Problem Statement

🧾 Document 1: Problem Statement — CarbonKanakku

Project Title:

CarbonKanakku – Making Sustainability Measurable for the Textile Industry

---

1. Background

The textile industry is one of the world's largest contributors to environmental pollution. Small and medium-scale textile manufacturers often lack access to affordable tools for tracking and reducing their carbon footprint. With growing government regulations and ESG (Environmental, Social, and Governance) compliance requirements, textile businesses struggle to measure, monitor, and report their sustainability metrics.

---

2. Problem Definition

There is no simple, automated system for small textile industries to:

- Calculate their carbon emissions,
- Track water and waste usage,
- Understand their ESG compliance level, and
- Generate sustainability reports for brand reputation and legal compliance.

This leads to:

- Inaccurate or missing data,
- Higher audit and penalty risks,
- Missed business opportunities with sustainable brands.

---

3. Problem Scope

The problem affects:

- **Small & medium textile manufacturers** who lack sustainability tracking systems.
- **ESG consultants and auditors** who rely on manual data collection.
- **Brands** aiming for sustainable sourcing but unable to verify supplier compliance.

---

4. Objective

To develop a **web-based application** that calculates and tracks carbon emissions, water usage, and waste generation — providing real-time ESG dashboards and automated sustainability reports.

---

5. Impact of the Problem

Without a proper tracking system:

- Textile companies lose ESG credibility.
- Non-compliance fines increase.
- Global buyers avoid non-sustainable suppliers.

Business Requirements Document (BRD)

🗒️ Document 2: Business Requirements Document (BRD)

Project Name: CarbonKanakku
Prepared By: Business Analyst – *B.R. Vishwa*

---

1. Purpose

The purpose of CarbonKanakku is to help textile industries **track, measure, and reduce** their environmental impact.
It provides ESG (Environmental, Social, Governance) dashboards, carbon emission reports, and sustainability insights to support **data-driven decision-making** and **regulatory compliance**.

---

2. Business Objectives

- Automate the process of calculating **carbon emissions**, **water consumption**, and **waste generation**.
- Provide real-time **ESG dashboards** with visual analytics.
- Enable **auto-generated sustainability reports** for compliance and branding.
- Help companies **compare performance over time** and improve sustainability scores.

---

3. Scope

In-Scope:

- Web-based app for textile industries
- Data input for production, energy, and logistics
- Carbon emission & ESG calculations using standard formulas
- Auto-generated PDF sustainability reports

- Admin and company user dashboards
- ML-based sustainability recommendations

Out-of-Scope (for MVP):

- Mobile app version
- Blockchain traceability
- Integration with IoT sensors (future phase)

---

4. Stakeholders

Role	Responsibility
Business Analyst	Requirements gathering, documentation, validation
Product Manager	Feature planning, sprint coordination
Developers	Frontend, backend, and API implementation
Data Scientist	ML model integration
End Users (Factories)	Input production data, view ESG dashboard
Admin (Organization)	Monitor platform activity, verify data

---

5. Functional Requirements

1. **User Registration & Login**
    - Company registration (factory name, type, size, location, email).
    - Secure login with authentication.
2. **Data Input Module**
    - Input for energy usage, water consumption, raw materials, logistics.

    - Automatic emission factor selection based on factory type.

3. **Dashboard Module**
    - Display CO_2, H_2O, and waste metrics in charts.
    - Show compliance progress and comparison to targets.
4. **Report Generation**
    - Downloadable PDF report with ESG summary.
5. **Recommendation Engine**
    - Suggest sustainability improvements using ML models.

---

6. Non-Functional Requirements

- **Performance:** Real-time calculations under 3 seconds.
- **Security:** Data encryption & secure login.
- **Scalability:** Should support 100+ factories.
- **Usability:** Simple UI for non-technical users.

---

7. Success Metrics

- ≥ 90% accuracy in emission calculation.
- ≥ 80% of users generate reports monthly.
- ≥ 30% improvement in ESG compliance score over 3 months.

Functional Requirements Specification (FRS)

📘 Document 3: Functional Requirements Specification (FRS)

Project Name: CarbonKanakku
Prepared By: Business Analyst – *B.R. Vishwa*

---

1. Purpose

This document defines the **functional features, workflows, and data inputs/outputs** of CarbonKanakku — a sustainability tracking web app designed for textile industries. It ensures that every module is developed according to user and business needs.

---

2. System Overview

CarbonKanakku enables textile factories to:

- Record energy, water, and material usage.
- Automatically calculate **carbon, water, and waste emissions**.
- View **ESG compliance dashboards**.
- Generate **auto ESG reports**.
- Receive **AI-based sustainability recommendations**.

---

3. System Users

User Type	Description	Key Actions
Factory User	Textile industry representative	Register, log in, input data, view dashboards, generate reports
Admin	Platform manager	Approve new factories, manage data, verify emissions, handle user access

---

4. Functional Modules

🔹 **Module 1: Authentication & Onboarding**

Description: Handles secure login, registration, and user profiles.

- **Inputs:** Factory name, type (spinning, dyeing, garmenting, etc.), size, address, email, password.
- **Process:** Validate details → store in DB → send confirmation mail.
- **Outputs:** User session created → redirected to dashboard.

Error Handling: Invalid email, existing account, weak password.

🔹 **Module 2: Data Input Module**

Description: Factories input their operational data for analysis.

- **Inputs:**
    - Energy usage (kWh, diesel, coal, etc.)
    - Water usage (liters/day)
    - Raw materials (cotton, polyester, chemicals, dyes)
    - Waste generated (kg/day)
    - Transport (distance & vehicle type)
- **Process:**
    - Validate units
    - Apply emission factor formulas
    - Store historical data
- **Outputs:**

    - Total emission (CO_2e)
    - Water footprint
    - Waste score

🔹 **Module 3: Dashboard Module**

Description: Displays key metrics, charts, and compliance levels.

- **Inputs:** Emission data from DB.
- **Process:** Summarize metrics by week/month → generate graphs.
- **Outputs:**
    - CO_2, Water, Waste charts
    - Compliance level progress bar
    - Comparison vs industry average

🔹 **Module 4: ESG Report Generator**

Description: Creates auto sustainability reports for branding and compliance.

- **Inputs:** Factory data + calculated emission + compliance data.
- **Process:**
    - Format report using templates
    - Add recommendations & summary charts
    - Export as PDF
- **Outputs:** Downloadable “CarbonKanakku Report.pdf”

🔹 **Module 5: Recommendation Engine (AI/ML)**

Description: Suggests actions to improve ESG score.

