EcoPackAI – AI-Powered Sustainable Packaging Recommendation System

Author: Nilesh Gawhale
Infosys Springboard Internship – Batch 11

---

1. Abstract

EcoPackAI is an AI-driven decision support system designed to recommend sustainable packaging materials based on product characteristics and sustainability priorities. The system integrates Machine Learning models for cost and CO_2 emission prediction, applies a weighted ranking framework, and provides an interactive dashboard for visualization.

The application is built using Flask (backend), HTML/Bootstrap (frontend), PostgreSQL (cloud database), and deployed using Render with production-ready configuration.

The project demonstrates the end-to-end lifecycle of an AI system: data processing, model training, backend integration, dashboard visualization, database logging, and cloud deployment.

---

2. Problem Statement

Sustainable packaging selection is a complex decision-making process involving trade-offs between:

- Cost
- Environmental impact (CO_2 emissions)
- Fragility requirements
- Shipping type
- Sustainability priority

Organizations often rely on static decision rules instead of data-driven approaches.

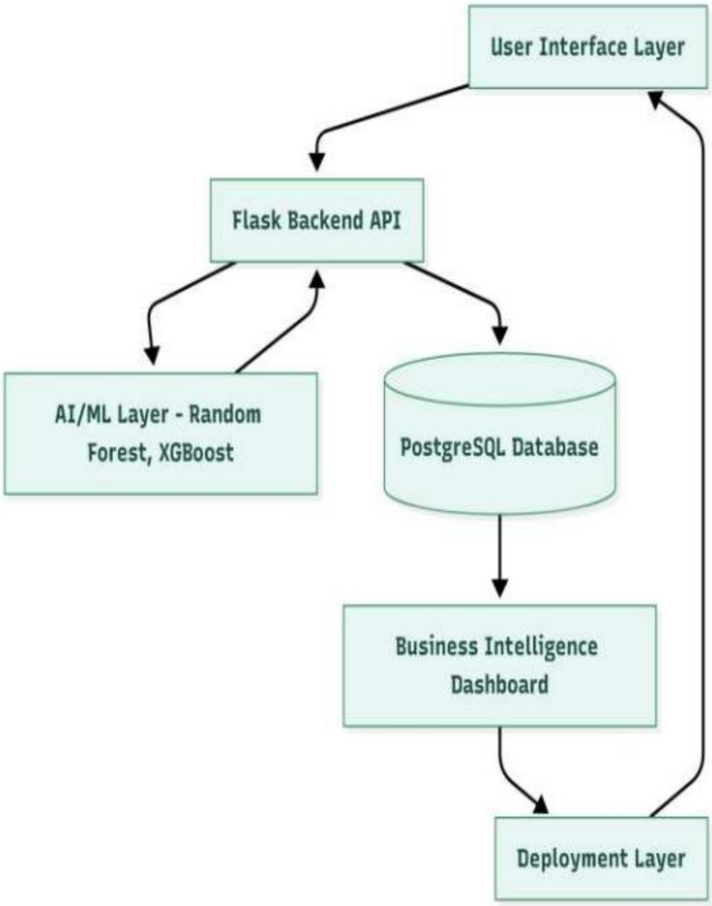
The objective of EcoPackAI is to:

- Predict packaging cost
- Predict CO_2 emission impact
- Rank materials dynamically
- Provide actionable recommendations
- Log usage data for analytics

3. System Architecture

3.1 High-Level Architecture

User (Browser)
→ Frontend (HTML + Bootstrap + JS)
→ Flask Backend
→ Feature Engineering
→ ML Models (Random Forest + XGBo
→ Ranking Engine
→ PostgreSQL (Cloud Logging)
→ Dashboard (Charts + Reports)
→ Export (Excel + PDF)



3.2 Component Description

Frontend

- Collects user inputs
- Displays recommendations
- Shows dashboard charts
- Provides export options

Backend (Flask)

- Handles API routes
- Loads ML models
- Generates predictions
- Applies ranking logic
- Logs usage to PostgreSQL

Machine Learning Layer

- Cost Prediction Model
- CO_2 Prediction Model
- Feature Scaling

Database Layer

- Cloud PostgreSQL
- Stores recommendation logs
- Tracks material usage frequency

4. Dataset Description

4.1 Dataset Name

EcoPackAI_materials.csv

4.2 Dataset Purpose

The dataset contains baseline packaging materials and their attributes, used for prediction and ranking.

