CONSTRUCTION COMPILER LAB TERMINAL

SUBMITTED BY: FAJAR AAMIR SHEIKH

RGISTRATION NO: SP22-BCS-031

SUBMITTED TO : SIR BILAL BUKHARI

SUBMISSION DATE: 18TH JUNE2025

QUESTION NO 1:

Project Overview

- A **Mini Compiler** is a simplified tool that mimics the functioning of a real-world compiler for educational or prototype purposes.
- It takes a small, structured subset of a programming language as input and translates it into an intermediate representation or detects syntax/semantic errors.
- This project helps students understand the key phases of compilation such as lexical analysis, syntax parsing, and semantic validation.

Key Objectives

- To develop a tool that can read and analyze input code.
- To simulate the basic phases of a compiler.
- To identify and display syntax or semantic errors.
- To provide intermediate output such as token streams or parse trees.
- To enhance understanding of compiler design concepts.

Phases of Compilation (Mini Compiler)

Phase	Description
Lexical Analysis	Tokenization using finite automata or regex
Syntax Analysis	Grammar rules via parsers (LL(1), Bottom-Up)
Semantic Analysis	Type checks, undeclared variables, etc.
Intermediate Code (Optional)	Generation of simplified pseudo-code
Symbol Table	Tracks variables, types, scopes

Advantages

- Demonstrates **compiler phases in isolation**.
- Encourages **modular design** and clear thinking about language rules.
- Provides **error diagnostics** for code input.

Limitations

- Only supports a **small subset** of language grammar.
- Not capable of **full code optimization** or **machine code generation**.