

- Project summary

- [The issue we are hoping to solve](#)
- [How our technology solution can help](#)
- [Our idea](#)

- Technology implementation

- [IBM AI service\(s\) used](#)
- [Other IBM technology used](#)
- [Solution architecture](#)

- Presentation materials

- [Solution demo video](#)
- [Project development roadmap](#)

- Additional details

- [How to run the project](#)
- [Live demo](#)

- About this template

- [Contributing](#)
- [Versioning](#)
- [Authors](#)
- [License\(MIT\)](#)
- [Acknowledgments](#)

Project Summary

The Issue we are hoping to solve

For our standard desktop PC and screen, operated over a six-year period, the annual GHG impact (aka carbon footprint, CFP) will be around 778kg CO₂e. Based on supplier data for PC models used extensively within the University, scaled to use over a six-year period, the carbon footprints of some typical working setups are:

- Desktop + screen: 621kg CO₂e
- Laptop* + screen: 691kg CO₂e (+11%)
- Desktop + 2 screens: 903kg CO₂e (+45%)
- Laptop* + screen at office + screen at home: 928kg CO₂e (+49%)
- Desktop + screen + laptop*: 1,030kg CO₂e (+66%)
- Sample PC kept 'active' continuously: 73kg CO₂e
- With default power saving features: 37kg CO₂e (-49%)
- Shutdown when not in use: 17.6kg CO₂e (-76%)
- Turned off at wall when not in use: 14.7kg CO₂e (-80%)

Not only this but Video Gaming Machines, Crypto Mining Rigs, Massive Central Servers to Embodied Carbon(CO₂ produced during manufacturing, using and disposal of electronic devices or components) is huge and growing by all means. Leaving a gigantic issue to be addressed and resolved.

How our technology solution can help

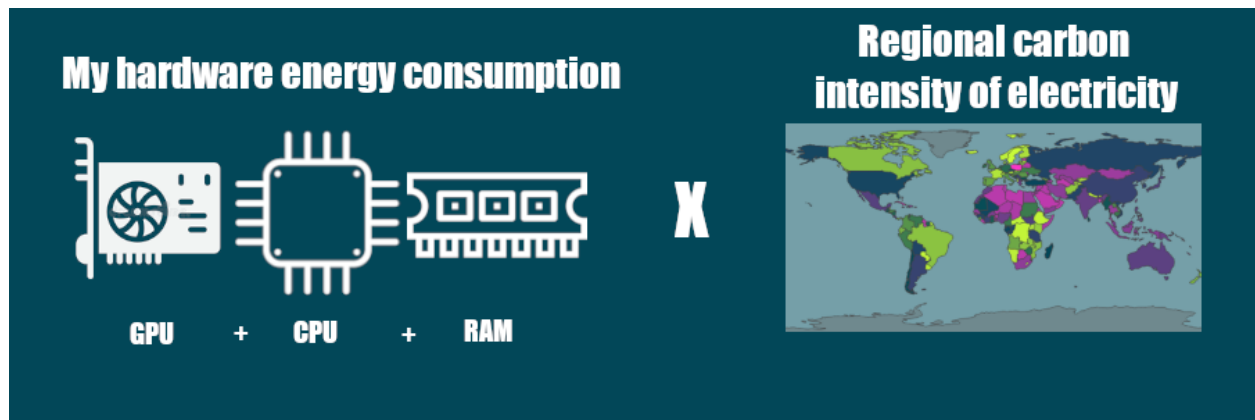
Our Solution CodeCarbon analyzes and detects the different factors which help in estimating the carbon footprint of electronic devices and suggest how to reduce it using technologically advanced solutions and well solutions based on conservative approaches.

Our Idea

CodeCarbon estimates and tracks carbon emissions from electronic devices, quantifying and analyzing their impact. We found some global data like "computing currently represents roughly 0.5% of the world's energy consumption" but nothing on our individual/organization level impact. At CodeCarbon, we believe, along with Niels Bohr, that "Nothing exists until it is measured". So we found a way to estimate how much CO₂ we produce while running our code.

How?

We created a Python package that estimates your hardware electricity power consumption (GPU + CPU + RAM) and we apply to it the carbon intensity of the region where the computing is done.



We explain more about this calculation in the following section of the documentation.

Our hope is that this package will be used widely for estimating the carbon footprint of computing, and for establishing best practices with regards to the disclosure and reduction of this footprint.

