Project Report Format

1.INTRODUCTION

Project Title:

CleanTech: Transforming Waste Management with Transfer Learning

1.1 project view:

CleanTech: Transforming Waste Management with Transfer Learning is a forward-thinking AI-powered solution that automates the classification of municipal waste using image recognition and transfer learning techniques. The project addresses inefficiencies in traditional waste segregation by enabling users—individuals, households, or municipalities—to upload images of waste and receive instant classification results (e.g., biodegradable, non-biodegradable, recyclable).

1.2 purpose:

The purpose of the **CleanTech** project is to modernize municipal waste management by using **transfer learning** to automate the classification of waste through image recognition. This system enables users to quickly and accurately identify whether waste is biodegradable, non-biodegradable, or recyclable, reducing human error, improving recycling rates, and supporting smart city sustainability goals. The project promotes environmental awareness while streamlining waste processing for both individuals and municipalities.

2.IDEATION PHASE

2.1Problem Statement

Urban areas face increasing challenges in managing the growing volume and complexity of municipal waste. Manual sorting is inefficient, error-prone, and lacks scalability. Current AI-based systems often require large datasets and are not easily adaptable across regions.

Thinks	**“Is this recyclable or biodegradable?”“I wish there were a quicker way to know.”**
Feels	Confused by unclear waste categoriesFrustrated by lack of consistent guidance
Says	“I want to help the environment, but I’m not sure how.”“There’s no time to sort it manually.”
Does	Tosses waste into general binsGuesses classification or avoids sorting

2.2 Empathy Map Canvas

Thinks	**“Is this recyclable or biodegradable?”“I wish there were a quicker way to know.”**
Feels	Confused by unclear waste categoriesFrustrated by lack of consistent guidance
Says	“I want to help the environment, but I’m not sure how.”“There’s no time to sort it manually.”
Does	Tosses waste into general binsGuesses classification or avoids sorting

2.3 Brainstorming

Method Used:
Open, collaborative digital brainstorming using idea clustering and prioritization (based on feasibility vs. impact).

Idea Group	Ideas
AI/ML	Use transfer learning with pre-trained models for classifying waste images
UX/UI	Build a mobile-first web app with a drag-and-drop interface and instant results
Integration	Connect with municipal dashboards for real-time waste analytics
Engagement	Add gamification and user rewards for proper waste sorting
Education	Provide interactive visual guides and eco-friendly tips
Deployment	Use Flask for backend + Ngrok for quick demos; later migrate to cloud APIs

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

Stage	User Actions	Touchpoints	Pain Points	Opportunities
Awareness	Learns about CleanTech through social media or demo	Social media, website landing page	Lack of clarity on how it works	Clear value proposition and educational content
Consideration	Visits platform, explores how it works	Home, About sections	Doubts on accuracy or benefit	Provide use cases and sample predictions
Engagement	Uploads an image for waste classification	Prediction interface	Confusion about uploading or results interpretation	Provide tooltips, confidence score, and image preview
Action	Gets waste type result and takes correct disposal action	Results section	Uncertainty about disposal instructions	Show proper bin type, disposal guide, and eco tips

Stage	User Actions	Touchpoints	Pain Points	Opportunities
Feedback & Loop	Provides feedback, checks history, or returns for another session	Feedback form, history (future)	No persistent storage of feedback or session	Add login + history in future version

