**Automated Test Case Generation from Requirements**

**Introduction:**

The project focuses on developing a tool that automatically generates test cases from software requirements written in natural language. The goal is to efficiently and accurately convert textual requirements into structured test cases.

**Approach:**

**1.Preprocessing:**

Data Cleaning: Text is cleaned by removing non-ascii characters and extra spaces.

**2.Categorization:**

Requirements are categorized based on prefixes (e.g., Functional, Non-Functional) for structured analysis.

**3.Text Parsing**

This step is crucial for accurately identifying the key components needed for test case generation. For each entity (Actor, Action, Condition, Outcome) some rules are defined, based on those rules entities are extracted.

**4.Test Case Generation:**

A rule-based approach along with some templates (for test steps) is used to generate test cases.

**5.Evaluation:**

The tool’s effectiveness is measured using Precision (1.00), Recall (0.67), F1 Score (0.80), and Accuracy (0.75).

**Workflow:**

**Dataset** (nrf.txt)

**|**

**Preprocessing**

**|** cleaned\_nrf.txt

**Categorization**

**|** requirements (default dict)

**Text Parsing**

**|** extracted\_entities.json

**Rules based Testcase Generation**

**|** testcases(list)

**Evaluation**