Technical Project Report Template using Latex

A Project Report
Submitted to the Bharati Vidyapeeth College of Engineering
in partial fulfillment of requirements for the award of degree

Bachelor of Engineering
in
Information Technology

by

Bhoir Ankit Bharat (4407)
Ankolekar Vaibhav Pandurang (4401)
Chouhan Aman Ravi (4414)
Badge Aman Ramesh (4404)

DEPARTMENT OF INFORMATION TECHNOLOGY
BHARATI VIDYAPEETH COLLEGE OF ENGINEERING
NAVI MUMBAI
October 2022

DEPARTMENT OF INFORMATION TECHNOLOGY
BHARATI VIDYAPEETH COLLEGE OF ENGINEERING
NAVI MUMBAI
2022 - 23

CERTIFICATE

This is to certify that the report entitled **Technical Project Report Template using Latex** submitted by **Bhoir Ankit Bharat** (4407), **Ankolekar Vaibhav Pandurang** (4401), **Chouhan Aman Ravi** (4414) and **Badge Aman Ramesh** (4404) to the Bharati Vidyapeeth College of Engineering in fulfillment of the B.E. degree in Information Technology is a bonafide record of the project work carried out by him under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Prof. Hanamant B. Sale
(Subject Co-ordinator)

Dr. Shankar M. Patil
(Head of Department)

(External)

DECLARATION

We hereby declare that the project report **Technical Project Report Template using Latex**, submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Engineering of the Bharati Vidyapeeth College of Engineering, Navi Mumbai is a bonafide work done by us under supervision of Prof. Hanamant B. Sale

This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources.

We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Bhoir Ankit Bharat
Ankolekar Vaibhav Pandurang
Chouhan Aman Ravi
Badge Aman Ramesh

Navi Mumbai
04-10-2022

Abstract

This document contains essential templates required to write technical reports using LaTeX. This template may be used for the preparation of B.Tech project reports of Bharati Vidyapeeth College of Engineering, Navi Mumbai. Also minimum working examples to create equations, include figure, include table, table of contents symbols list and bibliographic citation in a LaTeX document are provided.

Students taking the undergraduate degrees in Engineering at the Bharati Vidyapeeth College of Engineering are required to undertake several a projects and to write them up as reports. This document provides some advice on using LaTeX to typeset these reports. It is also an example of how to go about making a LaTeX document: you should read the source as well as the typeset document in order to see how it was all done. This document originally conformed to the regulations laid down for the project report: it had the correct font size, margins etc. So you could use it as a template for your project report by removing all of our content and inserting your own. You should check with the current course organiser for projects whether this is still true.

Acknowledgement

It gives us a great sense of pleasure to present the report of the B.E Project undertaken during Bachelor of Engineering Final Year. We owe special debt of gratitude to Prof. Hanamant B. Sale, Subject Co-ordinator, Information Technology, Bharati Vidyapeeth College of Engineering, Navi Mumbai for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of Dr. Shankar M. Patil, Head of Department, Information Technology, Bharati Vidyapeeth College of Engineering, Navi Mumbai for his full support and assistance during the development of the project.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Bhoir Ankit Bharat
Ankolekar Vaibhav Pandurang
Chouhan Aman Ravi
Badge Aman Ramesh

Contents

List of Figures

List of Tables

Chapter 1
Introduction

The generally used documents such as project reports, notices, internal notes etc. in any technical institute are expected to be submitted in a standard specified format. The commonly used editing tools for this purpose are Microsoft Word, notepad, wordpad, etc. Taking into consideration various formatting constraints namely alignment, font styles, paragraphs, sections, subsections, etc. becomes a bit tedious using MS Word or other tools. Also maintaining the subscripts and superscripts to obtain the various mathematical equations becomes difficult and time consuming.

To overcome this drawback we have LaTeX a documentation preparation system that enables the document writer to concentrate on the contents of their text, without bothering too much about the formatting of it.

1.1 LaTeX : what is it?

LaTeX is a document preparation system for high-quality typesetting. It is most often used for medium-to-large technical or scientific documents but it can be used for almost any form of publishing.

LaTeX is not a word processor. LaTeX is based on the idea that authors should be able to focus on the content of what they are writing without being distracted by its visual presentation. In preparing a LaTeX document, the author specifies the logical structure and lets the LaTeX system worry about the presentation of these structures. It therefore encourages the separation of layout from content while still allowing manual typesetting adjustments where needed.

