



**AUTHOR'S NAME:**

**MUNEEB UR REHMAN (160655)**

**ZAEEM DASTAGHIR (160335)**

## **IN DOC DATA ANALYSIS**

**Bachelor of Science in Computer Science**

**SUPERVISOR NAME: MUHAMMAD SHOAIB MALIK**

**CO-SUPERVISOR NAME: IMRAN IHSAN**

**DEPARTMENT OF COMPUTER SCIENCE**

**AIR UNIVERSITY, ISLAMABAD**

**May 2020.**

© Muneeb, Zaeem, 2020

## Certificate

We accept the work contained in this report titled “In Doc Data Analysis”, written by Muneeb ur Rehman And Zaeem Dastaghir as a confirmation to required standard for the partial fulfillment of the degree of Bachelor of Science in Computer Science.

### Approved by:

Supervisor: Muhammad Shoaib Malik

Internal Examiner:

External Examiner:

Project Coordinator: Imran Ihsan

Head of the Department: Dr. Mehdi Hassan

May, 2020.



## Abstract:

The research work done is always somehow dependent on the previous findings. So, for the further advancements and testing the authenticity of the research and availability of the previous datasets, algorithms and others important contents is very critical. However, our research culture differs in unusual ways. Most of the times only the well observed findings and polished results are included

However, the algorithms, input datasets, raw result datasets, as well as scripts that were used to produce the results in the original research document are sporadically made obtainable for the public reviewers and typically not available to other researchers also. Here it leads to springing up of many questions related to the legitimacy of the research work, data sets and results published.

Unfortunately, till now instead of a long workflow which produced results for research paper, only the final result is published in journals.

Hence some of the steps which have played vital role in the research results are skipped. This is very forlorn and has been denounced in several research communities. In our system we have argued on one fragment of the problem and our dated perspective on what a research document is and how it can be refined or should look like.

As a remedy, we introduce Janiform document format. These documents are of hybrid format, i.e. they are at the same time a standard static pdf as well as a highly dynamic (offline) HTML-document. PDbF's allow you to access the raw data and reproduce your own graphs from the raw data all of this *within* a portable document.

One of the many problems that are being faced by the research community is of raw data unavailability. As of today, data-science is becoming a hot field. Most of the research work is being carried out in this branch. Most of the times the data upon which several operations were performed become obsolete and is not available to reviewers and hence it becomes a hectic task for further advancements in that work. What we have proposed is to embed that data within a document. This would also allow the long-term preservation of the data. It would also resolve its availability questions as it is just one click away from the reader or a reviewer.

On parallel to this we also have provided visualization of the data within offline HTML format of the document. Hence, we have addressed these

issues with our utmost efforts and we hope that our work will somehow ease the issue faced in making the research work more organized.

We demonstrated a system that authorize its users to compile Janiform documents smoothly from within LATEX editors. This tool allows to preserve the workflow of raw measurement data to its culminated graphical output through all transitioning steps.







## **Acknowledgment:**

I would like to express profound gratitude to my guide Sir Muhammad Shoaib Malik and Sir Imran Ihsan for their invaluable support, encouragement, supervision and useful suggestion throughout this project work. Their moral support and continuous guidance enabled me to complete my work successfully.

I am grateful for the corporation and continuous encouragement for my honorable Dean of the Computer and AI Dr. Kashif Kifayat and Head of Department Dr. Mehdi Hassan.

Last but not the least, I am thankful and indebted to all those who helped me directly or indirectly in completion of this project and project report.

Muneeb ur Rehman  
Zaeem Dastaghir,

**Bachelor of Science of Computer Science**



“We think someone else, someone smarter than us, someone more capable, someone with more resources will solve that problem.

But there isn’t anyone else.”

Regina Dugan

## Table of content

### Table of Contents

<b>Abstract:</b> .....	5
<b>Acknowledgment:</b> .....	9
<b>Chapter 1: Introduction</b> .....	17
<b>1.1 Background:</b> .....	17
<b>1.2 Objective:</b> .....	18
<b>1.3 Scope:</b> .....	18
<b>1.4 Problem Statement:</b> .....	19
<b>Chapter 2: Literature Review</b> .....	21
<b>2.1 Related Projects:</b> .....	21
<b>2.1.1 Janiform IntraDocument Analytics for Reproducible Research</b> .....	21
<b>Chapter 3: Requirement Specification</b> .....	23
<b>3.1 Existing System:</b> .....	23
<b>3.2 PROPOSED SYSTEM:</b> .....	23
<b>Chapter 4: Design</b> .....	26
<b>4.1 SYSTEM ARCHITECTURE</b> .....	26
<b>4.2 DESIGN CONSTRAINTS</b> .....	26
<b>4.3 DESIGN METHODOLOGY</b> .....	27
<b>4.4 GUI DESIGN</b> .....	28
<b>Chapter 5: System Implementation</b> .....	31
<b>5.1 System Architecture</b> .....	31
<b>5.1.1 Architecture Description:</b> .....	31
<b>5.1.2 Project Flow</b> .....	32
<b>5.1.3 Tools and Technology:</b> .....	33
<b>Chapter 6: System Testing Evaluation</b> .....	35
<b>Chapter 7: Conclusion</b> .....	38
<b>Appendix A</b> .....	40
<b>USER MANUAL</b> .....	40
<b>References:</b> .....	41

## List of Figures

## List of Tables

## **Acronyms and Abbreviation**

# Chapter 1



## Chapter 1: Introduction

In our project we are working on the problem faced by the research community most of the times. As the availability of data along with research results and benchmarks is very important in every research conducted so far. Here arise an issue of data availability and its validation. If someone wants to carry a research on already published research paper, it is importunate for the researcher to get their hands on the previous conducted research and data which was previously gathered by the author. Unfortunately, in many cases data is not available. So, one has to regather all the previous data or to contact the previous author for the data set, which god knows better is valid or not.

