IoT - Based Tobacco Curing Chamber

Thank you for taking the time to visit my portfolio!

🌿 IoT-Based Tobacco Curing Chamber

📌 Project Overview

The **IoT-Based Tobacco Curing Chamber** is a smart agricultural automation project designed to modernize the traditional tobacco curing process. Conventional curing methods rely heavily on burning firewood, require continuous manual labor, cause deforestation, and take **7–10 days** to complete.

This project replaces the traditional approach with an **automated, sensor-based curing system** using **ESP8266, temperature and humidity control, heater coils, and exhaust fans**, all monitored and configured through a **web application integrated with Firebase**.

The system reduces curing time to **4 days**, eliminates the use of firewood, minimizes labor health risks, and ensures consistent curing quality.

🎯 Objectives

- Automate tobacco curing using IoT
- Maintain precise temperature and humidity levels
- Reduce curing time from 7–10 days to 4 days
- Eliminate wood usage and prevent deforestation

- Allow remote temperature scheduling through a web page
- Improve worker safety and curing consistency

Problems in Traditional Tobacco Curing

- Large-scale **deforestation** due to firewood usage
- High **labor effort** and exposure to smoke and heat
- Poor temperature and humidity control
- Longer curing duration
- Inconsistent quality of cured tobacco leaves

💡 **Proposed Smart Solution**

The IoT-based curing chamber uses:

- **Heater coils** for controlled heat generation
- **Draft fan** to distribute heat uniformly
- **Top exhaust fan** to remove excess moisture and humid air
- **PT100 temperature sensor** for accurate temperature measurement
- **ESP8266 (NodeMCU)** for IoT connectivity

- **Web page with Firebase** to set temperature values day-wise

🛠️ Hardware Components Used

Component	Description
ESP8266 (NodeMCU)	Main IoT controller
PT100 Temperature Sensor	Accurate temperature sensing
Humidity Sensor	Detects moisture content
Heater Coils	Generates required heat
Draft Fan	Circulates hot air
Top Exhaust Fan	Removes humid air
Relay Module	Controls heaters and fans
Power Supply	System power

🌐 Software & Technologies

- Arduino IDE
- Embedded C / C++

- Firebase Realtime Database
- Web Page (HTML/CSS/JS)
- ESP8266 Wi-Fi communication

🔌 Working Principle

🔥 Temperature Control

- If chamber temperature falls **below the set value**:
    - Heater coils turn ON
    - Draft fan turns ON to distribute heat
- If temperature reaches the desired value:
    - Heater coils turn OFF

💨 Humidity Control

- If humidity at the top of the chamber becomes **too high**:
    - Top exhaust fan turns ON
    - Excess moisture and unwanted air are pushed out

🌍 Remote Control

- Temperature values are set through a **web page**

- Data is stored and synced using **Firebase**
- ESP8266 fetches real-time setpoints automatically

📅 Day-Wise Temperature Scheduling

Day	Temperature Setpoint
Day 1	55°C
Day 2	49°C
Day 3	38°C
Day 4	30°C

> These values can be **modified remotely** using the web interface.

🔁 System Flow

1. User sets temperature values on web page
2. Firebase updates real-time database
3. ESP8266 reads temperature setpoint
4. PT100 sensor measures chamber temperature

5. Heater & fans operate automatically

6. Humidity is controlled using top exhaust fan

📈 Results & Performance

Parameter	Traditional Method	Proposed System
Curing Time	7–10 days	**4 days**
Wood Usage	High	**Zero**
Labor Risk	High	**Low**
Temperature Accuracy	Poor	**High**
Environmental Impact	High	**Eco-friendly**

🌍 Environmental & Social Impact

- 🌳 Prevents deforestation (no firewood)
- 👷 Improves labor safety (less smoke & heat exposure)
- ⚡ Energy-efficient curing
- 📉 Reduces curing losses
- 📈 Improves tobacco quality consistency

📦 Applications

- Tobacco curing chambers
- Agricultural drying systems
- Herbal leaf drying
- Smart agro-processing units

