Project Initialization and Planning Phase

Date	13 June 2025
Team ID	SWTID1749709340
Project Title	Predicting Co2 Emission by countries Using Machine Learning
Maximum Marks	3 Marks

Project Proposal (Proposed Solution) template

This project proposal outlines a solution to address a specific problem. With a clear objective, defined scope, and a concise problem statement, the proposed solution details the approach, key features, and resource requirements, including hardware, software, and personnel.

Project Overview	
Objective	To develop a machine learning-based web application that predicts CO_2 emissions of countries based on user-provided year and country input for aiding policy planning and environmental research.
Scope	The project covers data collection, preprocessing, training regression models, developing a web interface for user input, visualizing prediction results, and deploying the application for public and academic use.
Problem Statement	
Description	It is difficult for policymakers, researchers, and citizens to predict future CO_2 emissions for countries easily, as available data is static, scattered, and not user-friendly. This hinders effective planning and climate research.
Impact	Solving this enables data-driven climate action, easy trend analysis, and informed policy decisions by providing accessible, interactive, and accurate CO_2 emission forecasts for each country.
Proposed Solution	
Approach	We will collect historical CO_2 emission data, preprocess and train machine learning regression models, and integrate them with a Flask-based web application where users can input a country and year to

	receive predictions with visual plots
Key Features	- User-friendly web interface for input and prediction - Country-wise CO_2 emission prediction with graphical visualization - Uses machine learning for accurate forecasts - Accessible for policymakers, researchers, and students

Resource Requirements

Resource Type	Description	Specification/Allocation
Hardware		
Computing Resources	CPU for development, model training, and testing	Intel i5/i7, 4 cores
Memory	RAM for data processing and model execution	8 GB
Storage	Storage for datasets, models, logs	256 GB SSD
Software		
Frameworks	Python frameworks	Flask
Libraries	Additional Python libraries	scikit-learn, pandas, numpy matplotlib
Development Environment	IDE, version control	Jupyter Notebook, GitHub
Data		
Data	Source, size, format	Kaggle/World Bank CO_2 dataset, CSV format