WINTER TRAINING REPORT

OF

SCIENTIFIC DOCUMENT ANALYZER

Defence Scientific Information & Documentation centre

Defence Research and Development Organisation

Metcalfe House, Civil Lines, Delhi – 110054

in partial fulfillment of the degree

Bachelor of technology

In

Information Technology and Engineering

Maharaja Agrasen Institute of Technology

Submitted by

Akash Jha

Under the guidance of

SHRI DEEPAK KUMAR VERMA (SCIENTIST D)

DECLARATION

This is to certify that Winter Training report entitled of SCIENTIFIC DOCUMENT ANALYZER which is submitted by Akash Jha in partial fulfillment of the requirement for the award of degree B.Tech in Department of Information Technology and Engineering of Maharaja Agrasen Institute of Technology, is a record of the candidate own work carried out by him under my/our supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree. I hereby declare that this submission was his own work and that, to the best of his knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute or other institute of higher learning, except where due acknowledgement has been made in the text.

SIGNATURE

SHRI DEEPAK KUMAR VERMA (SCIENTIST D)

ACKNOWLEDGEMENT

I record my sincere thanks to the Director "Dr K Nageswara Rao" Defence Scientific Information & Documentation centre Defence Research and Development Organisation for providing me an excellent opportunity to undergo training in his esteemed organization through which I could gain an exposure to the Research and Development environment and getting acquainted with the software. I would sincerely like to thank SHRI DEEPAK KUMAR VERMA, Scientist 'D' for permitting me to work in his division and making me aware of recent trends and technologies. I express my thanks to all the members of DESIDOC group for providing me support and their constant encouragement during training. Last but not the least, I would like to thank the library staff for allowing me to access the library and its valuable material.

AKASH JHA

About DRDO

Defence Research & Development Organisation (DRDO) works under Department of Defence Research and Development of Ministry of Defence. DRDO dedicatedly working towards enhancing self-reliance in Defence Systems and undertakes design & development leading to production of world class weapon systems and equipment in accordance with the expressed needs and the qualitative requirements laid down by the three services.

DRDO is working in various areas of military technology which include aeronautics, armaments, combat vehicles, electronics, instrumentation engineering systems, missiles, materials, naval systems, advanced computing, simulation and life sciences. DRDO while striving to meet the Cutting edge weapons technology requirements provides ample spinoff benefits to the society at large thereby contributing to the nation building.

Vision

Make India prosperous by establishing world-class science and technology base and provide our Defence Services decisive edge by equipping them with internationally competitive systems and solutions.

DRDO have many research and development labs spread across the country. The different R&D labs are meant to work on different projects.one of the lab of DRDO is DESIDOC (Defence Scientific Information & Documentation centre) Metcalfe House, Civil Lines, Delhi - 110054.

The Defence Scientific Information & Documentation Centre (DESIDOC) is a division of the Defence Research and Development Organisation (DRDO). Located in Delhi, its main function is the collection, processing and dissemination of relevant technical information for DRDO scientists. The present director of DESIDOC is "Mr. K Nageswara Rao."

- Graphic designing and digital printing
- Design and development of web-based knowledge repositories; digital data storage and retrieval; e-publishing and e-library
- End-to-end multimedia services
- IT enabled services through DRDO Intranet and DRDO Website.

CONTENTS

LIST OF FIGURES

Abstract

Data extraction refers to the process of retrieving information from unstructured or poorly structured data sources for further processing or storage. This typically involves importing experimental data into a computer from primary sources such as measuring or recording devices. The imported data is then transformed and augmented with metadata before being exported to other stages in the data workflow. Techniques for adding structure to unstructured data include text pattern matching, table-based approaches, and text analytics.

Python is a high-level, general-purpose programming language that emphasizes code readability through significant indentation. It supports multiple programming paradigms, including structured, object-oriented, and functional programming. Python's comprehensive standard library has earned it a reputation as a "batteries included" language.