3.2 Solution Requirements

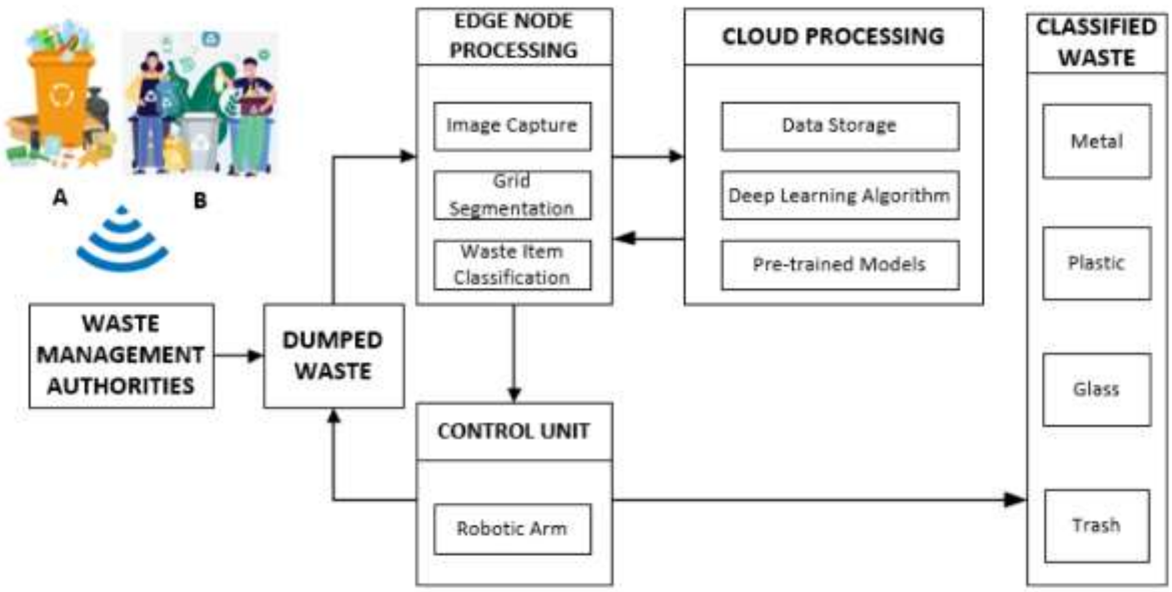
🔷 *Functional Requirements*

- Upload an image and classify it as biodegradable/non-biodegradable/recyclable
- Simulate or return real-time predictions
- Display classification result with visual feedback
- Responsive design compatible with mobile and desktop
- Admin interface (planned) for monitoring and logs
- Future: User login, history, and feedback system

🔷 *Non-Functional Requirements*

- Fast response time (< 2 seconds per prediction)
- Scalable deployment using cloud/containerization
- Privacy-friendly (no personal data collected in MVP)
- Modular, maintainable codebase

3.3 Data Flow Diagram



3.4 Technology Stack

Layer	Technology	Purpose
Frontend	HTML5, Tailwind CSS, JavaScript	Responsive UI, file upload, routing, and modals
Backend	Flask (Python), Pyngrok	Serve HTML, simulate backend logic, expose via public URL
ML Engine	(Planned) TensorFlow / PyTorch + Transfer Learning	Waste image classification using pre-trained CNN models
Database	(Planned) MongoDB	Store classification logs, user history, and feedback
Deployment	Ngrok, Localhost (Dev)	Quick public demo access
Future Additions	Docker, Cloud (Render, Heroku, etc.)	Scalable, production-ready deployment

4. PROJECT DESIGN

4.1 Problem Solution Fit

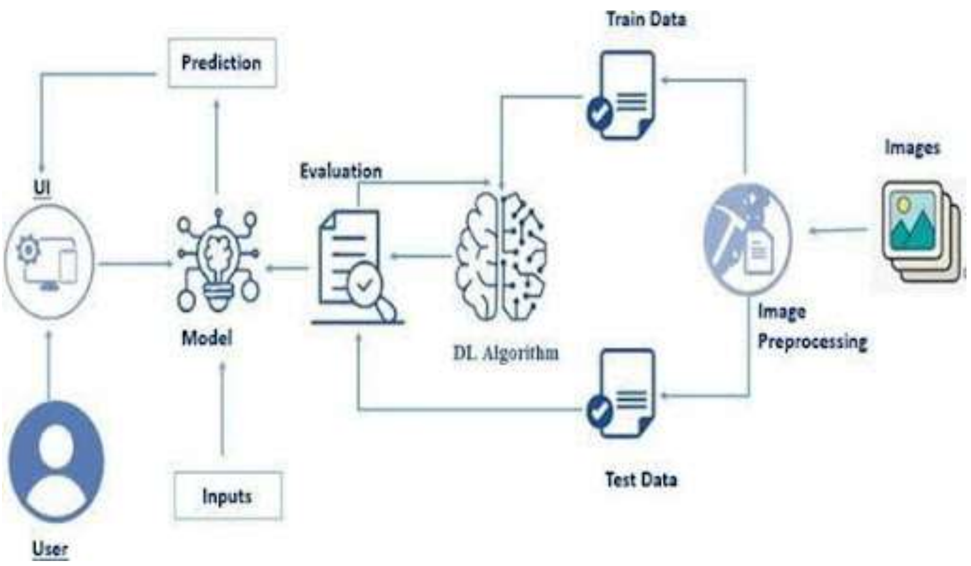
Problem Recap:

Manual waste classification is inefficient, error-prone, and lacks scalability. Citizens often lack clarity on how to properly dispose of waste, and municipalities face challenges in optimizing waste segregation and recycling processes.

4.2 Proposed Solution

CleanTech offers a lightweight, AI-driven platform where users can upload images of waste and instantly receive classification results. This empowers individuals to make informed disposal decisions, enhances recycling accuracy, and creates a data-driven foundation for smarter urban waste management.

4.3 Solution Architecture



5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

The CleanTech project is structured into agile development phases to ensure rapid prototyping, iterative improvement, and future scalability. The planning focuses on delivering a functional MVP (Minimum Viable Product) while preparing for machine learning integration and production deployment.

Phase	Timeline	Milestones
Phase 1: Research & Ideation	Week 1	Problem definition, empathy mapping, solution brainstorming
Phase 2: MVP UI Development	Week 2	Static frontend (HTML/CSS/JS), responsive layout, file upload preview
Phase 3: Flask Backend Setup	Week 3	Flask server setup, ngrok tunneling, simulated prediction logic
Phase 4: Model Integration (Planned)	Week 4–5	Connect pre-trained transfer learning model, create `/predict` API
Phase 5: Database Integration (Planned)	Week 6–7	MongoDB setup for storing history and feedback
Phase 6: Testing & Optimization	Week 8	UI testing, API testing, user testing with sample data
Phase 7: Final Deployment & Demo	Week 9	Host app, prepare project documentation and final presentation/demo

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

Performance testing ensures that the CleanTech application delivers a smooth and responsive user experience under expected workloads. For this MVP version, the focus is on frontend responsiveness, backend response time, and simulated load handling.

Component	Test Type	Tool / Method	Observation
Frontend UI	Load & responsiveness	Manual + Chrome DevTools (Lighthouse)	Fast image preview & section scrolls
Flask Backend	API response time	`time curl http://localhost:5000`	Initial load < 500ms (static content)
Ngrok Tunnel	External access latency	Browser testing + mobile network access	Average delay: ~1s (acceptable demo)
Future Prediction API	Stress testing (planned)	Locust / Postman + mock payloads	Scheduled post-ML integration

7. RESULTS

7.1 Output Screenshots

```
Class: Biodegradable Images
Image: TRAIN.2_BIODEG_ORI_11280.jpg
```

```
Class: Recyclable Images
Image: metal179.jpeg
```

```
Class: Trash Images
Image: trash68.jpeg
```


```
Class: Recyclable Images
Image: cardboard130.jpeg
```

8. ADVANTAGES & DISADVANTAGES

🔷 Advantages

1. **AI-Driven Automation**
   Uses transfer learning to reduce manual effort and errors in waste classification.
2. **User-Friendly Interface**
   Clean and responsive UI with image upload and real-time feedback ensures ease of use across all devices.
3. **Scalable Architecture**
   Modular design allows future integration with cloud APIs, databases, and advanced ML models.
4. **Environmentally Impactful**
   Promotes better waste segregation and supports municipal sustainability goals.
5. **Rapid Deployment**
   Lightweight Flask + Ngrok setup allows fast prototyping and remote access without heavy infrastructure.
6. **Open for Customization**
   Built with flexibility to accommodate additional features like gamification, dashboards, and authentication.

