FindItBack – Lost & Found AI-Powered App

# ✅ Project Summary

FindItBack is an AI-powered campus lost and found app that helps students post lost/found items, match them using image recognition, and contact the uploader to return the item easily.

# 💡 Features Implemented

* 🖼️ Image Upload (Lost & Found): Users upload item photos with metadata.
* 🤖 AI-Based Image Matching: MobileNetV2 + Cosine Similarity to find similar items.
* 📍 Place Tagging: Replaces category with actual place of lost/found.
* 📅 Date Logging: Date picker for event date.
* 👤 Uploader Info & Contact: Name and phone number stored with post.
* 🔎 Smart Search: Search by name, place, or item ID.
* 📈 Top 3 Match Display: Shows top 3 matches based on visual similarity.
* 🖼️ Gallery Display: Shows thumbnails of all lost/found items.
* 🔐 Robust Error Handling: Handles missing files, data errors safely.

# ⚙️ How It Works

1. 1. Upload: Users upload image + metadata. Stored in folders and items.json.
2. 2. Feature Extraction: MobileNetV2 gets embedding vector of the image.
3. 3. Matching: Cosine similarity compares new image to existing ones.
4. 4. Search: Searches items.json based on user input (name/place/ID).
5. 5. Display: Matched results shown with uploader info and image.

# 🧠 Technologies Used

* Streamlit – Frontend & backend app interface
* TensorFlow (MobileNetV2) – Image feature extraction
* scikit-learn – Cosine similarity computation
* Pillow – Image I/O and processing
* JSON – Metadata storage
* uuid – Unique item identification

# 🎤 Presentation / Demo Script

We built FindItBack, a smart Lost & Found app for campuses that uses AI-powered image matching to help return lost items faster. Users can upload lost or found item photos with details like place and date, and our system shows top visual matches from the other category. We also include contact details to speed up item return. Everything is searchable by name, place, or item ID, and we've built the app using Streamlit, TensorFlow, and Python. It’s simple, scalable, and student-friendly.