**Simple Compiler project Documentation**

**Team Members:**

* **Yahiaelreedy@gmail.com 222189**
* **seifeldin.amr3@msa.edu.eg 224703**
* **ahmed.abdelmoneim2@msa.edu.eg 223135**
* **aly.essameldi@msa.edu.eg 221071**

**Description:**

This project is a compiler developed using Python and Gradio. It processes input code to perform lexical analysis, syntax checking, and basic semantic analysis. It identifies variables, reserved words, and symbols, handles array definitions and operators, and checks for common errors such as missing semicolons and invalid case statements in switch-case structures.

**Code Structure**

code scanning function **scanString** and the setup of a graphical user interface (GUI) using the Gradio library for a compiler design project. The GUI allows users to input code, scan it for various elements, and receive outputs including code explanations, error lists, and a memory table.

**Function: scanString(Code)**

This function scans the input code for various syntactical and structural elements, checks or errors, and generates outputs that detail the findings of the scan.

**Parameters**: **Code** (str): The code to be scanned as a string.

**Scanner Initialization:**

A Scanner object s is instantiated.

Various checks are performed using methods of the Scanner class:

* + - **VariableCheck(Code)**
    - **IdentfiersCheck(Code)**
    - **SymbolsCheck(Code)**
    - **ArrayChecker(Code)**
    - **NumberCheck(Code)**
    - **simicolon(Code)**
    - **checker(Code)**
    - **definemuloperatorArray(Code)**
    - **defineArray(Code)**
    - **define(Code)**
    - **definemuloperator(Code)**
    - **ReserverdWordCheck(Code)**

**Table Generation:**

* Creates an HTML table to display memory values.
* Splits the string data and formats it for output.

**Error Handling:**

* Collects and clears errors from the Scanner object.
* Checks for bracket mismatches using the check\_brackets function.

**Cleanup:**

* Clears various lists and memory data from the Scanner object.
* Deletes the Scanner object to free resources.

**Return Values:**

* + Returns a tuple containing:
    - **done** (str): Processed code string with newline-separated elements.
    - **errStmt** (str): Concatenated error messages.
    - **table** (str): HTML table of memory values.

**Gradio Interface**

The GUI is set up using the Gradio library to provide a user-friendly interface for the compiler project.

**Components**

1. **Code Input**:
   * **gr.Code(label="Code")**: Text area for user to input code.
2. **Output Display**:
   * **gr.Text(label="Code Explanation")**: Text box to display the explanation of the scanned code.
   * **gr.Text(label="Error List")**: Text box to display the list of errors found in the code.
   * **gr.HTML(label="Memory Table")**: HTML container to display the memory table.
3. **Buttons**:
   * **gr.Button("Scan...", variant="primary")**: Button to initiate the scanning process.
   * **gr.Button("Clear")**: Button to clear the inputs and outputs.