DBMS PROJECT REPORT

Industry Carbon Footprint Management System

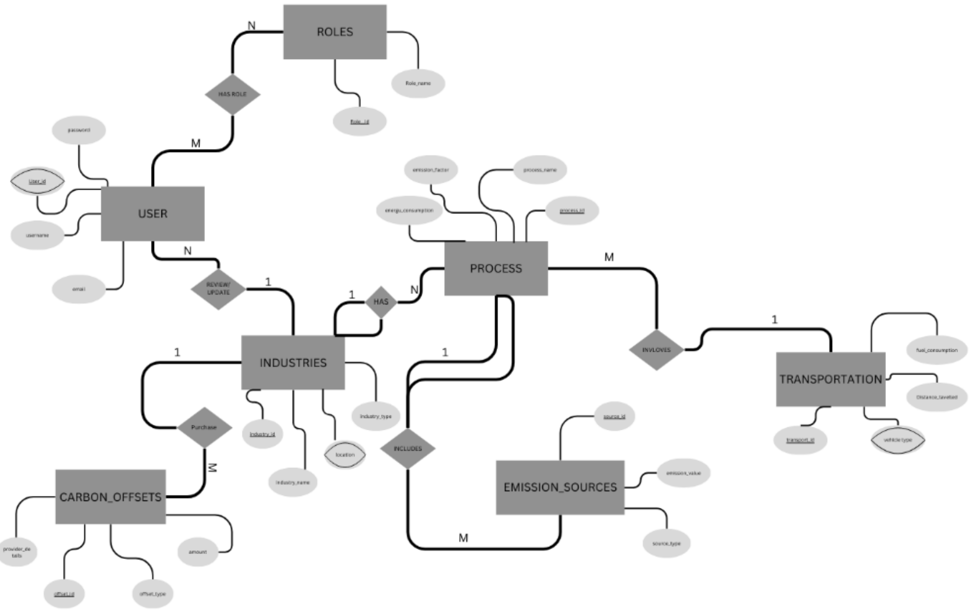
PES1UG22CS324 – Maitreya Tiwary

PES1UG22CS350 – M K Gagan

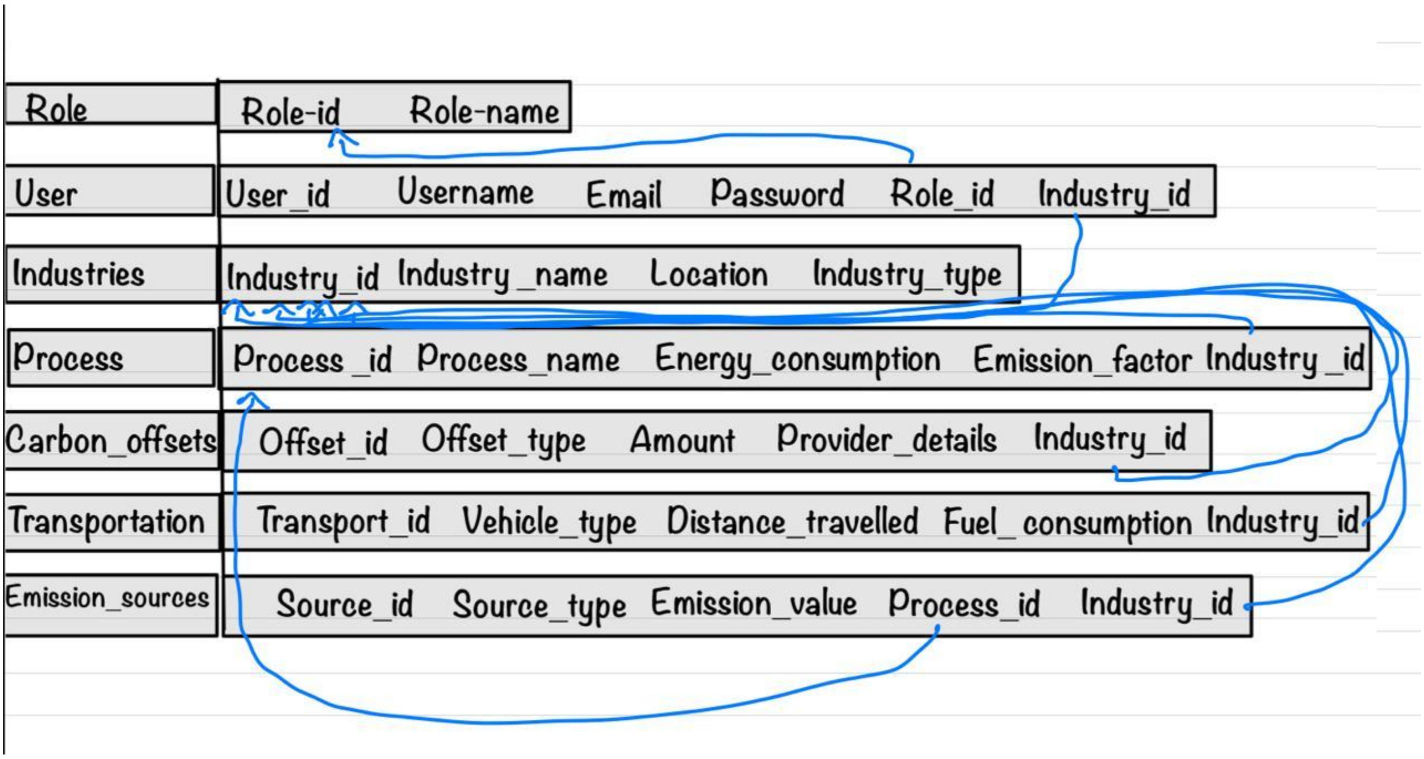
Introduction:

The Industry Carbon Footprint Management System (ICFMS) is a robust database solution aimed at tracking, managing, and reducing carbon emissions in the industrial sector. Utilizing MySQL architecture, it efficiently organizes critical data on industrial activities, emissions, and carbon offsets, ensuring data integrity through the use of foreign key constraints and triggers. The system offers real-time monitoring, compliance reporting, and actionable insights, assisting industries in meeting regulatory standards and enhancing sustainability efforts. Designed to be secure, scalable, and efficient, ICFMS enables industries to make informed, environmentally-conscious decisions.

ER Diagram:



Relational Schema:



Role	Role-id	Role-name				
User	User_id	Username	Email	Password	Role_id	Industry_id
Industries	Industry_id	Industry_name	Location	Industry_type		
Process	Process_id	Process_name	Energy_consumption	Emission_factor	Industry_id	
Carbon_offsets	Offset_id	Offset_type	Amount	Provider_details	Industry_id	
Transportation	Transport_id	Vehicle_type	Distance_travelled	Fuel_consumption	Industry_id	
Emission_sources	Source_id	Source_type	Emission_value	Process_id	Industry_id	

Triggers:

1. Trigger: before_industry_delete

This trigger executes before a row is deleted from the industries table.

When an industry is deleted, it also deletes all associated rows in the Process table where the industry_id matches the industry_id of the industry being deleted.

Ensures referential integrity by cleaning up dependent records in the Process table.

Prevents orphaned rows in the Process table that reference an industry that no longer exists.

2. Trigger: before_process_delete

This trigger executes before a row is deleted from the Process table.

When a process is deleted, it also deletes all related rows in the Emission_Sources table where the process_id matches the process_id of the process being deleted.

Ensures referential integrity by cleaning up dependent records in the Emission_Sources table.

Prevents orphaned rows in the Emission_Sources table that reference a process that no longer exists.

Functions:

1. Function: get_total_carbon_offset

This function takes an industryId as input and calculates the total carbon offset for that industry by summing the amount_offset values in the Carbon_Offsets table for the given industry_id.

The function returns the total carbon offset as a DECIMAL(10,2) value.

