

Analysing the Commitment of US Technology Companies Towards ESG Goals.

Leon Chipchase

Department of Computer Science

University of Warwick

Supervised by Yu Guan

Year of Study: 3rd

2 May 2023

Acknowledgements

First, I would like to thank my project supervisor, Dr Yu Guan, for his support and guidance over the last two terms and for his accommodation of my circumstances. Second, I would also like to thank Giles Palmer for helping me conceive the project and pointing me toward a feasible one. Finally, my family for providing me with the resources required for the project, and my friends for testing and providing feedback on the system.

Abstract

Climate change has emerged as a pressing issue in the last century. Kick-started by the industrial revolution, the environmental impact of corporations is an issue of particular importance; with millions of employees and revenues larger than entire countries, monitoring and regulating their footprint is imperative. The first of many steps to achieve this is finding a way to extract and measure their emissions data for a clearer insight.

In this project, we aim to create a system that extracts the desired environmental indicators from a company's environmental report, stores that data, and produces a report containing visual info-graphics to present this information clearly and concisely to the user. The process should be affordable, robust and repeatable, allowing individuals without corporate resources to use it and gain insights. We analyse existing solutions and the current issues with those. And examine the process developed to achieve these goals. Finally, we assess the developed approach's effectiveness, limitations, and potential future improvements.

Keywords

Environment, Key Performance Indicator, PDF analysis, Table Extraction, ESG Reporting, Corporate Responsibility

Contents

1	Introduction	8
1.1	Motivation	9
1.2	Client	9
1.3	Project Scope	10
2	Background	11
2.1	Environmental, Social and Corporate Governance (ESG) Reporting	11
2.2	Existing Solutions	12
2.2.1	Bloomberg Terminal	12
2.2.2	Chat-GPT	13
2.2.3	IdealRatings ESG Metric Dataset	14
2.2.4	Summary	15
3	The Problem	16
3.1	Finding and Collecting Data	16
3.2	Identifying the Relevant Data	17

Analysing the Commitment of US Technology Companies Towards ESG Goals. 5

- 3.2.1 Scope Reduction 18
- 3.2.2 Chosen KPIs 18
- 3.3 Extracting the Identified Data 22
 - 3.3.1 Table Extraction 23
 - 3.3.2 Identifying Relevant Tables 23
 - 3.3.3 KPI Matching 24
 - 3.3.4 Data Extraction 24
- 3.4 Presenting the Extracted Data to the user 24
 - 3.4.1 PDF Reporting 25
 - 3.4.2 Interactive Reports Through a User Interface 25
- 3.5 Legal and Ethical Considerations 26
 - 3.5.1 Ethical Consideration in Evaluation 27

4 Requirements 28

5 Design 33

- 5.1 Software Stack 33
 - 5.1.1 Front-End 33
 - 5.1.2 Back-End and Key Libraries 35
 - 5.1.3 Database 39
- 5.2 System Design 40
 - 5.2.1 Table Extraction 41
 - 5.2.2 Data Extraction 42

5.2.3	Report Generation	44
5.2.4	Database Design	44
6	Implementation	47
6.1	Research Methods	47
6.2	Extracting Identified Data	48
6.2.1	Table Extraction	48
6.2.2	Identifying Relevant Tables	49
6.2.3	KPI Matching	49
6.2.4	Data Extraction	51
6.3	Data Representation	55
6.3.1	PDF Reporting	55
6.3.2	Graphing Results	56
6.3.3	Company Ranking	63
7	Project Management	67
7.1	Methodology	67
7.2	Source Control	68
7.3	Timeline	69
7.3.1	Original Timeline	69
7.3.2	Actual Timeline	71
7.4	Risk Assessment	72

- 8 Evaluation 75**
 - 8.1 Evaluation of the developed process 75
 - 8.1.1 Table Extraction 76
 - 8.1.2 KPI Matching and the Learning Process 79
 - 8.1.3 Data Extraction 86
 - 8.1.4 Report Generation 91
 - 8.1.5 Conclusion 96
 - 8.2 Requirements Success 97
 - 8.3 Findings 102
 - 8.4 Limitations 103

- 9 Conclusions 104**
 - 9.1 Future work 105
 - 9.2 Authors Assessment 107

Chapter 1

Introduction

Significant progress has been made in the last decade towards global environmental goals. This is an effort which requires collaboration from countries and corporations worldwide. Despite the recent shift in awareness and progress, this is still far off from what is required to reach the goals of the Paris Agreement, such as "keeping a global temperature rise this century well below 2 degrees Celsius above pre-industrial levels" (United-Nations).

This is particularly prevalent in the corporate world, where at the date of the project conception (September 2022), there were no strict environmental, social and corporate governance reporting guidelines for public companies in the United States. This means the figures reported by different companies could have been clearer regarding the reporting format and KPIs. Therefore, getting a clear idea about the raw environmental impact of a particular company is very difficult for someone without the resources of a corporate client due to the astronomical costs of the existing solutions.

Therefore, the goal was to create a platform or system which would make analysing the impact of a company a robust, repeatable process accessible to the average person and research a method on how to do this.

1.1 Motivation

My motivation for this project stemmed initially from my family, who have always held sustainability in high regard. However, I wanted to combine this with a topic relevant to myself and my career - 'Big Tech', the field I intend to work in after my graduation; therefore wanted a deeper understanding of the companies I could work for (Chipchase, 2022).

The timing of this project aligns with the SEC ruling for the 2023 Fiscal year, where it could become a requirement for all publicly listed US companies to produce their ESG/Corporate Responsibility reports (Beardsley, 2023). This will then act as a tool to determine whether those companies are working towards those goals effectively.

1.2 Client

The intended client for this project would be users who cannot afford the excessive price tag of over \$10,000 for the existing solutions such as Bloomberg Terminal (Bloomberg, 2023). In addition, this would include anyone who requires more in-depth insight into companies' environmental impact, whether for professional, informative, or personal reasons. This is not intended to be

a financial tool for evaluating the company's success towards ESG goals for investment decisions.

1.3 Project Scope

This project's scope will be limited to the top 100 publicly listed technology companies by market capitalisation (listed on the NASDAQ and the NYSE). This is because companies in the same country are likely to have a similar reporting structure in reports and figures published. Furthermore, only the top 100 companies were considered since some companies did not publish the required information at the bottom end of the list.

Initially, the project scope included environmental, social and corporate governance data; however, this was later limited to purely environmental data due to the significant inconsistencies in the reporting standards for the social and corporate governance data. Moreover, the added difficulty of analysing non-numerical data meant that an entirely different approach would be required for the analysis.

Chapter 2

Background

2.1 Environmental, Social and Corporate Governance (ESG) Reporting

ESG reporting, or environmental, Social, and Governance reporting, refers to disclosing a company's non-financial performance on environmental, social, and governance issues to its stakeholders. In the United States, the guidelines for reporting are currently voluntary; this makes it difficult for stakeholders to compare the performance of different companies against each other.

Having multiple ESG frameworks makes ESG reporting difficult (Tocchini and Cafagna, 2022) and causes issues for both companies and investors. As well as this, it prevents a company from following a clear goal to reduce its impact.

The Securities and Exchange Commission (SEC) recently announced a change that it would implement new regulations for publicly listed companies in ef-

fect in the 2023 fiscal year. These regulations will require companies to report climate-related risks and greenhouse gas emissions (environmental data), board diversity data, and human metrics (social and governance data). As well as this, they will be required to implement set goals and plans to achieve those goals (SEC, 2022).

Moving to a consistent set of standards will hopefully increase company transparency and accountability. This will hopefully come with an impactful change towards a more sustainable future.

2.2 Existing Solutions

Due to the significant focus on the environment in recent years, particularly regarding sustainable investing, a wide range of available financial tools can provide this data with in-depth insights and analysis about what the figures mean. However, due to the economic nature of the tools, these are often only available to corporate clients or institutions due to the price tag associated with each product.

2.2.1 Bloomberg Terminal

Bloomberg Terminal is a tool that provides users access to a vast amount of financial data. However, one of the areas in which it excels is its ability to analyse and extract environmental data.

With the increasing importance of environmental, social, and governance (ESG) factors, the Terminal provides users with a wide range of ESG metrics. For example, the platform can pull information on KPIs such as carbon footprint, water usage, energy consumption, and other critical environmental metrics. The Terminal also enables users to compare companies within a given sector or industry and across different regions or countries. Furthermore, it offers a range of analytical tools to help 'explain' the data presented. These include charts, graphs, and reports that can be tailored to specific metrics. Bloomberg Terminal is a powerful resource. With the wide range of data and analytical tools available, the platform can help users gain an in-depth understanding of the impact of each company they would like to consider.

This is unsuitable because a subscription to the Terminal comes with a cost of \$24,000 per seat per year, far outside of the range suitable for an everyday user. And when we contacted support stating my intentions as a student to try a demo, no contact or email was returned.

2.2.2 Chat-GPT

After the conception of the original project, Chat-GPT was released. This provided a powerful tool which could be used to query data such as an ESG report and extract a set of figures from that report, as well as an analysis of what that data means.

We opted against incorporating Chat-GPT into the project due to the language model's complex nature. It is far beyond the scope of understanding of any

single person, but possibly the team members behind the model. This means that when querying more complex tables or data, there will be little or no understanding of the process of that extraction, as well as why those are the correct figures.

Example output from using GPT-4 (OpenAI, 2023) queries in the format of: "Treating the following text as a table in CSV format. The table contains a set of KPIs in the first column; then, each year has a value for that indicator. Output the data in the same format, however, only extracting from the following set of KPIs: 'KPI_LIST'. The table to analyse is as follows: 'TABLE'. This query works both for Chat-GPT and GPT-4.

KPI	2017	2018	2019	2020	2021	Unit
Scope 1	66,549	63,521	66,686	38,694	45,073	tCO ₂ e
Scope 2 Market Based	509,334	684,236	794,267	911,415	1,823,132	tCO ₂ e
Scope 2 Location Based	3,301,392	4,344,686	5,116,949	5,865,095	6,576,239	tCO ₂ e
Scope 3	2,719,024	12,900,467	11,669,000	9,376,000	9,503,000	tCO ₂ e
Total Scoped Emissions	3,294,907	13,648,224	12,529,953	10,326,109	11,371,205	tCO ₂ e

Figure 2.1: Example GPT-4 Output.

2.2.3 IdealRatings ESG Metric Dataset

Furthermore, there are datasets which contain vast amounts of information about a set of companies. An example of one which provides a comprehensive

set of ESG metrics is the IdealRatings ESG Metric Dataset (Catania and Keefer, 2022). This would be ideal for the analysis of companies worldwide, providing all information required to provide an in-depth assessment. However, once again, this comes with a cost of \$25,000 for 12 months of access. This is more than the Bloomberg terminal, without providing the capability of report generation. Once again far outside the scope of cost suitable for a non-corporate client.

2.2.4 Summary

Despite the range of different solutions available, these are very rarely accessible to the everyday user. This means that finding more in-depth information on the performance of these companies is often unfeasible due to the time taken to analyse a report manually for a set of companies. Therefore, this shows the value provided by this system to users who do not have the resources of a corporate client available to them.

Chapter 3

The Problem

This project consisted of four main stages and challenges to overcome. Solving these problems was an iterative process of research, trial and error and discussion with my supervisor. Through these meetings, which occurred fortnightly during term time, we could effectively find the tools required to overcome these challenges systematically.

3.1 Finding and Collecting Data

Due to the high costs of purchasing a dataset (often in the price ranges of \$10,000 or more, another source of data was required, ideally free and accessible. Even though the companies were not required by law to publish their ESG reports (VinciWorks, 2022), many did due to the global exposure and the large market capitalisation of the companies; therefore, producing these figures is important to the shareholders and potential investors.

To collect each of these reports, we initially searched for an API that could

scrape these ESG reports from the company websites. However, once again, these often came with a price tag outside an affordable price range, particularly for a university student. Therefore we opted for manually downloading the report from the respective company website. However, some companies did not publish these reports, particularly the smaller cap¹ companies at the bottom end of the list.

3.2 Identifying the Relevant Data

ESG data is split into three different sections (Delubac, 2023),

1. **Environmental:** "Environmental factors involve considerations of an organization's overall impact on the environment and the potential risks and opportunities it faces because of environmental issues, such as climate change and measures to protect natural resources" (Mathis, 2023).
2. **Social:** "Social factors address how a company treats different groups of people – employees, suppliers, customers, community members and more" (Mathis, 2023).
3. **Corporate Governance:** "Governance factors examine how a company polices itself, focusing on internal controls and practices to maintain compliance with regulations, industry best practices and corporate policies" (Mathis, 2023).

Each of these sections consists of a set of KPIs (Key Performance Indicators)²

¹Companies with a lower total market value

²KPI: Short for key performance indicator, a measurable and quantifiable metric used to track progress towards a specific goal or objective (Klipfolio, 2023).

published by the company. Due to the lack of consistent guidelines present, the indicators that each company publishes may vary significantly.

Therefore, the challenge here was identifying the data published most ubiquitously across the entire set of companies, as well as identifying the KPIs which gave the most impactful representation of the impact of a particular company. To identify these, we read through a large set of reports recording which KPIs were most commonly present, as well as domain research online and through different environmental websites to identify the most impactful set.

3.2.1 Scope Reduction

Initially, the aim was to include KPIs about the company's environmental, social and corporate governance impact. However, after analysing a large set of reports, we identified that many of the reports would only publish their environmental data in a tabular format alongside the respective KPI. In addition, typically, social and corporate governance data was typically spread out in large chunks of text throughout the document. Therefore, identifying the social and corporate governance data would require an entirely different approach. Therefore, only environmental KPIs were included.

3.2.2 Chosen KPIs

The table below 3.1 shows the chosen set of KPIs, along with the unit and the description for each. Even though these were most commonly reported by the

entire collection of companies, in some reports, a subset of these may still have been unavailable or unpublished, particularly with the smaller companies.

KPI	Unit	Description
Carbon Intensity Per FTE Employee	tCO ₂ e/ FTE	"A measure of carbon dioxide and other greenhouse gases (CO ₂ e) per"(Chevron Policy and Affairs, 2022) Full Time Equivalent (all employee hours calculated to the number of full-time workers).
Carbon Intensity Per Megawatt-Hour of Energy	tCO ₂ e/MWh	"A measure of carbon dioxide and other greenhouse gases (CO ₂ e) per" (Chevron Policy and Affairs, 2022) Mega-Watt hour of energy
Carbon Intensity Per Unit of Revenue	tCO ₂ e/ Million USD (\$)	"A measure of carbon dioxide and other greenhouse gases (CO ₂ e) per" (Chevron Policy and Affairs, 2022) Million USD of revenue

Landfill Diversion Rate (%)	%	The proportion of waste not sent to landfill(ReWorksSA, 2023).
Proportion of Renewable Electricity Used/Purchased (%)	%	The proportion of the electricity used or purchased by the company that is from renewable sources.
Proportion of Renewable Energy Used/Purchased (%)	%	The proportion of the energy used or purchased by the company that is from renewable sources.
Scope 1	tCO ₂ e	"Direct greenhouse (GHG) emissions that occur from sources that are controlled or owned by an organization" (EPA, 2022)
Scope 2 - Market Based	tCO ₂ e	A measure of "emissions based on the electricity that organizations have chosen to purchase, often spelled out in contracts or instruments" (Bock, 2023).

Scope 2 - Location Based	tCO2e	A measure of "emissions based on the emissions intensity of the local grid area where the electricity usage occurs" (Bock, 2023).
Scope 3	tCO2e	A measure of "all other indirect emissions that occur in the upstream and downstream activities of an organisation" Trust (2023).
Total Electricity Used/Purchased	MWh	The total electricity used or purchased by the company.
Total Energy Used/Purchased	MWh	The total energy used or purchased by the company.
Total Renewable Electricity Used/Purchased	MWh	The total electricity used or purchased by the company from renewable sources.
Total Renewable Energy Used/Purchased	MWh	The total energy used or purchased by the company from renewable sources.
Waste Generated	Million Tonnes	The total waste generated by the company.

Water Consumption	Million Gallons	A measure of "the portion of water use that is not returned to the original water source after being withdrawn" (Reig, 2013).
Water Discharge	Million Gallons	The measure of "water discharges from agricultural, industrial, and commercial operations, wastewater treatment plants, or residential properties" (Insider, 2023).
Water Withdrawal	Million Gallons	A measure of "freshwater taken from ground or surface water sources, either permanently or temporarily, and conveyed to a place of use" (OECD, 2023).

Table 3.1: Chosen Key Performance Indicators (KPIs)

3.3 Extracting the Identified Data

After identifying the desired set of key performance indicators, the next challenge was extracting this data from the document. We spotted that in each

report, the published metrics were present in a table at the end of the paper. This meant that if we could extract the set of tables from the report, we could narrow the tables to those containing relevant information and match the KPIs in the tables to the KPIs we wanted to store.

