



## PROJECT AND TEAM INFORMATION

### Project Title

*File compression and decompression using Huffman coding*

### Student / Team Information

<i>Team Name:</i> <i>Team #</i>	<i>BitBusters</i> <a href="#"><u>DAA-IV-T145</u></a>
<b>Team member 1 (Team Lead)</b> <i>(Last Name, name: student ID: email, picture):</i>	<i>Chaudhary, Manvi – 23022908</i> <a href="mailto:nainachaudharydehradun@gmail.com"><u>nainachaudharydehradun@gmail.com</u></a> 
<b>Team member 2</b> <i>(Last Name, name: student ID: email, picture):</i>	<i>Nivedan, Harsh – 230211585</i> <a href="mailto:harshnivedan@gmail.com"><u>harshnivedan@gmail.com</u></a> 

**Team member 3**

(Last Name, name: student ID: email, picture):

Agarwal, Vedit Saran – 23021864

[viditagarwal60@gmail.com](mailto:viditagarwal60@gmail.com)

**Team member 4**

(Last Name, name: student ID: email, picture):

Mittal, Rishika – 23022417

[rishikamittal2005@gmail.com](mailto:rishikamittal2005@gmail.com)



## PROPOSAL DESCRIPTION (10 pts)

### Motivation (1 pt)

*In today's data-driven world, handling large volumes of text-based information has become routine — whether it's logs, CSV datasets, or configuration files. However, as the amount of textual data grows, so does the storage requirement. Our project aims to tackle this problem by offering an efficient and lossless method to compress text files using **Huffman Coding**. It not only saves disk space but also improves the speed of data transfer and storage efficiency — especially important for cloud-based systems or resource-limited devices.*

### State of the Art / Current solution (1 pt)

*Currently, text compression is widely handled by utilities like **ZIP, GZip, and 7-Zip**, which internally use algorithms like **LZ77** or variants of **Huffman Coding**. However, these are general-purpose tools. Our approach focuses entirely on **pure Huffman Coding**, giving users and developers a clear and educational implementation that highlights how the algorithm actually works under the hood — ideal for learning, research, and lightweight applications.*

### Project Goals and Milestones (2 pts)

- Build a tool that can **compress and decompress text files** using Huffman Coding.
- Support file formats like `.txt`, `.csv`, `.json`, `.xml`, `.html`, `.log`, `.md`, and `.yaml`.
- Provide a **user-friendly interface** for selecting files and viewing results.
- Include features for viewing original size, compressed size, and compression ratio.
- Key milestones:
  - Week 1: Research & algorithm implementation
  - Week 2–3: Build compression & decompression modules
  - Week 4–5: Test with various file types and measure performance
  - Week 6–7: Build UI and integrate
  - Final Week: Documentation, optimization, and submission

