Carbon Emission Analysis in Coal Mining

Abstract

This project introduces a specialized software solution designed to estimate carbon emissions and evaluate carbon sinks in coal mining operations. The tool enables users to input operational data such as equipment used, fuel type, and consumption levels, and calculates emissions using standard emission factors. It also incorporates a carbon sink analysis module, which helps assess the offset potential based on environmental and site-specific data.

In addition to estimation, the system offers remedial suggestions to reduce or balance out emissions, such as fuel optimization or land restoration measures. By providing accurate, activity-based emission data and actionable insights, the software serves as a decision-support tool for mining stakeholders, contributing to more sustainable and environmentally responsible mining practices in line with India's climate goals.

Problem Statement

Coal mining is a major contributor to carbon emissions, posing a challenge to India's carbon neutrality goals. Existing solutions lack granularity in estimating emissions and do not effectively integrate carbon sink analysis or remedial suggestions. This project aims to develop a software tool that calculates carbon emissions based on user-defined inputs such as equipment and fuel data, estimates associated carbon sinks, and suggests actionable strategies to reduce or offset emissions. The goal is to support informed decision-making for sustainable mining practices

Introduction

As India continues to pursue sustainable development and climate commitments, balancing energy production with environmental responsibility becomes increasingly important. Coal mining, while crucial to the country's energy infrastructure, is one of the major contributors to carbon emissions. Addressing these emissions requires a shift from broad estimations to more granular, activity-specific assessments. However, most existing tools fail to offer detailed emission calculations based on operational parameters or integrate carbon sink evaluations effectively

This project seeks to bridge that gap by developing a software tool that enables accurate estimation of carbon emissions in coal mining operations. By incorporating user-defined inputs such as fuel type, equipment specifications, and emission factors, the tool ensures tailored and reliable outputs. Moreover, by evaluating region-specific carbon sinks and suggesting appropriate mitigation strategies, it empowers stakeholders to make data-informed decisions that support the national goal of carbon neutrality while sustaining productivity.

Theory Concepts Used:

The project structures emission-related activities using a base class that encapsulates common attributes, with subclasses representing specific activity types. This reflects the use of inheritance to design a flexible and extensible class hierarchy aligned with object-oriented best practices.

1. **JDBC** – Java Database Connectivity: JDBC enables Java applications to interact with relational databases by providing an interface for executing SQL queries. It simplifies database operations with components like Driver Manager, Connection, Statement, and Result Set. The use of Prepared Statement allows for parameterized queries, improving performance and security by preventing SQL injection. In your project, JDBC handles interactions with a MySQL database, allowing for dynamic storage, retrieval, and manipulation of emission data, ensuring smooth integration between the Java application and the backend database.
2. **Exception Handling**: Exception handling is a key feature in Java that allows developers to manage runtime errors in a controlled manner, ensuring that the program doesn't terminate unexpectedly. The try, catch, finally, throw, and throws keywords provide a structured way to

catch and manage exceptions, improving program robustness. By distinguishing between checked and unchecked exceptions, Java allows for both error management and cleaner code. In your project, exception handling ensures that errors in database connections, query execution, or input validation are properly managed, enhancing the user experience and system stability.

3. **Inheritance**: Inheritance is a fundamental concept of Object-Oriented Programming (OOP) that enables classes to inherit attributes and methods from other classes, promoting code reuse, flexibility, and easier maintenance. Java supports various inheritance types such as single, multilevel, and hierarchical inheritance. In your project, inheritance is used to model emission activities, with a base class defining shared properties and methods, while specific emission types extend the base class, adding their own unique behaviors. This structure promotes modularity, reduces redundancy, and allows easy scalability by adding new emission types or features without altering existing code.

