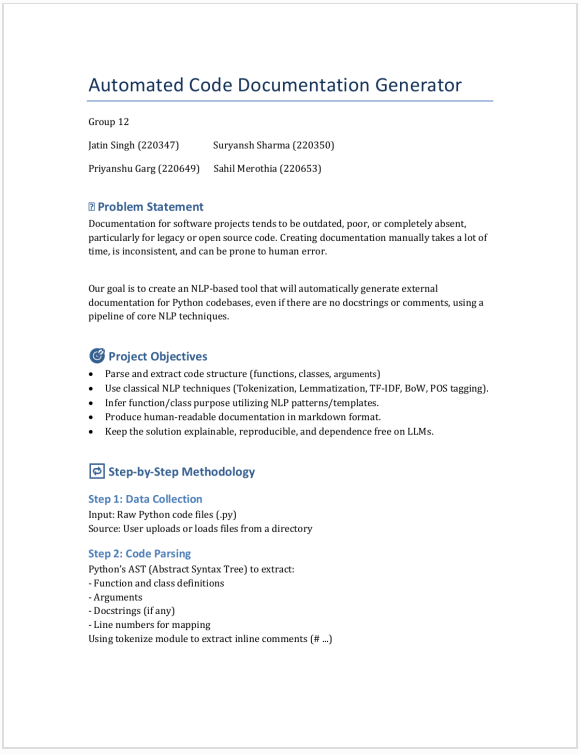
Digital Receipt

This receipt acknowledges that Turnitin received your paper. Below you will find the receipt information regarding your submission.

The first page of your submissions is displayed below.

Submission author:	Jatin Singh
Assignment title:	System and Architecture Design
Submission title:	NLP Project Methodology Group 12.docx
File name:	NLP_Project_Methodology_Group_12.docx
File size:	110.38K
Page count:	4
Word count:	473
Character count:	2,885
Submission date:	14-Apr-2025 10:21PM (UTC+0530)
Submission ID:	2645923241



Automated Code Documentation Generator

Group 12

Jatin Singh (220347) Suryansh Sharma (220350)

Priyanshu Garg (220649) Sahil Merothia (220653)

Problem Statement

Documentation for software projects tends to be outdated, poor, or completely absent, particularly for legacy or open source code. Creating documentation manually takes a lot of time, is inconsistent, and can be prone to human error.

Our goal is to create an NLP-based tool that will automatically generate external documentation for Python codebases, even if there are no docstrings or comments, using a pipeline of core NLP techniques.

Project Objectives

- Parse and extract code structure (functions, classes, arguments)
- Use classical NLP techniques (Tokenization, Lemmatization, TF-IDF, BoW, POS tagging).
- Infer function/class purpose utilizing NLP patterns/templates.
- Produce human-readable documentation in markdown format.
- Keep the solution explainable, reproducible, and dependence free on LLMs.

Step-by-Step Methodology

Step 1: Data Collection

Input: Raw Python code files (.py)
Source: User uploads or loads files from a directory

Step 2: Code Parsing

Python's AST (Abstract Syntax Tree) to extract:
- Function and class definitions
- Arguments
- Docstrings (if any)
- Line numbers for mapping
Using tokenize module to extract inline comments (# ...)