LaTeX is based on Donald E. Knuth's TEX typesetting language or certain extensions. LaTeX was first developed in 1985 by Leslie Lamport.

1.2 What are the advantages and disadvantages?

You might want to consider using LaTeX for your project write-up and your other reports for the following reasons:

- The typeset text looks nicer than word-processor output.
- The table of contents, cross-references and bibliography can be generated automatically.
- Many journals provide LaTeX styles and templates, so that when you come to write scientific papers you can do so in LaTeX without thinking about the typesetting at all. LaTeX is threfore a useful skill to have learned for any career in the hard sciences.
- LaTeX is available for FREE, for all popular operating systems (and most of the unpopular ones).

- The LaTeX way of typesetting equations occurs in other contexts, e.g. markdown cells in jupyter notebooks and any text items in matplotlib plots.
- You like programming. You hate word processors and find them frustrating.

Reasons that you might not want to choose LaTeX include:

- LaTeX is not WYSIWYG. You type your document into an ordinary text editor and typeset it in a separate step. Arguably, this helps one to separate layout (the computer's job) from content (your job).
- While it is easy to get LaTeX to typeset your document in its default style, it can be tiresome or obscure to get it to do anything different.
- The learning curve can be a little steep at first.
- You like word processors. You are good at getting them to do your bidding. You want to make use of this expertise.

1.3 Why LaTeX ?

What you see is what you get (WYSIWYG) programs make it easy to put text wherever you want in whatever size and style of type you want, i.e., WYSIWYG programs offer visual design. The visual design is fine for short, simple documents like letters and memos. The visual design is not good for more complex documents such as scientific papers. For this purpose, we use LaTeX that offers logical design.

LaTeX is intended to provide a high-level language that accesses the power of TEX. LaTeX comprises a collection of TEX macros and a program to process LaTeX documents. Because the TEX formatting commands are very low-level, it is usually much simpler for end-users to use LaTeX . As LaTeX is distributed under the terms of the LaTeX Project Public License (LPPL), LaTeX is free software.

1.4 Required Parts of LaTeX input file

A few commands must appear in every LaTeX input file in a certain order. They are:

```
\documentclass{classname}
\begin{document}
\end{document}
```

The documentstyle has a required argument stylename to select an overall type-setting style for the document; the one normally used is article (there are also book, report, letter, and

memo). It also has an optional argument to select 11pt or 12pt normal type size (10pt is the default size). The actual text of user document and associated commands go between the begin and end commands.

1.5 Customizing LaTeX

There are situations where LATEXdoes not provide a command or environment that matches user needs, or the output produced by some existing command may not meet user requirements. To add your own commands, use

```
\newcommand{name}[num]{definition}
```

The command requires two arguments: the name of the command you want to create, and the definition of the command. The num argument in square brackets is optional.

1.6 Creation of style file

Typically, a style sheet is specified at the beginning of document. This style sheet applies to the entire document. To create your own style file, at the very beginning of the text document just write,

```
\ProvidesPackage{mypack}
```

where mypack is the name of package. Write whatever you want in it using all the LATEXcommands you know. The style file should have the same name as that of the package name. Save this style file with extension .sty. Now, it is necessary to import this style file in your TEXdocument which can be done using following command.

```
\usepackage{mypack}
```

Chapter 2
Literature Review

2.1 Technical Papers

Sr. No.	Title	Results
1	Free and works on all operating systems, 2021	LaTeX is free and there are a multitude of text editors to choose from for each operating system. If you create a LaTeX .tex file at work on Windows and resume working on it at home on your Mac or Linux computer, the file will compile the same regardless.
2	Editing, Versioning, Automation and Outputs, 2020	Software developers commonly use source control systems to track changes to files over time. LaTeX is very amenable to using Git to track changes to your documents, something that is difficult to do for documents from a Word Processor
3	Focus on Content, 2020	LaTeX separates content and styling/design, and this is especially true when using templates. This ensures consistency since the formatting is handled separately and en masse.
4	Time Investment, 2019	When using a word processor, time spent trying to fix a formatting problem is essentially time wasted; it is unlikely the same problem will arise again. In LaTeX, once you figure out how to make a formatting change, you now know how to do this forever and it becomes quite natural given some repetition.
5	Longevity, 2019	Documents written 20 years ago in LaTeX are almost guaranteed to compile today. Compare this with a document produced using a word processor from 20 years ago, it is likely the software no longer exists and if it does, it has undergone so many changes that it may not display the document correctly.