4.3 Key Attributes

- Material Name
- Cost
- CO_2 Score
- Sustainability Score
- Weight-related metrics

4.4 Data Preprocessing

- Missing value handling
- Feature encoding
- Standardization using StandardScaler
- Feature vector creation based on user inputs

The dataset serves as the base reference for model prediction and ranking.

material_id	material_name	strength	weight_capacity	cost	biodegradability_score	co2_score	recyclability_percent
1	Corrugated Cardl	3	25	2	9	3	85
2	Kraft Paper	2	15	1	8	4	80
3	Recycled Paper	2	12	1	9	3	90
4	Molded Pulp	2	10	2	10	2	95
5	Bagasse	3	20	3	10	2	85
6	PLA Bioplastic	2	8	4	8	4	70
7	Cornstarch Packa	1	5	3	9	3	60
8	Bamboo Fiber	3	30	4	10	2	85
9	Palm Leaf Packag	3	18	4	10	2	80
10	Jute Fabric	3	40	5	9	3	75
11	Hemp Fiber	3	35	5	9	3	78
12	Recycled PET	3	50	6	6	5	90
13	Paperboard	2	15	2	8	4	82
14	Cellulose Film	1	4	3	9	3	65
15	Mushroom Packa	2	12	4	10	1	90
16	Seaweed Packagi	1	3	4	10	1	55
17	Sugarcane Fiber	3	25	3	10	2	88
18	Wheat Straw	2	14	3	9	3	80
19	Rice Husk Packag	2	16	3	9	3	78
20	Recycled Aluminu	3	60	6	5	6	95
21	Glass Packaging	3	70	7	4	7	100
22	Wooden Crates	3	80	9	6	6	85
23	Paper Foam	1	6	2	9	3	70
24	Biodegradable Fc	1	5	2	8	4	60
25	Cotton Fabric	2	20	4	8	4	75

5. Machine Learning Model Design

5.1 Models Used

RandomForestRegressor

Used for cost prediction.

Reasons:

- Handles nonlinear relationships
- Strong performance on structured datasets

XGBoostRegressor

Used for CO_2 prediction.

Reasons:

- Gradient boosting optimization
- High predictive performance

StandardScaler

Used for feature normalization before model inference.

```python
from xgboost import XGBRegressor

xgb_co2 = XGBRegressor(
    n_estimators=100,
    learning_rate=0.1,
    random_state=42
)

xgb_co2.fit(X_train_scaled, y_co2_train)

y_co2_pred = xgb_co2.predict(X_test_scaled)
```

```python
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import numpy as np

mae = mean_absolute_error(y_co2_test, y_co2_pred)
rmse = np.sqrt(mean_squared_error(y_co2_test, y_co2_pred))
r2 = r2_score(y_co2_test, y_co2_pred)

print("CO2 MAE:", mae)
print("CO2 RMSE:", rmse)
print("CO2 R2:", r2)
```

```
CO2 MAE: 0.09907898035916415
CO2 RMSE: 0.242359110313227
CO2 R2: 0.9596176743507385
```

5.2 Prediction Workflow

1. User selects input filters.
2. Feature vector is generated.
3. Features are scaled.
4. Cost is predicted using Random Forest.
5. CO_2 emission is predicted using XGBoost.
6. Results are normalized.
7. Ranking score is calculated.

6. Ranking Logic

The final material ranking is computed using a weighted scoring system.

Final Score =
$(W_1 \times \text{Normalized Cost}) + (W_2 \times \text{Normalized } CO_2)$

Where:

- W_1 and W_2 vary based on sustainability priority.
- Higher sustainability priority increases CO_2 weight.
- No negative CO_2 values are allowed (clipped to zero).

The system ensures:

- No equal ranking scores
- No uniform cost for all materials
- Consistent material ordering

---

7. Evaluation Metrics

Model performance was evaluated using:

- Mean Absolute Error (MAE)
- Root Mean Square Error (RMSE)
- R^2 Score

These metrics help measure:

- Prediction accuracy
- Error magnitude
- Model reliability

The models demonstrated stable performance suitable for deployment.

---

8. Dashboard Explanation

The dashboard provides:

8.1 Top 5 Recommendations

Displays best materials based on final ranking score.

8.2 Full Ranking Table

Shows all materials sorted by final score.

8.3 Visualizations

- Vertical Bar Chart – Cost comparison
- Horizontal Bar Chart – CO_2 comparison
- Line Chart – Material comparison trend
- Pie Chart – Sustainability distribution

Charts are rendered using Plotly for interactive visualization.

---

9. Database Integration (PostgreSQL)

9.1 Purpose

To log recommendation usage for analytics and tracking.

9.2 Implementation

- Cloud PostgreSQL database (Render)
- psycopg2 for connection
- Environment variable-based credentials
- No hardcoded passwords

Each recommendation request logs:

- Selected filters
- Top recommended material
- Timestamp

---

10. Security Implementation

- Environment variables for database credentials
- Debug mode disabled in production
- No exposed passwords in repository
- .gitignore prevents sensitive files from upload

---

11. Deployment Process

11.1 Local Setup

1. Clone repository
2. Install dependencies:
   pip install -r requirements.txt
3. Set .env file
4. Run:
   python app.py

11.2 Production Deployment

Platform: Render

Steps:

- Create Web Service
- Connect GitHub repository
- Add environment variables
- Configure gunicorn:
  gunicorn app:app
- Deploy application

Live deployment ensures production-ready API and dashboard access.

