Graphic Era Hill University
DEHRADUN • BHIMTAL • HALDWANI

PROJECT AND TEAM INFORMATION

Project Title

(Try to choose a catchy title. Max 20 words).

ASM Fusion – Assembly to C code Translator

Student/Team Information

Team Name:	**Byte - Crafters**
Team member 1 (Team Lead) (Aaryan Goel, 22011987, aaryan.goel03@gmail.com):	
Team member 2 (Rishabh Pant, 22011962, rishabhpant1395@gmail.com):	

Team member 3 (Manya Rajput, 22012668, manyarajput.9b@gmail.com):	
Team member 4 (Priyanshu Vishwakarma, 220111135, priyanshuvi420@gmail.com):	

PROJECT PROGRESS DESCRIPTION

Project Abstract

(Brief restatement of your project's main goal. Max 300 words).

ASM Fusion is an innovative dual-interface translator that converts assembly language code into equivalent C code. Designed for both academic and practical purposes, it includes a terminal-based backend which processes input from input.asm and writes to output.c. Alongside, it features a real-time interactive GUI implemented in HTML/CSS/JavaScript to provide live code conversion. This project helps learners and professionals understand low-level hardware instructions in high-level logic, bridging educational and technical gaps.

Updated Project Approach and Architecture

(Describe your current approach, including system design, communication protocols, libraries used, etc. Max 300 words).

- **Backend**: Built lexer and parser using C++ (lexer_parser.cpp) with modular headers (asm_fusion.h, converter.h) and logic in converter.cpp. The parsing and instruction-mapping logic handles core assembly instructions.
- **Frontend**: Implemented in index.html, styled with CSS. JavaScript handles tokenization, line numbering, and live conversion using an in-browser interpreter.
- **Flow**: Terminal mode reads input.asm and writes to output.c. GUI mode offers live conversion with buttons to download .asm and .c files.
- **Libraries**: Standard C++ STL, HTML5, CSS3, vanilla JavaScript (DOM API).

Tasks Completed

(Describe the main tasks that have been assigned and already completed. Max 250 words).

Task Completed	Team Member
Lexer Module Implementation →	Aaryan Goel & Rishabh Pant
Parser Module Implementation →	Aaryan Goel & Rishabh Pant
Instruction to C conversion →	Manya Rajput & Priyanshu Vishwakarma
Basic Instruction Set Integration →	Manya Rajput & Priyanshu Vishwakarma
GUI Implementation →	Each member contributed

Challenges/Roadblocks

(Describe the challenges that you have faced or are facing so far and how you plan to solve them. Max 300 words).

- **Instruction Variability**: Handling numerous variations of assembly instructions required extensive mapping and validation logic.
- **Edge Cases**: Incorrect or malformed instructions needed fallback handling to avoid breaking conversion.
- **Frontend Syncing**: Ensuring real-time GUI translation and updating stats without lag required JavaScript debouncing and DOM optimization.
- We plan to refine the parser with regex/token grammar for better robustness.

Tasks Pending

(Describe the main tasks that you still need to complete. Max 250 words).

Task Pending	Team Member (to complete the task)
N/A	N/A

Project Outcome/Deliverables

(Describe what are the key outcomes / deliverables of the project. Max 200 words).

- A working C++ CLI tool to convert assembly code (input.asm) to output.c.
- A fully functional web GUI supporting live translation, syntax stats, and file download.
- Core code organized into header + logic files, ready for extension.
- Final demo video and user documentation.

Progress Overview

(Summarize how much of the project is done, what's behind schedule, what's ahead of schedule. Max 200 words.)

- ~100% of the project is completed.
- Backend logic and GUI are both functional.
- Code optimization and packaging are completed.
- GUI styling and validation handling are completed.

Codebase Information

(Repository link, branch, and information about important commits.)

Repository Link: https://github.com/RishabhPant1/ASM-Fusion
Branch: Btech CSE
Important Files:
lexer_parser.cpp: Tokenization and Parsing.
converter.cpp: Core conversion logic
converter.h and asm_fusion.h: Data structures and declarations.
index.html: Full-feature GUI for live conversion.

Testing and Validation Status

(Provide information about any tests conducted)

Test Type	Status (Pass/Fail)	Notes
Manual Unit Tests	Pass	Validated common assembly instructions like MOV, MVI, ADD.
GUI Interaction	Pass	Tested live input, scrolling sync, line stats.
Edge Instruction Tests	Pass	Some rare instructions.

Deliverables Progress

(Summarize the current status of all key project deliverables mentioned earlier. Indicate whether each deliverable is completed, in progress, or pending.)

- CLI Translator – ✅
- Live Web GUI – ✅
- Unit Test Suite – ✅
- User Documentation – ✅
- Code Packaging (optional EXE/Electron) – ✅