Table 2.1: List of papers reviewed

Chapter 3
Problem Definition

Technical report writing is an important skill, and will stand you in good stead in your future career. It is much more precise than many other forms of writing. A project report is not quite the same as a technical report2 , however it should show the same level of care and attention to detail.

MS Word is for general purpose documentation. As against that LaTex is a strongly typed language engineered for technical documentation. In crude sense some of the advantages of using LaTex is the programming kinda approach to putting your stuff in place. Various packages help in creating a sharp and nicely written articles, reports, etc. There are certain commands for many mathematical building blocks.

For graphics again we have packages and some more commands. But once you get to know these commands, writing and creating high quality documents is but a walk in a dream. M.S. words does have many facilities, but the format and the way one uses it is very dis-ordered and not so sturdy. Across different versions, many basic properties like alignment, are found to be varying giving one a not so uniform rendering.

The proposed system attempts to use LATEX to make documentation of generally used documents like project thesis, reports, notices, etc. in the technical institutes a bit easier. The proposed system will provide class or style files for a technical institute to write

1. notice
2. project reports

The user simply needs to include the package name in his file and use the commands defined in the package to format his documents. Thus, the proposed system will reduce the efforts of the user in formatting the documents.

Chapter 4
Objectives of Work

4.1 List of Objectives

- LaTeX Report Template provides core and advanced concepts of LaTeX .
- Our Latex Report Template is designed for beginners and working professionals.
- The LaTeX is a high-quality typesetting system, used for the documentation of scientific and technical documents.
- It is widely used in academia for the communication and the publication of scientific papers popularly in fields such as economics, sociology, mathematics, chemistry, physics, engineering, etc.
- It also handles the formatting layout of different structures. The name is stylized as LaTeX.
- The existing word processors have several limitations which can be overcome by LATEX.

The main advantages of LATEX over normal word processors are the following:

- Professionally crafted layouts are available, which make a document really look as if "printed".
- The typesetting of mathematical formulae is supported in a convenient way.
- Users only need to learn a few easy-to-understand commands that specify the logical structure of a document. They almost never need to tinker with the actual layout of the document.
- Even complex structures such as footnotes, references, table of contents, and bibliographies can be generated easily.
- Free add-on packages exist for many typographical tasks not directly supported by basic LATEX. For example, packages are available to include PostScript graphics or to typeset bibliographies conforming to exact standards.
- LATEX encourages authors to write well-structured texts by specifying structure