Our motive is to address these problems and to come up with simple and viable solution. Our proposed solution is to embed that data with the PDBF document which in turn eradicates the question of data availability. As an additive feature we also aim to provide ML algorithms that can be applied on the data set within the PDBF document.

We do not aim to provide all the ML algorithms, but to prove that this concept can be applied realistically our motivation is to implement one model and its results on appropriate dataset.

### 1.1 Background:

Over the course of time with the computer sciences has revolutionized. The main issue which is being faced by the research community is the management, providence and preservice of the data. However, one of the many issues of our research publication culture is that we only publish a brief summary of the final results of the long projects. Only the well-polished graphs, brief result outputs and diagrams are not sufficient for the reviewer to check the validity and the quality of the research work done by the authors.

This issue has been discussed and criticized in many research and scientific communities but the efficient till date no one has come up with the efficient solution to meet the benchmarks.

One of the very basic solution that has being used is “**ask and provide**” method. One should have to contact the author and ask them to provide the reviewer with the data and algorithms of the research problems. In many cases this is not a viable solution. Some also have reported that the actual data do not even exists. In these cases, the reviewer have to improvise and gather the

data again for performing the same testing and experimentation, which is a very time consuming and hectic task. Even after such tasks the results are not reproduced with the precision that has been mentioned or shown in the published research papers. This in turn puts a question mark on the validity of the research work.

To avoid this drawback a viable and efficient solution that should address the problem stated above is impotunate. This is where our Final Year Project idea comes from. The project statement clearly describes the solution that is devised by our group to eradicate this issue.

## **1.2 Objective:**

The objectives of our project are as follows:

- Make research paper easier to understand for the reviewer.
- Embed related data inside a document.
- Allow the user to create ML models on embedded data.
- Provide useful graphical visualizations of data.
- Allow manipulation of visualization by the reviewer for validating correctness of tested data.

## **1.3 Scope:**

Our motivation behind choosing this project is the gap between the problem and its solution. Our project directly addresses the predicaments with very simple and viable solution. The toolkit we aim to develop is providing functionalities like data embedding and viewing within the document. In order to achieve and deliver these functionalities our proposed solution is to make research papers dynamic. We have achieved dynamicity by making research paper an offline HTML document in which dynamic functions are being provided for the reviewer.

We also have affirmed intentions of developing add on feature of providing the reviewer with machine learning algorithms. These machine learning algorithms can be applied on the embedded data. We also hope for producing results by the applied algorithm. This task is just for a spoof concept of work that it can be done using the resources that we have today.

Soon it can be a possibility that it would become possible that is impossible today.

#### **1.4 Problem Statement:**

This project plans to tackle this issue by producing Janiform reports from LaTeX documents. A Janiform document is a document that is valid PDF and HTML simultaneously. This enables the researchers to embed raw dataset inside the document and permits different reviewers to see that dataset and furthermore produce ML models on the embedded datasets utilizing the dynamic HTML version of the document.

# Chapter 2

## Chapter 2: Literature Review

The problem of data and results irreproducibility is always an issue for the research community. Once a research paper is published, the data which is used by the author for the results is not always available for the reviewers. They must contact the author or to search for other online repositories for having their hands on the raw dataset. This project aims to solve this problem by generating Janiform documents from LaTeX files. A Janiform document is a document that is valid PDF and HTML at the same time. This allows the author to embed raw dataset within the document and allows other users such as reviewers to view that dataset and also generate ML models on the embedded datasets using the dynamic HTML version of the document.

### 2.1 Related Projects:

There is only single research related to this project and that research is the base of our project.

#### 2.1.1 Janiform IntraDocument Analytics for Reproducible Research

This research shows that the research community always face a problem of data availability and irreproducibility of results, if they want to get the algorithm, graphs, dataset of the research they had personally had to contact the author of the research. That's why this project has come into the spotlight as this problem is solved by introducing a portable database file (PDBF). These files are Janiform i.e; static PDF and highly dynamic HTML at the same time. This PDBF file also allows the user to access raw data behind the graph, perform OLAP style analysis and reproduce your own graph from the raw data, all within the same file. The user can generate PDBF document from LaTeX files.

# Chapter 3

## Chapter 3: Requirement Specification

### 3.1 Existing System:

The system proposed us as project is novel. No such system exists which can provide similar functionality like our proposed system is proffering. There are some systems which are using similar approaches like the one proposed by our system.

