Smart brake wear Monitoring System

**🛞 1. Problem Statement**

**Brake pad wear** is often overlooked in two-wheelers, leading to sudden brake failure and road accidents.  
Currently, riders rely on manual inspections or performance drop, both of which are unreliable and reactive.

**💡 2. Proposed Solution**

Design a sensor-based IoT system that:

* Continuously monitors **brake pad thickness or pressure response**
* Sends **real-time alerts** to the rider via a dashboard LED or mobile app
* Logs wear data to **predict maintenance schedules**
* Uses **Bluetooth or WiFi** for communication between device and phone/cloud

**⚙️ 3. System Architecture**

Brake → Sensor → Microcontroller → Bluetooth/WiFi → Mobile App + Cloud Logging

**📦 4. Hardware Components**

| **Component** | **Purpose** |
| --- | --- |
| Linear Potentiometer / Load Cell | Detect brake wear (thickness/pressure) |
| ESP32 / Arduino Nano 33 IoT | Process & transmit data |
| Power Source (DC tap or battery) | Power the module |
| RGB LED (optional) | Visual indicator on dashboard |
| Brake housing mount | Secure sensor placement |

**📲 5. Software Modules**

**a. Mobile App**

* Real-time brake status (“Safe”, “Replace Soon”, “Critical”)
* Wear history logs
* Alert notifications
* Garage finder / schedule tracker

**b. Firmware (Microcontroller)**

* Read sensor data
* Apply threshold logic or ML
* Send alerts via Bluetooth

**c. Cloud Backend (optional)**

* Store user history
* Push maintenance reminders
* Aggregate wear pattern analytics (for fleets or garages)

**🌐 6. Communication Protocols**

| **Function** | **Protocol** | **Usage Scenario** |
| --- | --- | --- |
| Device ↔ App | Bluetooth LE | Local alerts & setup |
| Device ↔ Cloud | MQTT / HTTP | Remote logging & maintenance data |
| App ↔ Cloud | REST API | Display user history |
| OTA Updates (Optional) | WiFi / BLE | Update firmware remotely |

**🤖 7. Optional ML Model**

* Inputs: Brake pressure, ride time, braking frequency
* Output: Remaining brake pad life
* Toolchain: **Edge Impulse**, **TinyML**, or basic regression in Python
* Deployment: On-device inference or cloud processing

**🧠 8. Unique Value Proposition**

* Two-wheeler focus (rare in existing systems)
* Compact, retrofit-friendly module
* Bluetooth-based alerts—ideal for mobile-first users
* Predictive maintenance = safer roads + smarter riding

**⏱️ 9. Development Timeline**

| **Phase** | **Time Estimate** |
| --- | --- |
| Hardware setup | 2–3 hours |
| Sensor integration | 3–4 hours |
| Firmware coding | 4 hours |
| Mobile app UI + alerts | 5 hours |
| Testing & final demo | Remaining time |

🚀 **Conclusion**

The Smart Brake Wear Monitoring System stands as a forward-thinking innovation aimed at enhancing road safety, especially for two-wheeler riders. By integrating real-time wear detection, predictive maintenance alerts, and user-friendly mobile connectivity, this solution offers a proactive approach to one of the most critical aspects of vehicle health: braking efficiency.