

pinMode(IR\_RIGHT, INPUT);

Serial.begin(9600);

}

void loop() {

long duration;

int distance;

digitalWrite(TRIG, LOW);

delayMicroseconds(2);

digitalWrite(TRIG, HIGH);

delayMicroseconds(10);

digitalWrite(TRIG, LOW);

duration = pulseIn(ECHO, HIGH);

distance = duration \* 0.034 / 2;

int pirState = digitalRead(PIR);

int leftIR = digitalRead(IR\_LEFT);

int rightIR = digitalRead(IR\_RIGHT);

if (distance > 10 && pirState == HIGH) {

motor1.setSpeed(150);

motor2.setSpeed(150);

motor1.run(FORWARD);

motor2.run(FORWARD);

}

else if (leftIR == LOW) {

motor1.run(BACKWARD);

motor2.run(FORWARD);

}

else if (rightIR == LOW) {

motor1.run(FORWARD);

motor2.run(BACKWARD);

}

else {

motor1.run(RELEASE);

motor2.run(RELEASE);

}

}

**Functionality:**

* Moves forward **if no obstacles & detects human motion**.
* Uses **IR sensors** for **line tracking**.
* Stops if an **obstacle is detected**.

**Code for RFID & Display**

#include <Wire.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

#include <SPI.h>

#include <MFRC522.h>

#define SCREEN\_WIDTH 128

#define SCREEN\_HEIGHT 64

#define OLED\_RESET -1

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, OLED\_RESET);

#define SS\_PIN 21

#define RST\_PIN 22

MFRC522 mfrc522(SS\_PIN, RST\_PIN);

void setup() {

Serial.begin(115200);

SPI.begin();

mfrc522.PCD\_Init();

if(!display.begin(SSD1306\_SWITCHCAPVCC, 0x3C)) {

Serial.println(F("SSD1306 allocation failed"));

for(;;);

}

display.clearDisplay();

display.setTextSize(1);

display.setTextColor(WHITE);

display.setCursor(0, 10);

display.println("Scan Product");

display.display();

}

void loop() {

if (mfrc522.PICC\_IsNewCardPresent() && mfrc522.PICC\_ReadCardSerial()) {

String productID = "";

for (byte i = 0; i < mfrc522.uid.size; i++) {

productID += String(mfrc522.uid.uidByte[i], HEX);

}

display.clearDisplay();

display.setCursor(0, 10);

display.println("Product ID:");

display.println(productID);

display.display();

Serial.println("Scanned Product: " + productID);

delay(2000);

}

}

**Functionality:**

* Reads **RFID tag / barcode**.
* Displays **product details on OLED screen**.

**Code for Cloud Update**

#include <WiFi.h>

#include <FirebaseESP32.h>

#define WIFI\_SSID "Your\_WiFi\_Name"

#define WIFI\_PASSWORD "Your\_WiFi\_Password"

#define FIREBASE\_HOST "your\_project.firebaseio.com"

#define FIREBASE\_AUTH "your\_database\_secret"

FirebaseData firebaseData;

void setup() {

Serial.begin(115200);

WiFi.begin(WIFI\_SSID, WIFI\_PASSWORD);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println("WiFi connected!");

Firebase.begin(FIREBASE\_HOST, FIREBASE\_AUTH);

Firebase.reconnectWiFi(true);

}

void loop() {

String scannedProduct = "Product123";

if (Firebase.pushString(firebaseData, "/cart/products", scannedProduct)) {

Serial.println("Product uploaded to cloud!");

} else {

Serial.println("Failed to update cloud");

}

delay(5000);

}

**Functionality:**

* Connects **ESP32 to WiFi**.
* Uploads **scanned product to Firebase database**.

**8. RESULTS & DISCUSSION**

**8.1 Expected Outcomes**

* Customers can scan items while shopping, avoiding long queues.
* The trolley can autonomously follow customers, reducing physical effort.
* Inventory is updated in real time, ensuring better stock management.

**8.2 Advantages of the System**

✅ Reduces checkout time and improves efficiency.  
✅ Enhances accessibility for physically challenged users.  
✅ Minimizes human errors in billing and inventory updates.  
✅ Smart & automated movement for better shopping experience.

**9. CONCLUSION**

This project presents an IoT-based Smart Trolley that automates the billing process, product scanning, and trolley navigation while providing real-time inventory updates. The integration of RFID/barcode scanning, robotic movement, and cloud connectivity makes shopping more efficient and accessible.

With future improvements like AI-based product recommendations, voice assistance, and integration with payment gateways, the Smart Trolley can revolutionize the retail shopping experience, making it faster, smarter, and more user-friendly.

**10. REFERENCES**

1️ SMART: Smart Mobile Autonomous Robotic Trolley   
2️ Automated Billing System Using RFID and Cloud   
3️ Robust Low-Cost Passive UHF RFID-Based Smart Shopping Trolley