For Example:

**R markdown** which is another documentation tool which provides the results insight through **R-Language** integration for producing results to be used in document. The resulting document of R-Markdown is pdf file. Although that document is integrated with the visual elements like charts, plots etc. All these elements are irreproducible and static images of results generated using **R** code.

These systems have many shortcomings like what if another person wants to recreate results or want to look on the data which was used to produce results. As today links of the datasets are provided but still questions arise about the authenticity of data. In some of the cases these links don't even provide data unless the paper is published in renowned journals.

Another drawback is less responsiveness. As most of the research papers include figures like charts tables etc. These figures are of utmost importance as it is best possible way to summarize findings and results produced. What if these figures can be made interactive. In this way it would become easy to evaluate research results. Conventional systems lack such properties.

### 3.2 PROPOSED SYSTEM:

Our vital aim is to make document dynamic and more responsive. We have been working on hybrid document format for our project. These documents have more than one valid formats. One can easily switch between formats by simply renaming the file.

The simple document format that is used most of the times just for reading purpose is PDF. So we are also using this format for our document. Accompanying it, as the document is in hybrid format another format will be html.

In html version of the document the reader will be provided with the services like downloading data file into local storage, viewing data table, performing SQL queries on data table, viewing plot of the data inserted into the data table.

Another functionality that we are working on to include is providing pre-trained ML model that can be applied on to the embedded data. This service can only be available in html format of the document.



# Chapter 4

## Chapter 4: Design

### 4.1 SYSTEM ARCHITECTURE

Our PDBF compiler coined as JANIC is written in java. Its architectural methodologies can be subdivided into further processes.

Firstly, a valid latex file is passed as an argument to Janic. Janic compiler will then invokes the standard latex compiler to validate and read the file. After that a PDbf-configuration is produced in first step. In addition to configuration file a simple pdf document will also be produced as draft version. Configuration file will then be read again by the Janic for relative placements of visual elements and dynamic content that has to be placed in the final hybrid document.

One can also include visual elements that are produced using in-document data for pdf version of document. For this purpose, it is also critical to produce static snapshots of those elements. These snapshots are very helpful to make a static document.

So, for this task we have used Phantom's which produces static snaps of the visual elements. This is for the pdf view of the document.

Now after successfully creating static snapshots and determining the relative positions of textual and visual elements. Janic compiler will again read the PDbf-configuration file. This time it is done for final placements of the elements in dynamic document.

Now a final output is produced which is a hybrid document.

### 4.2 DESIGN CONSTRAINTS

Our system, as previously mentioned is based upon java. Moreover, Latex is also very vital component in our toolkit. In designing process, we have faced problems like:

- How to efficiently handle different input files.
- How to provide enough solution of files passing to and from classes.
- How to keep our work as simple as possible for the reviewers and computer science related community.

So, that they can easily add their spadework to make our system more efficient and up-to the future advancements.

Keeping in-view the mentioned concerns. We hope that the designed system will meet the expectations and accomplish the given task in effectual manner. As we have aimed to embed data into the html-document. We have faced data related hurdles which are not negligible because of complexity and size related concerns.

Most of the times the data files provided as the standalone links are in csv or textual form. So, keeping in mind the majority we have designed our system to perform in well-ordered way for these data files. Dealing this issue was very important because in this way the efficiency and compilation time is at stake. It is not yet impossible to embed data files having too much size and incompatible data types. These reasons do not seem big but when we add them, they become considerable and adds up different factors that can in turn make our system's overall performance at risk.

We have assumed that every targeted user of our system has basic know how of Latex and can write simple LaTeX documents. The knowledge of command prompt will also help in using the system without any hurdles.

### **4.3 DESIGN METHODOLOGY**

We have done our utmost to design our system with low complexity and easy to modify in future.

Our implementation of the system is divided into two stages.

1. Pre-Compilation stage.
2. Compilation stage.

In this way it will be very easy for the the reviewer to understand the working of system. Another good factor is that, modifications can be enacted easily without effecting the other components of system.

**Pre-Compiler:**

The logical implementation, file handling, component related methods are invoked in this stage. Every component of the hybrid document has its own method declared separately with all the related logical providence.

### **Compiler:**

Here the finalized results along with the authentication flags is received and output is produced. Flags also serve as the error specifiers of the methods invoked in pre-compilation stage. Hence it becomes easy to trace the error back to the line where the system stopped execution.

This class hierarchy considered because it addresses the problem's solution in efficient and easy to understand manner. Other designs were also considered but this one was most feasible in our system's development.

## **4.4 GUI DESIGN**

Our system does not require that much graphical interface. It is because our toolkit is not any mobile or web-based application. They required graphical interfaces for attracting the users and making the product look new and refreshed.