Python includes several key programming constructs, such as the assignment statement, if statement (including else and elif), for statement, while statement, try statement (with new syntax except* in Python 3.11 for exception group), raise statement, class statement, def statement, with statement, break statement, continue statement, del statement, pass statement, assert statement, yield statement (used to implement coroutines), and return statement. The import and from statements are also used to import modules whose functions or variables can be used in the current program.

Data extraction is a critical process in data management and involves retrieving data from unstructured or poorly structured data sources. The extracted data is then processed and transformed for storage or further analysis.

Data extraction can be performed using various techniques such as text pattern matching, table-based approaches, and text analytics.

Python, on the other hand, is a high-level programming language that supports multiple programming paradigms, including object-oriented, functional, and structured programming. Python is known for its code readability, significant indentation, and comprehensive standard library, which make it a popular language among developers. Python has various constructs and statements such as the

assignment, if, for, while, try, raise, class, def, with, break, continue, del, pass, assert, yield, import, and from statements that developers can use to perform various tasks.

Understanding data extraction and programming constructs in Python is essential for data management and software development.

Introduction

Research papers are a valuable source of information for scientists and researchers to gather knowledge and insights in their respective fields. However, research papers often contain a large amount of data, making it difficult to extract relevant information from them. This is where data abstraction comes in. Data abstraction is the process of filtering out unnecessary data from a research paper to extract the relevant information needed for analysis.

Data abstraction involves the use of techniques such as data extraction, data transformation, and data loading. Data extraction involves identifying and collecting relevant data from the research paper, while data transformation involves the conversion of the extracted data into a structured format for easy analysis. Data loading involves the storage and management of the extracted and transformed data in a suitable system.

The process of data abstraction is crucial in research as it allows researchers to focus on the relevant information and draw meaningful insights from it. In addition, data abstraction can help in reducing the time and effort required to analyze large volumes of data. This makes it easier for researchers to gain valuable insights from research papers and use them to make informed decisions.

In this report, we will discuss the importance of data abstraction in research papers and the techniques used for data abstraction. We will also explore the challenges involved in data abstraction and the potential solutions to overcome them. Overall, the goal of this report is to provide a comprehensive understanding of data abstraction in research papers and its significance in scientific research.

The process of writing a research paper is included with a number of steps. The same is for the analysis of the research paper, to easy the process a platform is developed to extract data from the research papers. The platform helps to extract data from the research papers and show them on the website and option to download.

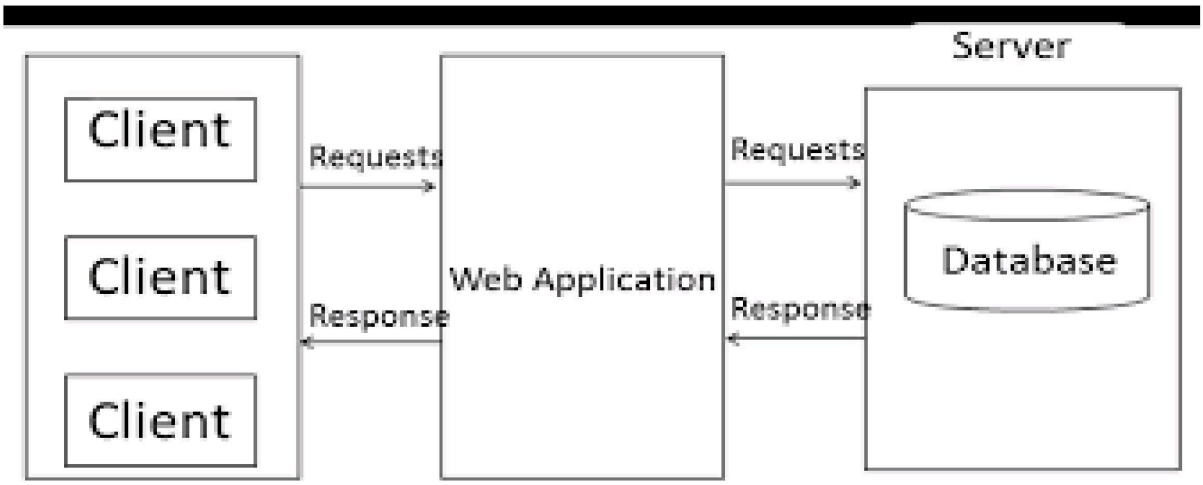

Figure 1.1 Process of the flow of Data

The above diagram shows the flow of the Software Project, the file is sent to the database and then is used by the python to extract the data and then this data is sent to the website to download. The software can be used by downloading in the local computer and used offline without internet. In this case the software will use the local storage as db.