3.3.1 Table Extraction

The problem with the tabular data in PDFs is that the format is often inconsistent (Timalsina, 2023); therefore, just analysing the PDF meta-data was insufficient. Initially, we looked at using the library PyMuPDF (PyMuPDF, 2023) to analyse the pdf and extract the tabular data in this manner. However, this did not work and often resulted in significant inconsistencies and incorrect data.

Therefore another method was required. Two main options were considered, Microsoft Excel and ExtractTable. This would allow the extraction of the tabular data from the report and return the data to be analysed in Python.

3.3.2 Identifying Relevant Tables

After the set of tables has been extracted from the report, these need to be narrowed down to determine which tables contain the relevant information. For this, we opted for an 'ask-the-user' process, where the user could examine the tables once they were returned to identify which tables contained the required information. Of course, this requires the user to have some knowledge as to what information is required.

3.3.3 KPI Matching

Due to the lack of reporting standards, there is a significant deviation from company to company with the exact way the figures are reported. A key factor was that for almost every company, the set of KPIs in the tables are spelt or worded differently. Given the requirement for 'fuzzy' string matching and a learning process that, over time, will become more automated as it learns the different variations for each representation of each KPI.

Once the table has been identified, the KPIs must be identified before inserting any information into the database.

3.3.4 Data Extraction

After the KPIs have been matched correctly, this leaves us with another challenge, which is extracting that numerical data from the resultant table to be stored for future use. A set of processing steps must be done on the table from the report. Including identifying which row contains the corresponding year and removing any unnecessary fields or information.

3.4 Presenting the Extracted Data to the user

Visualisation tools, such as graphs, are essential for enhancing the understanding of data. They enable clear communication of complex information by transforming it into a visual, easy-to-understand format (Few, 2009). By presenting the extracted data visually, the graphs allow for pattern recognition, allowing

the reader to clearly understand trends and relationships that may not have been uncovered by looking at the raw tabular data (Munzner, 2015).

This is particularly true with larger volumes of data, where humans are easily overwhelmed by large sets of numbers and consequently unable to comprehend what is presented to them effectively (Hiem and Keil, 2017). For this reason, the effective use of figures is instrumental as an aide in helping convey information to the user.

For the reasons stated, it was imperative that once all the relevant data had been collected, it was presented to the user clearly and concisely. Therefore, two main options were considered in this scenario, generating a PDF report for the user to read or an interactive User Interface.

3.4.1 PDF Reporting

While this option was considered, generating a clear PDF report for a user to read is a very complex, time-consuming task due to the difficulty of generating a PDF with Python code. Additionally, the focus on interactivity and user engagement meant this option was not chosen.

3.4.2 Interactive Reports Through a User Interface

This option was chosen because of the simplicity of using client-side JavaScript libraries to create interactive graphs, allowing the user to engage directly with the graph shown and the reports created, such as toggling different KPIs

within the graph and hovering over the points to view the exact numerical figure.

Furthermore, suppose a pdf of the report was required; the print web page functionality is provided by all mainstream browsers (Google, Safari, Edge). In that case, this allows the user to create a pdf for each company and interact directly with the report online.

3.5 Legal and Ethical Considerations

Since the data is being extracted from reports which belong to various companies, there are some legal and ethical considerations to be taken into account.

Firstly, the responsibility of producing the correct data. Therefore, there will be a disclaimer stating that there is a chance the information extracted may be incorrect and also no longer up to date - if there is a new report produced, for example.

Secondly, there is the consideration of transparency on how the data is extracted. However, since the user is present for these steps, this issue is removed.

Additionally, there is the case that a report contains any confidential information. If this were the case, this data should not be extracted and reported; however, since the reports are made public on their website, this is not the case. Furthermore, if the report contains personal data, this is important to

consider for GDPR regulations. However, since the tables extracted are for the environmental data and do not explicitly provide private personal information, this problem is alleviated.

Finally, there is the issue of the source of the data. Since the data is collected from the ESG reports from that company, this identifies the source of the data. Additionally, because the user themselves is extracting the data from the report rather than the system distributing information, the source of the data is clear.

In conclusion, whilst there are legal considerations regarding the data extracted, the considerations and actions required are minimal. Furthermore, due to the nature of this project being academic, the user of the ESG reports falls under 'fair use' or 'fair dealing'. However, regardless it is important to note that the extracted data 'Could' be incorrect to remove any liability.

3.5.1 Ethical Consideration in Evaluation

To evaluate the success of some of the components of the system, a survey was done. This was for friends and family who would feedback on the system. All questions were done with 'Google Forms'; no names or personal details were collected and all submissions were anonymous.

Chapter 4

Requirements

"Requirements gathering in Information Systems is a critical part of any project" (Lane et al., 2016), therefore defining these was a key pre-requisite to development. However, due to the exploratory nature of the project I was carrying out, the initial requirements were very flexible and subject to change throughout the project's lifetime. The key reason is that the project consisted of a large set of initial challenges and questions that needed to be overcome to produce a working result.

Based on the problem outlined in chapter 3, a set of requirements has been created, split into different categories of urgency. These follow the MoSCoW (Must, Should, Could, Won't) format. This will clearly define the project's scope into different parts and requirements that it has been split into, and "allows the delivery of the Minimum Usable Subset of requirements to be guaranteed" (Craddock and Craddock, 2014). This set of requirements is outlined in the table below 4.1.

Requirement	Urgency	Justification
<p>R1: Define a set of environmental KPIs that will be extracted and measured from each report.</p>	<p>M</p>	<p>Environmental KPIs consist of numerical data, allowing for a clear analysis and reporting through graphs and tables.</p>
<p>R2: Define a set of social KPIs that will be extracted and measured from each report</p>	<p>W</p>	<p>Measuring social KPIs allows for another angle of evaluation of a company regarding its impact on its social environment such as the communities it operates in.</p>
<p>R3: Define a set of corporate governance KPIs that will be extracted and measured from each report</p>	<p>C</p>	<p>Measuring governance KPIs gives a better outline on the running of the company itself, as well as the equality within its leadership.</p>
<p>R4: The system must be able to identify and extract the tabular data from the reports.</p>	<p>M</p>	<p>All ESG reports analysed contain a set of tables with data containing the figures for their KPIs. Therefore, identifying and extracting these is integral to producing meaningful report findings.</p>

<p>R5: An ask-the-user system must be implemented to identify the tables containing relevant data.</p>	<p>M</p>	<p>Once the set of tables has been extracted, the user manually identifies the relevant tables. This allows the system to extract the KPIs from that table then and ensures the user can see that the table to be analysed is correct.</p>
<p>R6: The system must be able to identify tables containing relevant data without user oversight.</p>	<p>C</p>	<p>This would streamline the process of extracting the data from the reports by removing a need for user input in this section of the system.</p>
<p>R7: The system must identify KPIs within the table and match those to the KPIs being measured.</p>	<p>M</p>	<p>This allows the system to identify which KPI the field in the table represents, identifying which KPI must be inserted into the database.</p>
<p>R8: The system must be able to learn different KPIs representations through the user's assistance.</p>	<p>M</p>	<p>This allows the system to streamline its process and requires less user input over time, increasing the system's efficiency and user experience over time.</p>

<p>R9: Once the KPIs have been matched, the system must be able to identify the corresponding values and years for each.</p>	<p>M</p>	<p>This allows the system to correlate the KPI to the correct year and correct value to be inserted into the database as a data point for that company.</p>
<p>R10: Once the data has been collected, the data must be presented to the user in a report.</p>	<p>M</p>	<p>This allows the data collected to provide meaningful insight to the user.</p>
<p>R11: A set of graphs must be presented to outline the KPI performance changes over time for a specific company.</p>	<p>M</p>	<p>Graphs and other visual aides significantly improve user retention and understanding when displaying and describing data (Bynes, 2023).</p>
<p>R12: The graphs must be interactive, allowing the user to toggle different series on and off to focus on another.</p>	<p>S</p>	<p>Providing a user with an interactive experience significantly increases attention retention, particularly in a world with social media, where "nearly three-quarters of markets rely on visuals" (Conner, 2017), therefore users are used to this kind of information.</p>

R13: A table must show all the collected data from a company.	M	Displaying the raw data extracted from the system for the company allows the user to gain an overview of the data collected.
R14: There must be a table showing the company's ranking for each of its KPIs.	M	Giving each of the KPIs a ranking puts the figures themselves into perspective for the user allowing for a better understanding of the figures.
R15: An overall rank must be calculated for the company compared to others in the set measured.	S	Giving an overall rank for a company indicates how the company is performing compared with the rest of the cohort.
R16: The system must be able to compare different companies side by side to compare their performance against each other.	C	Allowing a company by company allows a user to identify key differences between two companies.

Table 4.1: Requirements Table

Chapter 5

Design

5.1 Software Stack

The software stack of the project consisted of the following:

1. **Front-End:** The front end of the system consisted of HTML, JavaScript
2. **Back-End:** Python - Flask
3. **Database:** PostgreSQL

5.1.1 Front-End

HTML (Hypertext Markup Language), JavaScript and CSS (Cascading Style Sheet) are the global standard for front-end development. Integrating seamlessly to create interactive user interfaces (UIs) and ensuring a smooth user experience (UX). Additionally, the combination with the Bootstrap framework allows for a much simpler development process for interactive UIs, allowing for web applications to operate effectively with both desktop and mobile users.

Data Exchange Between the Front-End and Flask

Communication between the Front-End and Flask relies predominantly on HTTP requests and responses (Grinberg, 2014), with JavaScript often used for handling any asynchronous calls via AJAX (Asynchronous JavaScript and XML). Flask applications rely on routes and view functions to handle incoming requests, sending and receiving data within the application in various formats; typically, JSON (JavaScript Object Notation) allows the data from the Back-End to be presented to the user seamlessly (Echua, 2017).

Bootstrap

Bootstrap (Bootstrap, 2023) is used to create reactive, mobile-compatible sites quickly and easily. Allowing for a significantly reduced development time when making an effective UI. Furthermore, a wide range of resources online, such as Bootstrap documentation and online sites with tutorials, are available for learning the framework quickly and easily.

ChartJS

After the data was collected, a critical next part of the system was to present the collected data to the user. We chose the library ChartJS (ChartJS, 2023) for this task. The reasons for this were:

1. **Ease Of Integration:** Given two arrays of data, we could easily select the type of graph or plot required, with no additional data preprocessing needed on the Front-End of the system.
2. **Vast Range of Choice:** ChartJS offers a wide range of different options of

graphs to be used (in the 100s) for various other types of data representation.

3. **Customisation:** Each of the different graphs allowed for a range of customisation options, including but not limited to graph titles, axes labels, and adding a range of other series and colours.
4. **Interactivity:** One key feature which stood out was the ability to interact with each graph, such as toggling a particular series allowing you to focus on another and hovering over the points in the chart to view their exact value. Combining these two features gave the user a fully immersive experience with the presented data.

5.1.2 Back-End and Key Libraries

Flask

Flask is an ideal choice for developing small web applications. Built on top of Python, Flask allows a bare-bones approach to web development, allowing developers to quickly build, deploy and maintain web applications (Grinberg, 2018). This was key for the project since the UI development was only a small fraction of the task, where the main focus was extracting this data for the user.

Firstly, Flask's lightweight nature is ideal for this task. It neither imposes strict conventions nor includes many built-in components, reducing the project's overall complexity (Flask, 2023). Once again, a key requirement for this project where the main focus is extracting the data and then presenting the data to the user, neither of which Flask was involved in directly.

ExtractTable

To extract the tabular data from the report, the chosen library was ExtractTable (ExtractTable, 2023). This is for a few reasons, however, predominantly the ease of integration with the rest of the system. It is accessible via API through its Python library and its website. Within the API, a PDF file is passed to the API then the set of tables is returned as a list of Pandas DataFrames. This allows the conversion to a numpy array for further processing and analysis. Furthermore, the system's accuracy was the highest found, providing excellent results when tested with images and PDF files - the format used for this project.

There was, however, a drawback; this was the associated cost of access to the library. However, this was still very reasonable, allowing the processing of 10,000 pages for the cost of \$0.027 per page, allowing the analysis of hundreds of company reports, more than what is required for the set of companies considered for this project.

Additional Options

Another option considered was Microsoft Excel, which also had the functionality for OCR. Given an image of a table, it could convert the image to a set of values and insert them into an Excel spreadsheet. It would then be possible to convert that spreadsheet to a set of CSV files and consequently into a Pandas DataFrame for future operations. However, uploading an image to Excel had a significant drawback. It would then require a Macros to return that data to the Python program, which is likely much more inefficient than using ExtractTable. Furthermore, this would require much more human interaction with the process since the user would be required to read through the entire report

and then crop an image of the table which would then be uploaded to Excel.

How ExtractTable Works

ExtractTable is a Python library enabling the extraction of tabular data from PDF files and images using Optical Character Recognition (OCR). OCR is a process that enables the recognition and conversion of printed or handwritten text into machine-encoded text. By using OCR, ExtractTable allows the extraction of tabular data within PDFs, returning them in a set of formats including but not limited to CSV, JSON and Pandas DataFrames.

A key difficulty when working with PDF files is the wide range of formats in which tabular data may be presented, such as scanned images or embedded fonts. Therefore, identifying tabular data within the PDF file can be very difficult, requiring OCR to do this effectively (Timalsina, 2023). The library automatically detects the tables within the file, recognises the text within the cell (along with a certainty score), and converts that into a machine-readable format ready for processing.

Furthermore, a range of preprocessing steps, such as image binarisation, noise removal and line detection. This enhances the quality of the input images and significantly improves the performance and effectiveness of the OCR (R, 2021). Finally, uses a range of methods or algorithms to extract the data identifying the fields, cells, rows and columns. Please note that due to being a paid product, these algorithms are not shared or published). Finally, the recognised text is extracted and restructured into a tabular format, where the user can specify the exact format as one of the arguments for the API calls.

In conclusion, ExtractTable effectively extracts tabular data from PDF files with high accuracy and returns the data in an easy-to-use format integrating straight into a Python application. This is vital for this task due to the size of the documents (often over 100 pages) being analysed; streamlining the data extraction task to create a much more seamless process is imperative for a well-functioning process.

FuzzyWuzzy

Given the challenge of matching KPIs to their representations in the table, fuzzy string matching was required due to the significant differences in the terms within the table. The library I opted for was FuzzyWuzzy, which utilises an algorithm called the Levenshtien Distance to calculate the differences between sequences (Cohen, 2022).

The Levenshtein Distance algorithm provides a measure of the similarity between two strings. This metric is calculated as the minimum number of single-character edits required to transform one into the other (Navarro, 2001).

FuzzyWuzzy was chosen because of the substantial documentation and online resources about the library's effectiveness and the ease of implementation in Python.

NumPy

The Python library NumPy (Developers, 2022) for efficient mathematical computations on the data extracted from the reports.

Pandas

The Python library Pandas (Inc, 2023) stores the tables returned from Extract-Table and data returned by the database.

psycopg2

Since the database system I opted to use was PostgreSQL, some database adapter was required. The option I chose was psycopg2 (Di-Gregorio and Varrazzo, 2022). This was for various reasons; however, predominantly being the most popular Python database adapter for PostgreSQL meant that significant resources would be available to me, allowing me to learn this framework. Since it was a framework I was unfamiliar with, having many resources available was important to ensure I stayed on track with development as effectively as possible.

5.1.3 Database

The choice of RDBMS was PostgreSQL because of the ease of integration with Python, my familiarity with PostgreSQL, and the ease of set-up across various machines.

Furthermore, using additional applications such as PG-Admin4 provided a graphical user interface (GUI) to manage, maintain and monitor PostgreSQL databases (Regina O. Obe, 2017). This was key to quickly and easily setting up the database for the project without needing to deal directly with command line interaction to the database.

5.2 System Design

As shown in the diagram below [Fig 5.1], the ESG-X system will be split into three components. The Web App contains the Table Extraction (5.2), Data Extraction (5.3) and Report Generation System (5.5). The ExtractTable API - is used to extract the tables and Microsoft Azure Database.

