Project Title: Test Case Generator for Unit Testing

An AI-Powered Tool for Automated Unit Test Case Generation

Contact Info:

- Manan Kathrecha: kmanan@umich.edu
- Neeraj Saini: neerajsa@umich.edu
- Siddharth K Ram: sidkram@umich.edu

Project Specifications

Context

Unit testing is a critical part of software development, ensuring that individual components of a system work as expected. However, writing unit tests is time-consuming and often overlooked due to tight deadlines. Manual test case creation is prone to human error and may miss edge cases or boundary conditions. Automating this process can save time, improve code quality, and ensure better test coverage.

This project aims to develop a **Test Case Generator** that automatically generates unit test cases for functions based on their signatures and expected behavior. By integrating AI/ML, the tool will learn from existing codebases and improve its ability to generate meaningful and comprehensive test cases.

Motivation / Problem Statement

- **Manual Effort**: Writing unit tests manually is tedious and error prone.
- **Incomplete Coverage**: Developers often miss edge cases and boundary conditions.
- **Time Constraints**: Tight deadlines lead to insufficient testing.
- **Lack of Expertise**: Junior developers may struggle to write effective test cases.

The **Test Case Generator** addresses these challenges by automating test case creation, reducing manual effort, and improving test coverage.

Expected Contributions & Objectives

Solution

The project will deliver a tool that:

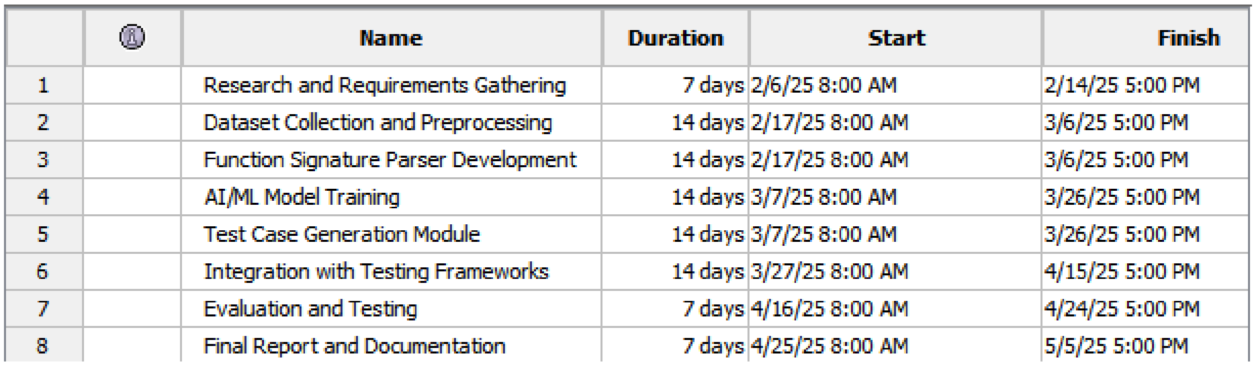
1. Parses function signatures to understand input parameters and return types.
2. Uses AI/ML to infer possible inputs, outputs, and edge cases.
3. Generates unit test cases in popular testing frameworks (e.g., pytest, JUnit).
4. Learn from existing codebases to improve test case generation over time.

Objectives

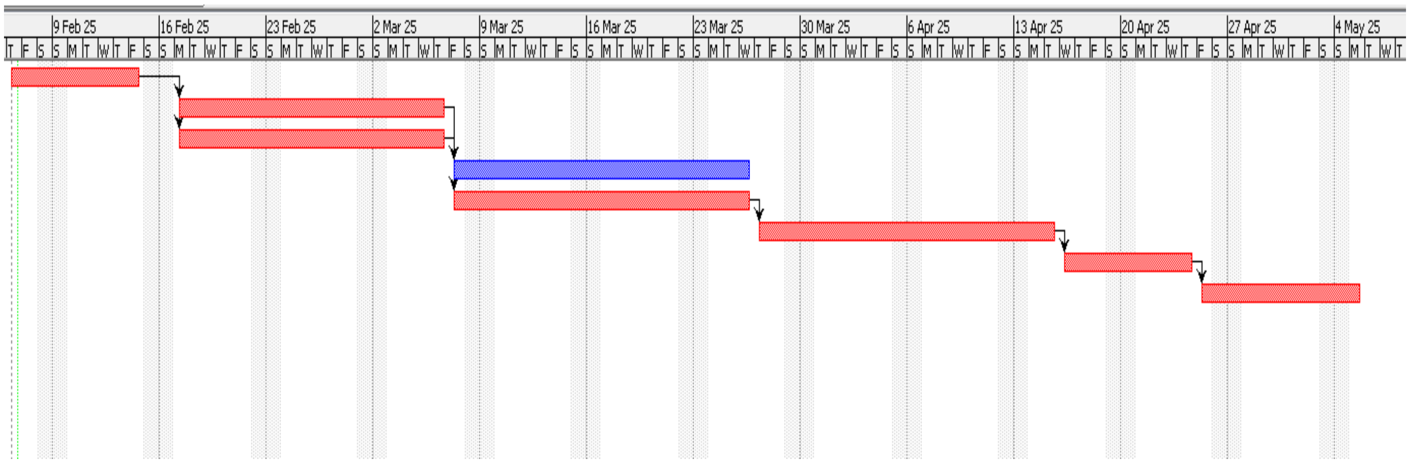
1. Develop a function signature parser for multiple programming languages.
2. Train an AI/ML model to predict test cases based on function behavior.
3. Generate test cases in a user-specified testing framework.
4. Provide a user-friendly interface for developers to use the tool.
5. Evaluate the tool's effectiveness on real-world codebases.

Task Management

Task Management - Test Case Generator for Unit Testing



		Name	Duration	Start	Finish
1		Research and Requirements Gathering	7 days	2/6/25 8:00 AM	2/14/25 5:00 PM
2		Dataset Collection and Preprocessing	14 days	2/17/25 8:00 AM	3/6/25 5:00 PM
3		Function Signature Parser Development	14 days	2/17/25 8:00 AM	3/6/25 5:00 PM
4		AI/ML Model Training	14 days	3/7/25 8:00 AM	3/26/25 5:00 PM
5		Test Case Generation Module	14 days	3/7/25 8:00 AM	3/26/25 5:00 PM
6		Integration with Testing Frameworks	14 days	3/27/25 8:00 AM	4/15/25 5:00 PM
7		Evaluation and Testing	7 days	4/16/25 8:00 AM	4/24/25 5:00 PM
8		Final Report and Documentation	7 days	4/25/25 8:00 AM	5/5/25 5:00 PM


9 Feb 25
16 Feb 25
23 Feb 25
2 Mar 25
9 Mar 25
16 Mar 25
23 Mar 25
30 Mar 25
6 Apr 25
13 Apr 25
20 Apr 25
27 Apr 25
4 May 25