



## **AKILESH**

**Final Project** 



# PROJECT TITLE



Transformers for Code Generation&Language Understanding

# **AGENDA**

Week 1:Plannning

Define Goals: Objectives and scope.

Literature Review: Research overview.

Data Setup: Preprocess datasets.

Environment: Setup tools.

Week 2-3: Model Development

Transformer: Implement model.

Training: Train on data.

Metrics: Define performance metrics.

Week 4-5: Fine-tuning

Fine-tuning: Optimize on specific data.

Optimization: Improve efficiency.

Testing: Validate model.

Week 6-7: App Development

Code Gen App: Build code tools.

Lang App: Create language tools.

UI Design: Design interfaces.

Week 8: Documentation

Docs: Detail project.

Report: Compile report.

Presentation: Prepare summary.

Week 9: Review

Internal Review: Identify issues.

Feedback: Revise as needed.

Week 10: Completion

Final Testing: Validate apps.

Delivery: Deliver project.

Presentation: Showcase to stakeholders.



## PROBLEM STATEMENT

This project focuses on leveraging transformer-based models for code generation and language understanding. Despite advancements in natural language processing (NLP), there's a gap in effectively applying these models to automate code writing and text analysis tasks. The challenge is to develop optimized transformer models capable of generating, completing, and translating code from natural language descriptions, while also performing tasks like sentiment analysis, named entity recognition (NER), and question answering accurately. By implementing tailored models, developing user-friendly applications, and documenting methodologies, the project aims to enhance software development and NLP. Constraints include dataset availability, computational resources, and project timeframe.



## PROJECT OVERVIEW

This project aims to harness transformer-based models to automate code generation and enhance language understanding. By developing optimized models, creating user-friendly applications, and documenting methodologies, the project seeks to advance software development and NLP capabilities. The initiative addresses the gap in effectively applying NLP to specialized domains, benefiting software developers, data scientists, and industry professionals. Constraints include dataset availability, computational resources, project timeframe, and the need for tailored solutions in code writing and text analysis tasks.



### WHO ARE THE END USERS?

The End users of this project include:



**Software Developers:** Those who will utilize the code generation and completion tools to streamline their coding process.

**Data Scientists:** Individuals who will benefit from the language understanding tools for tasks such as sentiment analysis, named entity recognition, and question answering.

**NLP Researchers:** Professionals and academics interested in exploring the capabilities and advancements in transformer-based NLP models.

**Industry Professionals:** Those in the software and NLP domains who can integrate the developed applications into their workflow to improve efficiency and accuracy.

**Academia:** Researchers, educators, and students who can leverage the tools and methodologies developed in this project for further studies and applications in the field of NLP and software development.

## YOUR SOLUTION AND ITS VALUE PROPOSITION





We are developing transformer-based models optimized for:

- Code Generation: Automating code writing and translation based on natural language.
- Language Understanding: Enhancing tasks like sentiment analysis, named entity recognition (NER), and question answering.

#### Our project offers:

- Efficiency: Streamlining code writing and enhancing accuracy in language tasks.
- User-Friendly Applications: Intuitive tools for software developers, data scientists, and industry professionals.
- Advancement in NLP: Filling the gap in specialized transformer-based NLP applications.

# THE WOW IN YOUR SOLUTION

- **Seamless Integration:** Our tools will seamlessly integrate into existing development environments, providing real-time code suggestions and analysis.
- **Personalization:** The models will adapt and learn from user interactions, providing personalized and context-aware suggestions and insights.

We aim to revolutionize productivity and effectiveness in software development and NLP domains.

## MODELLING

### **Data Collection and Preprocessing**

Datasets: Gather and preprocess datasets.

#### Transformer Model Architecture

Customized Transformer: Implement and tailor the model for NLP tasks.

### Training and Fine-tuning

- Training: Train on datasets.
- Fine-tuning: Optimize for code generation and language tasks.

#### **Optimization Techniques**

- · Efficiency: Apply pruning and quantization.
- Personalization: Incorporate user interaction for adaptive suggestions.

#### **Evaluation Metrics**

• Performance Metrics: Monitor accuracy and BLEU/F1 scores.

#### **Application Development**

- Code Generation App: Develop code completion and translation features.
- Language App: Create tools for sentiment analysis, NER, and QA.

# **RESULTS**

### **Expected Results**

#### **Performance**

 Accuracy: Achieve high accuracy in code generation and language understanding tasks.

#### Efficiency

• Speed: Improve model efficiency through optimization techniques like pruning and quantization.

#### **User Experience**

Personalization: Deliver adaptive and personalized suggestions through user interaction.

#### **Application Impact**

• Utility: Develop user-friendly applications that streamline code writing and enhance language analysis tasks.

#### **Documentation**

 Comprehensive Report: Provide detailed documentation and a comprehensive project report.

### **Demo Link**