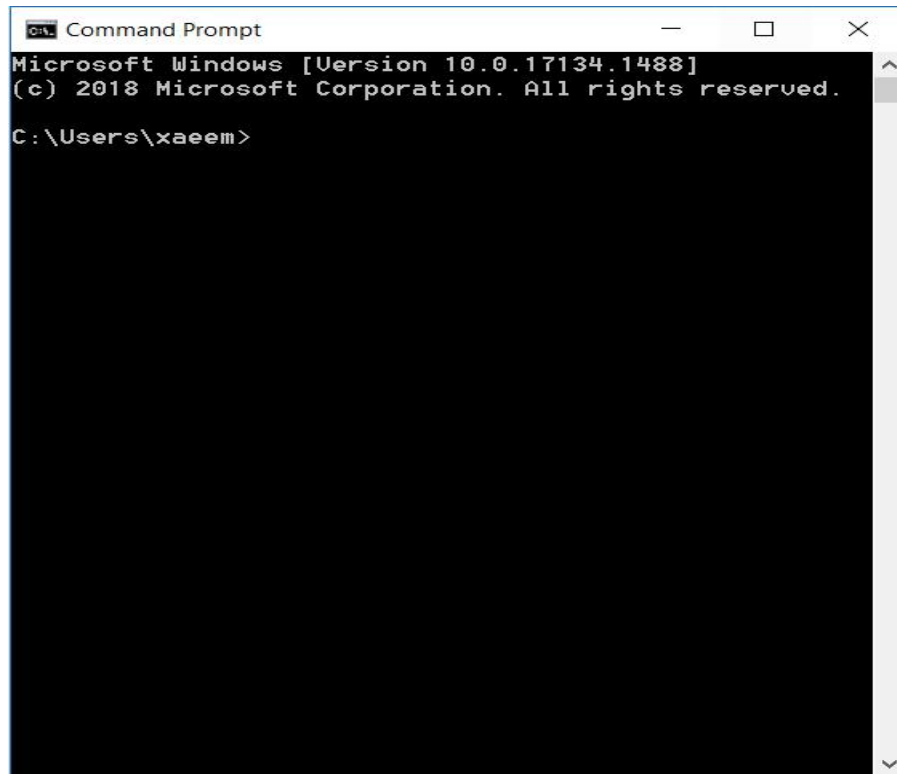
As we can address our system as the document editing tool and keeping in view the other document producing tools none of them have graphics like MS Word, WPS Office etc.

This was the main reason due to which there's no need to develop a separate interface.

Our user will use LaTeX editor for writing .tex file. Now it depends on the user and availability of LaTeX editors. One can any of them like MikTeX, TexMaker, TexWorks etc. These editors are not that complex, and one can easily understand their working methodology.

As for keeping the system easy to use we have utilized the OS built-in Command line interpreter which is commonly known as command prompt.

After successful completion of writing latex file. User must start command prompt. A window will open having black background.



```
Command Prompt
Microsoft Windows [Version 10.0.17134.1488]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\xaeem>
```

Here a user is required to type custom command to evoke JANIC compiler which after some processing creates a Janiform document.

# Chapter 5

## Chapter 5: System Implementation

### 5.1 System Architecture

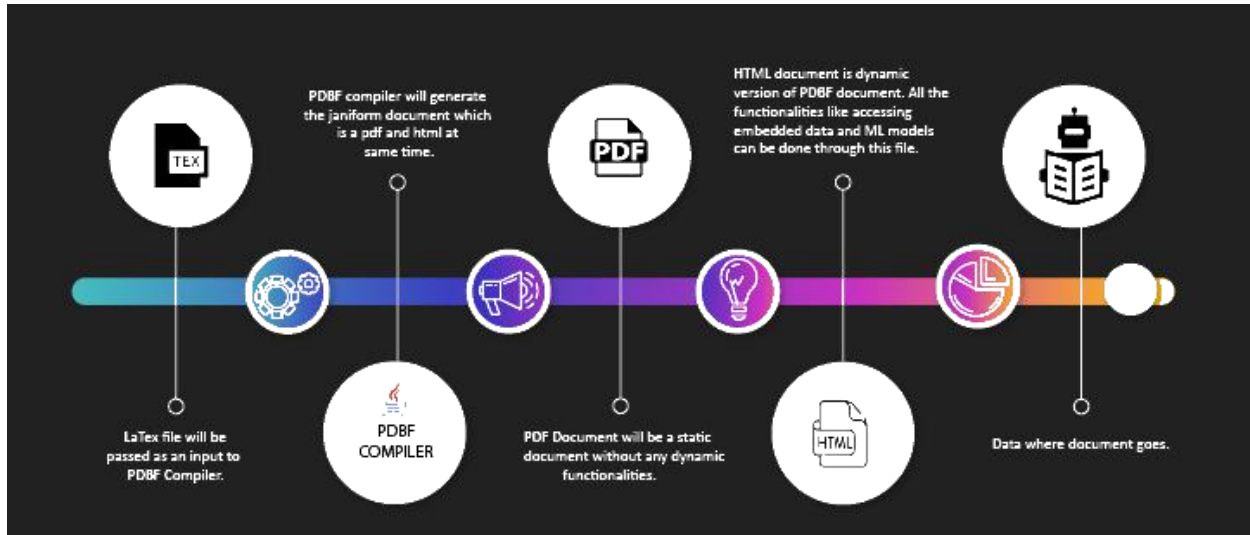


Figure 5.1

#### 5.1.1 Architecture Description:

The Architecture of our project is as follow:

- LaTeX file will be passed as an input to PDBF compiler.
- PDBF compiler will generate Janiform document which is a valid PDF and HTML document at the same time.
- PDF document will be static document without any dynamic functionalities.
- HTML document will be dynamic version of PDBF document. All functionalities like accessing embedded data and creating ML models will be done through this file.
- Data goes where the document goes.