File Upload > File stored in the Database > Data Extraction > File download on the website.

Project Undertaken

SCIENTIFIC DOCUMENT ANALYZER

The main objective of this project is to develop a web application that can extract relevant data from research papers in PDF format. The web application will use Python as a backend and React as a frontend. The specific objectives of this project include:

1. Developing a user-friendly web interface for uploading research papers in PDF format.
2. Developing a data extraction algorithm that can extract relevant data such as images, tables, and word counts from research papers.
3. Storing the extracted data in a structured format for further processing and analysis.
4. Providing the user with the ability to download the extracted data in CSV format for further analysis.

Developed a Data Extraction platform that helps to extract data from the pdf and shows the data on the website and download. The project has:

- **Full Stack Website.**
- **Python for Data Extraction**

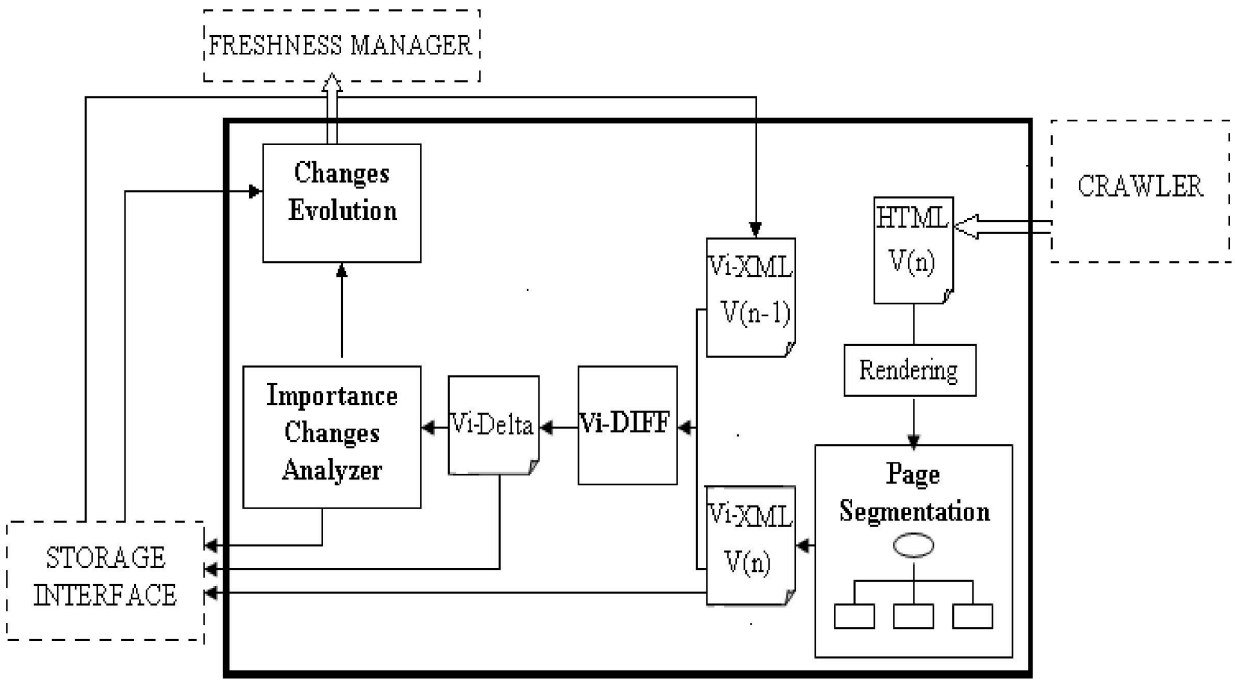

Figure 2.1 Working of inside the code

Extract Data and get information on the data with the help of this project. The data extracted and information provided by the project:

- No. of Pages
- No. of Words
- No. of Images
- No. of Tables
- Extract images in a folder
- Extract tables in a folder

Project has a website with interactive interface to upload the file. It shows the data on the website and helps to download the extracted data, images and tables in the folder.