Chapter 5
Models/Block diagram

5.1 Document Structure

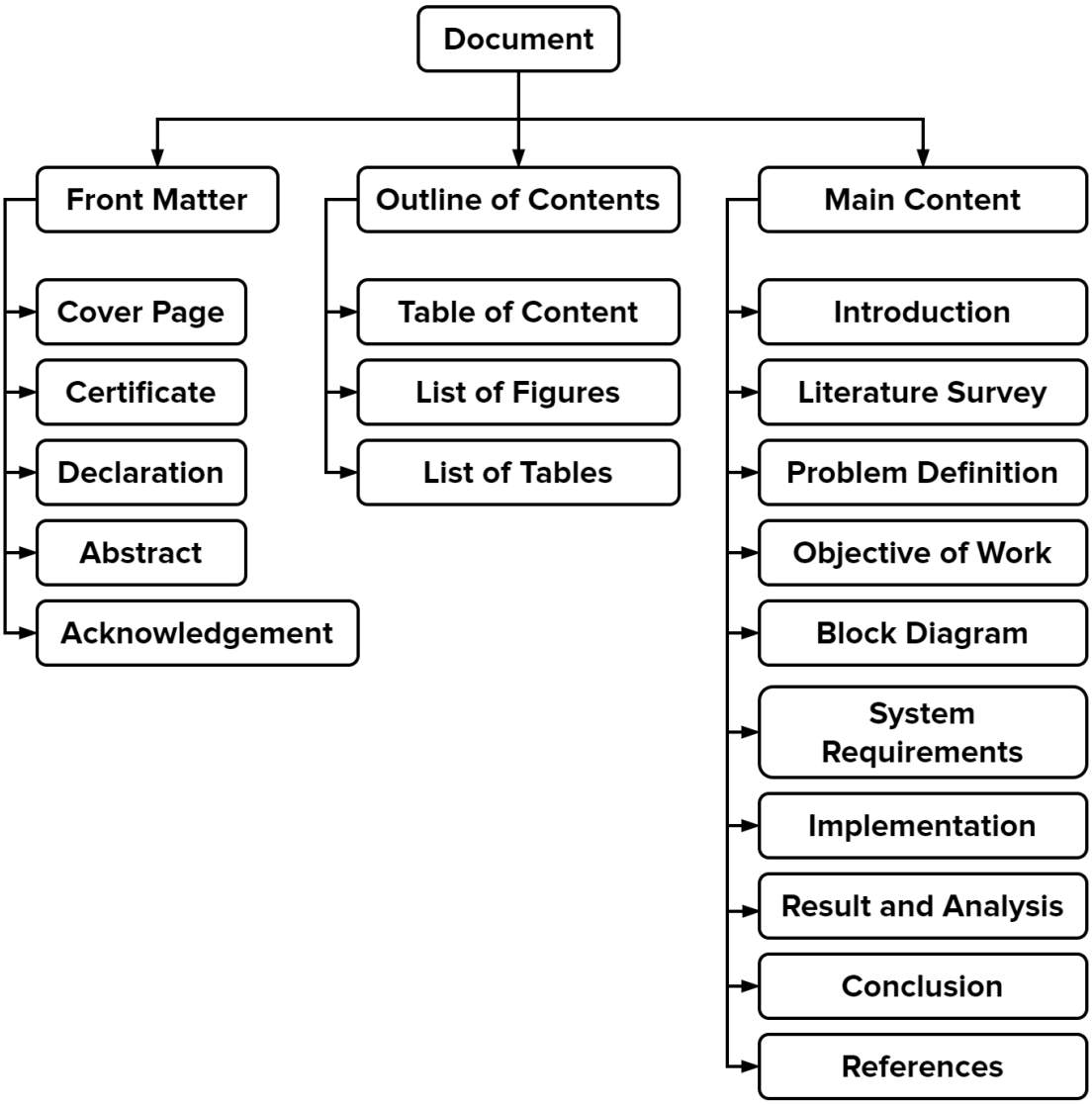


Figure 5.1: Block Diagram

Chapter 6
System requirements

6.1 Using Overleaf

Overleaf provides a rich-text editor so you don't need to know any code to get started—you can just edit the text, add images, and see the typeset document automatically created for you as you type. Our tutorial provides a quick three-step introduction to the main features.

Overleaf is an online collaborative writing and publishing tool that makes the whole process of writing, editing and publishing scientific documents much quicker and easier. Overleaf provides the convenience of an easy-to-use LaTeX editor with real-time collaboration and the fully compiled output produced automatically in the background as you type.

If you're familiar with LaTeX, using Overleaf couldn't be simpler as we provide full support for direct LaTeX editing, and automatically compile your document for you on our servers (so there's nothing to install). All you need to do is create a document and choose source mode in the editor to edit the LaTeX code for your paper.

The key features if Overleaf are :

- All you need is a web browser
- Always have the latest version
- Effortless sharing
- Automatic real-time preview
- Real-time track changes and commenting
- Complementary Rich Text and LaTeX modes
- Quickly find LaTeX errors

If you're new to LaTeX and would like to learn more about it, we recommend completing our online introduction to LATEX course, prepared by Dr John Lees-Miller and originally presented at the University of Bristol.

6.2 Using MiKTEX and Texmaker

The recommended TEX distributions are:

- TEXLive is a major TEXdistribution for Unix/Linux, Mac OS and Windows.
- MiKTeX is a Windows-specific distribution.
- MacTeX is a Mac OS-specific distribution based on TEXLive.

The Texmaker is an editor with the text window, structure window, toolbars, functions, and status bar. The white portion shows the text or writing window, and the black part shows the structure window. It contains a link to chapters, sections, tables, equations, etc. The features of Texmaker are given below:

- It includes spell checking while typing.
- It supports a variety of encodings.
- It contains a 'structure view,' which gets automatically updated while typing.
- In Texmaker, with the use of keyboard triggers, you can define an unlimited number of snippets.
- It gives you the full asymptotic support.
- It includes the built-in PDF viewer and the 'Quick Build' command.
- The Texmaker includes 37O mathematical symbols, which can be inserted in just one click.
- The extensive Latex document is furnished with the Texmaker.
- It automatically detects the warnings and errors with the corresponding line number after the compilation. It also contains the detail of each error.
- It also allows us to work efficiently onto the documents separated in several files with the "master mode."