## 5.1.2 Project Flow

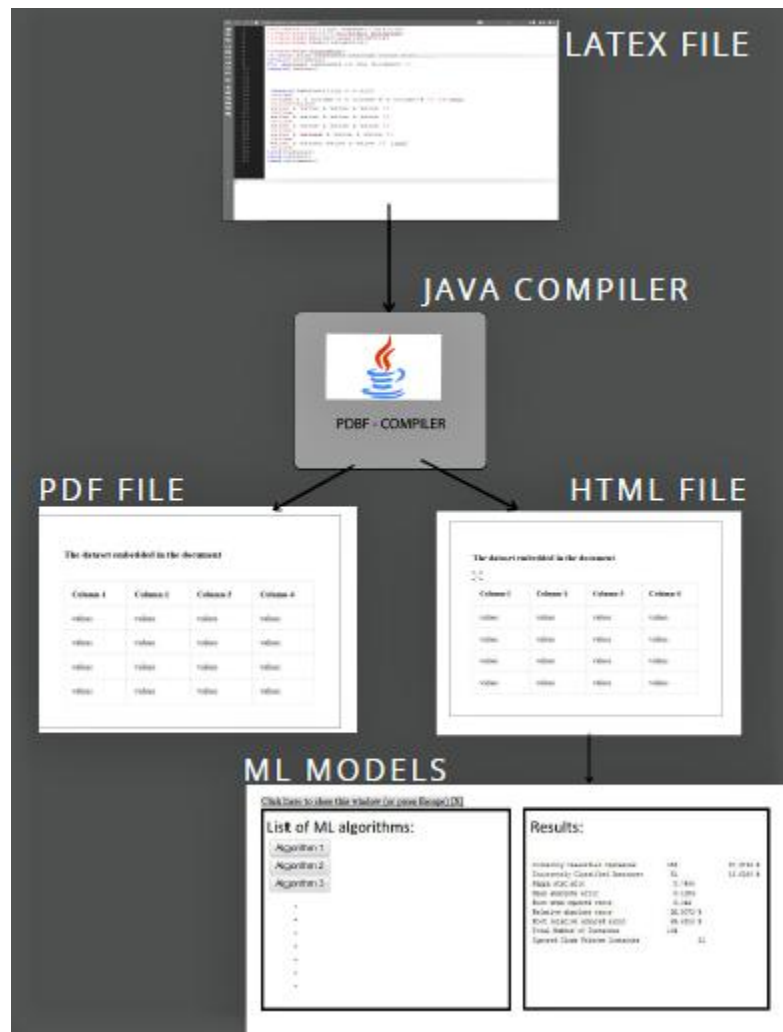


Figure 5.2

### Working:

This flow shows the working of our project as the user write it's LaTeX file in TeX Studio/MiTex the user gives that LaTeX file as an input to the PDBF compiler. The PDBF compiler compiles the file and generate a Janiform document which is a static PDF as well as dynamic HTML versions of the document and that dynamic HTML document with all functionalities like accessing raw data, embedded data and creating ML through that file. The benefit of PDBF file is the raw data and the embedded data in the document will also goes where the file goes that means anybody can access that data.



### **5.1.3 Tools and Technology:**

#### **Any LaTeX Distribution:**

LaTeX distribution like Tex works, Tex studio, MikTex are the platforms where the user writes and compile their LaTeX file.

