

Name of Institution
Department of Computer Studies

Collection of L^AT_EX Sections for Structured Academic and Technical Report Documents

A Template Collection

PREPARED BY:
Woodrow "Wooderson" Fajardo

SUBMITTED TO:
[Instructor / Supervisor Name]

**BLK1-FRGMT-001
NOVEMBER 2025**

Contents

Tables	ii
Figures	iii
1 Typography and Formatting	1
1.1 Text Sizes	1
1.2 Text Styles.....	1
1.3 Enumeration and Lists.....	1
2 Tables	3
2.1 Standard Professional Table (No Stripes)	3
2.2 Zebra Striped Table	3
2.3 Full Width Table	3
3 Figures and Images	4
3.1 Standard Figure.....	4
3.2 Side-by-Side Figures	4
3.3 Custom Styled Figure.....	5
4 Code Implementation.....	6
4.1 Inline Code.....	6
4.2 Multi-Language Support	6
4.3 System Output	7

List of Tables

1	A clean academic table using Booktabs.	3
2	Inventory list with alternating row colors.	3
3	A full-width table utilizing available margin space.	3

List of Figures

1	A standard centered figure.....	4
2	Comparative analysis showing two figures side-by-side.....	4
3	Extended detail overview of the figure shown below.....	5

1 Typography and Formatting

1.1 Text Sizes

The following demonstrates the standard L^AT_EX font sizing hierarchy:

- This is tiny text
- This is scriptsize text
- This is footnotesize text
- This is small text
- This is normalsize text (Default)
- This is large text
- This is Large text
- This is LARGE text
- This is huge text
- This is Huge text

1.2 Text Styles

Standard formatting options include:

- **Bold Text Example**
- *Italicized Text Example*
- Underlined Text Example
- ***Combined Bold and Italic***
- Monospaced/Typewriter Font
- SMALL CAPS TEXT

1.3 Enumeration and Lists

Using the `enumitem` package, we can customize list markers.

Arabic Numerals (Default)

1. First item (Standard)
2. Second item
3. Third item

Roman Numerals

- I. Major Point One
- II. Major Point Two
 - i. Minor point i
 - ii. Minor point ii

Alphabetical Lists

- A. Section A
- B. Section B
 - a) subsection a
 - b) subsection b

2 Tables

2.1 Standard Professional Table (No Stripes)

Table 1: A clean academic table using Booktabs.

Component	Material	Cost (\$)
Chassis	Aluminum	45.00
Microcontroller	Silicon	12.50
Power Supply	Li-Ion	22.00

2.2 Zebra Striped Table

This table uses the manila background color defined in the main file.

Table 2: Inventory list with alternating row colors.

Item Name	Stock ID	Quantity
Network Switch	NS-001	15
Patch Cable (5m)	PC-005	120
Server Rack	SR-990	4
Cooling Unit	CU-202	8
UPS Battery	UP-110	20

2.3 Full Width Table

This table automatically expands to fill the text width between margins.

Table 3: A full-width table utilizing available margin space.

Project Phase	Status	Completion Date
Phase 1: Planning	Complete	Jan 2024
Phase 2: Development	In Progress	Mar 2024
Phase 3: Testing	Pending	May 2024
Phase 4: Deployment	Pending	Jul 2024

3 Figures and Images

3.1 Standard Figure

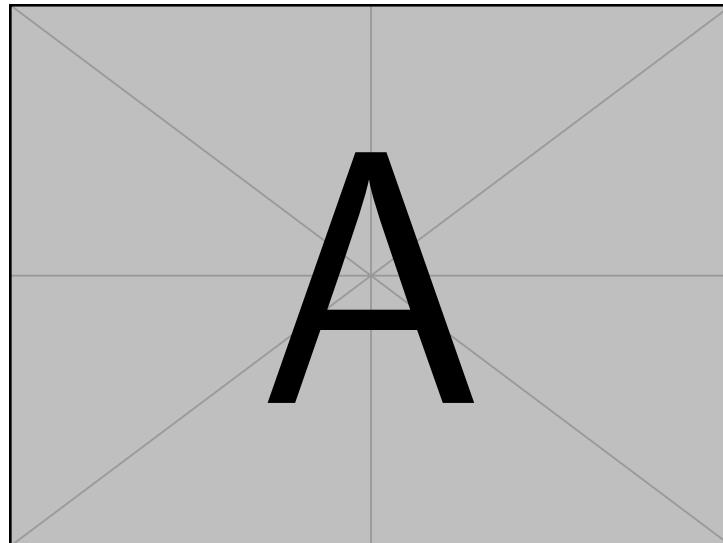
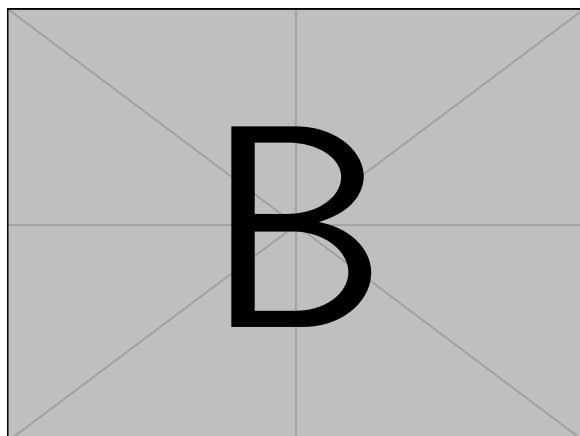