- Steps/Logic:

1. Declares a variable totalOffset to hold the sum.
2. Executes a SELECT query with a SUM aggregate function to calculate the total offset for the specified industry_id.
3. Assigns the result to totalOffset.
4. Returns the calculated value.

Provides a quick way to retrieve the total carbon offset for a specific industry.

Can be used in other SQL queries, reports, or applications that need to access this information.

Aggregated Queries:

1. The function get_total_carbon_offset includes an aggregated query:

```
SELECT SUM(amount_offset) INTO totalOffset

FROM Carbon_Offsets

WHERE industry_id = industryId;
```

> It calculates the total (SUM) of the amount_offset column for rows in the Carbon_Offsets table where the industry_id matches the given input parameter (industryId).

Joins:

1. SELECT p.process_id, p.process_name, p.energy_consumption, p.emission_factor, i.industry_name

FROM Process p

JOIN Industries i ON p.industry_id = i.industry_id

WHERE p.industry_id = %s

- **Join Type**: INNER JOIN (default JOIN).
- **Tables Involved**:
  - Process (p) table, which contains information about different processes (e.g., process_id, process_name, etc.).
  - Industries (i) table, which contains industry details (e.g., industry_id, industry_name, etc.).
- **Join Condition**: ON p.industry_id = i.industry_id, meaning it links processes to the industries they belong to.
- **Purpose**: This query fetches the processes belonging to the specific industry assigned to the logged-in Industry Manager. The WHERE p.industry_id = %s filters the results to only include processes for the industry_id associated with the Industry Manager.

2. SELECT p.process_id, p.process_name, p.energy_consumption, p.emission_factor, i.industry_name

FROM Process p

LEFT JOIN Industries i ON p.industry_id = i.industry_id

- **Join Type**: LEFT JOIN.
- **Tables Involved**:

  - Process (p) table.
  - Industries (i) table.

- **Join Condition**: ON p.industry_id = i.industry_id.
- **Purpose**: This query fetches all processes, regardless of whether they are associated with an industry. The LEFT JOIN ensures that processes without a matching industry_id in the Industries table are still included in the results, with the industry_name field being NULL for those rows.

3. SELECT co.offset_id, co.offset_type, co.amount_offset, co.date_purchased, co.provider_details, i.industry_name

FROM Carbon_Offsets co

JOIN Industries i ON co.industry_id = i.industry_id

WHERE co.industry_id = %s

- **Join Type**: INNER JOIN (default JOIN).
- **Tables Involved**:
  - Carbon_Offsets (co) table, which contains details about carbon offset purchases
    (e.g., offset_id, offset_type, amount_offset, etc.).
  - Industries (i) table, which provides information about industries (e.g., industry_id, industry_name, etc.).
- **Join Condition**: ON co.industry_id = i.industry_id, linking the carbon offset records to the industries they belong to.
- **Purpose**: This query fetches all carbon offset records associated with the specific industry_id of the Industry Manager.
  The WHERE co.industry_id = %s filters results to include only those related to the logged-in manager's assigned industry.

4. SELECT co.offset_id, co.offset_type, co.amount_offset, co.date_purchased, co.provider_details, i.industry_name

FROM Carbon_Offsets co

LEFT JOIN Industries i ON co.industry_id = i.industry_id

- **Join Type**: LEFT JOIN.
- **Tables Involved**:
    - Carbon_Offsets (co) table.
    - Industries (i) table.
- **Join Condition**: ON co.industry_id = i.industry_id.
- **Purpose**: This query retrieves all carbon offset records, regardless of whether they are linked to an industry. The LEFT JOIN ensures that carbon offset entries without a matching industry_id in the Industries table are still included, with the industry_name field being NULL for those rows.

5. SELECT t.transport_id, t.vehicle_type, t.distance_travelled, t.fuel_consumption, i.industry_name

FROM Transportation t

JOIN Industries i ON t.industry_id = i.industry_id

WHERE t.industry_id = %s

- **Join Type**: INNER JOIN (default JOIN).
- **Tables Involved**:
    - Transportation (t) table, which contains records of transportation details
      (e.g., transport_id, vehicle_type, distance_travelled, etc.).