- **Inputs:** Factory metrics (energy, water, waste).
- **Process:** Compare with trained dataset → identify improvement areas.
- **Outputs:**
    - Action suggestions (e.g., switch to solar, reuse water)
    - Estimated % improvement in emissions

🔹 **Module 6: Admin Panel**

Description: Enables admins to manage factories and verify reports.

- **Inputs:** Factory list, reports, activity logs.
- **Process:** Approve/reject registration, validate emission records.
- **Outputs:** Updated database, verified user list.

---

5. System Flow (Simplified)

1. User registers → email verification.
2. Logs in → fills data in input forms.
3. System calculates emissions automatically.
4. Dashboard updates with new values.
5. User generates ESG report.
6. ML model gives recommendations.

7. Admin verifies and approves reports.

---

6. Data Flow Example

Input	Process	Output
Electricity usage: 1000 kWh	× emission factor (0.82 kg CO_2/kWh)	820 kg CO_2 emitted
Water: 5000 L/day	× reuse ratio	Water sustainability score
Waste: 30 kg/day	Categorize recyclable/non-recyclable	Waste management score

---

7. Assumptions

- User inputs are accurate.
- Factories have consistent monthly data.
- Standard emission factors are used (per UNFCCC/ISO).

---

8. Constraints

- Limited internet connectivity at factories.
- Variation in data formats (manual vs IoT readings).
- Initial version supports only textile domain.

✅ Outcome

By completing these modules, the system enables **end-to-end carbon tracking** and **automated ESG compliance** for small to mid-level textile units.

Software Requirement Specification (SRS)

📘 Document 4: Software Requirements Specification (SRS)

Project Name: CarbonKanakku
Prepared by: Business Analyst – *B.R. Vishwa*
Reviewed by: Product Manager – *Vishwanathan*

---

1. Purpose

The purpose of this document is to define the **complete technical and functional requirements** for the *CarbonKanakku* web app — a sustainability and carbon footprint tracking system for the textile industry.
It acts as a bridge between the business problem and the developer's implementation plan.

---

2. System Overview

CarbonKanakku is a full-stack web app that helps textile industries calculate, monitor, and reduce their carbon emissions.
It integrates data collection, analysis, visualization, and reporting features into one unified dashboard.

---

3. Scope

The system will:

- Record operational data such as energy, water, and material usage.
- Calculate **CO_2, water, and waste emissions**.
- Display performance dashboards.
- Generate ESG reports automatically.
- Provide **AI-based sustainability recommendations**.
- Allow admin verification and monitoring.

---

4. System Users

Role	Description	Responsibilities
Factory User	Industry representative	Input data, view dashboard, generate reports
Admin	System manager	Validate data, manage factories, approve reports
Developer	System maintainer	Implement APIs, maintain backend
Business Analyst	Requirement manager	Ensure user needs match output features

---

5. Functional Requirements

🔹 **Module 1 – Authentication**

Functions:

- User registration
- Email verification
- Login / Logout
- Password recovery

APIs:

- `POST /api/register`
- `POST /api/login`
- `POST /api/verify-email`
- `POST /api/reset-password`

Dummy Data Example:

```
{
  "email": "textile@factory.com",
  "password": "Carbon@2025",
  "factoryName": "GreenTextile Pvt Ltd",
  "factoryType": "Dyeing Unit"
}
```

🔹 **Module 2 – Data Input & Emission Calculation**

Functions:

- Input energy, water, waste, material, and transport data.
- Apply emission factor formulas.
- Save data historically (daily, weekly, monthly).

APIs:

- `POST /api/data-input`
- `GET /api/emission-calc`
- `GET /api/factory-data/:id`

Calculation Formula:

CO_2 Emission=Activity Data×Emission Factor\text{CO_2 Emission} = \text{Activity Data} \times \text{Emission Factor}CO_2 Emission=Activity Data×Emission Factor

Example:
Electricity (1000 kWh × 0.82 kg CO_2/kWh) = 820 kg CO_2

Dummy Input Example:

```
{
  "factoryId": "F101",
  "energy": {"electricity": 1000, "diesel": 30},
  "water": 5000,
```

```
  "materials": {"cotton": 200, "polyester": 50},
  "waste": {"solid": 40, "liquid": 20}
}
```

🔹 **Module 3 – Dashboard**

Functions:

- Display CO_2, water, and waste in visual charts.
- Show compliance progress.
- Compare with industry average.

APIs:

- `GET /api/dashboard/:factoryId`
- `GET /api/compare-industry`

Dummy Data Example:

```
{
  "totalEmission": 820,
  "waterUse": 5000,
  "wasteGenerated": 40,
  "complianceLevel": "75%"
}
```

🔹 **Module 4 – Report Generator**

Functions:

- Generate ESG report based on input data.
- Export report as PDF or downloadable file.

APIs:

- `POST /api/generate-report`
- `GET /api/report/:id`

Dummy Output Example:

```
{
  "reportId": "R101",
  "factoryId": "F101",
  "totalEmission": "2.3 tons/month",
  "waterFootprint": "12 KL",
  "wasteOutput": "40 kg",
  "complianceScore": "78%",
  "recommendations": ["Install solar panels", "Reduce dye chemical
usage"]
}
```

🔹 **Module 5 – AI Recommendation System**

Functions:

- Analyze factory data.
- Provide actionable suggestions to reduce emissions.

APIs:

- `GET /api/recommendations/:factoryId`
- `POST /api/retrain-model`

Dummy Output:

```
{
  "suggestions": [
    "Switch to renewable power sources (saves 18% CO₂)",
```

```
    "Recycle 30% wastewater to cut 25% water footprint"
  ]
}
```

🔹 **Module 6 – Admin Panel**

Functions:

- Manage factories.
- Approve or reject emission data.
- View reports and activity logs.

