|  |
| --- |
|  |
| PROJECT PROPOSAL |
|  |
|  |
|  |

***Syed Samama Ali B19103062***

***Sufiyan Mohammad Salman B20103065***

***Abu Ubaida B20103007***

***Muhammad Ibrahim khan B20103041***

***Muhammad Shehroz B20103052***

***Faraz Javed B20103022***

***Muhammad Mairaj B20103043***

***Mohammad Zoren Khan B20103033***

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | ***Introduction***  Compilers are an essential part of programming languages that convert source code into executable code. The development of a compiler is a complex and challenging task, requiring a thorough understanding of programming languages, syntax analysis, and code generation.  ***Objectives***  The primary objectives of this project are to:   * Develop a compiler that can convert source code written in a specific programming language into executable code. * Understand the theory and principles behind compilers, such as syntax analysis, semantic analysis, and code generation. * Implement the various phases of the compiler, such as lexical analysis, syntax analysis, semantic analysis, and code generation. * Test and validate the compiler using a sample set of programs and ensure it produces the expected output.   ***Methodology***  The compiler will be developed using Java and JavaScript. React will be used for the front-end of the compiler, which will include lexical analysis, syntax analysis, and semantic analysis. Java will be used for the back-end of the compiler, which will include code generation and optimization.  The project will be divided into the following ***phases***:   * Design the overall architecture of the compiler, including the front-end and back-end components. * Implement the lexical analysis phase, which will involve tokenizing the source code into a sequence of tokens. * Implement the syntax analysis phase, which will involve parsing the sequence of tokens and building an abstract syntax tree (AST). * Implement the semantic analysis phase, which will involve analyzing the AST and verifying the semantics of the source code. * Implement the code generation phase, which will involve generating executable code from the AST. * Optimize the generated code to improve its performance. * Test and validate the compiler using a set of sample programs.   ***Early Language Specification***  I. Words  a. Keywords are reserved words in the programming language that have special meanings and cannot be used as variable names. This language has the following keywords:  •if  •else  •for-each  •while  •for  •continue  •break  •return  •void  •class  •interface  •final  •static  •public  •private  •protected  •extends  •implements  •try  •catch  •finally  •throw  •this  •super  b. Datatypes are used to define variables and their values in a program, but we might prefer dynamic data type declaration.   * int - Integer * float - Float * char – Character * String – Strings * Arr – Array   II. Operators   * Arithmetic   +, -, \*, /, %   * Relational   <, <=, >, >=, !=, ==  III. Punctuators are used to separate and structure code in a program. This language has the following punctuators:  •Semicolon (;): used to terminate lines of code  •Comma (,): used to separate items in a list or function arguments  •Dot (.): used to access attributes and methods of an object  •Parentheses (): used to declare functions and pass arguments  •Braces {}: used to define the scope of methods and code blocks  •Brackets []: used to index items in an array  •Colon (:): used in some cases to separate code statements  PROJECT SCOPE  ***Main* *Idea***  The scope of this project is to develop a compiler, for a custom programming language, using Java and JavaScript with React used for the front-end. This compiler will be designed to allow simple program compilation. The compiler will go through several phases, including lexical analysis, syntax analysis, semantic analysis, and code generation, all of this will be implemented in Java. The front-end developer will use React to provide a user interface for inputting the source code and displaying the output.  The overall architecture of the compiler will be designed to be scalable and modular, allowing for easy future modifications and improvements. The compiler will be tested and validated using a set of sample programs to ensure that it produces the expected output. Additionally, the generated code will be optimized for improved performance.  The primary goal of this project is to build a fully functional compiler for a custom programming language that can convert source code into executable code. The secondary goal is to provide a user-friendly front-end interface for inputting source code and displaying output. Overall, the scope of the project is to build a high-quality, efficient, and reliable compiler for a custom programming language.  ***Target* *audience***  The target audience for this custom programming language compiler includes anyone interested in programming languages, software development, and computer science, who may benefit from a tailored programming language or learning about the inner workings of compilers. Additionally, the front-end interface provided by React may be useful for those who prefer a user-friendly way of inputting and displaying source code and output.  ***Timeframe***  The project will be completed over a period of three months, starting in March and ending in June. The timeline for the various phases of the project is as follows:  March: Design and architecture  April: Lexical analysis and syntax analysis  May: Semantic analysis, code generation, and testing  ***Conclusion***  In conclusion, the proposed project aims to build a compiler for a custom programming language using aforementioned languages. The language specification provides the details of the language, which will be used to develop the compiler. The compiler will have three major components: the lexical analyzer, the parser, and the code generator. The compiler will be tested by writing programs in the custom programming language and compiling them using the compiler. | |  |