**Smart Temple Automation System  
Using Arduino with Ultrasonic and PIR Sensors**

Project Report  
Prepared as part of an Independent Project

**Prepared By:**

**Prasad Goregaonkar**  
Software Developer & Electronics Enthusiast

✉️ Email: your\_email@example.com  
🌐 GitHub: github.com/yourgithub  
📍 Location: Pune, India

September 2025

**Smart Temple Automation System with Ultrasonic and PIR Sensors**

**Chapter 1: Introduction**

Automation has become a key part of modern living, ranging from industrial applications to home decorations. Festivals, events, and ceremonies often demand sophisticated lighting and sound effects to enhance the atmosphere. Traditionally, these effects were controlled manually, leading to inefficiency, delays, and high energy consumption. With the advancement of microcontrollers like Arduino and sensors such as ultrasonic and PIR, it has become possible to automate these processes in a cost-effective and reliable way.

This project focuses on creating a **Smart Decorative Automation System** that integrates **Arduino Uno, ultrasonic sensors, PIR motion sensors, relays, and a DFPlayer Mini module for audio playback**. The goal is to synchronize lighting and audio effects with human presence, ensuring both energy efficiency and an enhanced decorative experience.

The system is designed for scenarios such as **Ganpati decoration, Diwali lighting, weddings, and cultural events**, where synchronized lights and music can elevate the festive spirit.

**Chapter 2: Objectives**

The objectives of this project are:

1. To design and implement an Arduino-based automated decoration system.
2. To use an ultrasonic sensor for **distance-based activation** of lights and sound.
3. To include a PIR sensor for **motion detection**, improving accuracy and responsiveness.
4. To control multiple relays for switching high-power decorative lights.
5. To integrate a DFPlayer Mini module for **background music playback**.
6. To ensure reliable operation with proper power management.
7. To make the system cost-effective, modular, and easy to deploy.

**Chapter 3: Literature Review**

Automation in lighting and decorative systems has been explored widely in recent years. Some relevant works include:

* **Home Automation Projects**: Systems that use relays and microcontrollers to switch lights and fans remotely. However, most of them lacked synchronized music integration.
* **PIR Sensor Lighting**: Many security systems use PIR sensors for detecting human presence. While effective, they are often prone to false triggers from heat sources.
* **Ultrasonic Distance Measurement**: Ultrasonic sensors are widely used for proximity detection. They provide more precise control compared to PIR when distance-based activation is required.
* **Event Management Systems**: Advanced systems use DMX controllers for large-scale events. These are expensive and complex compared to an Arduino-based approach.

From the literature, it is clear that combining **ultrasonic sensing, PIR motion detection, and audio synchronization** in a **low-cost Arduino-based project** is still an area of innovation and practical importance.

**Chapter 4: System Requirements**

**Hardware Requirements:**

1. **Arduino Uno** – The microcontroller for control logic.
2. **Ultrasonic Sensor (HC-SR04)** – For detecting distance (object within 30 cm).
3. **PIR Sensor** – For motion detection.
4. **Relay Module (6-channel)** – To control lights and decorative appliances.
5. **DFPlayer Mini Module** – To play MP3 songs from SD card.
6. **Speaker** – For audio output.
7. **External Power Supply** – 9V/12V adapter for Arduino, 5V/12V for relays.
8. **Resistors, jumper wires, breadboard, PCB** – For circuit assembly.
9. **Separate power supply for relays** to avoid overloading Arduino.

**Software Requirements:**

1. **Arduino IDE** – For coding and uploading programs.
2. **Proteus / Tinkercad** – For circuit simulation.
3. **Fritzing** – For wiring diagrams.
4. **GitHub** – For project version control and hosting.

**Chapter 5: System Design**

**Block Diagram**

Ultrasonic Sensor ----┐

│

PIR Sensor -----------│------> Arduino Uno -----> Relay Driver -----> Decorative Lights

│

DFPlayer Mini --------│------> Speaker (Music Output)

* The **Ultrasonic sensor** detects if someone is within 1 foot.
* The **PIR sensor** confirms motion.
* The **Arduino Uno** processes inputs and runs a timing sequence.
* The **Relay module** switches on lights at different intervals.
* The **DFPlayer Mini** plays music simultaneously.

**Circuit Description**

* Ultrasonic sensor connected to D8 (TRIG) and D9 (ECHO).
* PIR sensor connected to D12.
* Relays connected to D2–D7.
* DFPlayer Mini connected via SoftwareSerial (D10 RX, D11 TX).
* Power distribution carefully designed to prevent overloading.