Code

Main class including operations:

```
package com.coalemission.beans;
import java.util.*;
// Main application class
public class CoalMineEmissionApp {
    private static Scanner sc = new Scanner(System.in);
    private static List<MiningActivity> activities = new ArrayList<>();
    public static void main(String[] args) {
        boolean exit = false;
        while (!exit) {
            System.out.println("\n=== Coal Mine Emission Management ===");
            System.out.println("1. Add Activity");
            System.out.println("2. Display Activities");
            System.out.println("3. Update Emission Factor");
            System.out.println("4. Show Emission Report");
            System.out.println("5. Exit");
            System.out.print("Enter choice: ");
            int choice = sc.nextInt();
            switch (choice) {
            case 1 -> addActivity();
            case 2 -> displayActivities();
            case 3 -> updateEmissionFactor();
            case 4 -> showEmissionReport();
            case 5 -> exit = true;
            default -> System.out.println("Invalid choice.");
            }
```

```
		}
	}
	public  static void showEmissionReport() {
		// TODO Auto-generated method stub
		for (MiningActivity a : activities)
		{
			double calemission=a.fuelConsumed *a.emissionFactor;
			System.out.printf("%-10s %-15s %-10.2f %-10.2f %-10.2f\n",
					a.mineID, a.equipment, a.fuelConsumed,
a.emissionFactor,calemission);
			if(calemission<=100)
			{
				System.out.println("No action is triggered\nThe emissions are within
a tolerable range, and hence, no remedial action is deemed necessa"
					+ "ry.");
			}
			else if(calemission>100 && calemission<=300)
			{
				System.out.println("Carbon sink calculation\nModerate emissions
detected; basic carbon sinks like tree planting can effectively neutralize this impact.");
			}
			else if(calemission>300 && calemission<=600)
			{
				System.out.println("Sink + Medium Remedial Action\nHigh carbon
emissions identified; appropriate remedial steps such as afforestation and soil carbon sequestration
are advised");
			}
			else
			{
```

```
                                System.out.println("Sink + Aggressive Remedial\nCritical emission level recorded; large-scale carbon offset mechanisms like wetland restoration or reforestation must be initiated");
                        }
                }
        }
        // Add a mining activity
        private static void addActivity() {
                System.out.print("Enter Mine ID: ");
                String mineID = sc.next();
                System.out.print("Enter Equipment: ");
                String equipment = sc.next();
                System.out.print("Enter Fuel Consumed: ");
                double fuel = sc.nextDouble();
                System.out.print("Enter Emission Factor: ");
                double factor = sc.nextDouble();
                System.out.print("Enter Activity Type (excavation/drilling/transportation/hauling)”);
                String type = sc.next().toLowerCase();
                MiningActivity activity = switch (type) {
                case "excavation" -> new Excavation(mineID, equipment, fuel, factor);
                case "drilling" -> new Drilling(mineID, equipment, fuel, factor);
                case "transportation" -> new Transportation(mineID, equipment, fuel, factor);
                case "hauling" -> new Hauling(mineID, equipment, fuel, factor);
                default -> null;
                };
                if (activity != null) {
                        activities.add(activity);
```

```
                System.out.println("Activity added successfully!");
            } else {
                System.out.println("Invalid activity type.");
            }
        }
        // Display all stored activities
        private static void displayActivities() {
            if (activities.isEmpty()) {
                System.out.println("No mining activity data found.");
                return;
            }
            System.out.printf("\n%-10s %-15s %-10s %-10s %-10s\n", "MineID",
"Equipment", "Fuel", "Factor", "Emission");
            for (MiningActivity activity : activities) {
                activity.displayInfo();
            }
        }
       // Update emission factor by mineID
        private static void updateEmissionFactor() {
            System.out.print("Enter Mine ID to update: ");
            String id = sc.next();
            boolean found = false;
            for (MiningActivity activity : activities) {
                if (activity.getMineID().equals(id)) {
                    System.out.print("Enter new emission factor: ");
                    double newFactor = sc.nextDouble();
                    activity.setEmissionFactor(newFactor);
```

```
                        System.out.println("Emission factor updated successfully.");
                        found = true;
                        break;
                    }
                }
                if (!found) {
                    System.out.println("Mine ID not found.");
                }
            }
        }
```