Steps of the Extraction:

- **Upload the file to the database**
- **Extract data**
- **Show output on the website and download**

Tools Developed

There are Two major Tools Developed for the Project:

- **Website** for Uploading File.
- **Backend** that runs **Python** for **Data Extraction**.

Website:

Developed a Data Abstraction Program for Extraction of data and help in the development of Research papers. The project used Machine learning and used Python to Extract data like **No. of Pages, No. of words, No. of Images, Images, Tables. from the Research Papers**. The project also used JavaScript for the Frontend and Nodejs for the Backend part.

The project will be implemented using **Python as a backend and React as a frontend**. The backend will be responsible for data extraction, while the frontend will provide a user-friendly interface for uploading research papers and downloading extracted data.

The data extraction algorithm will use Python libraries such as **PyPDF2, pdfminer, and PIL** to extract relevant data such as images, tables, and word counts from research papers. The extracted data will then be stored in a database in a structured format for further processing and analysis.

The frontend will be developed using React, a popular JavaScript library for building user interfaces. The frontend will provide a user-friendly interface for uploading research papers and downloading extracted data. The user will be able to upload research papers in PDF format, and the data extraction algorithm will automatically extract relevant data from the research papers. The extracted data will then be displayed in a structured format on the frontend, and the user will be able to download the data in CSV format for further analysis.

The Project uses the HTML, CSS and JavaScript for the development of website to Upload the File and to check the data extracted. The data extracted can be downloaded and can also be seen in the website.

The project also uses the use of **Nodejs** for Backend of the website. The data is posted on the database with the help of the backend, then the python file is run that

imports the file and gives the output and the output is then shown with the help of NodeJS on the frontend.

