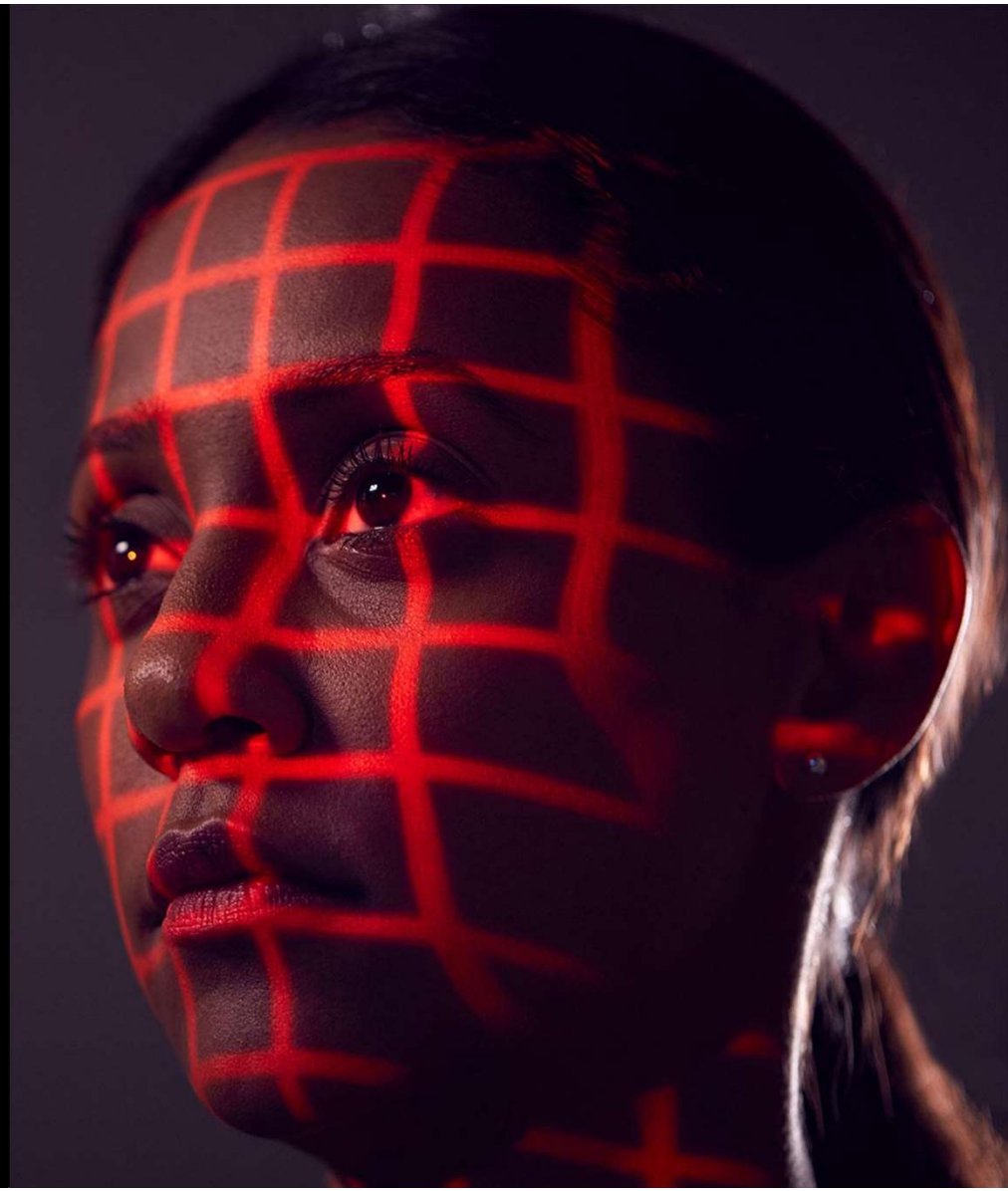
Let's go further

Team Name:
JD CODER



Add your Team members details in this slide

1. Jabasingh Daniel
2. Srinivas Institute of Technology
3. 8th SEM, BE- Artificial Intelligence and Machine Learning

1. Nihal Athri
2. Srinivas Institute of Technology
3. 8th SEM, BE- Artificial Intelligence and Machine Learning

1. Adarsh S M
2. Srinivas Institute of Technology
3. 6th SEM, BE- Artificial Intelligence and Machine Learning

1. Swathi
2. Srinivas Institute of Technology
3. 6th SEM, BE- Artificial Intelligence and Machine Learning

Proposed Solution:

1. Code Summarization Entity:

- Develop a module that takes source code as input.
- Implement a parser to understand the structure and semantics of the source code.
- Extract key elements from the parsed code to build a summary.
- This could involve extracting comments, function/method names, or using natural language processing techniques.