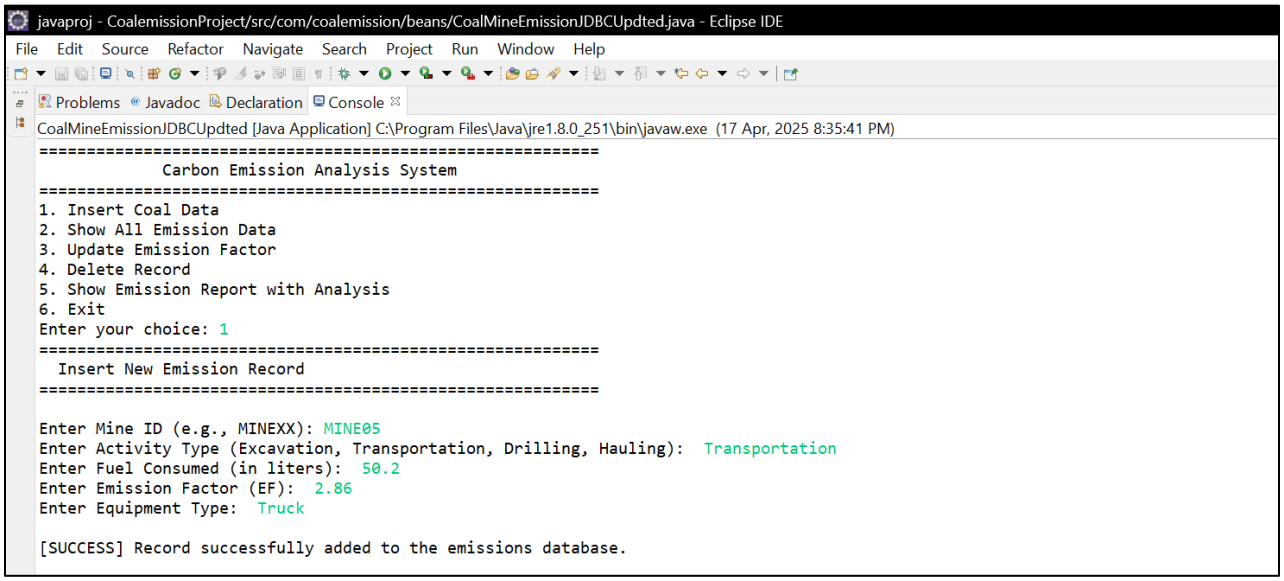
Use of inheritance:

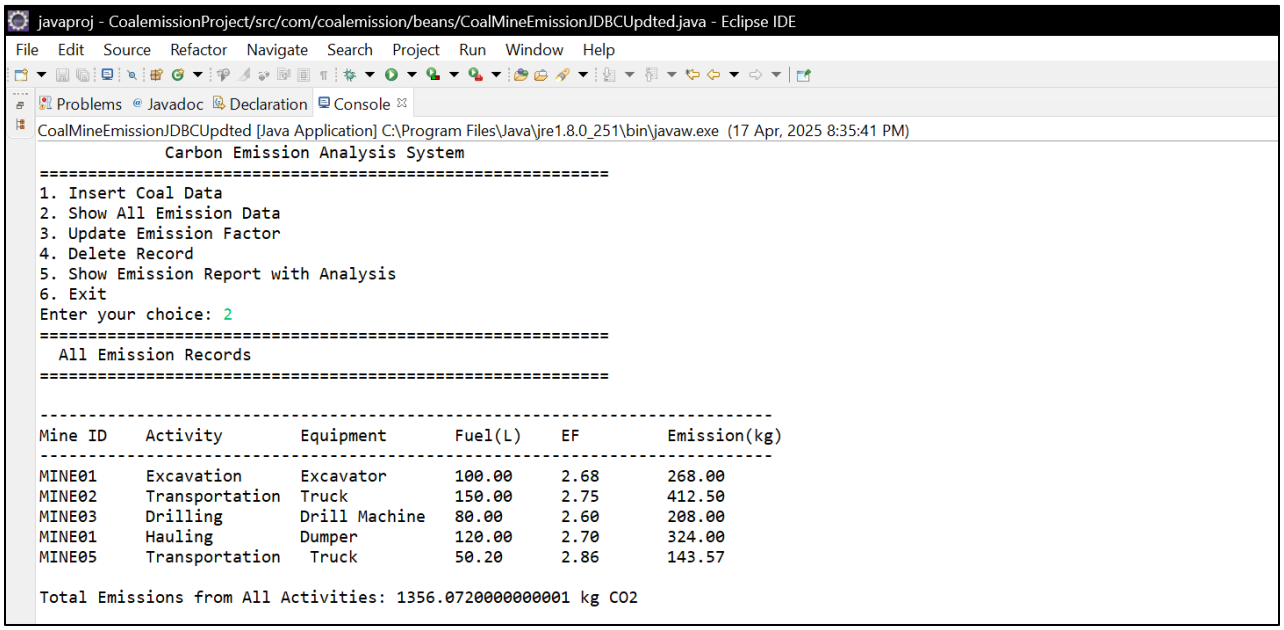
```
package com.coalemission.beans;
class MiningActivity {
    protected String mineID;
    protected String equipment;
    protected double fuelConsumed;
    protected double emissionFactor;
    public MiningActivity(String mineID, String equipment, double fuelConsumed, double emissionFactor) {
        this.mineID = mineID;
        this.equipment = equipment;
        this.fuelConsumed = fuelConsumed;
        this.emissionFactor = emissionFactor;
    }
    public double calculateEmission() {
        return fuelConsumed * emissionFactor;
    }
```