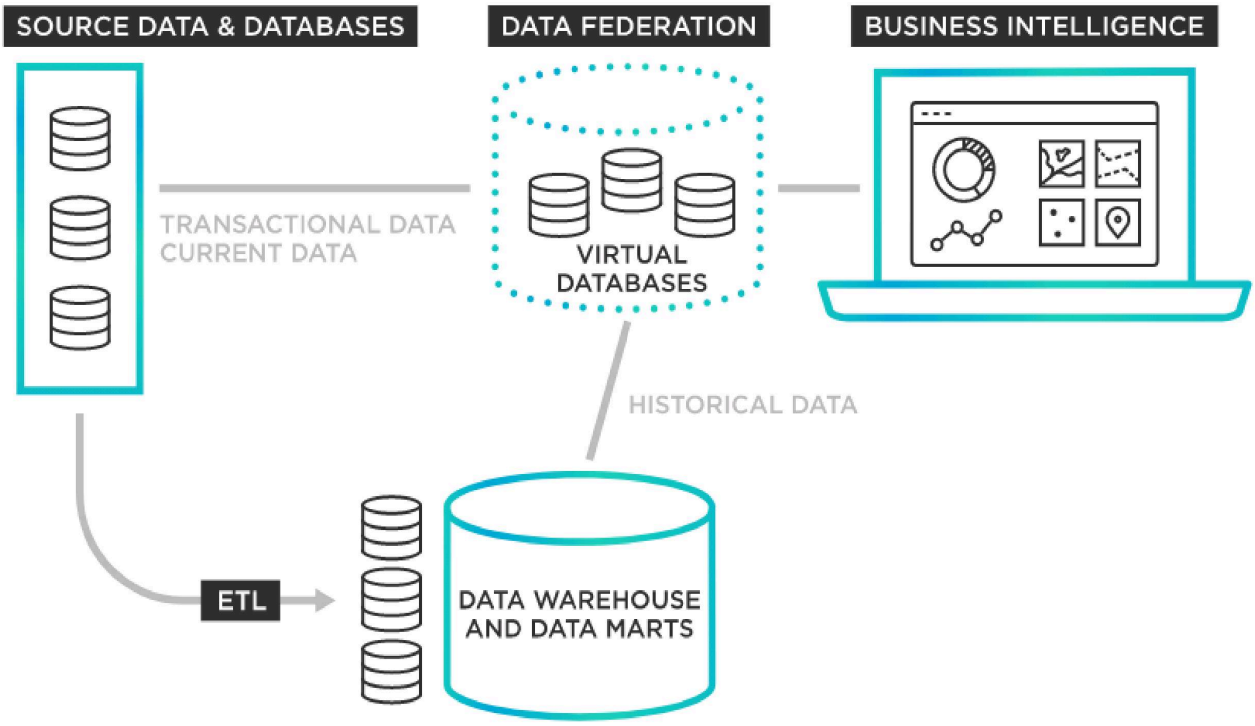

Figure 3.1 Using of virtual Database and Implementation of output

Backend:

Python file that runs on the backend takes the file as input and extracts data and returns the output. It uses libraries like **re, pdfminer.high_level, sys, Fitz, Pil. Image**, io. It extracts data from the file and shows it on the website and is available to download.

Result and Analysis

The Functions of the Project:

- The project uses the Website to Upload the File.
- The project runs python to extract data from the file.
- The project displays the output on the website.
- With the help of the project data can also be downloaded.

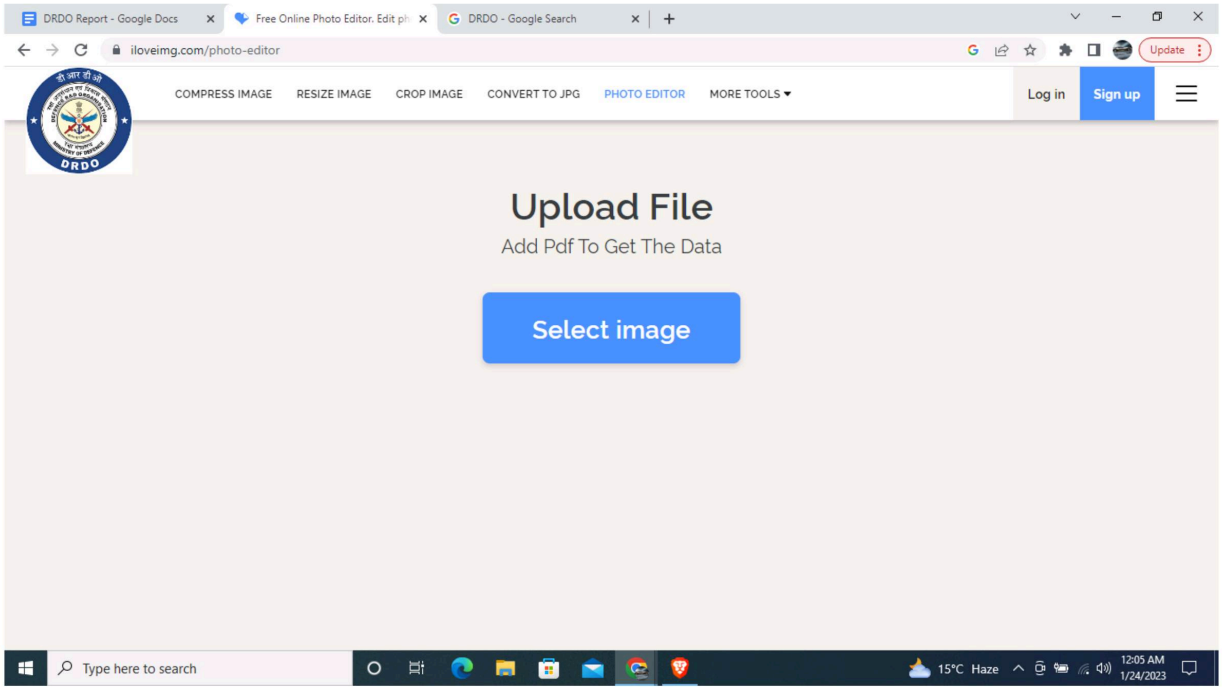

Figure 4.1 Frontpage of the project

Python Code to Extract Data:

 # Importing libarary

```
import re
from pdfminer.high_level import extract_pages, extract_text
import sys
import tabula
import fitz
import PIL.image
import io
```

 # This code for extracting text

```
text = extract_text("E:/DRDOproject2/test.pdf")
```

#Print the value of text

```
print(text)
pattern = re.compile(r"[a-zA-z]+,{1}\s{1}")
matches = pattern.findall(text)
print(matches)
print(len("test"))
```

 # image abstraction

```
pdf = fitz.open("pdf")
counter=1
for i in range(len(pdf)):
    page = pdf[i]
    images = get.extract_image(img[0])
    for image in images:
        base_img = pdf.extract(image[0])
        image_data = base_img["image"]
        img = PIL.Image.open(io.BytesIO(image_data))
```

```
        extention = base_img["ext"]
```

#Creating a pdf of all the images

```
        img.save(open(f"image{counter}.{extention}","wb"))
        counter +=1


print(counter)
```

 # Tables extraction

```
tables = tabula.read_pdf("sample.pdf", pages="all")
df = tables[0]
print(df).
```

The Python project uses Libraries for the process of Extraction :

- **Import re**
- **Import pdfminer.high_level**
- **Import sys**
- **Import tabula**
- **Import fitz**
- **Import Pil.image**

HTML Code

```
<!DOCTYPE html>
<html>
  <head>
    <title>Title of the document</title>
  </head>
```

```
  <body>
    <div class="container">
      <div class="button-wrap">
        <label class="button" for="upload">Upload File</label>
        <form action="/action_page.php">
        <input type="file" id="myFile" name="filename">
        <input type="submit">
        </form>
        <input id="upload" type="file">
      </div>
    </div>
  </body>
</html>
```

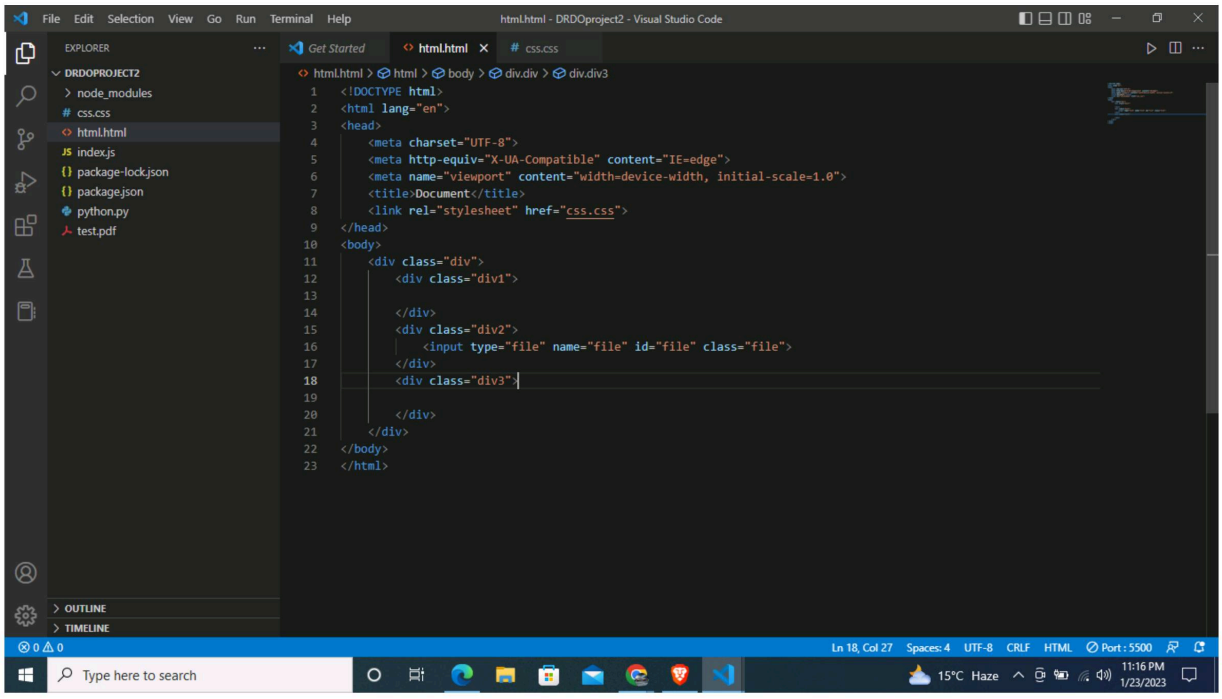

Figure 4.2 Screenshot of HTML Code

CSS code :

```
.container {
  display: flex;
  align-items: flex-start;
  justify-content: flex-start;
  width: 100%;
}


input[type="file"] {
  position: absolute;
  z-index: -1;
  top: 10px;
  left: 8px;
  font-size: 17px;
  color: #b8b8b8;
}


.button-wrap {
  position: relative;
}


.button {
  display: inline-block;
```

```
    padding: 12px 18px;
    cursor: pointer;
    border-radius: 5px;
    background-color: #8ebf42;
    font-size: 16px;
    font-weight: bold;
    color: #fff;
}
```

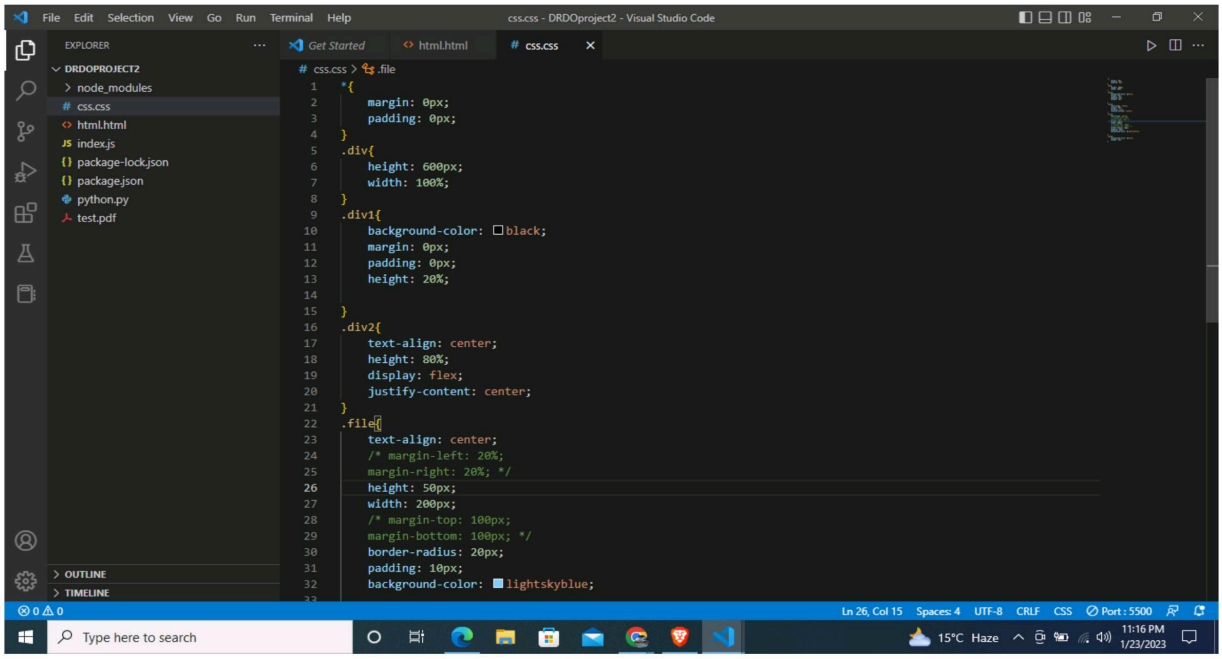

Figure 4.3 Screenshot of CSS Code

Challenges

1. Integration of backend with frontend.
2. Implementing code for creating pdf of all the images
3. Abstracting tables was really a critical part because when I implement this facing a lot of bugs in console

Limitations

1. Doesn’t abstract more than 1000 images in a single row.
2. Not abstract large amount of data in a table.

Conclusion and Future Scope

The project helps in the development of research papers to scientists and to others to extract data and analytics. The analytics save time and help to easily work on the paper, providing with the data that can be shown on the website and on download. The project is hosted online and can be used by many users.

In conclusion, this project aims to automate the process of data extraction from research papers using a web application developed using Python as a backend and React as a frontend. The web application will extract relevant data such as images, tables, and word count from research papers in PDF format, store the data in a structured format, and provide the user with the ability to download the data in CSV format for further analysis. The project has the potential to streamline the process of data extraction from research papers and reduce the time and resources required for manual data extraction.
The project can further be developed to help in the citation of the research papers. With the help of Machine Learning it can change the citation in the references and easy the process of changing the citation. It can be developed to change the order of the pages, change the outline and it can also be used as a paper development environment with all the tools for the development of the paper and ease the process of research paper development for the scientists and students.

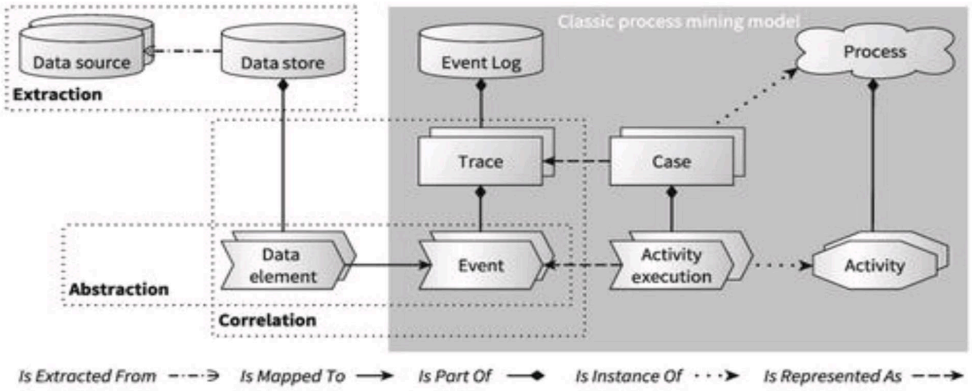

Figure 5.1 Graphical Abstract

References

DRDO. "About DRDO." *Defence Research and Development Organisation*, 2023, https://www.drdo.gov.in/about-drdo. Accessed 21 January 2023.

Stitchdata. "What is Data Extraction? Data Extraction Tools & Techniques." *Stitch Data*, https://www.stitchdata.com/resources/what-is-data-extraction/. Accessed 25 January 2023.

"Data extraction." *Wikipedia*, https://en.wikipedia.org/wiki/Data_extraction. Accessed 25 January 2023.

IBM. "What is Machine Learning?" *IBM*, https://www.ibm.com/topics/machine-learning. Accessed 25 January 2023

Machine learning."*Wikipedia*,2023,https://en.wikipedia.org/wiki/Machine_learning Accessed 25 January 2023.

"Python Tutorial." *W3Schools*, 2023, https://www.w3schools.com/python/. Accessed 25 January 2023.

Van Rossum, Guido. "Python (programming language)." *Wikipedia*, 2023, https://en.wikipedia.org/wiki/Python_(programming_language). Accessed 25 January 2023.