APIs:

- `GET /api/admin/factories`
- `PUT /api/admin/verify/:factoryId`
- `DELETE /api/admin/remove/:factoryId`

---

6. Non-Functional Requirements

Category	Requirement
Performance	Handle 100+ simultaneous users
Scalability	Support multiple industries in future
Security	JWT Authentication + hashed passwords
Availability	99% uptime target
Usability	Clean and minimal UI (for factory operators)
Maintainability	Modular architecture for easy updates

---

7. System Architecture

Frontend: React.js / Next.js
Backend: Node.js + Express
Database: Supabase / PostgreSQL
AI Model: Python (Flask/FastAPI service)
Hosting: Vercel (Frontend) + Render (Backend)

---

8. Data Flow (Simplified)

User → Input Data → Backend → Emission Calculator → Dashboard → Report → Recommendations

---

9. Expected Output

✅ Clean sustainability dashboard
✅ Auto-calculated carbon emission metrics
✅ AI-powered recommendations
✅ ESG report PDF
✅ Admin verification portal

---

10. Acceptance Criteria

- User can register and log in successfully.
- Factory data is stored and emission calculated correctly.
- Dashboard loads with accurate charts.
- PDF report generation works without error.
- Admin can verify or reject factory reports.

Test Plan & Test Case

🧪 Document 5: Test Plan & Test Case Document

Project Name: CarbonKanakku
Prepared by: Haris (Tester)
Reviewed by: B.R. Vishwa (Business Analyst)
Sprint: 4

---

1. Objective

The purpose of this test plan is to ensure that all functionalities of *CarbonKanakku* — from user registration to ESG report generation — perform as expected, are bug-free, and meet the business and functional requirements defined in the SRS.

---

2. Testing Scope

In Scope:

- Authentication (Login, Register)
- Data Input & Emission Calculation
- Dashboard Visualizations
- Report Generation
- AI Recommendations
- Admin Verification

Out of Scope:

- Mobile responsiveness
- Integration with external IoT devices

---

3. Testing Types

Type	Description	Tools / Approach
Unit Testing	Testing individual API endpoints and functions.	Postman / Jest
Integration Testing	Verifying data flow between modules (input → calculation → report).	Postman
System Testing	Checking the whole web app flow as a single unit.	Browser Testing
User Acceptance Testing (UAT)	Final validation based on end-user perspective.	Manual Testing
Regression Testing	Retesting after code updates or bug fixes.	Postman / Manual
Performance Testing	Testing app speed and load time.	Chrome Lighthouse

---

4. Test Environment

Component	Configuration
Frontend	React (Vercel deployment)
Backend	Node.js (Render deployment)
Database	Supabase (PostgreSQL)
Browsers Tested	Chrome, Edge

---

5. Test Scenarios and Test Cases

Test ID	Module	Test Case Description	Input	Expected Output	Status
TC-01	Authentication	Verify registration form works properly	Factory Name, Email, Password	User registered successfully	✅ Pass

TC-02	Authentication	Login with valid credentials	Email, Password	Redirect to dashboard	✅ Pass
TC-03	Authentication	Login with invalid password	Wrong Password	Display error message	✅ Pass
TC-04	Data Input	Add energy and water data	1000 kWh, 5000L	Data saved successfully	✅ Pass
TC-05	Emission Calculation	Verify CO_2 formula correctness	Electricity = 1000 kWh	Correct value = 820 kg CO_2	✅ Pass
TC-06	Dashboard	Check dashboard loads correctly	Factory ID = F101	Display charts and KPIs	⚠️ Partial (slow load)
TC-07	Report Generator	Generate PDF report	Factory data available	Downloadable ESG report	✅ Pass
TC-08	Recommendations	AI generates valid suggestions	Energy = 1000, Water = 5000	2 sustainability tips	✅ Pass
TC-09	Admin	Verify factory data approval	Factory ID = F101	Status = Approved	✅ Pass
TC-10	Security	Test unauthorized access	No login token	Redirect to login	✅ Pass

---

6. Bugs & Issues Identified

Bug ID	Module	Description	Severity	Status
B-01	Dashboard	Chart load delay after data input	Medium	Fixed
B-02	Report Generator	Alignment issue in PDF	Low	Fixed
B-03	Auth	Password reset email not triggering	High	Pending

---

7. Test Summary

✅ **Total Test Cases:** 10
✅ **Passed:** 9
⚠️ **Pending:** 1
🚫 **Failed:** 0

Overall Test Result: *Successful with minor UI and performance issues.*

---

8. Tester’s Remarks

The overall system is stable.
Core functions like data input, calculation, report generation, and AI suggestions are working fine.
Performance optimization needed in the dashboard module for faster chart rendering.

---

9. Business Analyst Review (by B.R. Vishwa)

The system fulfills the business requirements for ESG tracking and emission management.
The issues found do not impact core calculations or compliance reports, hence the MVP is ready for demonstration and future scale-up.

Project Summary & Future Scope

Project Title: CarbonKanakku
Domain: ESG + Sustainability Tracking for Textile Industry
Team: Pixel Piratech
Prepared by: B.R. Vishwa (Business Analyst)

---

1. Project Overview

CarbonKanakku is a smart sustainability tracking platform designed for **textile industries** to monitor, calculate, and reduce their **carbon footprint**, **water consumption**, and **waste generation**.

The tool helps factories transition toward **ESG compliance** by automating data collection, generating **real-time dashboards**, and providing **AI-powered recommendations** to improve sustainability performance.

In simple terms — it makes sustainability *measurable, actionable, and visible*.

---

2. Problem We Solved

Textile industries contribute to nearly **10% of global carbon emissions**, yet most small and medium factories:

- Lack tools to calculate or track emissions.
- Struggle to meet **ESG compliance requirements** like BRSR and CSRD.
- Lose clients due to the absence of verified sustainability data.

Hence, **CarbonKanakku** empowers them with a low-cost digital solution to measure and reduce their footprint.

---

3. Our Proposed Solution

We developed a web application that allows textile units to:
✅ Input operational data (energy, water, waste, materials).
✅ Auto-calculate emissions using verified formulas.
✅ View ESG performance on a clean dashboard.

✅ Generate downloadable sustainability reports.
✅ Get AI-driven recommendations to reduce emissions.

This helps industries build **brand trust**, **regulatory compliance**, and **environmental responsibility** simultaneously.

---

4. Key Features

Feature	Description
🧾 **Emission Calculator**	Calculates CO_2, water, and waste emissions from input data.
📊 **Sustainability Dashboard**	Displays key metrics visually for better decision-making.
🤖 **AI Recommendations**	Suggests ways to reduce emissions and improve sustainability.
📑 **Automated ESG Report**	Generates a verified compliance report for submission.
🔐 **Admin Panel**	Allows verification, factory management, and report approvals.

---

5. Impact

Stakeholder	Benefit
Factory Owners	Get real-time insight into their environmental impact.
Auditors / Consultants	Access verified ESG data for compliance checks.
Government & Brands	Validate sustainable sourcing easily.
Environment	Promotes measurable carbon reduction.

