**INTEL UNNATI INTERNSHIP – 2025**

**Problem Title:** Bug Detection and Fixing

## Name: Subashini J

## College: Sri Sairam Engineering College

## Internship Organization Name: Intel (Unnati Industrial Training)

## Internship Duration: 36 days (01.03.2025 to 05.04.2024)

## Mentor: Dr.P.Vijayakumari

## Date of Submission: 04.04.2025 ACKNOWLEDGMENT

I sincerely express my gratitude to Intel Unnati for providing me with the opportunity to work on the Bug Detection and Fixing project as part of their internship program. This experience has been invaluable in enhancing my technical skills and problem-solving abilities in software debugging and automation.

I extend my heartfelt thanks to my mentors and project guides for their continuous support, insightful feedback, and encouragement throughout the project. Their expertise and guidance played a crucial role in refining my approach and improving the quality of my work. I also acknowledge the contributions of my team members, whose collaboration, dedication, and innovative thinking were instrumental in successfully executing the project.

Lastly, I appreciate the vast learning resources and technical support provided by Intel Unnati, which enabled me to explore cutting-edge methodologies in bug detection and resolution, thereby enriching my understanding of Python debugging and software reliability.

Thank you to Intel and my mentors.

**PROBLEM STATEMENT:**

A system which aims to enhance software code quality, reduce debugging time and improve developer productivity by identifying bugs in software code and provide recommended fixes.

**AIM:**

The aim of this project is to develop an interactive interface that allows users to detect errors in their Python code and receive recommended fixes along with relevant examples.

**DESCRIPTION:**

The Python Bug Detector is a web-based application designed to help developers identify and fix errors in their Python code efficiently. The tool provides syntax error detection, runtime error analysis and common logical error identification, making it a valuable resource for both beginners and experienced programmers. With features like line highlighting for errors and a light/dark theme toggle, the interface ensures a seamless debugging experience.

Built using Flask for the backend, JavaScript for interactivity and Code Mirror for an intuitive code editing experience, the application allows users to upload Python files or paste code directly into the editor. Once analyzed, the system highlights errors and offers contextual recommendations for fixes, aiding in faster debugging and improved code quality.

The project follows the MIT License, enabling open-source contribution and free usage, allowing developers to modify and enhance the tool as needed.

**MY CONTRIBUTION:**As part of the Python Bug Detection and Fixing project under the Intel Unnati Internship, I, Subashini J, played a key role in front-end development by designing and implementing CSS styling for the project’s website. My primary objective was to develop a responsive and user-friendly interface that enhances visual appeal, accessibility, and usability. Additionally, I integrated a theme-switching feature to improve user experience and leveraged AI-powered tools to optimize the CSS code, ensuring efficiency, consistency, and maintainability.

**Key Contributions:**

* **Theming Support:** Implemented dark/light mode using CSS variables for seamless transitions.
* **Global Styling:** Applied reset styles for uniform spacing, typography, and layout.
* **Navigation Bar:** Designed a responsive navigation bar with shadow effects.
* **Code Editor UI:** Styled the embedded CodeMirror editor for readability.
* **File Upload Section:** Created a drag-and-drop upload feature for user convenience.
* **Responsive Design:** Ensured mobile compatibility using media queries and flexbox/grid.
* **Icon Integration:** Used Font Awesome icons for improved UI aesthetics.
* **AI-Enhanced Optimization:** Utilized AI tools to refine CSS, reducing redundancy and improving performance.

**Performance & Optimization:**

* Used CSS variables for theme management and maintainability.
* Implemented smooth transitions and structured CSS for readability.
* Minified and optimized CSS to reduce load times.
* Leveraged AI-generated recommendations for better performance.

**Challenges & Solutions:**

* **Cross-Browser Compatibility:** Conducted testing for consistent rendering.
* **Dark Mode Performance:** Used transitions and variables for smooth switching.
* **Load Time Optimization:** Minified CSS to enhance page speed.
* **Scalability:** Structured CSS efficiently for future expansion.