⚠️ Limitations

- Requires stable power supply
- Internet connectivity needed for web control
- Single-chamber implementation

🚀 Future Enhancements

- Mobile application support
- Multiple chamber control
- Automated humidity scheduling
- Solar power integration
- Data analytics for curing optimization

📸 Project Outputs

- ✔️ Fully working curing chamber prototype
- ✔️ Web page with Firebase integration
- ✔️ Project demonstration video
- ✔️ Group photographs
- ✔️ Project completion certificate

🏁 Conclusion

The **IoT-Based Tobacco Curing Chamber** demonstrates how smart automation can replace traditional, inefficient, and environmentally harmful agricultural practices. By integrating IoT, sensors, and web-based control, the project achieves faster curing, better quality, reduced labor risk, and zero deforestation.

> **"Smart agriculture is not just about automation—it's about sustainability."**

Code:

```
#include <ESP8266WiFi.h>

#include <FirebaseESP8266.h>

#include <DHT.h>


/* ---------- WIFI DETAILS ---------- */

#define WIFI_SSID "YOUR_WIFI_NAME"

#define WIFI_PASSWORD "YOUR_WIFI_PASSWORD"


/* ---------- FIREBASE DETAILS ---------- */
```

```
#define FIREBASE_HOST "your-project.firebaseio.com"
#define FIREBASE_AUTH "YOUR_FIREBASE_DATABASE_SECRET"

/* ---------- PIN DEFINITIONS ---------- */
#define HEATER_RELAY D1
#define DRAFT_FAN    D2
#define TOP_FAN      D3

#define DHTPIN D4
#define DHTTYPE DHT11   // or DHT22

#define PT100_PIN A0    // Analog PT100 module

/* ---------- OBJECTS ---------- */
FirebaseData fbdo;
DHT dht(DHTPIN, DHTTYPE);

/* ---------- VARIABLES ---------- */
float currentTemp = 0;
float humidity = 0;
float setTemp = 0;
float humidityLimit = 70;

void setup() {
  Serial.begin(9600);

  pinMode(HEATER_RELAY, OUTPUT);
  pinMode(DRAFT_FAN, OUTPUT);
```

```
  pinMode(TOP_FAN, OUTPUT);

  digitalWrite(HEATER_RELAY, HIGH);
  digitalWrite(DRAFT_FAN, HIGH);
  digitalWrite(TOP_FAN, HIGH);

  dht.begin();

  /* ---------- WIFI ---------- */
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to WiFi");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\nWiFi Connected");

  /* ---------- FIREBASE ---------- */
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  Firebase.reconnectWiFi(true);

  Serial.println("System Ready");
}

void loop() {

  /* ---------- READ SENSORS ---------- */
  int pt100Value = analogRead(PT100_PIN);
```

```
currentTemp = map(pt100Value, 0, 1023, 0, 100); // Approximation
humidity = dht.readHumidity();

Serial.print("Temp: ");
Serial.print(currentTemp);
Serial.print(" °C | Humidity: ");
Serial.println(humidity);

/* ---------- READ FIREBASE ---------- */
if (Firebase.getFloat(fbdo, "/curing/setTemp")) {
  setTemp = fbdo.floatData();
}

if (Firebase.getFloat(fbdo, "/curing/humidityLimit")) {
  humidityLimit = fbdo.floatData();
}

/* ---------- TEMPERATURE CONTROL ---------- */
if (currentTemp < setTemp) {
  digitalWrite(HEATER_RELAY, LOW);   // Heater ON
  digitalWrite(DRAFT_FAN, LOW);      // Draft Fan ON
} else {
  digitalWrite(HEATER_RELAY, HIGH);  // Heater OFF
  digitalWrite(DRAFT_FAN, HIGH);     // Draft Fan OFF
}

/* ---------- HUMIDITY CONTROL ---------- */
if (humidity > humidityLimit) {
```

```
   digitalWrite(TOP_FAN, LOW);        // Exhaust Fan ON
  } else {
   digitalWrite(TOP_FAN, HIGH);       // Exhaust Fan OFF
  }

  delay(2000);
}
```