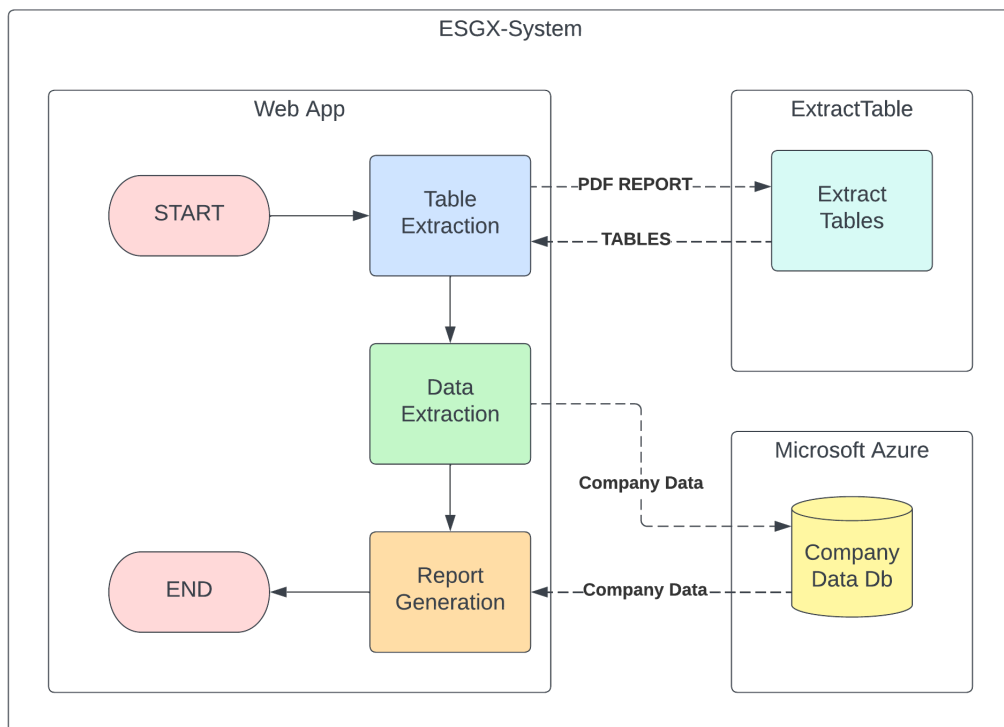


Figure 5.1: ESGX System Diagram.

5.2.1 Table Extraction

The table extraction system [Fig 5.2] allows the user to either enter the details for a report which may be stored in the system, or upload a report and enter the company_code and year corresponding to the uploaded report. This will then be sent to ExtractTable to extract the tables within the report. ExtractTable returns the tables within the report as a list of Pandas DataFrames, which are stored as a set of CSV files with that company code and the corresponding year. These are then accessed later on in data extraction.

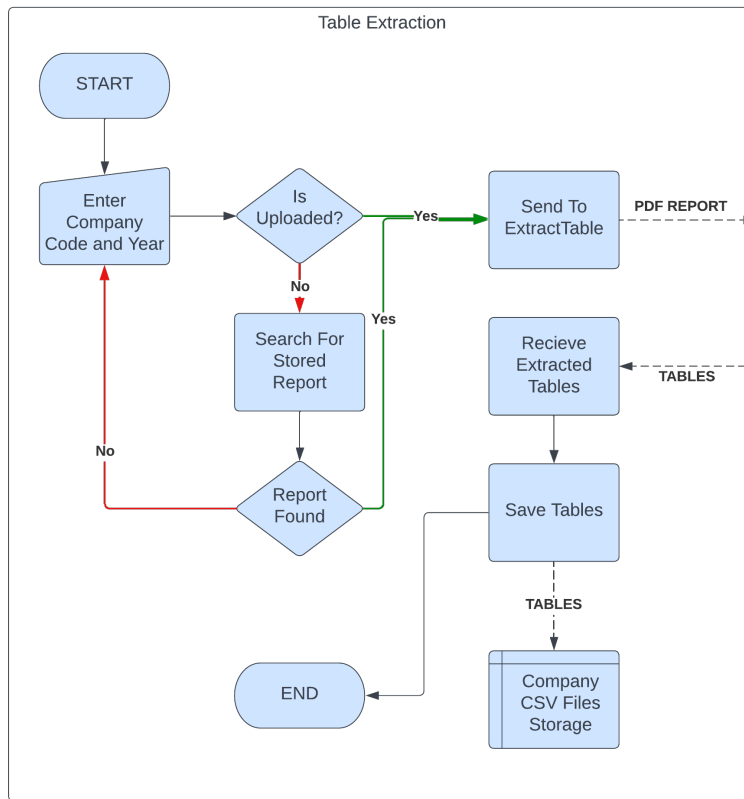


Figure 5.2: Table Extraction System Diagram.

5.2.2 Data Extraction

The data extraction system [Fig 5.3] allows the user to extract data from the collected tables. First of all, the user iterates through the list of tables. For each one, check if the table contains relevant data - has the KPIs required. If so, they state that the table is relevant and will match the KPIs within the table. This is automated - if matches are close enough to the queries or KPIs are stored. Otherwise, the user matches the KPI manually, and the connection between the corresponding KPI and the query is stored.

Once the KPIs have been matched, the corresponding values for each KPI and year are extracted from the table and stored in the database for later use in report generation.

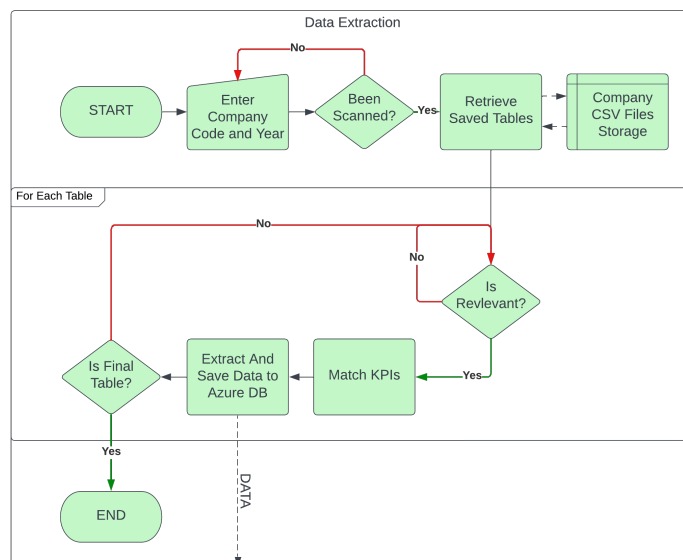


Figure 5.3: Data Extraction System Diagram.

Within the diagram (5.3), the 'KPI matching' process consists of a series of sub-processes outlined in the diagram below (5.4). Once the KPI column has been identified, the KPIs in that column are matched with the KPIs stored in the database. Fuzzy string matching is used here to match the KPIs, with the threshold score being 92. If the KPI is not automatically matched, the user selects the corresponding KPI manually. From here, a query (the KPI representation within the table) is stored along with the ID of the correct KPI. Therefore, this allows the system to match the KPI in the table to the correct KPI in the future - if the representation is similar enough.

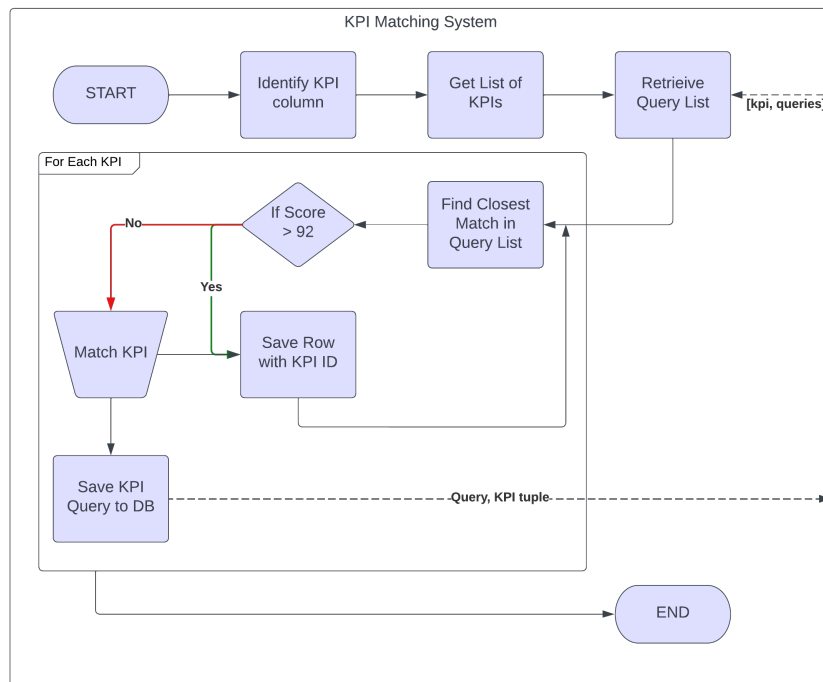


Figure 5.4: KPI Matching System.

5.2.3 Report Generation

Once the data for the company has been stored, the user can enter a company code which will then collect all relevant data for that company from the database. From there, it will separate the data into the different series required for each graph to present a set of charts to the user. As well as this, a ranking for each of the KPIs and for the company as a whole will be calculated and presented as part of the report.

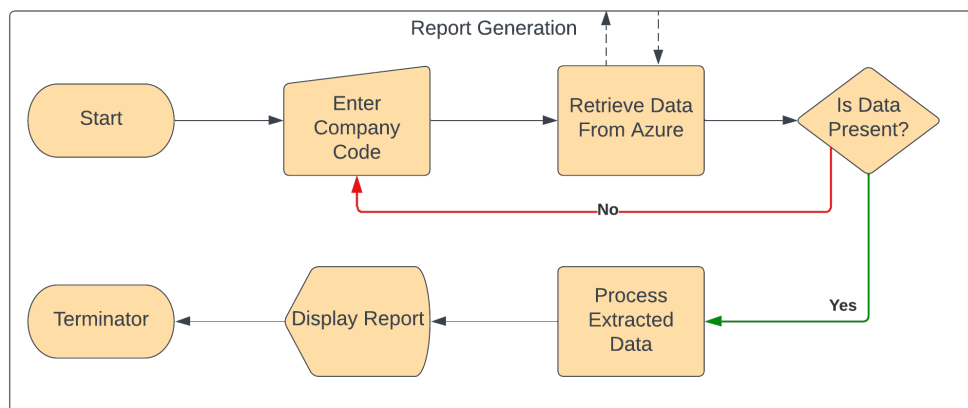


Figure 5.5: Report Generation System Diagram.

5.2.4 Database Design

During the development of the system, the database was stored in Microsoft Azure. This hosting service was chosen due to the relatively low cost associated with hosting on Azure through student subscriptions. Furthermore, having a remote database, even though it resulted in slower database access,

allowed the software to be developed on different machines without editing the connection strings for the database or ensuring that any updates to the database would then be the same for all instances or locations for development.

The entity relationship diagram (ERD) is shown below in figure [Fig 5.6].

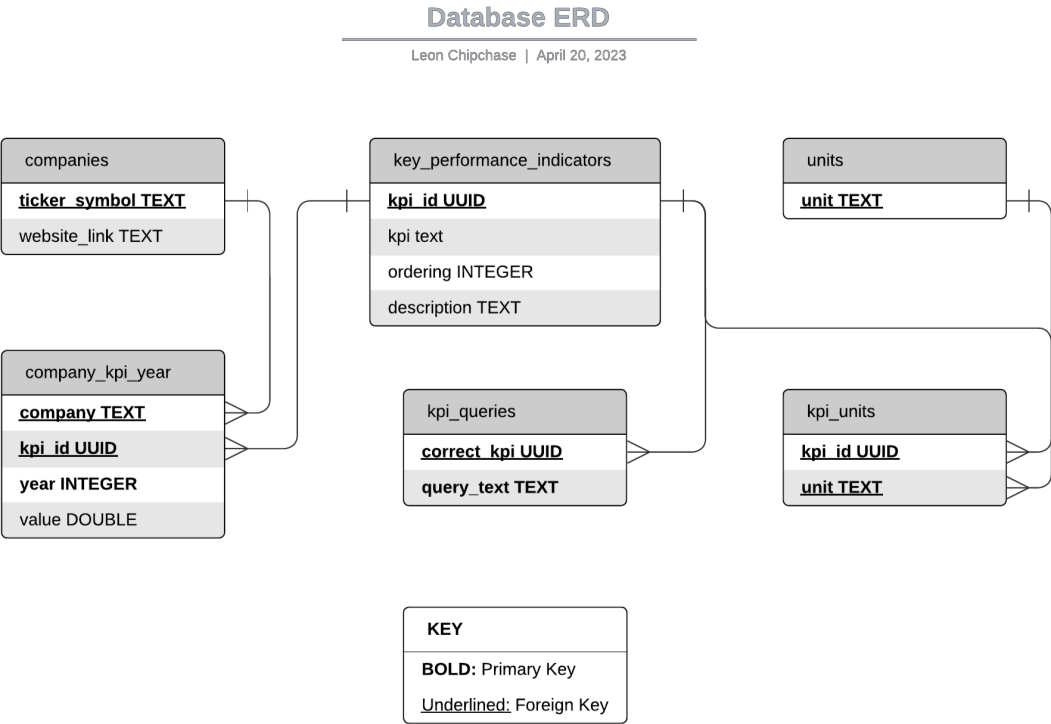


Figure 5.6: Database Entity Relationship Diagram.

The database is straightforward, consisting of only six tables containing all information required for the system to function. Additional materialised views are created for the rankings calculations, which are covered in section 6.3.3.

Table	Description
<i>companies</i> PK: ticker_symbol	This stores the companies analysed and their respective websites.
<i>key_performance_indicators</i> PK: kpi_id	This table stores the KPIs that are measured. The name of the KPI, An ordering (all are 0 but Not an Indicator which takes the value of -1), and a description of its meaning.
<i>company_kpi_year</i> PK: <company,kpi_id,year>	This stores a value of a specific KPI for a specific company for a specific year, the key is composed of (company, kpi_id, year).
<i>kpi_queries</i> PK: kpi_id	This table stores the queries used for the learning system. It stores the query itself and a reference to the KPI (kpi_id) to which that query corresponds to.
<i>units</i> PK: unit	This table stores the different units.
<i>kpi_units</i> PK: kpi_id, unit	This table stores the relationships between each of the KPIs (kpi_id) and the corresponding unit (unit).

Table 5.1: Database Tables

Chapter 6

Implementation

6.1 Research Methods

Since this project was predominantly based on identifying a way to identify a process for extracting the data from the reports and presenting the data. The majority of the time was focused on identifying the appropriate tools and methods how to do this. This process consisted of a range of steps.

1. Collecting the data
2. Identifying the data present within the reports, how they are structured, and different tools on how how to extract this
3. Once the location of the data in the reports has been identified, how to take get this from the report
4. Once the data has been extracted, how to best match the data to the KPIs that are relevant

5. Once the KPIs have been matched, how to extract the data from the table and how to input that into the database
6. Once the data has been collected, how to best report and show that data

The way each of these methods was identified was through research online, testing to see how the tools or methods found performed and then making a decision to adopt or to continue looking further.

The predominant testing environment used was Jupyter Notebooks, where countless methods were tested to see if they were suitable, very quickly cutting out the majority of tools found.

6.2 Extracting Identified Data

In this subsection, for each code snippet presented, any libraries required in the snippet will be shown as imports in the snippet itself.

6.2.1 Table Extraction

Since we had identified that each report contained a set of tables - typically at the end of the report. Extracting these tables was a key challenge to overcome. There were various options considered for this task, however the chosen option was (ExtractTable, 2023).

6.2.2 Identifying Relevant Tables

To identify which tables were relevant, the user would iterate through the tables returned from the reports via `ExtractTable`. By inspecting the table's contents, the user could either skip the table or state that the table was relevant and then match the relevant KPIs.

Even though ideally, this process would be automated, through trying various techniques, this was very inaccurate and very often included irrelevant tables. Therefore, this would have propagated the user interaction step to a different stage, requiring them to state the information was irrelevant.

6.2.3 KPI Matching

After the user had identified the table as containing relevant information, the system searched the table to determine which columns contained the desired KPIs. As shown in the code snippet [Fig 6.1], the column with the most matches is returned. When the user selects the relevant table, a column will contain a subset of the KPIs stated in the table [Table 3.1]. In this case, the threshold is set to 80 since the exact match of the KPI is irrelevant. Rather than there is a KPI present in this field.

```
import numpy as np
from fuzzywuzzy import process, fuzz

def find_best_matching_column(table, kpis, threshold):
    """
    Find the column which is the best match for the list of KPIs.
    Params:
        table [[String]]: The table to analyse
        kpis [String]: The list of KPIs to match too
        threshold int: The threshold for the accuracy of matching
    Returns:
        best_matching_column int: The index of the best matching column
    """

    best_matching_column = None
    highest_match_count = 0

    # Get the number of columns
    num_columns = table.shape[1]

    # For each of the columns
    for column_index in range(num_columns):
        match_count = 0

        # Convert the data to strings for matching
        column_data = table[:, column_index].astype(str).tolist()

        # For each of the KPIs
        for kpi in kpis:

            # Get the best match and the score
            best_match = process.extractOne(kpi, column_data, scorer=fuzz.token_set_ratio)

            # Check if it is above the threshold
            if best_match and best_match[1] >= threshold:
                match_count += 1

        # Check if the match count is the highest
        if match_count > highest_match_count:
            highest_match_count = match_count
            best_matching_column = column_index

    # Return the best match
    return best_matching_column
```

Figure 6.1: Finding KPI Column.

After the system identifies the column containing the KPIs, the next step is matching the KPIs within that column to the set of KPIs on which we are aiming to extract data. Due to the lack of reporting standards, there was a significant variation in how the KPIs were represented. Each of the KPIs significantly varied in their representation. This meant reducing the requirement for the user needing to identify each of the KPIs each time manually; the requirement for a system which 'learnt' based on experience was key.