#### **Intelli J:**

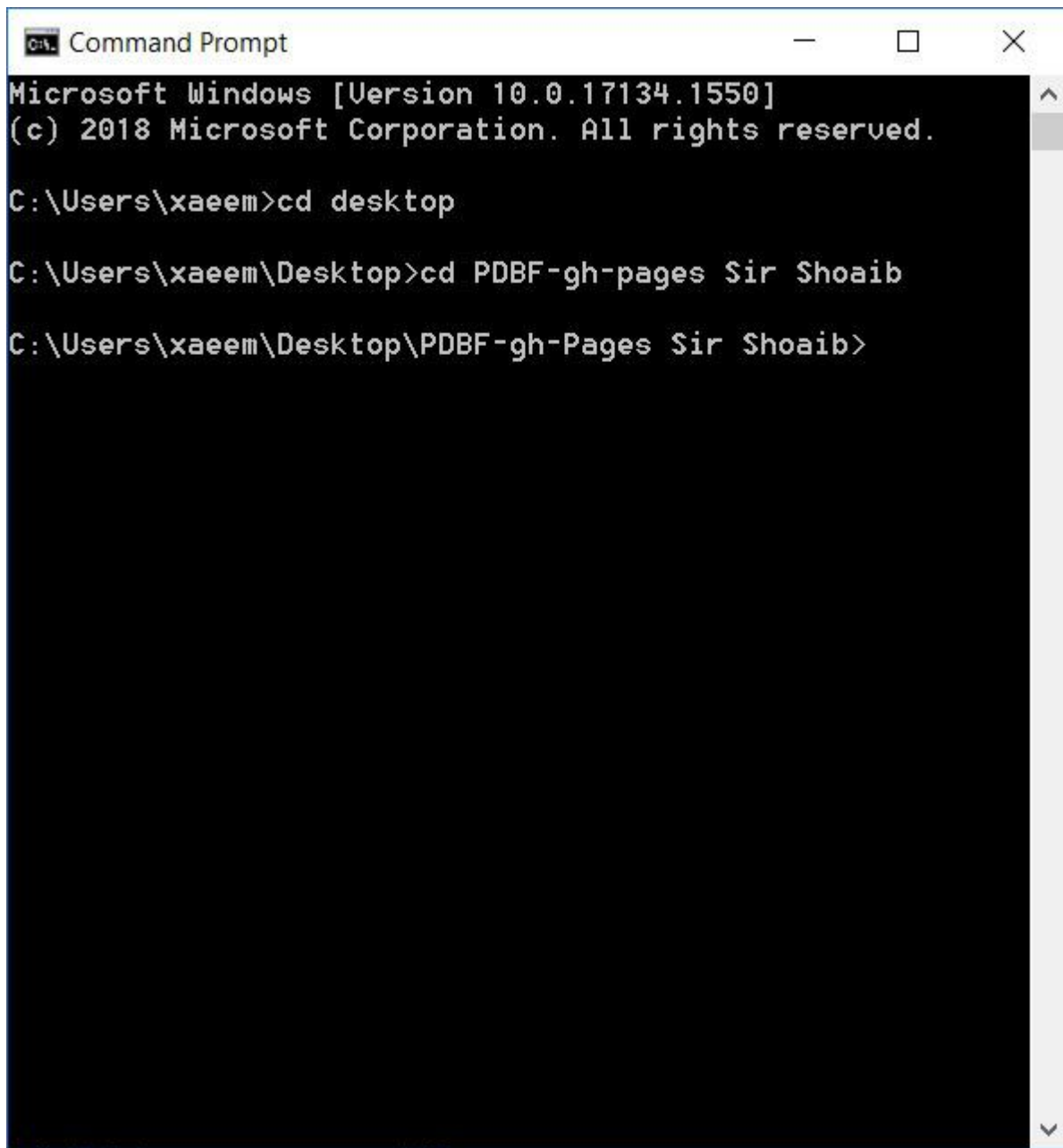
Intelli J is the Java JDE compiler which we used to build the compiler that generates PDBF form of the document.

#### **VS Code:**

VS Code is used to write and compile Java Script file within the compiler which we used to build Janiform of the document.

# Chapter 6

## Chapter 6: System Testing Evaluation



```
Command Prompt
Microsoft Windows [Version 10.0.17134.1550]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\xaeem>cd desktop

C:\Users\xaeem\Desktop>cd PDBF-gh-pages Sir Shoaib

C:\Users\xaeem\Desktop\PDBF-gh-Pages Sir Shoaib>
```

The image shows a Windows Command Prompt window with a black background and white text. The title bar at the top reads 'Command Prompt'. The window displays the following text: 'Microsoft Windows [Version 10.0.17134.1550]', '(c) 2018 Microsoft Corporation. All rights reserved.', 'C:\Users\xaeem>cd desktop', 'C:\Users\xaeem\Desktop>cd PDBF-gh-pages Sir Shoaib', and 'C:\Users\xaeem\Desktop\PDBF-gh-Pages Sir Shoaib>'. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.



# Chapter 7

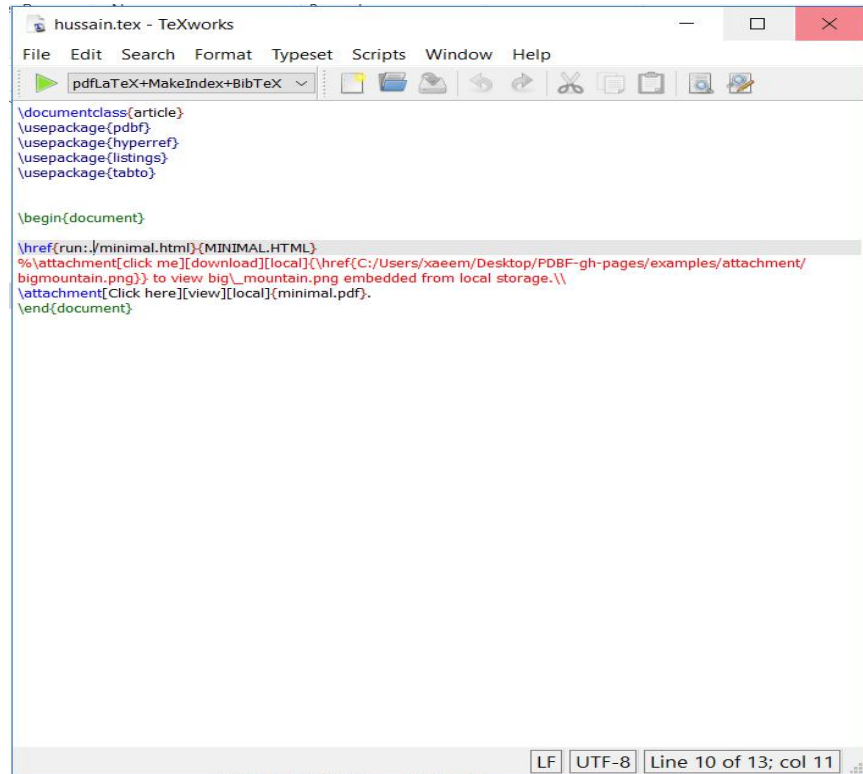
## Chapter 7: Conclusion

This system has opened a new horizon of portable data files and hybrid documents. As the documents have two formats pdf and HTML. So, there are many possibilities and ways to improve this system. The dynamic copy of the document has the potential of further advancements which can be done. As till now we have accomplished the task of embedding the data into the document which solved the issue of raw data availability and preservice for the long terms. Yet there are many horizons that are still undiscovered.