---

12. Results

The system successfully:

- Predicts cost and CO_2 emissions
- Dynamically ranks materials
- Displays interactive dashboard
- Logs data to PostgreSQL
- Exports Excel and PDF reports
- Runs in cloud production environment

No 500 errors were observed after final deployment.
Ranking logic works consistently.
Charts load correctly.
Application is mobile responsive.

13. Future Enhancements

- Real-time carbon emission API integration
- User authentication system
- Role-based access control
- Scalable microservices architecture
- CI/CD automation
- Multi-dataset support

---

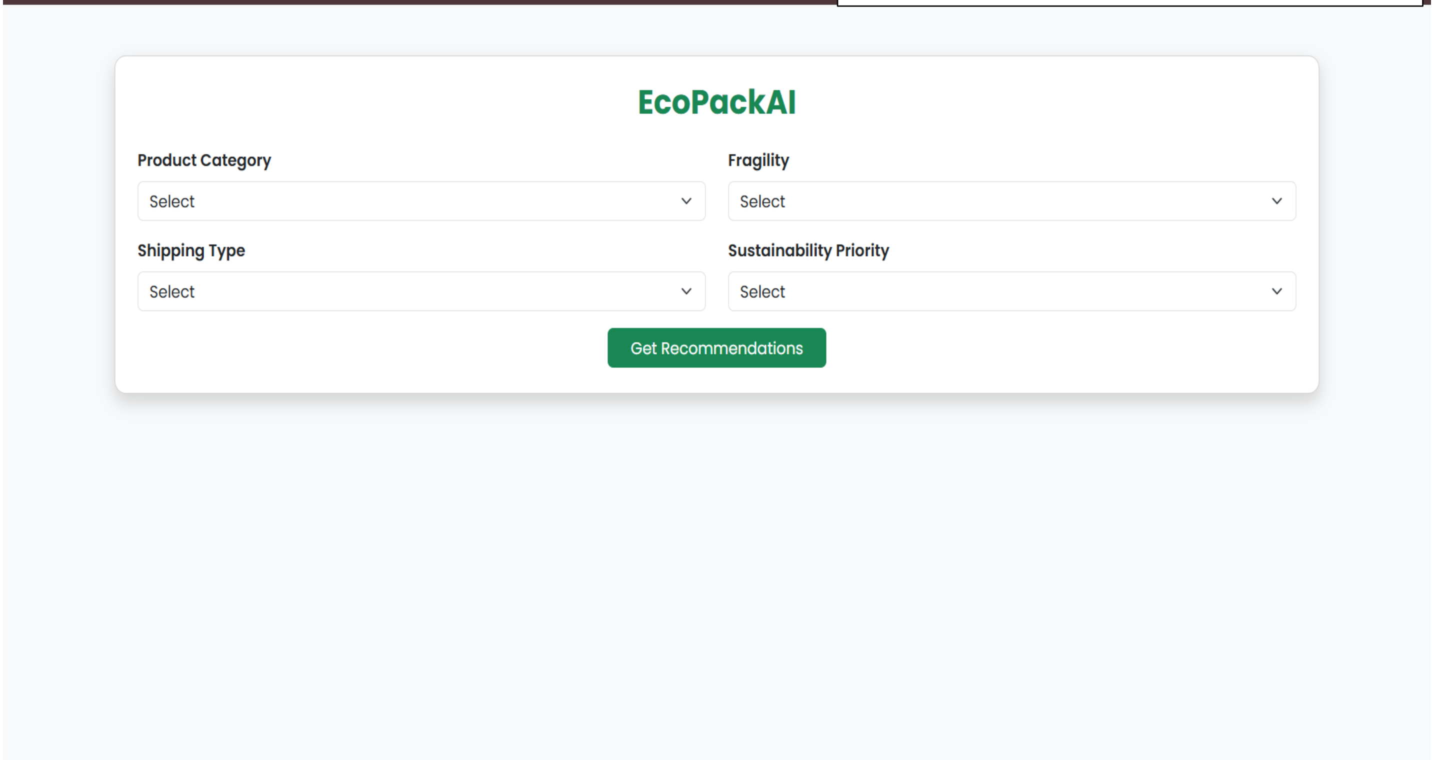
14. Conclusion

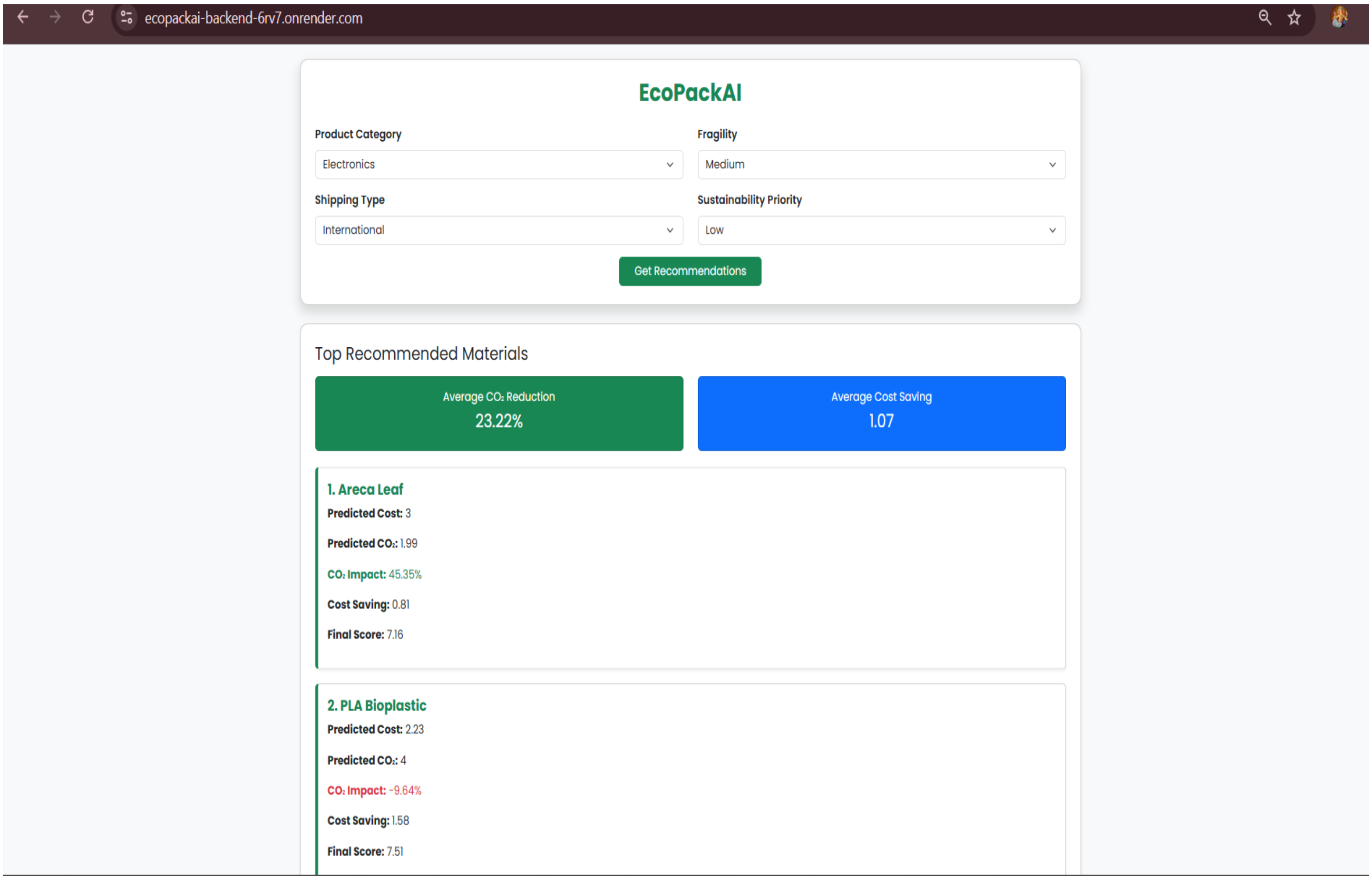
EcoPackAI demonstrates a complete AI-powered decision support system integrating machine learning, backend engineering, database logging, visualization, and cloud deployment.

The project successfully bridges theoretical ML concepts with real-world production deployment, showcasing practical AI system design and implementation.