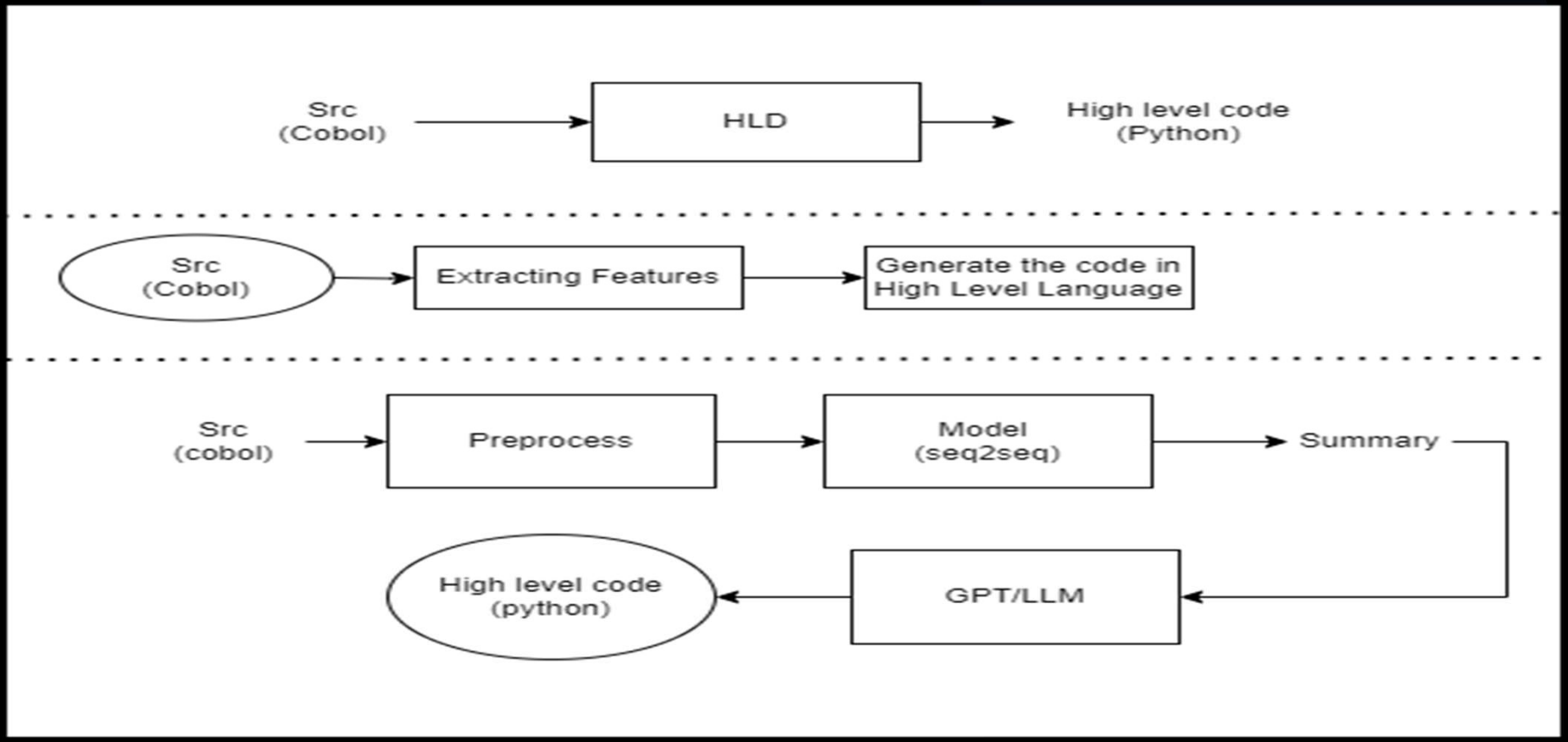
2. Code Generation Entity:

- Develop a second module that takes the summary from the first entity as input.
- Implement Natural Language Processing (NLP) and Machine Learning (ML) techniques to interpret the summary and map it to Python constructs.
- Generate Python code based on the interpreted summary.

3. Integration of Entities:

- Integrate the two entities to form a pipeline where the output of the first entity (summary of source code) becomes the input for the second entity (Python code).

Flowchart:


Src
(Cobol)
HLD
High level code
(Python)
Src
(Cobol)
Extracting Features
Generate the code in
High Level Language
Src
(cobol)
Preprocess
Model
(seq2seq)
Summary
High level code
(python)
GPT/LLM

Innovative feature

1. Comprehensive Python Utilization:

- Our solution leverages all core concepts and best practices of Python programming, ensuring the translated code is efficient, readable, and maintainable.

2. Advanced Code Summarization:

- Provides detailed summaries of COBOL code, including function breakdowns, data structure mappings, and control flow visualizations, to aid in understanding and translation.

3. Automated and Optimized Code Generation:

- Utilizes template-based methods to translate COBOL to Python accurately and consistently, while ensuring PEP 8 compliance and performance optimization.

4. Built-in Testing and Validation:

- Automatically generates unit tests for the translated Python code and validates the output against the original COBOL code, ensuring functional equivalence.

5. Interactive and User-Friendly Interface:

- Features an interactive code editor with real-time suggestions and visual feedback, facilitating a seamless and intuitive translation process.

Technologies Used

1. Python: The Python programming language will be used to implement the seq2seq model and handle the code migration process.

2. TensorFlow : These deep learning frameworks can be used to build and train the seq2seq model. This frameworks provide implementations of the necessary components for sequence-to-sequence modeling.

3. Github API: For collecting COBOL dataset.

4. UI Tools: React JS

5. Tools : Google Colab or Jupyter, VS Code, Git.

6. Backend/ API's: FastAPI

THANK YOU