Chapter 7
Implementation

7.1 Editing your document

You can edit your LaTeX document with any text editor. EMACS, vi, kate, gedit or even (ugh!) MicroSoft Notepad can be used. Save your document with a name that ends in .tex so that you and LaTeX know that it contains LaTeX source code. An absolute minimum LaTeX document is:

```
\documentclass{article}
\begin{document}
Here is some text for \LaTeX\ to typeset.
\end{document}
```

Note that things beginning with a are instructions to the LATEX program. Everything else is text to be typeset.

7.2 Organize

LaTeX can organize, number, and index chapters and sections of document. There are up to 7 levels of depth for defining sections depending on the document class:

```
-1 \part{part}
 0 \chapter{chapter}
 1 \section{section}
 2 \subsection{subsection}
 3 \subsubsection{subsubsection}
 4 \paragraph{paragraph}
 5 \subparagraph{subparagraph}
```

Usually, `\section` is the top-level document command in most documents. However, in reports or books, and similar long documents, this would be `\chapter` or `\part`.

In our template we have used report document so `\chapter` is the top-level document command.

7.3 Sectioning

You can generate section, subsection, subsubsection headings using `\section` etc. The sections are numbered automatically.

7.3.1 Avoiding numbering

If you want a section without its number, put a star on the end of the command, like this: Un numbered subsubsection Generated with the `\subsubsection*{}` command.

7.3.2 Unnumbered sections in the table of contents

To add an unnumbered section to the table of contents, use the `\addcontentsline` command like this:

```
\addcontentsline{toc}{section}{Title of the section}
```

7.4 Chapters

The LaTeX default starts each chapter on a fresh page, an odd-numbered page if the document is two-sided. It produces a chapter number such as 'Chapter 1' in large boldface type (the size is `\huge`). It then puts title on a fresh line, in boldface type that is still larger (size `\Huge`). It also increments the chapter counter, adds an entry to the table of contents, and sets the running header information.

This produces a chapter.

```
\chapter{Loomings}
Call me Ishmael.
    Some years ago--
```

In our template we have seperated chapter for each topic. The list goes as

```
\chapter{Introduction}
\chapter{Literature Review}
\chapter{Problem Definition}
\chapter{Objective of Work}
\chapter{Models/Block diagram}
\chapter{System requirements}
\chapter{Implementation}
\chapter{Results and Analysis}
\chapter{Conslusion anf Future Scope}
```

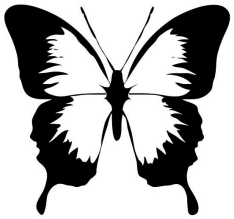
7.5 Including figures

Figures are inserted using the `\includegraphics` command. For this to work, you need to have used `\usepackage{graphicx}` near the start of the document. You can use `\includegraphics` anywhere you like.

The below figure has to appear exactly where I put it, between ". . . you like." and "The above figure. . . ". This is not smart for anything except the smallest graphics, because it can lead to large spaces at the foot of a page. More usually, we put figures into a float, like figure 7.1.

The source code for a floating figure looks like this:

Figure 7.1: This picture is in a float, which means that LaTeX will put it in a suitable position, somewhere close to where you put it in your LATEX source.



```
\begin{figure}[h!]
Figure content goes here
\caption{\label{myfiglabel} Your caption goes here}
\end{figure}
```

Note that LATEX provides a numbered caption. Make sure you put your `\label{}` inside the caption or the cross-referencing may not work properly.

The optional `[htbp]` argument gives you a little control over where the figure goes: you can put any combination of here `[h]`, top (of a page) `[t]` , bottom of a page `[b]`, (whole) page (containing only figures) `[p]` and LATEX will do its best give you the best of the options you permitted. Note that you need to be relaxed about figures not appearing exactly "here". For this reason, always refer to a figure by its number, never as "the figure below" or "the figure above". (This is good practice for real life: when you write a paper for a scientific journal, the journal staff always take control over figure placement and you never know where they are going to put the figures.) New LATEX users often try to be too restrictive about figure placement; this often results in LATEX deciding that there is no good answer and putting all of your figures right at the end of the document.