15. Screenshots Section

DashBoard – Filters & Summary Cards





Top 5 Recommendate Materials, Comparisons & Ranking Graphs

3. Wheat Straw

Predicted Cost: 1.75

Predicted CO_2: 3

CO_2 Impact: 17.77%

Cost Saving: 2.06

Final Score: 7.53

4. Mushroom Packaging

Predicted Cost: 3

Predicted CO_2: 1.01

CO_2 Impact: 72.24%

Cost Saving: 0.81

Final Score: 7.57

5. Biodegradable Mailers

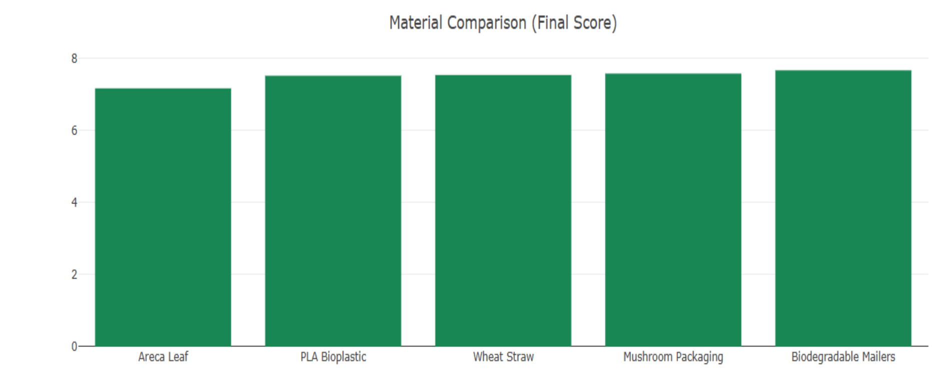
Predicted Cost: 3.74

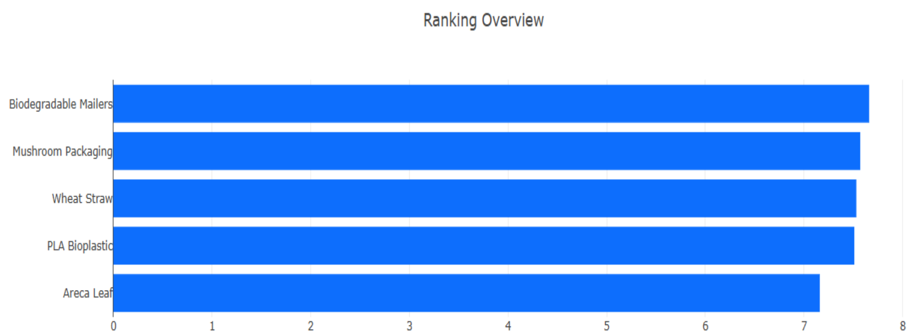
Predicted CO_2: 4

CO_2 Impact: -9.64%

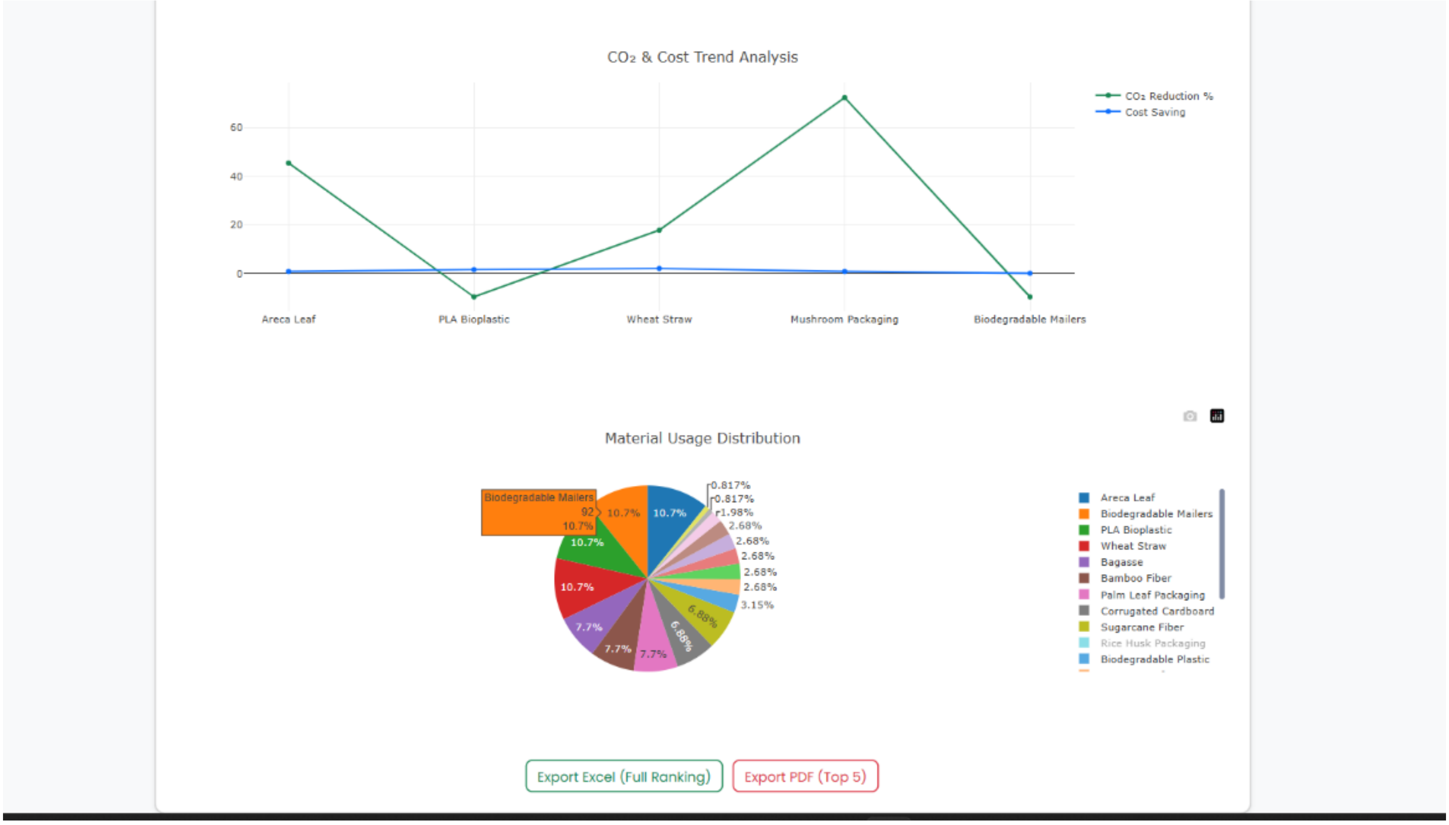
Cost Saving: 0.07

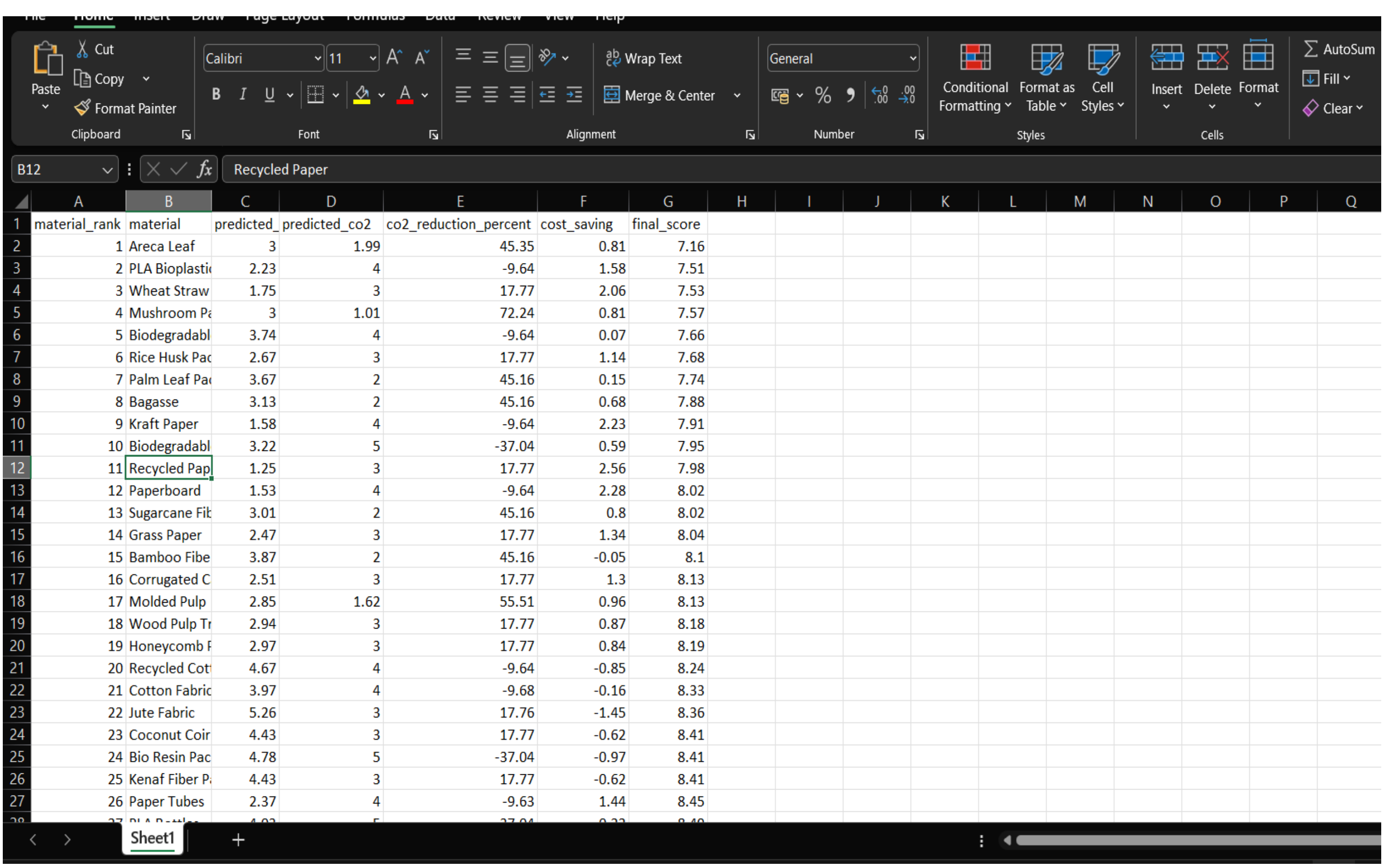