HTML / CSS / JS :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>IoT Tobacco Curing Chamber</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <!-- Firebase SDKs -->
  <script src="https://www.gstatic.com/firebasejs/8.10.0/firebase-app.js"></script>
  <script src="https://www.gstatic.com/firebasejs/8.10.0/firebase-database.js"></script>

  <style>
    body {
      font-family: Arial, sans-serif;
      background: #f4f6f8;
      margin: 0;
      padding: 0;
    }
```

```
header {
  background: #2e7d32;
  color: white;
  padding: 15px;
  text-align: center;
}

.container {
  max-width: 600px;
  margin: 20px auto;
  background: white;
  padding: 20px;
  border-radius: 8px;
  box-shadow: 0 0 10px rgba(0,0,0,0.1);
}

h2 {
  text-align: center;
  color: #333;
}

label {
  display: block;
  margin-top: 15px;
  font-weight: bold;
}

input, select, button {
```

```
  width: 100%;
  padding: 10px;
  margin-top: 5px;
  font-size: 16px;
}

button {
  background: #2e7d32;
  color: white;
  border: none;
  border-radius: 5px;
  cursor: pointer;
  margin-top: 20px;
}

button:hover {
  background: #1b5e20;
}

.status {
  margin-top: 20px;
  padding: 10px;
  background: #e8f5e9;
  border-left: 5px solid #2e7d32;
}

footer {
  text-align: center;
```

```
    margin-top: 30px;
    color: #777;
    font-size: 14px;
   }
  </style>
</head>

<body>

<header>
  <h1>IoT-Based Tobacco Curing Chamber</h1>
  <p>Web Control Panel</p>
</header>

<div class="container">

  <h2>Temperature & Humidity Control</h2>

  <label for="daySelect">Select Day</label>
  <select id="daySelect">
    <option value="55">Day 1 – 55°C</option>
    <option value="49">Day 2 – 49°C</option>
    <option value="38">Day 3 – 38°C</option>
    <option value="30">Day 4 – 30°C</option>
  </select>

  <label for="temp">Set Temperature (°C)</label>
  <input type="number" id="temp" placeholder="Enter Temperature">
```

```
  <label for="humidity">Humidity Limit (%)</label>
  <input type="number" id="humidity" value="70">

  <button onclick="updateValues()">Update Values</button>

  <div class="status" id="statusBox">
    Status: Waiting for input...
  </div>

</div>

<footer>
  © IoT Tobacco Curing Chamber | ESP8266 + Firebase
</footer>

<script>
  // 🔥 Firebase Configuration (REPLACE WITH YOUR DETAILS)
  var firebaseConfig = {
    apiKey: "YOUR_API_KEY",
    authDomain: "your-project.firebaseapp.com",
    databaseURL: "https://your-project.firebaseio.com",
    projectId: "your-project",
    storageBucket: "your-project.appspot.com",
    messagingSenderId: "XXXXXXX",
    appId: "XXXXXXX"
  };
```

```
// Initialize Firebase
firebase.initializeApp(firebaseConfig);

var database = firebase.database();

// Auto-fill temperature when day selected
document.getElementById("daySelect").addEventListener("change", function () {
  document.getElementById("temp").value = this.value;
});

function updateValues() {
  var tempValue = document.getElementById("temp").value;
  var humidityValue = document.getElementById("humidity").value;

  if (tempValue === "" || humidityValue === "") {
    alert("Please enter all values");
    return;
  }

  database.ref("curing").set({
    setTemp: parseFloat(tempValue),
    humidityLimit: parseFloat(humidityValue)
  });

  document.getElementById("statusBox").innerHTML =
    "Updated Successfully ✔<br>Temperature: " + tempValue +
    " °C<br>Humidity Limit: " + humidityValue + " %";
}
```

</script>

</body>

</html>