The caption is usually positioned below the figure. With a bit more work you can position the caption at the side of the figure; if your figure is square-ish this saves you some space. Figure 7.2 shows how this is done.

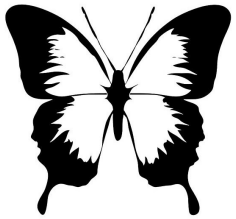


Figure 7.2: For this figure, the caption is placed inside a minipage; this is a small block of text which is typeset as a unit and then treated as a single item, as if it were a picture, or a very large character.

LATEX can not include all types of graphics, so you need to make your pictures in the correct format, or convert them to it.

- If you use latex and dvips then all of your figures must be encapsulated PostScript.
- If you use pdflatex then your figures may be any of
  - Portable Document Format (PDF)
  - JPEG image
  - PNG image

If you miss off the filetype extension, then LATEX will look for a file with a suitable extension. So if you do

```
\includegraphics[width=4cm]{lara_croft}
```

then latex will look for a file called `lara_croft.eps`, but pdflatex will look for files called `lara_croft.pdf`, `lara_croft.jpg` and `lara_croft.png` (in I don't know what order).

7.6 Tables

The `tabular` environment is the default LATEX method to create tables. You must specify a parameter to this environment; here we use `{c c c}` which tells LaTeX that there are three columns and the text inside each one of them must be centred.

7.6.1 Creating a simple table in LATEX

The tabular environment provides additional flexibility; for example, you can put separator lines in between each column:

```
\begin{tabular}{ ccc }
   \hline
   cell1 & cell2 & cell3 \\
   cell4 & cell5 & cell6 \\
   cell7 & cell8 & cell9 \\
   \hline
\end{tabular}
```

This produces :

cell1	cell2	cell3
cell4	cell5	cell6
cell7	cell8	cell9

7.7 Page Layout

Fitting a LATEX document to a strict page layout can be a bit fraught. I set this document up so that it matched the requirements of the projects course organiser:

- 12 point font
- 2.5 cm margins
- 1.0 line spacing
- A space (size unspecified) between paragraphs

But the exact size and position on the paper may be messed up a bit in the printing process. The important thing is to ensure that page scaling and auto-centre and rotate are turned OFF when you print the page. In the evince document viewer, the print dialog has a "Page Handling" tab; select "Page Scaling=None" and uncheck the "Auto-Rotate and Centre" box. The print dialog in Adobe reader has a page handling section with similar controls.

```
\textheight=23cm
\topmargin=0.5cm
\oddsidemargin=0.5cm
\textwidth=15.0cm
```

7.8 Fonts and sizes

Try to avoid controlling font size by hand. LATEX's default style is pre-set to use sensible font sizes for most things. If you insist, LATEX comes with several pre-defined sizes: tiny, scriptsize, footnotesize, small, normalsize, large, large, LARGE, huge, and Huge.

7.9 Page Border

Page border for a certain page can be added using following code :

```
\begin{tikzpicture}[overlay,remember picture]
    \draw [line width=3pt]
        ($ (current page.north west) + (1.3cm,-1.3cm) $)
            rectangle
        ($ (current page.south east) + (-1.3cm,1.3cm) $);
    \draw [line width=1pt]
        ($ (current page.north west) + (1.45cm,-1.45cm) $)
            rectangle
        ($ (current page.south east) + (-1.45cm,1.45cm) $);
```

```
\end{tikzpicture}
```

In our template, we have added page border in cover page, certificate and declaration page.

7.10 User defined variables

```
\def<cs><arg syntax>{<definition>}
```

This defines <cs> without checking if it already exists. The ⟨definition⟩ part is the same as that for \newcommand, but with \def you don't just specify the number of arguments. Instead you declare the argument syntax in <arg syntax> where each parameter is identified by #<n> (where<n> is a number from 1 to 9).

The simplest form is when you define a command that has no arguments. For example:

```
\def\test{This is a test.}% definition
\test
```

produces:

This is a test.

\def can be preceded by \global (the redefinition isn't confined to the current group, also controlled by \globaldefs), \long (the command admits long arguments), \outer (the command is not tolerated in parts of the code which are read at high speed).