Final Score: 7.66



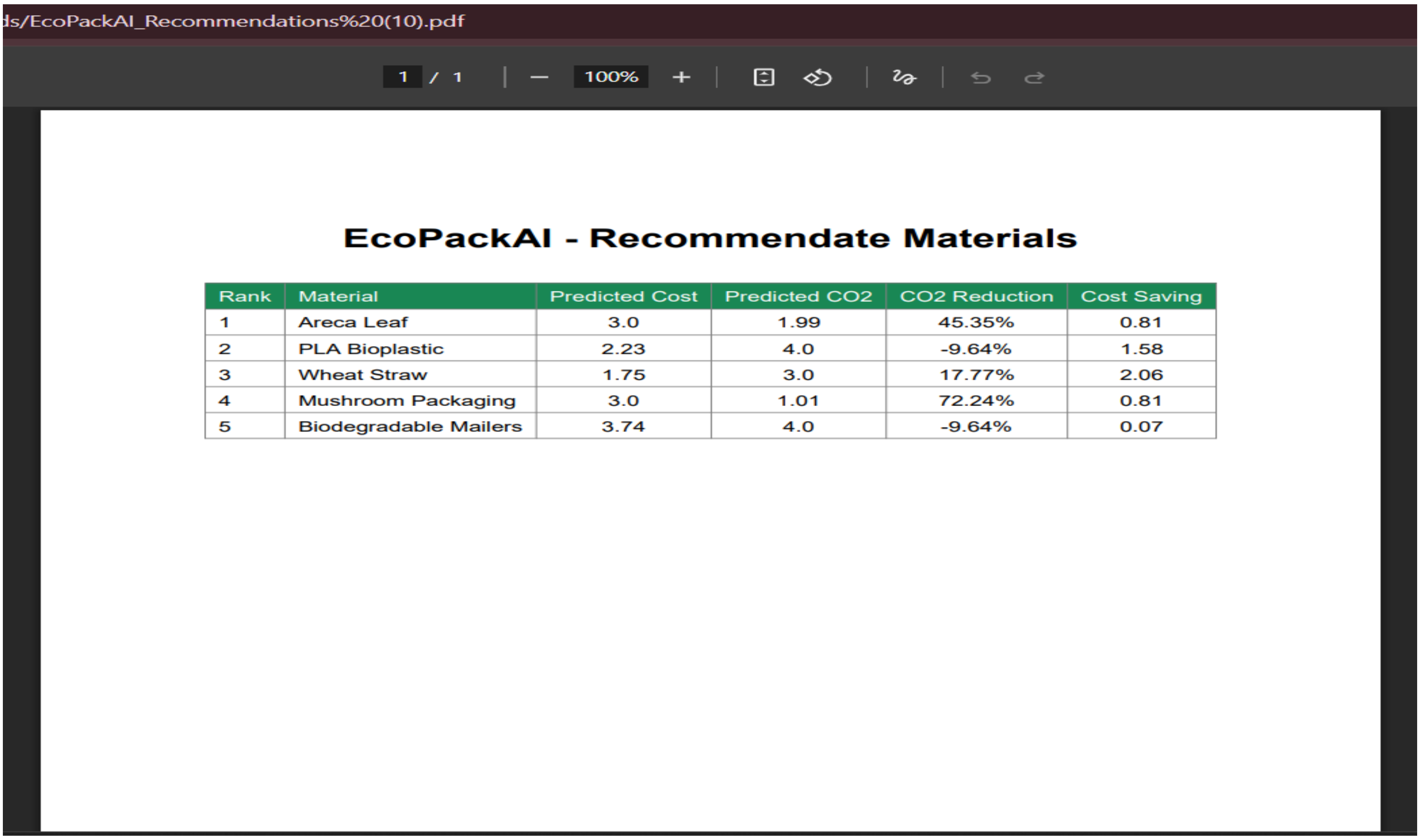


Materials Usage Pie -Chart & Exports Excels + PDF



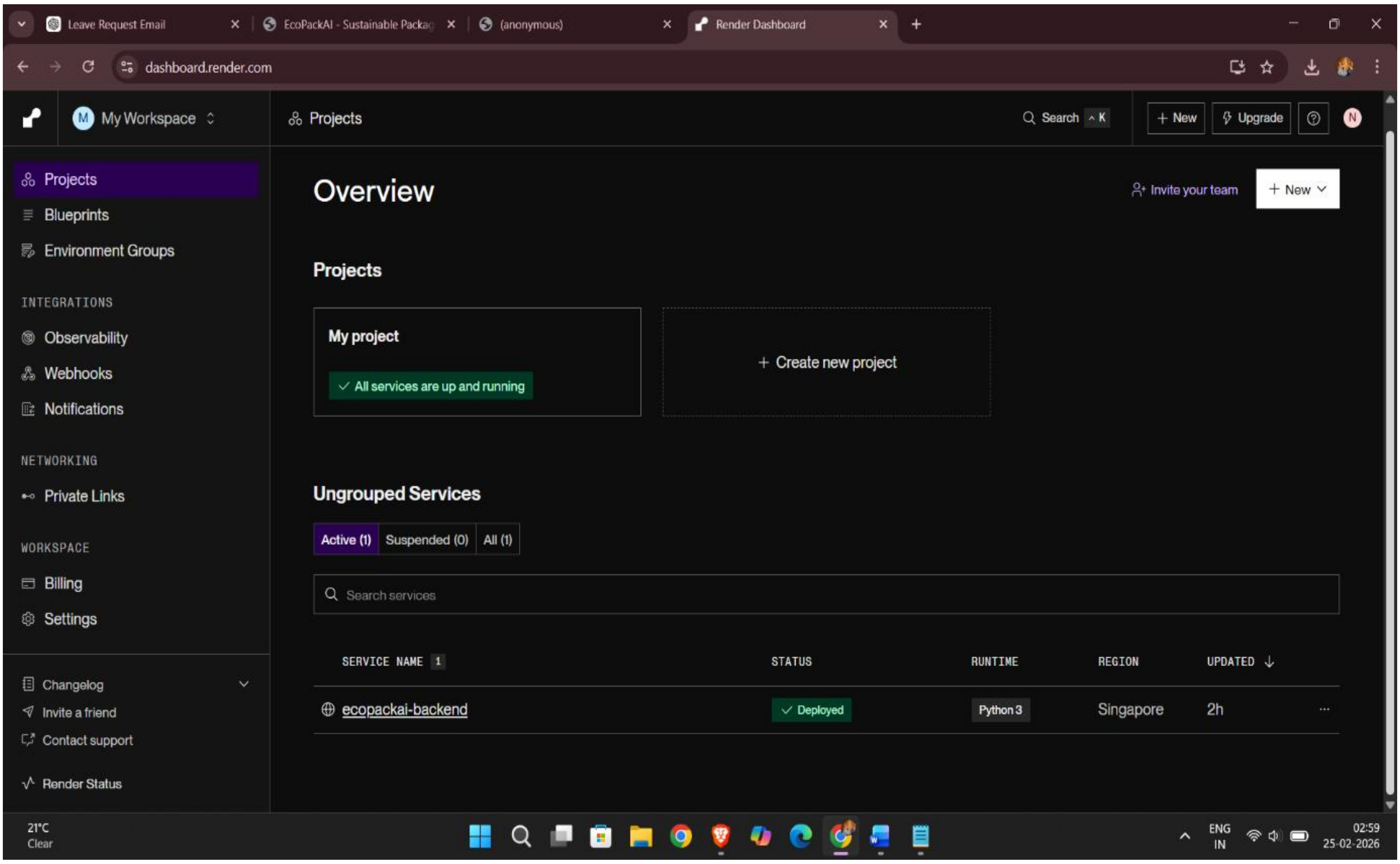


material_rank	material	predicted_	predicted_co2	co2_reduction_percent	cost_saving	final_score
1	Areca Leaf	3	1.99	45.35	0.81	7.16
2	PLA Bioplastic	2.23	4	-9.64	1.58	7.51
3	Wheat Straw	1.75	3	17.77	2.06	7.53
4	Mushroom Pa	3	1.01	72.24	0.81	7.57
5	Biodegradabl	3.74	4	-9.64	0.07	7.66
6	Rice Husk Pac	2.67	3	17.77	1.14	7.68
7	Palm Leaf Pac	3.67	2	45.16	0.15	7.74
8	Bagasse	3.13	2	45.16	0.68	7.88
9	Kraft Paper	1.58	4	-9.64	2.23	7.91
10	Biodegradabl	3.22	5	-37.04	0.59	7.95