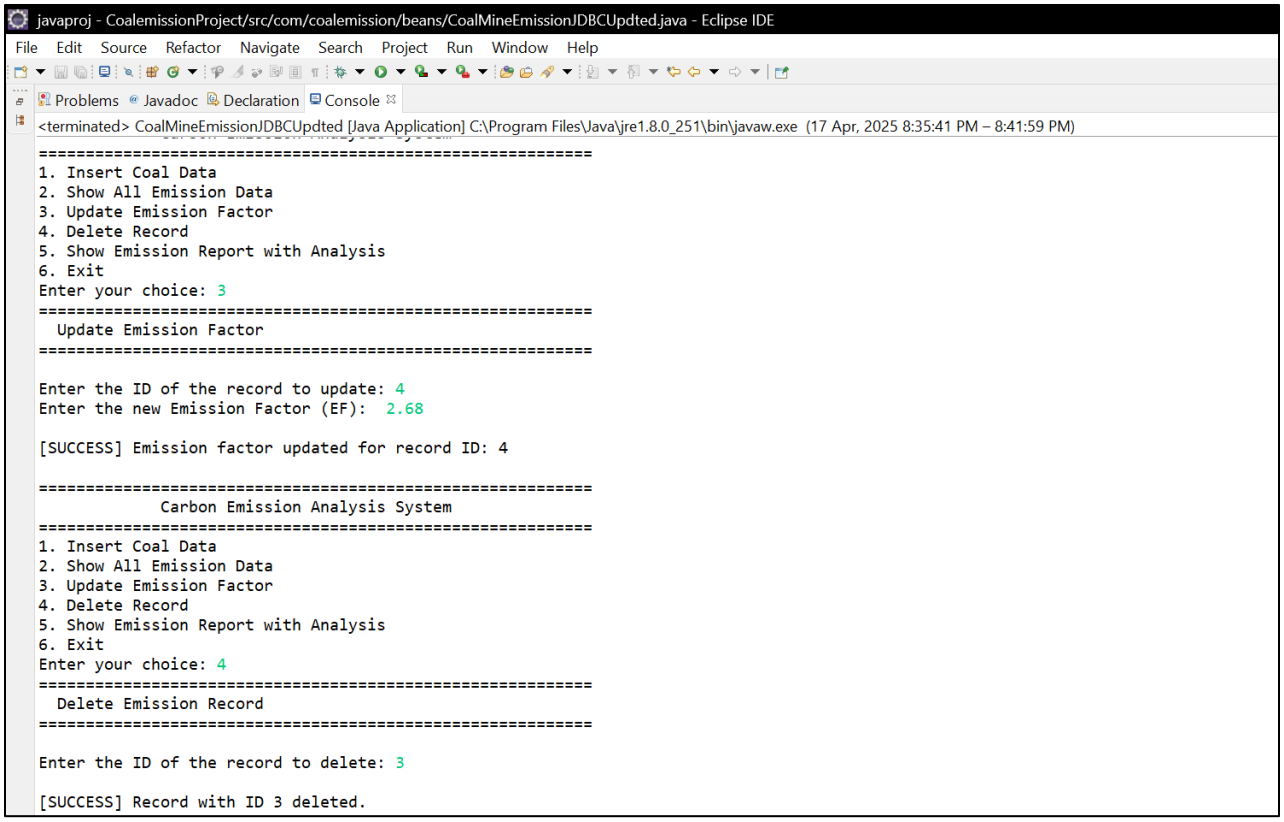
```
    public void displayInfo() {
        System.out.printf("%-10s %-15s %-10.2f %-10.2f %-10.2f\n",
                mineID, equipment, fuelConsumed, emissionFactor, calculateEmission());
    }
    public String getMineID() {
        return mineID;
    }
    public void setEmissionFactor(double emissionFactor) {
        this.emissionFactor = emissionFactor;
    }
}
// Subclasses
class Excavation extends MiningActivity {
    public Excavation(String mineID, String equipment, double fuelConsumed, double
emissionFactor) {
        super(mineID, equipment, fuelConsumed, emissionFactor);
    }
}
class Transportation extends MiningActivity {
    public Transportation(String mineID, String equipment, double fuelConsumed, double
emissionFactor) {
        super(mineID, equipment, fuelConsumed, emissionFactor);
    }
}
class Drilling extends MiningActivity {
    public Drilling(String mineID, String equipment, double fuelConsumed, double emissionFactor)
{
        super(mineID, equipment, fuelConsumed, emissionFactor);
```

