Project Initialization and Planning Phase

Date	13 july 2024
Team ID	SWTID1720078627
Project Title	BugBuster-Automated-Code-Debugging-Tool-Using-Palm-s-Chat-Baison-001
Maximum Marks	3 Marks

Project Proposal (Proposed Solution) report

Problem: Finding and fixing bugs in computer programs is hard and time-consuming.

Solution: BugBuster is a tool that uses artificial intelligence to automatically find and fix bugs in computer code. It's like having a smart assistant that helps programmers work faster and make fewer mistakes.

How it works:

- Examines code to find problems.
- Explains what's wrong and how to fix it.
- Suggests code changes to solve the problem.

Benefits:

- Saves programmers time.
- Helps write better code.
- Makes software more reliable.

Project Overview	
Objective	To develop an AI-powered debugging tool, BugBuster, that significantly reduces debugging time and effort for developers by automatically identifying, analyzing, and suggesting solutions for code defects.
Scope	• Find mistakes in computer code. • Explain why the mistakes happened. • Suggest how to fix the mistakes. • Work with common types of code. • Be easy to use.
Problem Statement	
Description	BugBuster is an AI tool designed to automatically detect, analyze, and suggest fixes for errors in computer code, saving developers time and improving code quality.
Impact	BugBuster aims to significantly accelerate software development by automating the debugging process, leading to increased developer productivity and higher software quality.
Proposed Solution	
Approach	BugBuster will combine code analysis, machine learning, and natural language processing to automatically detect, analyze, and suggest fixes for code errors.
Key Features	BugBuster automates code analysis, detects and classifies bugs, suggests potential fixes, and provides clear explanations. It integrates seamlessly into developer workflows.

Resource Type	Description	Specification/Allocation
Hardware		
Computing Resources	CPU/GPU specifications, number of cores	• Multi-core processor • 4-8 cores might suffice • NVIDIA GPU highly recommended • VRAM Depends on the size of your model and datasets
Memory	RAM specifications	**Minimum 16GB:** For basic operations, but more is often required for complex tasks
Storage	Disk space for data, models, and logs	1 TB SSD
Software		
Frameworks	Python frameworks	streamlit
Libraries	Additional libraries	google.generetiveai
Development Environment	IDE	Jupyter Notebook, visual studio code