---

6. Team Contributions

Member	Role	Key Work
Vishwanathan	Product Manager	Defined project roadmap, sprint plans, and coordination

B.R. Vishwa	Business Analyst	Collected requirements, designed architecture, and managed documentation
Rohit	Developer	Implemented frontend and API integration
Haris	Tester	Performed test plan execution and bug tracking
Dilip Kumar	Documentation	Prepared reports, posters, and project presentation

---

7. Sprint Overview

Sprint	Objective	Key Deliverable
Sprint 1	Problem Understanding & Industry Selection	Textile Industry Focus
Sprint 2	Backlogs & Architecture	Module and API Planning
Sprint 3	UI/UX Design & Prototyping	Wireframes + Frontend
Sprint 4	Testing & Integration	Working Prototype + Report

---

8. Limitations

- Currently supports only textile manufacturing industries.
- Data input is manual (no IoT integration yet).
- Limited to web interface (mobile version in progress).

---

9. Future Scope

💡 **Next Steps for CarbonKanakku**

1. **Industry Expansion** – Extend system to automotive, construction, and agriculture sectors.
2. **IoT Integration** – Auto-fetch data from sensors for real-time monitoring.
3. **AI Forecasting** – Predict future emissions based on production trends.

4. **Blockchain Verification** – Secure and trace sustainability reports.

5. **B2B Marketplace** – Connect sustainable factories with green buyers.

---

10. Conclusion

CarbonKanakku is not just a project — it's a step toward a **sustainable textile ecosystem**.
By empowering small factories with ESG intelligence, it bridges the gap between environmental responsibility and business growth.

> "Our mission is simple — to make every thread of production count towards a greener planet." 🌱

Process Flow Document

🧾6️⃣ Process Flow Document

🎯 Purpose

The Process Flow Document visually and descriptively explains **how your CarbonKanakku system works step-by-step** — from when a user logs in to when reports are generated.
It helps developers, testers, and stakeholders understand **the journey of data** and **user actions** within the app.

⚙️ Overview

The process flow represents how a textile business user interacts with the system and how the system responds.
In **CarbonKanakku**, there are mainly **three user roles**:

- 👤 **Factory Owner / Business User**
- 🧑‍💼 **Admin (ESG Officer)**
- 🧠 **ML Engine (Backend Service)**

🧩 Step-by-Step Flow (Textual Description)

1️⃣ User Registration & Login

- The user (factory owner) visits the CarbonKanakku web app.
- Creates an account by entering details like factory name, type, location, contact info, etc.
- Credentials are verified via backend (Supabase Auth).
- Upon success → user is redirected to the dashboard.

Output: Authenticated session with personalized dashboard.

2️⃣ Data Input Section

- The user enters operational details:
    - Factory Type (e.g., Cotton Dyeing, Garment, Spinning Unit)
    - Energy Source (Electricity, Diesel, Coal)
    - Water Usage (Litres per day)
    - Waste Generated (Kg per day)
    - Production Quantity (Meters/Tons)
- The system validates these inputs.

Output: Validated data stored in the database.

3️⃣ ML Calculation (Emission Engine)

- The backend sends data to the ML model.
- The model calculates:
    - 🌫️ Carbon Emission (CO_2e)
    - 💧 Water Footprint
    - 🗑️ Waste Footprint
- Results are returned to the backend and stored in the database.

Output: Emission data ready for dashboard visualization.

4️⃣ ESG Dashboard Generation

- The dashboard visualizes data through:
    - Bar charts for CO_2 emission per unit type
    - Pie charts for water and waste consumption

    - ESG score indicators (A, B, C grade)
- The user can filter data by time or process.

Output: Interactive sustainability dashboard.

5️⃣ Report Generation

- The user can click **“Generate Report”** to create an ESG summary.
- The report includes:
    - Company info
    - Monthly or yearly emission summary
    - Recommendations to reduce emissions
- The report is downloadable as a PDF.

Output: ESG Report PDF.

6️⃣ Logout or Session End

- The user can log out manually or session auto-expires.
- All unsaved data is automatically stored securely.

Output: User safely exits the system.

🧠 Process Flow (Diagram Description)

You can visualize this using BPMN or a simple flowchart tool like Lucidchart or Draw.io.

```
[Start]
   ↓
[User Login/Registration]
```

```
   ↓
[Data Input Page]
   ↓
[Emission Calculation Engine (ML)]
   ↓
[Display Dashboard]
   ↓
[Generate ESG Report]
   ↓
[Logout/End]
```

🧩 Key Benefits of Process Flow Document

- Helps developers understand system logic easily.
- Allows testers to derive scenarios from each flow step.
- Acts as a visual reference for future enhancement discussions.
- Improves communication between technical and non-technical stakeholders.

Data Flow Diagram (DFD)

🧾7️⃣Data Flow Diagram (DFD)

🎯 Purpose

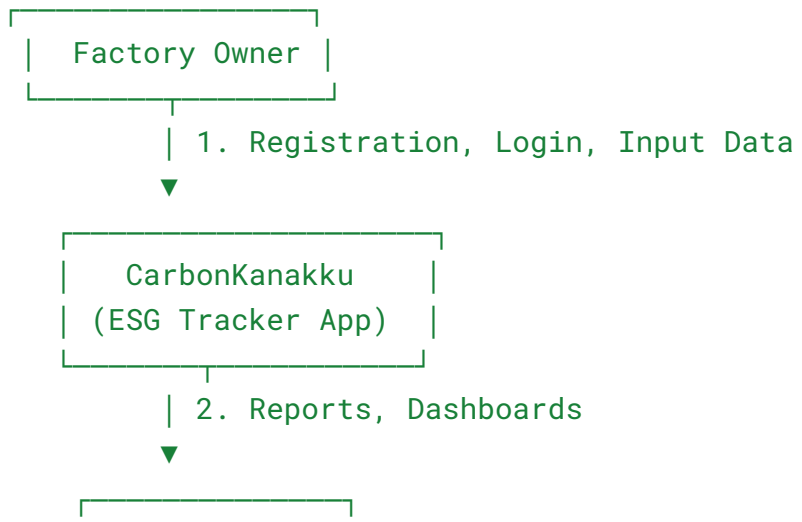
A **Data Flow Diagram (DFD)** shows **how data moves** from input to output within the CarbonKanakku system.
It represents **processes, data stores, external entities, and data flows** clearly — helping both developers and non-technical members understand **what happens under the hood.**

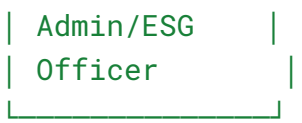
🧩 Key Components Used in the DFD

Symbol	Component	Meaning
🧍	**External Entity**	Person or system that interacts with CarbonKanakku (e.g., Factory Owner, Admin)
🔄	**Process**	Activity or function that transforms data (e.g., Calculate Emissions)
🗂️	**Data Store**	Database or file where information is stored (e.g., Supabase DB)
➡️	**Data Flow**	The movement of data between components

🔹 Level 0 DFD (Context Diagram)

This is the **simplest view**, showing CarbonKanakku as a single system.



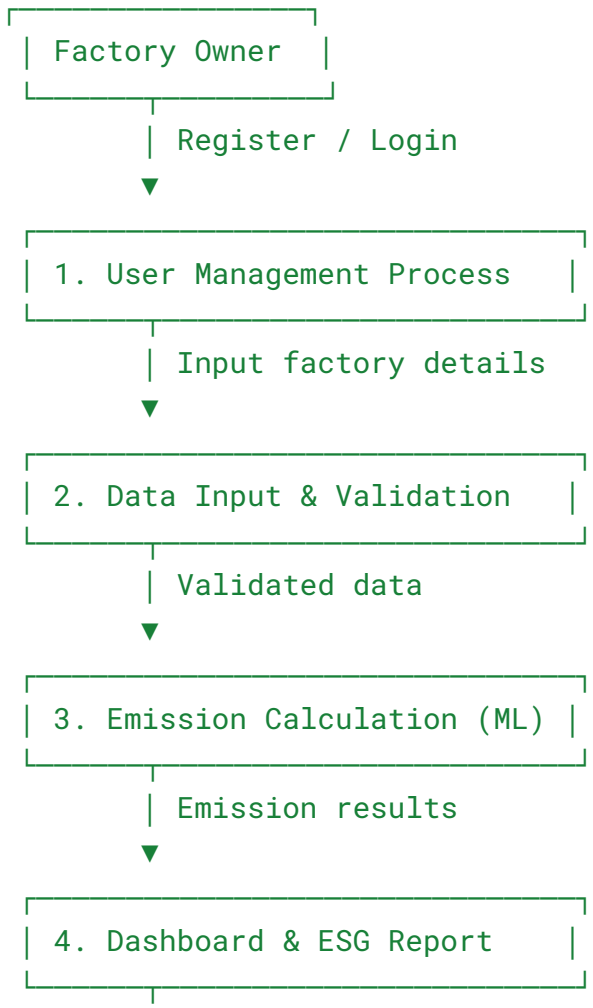


📘 **Explanation:**

- Factory Owner inputs data → system processes it → sends results (dashboard + report).
- Admin can view reports, compliance, and user details.

🔹 Level 1 DFD – Expanded System View

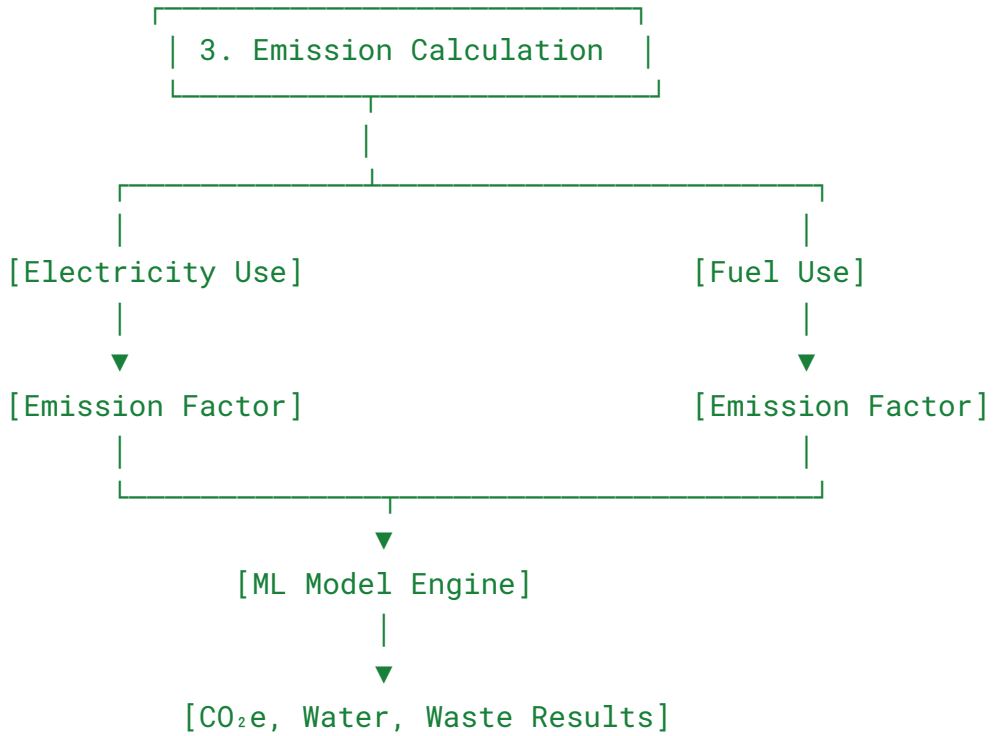
This expands the CarbonKanakku system into major processes.



📘 **Explanation:**

1. The **User Management** process handles registration/login.
2. The **Data Input & Validation** process ensures accuracy of factory info.
3. The **Emission Calculation** process uses ML to calculate carbon, water, and waste footprints.
4. The **Dashboard & Report Generation** process displays results and provides reports.

🔹 Level 2 DFD – Detailed Breakdown (Example: Emission Calculation)



📘 **Explanation:**

- Each input (electricity, fuel, water) is multiplied by a standard **emission factor**.
- The ML model refines accuracy using historical data.
- Results are stored and displayed on dashboards.

🗂️ Data Stores Used

Data Store	Description
🧾 **User Database**	Stores user credentials, factory info
🧮 **Emission Data**	Stores calculated CO_2e, water, waste data
📊 **Report Data**	Contains generated ESG reports
🧠 **ML Model Store**	Stores model weights and parameters for emission estimation

---

📈 Data Flow Summary

From	To	Data Transferred
User	System	Registration details, factory data
System	ML Engine	Input parameters for emission
ML Engine	System	CO_2, Water, Waste values
System	Dashboard	Processed metrics, ESG score
System	Report Generator	Formatted ESG report (PDF)

💡 Why DFD is Important

- Helps visualize data transformation clearly.

- Simplifies communication between the **business analyst**, **developer**, and **tester**.
- Detects redundant or missing data processes early.
- Useful reference for future enhancements or audits.

System Architecture Diagram (SAD)

🧾8️⃣System Architecture Diagram (SAD)

Project: CarbonKanakku
Prepared by: B.R. Vishwa (Business Analyst)
Sprint Phase: Post-Sprint 2 — Architecture Design

🎯 Purpose

The **System Architecture Diagram (SAD)** visually explains how all parts of **CarbonKanakku** — frontend, backend, database, and ML model — interact with each other.
It helps the **developers, testers, and product manager** clearly understand the structure, communication flow, and integration points.

🧠 High-Level Overview

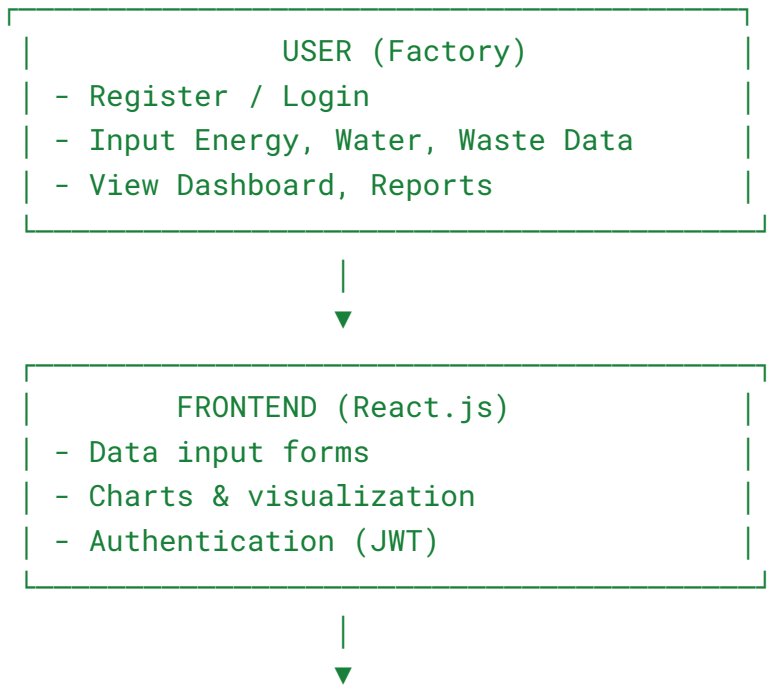
CarbonKanakku is a **web-based ESG + Carbon Footprint tracking platform** designed for the **textile industry** to measure emissions, waste, and water consumption while generating **ESG compliance dashboards and recommendations.**

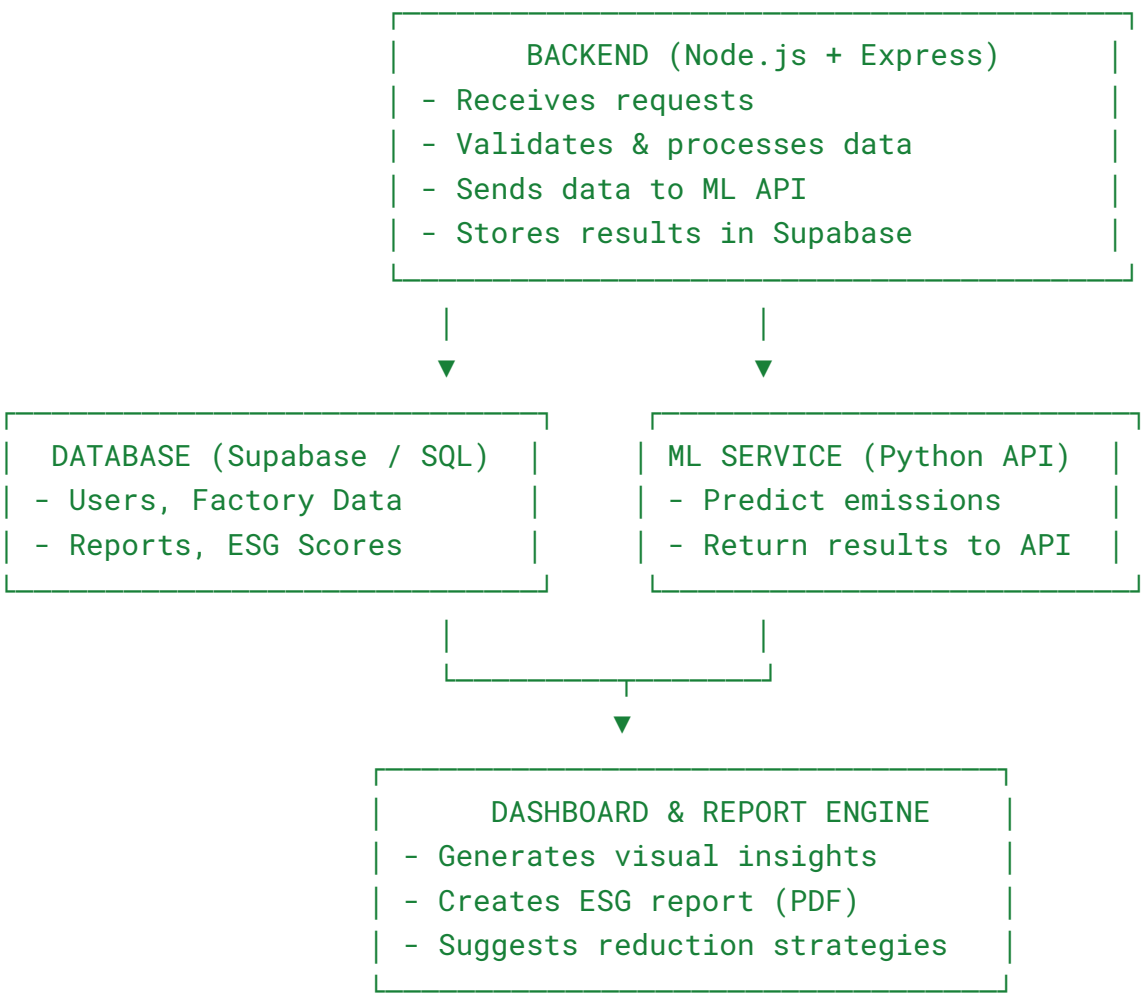
🧩 Main Components

1. 🌐 **Frontend (React.js)**
    - Built with **React + Tailwind CSS**.
    - Provides user-friendly dashboards, forms, and visualization charts.
    - Handles user registration, login, input data, and viewing results.
2. ⚙️ **Backend (Node.js + Express)**
    - Manages all **API requests** from frontend.
    - Handles **authentication, validation, and report generation**.
    - Connects to both **database** and **ML microservice**.

3. 🧮 **Machine Learning Service (Python + Flask / FastAPI)**
    - Predicts **carbon, water, and waste emissions** using standard emission factors and past data.
    - Returns results to backend via REST API calls.
4. 🗄️ **Database (Supabase / PostgreSQL)**
    - Stores **user data, factory data, emission records, ESG scores**, and **recommendations**.
    - Ensures **secure and scalable** cloud storage.
5. ☁️ **Deployment Layer**
    - Frontend: **Vercel**
    - Backend + ML API: **Render / Railway**
    - Database: **Supabase Cloud**

🧭 System Architecture Diagram (Textual Representation)





🧱 Data Flow Summary

Step	Source	Destination	Description
1️⃣	User	Frontend	User enters data (factory info, inputs).
2️⃣	Frontend	Backend	Data sent via POST API (JSON format).
3️⃣	Backend	ML API	Backend forwards data for analysis.
4️⃣	ML API	Backend	ML model returns emission predictions.
5️⃣	Backend	Database	Results stored securely in Supabase.

6️⃣	Backend	Frontend	Sends dashboard metrics + recommendations.
7️⃣	Frontend	User	Dashboard visualizes results.

🔐 Security Considerations

- JWT-based authentication for secure sessions.
- Role-based access control (Admin / Factory Owner).
- HTTPS communication between all layers.
- Data validation before ML processing.

🚀 Scalability Features

- Microservice-based architecture (backend and ML are independent).
- Cloud-hosted database for performance.
- API-driven integration for future industry expansion.

💡 Benefits of This Architecture

✅ Modular — easy to scale or modify components.
✅ Cloud-ready — supports hackathon deployment (Vercel + Render).
✅ Secure — minimal data risk with JWT + validation.
✅ Future-friendly — can expand to multiple industries (food, steel, chemical).

Use Case Diagram & Use Case Descriptions

🧾9️⃣ Use Case Diagram & Description

Project: CarbonKanakku
Prepared by: B.R. Vishwa (Business Analyst)
Sprint Phase: Sprint 2 — Requirement Analysis

🎯 Purpose

The **Use Case Diagram** helps visualize how **different users** interact with the **CarbonKanakku system** — what actions they perform, what features they access, and how the system responds.

This is a **bridge between user requirements and system functionality**, helping developers and testers understand all key interactions clearly.

👥 Actors in the System

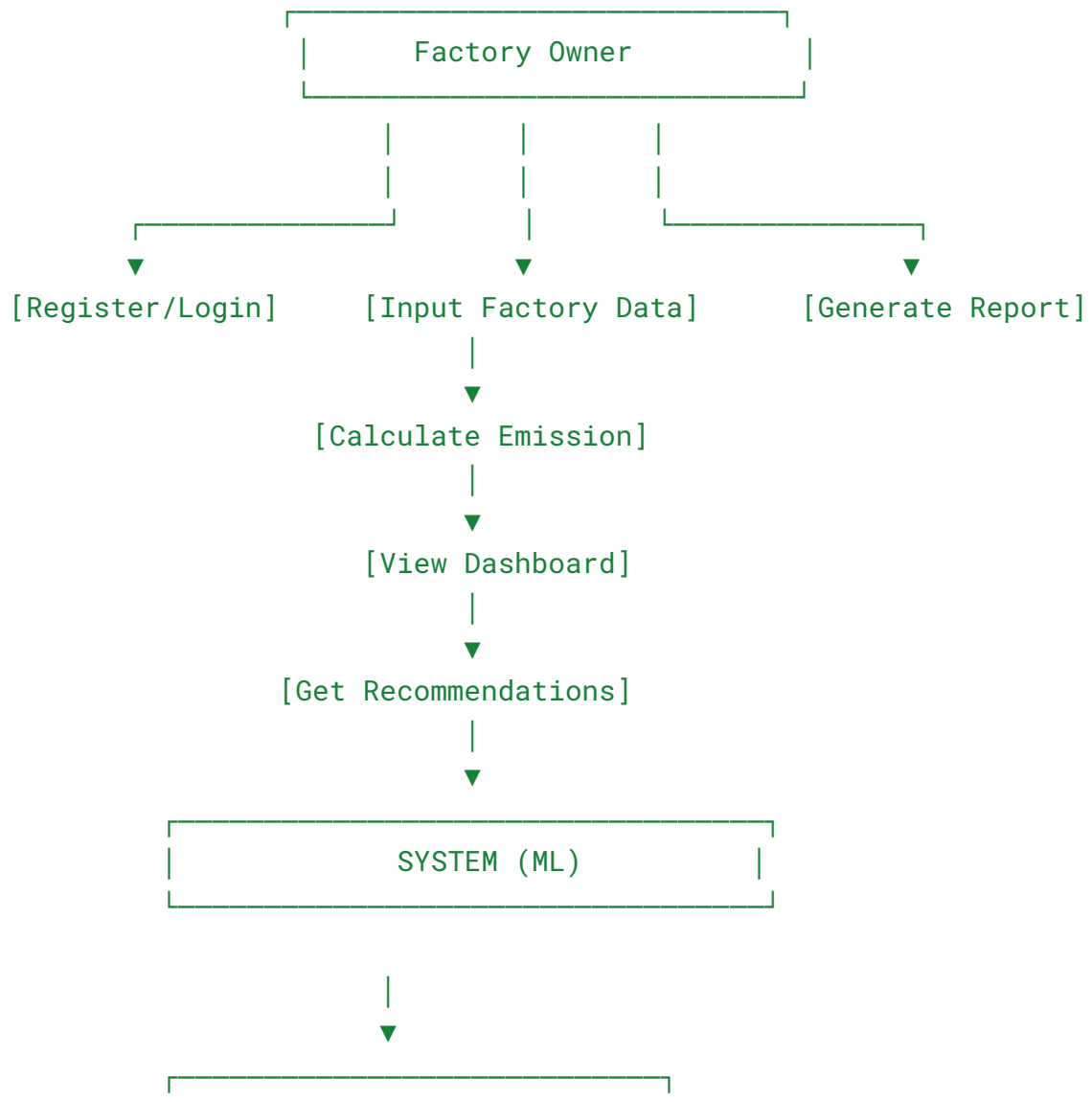
Actor	Description
🧍 **Factory Owner / Business User**	Main user of the system who inputs factory data, checks emission, and generates reports.
🧑‍💼 **Admin (ESG Analyst)**	Reviews overall industry-level data, validates reports, and monitors compliance levels.
🤖 **System / ML Engine**	Processes emission calculations and provides recommendations for reduction.