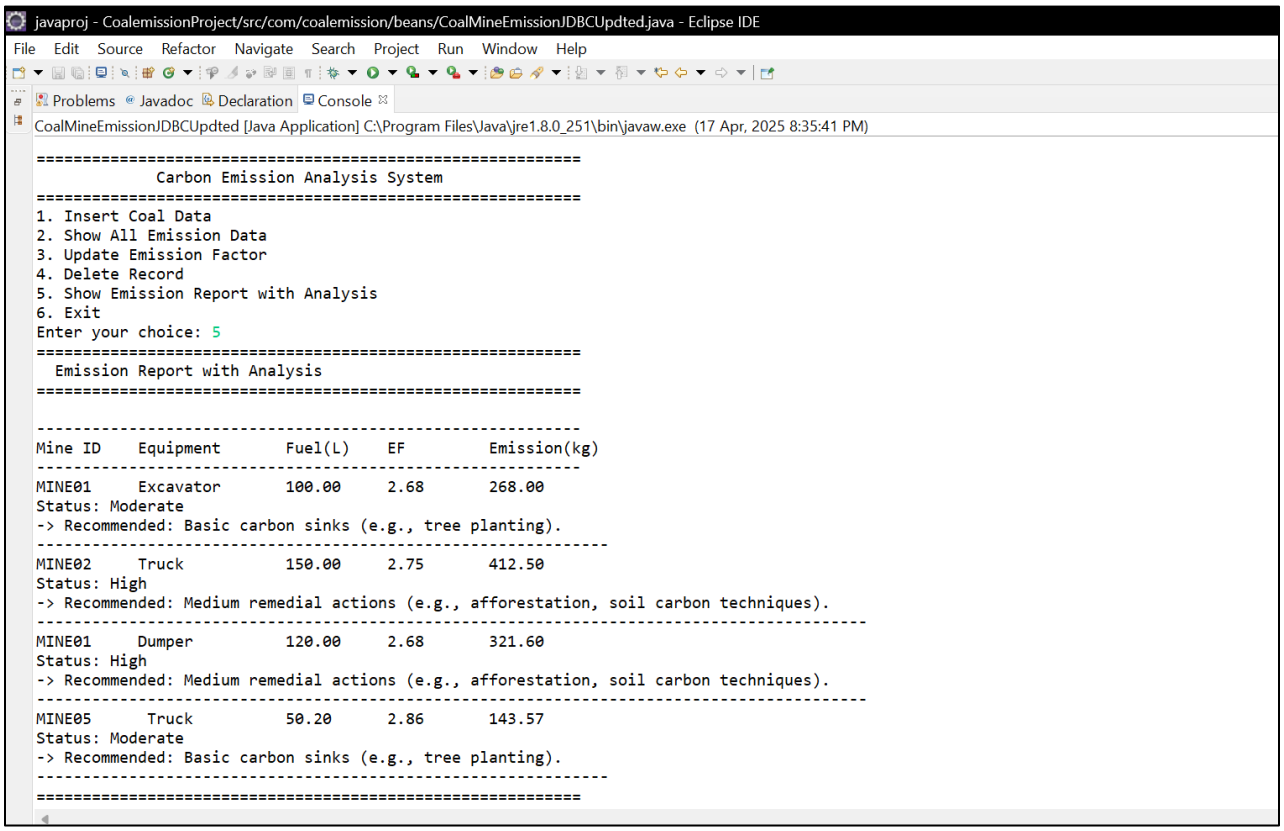
```
    }
}
class Hauling extends MiningActivity {
    public Hauling(String mineID, String equipment, double fuelConsumed, double emissionFactor)
{
        super(mineID, equipment, fuelConsumed, emissionFactor);
    }
}
```