First of all, fuzzy string matching was used to match the KPIs in the respective column. However, in this case, the threshold for a match was set to 92, significantly higher than previously for identifying the column since it is imperative to match the correct indicator in this stage. Furthermore, the requirement for 'Not an Indicator' was required, meaning that the user could select that they did not want to store the values for that row.

After this, if the match is not automatic, the user matches the KPI in the table to the desired KPI. The KPI in the table is stored as a 'Query' alongside the *kpi_id* of the correct KPI in the *kpi_queries* table. In future, the system will then attempt to match the KPIs in the KPI column to the stored queries, meaning that as more queries were 'learnt', less and less interaction was required from the user. Very often, when the user stated that a table contained the relevant KPIs, no additional input was required from the user to identify those KPIs. A diagram outlining the KPI matching process is shown in figure [Fig 5.4].

6.2.4 Data Extraction

After the KPIs are matched, further table processing is required. First, the row containing the years must be found to ensure that each of the values in the table can be matched to a $(kpi, year, value)$ tuple.

Two example representations of tables containing emissions data from Google [Fig 6.2] and Microsoft [Fig 6.3] outline two possible examples for the representation of the years' column. Therefore these cases must be handled correctly.

Key performance indicator	Assured for 2021 ²³	Unit	Fiscal year ²⁴				
			2017	2018	2019	2020	2021
OUR OPERATIONS							
GHG EMISSIONS							
Emissions inventory^{25,26}							
Scope 1	●	tCO ₂ e ²⁷	66,549	63,521	66,686	38,694	45,073
Scope 2 (market-based) ^{28,29}	●	tCO ₂ e	509,334	684,236	794,267	911,415	1,823,132
Scope 2 (location-based)	●	tCO ₂ e	3,301,392	4,344,686	5,116,949	5,865,095	6,576,239
Scope 3 (total) ³⁰		tCO ₂ e	2,719,024	12,900,467 ³¹	11,669,000	9,376,000	9,503,000
Scope 3 (business travel and employee commuting, including teleworking) ³²	●	tCO ₂ e	356,060	463,467	542,000	213,000 ³³	136,000
Scope 3 (other)		tCO ₂ e	2,362,964	12,437,000	11,127,000	9,163,000	9,367,000
Total (Scope 1, 2 [market-based], and 3 [total])		tCO ₂ e	3,294,907	13,648,224 ³⁴	12,529,953	10,326,109	11,371,205
Biogenic emissions	●	tCO ₂	14,708	22,862	21,905	5,417	3,797

Figure 6.2: Google Scope 1,2,3 Emissions.

Table 1
GHG emissions by Scope (mtCO₂e)

	FY17	FY18	FY19	FY20	FY21
Scope 1	107,452	99,008	117,956	118,100	123,704
Scope 2					
Location-Based	2,697,554	2,946,043	3,557,518	4,102,445	4,745,197
Market-Based	139,066	183,329	275,420	228,194	163,935
<i>Subtotal emissions (Scope 1 + 2 market-based)</i>	<i>246,518</i>	<i>282,337</i>	<i>393,376</i>	<i>346,294</i>	<i>287,639</i>
Scope 3					
Category 1 – Purchased Goods & Services ^{1,2,5}	4,058,000	4,452,000	4,411,000	4,156,000	4,930,000
Category 2 – Capital Goods ^{3,2,5}	1,666,000	2,185,000	2,340,000	2,962,000	4,179,000
Category 3 – Fuel- and Energy-Related Activities (Location-Based) ¹	540,000	550,000	650,000	770,000	810,000
Category 3 – Fuel- and Energy-Related Activities (Market-Based) ^{1,4}	250,000	220,000	270,000	310,000	310,000
Category 4 – Upstream Transportation ^{1,2}	52,000	53,000	96,000	102,000	225,000
Category 5 – Waste ^{1,6}	700	500	10,500	9,500	5,700
Category 6 – Business Travel	419,020	461,787	476,457	329,356	21,901
Category 7 – Employee Commuting ¹	343,000	345,000	411,000	317,000	80,000
Category 9 – Downstream Transportation ^{1,3}	85,000	98,000	57,000	47,000	45,000
Category 11 – Use of Sold Products ^{1,3,5}	3,757,000	3,910,000	3,375,000	2,983,000	3,950,000
Category 12 – End of Life of Sold Products ^{1,3}	31,000	18,000	18,000	17,000	19,000
Category 13 – Downstream Leased Assets ¹	700	1,700	800	6,100	18,900
<i>Subtotal emissions (Scope 3 market-based)⁷</i>	<i>10,662,000</i>	<i>11,745,000</i>	<i>11,466,000</i>	<i>11,239,000</i>	<i>13,785,000</i>
Total emissions (Scope 1 + 2 + 3) ⁷	10,909,000	12,027,000	11,859,000	11,585,000	14,073,000

Figure 6.3: Microsoft Scope 1,2,3 Emissions.

Since the company decides the exact format of the report, the years differed in representation as well. The variation of this was: in descending order, ascending order, in the format *xxxx*, in the form *xx*, in the form *FYxx*, and in the

form *FYxxxx*, where *x* represents a number between 0 and 9. Consequently, each of these cases needed to be considered. The code snippet for extracting the years' row is shown in figure [Fig 6.4]

```

import pandas as pd
import re

def find_years_row(table):
    """
    Find the row containing the years.
    Params:
        table DataFrame (Pandas): The table containing the data
    Returns:
        index int: The index of the row
    """
    # For each index, row pair in the table
    for index, row in table.iterrows():
        years_found = 0
        prev_year = None

        # For each item in the row
        for item in row:

            # Try where the year is an in the integer format
            try:
                current_year = int(item)

                # If the difference between the items is 1 (asc or desc years)
                if prev_year is not None and abs(current_year - prev_year) == 1:

                    # Increment th year count
                    years_found += 1 if years_found > 0 else 2
                else:
                    years_found = 0
                    prev_year = current_year

            # If the year did not match the int format
            except:
                try:

                    # Check if there is a are integers there and extract the group
                    match = re.search(r'\d{2,4}', item)
                    if match is not None:

                        # Get the year from the match of the regex, then perform same check as above
                        current_year = int(match.group())
                        if prev_year is not None and abs(current_year - prev_year) == 1:
                            years_found += 1 if years_found > 0 else 2
                        else:
                            years_found = 0
                            prev_year = current_year
                    else:
                        prev_year = None
                        continue
                except:
                    pass

            # If there are two or more years present, return that index
            if years_found >= 2:
                return index

    # Else return none (no year present)
    return None

```

Figure 6.4: Finding the Years Row.

After this, the unit column is removed from the table because there is already

a database table (*kpi_units*) that stores the units for each of the KPIs. It also reduces the requirement for processing the units when inserting the values into the database. Once again, fuzzy string matching is used to identify the units, using the library *fuzzywuzzy* and the functions *process* and *fuzz*. This will match the items in the columns to a list of units (stored in the database).

```
from fuzzywuzzy import process, fuzz

def find_and_remove_unit_column(table, unit_list):
    """
    Find and remove the unit column from a table.
    Params:
        unit_list [String]: The list of units to match too
    Returns:
        table [[]]: The table without the unit column if there is one
    """

    # For each of the columns in the table
    for col_idx in range(table.shape[1]):
        unit_matches = 0

        # For each of the cells, see if there is a match in the column
        for cell in table[:, col_idx]:

            # If there are any matches above score 80 add to the count
            if any([fuzz.token_set_ratio(str(cell), unit) > 80 for unit in unit_list]):
                unit_matches += 1

            # If there are two or more matches, remove that column and return the table
            if unit_matches >= 2:
                return np.delete(table, col_idx, axis=1)

    # Else return the table
    return table
```

Figure 6.5: Finding the Unit Column.

6.3 Data Representation

The final challenge for the project was displaying the data to the user effectively. As stated in chapter 4, effective data representation is crucial in conveying the data and a message to the user as well as possible. Two main approaches were chosen for this part, one using PDF reporting and one using Interactive Reports.

6.3.1 PDF Reporting

Though this was initially considered an option and developed for approximately a week until producing a report that provided the same value and quality as one from using an interactive UI was deemed unfeasible. An example output from a graph using the libraries *matplotlib* and *reportlab* is shown in the figure below [Fig 6.6].

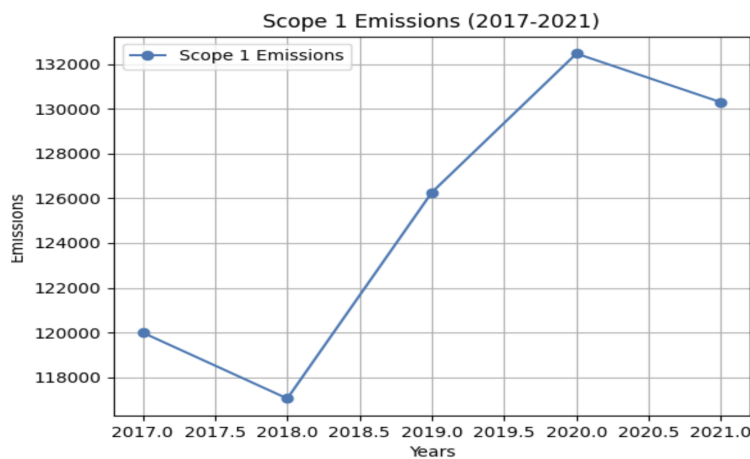


Figure 6.6: Example PDF Generated Report.

Though this graph effectively shows the data for the scope one emissions, there

are various issues with PDF report generation. Firstly, the lack of customisation with libraries like *Matplotlib* and *ReportLab* in comparison to others such as ChartJS. Finally, in this case, the critical issue was the lack of interactivity; since the PDF reports are static, they do not support features such as interactive graphs and filters.

6.3.2 Graphing Results

The option of creating an interactive report was chosen due to: the increased interactivity, the ease of implementation, and the versatility of having a non-static UI. To outline the process for preparing the data for the Scope Emissions Graph

First, the company data is retrieved from the database. The database adapter used for this task was *psycopg2*. This allowed simple interaction with the database from within Python. An additional benefit of this was binding the *company_code* to the SQL query preventing an SQL injection. Once the data has been retrieved from the database, as shown in figure [Fig 6.7], the rows are retrieved with *cursor.fetchall()*, then converted into a DataFrame and returned.


```

def all_company_data_raw(self, company_code):
    """
    Get all the stored data for a company without any further processing.
    Params:
        company_code (string): The code to filter the company
    Returns:
        DataFrame(rows) [Pandas]: A pandas dataframe with the company data in the database
    """
    try:
        # Define the connection cursor
        cursor = self.conn.cursor()

        # Execute the SQL with the company_code bound to the statement
        cursor.execute("""
            SELECT
                kpi_id, year, value
            FROM
                company_kpi_year
            WHERE
                company=(%s)
            ORDER BY
                year;
            """,
                (company_code,))

        rows = cursor.fetchall()
        cursor.close()

        # Convert to a DF and return
        return pd.DataFrame(rows)
    except:
        pd.DataFrame()

```

Figure 6.7: Retrieving Data From DB.

To retrieve this data within the *views.py* file, First, a DatabaseConnector object is defined. This contains the connection strings to the database and all functions to interact with the PostgreSQL database. As shown in figure [Fig 6.8], an additional check is done to ensure that there is data for the company present.

```
# Get the database connector object
database_connector = DatabaseConnector()

# Get all the data associated with that company
all_company_data = database_connector.all_company_data_raw(company_code)

# Check the data is present for that company
if database_connector.company_data_length(company_code) > 0:
    existing = 1
else:
    existing = 0
```

Figure 6.8: Getting all Data.

Once the company data has been retrieved from the database, this data must be filtered for each KPI. The filter function is shown in figure [Fig 6.9].

```
def filter_rows(x, data):
    """
    Filter a DataFrame based on a UUID x and return the filtered data.
    Params:
        x (UUID): The UUID of the KPI to be filtered
        data (DataFrame [Pandas]): The dataframe containing all company data
    Returns:
        data ([numpy]): A numpy array containing the filtered data
    """
    # Convert the DF to numpy
    data = data.to_numpy()

    # If there is data present, then filter based on x and return
    if len(data) > 0:
        return data[np.where(data[:, 0] == uuid.UUID(x))]

    # Else return an empty list
    return []
```

Figure 6.9: Filter Function.

As shown in figure [Fig 6.10], *all_company_data* is filtered according to each of the *kpi_ids*, and defined as a list which will later be sent to the front-end to be processed and used within the graph. In this case, there is an additional operation of summing the data in the different lists to create a *total_scoped_data* array - a value sometimes not explicitly published within the report.



```

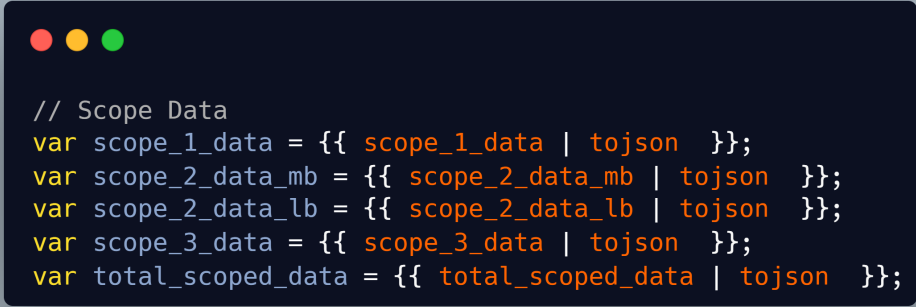
# Get data for scope 1,2,3 emissions
scope_1_data = [x[2] for x in list(filter_rows("39da681c-83c9-4986-9694-7c882eed3109", all_company_data))]
scope_2_data_mb = [x[2] for x in list(filter_rows("3db40d01-6b90-4a20-98f8-2d34f844c7d1", all_company_data))]
scope_2_data_lb = [x[2] for x in list(filter_rows("2e8c455a-e6f7-492f-a688-b32206c664ca", all_company_data))]
scope_3_data = [x[2] for x in list(filter_rows("059e0b65-a6e3-42b4-81b7-7501ee3e3a09", all_company_data))]
total_scoped_data = []

# If they have sufficient data for all the fields add together, if not then total scoped would produce incorrect
## information
try:
    total_scoped_data = list(np.sum([scope_1_data, scope_2_data_mb, scope_3_data], axis=0))
except:
    pass

```

Figure 6.10: Filtering KPIs.

Within the HTML file, the data sent from the Flask back-end must be converted into JavaScript arrays for use within the chart. This is converted using the Jinja templating engine (Jinja, 2022) - each of the variables they are assigned the value of the server-side variable. For example, the `| tojson` is a filter provided by Jinja2. The value is then assigned to the variable once converted to a JSON string.



```
// Scope Data
var scope_1_data = {{ scope_1_data | tojson }};
var scope_2_data_mb = {{ scope_2_data_mb | tojson }};
var scope_2_data_lb = {{ scope_2_data_lb | tojson }};
var scope_3_data = {{ scope_3_data | tojson }};
var total_scoped_data = {{ total_scoped_data | tojson }};
```

Figure 6.11: Converting to JS data.

Finally, once the required data has been retrieved from the server side, the graph is created using the code shown in figure [Fig 6.12]. Firstly, a context *scopeCTX* is created by getting an HTML element by their id (in this case), *scopeData*. Next, the data is defined, with the x-axis being the years and the different series defined like the example. This is done for each of the series; however, in the snippet, only *scope_1_data* is shown for purposes of clarity. Finally, the chart itself is defined with: the type of chart, the data, and any additional options, such as the chart titles and which axes each series corresponds to (not shown in this example).

For further details on (ChartJS, 2023), visit the documentation page.



```

// Scoped Data Chart
var scopeCTX = document.getElementById('scopeData').getContext('2d');
var data = {
  // Define the x-axis labels
  labels: years,
  datasets: [
    // For each of the arrays of data,
    {
      label: 'Scope 1',
      borderColor: 'rgb(153, 102, 255)',
      fill: false,
      data: scope_1_data,
    },
    // Repeat for each additional series
  ],
};
var scopeChart = new Chart(scopeCTX, {
  type: 'line',
  data: data,
  // Defining additional options
  options: {
    title: {
      display: true,
      text: 'Carbon Emissions (Tonnes CO2e)',
    },
    scales: {
      yAxes: [
        {
          ticks: {
            beginAtZero: true,
          },
        },
      ],
    },
  },
});

```

Figure 6.12: Creating the graph within JavaScript.

Finally, we are left with the graph displayed. This example is for data collected from the 2022 ESG Report for Google. It shows the series for five different metrics, and each of these can be toggled to focus each of the series themselves; furthermore, the scale of the axes will automatically adjust to fit the data as well and as clearly as possible, giving the user an immersive, interactive experience, effectively showing all collected data to the user.