\gdef works as \def except that the definition is automatically set as \global, or in other words, visible from the outside of the current group. In our template we have created \gdef definitions such as

```
\gdef \title {Technical Project Report Template using Latex}
\gdef \dept {Information Technology}
\gdef \degree {Bachelor of Engineering}
\gdef \branch {Information Technology}
\gdef \college {Bharati Vidyapeeth College of Engineering}
\gdef \collegeplace {Navi Mumbai}
\gdef \studentA {Bhoir Ankit Bharat}
\gdef \studentAroll {4407}
\gdef \studentB {Chouhan Aman Ravi}
\gdef \studentBroll {4414}
\gdef \studentC {Badge Aman Ramesh}
\gdef \studentCroll {4404}
\gdef \studentD {Ankolekar Vaibhav Pandurang}
\gdef \studentDroll {4401}
\gdef \guide {Prof. Hanamant B. Sale}
\gdef \guidedes {Subject Co-ordinator}
\gdef \hod {Dr. Shankar M. Patil}
\gdef \hoddes {Head of Department}
```

Chapter 8
Results and Analysis

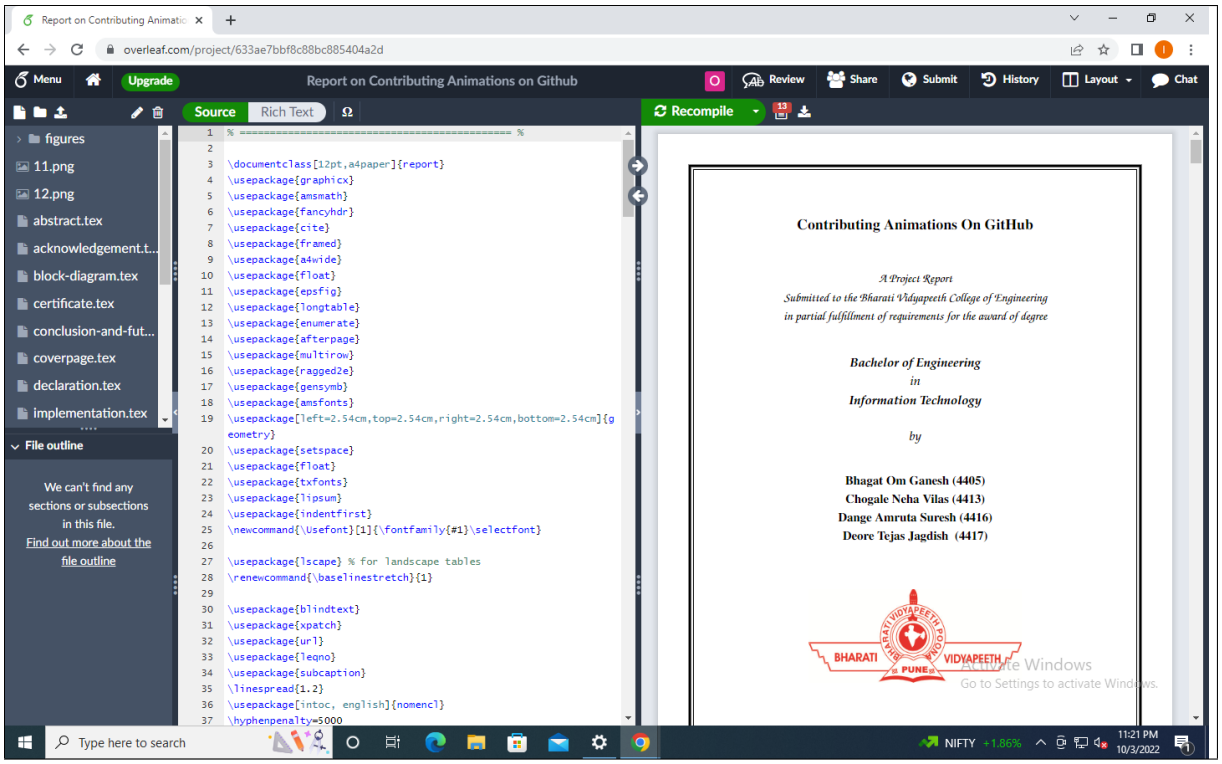


Figure 8.1: Result of cover-page with border

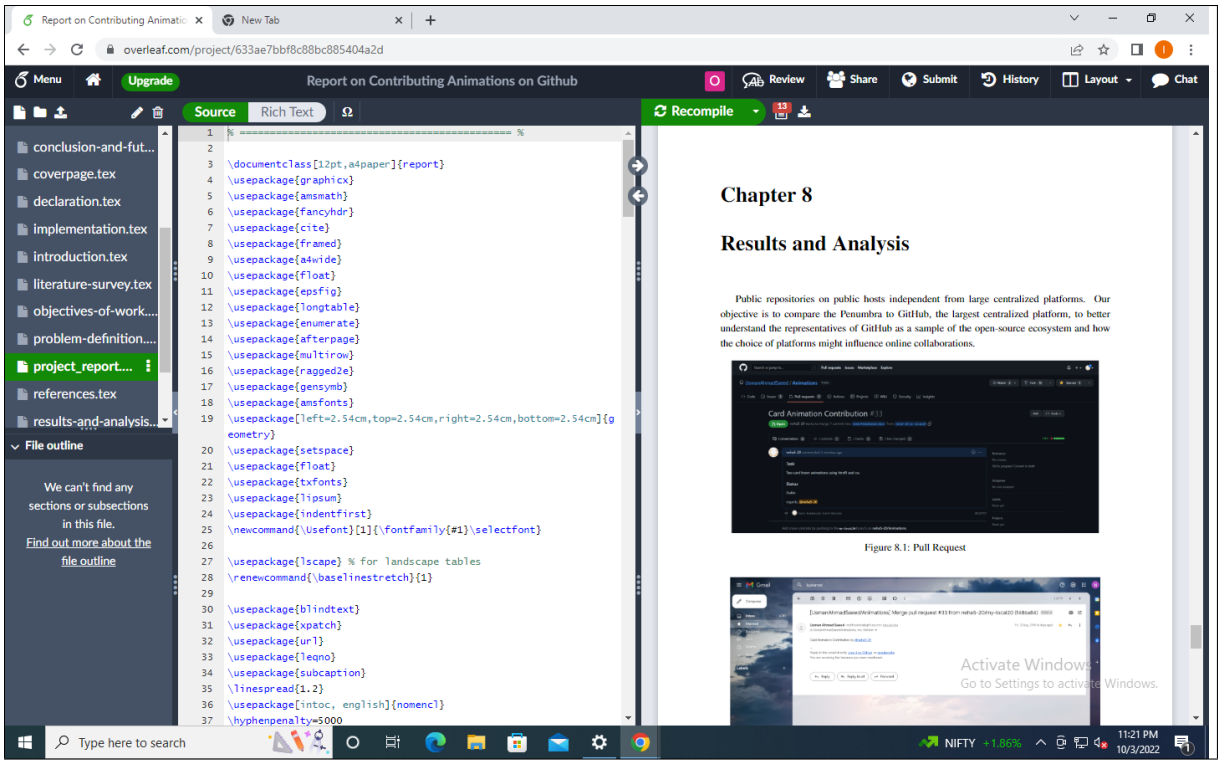


Figure 8.2: Result of images added in template

Chapter 9
Conclusion and Future Scope

9.1 Conclusion

Thus our project "Technical Project Report Template using Latex" concentrates on the documentation of project reports which includes project title page, certificate etc. The standard formatting constraints for these documents are defined in packages developed under this project, which will thus help the user to complete his/her work in stipulated time and making it less tedious. This template is used by other project groups of Information Technology department of Bharati Vidyapeeth College of Engineering. This template can also be used by other College or Universities by doing some changes in names and college logo.

9.2 Future Scope

In order to make use of class files developed in the project, the basic requirement is that the user must have the sound knowledge of LaTeX. So to make it more handy, user interface can be developed which will take only the required values as input viz; the content of notice and the packages in the back-end will take care of formatting without the user having to know LaTeX commands.

References

[1] https://www.overleaf.com/learn

[2] https://miktex.org/docs

[3] https://www.southampton.ac.uk/~fangohr/computing/latex/latex_for_report_writing_HF.pdf

[4] http://tug.ctan.org/tex-archive/macros/latex/contrib/famt/ProjectReport/Report.pdf

[5] https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.365.502&rep=rep1&type=pdf

[6] https://www.geos.ed.ac.uk/~hcp/latex_examples/gp_project_in_latex.pdf

[7] https://www.srmcollege.in/wp-content/uploads/2020/02/Report-wrtining-using-Latex.pdf

[8] https://www.phys.ufl.edu/courses/phy4803L/sample-paper.pdf