Calculations:

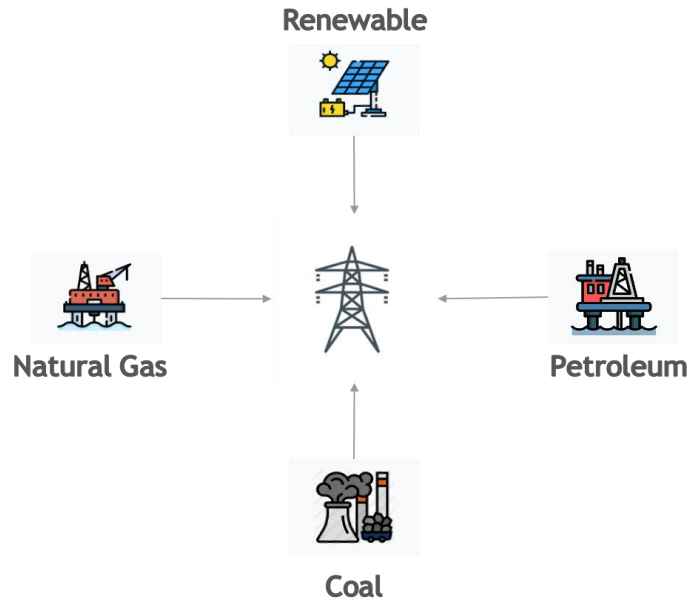
Carbon dioxide (CO₂) emissions, expressed as kilograms of CO₂-equivalents [CO₂eq], are the product of two main factors :

C = Carbon Intensity of the electricity consumed for computation: quantified as g of CO₂ emitted per kilowatt-hour of electricity.

E = Energy Consumed by the computational infrastructure: quantified as kilowatt-hours.

Carbon dioxide emissions (CO₂eq) can then be calculated as $C * E$

Carbon Intensity of the consumed electricity is calculated as a weighted average of the emissions from the different energy sources that are used to generate electricity, including fossil fuels and renewables. In this toolkit, the fossil fuels coal, petroleum, and natural gas are associated with specific carbon intensities: a known amount of carbon dioxide is emitted for each kilowatt-hour of electricity generated. Renewable or low-carbon fuels include solar power, hydroelectricity, biomass, geothermal, and more. The nearby energy grid contains a mixture of fossil fuels and low-carbon energy sources, called the Energy Mix. Based on the mix of energy sources in the local grid, this package calculates the Carbon Intensity of the electricity consumed.



When available, CodeCarbon uses global carbon intensity of electricity per cloud provider ([here](#)) or per country ([here](#)). If we don't have the global carbon intensity or electricity of a country, but we have its electricity mix, we compute the carbon intensity of electricity using this table:

S.No:	Energy Source	Carbon Intensity(Kg/MWh)
1	Coal	995
2	Petroleum	816
3	Natural Gas	743
4	GeoThermal	38
5	Hydroelectric	26
6	Nuclear	29
7	Solar	48
8	Wind	26

Sources: 1. [for fossil energies](#) 2. [for renewables energies](#)

Then, for example, if the Energy Mix of the Grid Electricity is 25% Coal, 35% Petroleum, 26% Natural Gas and 14% Nuclear:

$$\text{Net Carbon Intensity} = 0.25 * 995 + 0.35 * 816 + 0.26 * 743 + 0.14 * 29 = 731.59 \quad \text{kgCO}_2/\text{kWh}$$

If ever we have neither the global carbon intensity of a country nor its electricity mix, we apply a world average of 475 gCO₂.eq/kWh ([source](#)).

Technology Implementation

IBM AI service(s) that can be used

1. AI Factsheets - Included with Cloud Pak for Data - Used to organize and track lineage events, facts, and details for each of our machine learning model's lifecycle and increase transparency for model governance needs.
2. IBM Match 360 with Watson - Included with Cloud Pak for Data - Builds 360-degree view of our customers by matching data from disparate sources to create a single, complete, and trusted version of the truth.
3. Watson Machine learning - Included with Cloud pak for Data - Builds and trains machine learning models with tools for all skill levels. Deploy and manage models at scale.
4. Watson OpenScale - Included with Cloud Pak for Data - Infuses AI with trust and transparency. Understands how AI models make decisions to detect and mitigate bias.
5. Watson Studio - Included with Cloud Pak for Data - Unleashes the power of your data, Builds custom models and infuses your business with AI and machine learning.

Type of data

- Tabular data in delimited files or relational data in remote data sources
- Image files
- Textual (unstructured) data in documents

Type of tasks to be done

- Prepare data: cleanse, shape, visualize, organize and validate data.
- Analyze data: identify patterns and relationships in data and display insights
- Build models: build, train, test, and deploy models to make predictions or optimize decisions.

Automation required

- Code editor tools: Used to write code in Python, R or also spark
- Graphical builder tools: Use menus and drag and drop functionally on a builder to visually program
- Automated builder tools: Used to configure automated tasks that require limited user input.

Right tools used (Tabular or relational data).

- Jupyter notebook editor [code editor]{Prepare & Analyze data, Build models}

Use the Jupyter notebook editor to create a notebook in which you run code to prepare, visualize, and analyze data, or build and train a model.

Data format

Any

Data size

Any

How you can prepare data, analyze data, or build models

Write code in Python or R, all also with Spark.

Include rich text and media with your code.

Work with any kind of data in any way you want.

Use preinstalled or install other open source and IBM libraries and packages.

Schedule runs of your code

Import a notebook from a file or a URL.

Share read-only copies of your notebook externally.

Get started

To create a notebook, click New asset > Jupyter notebook editor.

Learn more

[Documentation about notebooks](#)

- JupyterLab [code editor]{Prepare & Analyze data, Build models}

Use the JupyterLab IDE to create a notebook or Python script in which you run code to prepare, visualize, and analyze data, or build and train a model. JupyterLab is integrated with a Git repository which must be associated with the project.

Data format

Any

Data size

Any

How you can prepare data, analyze data, or build models

Write code in Python.

Include rich text and media with your code.

Work with any kind of data in any way you want.

Use preinstalled or install other open source and IBM libraries and packages.

Import a notebook from a file.

Share your notebook or script in a Git repository.

Get started

To use JupyterLab, click Launch IDE > JupyterLab.

Learn more

[Documentation about JupyterLab](#)

- *Decision Optimization model builder [Graphical builder and code editor]{Prepare data, Build models}*

Use Decision Optimization to build and run optimization models in the Decision Optimization modeler or in a Jupyter notebook.

Required service

Decision Optimization

Data formats

Tabular: CSV files

Data size

Any

How you can prepare data

Import relevant data into a scenario and edit it.

How you can build models

Build prescriptive decision optimization models.

Create, import and edit models in Python DOpplex, OPL or with natural language expressions.

Create, import and edit models in notebooks.

How you can solve models

Run and solve decision optimization models using CPLEX engines.

Investigate and compare solutions for multiple scenarios.

Create tables, charts and notes to visualize data and solutions for one or more scenarios.

Get started

To create a Decision Optimization model, click New asset > Decision Optimization, or for notebooks click New asset > Jupyter notebook editor.

Learn more

[Documentation about Decision Optimization](#)

- *IBM Match 360 with Watson [Automated builder]{Prepare data}*

Use IBM Match 360 with Watson to create master data entities that represent digital twins of your customers. Model and map your data, then run the matching algorithm to create master data entities. Customize and tune your matching algorithm to meet your organization's requirements.

Required services

IBM Match 360 with Watson IBM Watson Knowledge Catalog

Data size

Any

How you can prepare data

Model and map data from sources across your organization.

Run the customizable matching algorithm to create master data entities.

View and edit master data entities and their associated records.

Get started

To create an IBM Match 360 configuration asset, click New asset > Master data configuration.

Learn more

[Documentation about IBM Match 360 with Watson](#)

- *Dashboard editor [Graphical builder]{Analyze data}*

Use the Dashboard editor to create a set of visualizations of analytical results on a graphical builder.

Required service

Cognos Dashboard

Data format

Tabular: CSV files

Relational: Tables in some relational data sources

Data size

Any size

How you can analyze data

Create graphs without coding.

Include text, media, web pages, images, and shapes in your dashboard.

Get started

To create a dashboard, click New asset > Dashboard editor. The *Dashboard editor* tile is in the Graphical builders section.

Learn more

[Documentation about dashboards](#)

- Watson Pipelines [Graphical builder]{Prepare and Analyze data, Build models}

Use the Pipelines canvas editor to create a flow to prepare, visualize, and analyze data, or build and train a model.

Required service

Watson Knowledge Catalog or Watson Studio

Data format

Any

Data size

Any

How you can prepare data, analyze data, or build models

Use a variety of nodes that each contain their own logs.

Incorporate notebooks into the flow to run any Python or R code.

Work with any kind of data in any way you want.

Schedule runs of your flow.

Import data from your mounted PVC, project, or ingest data from Github.

Create your custom component with a Python code.

Conditionalize your pipelines to monitor data quality however you want.

Use webhook to send emails or messages to keep up to date on the status of your flow.

Get started

To create a new pipeline, click New asset > Pipelines.

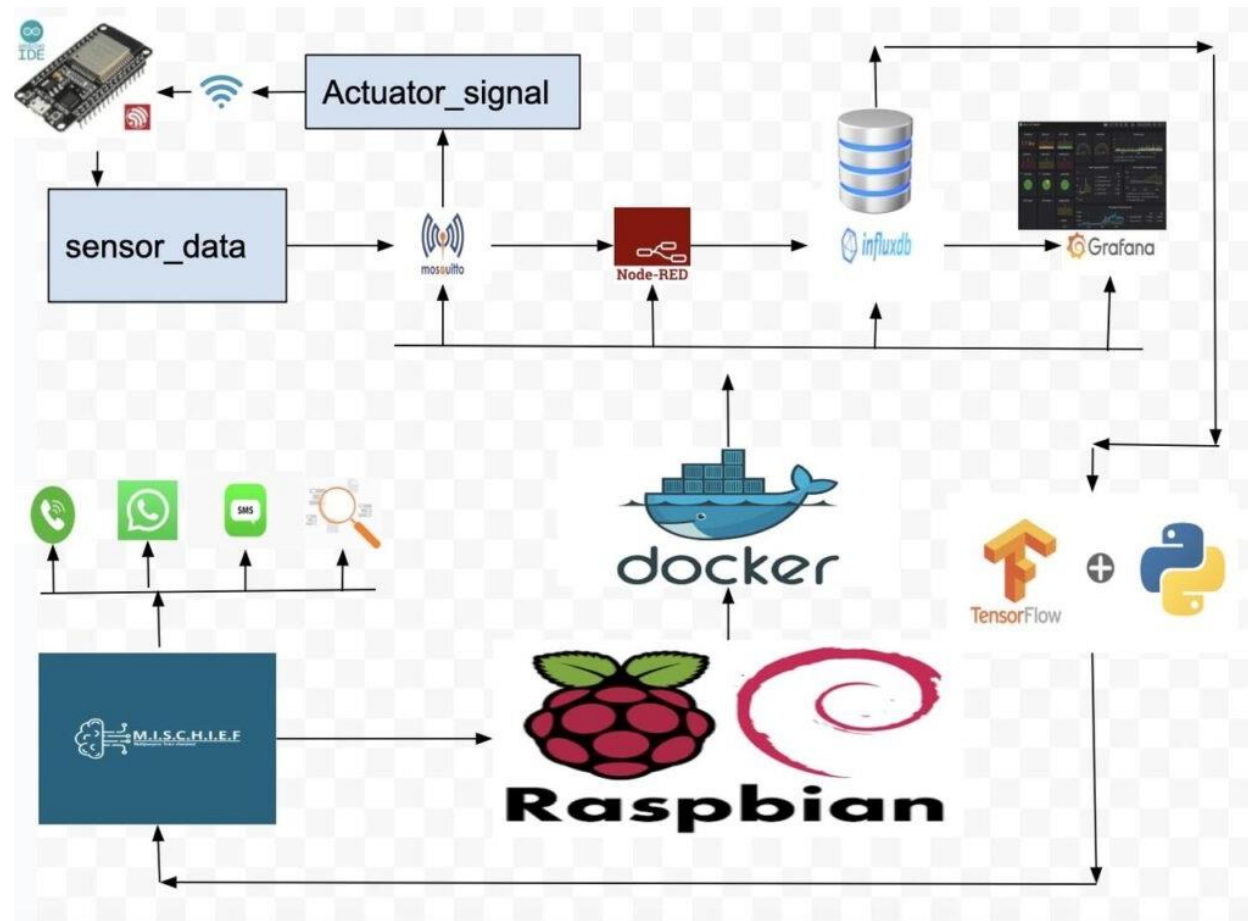
Parent topic: [Projects](#)