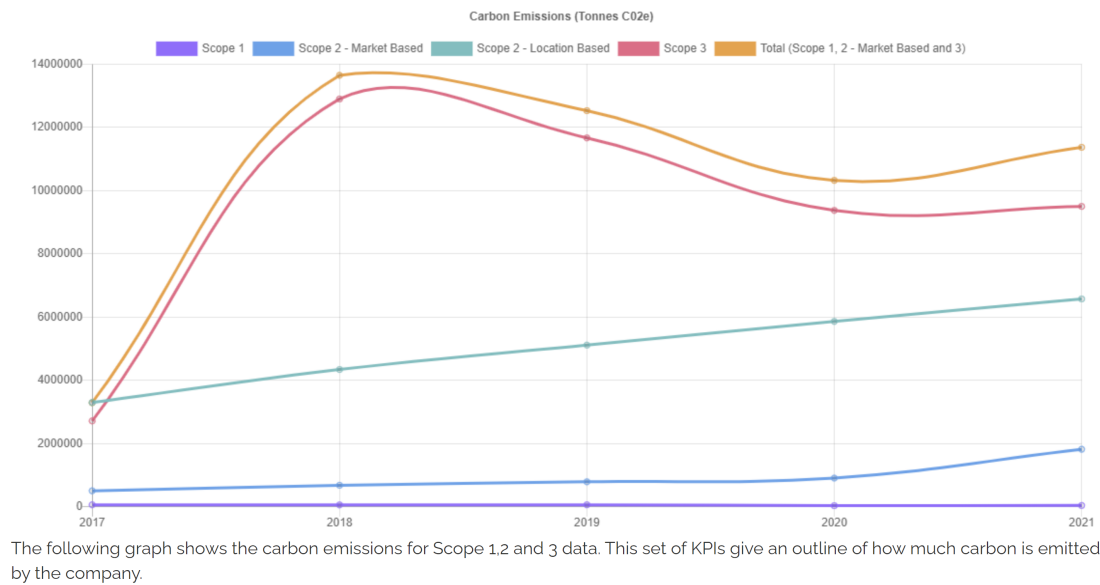


Figure 6.13: Scoped Emissions Data - Google 2017 - 2021.

The graphs shown in the report are Scoped Emissions, Energy Usage, Electricity Usage, Water Usage, and Carbon Intensity Metrics. The combination of these five graphs highlights the most significant direct impacts of the company on the environment. Suppose any of the data is not collected. In that case, this purely results in a graph without that particular series, meaning that if the company does not publish the data, this does not cause any additional issues and will still allow a report to be published.

6.3.3 Company Ranking

Another key feature of the project was to produce a ranking system for the set of companies. This was achieved with two approaches, one listing the rank for each of the KPIs for each company. Then secondly, taking the rank based on the mean position for the companies.

There were other options considered additionally, such as calculating a function with different weightings for each of the different metrics and assigning each of the companies a score from 0 to 1; however, since there was a large variation in which of the companies published which figures, this was opted against.

To calculate the rank, the following steps are taken.

1. Calculate the rank of each KPI for that company
2. Calculate the mean of the ranking
3. Get the rank/position based on the mean ranking across all of its KPIs

A company is not penalised for having a piece of missing information; this is because, in many cases, a company may not publish the data for that metric (particularly for a smaller company). Furthermore, for each company, its rank and value are presented for the metric that is given.

```

-- Calculate the rank of a company based on KPI values for a given year (2021 in the system)
CREATE OR REPLACE FUNCTION company_kpi_rank(p_company_code TEXT, p_year INTEGER)
RETURNS TABLE (company TEXT, kpi_id UUID, year INTEGER, value DOUBLE PRECISION, rank BIGINT) AS $$
BEGIN
RETURN QUERY
-- The CTE kpi_rankings calculates the rankings for each KPI, considering the input year
WITH kpi_rankings AS (
SELECT
company_kpi_year.company,
company_kpi_year.kpi_id,
company_kpi_year.year,
company_kpi_year.value,
-- Some KPIs in list in descending order else ascending
CASE
WHEN company_kpi_year.kpi_id IN (
'567bb0d9-5b3c-4549-95c5-13558699b546',
'89f2f8eb-8b06-45a9-8fe8-b7bb3f462c7e',
'b3b92136-1271-4319-a929-304d41b09ac9'
) THEN ROW_NUMBER() OVER (
PARTITION BY company_kpi_year.kpi_id
ORDER BY company_kpi_year.value DESC
)
ELSE ROW_NUMBER() OVER (
PARTITION BY company_kpi_year.kpi_id
ORDER BY company_kpi_year.value ASC
)
END AS rank
FROM company_kpi_year
WHERE company_kpi_year.year = p_year
)
-- This query returns the company, KPI, year, value, and rank for the given company code
SELECT
r.company,
r.kpi_id,
r.year,
r.value,
r.rank
FROM kpi_rankings AS r
WHERE r.company = p_company_code;
END; $$
LANGUAGE 'plpgsql';

```

Figure 6.14: Company Ranking Function (*company_kpi_rank(p_company, p_year)*).

An SQL function called *company_kpi_rank(p_company, p_year)*, shown in figure [Fig 6.14], will get the ranking for each of the KPIs, for a specific *company_code* and a specific year. The KPI IDs correspond to the KPIs in table

[Table 3.1]: Landfill Diversion Rate (%), Proportion of Renewable Electricity Used/Purchased (%), Proportion of Renewable Energy Used/Purchased (%).

Once the function has been defined, is used to create a MATERIALISED VIEW called *company_mean_rank_view*. Firstly, a MATERIALISED VIEW called *company_ranking_kpis* is created, which will get the ranking for all KPIs for each company in the companies table using *CROSS JOIN LATERAL*, and then appends the result of the function to the result. The SQL for this view is shown in figure [Fig 6.15]

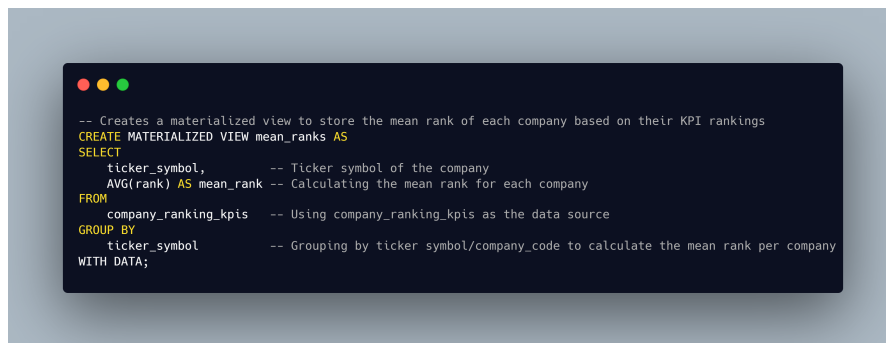
```

-- Create a materialized view to store company KPI rankings for the year 2021
CREATE MATERIALIZED VIEW company_ranking_kpis AS
SELECT
  c.ticker_symbol, -- Ticker symbol of the company
  k.kpi,          -- Key Performance Indicator name
  u.unit,        -- Unit of measurement for the KPI
  r.rank         -- Rank of the company for the given KPI
FROM
  companies AS c -- Companies table
CROSS JOIN LATERAL
  -- Using company_kpi_rank function to get KPI rankings for each company in 2021
  company_kpi_rank(c.ticker_symbol, 2021) AS r
NATURAL JOIN
  key_performance_indicators AS k -- Key performance indicators table
NATURAL JOIN
  kpi_units AS u -- KPI units table
ORDER BY
  c.ticker_symbol, -- Ordering by ticker symbol
  k.kpi,          -- Ordering by KPI name
  r.rank ASC     -- Ordering by rank, ascending
WITH DATA;

```

Figure 6.15: Company Ranking KPIs View.

From here, *company_ranking_kpis* is used in a second MATERIALISED VIEW called *mean_ranks*, which calculates the mean of the ranks grouped by *ticker_symbol*. The SQL for this view is shown in figure [Fig 6.16]



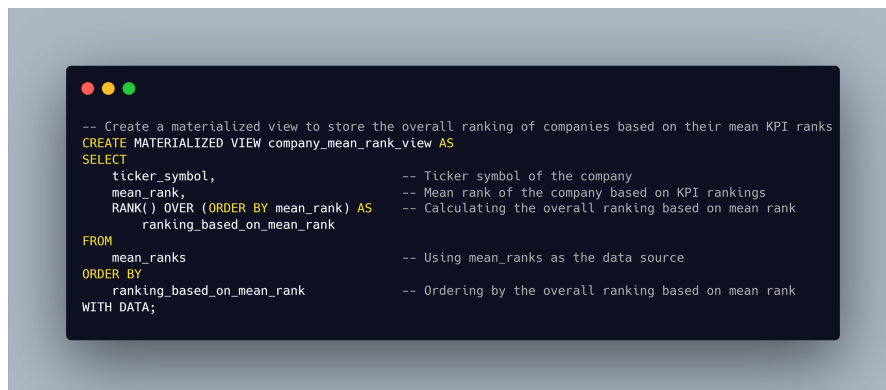
```

-- Creates a materialized view to store the mean rank of each company based on their KPI rankings
CREATE MATERIALIZED VIEW mean_ranks AS
SELECT
  ticker_symbol, -- Ticker symbol of the company
  AVG(rank) AS mean_rank -- Calculating the mean rank for each company
FROM
  company_ranking_kpis -- Using company_ranking_kpis as the data source
GROUP BY
  ticker_symbol -- Grouping by ticker symbol/company_code to calculate the mean rank per company
WITH DATA;

```

Figure 6.16: Mean Ranks View.

Finally, the MATERIALISED VIEW *company_mean_rank_view* is created, which selects the *ticker_symbol*, the *mean_rank* and the ranking based on *mean_rank*. Therefore, when finding company data, the ranking for each company does not need to be calculated each time, saving computation time. The SQL query for creating the view is shown in figure [Fig 6.17].



```

-- Create a materialized view to store the overall ranking of companies based on their mean KPI ranks
CREATE MATERIALIZED VIEW company_mean_rank_view AS
SELECT
  ticker_symbol, -- Ticker symbol of the company
  mean_rank, -- Mean rank of the company based on KPI rankings
  RANK() OVER (ORDER BY mean_rank) AS -- Calculating the overall ranking based on mean rank
  ranking_based_on_mean_rank
FROM
  mean_ranks -- Using mean_ranks as the data source
ORDER BY
  ranking_based_on_mean_rank -- Ordering by the overall ranking based on mean rank
WITH DATA;

```

Figure 6.17: Company Mean Ranks View.

To update the ranking views, 'REFRESH MATERIALIZED VIEW view_name' is used after the data from each table in the report has been extracted.

Chapter 7

Project Management

7.1 Methodology

The methodology used for this project was Agile SCRUM. Using an Agile-based methodology was key to the success of the project. Due to the exploratory nature of the project, where the majority of the task was identifying how best to achieve a specific goal - KPI matching, for example, a flexible timeline was key. Furthermore, due to the small team size (one person), a methodology which relied on "People, not Process" (Schwaber, 2015) was key, allowing for maximum flexibility and efficiency when working.

SCRUM was adopted for two key reasons. Firstly, meetings with my project supervisor - Yu Guan, were fortnightly. This set a time to start and end sprints. The basis of these meetings differed in terms one and two. In term one, they focused primarily on identifying tools, methods and libraries. During the time up to the meeting, I would research different tools to use and how I believed personally would be the best way to achieve each specific. During the meeting,

we would discuss the findings and any challenges identified and set a goal for the next meeting. This was key to the project's success because having an additional perspective with more experience meant that original, more progressive ideas were given.

In term two, once the majority of the methods had been determined, the majority of the focus was on implementing a user interface to allow the user to interact with the system and view the collected data in the reports. Here is where more aspects of SCRUM and Agile development came into effect since this was the actual implementation phase of the project. The format of fortnightly meetings with Yu Guan continued. This aligned with my sprints, where I would implement various project features and set up a project board and a backlog of any features or components that still need to be implemented.

Despite this, there were also some aspects of the Waterfall Methodology, with a set of defined documents required for the project and a set deadline for the final deliverables. However, excluding the required deadlines and documents, the rest of the project timeline was flexible.

7.2 Source Control

For this project, GitHub was used for source control. This came with many benefits; firstly, for each new feature implemented, a new branch is created. Once that feature is implemented, the changes are merged back into the original branch. Therefore, if the feature is still problematic or has caused any issues with the rest of the system, this can be fixed in the branch before continuing or making any additional changes.

7.3 Timeline

7.3.1 Original Timeline

The timeline was split into two key phases. As shown in figure [Fig 7.1], there was originally a proposed timeline for me to achieve different deliverables. However, since most of the project was researching different methods and techniques for extracting data from the reports, this dominated the first term of the project development.

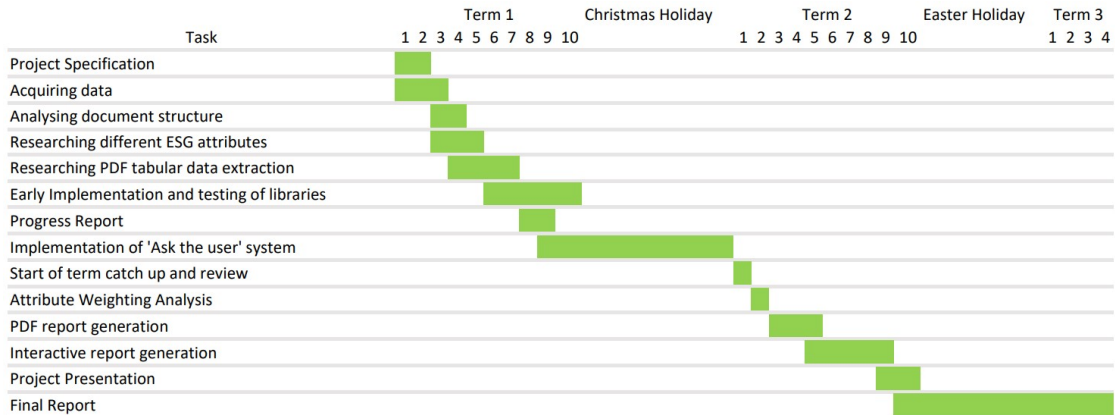


Figure 7.1: Project Timeline Gantt Chart.

Therefore, the actual timetable for term one is shown in figure [Fig 7.2], this accommodated any overruns due to extra workload or challenges.

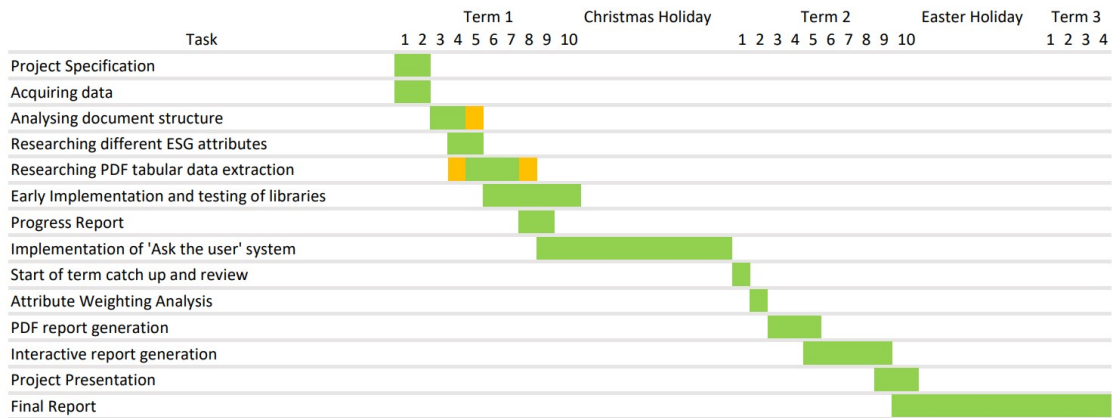


Figure 7.2: Revised Gantt Chart.

The fortnightly meetings with my project supervisor were key to the project’s success. Having insight from an external perspective, particularly one with more skill and knowledge than I have, meant that new ideas and techniques could be identified as quickly and efficiently as possible, guiding the project in its desired path and ensuring it keeps on track.

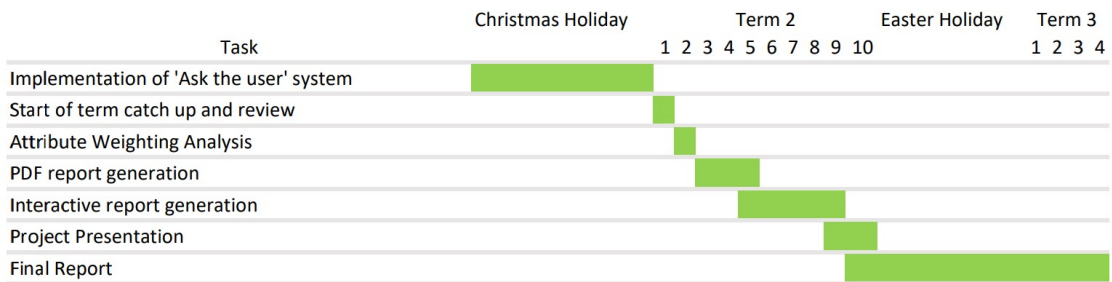


Figure 7.3: Term two project timeline.