11	Recycled Pap	1.25	3	17.77	2.56	7.98
12	Paperboard	1.53	4	-9.64	2.28	8.02
13	Sugarcane Fib	3.01	2	45.16	0.8	8.02
14	Grass Paper	2.47	3	17.77	1.34	8.04
15	Bamboo Fibe	3.87	2	45.16	-0.05	8.1
16	Corrugated C	2.51	3	17.77	1.3	8.13
17	Molded Pulp	2.85	1.62	55.51	0.96	8.13
18	Wood Pulp Tr	2.94	3	17.77	0.87	8.18
19	Honeycomb F	2.97	3	17.77	0.84	8.19
20	Recycled Cot	4.67	4	-9.64	-0.85	8.24
21	Cotton Fabric	3.97	4	-9.68	-0.16	8.33
22	Jute Fabric	5.26	3	17.76	-1.45	8.36
23	Coconut Coir	4.43	3	17.77	-0.62	8.41
24	Bio Resin Pac	4.78	5	-37.04	-0.97	8.41
25	Kenaf Fiber P	4.43	3	17.77	-0.62	8.41
26	Paper Tubes	2.37	4	-9.63	1.44	8.45



EcoPackAI - Recommendate Materials

Rank	Material	Predicted Cost	Predicted CO2	CO2 Reduction	Cost Saving
1	Areca Leaf	3.0	1.99	45.35%	0.81
2	PLA Bioplastic	2.23	4.0	-9.64%	1.58
3	Wheat Straw	1.75	3.0	17.77%	2.06
4	Mushroom Packaging	3.0	1.01	72.24%	0.81
5	Biodegradable Mailers	3.74	4.0	-9.64%	0.07



Deployment on Render

16. Repository & Deployment Details

GitHub Repository:

https://github.com/GPA-NileshGawhale/EcoPack-AI

Live Deployment (Render):

https://ecopackai-backend-6rv7.onrender.com

This project is deployed in a production environment using Flask, Gunicorn, PostgreSQL Cloud, and Render Web Services.