# **ShareBox**

**ShareBox** is a sustainability-focused donation platform that allows users to donate, browse, and request gently used items, promoting a circular economy. Users can upload items, categorize them, and help reduce waste while supporting their local communities.

## **Features**

* User Authentication & Profile Management
* Upload and browse items by categories (Clothing, Electronics, General, etc.)
* Image detection and classification using ML
* Quality checks to ensure donated items meet condition standards
* Firebase Realtime Database and Storage integration
* Mobile-first design with React Native
* Admin dashboard for managing donations

## **Tech Stack**

* **Frontend**: React Native
* **Backend**: Firebase (Auth, Realtime Database, Storage)
* **ML Models**: TensorFlow / scikit-learn for classification and defect detection
* **Testing**: Selenium, Unit tests
* **CI/CD**: GitHub Actions (optional)

## **Getting Started**

### **Prerequisites**

* Node.js & npm
* Firebase project setup
* Android Studio / Xcode for mobile testing
* Firebase CLI
* Python interpreter
* Libraries
  + Torch
  + Pandas
  + Numpy
  + Jupiternotebooks: optional

from torchvision import datasets, models, transforms

from torch.utils.data import Dataset, DataLoader

import torch

import cv2

import os

### **Installation**

git clone https://github.com/sharebox.git

cd sharebox

npm install

### **Firebase Configuration**

1. Create a Firebase project
2. Enable:  
   * Authentication (Email/Password)
   * Realtime Database
   * Firebase Storage
3. Replace firebaseConfig in the code with your project credentials

**Usage**

* Launch the app: npx expo start
* Create an account and log in
* Upload an item with an image and name
* ShareBox classifies and verifies the item condition
* Other users can view and request items

## **How it Works**

* **Image Classification**: ML model analyzes image to verify item type
* **Text Classification**: Item name and description are analyzed for consistency
* **Validation**: Items must meet specific condition rules to be accepted

## **Testing**

npm run test

Selenium is used for UI testing and form validation. Unit tests cover classification logic and database interaction.

## **Project Structure**

/components

/ItemCard

/UserProfile

/screens

HomeScreen.js

UploadScreen.js

Settings.js

/utils

firebaseConfig.js

imageClassifier.js

textClassifier.js

## **License**

MIT License. Feel free to fork, improve, and contribute!

## **Contributing**

Pull requests are welcome! For major changes, please open an issue first to discuss what you would like to change.

Let me know if you want to tailor this for Expo, add CI/CD steps, or link up your actual GitHub repo.