## Project Approach (3 pts)

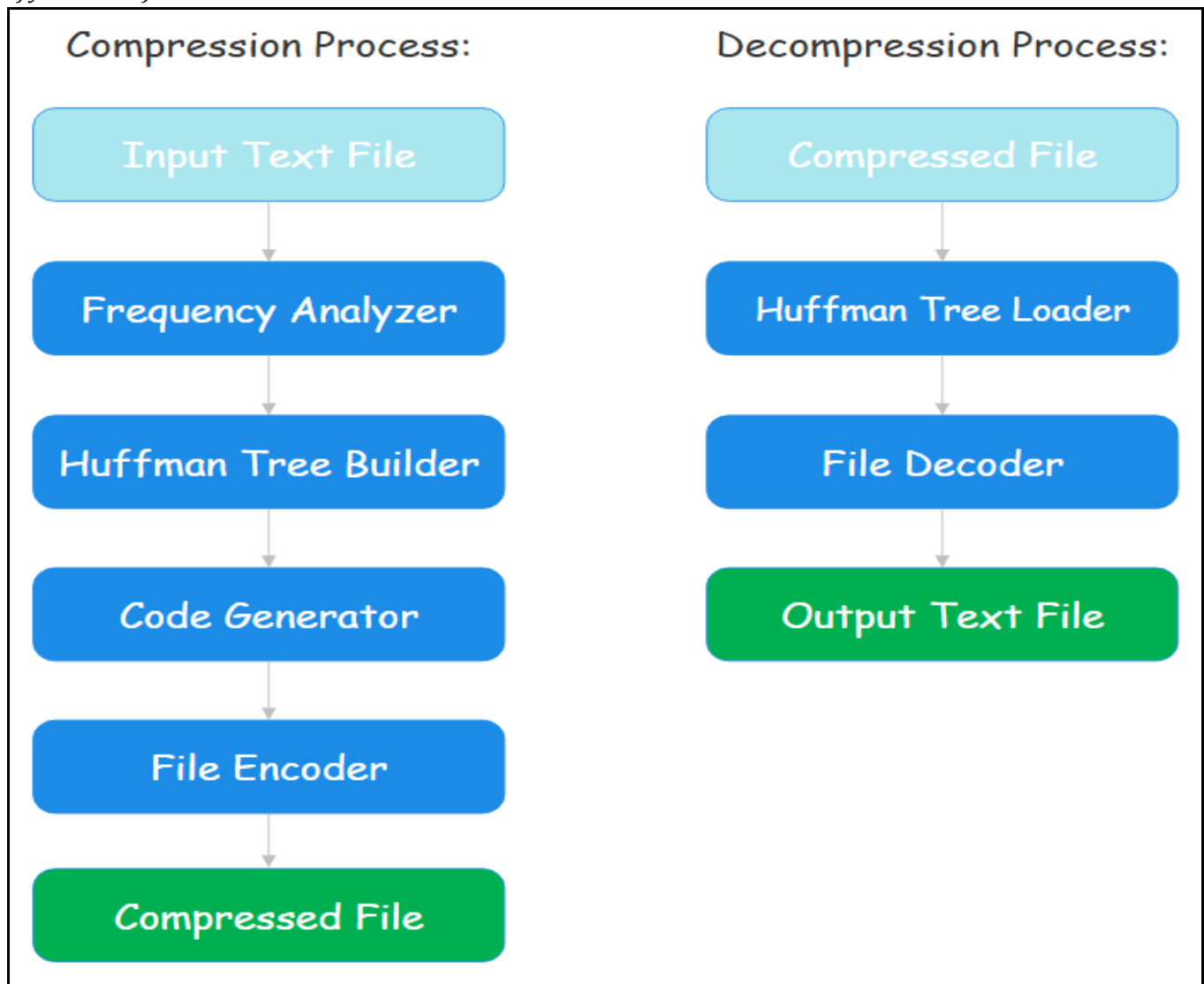
We will develop this project using **C++**, a powerful language well-suited for implementing algorithmic logic and memory-efficient data structures. The tool will:

- Read text files and count the frequency of each character.
- Build a **Huffman Tree** based on these frequencies.
- Generate prefix codes for each character and encode the file accordingly.
- Provide a decoding module to reconstruct the original text from the compressed data.

Optionally, we may also design a **simple web interface** using **HTML, CSS, and JavaScript** to allow users to upload files and visualize compression results. However, the core logic and processing will be done in C++. Testing will focus on various text formats like .txt and .csv to ensure accuracy and efficiency.

## System Architecture (High Level Diagram)(2 pts)

(Provide an overview of the system, identifying its main components and interfaces in the form of a diagram using a tool of your choice).



## Project Outcome / Deliverables (1 pts)

*By the end of this project, we'll deliver:*

- *A fully functional text file compression/decompression tool.*
- *Support for .txt, .csv, .json, .xml, .html, .log, .md, and .yaml formats.*
- *GUI interface for ease of use.*
- *Codebase with documentation and performance analysis (compression ratio, runtime, etc.)*

## Assumptions

- *The tool is built exclusively for **text-based files**.*
- *Input files are expected to be **UTF-8 encoded**.*
- *Works best with **English characters and symbols**.*
- *We assume users don't require compression of already compressed or multimedia files.*

## References

- [1] [Wikipedia: Huffman Coding](#)
- [2] [Text File Compression And Decompression Using Huffman Coding | GeeksforGeeks](#)
- [3] [Python Tkinter Documentation](#)
- [4] *Lecture notes and course materials on Algorithms and Data Structures*