

Project Abstract

The project proposes a fully **automated smart home system** powered by Internet of Things (IoT) and AI. At its core, it includes:

- A smart **refrigerator hardware module** that tracks the stock of items and automatically places orders from user-preferred stores (like Walmart) when items run low.
- A **service automation** system that monitors every home appliances automatically schedules maintenance if needed.
- A **home automation hub** that controls lights, fans, blinds, and windows — all dynamically managed via sensors and user-defined preferences.

The goal is to build an environment that thinks and acts on behalf of the user — reducing effort, optimizing resources, and ensuring convenience and efficiency.

Steps to Develop

1. System Design & Planning

- Identify supported home appliances and vendors (Walmart, Target, etc.).
- Design fridge inventory hardware logic (RFID, camera recognition, or weight sensors).
- Define automation rules and settings (e.g., fan ON when temp > 30°C).

2. Hardware Setup

- Use **Raspberry Pi** or **ESP32/ESP8266** as central microcontroller.
- Attach sensors:
 - **Fridge:** Load cells, RFID readers, image sensors (for computer vision).
 - **Room:** Temperature, humidity, light, motion sensors.
 - **Windows/Blinds:** Servo motors or stepper motors.
 - **AC:** Smart plugs with voltage/current sensors.

3. Software Development

- Develop firmware for IoT devices (Arduino/C++, MicroPython).
- Use **MQTT** or **HTTP API** for communication between devices and backend.

- Build mobile/web dashboard for users to set preferences and view status.

4. Inventory Management + Order Placement

- Train object detection model (YOLOv5, TensorFlow Lite) to recognize items.
- Use logic to keep a count and store in cloud DB (Firebase, MongoDB).
- When 5 or more items are low, trigger an **auto-order API** (Walmart, Amazon, etc.).

5. Service Automation for Appliances

- Monitor appliance health via smart plug data.
- When anomalies or scheduled service dates occur, call external APIs to book a service.

6. Full Home Automation

- Use **Home Assistant** or **OpenHAB** for integration.
- Voice and app control with **Google Assistant**, **Alexa**, or custom voice bot.
- Scene presets: e.g., “Sleep Mode” turns off lights, closes blinds, sets AC.

Key Uses & Benefits

- Minimizes grocery trips with **auto restocking**.
- Proactive appliance care reduces downtime and repair costs.
- Personalized automation increases **energy efficiency** and **comfort**.
- Full control and visibility through mobile/web UI.
- Integrates seamlessly with smart ecosystems and online stores.

Tools & Technologies

Microcontrollers	Raspberry Pi, ESP32, Arduino
Sensors	Load Cells, RFID, PIR, Temp, Light
Communication	MQTT, Bluetooth, Wi-Fi, Zigbee
Backend	Node.js, Firebase, MongoDB
Frontend	React.js, Flutter (for mobile app)
AI/ML	TensorFlow, OpenCV, YOLOv5
Automation Platform	Home Assistant, OpenHAB
Voice Integration	Google Assistant SDK, Alexa Skills Kit