By the time the GUI was being implemented, the 'KPI Matching System', for example, was already complete; only the way the user interfaces with the program had to be changed.

Since I was unfamiliar with the Flask framework at the beginning of the project, implementing the GUI using Flask in the back end of the system was a steep learning curve, once again highlighting the importance of a development methodology which allowed for flexibility and adaptation.

Testing was done throughout the project's lifetime, with each feature implemented, meaning that by the time a new section of the project was started, the previous was working functionally and was ready to be integrated and built upon. Consequently, the delay due to compatibility problems was negligible, ensuring the project stayed on track.

7.4 Risk Assessment

The uncertainty caused due to the research-oriented approach we opted for meant that a wide range of risks came associated with this project. These were of varying severity (**S**) and likelihood (**L**). The risk levels associated with this project were: Negligible - **N**, Low - **L**, Medium - **M** and High - **H**. Furthermore, the likelihood was assessed on a scale of 0 – 10, where 0 is extremely unlikely, and 10 is exceptionally likely.

Risk	S	L	Mitigation
The ESG reports are too unstructured to extract any relevant data effectively.	H	5	Research different sources of ESG data to try to extract and analyse in case extraction from the PDF reports is too infeasible.
The variations from report to report are too significant to produce a process which can efficiently extract data from more than one report format in the time allocated.	H	6	Same as above
Too many reports will not follow the desired structure to store their emissions data in tabular format.	H	1	All reports considered contained this information. Additionally, some companies also publish a document containing purely their KPI data in tables. Therefore this could also be scanned as a replacement.
Remote repository loss.	H	1	Ensure local backups across multiple machines and include a file history.
Failure of the hardware the project is being developed on, resulting in loss of code progress.	M	3	Use source control software (GitHub) to allow for backups and rollbacks in case new changes break the system.

Changes in public opinion focusing away from ESG metrics, preventing the requirement for companies from publishing this data.	H	1	The public's opinion can not be mitigated against. However, ensure the process would work for any KPIs which consist of numerical data to make the system as robust as possible.
inaccurate timeline estimation due to the difficulty in determining the time taken to complete a task given that the method is unknown.	M	8	Allow for a timeline giving additional time in components take too long to develop, along with a flexible scope, increasing if time permits, and decreasing if completely necessary.
Severe illness, Minjury or other external factors may cause delays.	M	2	Ensure that the planned deliverables timetable can accommodate delays or issues in case they occur.

Table 7.1: Risk Register

Chapter 8

Evaluation

This project was conducted as a research task with the goal of 'Analysing the Commitment of US Technology Companies Towards ESG Goals'. Therefore, though this was the goal, other aspects of software engineering and computer science were included in this project.

8.1 Evaluation of the developed process

Three main cases are considered to determine the success of the system developed. These cases will cover the Google 2021 Report (a large company where the process yielded the most significant success), Microsoft, which contains a differing representation of the tables to Google, as well as Advanced Micro Devices (AMD), where the process was less effective.

Google

Google, one of the largest companies considered, is in the constant public eye

and under constant scrutiny. Therefore, Google has a vast amount of resources under its control and a revenue larger than that of a small country. For these reasons, Google will likely report more figures than the rest of the cohort since there is a much larger requirement for investors to assess Google in terms of its ESG commitments and also have the capability to measure these factors. In addition, Google predominantly offers online services such as the Google search engine, cloud computing services, and data analytics due to its vast collected data.

Microsoft

Microsoft also falls into this category, however under a slightly different subsection, since they produce consumer goods such as electronics as well as offering online services such as Microsoft Azure.

8.1.1 Table Extraction

This part of the evaluation is based on how effectively the tables from the report itself were extracted. A score of 100% would indicate that all of the tables from the report themselves were extracted and then saved for the next process step. Additionally, the time taken to extract the tables is considered to see how quick this process is; however, the user can continue using the system whilst the scan is ongoing in the background. Therefore, this metric is less indicative of the success of the table extraction.

The tabular extraction method is outlined in section [Section 6.2.1], which details the process itself.

Google 2022

The table extraction method managed to extract 14 out of 14 of the tables from the report, giving this a score of 100%.

Microsoft 2021

The table extraction for this report successfully extracted 50 out of 50 tables from this report, giving a score o 100

AMD 2021

The table extraction for this report successfully extracted 61 out of 61 of the tables, giving a score of 100%.

Summary

The following table [Table 8.1] outlines the information about the report and the information relevant to the success of the extraction. The overall success rate is 100% since all three reports extracted all tables in the report. However, as well as this, the time taken to scan a report, particularly as the table and page count increase and the time taken to extract the data increases significantly too.

Report	Page Count	Table Count	Extraction Time (S)	Tables Extracted	Score (%)
GOOGL	17	14	24	14	100
MSFT	116	50	160	50	100

AMD	105	61	156	61	100
-----	-----	----	-----	----	-----

Table 8.1: Table Extraction Results

A graph containing the results in figure [Fig 8.1] for scanning times of 30 reports is shown below, indicating a direct relationship between the page count and the time taken for the scan. However, a proportion of this time will depend on the number of images within the report and the associated time to send the PDF to the API.

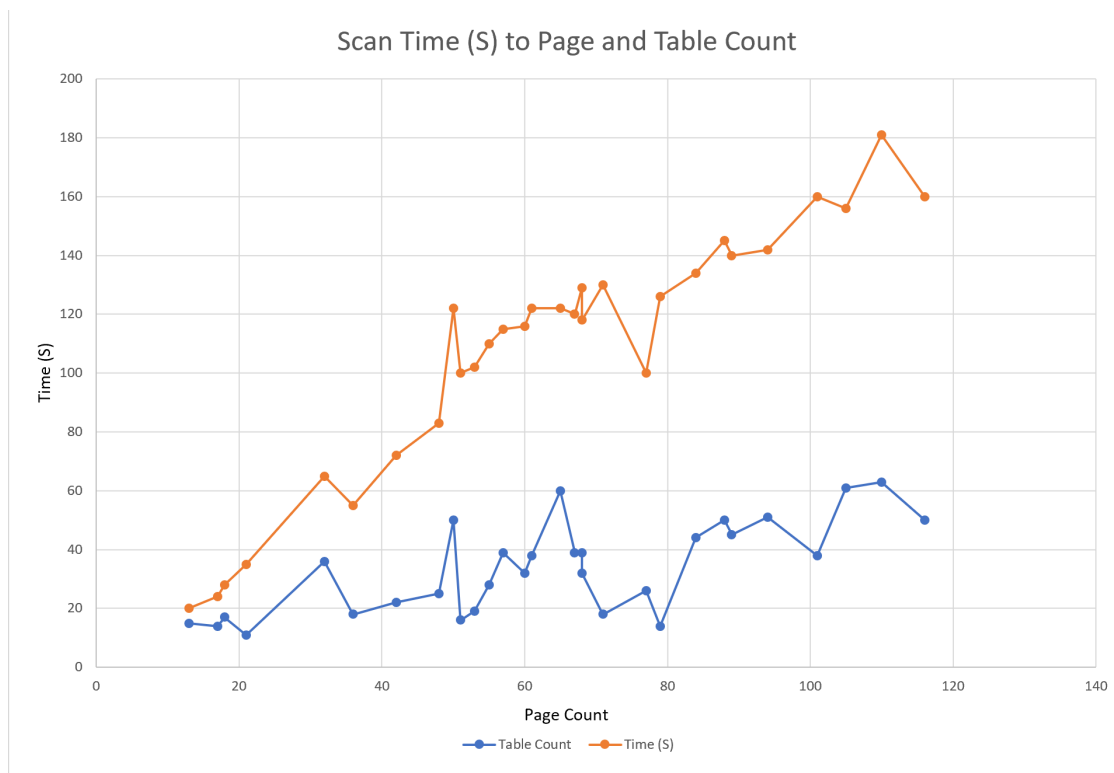


Figure 8.1: Term two project timeline.

A table of correlation coefficients between the values is shown in the table

below [Tab 8.2], further highlighting how closely related the page count and the time taken for a report scan are. Furthermore, there is a strong correlation between the table count and the time taken and the page count to the table count.

Relationship	Correlation Coefficient (R)
Page # : Table #	0.711
Page # : Time Taken (s)	0.946
Table # : Time Taken (s)	0.705

Table 8.2: Extraction Time Correlation Coefficients

Please Note: These tests were performed on a WIFI connection with an upload speed of less than 1Mb/S upload speed, and considering the size of the pdf reports, this will have had a significant impact on the results. Additionally, these tests can not be repeated due to the credit use of the API.

8.1.2 KPI Matching and the Learning Process

This part of the evaluation is based on how effectively the KPIs were matched from the report and which proportion of the published KPIs relevant to us were extracted from the report successfully. A score of 100% would indicate that all of the KPIs were matched - with or without user assistance.

This includes cases where the database of queries is empty and after the process has been used to extract the KPIs of a set of ten reports. This information will be used to evaluate the success of the KPI matching as a whole and also the learning process in itself. Identifying where the process works well and

any possible issues arising from this.

For this section of the evaluation, each of the KPIs will be referenced using the code outlined in the table below [Tab 8.3].

KPI Code	KPI
1	Carbon Intensity Per FTE Employee
2	Carbon Intensity Per Megawatt-Hour of Energy
3	Carbon Intensity Per Unit of Revenue Million USD (\$)
4	Landfill Diversion Rate (%)
5	Proportion of Renewable Electricity Used/Purchased (%)
6	Proportion of Renewable Energy Used/Purchased (%)
7	Scope 1
8	Scope 2 - Market Based
9	Scope 2 - Location Based
10	Scope 3
11	Total Electricity Used/Purchased
12	Total Energy Used/Purchased
13	Total Renewable Electricity Used/Purchased
14	Total Renewable Energy Used/Purchased
15	Waste Generated
16	Water Consumption
17	Water Discharge
18	Water Withdrawal
19	Not an Indicator

Table 8.3: KPIs codes

Furthermore, **MA** will stand for 'Matched Automatically' **WQT** will stand for 'With Query Table' - where the queries have been matched with the query table populated after ten report scans. Furthermore, **NQT** will stand for 'No Query Table'- where the table of queries has been emptied.

Google 2022

For this report, as shown in the table below [Tab 8.4], before the query table was used, only six out of eighteen of the KPIs were matched. However, of the six matches, all these values were matched correctly. However, once the query table was used after scanning a set of reports already, these KPIs were matched automatically and correctly with a 100% success rate. This will be due to the scanned reports containing a query similar enough to the KPI as it is presented in the Google report.

The data for KPI 14 [8.3], though present in the table, was split across multiple rows without a single data field present for that KPI. Therefore, a way to improve this would be to enable this capability.

KPI Code	MA-NQT	Correct NQT	MA-WQT	Correct WQT
1	N	N/A	Y	Y
2	N	N/A	Y	Y
3	N	N/A	Y	Y
4	N	N/A	Y	Y
5	N	N/A	Y	Y
6	N	N/A	Y	Y

7	Y	Y	Y	Y
8	N	N/A	Y	Y
9	N	N/A	Y	Y
10	Y	Y	Y	Y
11	N	N/A	Y	Y
12	N	N/A	Y	Y
13	N	N/A	Y	Y
15	Y	Y	Y	Y
16	Y	Y	Y	Y
17	Y	Y	Y	Y
18	Y	Y	Y	Y

Table 8.4: Google 2022 KPI Matching Results

Microsoft 2021

The results for the Microsoft report were very similar; there was difficulty in matching the data before the query table was used - in this case, only 3 of 12 KPIs were initially matched automatically and correctly. However, once the table of queries was populated after entering the queries, these KPIs were once again matched automatically and correctly; the exception for this was KPI 3,13,14 [8.3]. This was due to a vastly differing presentation of the KPI to the other queries. However, with the user selecting the correct option when asked by the system, this value was also matched correctly. Furthermore, in the case of KPI 13,14, the representation in the table differed from the desired representation.

Though less of KPIs were present in this report, the Microsoft report also offered a much more comprehensive range of different KPIs and, therefore, a wider range of tables; therefore, in future, it would be beneficial to include further KPIs for each company to be able to measure these metrics even if they are not present for as many companies as desired.

KPI Code	MA-NQT	Correct NQT	MA-WQT	Correct WQT
1	N	N/A	Y	Y
2	N	N/A	Y	Y
3	N	N/A	N	N/A
5	N	N/A	Y	Y
7	Y	Y	Y	Y
8	Y	N	Y	Y
9	N	N/A	Y	Y
10	Y	Y	Y	Y
11	N	N/A	Y	Y
12	N	N/A	Y	Y
13	N	N/A	N	N/A
14	N	N/A	N	N/A
16	N	N/A	Y	Y
17	N	N/A	Y	Y
18	N	N/A	Y	Y

Table 8.5: Microsoft 2021 KPI Matching Results

AMD 2021

Once again, the results for the AMD report were very similar and followed the same trend as the two above cases. Before the queries were learned, a low proportion of KPIs matched. However, those that were matched did so with an accuracy of 100%. Once again, when the table with the queries for the KPIs was present, the KPIs were matched automatically and correctly for all but KPIs 3,14, once again due to the significantly differing representation of the KPIs within the table.

Once again, like Microsoft, a different set of KPIs was present in the report. Therefore, some of the KPIs were not measured for this test. Once again, AMD included a large set of different KPIs, which would have outlined more information on the company.

KPI Code	MA-NQT	Correct NQT	MA-WQT	Correct WQT
1	N	N/A	Y	Y
3	N	N/A	N	N/A
5	N	N/A	Y	Y
7	Y	Y	Y	Y
8	Y	N	Y	Y
9	N	N/A	Y	Y
10	Y	Y	Y	Y
11	N	N/A	Y	Y
12	N	N/A	Y	
13	N	N/A	Y	Y
14	N	N/A	N	N/A

16	N	N/A	Y	Y
17	N	N/A	Y	Y
18	N	N/A	Y	Y

Table 8.6: AMD 2021 KPI Matching Results

Conclusion

The evaluation of the KPI matching system and the learning process has shown the system’s versatility in identifying the KPIs relevant across different reports (Google 2022, Microsoft 2021 and AMD 2021). It is evident that the learning process is integral to the overall success of the matching system.

Initially, the system had a low success rate in the KPI matching when the query table was empty. However, the accuracy of the matches was high (100%); this is due to the high matching threshold required for an automatic match [6.2.3]. However, this success rate improved significantly after the query table was populated with the queries from scanning ten other reports. Please note where the format matched the KPI; after scanning a report a second time, the KPI matches were 100% after they had learned the format of that particular company report.

The results also highlight the importance of the system’s ability to handle more complex representations of the KPI - across multiple rows, or where the two pieces of information required to determine a KPI were on two separate rows (say revenue and carbon emissions - this could be used to calculate carbon in-

tensity).

In conclusion, the KPI matching and learning system was a success, removing the requirement for user input as much as possible, however still shows that for this system, a user must oversee the process to correct any errors or make the connection to the correct KPI for the system in case the representations differ to significantly.

8.1.3 Data Extraction

This evaluation section is based on how effectively the data from the tables where the KPIs have been matched can be inserted into the database for the corresponding year as a record. Highlighting any cases where the method of data extraction worked successfully.

Instead of just using AMD 2021 for section 8.1.3, being used here, multiple reports with different examples of problematic data representation will be used to highlight difficulties in varying representations for the extraction of data.

Google 2022

Shown in the table below 8.7 is the extracted data for Google from the 2022 report. As shown, for all KPIs matched, the data were extracted correctly for the KPIs and the years. For example, for KPIs 16,17 8.3, and years 2017 and 2018, where the value in the table is 0. However, this is a placeholder since Google did not publish the results for these figures for those years.

Key Performance Indicator	2017	2018	2019	2020	2021	Unit
Carbon Intensity Per FTE Employee	7.6	8.36	7.96	7.49	12.87	tCO ₂ e/ FTE
Carbon Intensity Per Megawatt-Hour of Energy	0.0717	0.0707	0.0675	0.0615	0.1006	tCO ₂ e/ per megawatt-hour of energy consumed
Carbon Intensity Per Unit of Revenue	5.19	5.47	5.32	5.21	7.25	tCO ₂ e/ million US\$
Landfill diversion rate (%)	83	80	77	77	77	%
Proportion of Renewable Electricity Used/Purchased (%)	100	100	100	100	100	%
Proportion of Renewable Energy Used/Purchased (%)	100	100	100	100	100	%
Scope 1	66549	63521	66686	38694	45073	tCO ₂ e
Scope 2 Location-Based	3301392	4344686	5116949	5865095	6576239	tCO ₂ e
Scope 2 Market-Based	509334	684236	794267	911415	1823132	tCO ₂ e
Scope 3	2719024	12900467	11669000	9376000	9503000	tCO ₂ e
Total Electricity Used/Purchased	7609089	10104295	12237198	15138543	18287143	MWh
Total Energy Used/Purchased	8029409	10572485	12749458	15439538	18571659	MWh
Total Renewable Electricity Used/Purchased	7609089	10104295	12237198	15138543	18287143	MWh
Waste Generated	53363	57113	48126	28864	28153	Metric Tons
Water Consumption	0	0	3412	3749	4562	Million Gallons
Water Discharge	0	0	1749	1940	1735	Million Gallons
Water Withdrawal	3071	4170	5161	5689	6297	Million Gallons

Table 8.7: Google Extracted Data.