🔻 Disadvantages

1. **No Real ML Model Yet**
   Current version uses simulated predictions; real-time classification requires model integration.
2. **No Persistent Data Storage**
   Absence of database limits functionality like user history or analytics.
3. **Security Limitations**
   No authentication or access control makes the system open to unauthorized use during demo.
4. **Dependent on Ngrok**
   Temporary and unstable tunnel for public access; not suitable for production.
5. **Lack of Mobile App**
   Currently a web-only solution; lacks native app support for broader accessibility.

9. CONCLUSION

The **CleanTech** project presents a modern, scalable, and environmentally impactful approach to tackling the challenges of municipal waste management. By leveraging **transfer learning** and a clean web-based interface, the solution simplifies waste classification for users while setting the foundation for smarter and more sustainable urban ecosystems.

Even in its MVP stage, CleanTech demonstrates strong potential by simulating real-time waste classification, offering a responsive frontend, and planning for seamless backend expansion. It reflects a practical blend of AI innovation and public utility.

10. FUTURE SCOPE

The CleanTech platform is built with modularity and growth in mind. Several future enhancements can elevate the solution from a demo-ready prototype to a fully deployable smart waste management system:

🔶 Technical Expansion

- **Integrate Real-Time ML Model**: Deploy a trained CNN using TensorFlow or PyTorch via REST API.
- **Database Connectivity**: Use MongoDB to store classification logs, user feedback, and admin analytics.
- **User Authentication**: Enable secure user access with role-based features (citizen, admin, auditor).
- **Cloud Deployment**: Migrate from Ngrok to cloud platforms (e.g., Heroku, AWS, Render) for 24/7 access.

🔶 Feature Enhancements

- **Mobile App Development**: Build a cross-platform app for field workers and public users.
- **Gamification & Rewards**: Introduce user incentives to promote consistent participation.
- **Real-Time Analytics Dashboard**: Equip municipalities with dashboards to monitor trends and optimize collection routes.
- **Smart Bin Integration**: Connect with IoT-enabled waste bins for automated sorting and reporting.

11. APPENDIX

Source Code

```
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import ResNet50
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D,
Dropout
from tensorflow.keras.optimizers import Adam
import matplotlib.pyplot as plt

# Set up paths
train_dir = 'dataset/train'    # e.g., dataset/train/eosinophil,
dataset/train/lymphocyte...
val_dir = 'dataset/val'        # e.g., dataset/val/...

# Set up parameters
img_size = (224, 224)
batch_size = 32
num_classes = 3  # Change this if you have a different number of
classes

# Data generators with augmentation
train_datagen = ImageDataGenerator(
    rescale=1./255,
```

```
    zoom_range=0.2,
    horizontal_flip=True,
    rotation_range=20,
    shear_range=0.2
)

val_datagen = ImageDataGenerator(rescale=1./255)

train_gen = train_datagen.flow_from_directory(
    train_dir,
    target_size=img_size,
    batch_size=batch_size,
    class_mode='categorical'
)

val_gen = val_datagen.flow_from_directory(
    val_dir,
    target_size=img_size,
    batch_size=batch_size,
    class_mode='categorical'
)

# Load pre-trained ResNet50
base_model = ResNet50(weights='imagenet', include_top=False,
input_shape=(*img_size, 3))

# Freeze base layers
for layer in base_model.layers:
    layer.trainable = False

# Add custom classification head
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dropout(0.5)(x)
x = Dense(128, activation='relu')(x)
x = Dropout(0.3)(x)
predictions = Dense(num_classes, activation='softmax')(x)

model = Model(inputs=base_model.input, outputs=predictions)

# Compile model
model.compile(optimizer=Adam(learning_rate=0.0001),
              loss='categorical_crossentropy',
              metrics=['accuracy'])

# Train model
history = model.fit(
    train_gen,
```

```
    validation_data=val_gen,
    epochs=10
)

# Save model
model.save('hematovision_model.h5')

# Plot training results
plt.plot(history.history['accuracy'], label='Train Accuracy')
plt.plot(history.history['val_accuracy'], label='Val Accuracy')
plt.title('Accuracy over epochs')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```

Dataset Link

https://www.kaggle.com/datasets/elinachen717/municipal-solid-waste-dataset

GitHub & Project Demo Link

https://github.com/alluru-lekhana/CleanTech/tree/main