**4/12/2023**

Shri – An AI Code Generator

**Krishnakant Dubey 2817**

Introduction

**Shri - An AI Code Generator** is a project that aims to simplify the process of writing code by using machine learning and natural language processing to generate code snippets quickly and accurately. This application provides a user-friendly interface for developers to input their code requirements in natural language and receive code output. The generated code is validated to ensure that it is syntactically correct and semantically meaningful.

The project consists of several modules, including the user interface, natural language processing, code generation, and code validation. The application is built using Python as the primary programming language, while libraries such as TensorFlow is used for natural language processing and machine learning.

List of Modules and Functions

**User Interface**

The user interface module provides a user-friendly interface for users to input their code requirements in natural language and receive code output. The user interface is developed using the Flask web framework, which provides a lightweight and scalable platform for building web applications. The user interface module consists of several functions, including:

**User Input**: This function receives user input in natural language and passes it to the natural language processing module for further processing.

**Code Output**: This function receives the generated code from the code generation module and displays it to the user in a readable format.

**Natural Language Processing**

The natural language processing module interprets the natural language input and converts it into a structured format that can be used to generate code. This module uses the TensorFlow library for natural language processing and tokenization. The natural language processing module consists of several functions, including:

**Tokenization**: This function breaks down the natural language input into individual tokens, which can be used to identify keywords and syntax.

**Parsing**: This function analyzes the structure of the natural language input to identify its grammatical structure and the relationships between different words.

**Entity Recognition**: This function identifies entities in the natural language input, such as variables and functions, which can be used to generate code.

**Code Generation**

The code generation module generates code based on the structured input from the natural language processing module. This module uses a machine learning model trained on a dataset of code snippets to generate code that is syntactically correct and semantically meaningful. The code generation module consists of several functions, including:

**Code Generation**: This function generates code based on the structured input from the natural language processing module.

**Code Snippet Selection**: This function selects the most appropriate code snippet from the machine learning model's output based on the user's input.

**Code Validation**

The code validation module validates the generated code to ensure that it is syntactically correct and semantically meaningful. This module uses static analysis techniques to identify potential errors and inconsistencies in the generated code. The code validation module consists of several functions, including:

**Syntax Checking**: This function checks the generated code for syntax errors and provides feedback to the user if any errors are detected.

**Semantic Checking**: This function checks the generated code for semantic errors, such as incorrect variable types or incorrect function usage.

Utility of Application

**Shri** provides several benefits to developers and non-technical users alike. Some of the main benefits include:

**Time-Saving**: The application generates code quickly and accurately, saving developers time and effort.

**Learning Tool**: The application can be used to learn how to code by providing real-time feedback on code syntax and structure.

**Multi-Language Support**: The application can be used across a variety of programming languages and can generate code for a range of use cases.

**Non-Technical User Support**: The application can be used by non-technical users to generate code for basic use cases without requiring advanced programming skills.

**Free of Cost**: Aims to provide a free AI-based code generator service as existing options, such as Codex and Copilot, are not free.

Literature Review

**More sophisticated machine learning model:** Shri uses a model specifically trained for code generation, enabling it to understand code context and generate more accurate and efficient code.

**Wide language support:** Shri is built using Python, a widely used programming language.

**Free of charge:** Shri is available for use by anyone without cost, removing a potential financial barrier for developers looking to utilize code generation tools. This makes it more accessible to a wider range of developers, particularly those with limited financial resources.

**Open source:** Shri is an open-source project, allowing developers to modify and improve the code according to their needs. This promotes collaboration and innovation within the developer community, resulting in a constantly improving tool.

Overview of Shri: Features, Dataset, Limitations, and Security Measures

**Clarification**: It is important to note that Shri is intended to provide basic functionality in its initial stages, but with time and further development, it has the potential to support more complex algorithms as well. Additionally, the application will only support Python language in its initial stages, but the goal is to expand to other programming languages in the future.

**Dataset used**: The Hugging Face GitHub-Code dataset is being used for training Shri's machine learning model. This dataset consists of over 115 million code files, with a total size of over 1TB.

The dataset is diverse and covers various programming languages and coding scenarios, which allows for a more accurate and effective model.

**Limitations**: It is important to note that while Shri is designed to simplify the process of writing code, there may be limitations to its functionality. For example, the application may not be able to handle very complex programming scenarios or edge cases. Additionally, the generated code may not always be optimal or efficient, and may require further manual optimization by the developer.

**Security and Privacy**: Shri takes security and privacy seriously and employs several measures to protect user data. The application stores user data in a secure NoSQL database, which is designed for fast and efficient data storage and retrieval. Additionally, the application uses SSL encryption to protect user data during transmission. However, it is important to note that there may still be potential security risks, such as vulnerabilities in third-party libraries or unauthorized access to the server, and users should take necessary precautions to protect their data.

Technology Details

**Shri - AI Code Generator** application is built using Python as the primary programming language, with libraries such as TensorFlow is used for natural language processing and machine learning. The application also utilizes the Flask web framework for the user interface and static analysis libraries for code validation. The code generation module uses a pre-trained machine learning model based on a dataset of code snippets to generate code. The dataset is diverse and covers various programming languages and coding scenarios. The model was trained using deep learning techniques, including neural networks, to ensure high accuracy and effectiveness in generating code.

The application is designed to run on multiple operating systems and can be deployed on a local server or cloud-based platform. The application uses a microservices architecture, which allows for scalability and flexibility in adding new features or modules. The database used for the application is a NoSQL database, which provides fast and efficient data storage and retrieval.

Group Members

I am the only member of this project, Krishnakant Dubey. However, I intend to collaborate with several experts in the fields of natural language processing, machine learning, and software development to ensure that the application is accurate, effective, and user-friendly.

Awareness of Technology in Group

As the sole member of this project, I have knowledge in natural language processing, machine learning, and software development. I have completed several courses and certifications in these fields and have worked on several projects that involve these technologies. Additionally, I am actively involved in the community of developers and researchers in these fields, which allows me to stay up-to-date on the latest developments and advancements. I have also collaborated with experts in these fields to ensure that the application is built using best practices and the latest techniques.