For the success of data extraction for Google, the process scores 100%, showing the effectiveness of the data extraction process where the format of the document aligns with the system properly.

Microsoft 2021

Shown in the table below, 8.8, is the extracted data for Microsoft from the 2021 report,

Key Performance Indicator	2017	2018	2019	2020	2021	Unit
Carbon Intensity Per FTE Employee	0	0	0.0000139	0.0000125	0.0000111	tCO2e/ FTE
Carbon Intensity Per Unit of Revenue	64	66	67	69	72	tCO2e/ million US\$
Proportion of Renewable Electricity Used/Purchased (%)	96	100	100	100	100	%
Scope 1	107452	99008	117956	118100	123704	tCO2e
Scope 2 Location-Based	2697554	2946043	3557518	4102445	4745197	tCO2e
Scope 2 Market-Based	139066	183329	275420	228194	163935	tCO2e
Scope 3	10662000	11745000	11466000	11239000	13785000	tCO2e
Total Electricity Used/Purchased	6344700	7357636	8744834	10244377	12969393	MWh
Total Energy Used/Purchased	6756779	7781383	9249361	10757166	13481863	MWh
Total Renewable Electricity Used/Purchased	6344700	7357636	8744834	10244377	12969393	MWh
Waste Generated	26059	19066	46178	40570	22518	Metric Tons
Water Consumption	1913	3326	3946	3967	4478	Million Gallons
Water Discharge	3236	3393	3559	3651	3179	Million Gallons
Water Withdrawal	5148	6719	7505	7618	7657	Million Gallons

Table 8.8: Microsoft Extracted Data.

For the success of data extraction for Microsoft, the process scores 100%, showing the effectiveness of the data extraction process where the format of the document aligns with the system properly, even though fewer actual KPIs were collected in this case, this was due to a difference in what Microsoft published in their figures.

Problematic Reports

Though the KPI matching of the AMD data was very successful, some issues arose during the extraction itself. This was because tables in the AMD report were split across multiple tables. Therefore as shown in figure[Fig 8.2], there is no years column to match the indicators with. Therefore, this causes issues where the KPI has been matched, but there is no corresponding correct year for the tuple, meaning that the data will not be inserted so as not to insert

incorrect data.

Shanghai	8	7	8	10	11
Singapore	17	15	17	17	16
Sunnyvale	13	nan	nan	nan	nan
Other sites combined	5	5	5	5	5
Electricity (Indirect Energy, GWh)	122	116	120	116	101

Figure 8.2: AMD Table Extracted From Report.

Another scenario where there was an issue in extracting the data from the tables was in the LYFT 2022 report. As shown in the figure below [Fig 8.3]. A single year appears in multiple columns, with data for each of the metrics not just split into their respective years. This means that the system does not correctly identify the year row; therefore, once the KPIs are matched, it cannot insert the entry into the database since it would match the value to an incorrect year value.

Lyft GHG Emissions Inventory	2021 Location-Based	2021 Market-based	2020 Location-based	2020 Market-based
	metric tons CO2e	metric tons CO2e	metric tons CO2e	metric tons CO2e
Scope 1 Emissions	1,022	1,022	718	718
Natural Gas Consumption	1,022	1,022	718	718
Scope 2 Emissions Location-Based	6,167	-	8,191	-
Purchased Heating	795	-	1,650	-
Purchased Electricity	5,372	-	6,541	-
Scope 2 Emissions Market-Based (net)	-	795	-	1,650
Purchased Heating	-	795	-	1,650
Purchased Electricity	-	4,267	-	6,212
Purchased Electricity (applied RECs)	-	-4,267	-	-6,212
Scope 3 Emissions	1,490,752	1,488,067	1,357,212	1,355,642
Air Conditioning Refrigerant Leaks	1,533	1,533	1,514	1,514

Figure 8.3: Example Table - LYFT 2022.

Conclusion

The data extraction process demonstrated different levels of success in extracting the data from different reports. In cases where the format aligned well with the system, such as Google 2022 (8.1.3) and Microsoft 2021 (8.1.3), the reports achieved a 100% success rate showing the effectiveness of the extraction process under an optimal scenario.

However, unfortunately, not all reports are in an optimal representation. Two key cases where the reports presented problematic representations were for AMD 2021 report 8.2 and the LYFT 2022 report 8.3 with multiple columns for a single year, the system faced difficulties in accurately extracting and associating the values for the KPIs with the correct years. This highlighted the limitations of the current process when dealing with an unconventional structure and emphasised the need for further development and improvement to handle different data representations.

Overall, the data extraction process proved very effective in handling well-structured tables (present in most reports); however, it requires refinement to deal with more complex representations. Improving this further would significantly improve its overall performance and versatility for extracting data from a broader range of reports.

8.1.4 Report Generation

Finally, this section of the evaluation will be based on feedback from a survey of 10 users who submitted feedback for each of the graphs anonymously via a Google Form. Since there are variations in how much of the data is published by each company, this will also be considered. For this case, the evaluation will go through the created Google Report.

Google 2022

For each graph, the user can hover their mouse over the graph to view the exact value of a point; they can toggle a series on the graph on and off to get a more detailed view of the different series present.

Below in figure [Fig 8.4] is the graph for scoped emissions data for Google. This graph shows a total of 5 series Scope 1 Emissions, Scope 2 - Market-Based Emissions, Scope 2 - Location Based Emissions, Scope 3 Emissions and Total (Scope 1,2 - Market-Based and Scope 3) Emissions. This graph effectively displays the change in the performance over time for Google and effectively

highlights each of the different KPIs to the reader.

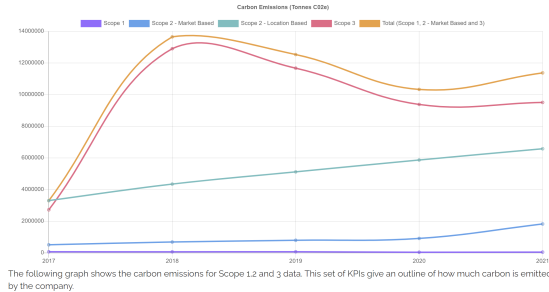


Figure 8.4: Google Reporting - Scoped Data Graph

Shown in figure [Fig 8.5] are the metrics for different KPIs relevant to the use of water usage of the company.

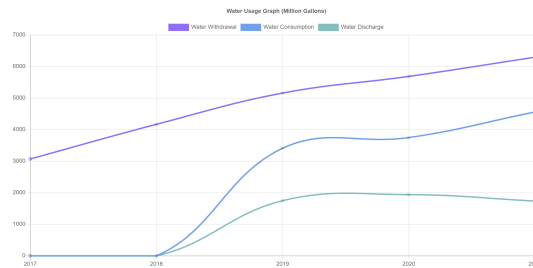


Figure 8.5: Google Reporting - Water Graph

Shown in figure [Fig 8.6] are the metrics for different KPIs relevant to the company's energy use. Please note that a series is present for Renewable Energy Use; however, this is obscured by the Energy Use series since they are the same value. However, the series can be toggled on and off to reveal the other.

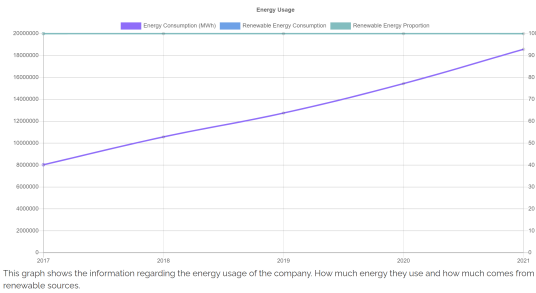


Figure 8.6: Google Reporting - Energy Data Graph

Figure [Fig 8.7] shows the metrics for different KPIs relevant to the company’s energy use. Please note there is a series present for Renewable Electricity Use; however, this is obscured by the Electricity Use series since they are the same value. However, the series can be toggled on and off to reveal the other.

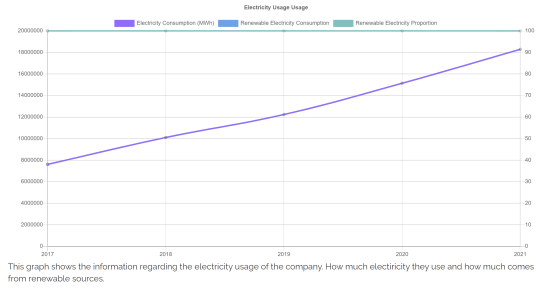


Figure 8.7: Google Reporting - Electricity Data Graph

Shown in figure [Fig 8.8] is the graph for the company’s carbon emissions normalised against different metrics such as per Full Time Employee, Unit Revenue, and MWh of Energy Used. This gives the user a different view of the company’s impact on the environment against different metrics.

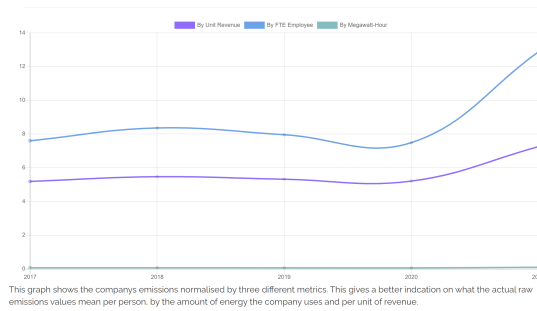


Figure 8.8: Google Reporting - Energy Intensity Data Graph

Figure [Fig 8.9] shows the complete set of results for Google; all of these figures are extracted purely from the 2022 report, giving five years of historical data from a single reports scan. Please note that the years published in different companies' reports may be between 2 to 5 years of historical data.

Key Performance Indicator	2017	2018	2019	2020	2021	Unit
Carbon Intensity Per FTE Employee	7.6	8.36	7.96	7.49	12.87	tCO2e/ FTE
Carbon Intensity Per Megawatt-Hour of Energy	0.0717	0.0707	0.0675	0.0615	0.1006	tCO2e/ per megawatt-hour of energy consumed
Carbon Intensity Per Unit of Revenue	5.19	5.47	5.32	5.21	7.25	tCO2e/ million US\$
Landfill diversion rate (%)	83	80	77	77	77	%
Proportion of Renewable Electricity Used/Purchased (%)	100	100	100	100	100	%
Proportion of Renewable Energy Used/Purchased (%)	100	100	100	100	100	%
Scope 1	66549	63521	66686	38694	45073	tCO2e
Scope 2 Location-Based	3301392	4344686	5116949	5865095	6576239	tCO2e
Scope 2 Market-Based	509334	684236	794267	911415	1823132	tCO2e
Scope 3	2719024	12900467	11669000	9376000	9503000	tCO2e
Total Electricity Used/Purchased	7609089	10104295	12237198	15138543	18287143	MWh
Total Energy Used/Purchased	8029409	10572486	12749458	15439538	18571659	MWh
Total Renewable Electricity Used/Purchased	7609089	10104295	12237198	15138543	18287143	MWh
Waste Generated	53363	57113	48126	28864	28153	Metric Tons
Water Consumption	0	0	3412	3749	4562	Million Gallons
Water Discharge	0	0	1749	1940	1735	Million Gallons
Water Withdrawal	3071	4170	5161	5689	6297	Million Gallons

Figure 8.9: Google Reporting - Complete Data

Finally, shown in figure [Fig 8.10] are the rankings for each of the KPIs for Google compared to other companies in the cohort. In some cases, Google is

ranked first; this will be either where Google is the only company to publish this figure in their report, or the case of the KPI - Proportion of Renewable Energy (%), this will be because multiple companies use 100% renewable energy.

Key Performance Indicator	Unit	Rank
Carbon Intensity Per FTE Employee	tCO2e/ FTE	58
Carbon Intensity Per Megawatt-Hour of Energy	tCO2e/ per megawatt-hour of energy consumed	1
Carbon Intensity Per Unit of Revenue	tCO2e/ million US\$	56
Landfill diversion rate (%)	%	2
Proportion of Renewable Electricity Used/Purchased (%)	%	5
Proportion of Renewable Energy Used/Purchased (%)	%	1
Scope 1	tCO2e	46
Scope 2 Location-Based	tCO2e	41
Scope 2 Market-Based	tCO2e	36
Scope 3	tCO2e	32
Total Electricity Used/Purchased	MWh	3
Total Energy Used/Purchased	MWh	7
Total Renewable Electricity Used/Purchased	MWh	6
Waste Generated	Metric Tons	5
Water Consumption	Million Gallons	4
Water Discharge	Million Gallons	1
Water Withdrawal	Million Gallons	2

Figure 8.10: Google Reporting - KPI Ranking Data

Summary

The table below 8.9 contains scores out of 10, 0 being the worst and 10 best. This gives a mean ranking of 9.14/10 across the different graphs and reports. Some further comments made to improve the reports were to allow company-by-company comparisons and allow the user to query the data further. Therefore, overall the reporting section of the project was very successful, given the data for a company and the associated set of KPIs gives a clear indication of the ranking of the company across the set of measured companies; it outlines clearly how different metrics change over time through the use of clear, interactive graphs giving the user an immersive experience into the data extracted from the report.

Report Component	7/10	8/10	9/10	10/10
Scoped Data (8.4)	0	1	3	6
Water Data (8.5)	0	2	5	3
Energy Data (8.6)	1	1	4	4
Electricity Data (8.7)	1	1	4	4
Intensity Data (8.8)	0	2	6	2
Complete Data (8.9)	0	2	3	5
KPI Ranking Data (8.10)	1	2	4	3

Table 8.9: Component Ratings

8.1.5 Conclusion

The evaluation of the different components of the system has demonstrated the effectiveness and potential of the developed system in multiple aspects.

The table extraction process was successful, with a 100% extraction rate across all tested reports. Despite this, extraction time was significantly increased as page and table count increased, with a clear correlation between these factors.

The matching system proved to be very versatile, learning various representations with high accuracy as it learned from different company reports. Despite this, some level of user input is required in case there is a new or different representation of the KPI in the table; however, the goal of minimising the need for user oversight was achieved.

Regarding data extraction, there were varying levels of success, from very high, where the representation was optimal to the system, to very low, depending on if the representation differed too significantly. This emphasised a key development area in order to handle a broader range of reports and representations to create a more versatile system.

The report generation component received an overall rating of 9.14/10, highlighting the effectiveness of the clear, interactive graphs and providing an immersive experience to the user. Whilst there were improvements mentioned, such as allowing for company-by-company comparisons, the overall success of this section is apparent.

In conclusion, this system has achieved high success in its goals and provides a strong foundation for further development and improvement. By addressing the improvements - particularly with handling more complex data representations, this system has a clear potential to be an invaluable tool for extracting and measuring the environmental commitment of companies, particularly as new regulations implemented by the SEC are implemented.

8.2 Requirements Success

Though this project was predominantly based on the research and identification of a process to gain insight into the question presented by the project title, there were also aspects of software engineering that needed to be included and implemented.

In this case, this was how the user interacted with the system through a graphical user interface. Since the development of a system relies on various components, this section will evaluate the system based on how well the project met the requirements outlined in section 4.

Requirement	Met	Explanation
R1	Yes	As outlined in table (3.1, a set of environmental KPIs were defined, and data was extracted for various companies. A table containing a set of KPIs and the data extracted along with those is shown in figure [Fig 8.9].
R2	No	This requirement was not met. This is of small effect to the overall evaluation since the urgency for this requirement was marked as Won't. Furthermore, analysis of social KPIs requires much further analysis of the text.
R3	No	This requirement was not met. This is again of small effect to the overall success since its urgency was marked as 'Could'. Additionally, not all companies presented this data in their data tables. Therefore the analysis of this data would require an entirely different approach to extract data across various report structures.
R4	Yes	As outlined in section 8.1.1, the tables within the reports were extracted with a 100% success rate (in the tests carried out).

R5	Yes	As shown in section 8.1.1, once the tables have been extracted from the report, they are presented to the user one by one, where the user can identify which table contains the relevant data.
R6	No	This requirement was not implemented due to the difficulty of automated identification of the tables, and various methods tested gave incorrect results. However, this is no detriment to the project’s overall success since the requirement was marked with urgency ‘Could’.
R7	Yes	As shown in section 6.2.3, the system was able to identify the KPIs within the table, either automatically or with user oversight.
R8	Yes	As shown in section 6.2.3, the system could identify the KPIs correctly within the table once it had learned queries of sufficient similarity. If there was no previously analysed query which matched
R9	Yes	As shown in section 8.1.3, the system was able to extract data with an extremely high success rate when the data format was optimal- covering a range of different representations. However, where the tabular data format differed significantly with a more complex representation, here, the system’s success was limited.

R10	Yes	As shown in section 8.1.4, a report containing all the information extracted for the company was presented to the user.
R11	Yes	As outlined in section 8.1.4, a range of graphs containing data on different KPIs are presented to the user where the information is present.
R12	Yes	As shown in section 8.1.4, each of the graphs allowed the toggling of different series to narrow the scope to a single KPI as well as hovering a mouse over specific data points to view the exact value of the KPI extracted.
R13	Yes	As shown by figure [Fig 8.9], an example of a complete set of results for the data collected for a company is present. A table is created for each company analysed when the user selects the company.
R14	Yes	As shown by figure [Fig 8.10], a ranking for each of the KPIs is present in a table, identifying to the user where the company stands compared to the rest of the cohort for that particular indicator.

R15	Yes	As shown in section 8.1.4, each company is given an overall ranking based on the ranking of each of the KPIs. Though this is a primitive ranking system, not considering the weighting of each of the KPIs, there was a significant disparity in the set of KPIs published, particularly when considering smaller companies; therefore, this method is sufficient for the meantime.
R16	No	This was not implemented due to time constraints; however, since this requirement was marked with urgency 'Could', this is no detriment to the system's success.

Table 8.10: Requirements Success Assessment

As shown in the table above 8.10, this project successfully met all of the defined 'Must' requirements. Some 'Should' requirements were also met, which further demonstrated the system's effectiveness.

Despite not meeting requirements R2, R3, R6 and R16, the project's overall success was not impacted significantly, as these requirements had lower priority levels - 'Could' and 'Won't'. The set of environmental KPIs were defined in table 3.1 (R1); the system was able to identify the tables containing these KPIs with the assistance of the user (R5), then identify these KPIs successfully (R7, R8). From this, the system could generate comprehensive reports for the user, which they could successfully interact with (R10-R15).

Despite this, though Requirement (R9) was met, there were still limitations to its success when dealing with complex table representations or representations which differ significantly from the desired format. However, since the majority of reports had a format which matched the desired format, this effect was marginal.

In conclusion, despite the difficulties in R9, this system succeeded overall and provided a strong baseline for further development and improvement. Whilst still providing a portal for analysis of companies on their Environmental impact based on their ESG reports - which will become mandatory for the 2023 fiscal year.

8.3 Findings

From the data collected from these reports, there were some associated findings. Firstly, some negatives, the raw impact of these corporations on the globe is vast, with lots of them releasing millions of tonnes of carbon into the atmosphere. Showing that they must lead the shift towards a more sustainable future by leading by example with the power and responsibility they have. Furthermore, since it is predominantly larger companies producing more reports, this means that typically, a larger company may rank better than a smaller company since the number of published indicators is much larger, 15 instead of 5, for example.

However, the findings were not only negative; lots of companies had 100% renewable energy and electricity usage, particularly for larger companies. And those who did not do this have goals to achieve this. Therefore, even though

there is a long way to go before we are at a sustainable place, there is a clear aim and goal to do so.

8.4 Limitations

Even though this project resulted in a complete system with the capability to analyse reports and extract data on a wide range of environmental information, there are some key limitations.

Firstly, the scope of data is only environmental. If there was the ability to include social and governance data in this report this would significantly increase the capability of the system

Furthermore, there is a reliance on ExtractTable. This is a paid service; therefore, for this project to be sustainable, either the table extraction and OCR must be done within the system itself or have some funding (less than £200 a year based on analysing a report from each of the 100 companies each year).

Finally, the capability to analyse the entire report rather than just the tables within the report. A report of over 100 pages will have an order of magnitude of data than what is stored in the tables, therefore, a more complete view of each company could be determined with this capability.

Chapter 9

Conclusions

Overall, this project is a success. This project began as an idea to understand further the field I was very interested in and relevant to my future - the impact of 'Big-Tech' companies on the environment and sustainability. Even though I reduced the project scope to accommodate challenges, I have created a system that allows for a robust, repeatable process of extracting environmental data from PDF reports. Furthermore, this project draws on various technologies and frameworks that are new to me, furthering my technical skills. In addition to this, my soft skills, such as project management communication with my supervisor and the ability to research and identify new innovative techniques to solve a problem, advanced significantly.

Of course, there are limitations and areas where the project could be improved, with a significantly reduced scope of analysis, the limitation to only analysing data within the tables of the report and the reliance on an external service. The project has a long way to go before it is ready to be released as a full-service or fully-functioning platform. However, this is a project I intend to continue

developing and working on in the future.

In conclusion, this project has provided me with a deeper insight into the world of 'Big-Tech', furthering a wide range of my skills and other potential users interested in the same field. Therefore, I intend to further develop this project over the rest of my education and post-graduation, working towards a more advanced solution providing more in-depth insights and value to its users.

9.1 Future work

Completing this project comes at a key time, with the implementation of the new SEC regulations on ESG reporting; this means the requirement for effective measuring and extraction from the reports of companies. However, this project is still a foundation and can be significantly built on in multiple ways.

First of all, the scope of analysis. Currently, the project only considers the environmental factors of the report. If social and corporate governance data were to be included and analysed through various language analysis methods, this would allow a complete evaluation of the commitment to the ESG goals of the companies, which means that a single ESG report from a company is used to give a well-rounded evaluation of the impact of the company. Furthermore, if this is done yearly (each time a company releases their report), the user can perform a year-on-year analysis of companies, showing their progress and change.

Another improvement area is adding further automation. Currently, the project still requires heavy human interaction. This could be reduced or eliminated at

various points of the system. Firstly, in collecting the reports. If reports for a designated set of companies were collected once a year via web-scraping, for example, this would reduce the requirement for human interaction, further streamlining the process, and secondly, regarding the relevant table identification. Developing a way to identify which tables contained pertinent data to the desired information would mean that after a report was scanned. The KPIs' values could be collated into a report for the user without additional user input. However, since further automation increases the risk of incorrect data being extracted or relevant data being missed, this would need to be done with significant care and moderation to ensure the correctness of any data presented.

Another way to improve is the integration of news reports and articles relevant to each company regarding ESG goals. Using NLP techniques such as Topic Modelling on news articles relevant to a set of companies would provide more frequent updates and insights. This, integrated with the system as a whole, would result in an in-depth, up-to-date portal allowing the analysis of a set of companies specified by a user.

In conclusion, the project's potential to significantly further ESG reporting and analysis comes from three main factors, expanding scope, enhancing automation, and incorporating external data sources. First, by including social and corporate governance data, automating the report collection and table identification process and conducting annual comparisons, the project would provide comprehensive ESG evaluations of companies. Furthermore, integrating news reports and articles using NLP techniques and emerging language models into the report analysis will further enrich this process, providing real-time insights

to the users. Although the apparent challenges in retaining data accuracy will remain, this set of advancements will contribute to a robust platform for assessing progress in companies' ESG performance.

9.2 Authors Assessment

The technical contribution of the project is bringing together a wide range of resources, tools and methods to produce a system which lays a foundation for the environmental analysis of companies. Relevant to computer science, various programming languages, frameworks and libraries and project management skills were used. Others can use the work to gain an in-depth understanding of the environmental impact of companies. This should be considered an achievement because this project started out as a challenge where I had no idea how to solve this problem, and through research, learning and commitment to the goal produced a functioning system. However, limitations include a scope reduction to environmental data, the requirement for user interaction and the reliance on external services.

Bibliography

- [1]. Beardsley, Elizabeth. Federal and state policies impacting esg reporting could be issued in 2023, Mar 2023. URL <https://www.usgbc.org/articles/federal-and-state-policies-impacting-esg-reporting-could-be-issued-2023>. Accessed: 16-04-2023.
- [2]. Bloomberg, . Bloomberg terminal | bloomberg professional services, 2023. URL <https://www.bloomberg.com/professional/solution/bloomberg-terminal/>. Accessed: 06-03-2023.
- [3]. Bock, Emma. What are market based and location based scope 2 emissions?, Apr 2023. URL <https://support.measurabl.com/hc/en-us/articles/4406903207565-What-are-Market-Based-and-Location-Based-Scope-2-emissions->. Accessed: 20-04-2023.
- [4]. Bootstrap, . Bootstrap - introduction, 2023. URL <https://getbootstrap.com/docs/5.0/getting-started/introduction/>. Accessed: 20-04-2023.
- [5]. Bynes, Grace. The impact of infographics on user engagement and retention, Apr 2023. URL <http://www.startupguys.net/impact-of-infographics-on-user-engagement>. Accessed: 18-04-2023.
- [6]. Catania, Patrick J. & Keefer, Nancy. Idealratings esg metric data-

- set, 2022. URL <https://aws.amazon.com/marketplace/pp/prodview-u3d5msmaoiqw#offers>. Accessed: 20-04-2023.
- [7]. ChartJS, . Chart.js, Oct 2023. URL <https://www.chartjs.org/docs/latest/>. Accessed: 14-04-2023.
- [8]. Chevron Policy, Government & Affairs, Public. Explainer: What is carbon intensity?, Nov 2022. URL <https://www.chevron.com/newsroom/2022/q4/explainer-what-is-carbon-intensity>. Accessed: 02-04-2023.
- [9]. Chipchase, Leon. 2022.
- [10]. Cohen, Adam. Fuzzywuzzy, 2022. URL <https://pypi.org/project/fuzzywuzzy/>. Accessed: 20-04-2023.
- [11]. Conner, Cheryl. The data is in: Infographics are growing and thriving in 2017 (and beyond), Oct 2017. URL <https://www.forbes.com/sites/cherylsnappconner/2017/10/19/the-data-is-in-infographics-are-growing-and-thriving-in-2017-and-beyond/?sh=5113c182137c>. Accessed: 18-04-2023.
- [12]. Craddock, Andrew & Craddock, Andrew. *Chapter 10.5*, page 70â76. DSDM Consortium, 2014.
- [13]. Delubac, Arnaud. What is esg data and how to use it?, Mar 2023. URL <https://greenly.earth/en-us/blog/company-guide/what-is-esg-data-and-how-to-use-it>. Accessed: 20-03-2023.
- [14]. Developers, NumPy. What is numpy?, 2022. URL <https://numpy.org/doc/stable/user/whatisnumpy.html>. Accessed: 20-04-2023.
- [15]. Di-Gregorio, Federico & Varrazzo, Daniele. Psycopg2, 2022. URL <https://pypi.org/project/psycopg2/>. Accessed: 16-04-2023.

- [16]. Echua, . Python - developing web applications with flask, 2017. URL https://www3.ntu.edu.sg/home/ehchua/programming/webprogramming/Python3_Flask.html.
- [17]. EPA, . Scope 1 and scope 2 inventory guidance, Sep 2022. URL <https://www.epa.gov/climateleadership/scope-1-and-scope-2-inventory-guidance>. Accessed: 04-04-2023.
- [18]. ExtractTable, . Extract table - api docs, 2023. URL <https://documenter.getpostman.com/view/6396033/SVfMS9xu>. Accessed: 14-04-2023.
- [19]. Few, Stephen. *Now you see it: Simple visualization techniques for quantitative analysis*. Analytics Press, 2009.
- [20]. Flask, . Welcome to flask, 2023. URL <https://flask.palletsprojects.com/en/2.1.x/>. Accessed: 10-04-2023.
- [21]. Gaspar, Daniel & Stouffer, Jack. *Mastering Flask web development: Build enterprise-grade, Scalable Python Web Applications*. Packt Publishing, 2018.
- [22]. Grinberg, Miguel. *Flask Web Development*. O'Reilly, 2014.
- [23]. Grinberg, Miguel. *Flask Web Development: Developing Web Applications with Python*. Web Development. O'Reilly Media, 2nd edition, 2018. ISBN 9781491991732; 1491991739.
- [24]. Hiem, Sabine & Keil, Andreas. Too much information, too little time: How the brain separates important from unimportant things in our fast-paced media world, Jun 2017. URL <https://kids.frontiersin.org/articles/10.3389/frym.2017.00023>. Accessed: 16-04-2023.
- [25]. Inc, NumFocus. Pandas documentation, Apr 2023. URL <https://pandas.pydata.org/docs/>. Accessed: 20-04-2023.

- [26]. Insider, Law. Water discharges definition, 2023. URL <https://www.lawinsider.com/dictionary/water-discharges>. Accessed: 20-04-2023.
- [27]. Jinja, , 2022. URL <https://jinja.palletsprojects.com/en/3.1.x/>.
- [28]. Klipfolio, . What is a key performance indicator (kpi)?, Apr 2023. URL <https://www.klipfolio.com/resources/articles/what-is-a-key-performance-indicator>. Accessed: 20-04-2023.
- [29]. Lane, Stephen & O'Raghallaigh, Paidi & Sammon, David. Requirements gathering: the journey. *Journal of Decision Systems*, 25(sup1):302–312, 2016. doi: 10.1080/12460125.2016.1187390. URL <https://doi.org/10.1080/12460125.2016.1187390>.
- [30]. Maia, Italo. *Building web applications with flask: Use python and flask to build amazing web applications, just The way you want them!* Packt Publishing Ltd., 2015.
- [31]. Mathis, Sandra. What is environmental, social and governance (esg)?, Mar 2023. URL <https://www.techtarget.com/whatis/definition/environmental-social-and-governance-ESG>. Accessed: 16-04-2023.
- [32]. Munzner, Tamara. *Chapter 7 Arrange Tables*, page 144â175. CRC Press, Taylor amp; Francis Group, 2015.
- [33]. Navarro, Gonzalo. A guided tour to approximate string matching. *ACM Computing Surveys* 2001-mar vol. 33 iss. 1, 33, mar 2001. doi: 10.1145/375360.375365.
- [34]. OECD, . Water withdrawals | water | oecd ilibrary, 2023. URL https://www.oecd-ilibrary.org/environment/water-withdrawals/indicator/english_17729979-en.

- [35]. OpenAI, , 2023. URL <https://openai.com/research/gpt-4>. Accessed: 15-03-2023.
- [36]. PyMuPDF, . Welcome to pymupdf, Apr 2023. URL <https://pymupdf.readthedocs.io/en/latest/>. Accessed: 18-04-2023.
- [37]. R, Mageshwaran. Survey on image preprocessing techniques to improve ocr accuracy, Jul 2021. URL <https://medium.com/technovators/survey-on-image-preprocessing-techniques-to-improve-ocr-accuracy-616ddb931b76>. Accessed: 14-03-2023.
- [38]. Regina O. Obe, Leo S. Hsu. *PostgreSQL. Up and Running. A Practical Guide to the Advanced Open Source Database*. O'Reilly Media, 3rd edition edition, 2017.
- [39]. Reig, Paul. What's the difference between water use and water consumption?, Mar 2013. URL <https://www.wri.org/insights/whats-difference-between-water-use-and-water-consumption>. Accessed: 04-04-2023.
- [40]. ReWorksSA, . Calculating your diversion rate, 2023. URL <https://www.reworkssa.org/Toolkit/Diversion-Rate>. Accessed: 04-04-2023.
- [41]. Schwaber, Ken. *Agile Project Management with scrum*. Microsoft, 2015.
- [42]. SEC, . Press release sec proposes rules to enhance and standardize climate-related disclosures for investors, Mar 2022. URL <https://www.sec.gov/news/press-release/2022-46>.
- [43]. Timalsina, Amit. Pdf data extraction - how to capture tables from pdf/images?, Apr 2023. URL <https://www.docsumo.com/blog/pdf-table-extraction>. Accessed: 19-04-2023.

- [44]. Tocchini, Fabrizio & Cafagna, Grazia, Jul 2022. URL <http://www.wolterskluwer.com/en/expert-insights/the-5-biggest-hurdles-to-effective-esg-reporting>. Accessed: 20-03-2023.
- [45]. Trust, Carbon. Briefing: What are scope 3 emissions?, Mar 2023. URL <https://www.carbontrust.com/our-work-and-impact/guides-reports-and-tools/briefing-what-are-scope-3-emissions?https%3A%2F%2Fwww.carbontrust.com%2Four-work-and-impact%2Fguides-reports-and-tools%2Fbriefing-what-are-scope-3-emissions%3>. Accessed: 10-04-2023.
- [46]. United-Nations, . Key aspects of the paris agreement. URL <https://unfccc.int/most-requested/key-aspects-of-the-paris-agreement>. Accessed: 1-04-2023.
- [47]. VinciWorks, . Is esg reporting mandatory in the uk, eu amp; us?: Vinciworks, Sep 2022. URL <https://vinciworks.com/blog/is-esg-reporting-mandatory-in-the-uk-the-eu-and-the-us/>. Accessed: 20-03-2023.
- [48]. Wikimedia, . Levenshtein distance, Apr 2023. URL https://en.wikipedia.org/wiki/Levenshtein_distance. Accessed: 14-04-2023.