    - Industries (i) table, which provides information about industries (e.g., industry_id, industry_name, etc.).

- **Join Condition**: ON t.industry_id = i.industry_id, linking transportation records to their respective industries.
- **Purpose**: This query fetches all transportation records associated with the industry_id of the logged-in Industry Manager. The WHERE t.industry_id = %s ensures only records for the manager's assigned industry are retrieved.

6. SELECT t.transport_id, t.vehicle_type, t.distance_travelled, t.fuel_consumption, i.industry_name

FROM Transportation t

LEFT JOIN Industries i ON t.industry_id = i.industry_id

- **Join Type**: LEFT JOIN.
- **Tables Involved**:
    - Transportation (t) table.
    - Industries (i) table.
- **Join Condition**: ON t.industry_id = i.industry_id.
- **Purpose**: This query retrieves all transportation records, regardless of whether they are linked to an industry. The LEFT JOIN ensures that entries without a matching industry_id in the Industries table are still included, with the industry_name being NULL for those rows.

7. SELECT e.source_id, e.source_type, e.emission_value, i.industry_name

FROM Emission_Sources e

JOIN Industries i ON e.industry_id = i.industry_id

WHERE e.industry_id = %s

- **Join Type**: INNER JOIN (default JOIN).
- **Tables Involved**:
  - Emission_Sources (e) table, which contains records of emission sources
    (e.g., source_id, source_type, emission_value, etc.).
  - Industries (i) table, which provides information about industries (e.g., industry_id, industry_name, etc.).
- **Join Condition**: ON e.industry_id = i.industry_id, linking emission sources to their respective industries.
- **Purpose**: This query retrieves emission source records associated with the specific industry_id of the logged-in Industry Manager. The WHERE e.industry_id = %s ensures only records for the manager's assigned industry are included in the results.

8. SELECT e.source_id, e.source_type, e.emission_value, i.industry_name

FROM Emission_Sources e

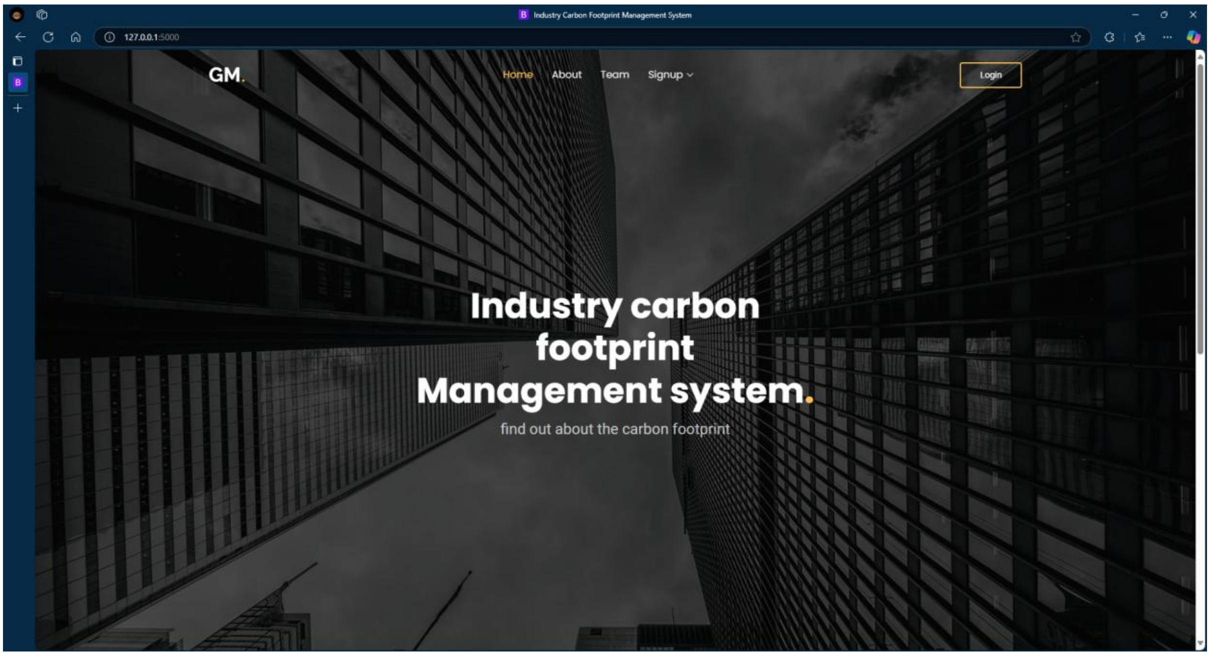
LEFT JOIN Industries i ON e.industry_id = i.industry_id

- **Join Type**: LEFT JOIN.
- **Tables Involved**:
  - Emission_Sources (e) table.
  - Industries (i) table.
- **Join Condition**: ON e.industry_id = i.industry_id.
- **Purpose**: This query retrieves all emission source records, regardless of whether they are linked to an industry. The LEFT

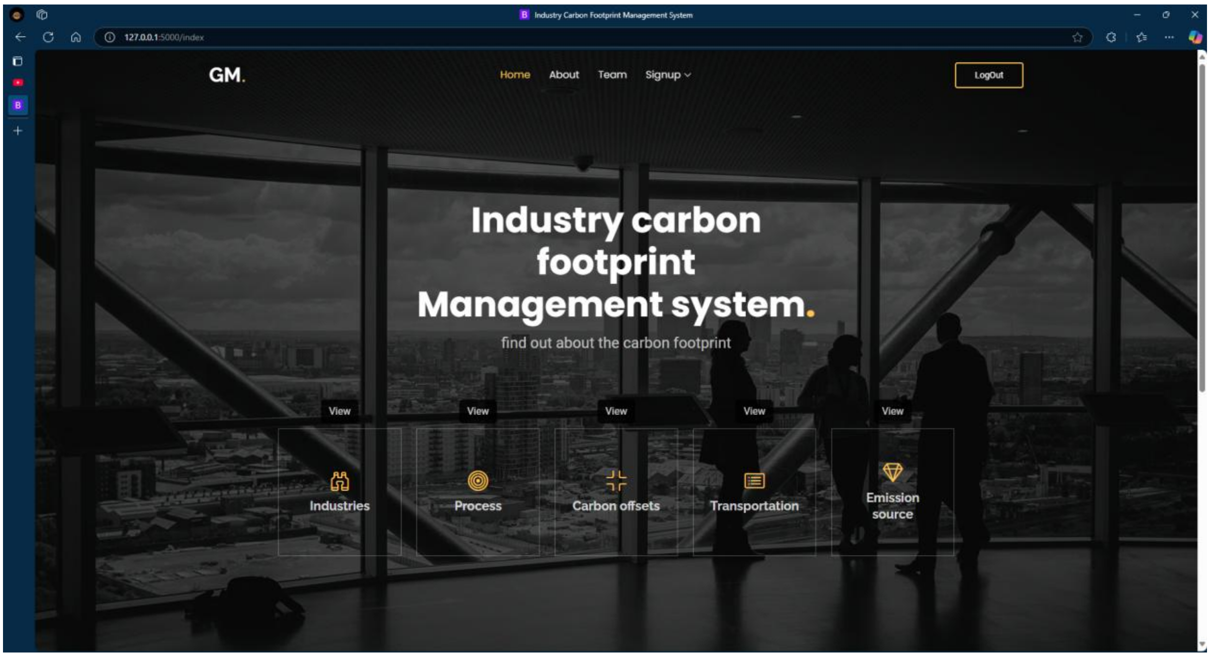
JOIN ensures that emission source records without a matching industry_id in the industries table are still included, with the industry_name field being NULL for those rows.

Working Instances:

Home screen:

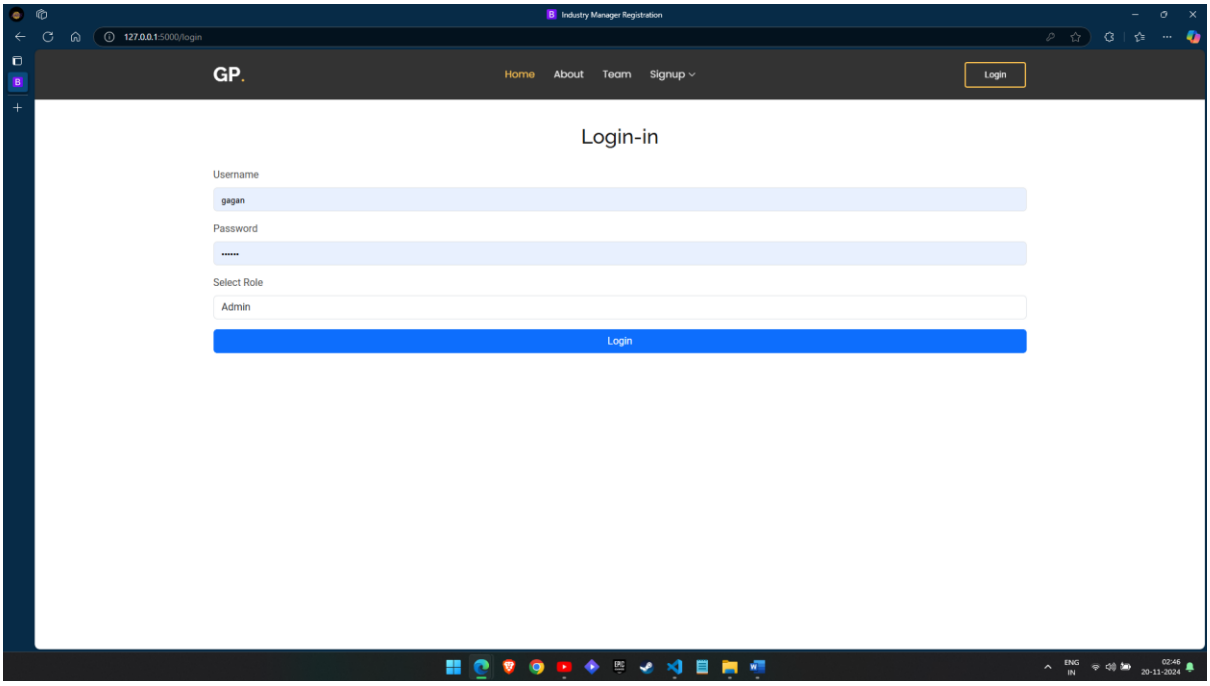

Industry Carbon Footprint Management System
127.0.0.1:5000
GM.
Home
About
Team
Signup
Login
Industry carbon footprint Management system.
find out about the carbon footprint

Dashboard:



This is the page the user will see once he has logged in. The different tiles take him into adding into the tables with the respective labels

Login page:



Login-in

Username
gagan

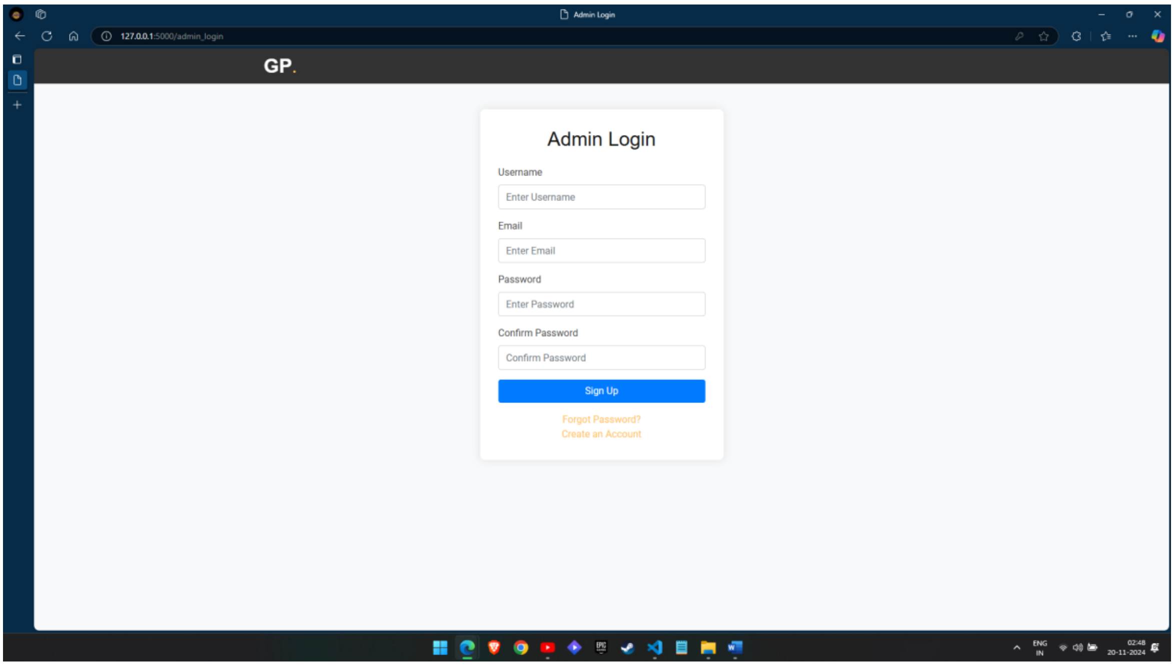
Password
••••••

Select Role
Admin

Login

Here, pre-existing users can login using their credentials, and based on the permission level allotted to their account, they will be able to create changes, or only view the details ahead.

Sign-up page:



Admin Login

Username
Enter Username

Email
Enter Email

Password
Enter Password

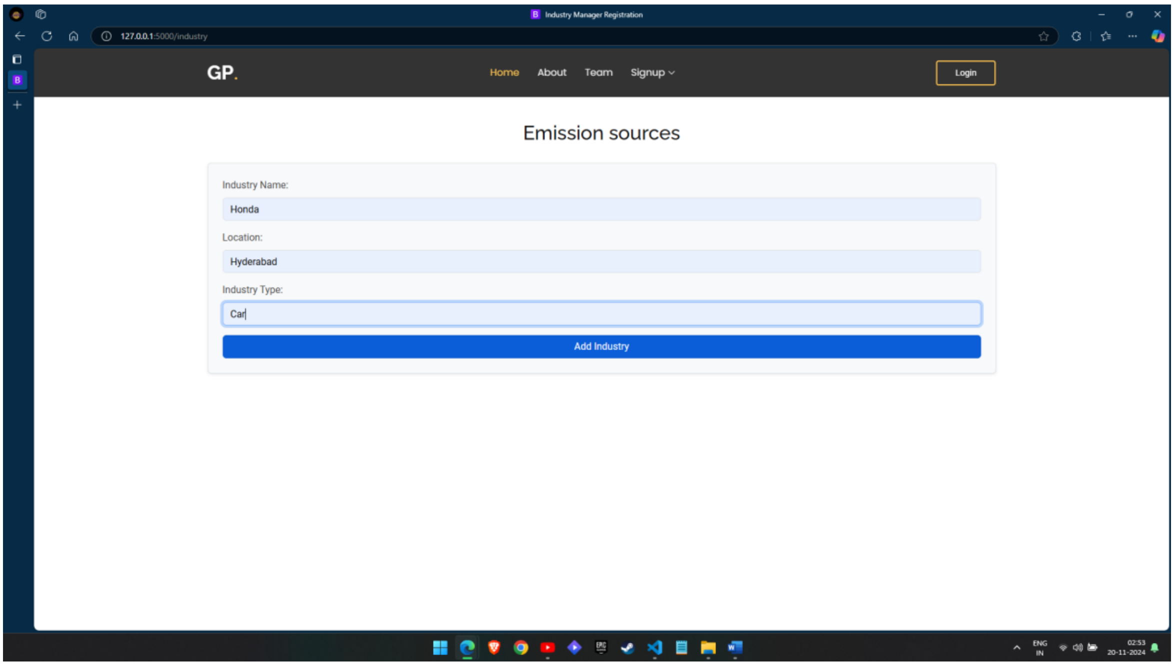
Confirm Password
Confirm Password

Sign Up

Forgot Password?
Create an Account

In the ‘sign-up’ tab, the user is prompted with options to sign up as ‘admin’, ‘industry manager’, ‘auditor’ or ‘guest’. Admins have access to view and/or alter any table, Industry managers can alter their respective data as well as view any data and auditors and guests can only view data.

Adding a new industry:



Emission sources

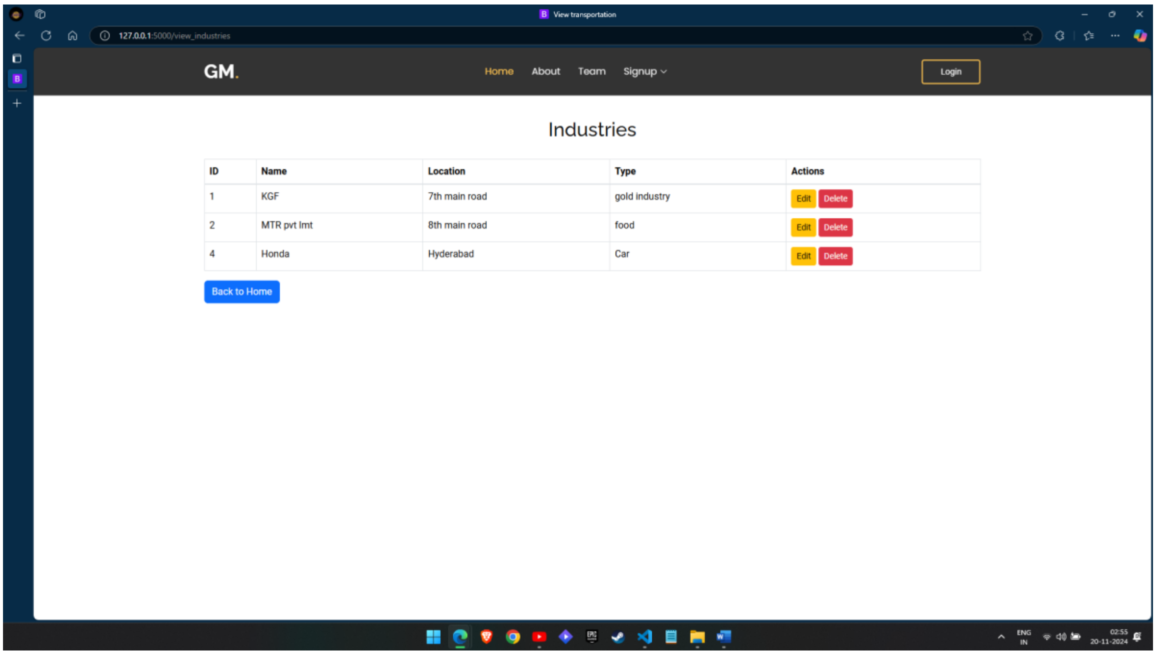
Industry Name: Honda

Location: Hyderabad

Industry Type: Car

Here, we can add a new industry into the table.

Viewing/deleting Industries:



Industries

ID	Name	Location	Type	Actions
1	KGF	7th main road	gold industry	Edit Delete
2	MTR pvt lmt	8th main road	food	Edit Delete
4	Honda	Hyderabad	Car	Edit Delete

Here, we can view the added industries, as well as edit the information of, or delete an industry. Similar tasks can be carried out for processes, carbon offsets, transportation and emission sources.