**Chapter 6: Working Principle**

1. When the system powers on, all relays are OFF.
2. The ultrasonic sensor continuously measures distance. If an object is detected within 30 cm, the cycle is triggered.
3. PIR sensor ensures that motion is real before activation.
4. DFPlayer Mini starts playing background music.
5. Relays turn ON in a programmed sequence:
   * Relay 1 at 80 seconds
   * Relay 2 at 123 seconds
   * Relay 3 at 140 seconds
   * Relay 4 at 208 seconds
   * Relay 5 at 240 seconds
   * Relay 6 at 300 seconds
6. After 4 minutes, all relays turn OFF, music stops, and system resets.
7. If motion is detected again, the cycle restarts.

**Chapter 7: Hardware Components in Detail**

(Each of these should take **half to one page** in your Word file.)

1. **Arduino Uno**
   * Based on ATmega328P microcontroller.
   * 14 digital I/O pins, 6 analog pins.
   * Operates at 16 MHz.
   * Ideal for beginners due to open-source support.
2. **Ultrasonic Sensor HC-SR04**
   * Works on the principle of sound wave reflection.
   * TRIG sends ultrasonic pulses; ECHO receives the reflected signal.
   * Distance = (Time × Speed of Sound) / 2.
3. **PIR Sensor**
   * Detects infrared radiation from human bodies.
   * Wide detection angle (~120°).
   * Prevents false triggers compared to ultrasonic alone.
4. **Relay Module (6 Channel)**
   * Acts as a switch for high-power devices.
   * Each relay can control up to 10A at 250V AC.
   * Provides isolation between Arduino and lights.
5. **DFPlayer Mini**
   * MP3 module with onboard amplifier.
   * Supports microSD card with thousands of songs.
   * Simple serial communication with Arduino.
6. **Power Supply**
   * Arduino: 9V adapter.
   * Relays: External 5V/12V supply with common ground.
   * Important to separate logic and load currents.

**Chapter 8: Software Implementation**

The Arduino program is written in **C++** and uploaded using Arduino IDE.

**Key Features in Code:**

* **Ultrasonic distance measurement** using pulseIn().
* **PIR motion detection** via digital read.
* **Timers with millis()** to schedule relay activation.
* **DFPlayer library functions** for play, stop, and volume control.
* **State management** with flags (cycleRunning, relayState[]).

**Chapter 9: Flowchart**

Start → Initialize → Wait for Sensor Trigger

↓

Ultrasonic detects < 30 cm?

↓ Yes

PIR detects motion?

↓ Yes

Start Cycle → Play Audio → Activate Relays Sequentially

↓

After 4 minutes → Turn OFF All → Stop Music → Reset

**Chapter 10: Applications**

1. **Festival Decorations** – Ganesh Chaturthi, Diwali, Christmas.
2. **Wedding Ceremonies** – Automated lighting with music.
3. **Temple Lighting Systems** – Synchronization with devotional songs.
4. **Event Management** – Stage decoration and effects.
5. **Smart Homes** – Automated ambiance creation.

**Chapter 11: Advantages**

* Cost-effective solution compared to professional DMX systems.
* Modular and scalable.
* Reduces manual intervention.
* Energy efficient (only works when people are nearby).
* Enhances user experience with synchronized music.

**Chapter 12: Limitations**

* Arduino Uno has limited memory and I/O pins.
* Ultrasonic sensors can be affected by temperature and soft surfaces.
* PIR sensors may trigger false positives in crowded environments.
* System is not waterproof, hence unsuitable for outdoor unprotected use.

**Chapter 13: Future Scope**

* Integration with **Bluetooth/WiFi** for remote control.
* Use of **voice assistants (Alexa/Google Home)**.
* **Mobile App** for customization of relay timings.
* Use of **addressable RGB LEDs (WS2812B)** for advanced lighting.
* Solar power supply for eco-friendly operation.
* Expansion to larger event halls with DMX protocol.

**Chapter 14: Testing and Results**

* Tested with **six relays controlling LED bulbs** in Tinkercad and hardware.
* Audio playback synchronized successfully.
* Ultrasonic detection range optimized to **30 cm** for reliability.
* Relay sequence worked as per custom timing schedule.
* System reset properly after 4 minutes.

**Chapter 15: Conclusion**

This project demonstrates the potential of low-cost microcontrollers in building advanced decorative systems. By integrating ultrasonic and PIR sensors, relays, and DFPlayer Mini, the project creates a **smart, responsive, and synchronized decoration system** suitable for various cultural and social events. The solution is efficient, scalable, and adaptable to future enhancements such as wireless control and smart home integration.

**Chapter 16: References**

1. Arduino Official Documentation – https://www.arduino.cc
2. DFPlayer Mini Datasheet – DFRobot
3. HC-SR04 Ultrasonic Sensor Datasheet
4. PIR Motion Sensor Technical Guide
5. Open-source Arduino Relay Projects on GitHub