PROJECT AND TEAM INFORMATION

Project Title

Efficient File Compression Using Huffman Encoding

Student / Team Information

Team Name: *Team #*	*Bytesquashers*
Team member 1 (Team Lead)	*Ashish Gupta* *9411742244* *theashish2004@gmail.com*

Team member 2	***Aditya Kimothi*** ***7505726811*** *adityakimothi31@gmail.com*
Team member 3	*Sameer Lohani* *7983956049* *sameerlohani200510@gmail.com*

PROPOSAL DESCRIPTION

Motivation

In today's digital age, efficient storage and transfer of data is a critical need. Large files consume considerable space and bandwidth, making file compression essential. This project aims to solve the problem of large file sizes using Huffman encoding, a proven lossless compression algorithm. The tool will be particularly useful in cloud storage and network transmission scenarios, where reducing file size can lead to significant cost and performance benefits.

State of the Art / Current solution

Currently, tools like WinRAR, 7-Zip, and gzip are used for file compression. While these are effective, they are closed-source and complex. Huffman coding, although a classical algorithm, is not commonly available in educational or customizable tools. Our project aims to implement this technique in a lightweight, educational, and open-source format.

Project Goals and Milestones

Develop a compression and decompression engine using Huffman encoding in C++.

Build a user-friendly web interface for file upload and download.

Integrate front-end with back-end to provide compression/decompression through a browser.

Conduct testing on various file types.
Milestones:

Week 1: Design and algorithm implementation

Week 2: File compression engine

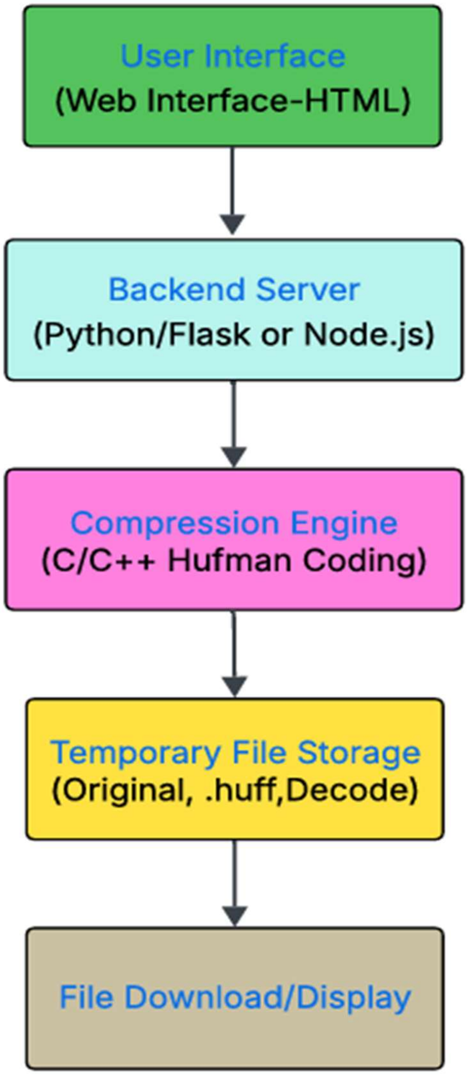
Week 3: Web integration

Week 4: Testing and final adjustments

Project Approach

We will use C++ for backend logic implementing the Huffman compression algorithm. HTML, CSS, and JavaScript will be used for the frontend to allow file uploads and downloads through a web interface. Node.js and Express.js may be used to bridge the front-end and the compression engine. The focus will be on making the system lightweight, fast, and easy to use.

System Architecture (High Level Diagram)



User Interface
(Web Interface-HTML)

Backend Server
(Python/Flask or Node.js)

Compression Engine
(C/C++ Hufman Coding)

Temporary File Storage
(Original, .huff,Decode)

File Download/Display

Project Outcome / Deliverables

C++ application for Huffman compression and decompression.

Web interface for file interaction.

Fully working demo hosted locally.

Project documentation and user manual.

Assumptions

Assumptions We assume the input files are binary-readable and that the system has basic support for running C++ executables through a server layer.

References

Huffman Coding Algorithm - Wikipedia

GeeksforGeeks: Huffman Coding Implementation

MDN Web Docs for HTML/CSS/JS

Node.js and Express.js documentation

C++ File I/O official documentation