FIGURE 1

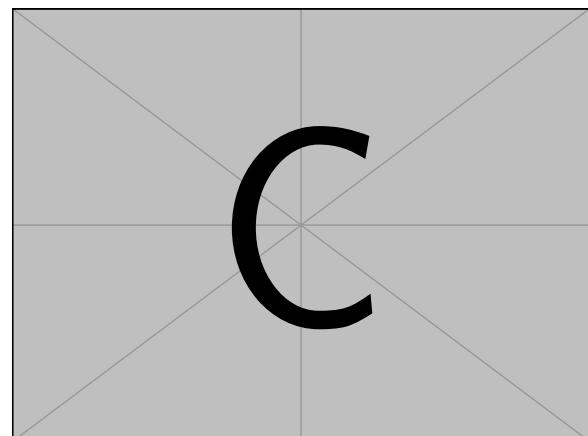
A standard centered figure.

Source: Generated by \LaTeX

3.2 Side-by-Side Figures



(a) Left Side Data



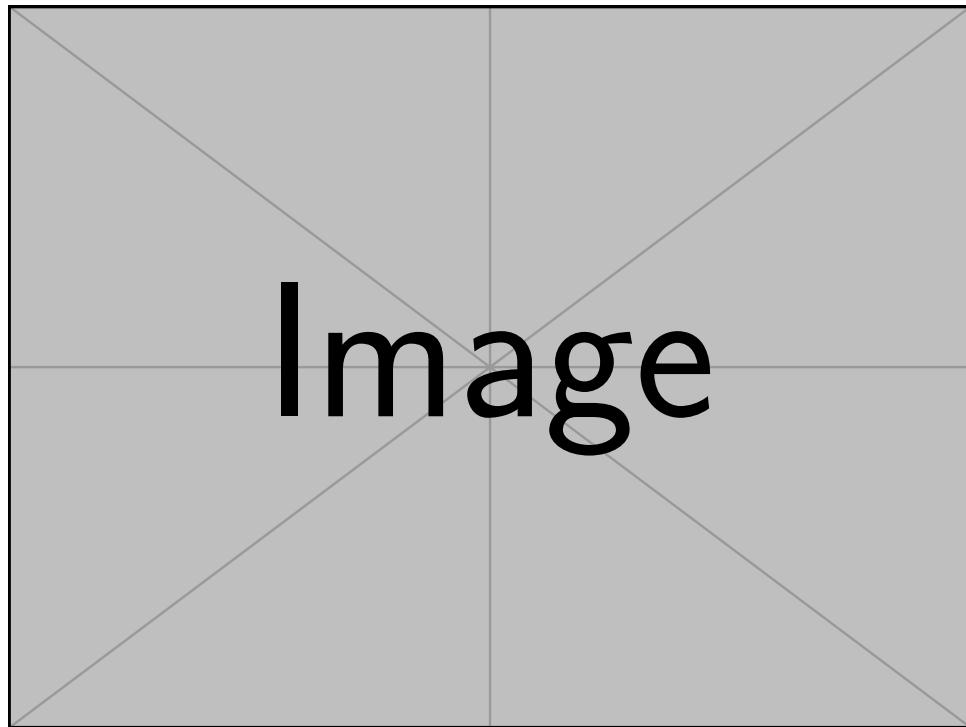
(b) Right Side Data

Figure 2: Comparative analysis showing two figures side-by-side.

3.3 Custom Styled Figure

FIGURE 3

Extended detail overview of the figure shown below...



Source: –Source of the Graph–

4 Code Implementation

4.1 Inline Code

You can mention variables or commands directly in the paragraph using **inline code styling**, which highlights the text with the Manila theme.

4.2 Multi-Language Support

MEMORY ALLOCATION ROUTINE

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main() {
5     int *ptr;
6     int n = 5;
7
8     // Dynamically allocate memory
9     ptr = (int*)malloc(n * sizeof(int));
10
11    if (ptr == NULL) {
12        printf("Memory not allocated.\n");
13        exit(0);
14    }
15
16    printf("Memory successfully allocated using malloc.\n");
17    free(ptr);
18    return 0;
19 }
```

DATA PROCESSING SCRIPT

```

1 import pandas as pd
2 import numpy as np
3
4 def calculate_metrics(data_frame):
5     # Calculate basic statistics
6     mean_val = np.mean(data_frame['value'])
7     max_val = np.max(data_frame['value'])
8
9     print(f"Processing Complete.")
10    print(f"Mean: {mean_val:.2f} | Max: {max_val}")
11    return mean_val
12
13 # Execute
14 df = pd.DataFrame({'value': [10, 20, 30, 40, 50]})
15 calculate_metrics(df)
```

STATISTICAL MODELING

```
1 # Load necessary library
2 library(ggplot2)
3
4 # Create sample dataset
5 data <- data.frame(
6   x = c(1, 2, 3, 4, 5),
7   y = c(2, 4, 6, 8, 10)
8 )
9
10 # Linear Regression Model
11 model <- lm(y ~ x, data = data)
12
13 # Output summary
14 print(summary(model))
```

4.3 System Output

TERMINAL OUTPUT

```
user@workstation:~/projects$ gcc main.c -o main
user@workstation:~/projects$ ./main
Memory successfully allocated using malloc.
user@workstation:~/projects$ _
```