We cannot provide many of the functionalities till now due to the lack of technologies and unavailability of resources.

Who knows that in future it might become possible. In future we want to embed trained and untrained machine learning models. These models can then be applied on embedded data and can also be trained on data within the document. We would also like to provide unconstrained data file limit by implementing most efficient data compression methods, but we cannot say when it would all become reality.

Someday it will be possible for us to make our future assumptions into reality, because we know that someday, someone can do this work more efficiently than us.

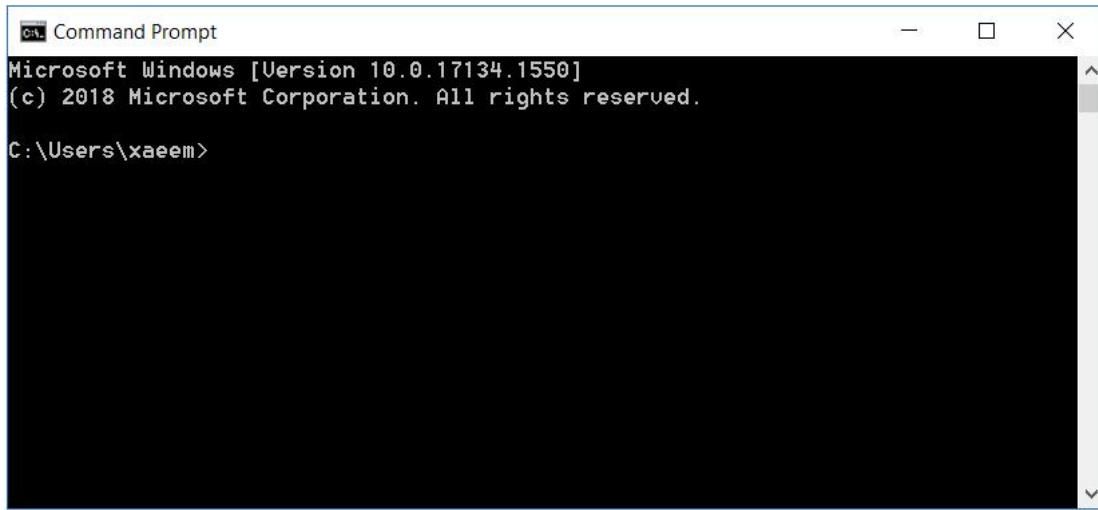


Different Tex editors have different

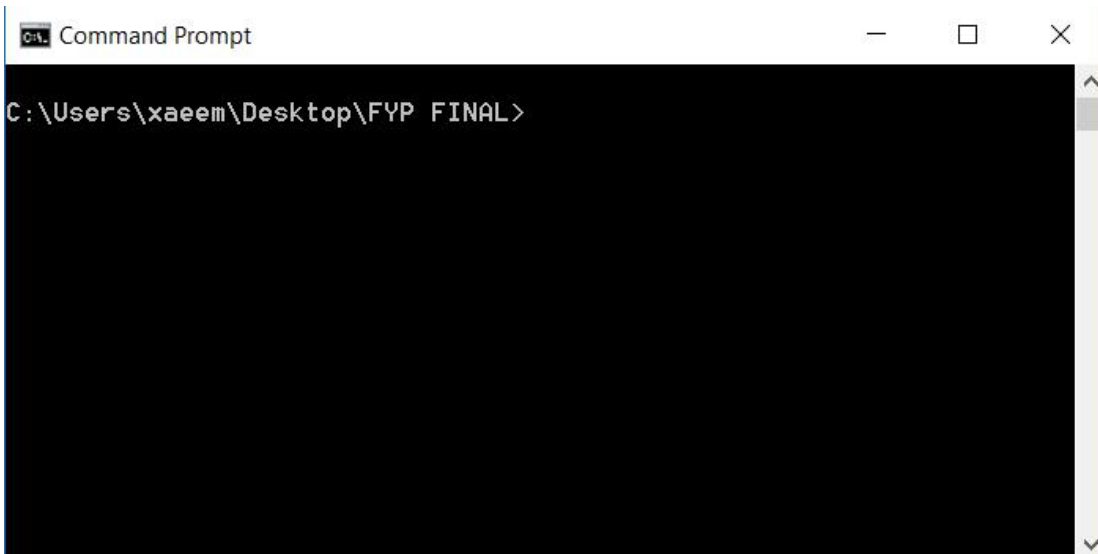
## Appendix A

### USER MANUAL

- Download toolkit from git or run it from your personal device if you have this toolkit already in your local storage.
- Save your latex file in the folder where toolkit is located.
- Run command prompt as administrator.