Output


javaproj - CoalemissionProject/src/com/coalemission/beans/CoalMineEmissionJDBCUpdted.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Problems Javadoc Declaration Console
CoalMineEmissionJDBCUpdted [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe (17 Apr, 2025 8:35:41 PM)
Carbon Emission Analysis System
1. Insert Coal Data
2. Show All Emission Data
3. Update Emission Factor
4. Delete Record
5. Show Emission Report with Analysis
6. Exit
Enter your choice: 1
Insert New Emission Record
Enter Mine ID (e.g., MINEXX): MINE05
Enter Activity Type (Excavation, Transportation, Drilling, Hauling): Transportation
Enter Fuel Consumed (in liters): 50.2
Enter Emission Factor (EF): 2.86
Enter Equipment Type: Truck
[SUCCESS] Record successfully added to the emissions database.


javaproj - CoalemissionProject/src/com/coalemission/beans/CoalMineEmissionJDBCUpdted.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Problems Javadoc Declaration Console
CoalMineEmissionJDBCUpdted [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe (17 Apr, 2025 8:35:41 PM)
Carbon Emission Analysis System
1. Insert Coal Data
2. Show All Emission Data
3. Update Emission Factor
4. Delete Record
5. Show Emission Report with Analysis
6. Exit
Enter your choice: 2
All Emission Records
Mine ID Activity Equipment Fuel(L) EF Emission(kg)
MINE01 Excavation Excavator 100.00 2.68 268.00
MINE02 Transportation Truck 150.00 2.75 412.50
MINE03 Drilling Drill Machine 80.00 2.60 208.00
MINE01 Hauling Dumper 120.00 2.70 324.00
MINE05 Transportation Truck 50.20 2.86 143.57
Total Emissions from All Activities: 1356.0720000000001 kg CO2


javaproj - CoalemissionProject/src/com/coalemission/beans/CoalMineEmissionJDBCUpdted.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Problems Javadoc Declaration Console
<terminated> CoalMineEmissionJDBCUpdted [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe (17 Apr, 2025 8:35:41 PM – 8:41:59 PM)
==========
1. Insert Coal Data
2. Show All Emission Data
3. Update Emission Factor
4. Delete Record
5. Show Emission Report with Analysis
6. Exit
Enter your choice: 3
==========
Update Emission Factor
==========
Enter the ID of the record to update: 4
Enter the new Emission Factor (EF): 2.68
[SUCCESS] Emission factor updated for record ID: 4
==========
Carbon Emission Analysis System
==========
1. Insert Coal Data
2. Show All Emission Data
3. Update Emission Factor
4. Delete Record
5. Show Emission Report with Analysis
6. Exit
Enter your choice: 4
==========
Delete Emission Record
==========
Enter the ID of the record to delete: 3
[SUCCESS] Record with ID 3 deleted.


javaproj - CoalemissionProject/src/com/coalemission/beans/CoalMineEmissionJDBCUpdted.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Problems Javadoc Declaration Console
CoalMineEmissionJDBCUpdted [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe (17 Apr, 2025 8:35:41 PM)
==========
Carbon Emission Analysis System
==========
1. Insert Coal Data
2. Show All Emission Data
3. Update Emission Factor
4. Delete Record
5. Show Emission Report with Analysis
6. Exit
Enter your choice: 5
==========
Emission Report with Analysis
==========
Mine ID Equipment Fuel(L) EF Emission(kg)
MINE01 Excavator 100.00 2.68 268.00
Status: Moderate
-> Recommended: Basic carbon sinks (e.g., tree planting).
MINE02 Truck 150.00 2.75 412.50
Status: High
-> Recommended: Medium remedial actions (e.g., afforestation, soil carbon techniques).
MINE01 Dumper 120.00 2.68 321.60
Status: High
-> Recommended: Medium remedial actions (e.g., afforestation, soil carbon techniques).
MINE05 Truck 50.20 2.86 143.57
Status: Moderate
-> Recommended: Basic carbon sinks (e.g., tree planting).

Conclusion

This project offers a practical solution for monitoring and analyzing carbon emissions in coal mining activities. By organizing different mining processes in a structured and efficient manner, the system helps track the amount of fuel used and the amount of carbon dioxide released.

Information is taken directly from a database and processed in a way that keeps the system organized, clean, and easy to manage. Though the technical backbone involves basic programming structures and methods to work with the data, the main focus remains on building a clear, understandable system that reflects how such tools can be used to support real-world environmental concerns. The project also leaves room for future improvement, where smarter data handling techniques can make the system even more effective in promoting sustainable practices