⚙️ Primary Use Cases

ID	Use Case Name	Description
UC1	Register / Login	User registers and logs into the CarbonKanakku platform securely.
UC2	Input Factory Data	User provides data like energy, waste, and water usage.

UC3	Calculate Emission	System uses ML logic to calculate carbon, water, and waste footprint.
UC4	View Dashboard	Displays real-time metrics and ESG compliance level.
UC5	Generate Report	Creates a downloadable ESG report in PDF format.
UC6	Get Recommendations	ML model gives actionable suggestions to reduce emissions.
UC7	Admin Validation	Admin verifies and manages factory reports and ESG data.

🎯 Use Case Diagram (Text Description)



🧾 Detailed Use Case Descriptions

Use Case UC1: Register / Login

Actor: Factory Owner
Description: User creates an account and logs in securely using Supabase Auth.
Precondition: User must have valid credentials.
Postcondition: User session starts, and dashboard is accessible.

Use Case UC2: Input Factory Data

Actor: Factory Owner
Description: User submits data about energy source, production, waste, and water use.
Precondition: User is logged in.
Postcondition: Data stored in database for analysis.

Use Case UC3: Calculate Emission

Actor: System / ML Engine
Description: System calculates CO_2e emissions based on the input data.
Precondition: Valid data is available.
Postcondition: Emission results saved and sent to dashboard.

Use Case UC4: View Dashboard

Actor: Factory Owner
Description: Displays key metrics — carbon footprint, water and waste usage, ESG compliance score.
Postcondition: User can analyze sustainability performance.

Use Case UC5: Generate Report

Actor: Factory Owner
Description: User clicks “Generate Report” to export a structured ESG compliance report (PDF).
Postcondition: Report generated and available for download.

Use Case UC6: Get Recommendations

Actor: System / ML Engine
Description: Provides AI-based suggestions to reduce energy and resource consumption.
Postcondition: Recommendations shown on dashboard.

Use Case UC7: Admin Validation

Actor: Admin
Description: Admin reviews company reports and validates compliance status.
Postcondition: Verified reports are flagged as “Approved” in the system.

💡 Key Benefits

✅ Clarifies **who does what** in the system.
✅ Helps developers design accurate user flows.
✅ Useful for testers to build **test cases** per use case.
✅ Clearly maps to **sprint goals** and **functional requirements**.

inal Summary & Report Document

🧾 Final Summary & Report Document

Project Name: CarbonKanakku

Role: Business Analyst – B.R. Vishwa

Industry Focus: Textile Industry (with scalability to other industries)

1️⃣ Project Overview

CarbonKanakku is a sustainability intelligence platform designed to help **textile industries** measure, monitor, and manage their **carbon footprint** and **ESG (Environmental, Social, and Governance) metrics**.

It tracks **carbon emissions, water usage, and waste generation** using simple data inputs from factories.
The system then generates **automated ESG reports**, **compliance dashboards**, and **recommendations** to help industries make **eco-friendly and cost-efficient decisions**.

2️⃣ Problem Statement

The **textile industry** is one of the **largest polluters globally**, yet small and medium factories lack affordable tools to measure and report sustainability data.
Currently:

- Calculations are manual or inconsistent
- ESG reporting requires consultants
- No simple platform provides real-time insights

3️⃣ Proposed Solution

CarbonKanakku simplifies sustainability tracking through automation:

- 🧮 Real-time calculation of **carbon emissions**, **waste**, and **water use**
- 📊 Auto-generated **ESG dashboards & compliance reports**
- 💡 Personalized **sustainability recommendations**
- 🌍 Data visualization to measure impact and progress

This solution can also adapt to **other industries** (manufacturing, food, logistics) with minor data adjustments.

4️⃣ Business Objectives

1. Help textile factories achieve **ESG compliance** easily.
2. Enable **data-driven sustainability decisions**.
3. Build trust and transparency for **eco-conscious consumers**.
4. Create a scalable, subscription-based model for long-term business.

5️⃣ Key Features

Feature	Description
Data Input Module	Factory managers enter energy, water, and waste data.
Emission Calculator	Calculates CO_2 equivalent using standard formulas (e.g., fuel × emission factor).
ESG Dashboard	Displays carbon footprint trends, water usage, and waste data visually.
ML-based Insights	Gives reduction tips and predicts future emissions.
Auto Reports	Generates branded ESG reports for clients or compliance bodies.
User Management	Role-based access (Admin, Factory Manager, Analyst).

6️⃣ Benefits for Users

✅ Save time and effort in sustainability reporting
✅ Increase compliance with global ESG standards
✅ Improve brand reputation
✅ Access real-time insights
✅ Make smarter resource decisions to reduce waste and cost

7️⃣ Business Model

Category	Description
Target Market	Small & medium textile manufacturers
Revenue Stream	SaaS model (subscription), Premium Reports, API integration
Scalability	Can extend to other industries (food, logistics, manufacturing)
Future Vision	Become India's first open ESG tracking ecosystem

---

8️⃣ Role of Business Analyst (You)

As the **Business Analyst**, your contributions include:

- Defined the **problem & scope**
- Conducted **requirement gathering**
- Created **process flow diagrams**
- Designed **data input structures** and validated logic
- Helped connect the **business goal to technical features**
- Coordinated with developers for **frontend & API clarity**
- Ensured **ESG compliance and measurable KPIs**

9️⃣ Testing & Evaluation Plan

Type	Description
Unit Testing	For emission calculation and input validation
Integration Testing	Between data input, dashboard, and report generation
UAT (User Acceptance Testing)	With factory users and analysts
Performance Testing	For real-time dashboard response

🔟 Conclusion

CarbonKanakku bridges the gap between sustainability and technology.
It empowers textile industries to become eco-conscious and compliant while staying competitive.
With continued innovation, this platform can evolve into a **nationwide sustainability intelligence system** for multiple sectors.