Other IBM technology used

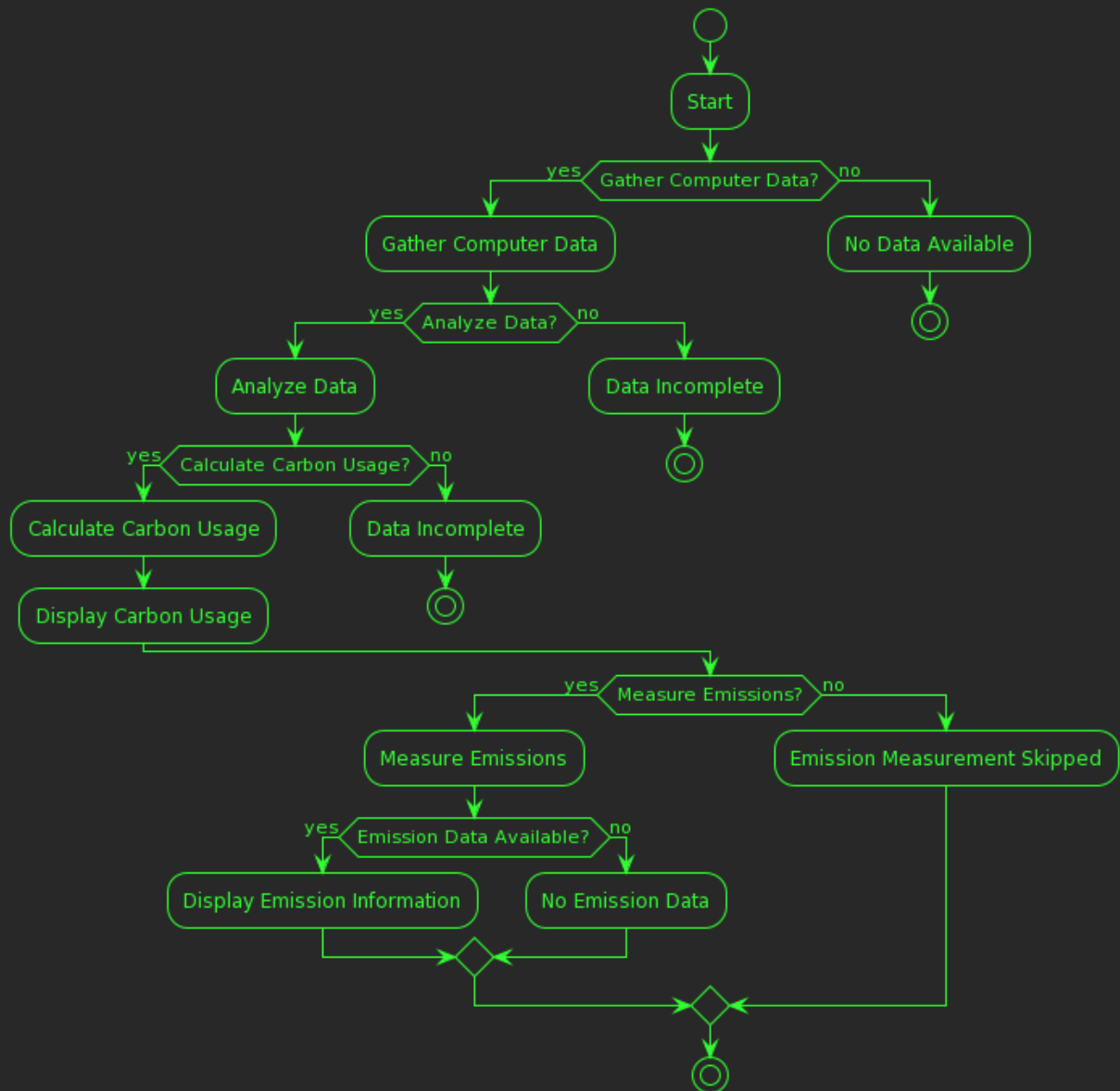
1. Nodered

Solution Architecture

Reference



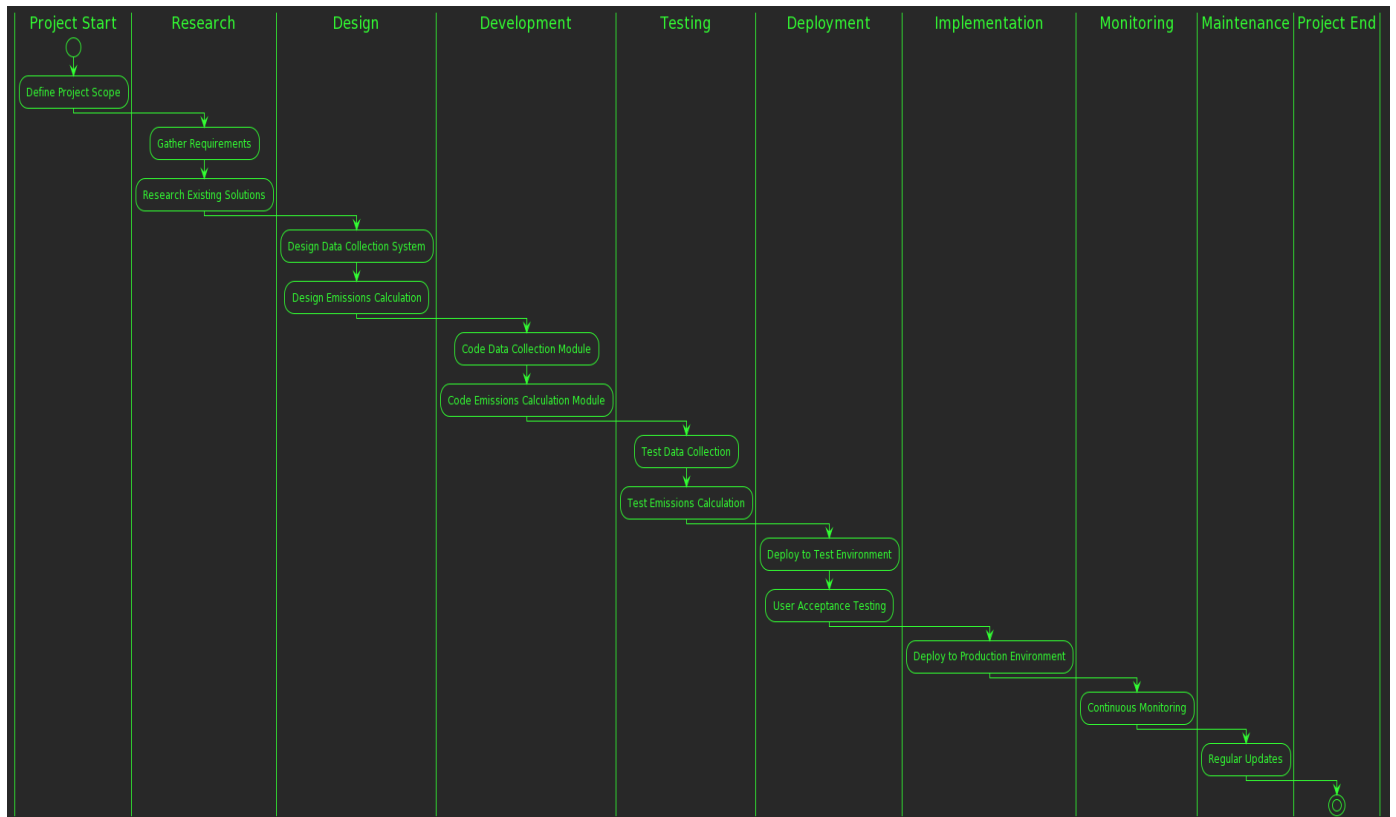
Carbon Usage Calculator and Emissions Measurement



Presentation materials

Solution demo video

Project development roadmap



Additional details

How to run the project

- 1: `pip install codecarbon`
- 2: `conda install -c conda-forge codecarbon`
- 3: `pip install codecarbon`
- 4: `conda install -c codecarbon -c conda-forge codecarbon`
- 5: `conda create --name codecarbon`
- 6: `conda activate codecarbon`
- 7: `codecarbon init`
- 8 : `conda install -c codecarbon -c conda-forge codecarbon`
- 9: `codecarbon init`
- 10 : `codecarbon monitor`

Live demo

We are developing Project CodeCarbon to analyze and estimate the carbon content and emissions from technological devices and systems.

Much as technology has seeped into all aspects of our lives, the costs that come with the use of these devices is growing exponentially. To add context, carbon emissions from devices and systems is

Licence: MIT