- Change the working directory using DOS commands to the folder where toolkit is stored in local storage.

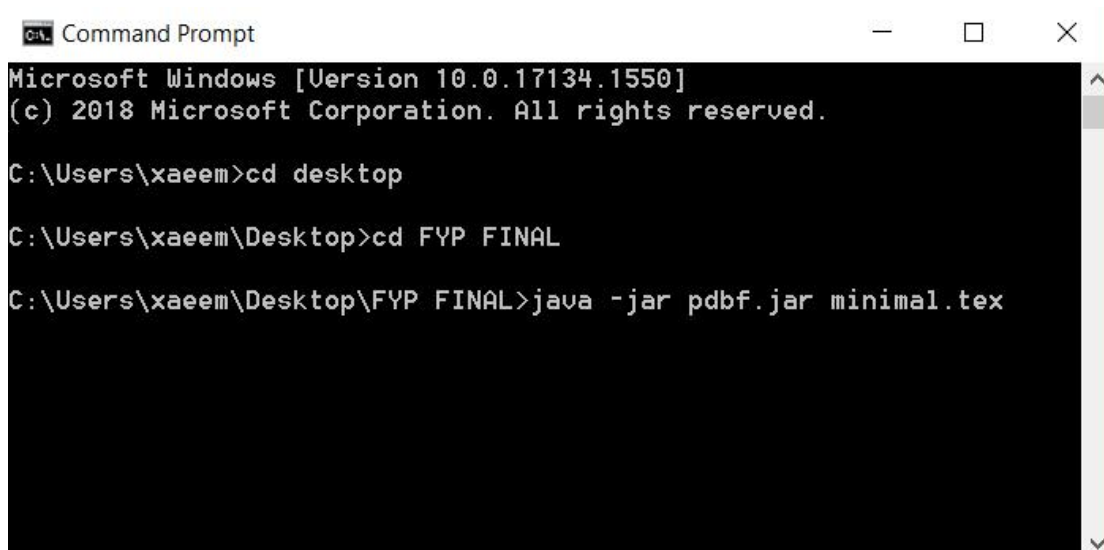


- Now after changing the working directory.



Type **java -jar pdbf.jar** (name of the tex file here with “.tex” at the end).

For example **java -jar pdbf.jar hello\_world.tex**.



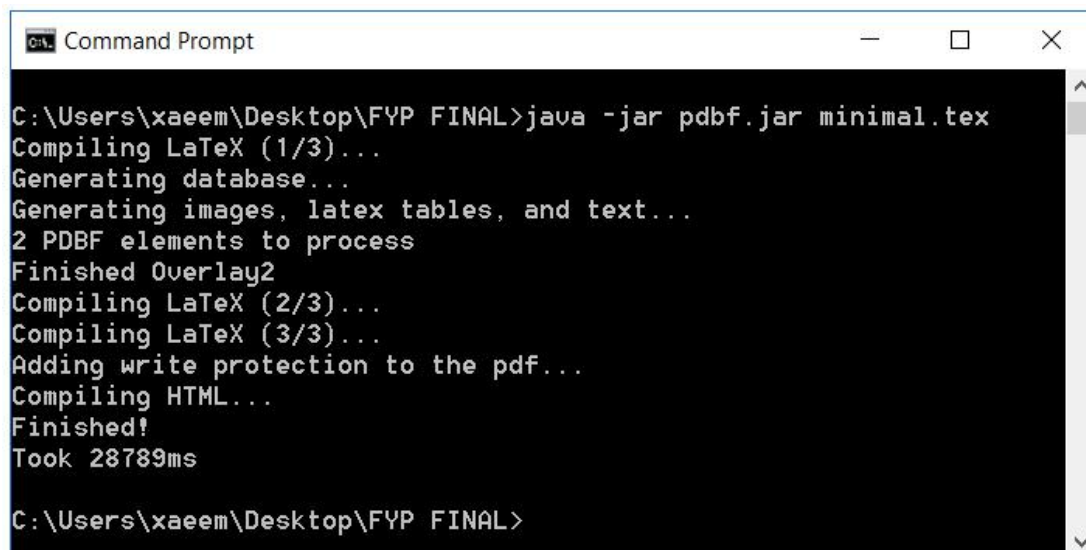
```
Command Prompt
Microsoft Windows [Version 10.0.17134.1550]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\xaeem>cd desktop

C:\Users\xaeem\Desktop>cd FYP FINAL

C:\Users\xaeem\Desktop\FYP FINAL>java -jar pdbf.jar minimal.tex
```

- After that the compilation steps would start and on successful completion HTML file will be generated as output. Change that file name to .pdf and it is valid pdf also.



```
Command Prompt

C:\Users\xaeem\Desktop\FYP FINAL>java -jar pdbf.jar minimal.tex
Compiling LaTeX (1/3)...
Generating database...
Generating images, latex tables, and text...
2 PDBF elements to process
Finished Overlay2
Compiling LaTeX (2/3)...
Compiling LaTeX (3/3)...
Adding write protection to the pdf...
Compiling HTML...
Finished!
Took 28789ms

C:\Users\xaeem\Desktop\FYP FINAL>
```

## References: