CodeSentinel AI

Automated Code Review Assistant

Intelligent code analysis, security det
ection, and performance optimization built
with BLACKBOX.AI, Groq API, and Llama Models

Team: QuantX - Blackbox Track

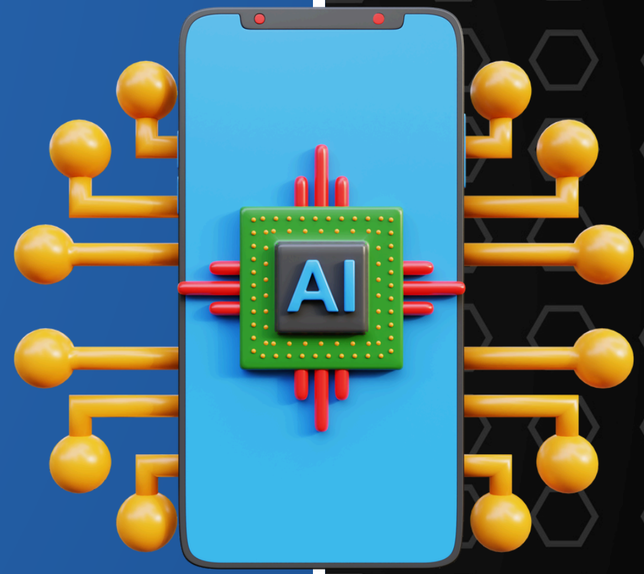
Problem

Code Reviews Take Too Much Time

Current Problems:

- Developers spend hours checking each other's code
- Bugs and security holes get missed by human eyes
- Junior developers wait days for feedback
- Projects get delayed because of slow reviews

The Result: Wasted time, more bugs, frustrated teams



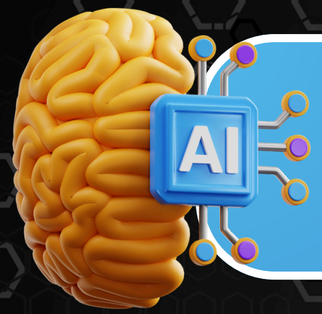
Our Solution

Let AI Do the Heavy Lifting

What our tool does:

- Reads your code in seconds (not hours)
- Finds bugs and security problems instantly
- Gives helpful suggestions to make the code better
- Works the same way every time
- Gives feedback immediately

Think of it as: A super-smart coding teacher that never gets tired



Core Features

01

Smart Code Checking

- Scores your code quality
- Tells you what's good and what needs work
- Checks if you follow good coding habits

02

Security Guard

- Finds places hackers could break in
- Spot weak passwords
- Warns about unsafe data handling

03

Speed Booster

- Shows where your code runs slowly
- Suggests ways to make it faster
- Save computer memory

Frontend UI

Hugging Face Deployment: https://huggingface.co/spaces/isana25/AI-Code-Assistant

🧠 AI Code Assistant

Powered by BLACKBOX.AI, Groq API, and Llama Models

Features:

- 🤖 **Automated Code Review**: Comprehensive quality assessment and review comments
- 🛡️ **Security Analysis**: Vulnerability detection and security recommendations
- 📊 **Performance Optimization**: Intelligent suggestions for better performance
- 📝 **Review Comments**: Generate detailed review comments for team collaboration

📝 Enter Your Code

```
function updateUserList(users) {
    const container =
document.getElementById('user-list');
    container.innerHTML = ''; // Clearing DOM

    for (let i = 0; i < users.length; i++) {
        const div =
document.createElement('div');
        div.innerHTML = `
            <h3>${users[i].name}</h3>
            <p>${users[i].email}</p>
            <button
onclick="deleteUser(${users[i].id})">Delete</bu
tton>
        `;
        container.appendChild(div); // Multiple
DOM operations
    }
}

// Memory leak potential
let globalCache = {};
```

🔍 Analysis Type

Automated code review with comprehensive analysis

- Automated Review

🚀 Analyze Code

📊 Quick Summary

Review Rating: Needs Significant Improvement

📊 Analysis Results

🤖 Groq + Llama Analysis | 📋 Comprehensive Report

CODE QUALITY ASSESSMENT

- Overall code quality score: 4/10
- Readability and maintainability: The code is somewhat readable, but the lack of proper indentation and inconsistent spacing make it difficult to follow. The code structure is not well-organized, with functions performing multiple unrelated tasks.
- Code structure and organization: The code is not well-organized, with functions performing multiple unrelated tasks. The `updateUserList` function is responsible for both updating the DOM and handling user data, while the `cacheUserData` function is responsible for storing user data without any cleanup mechanism.

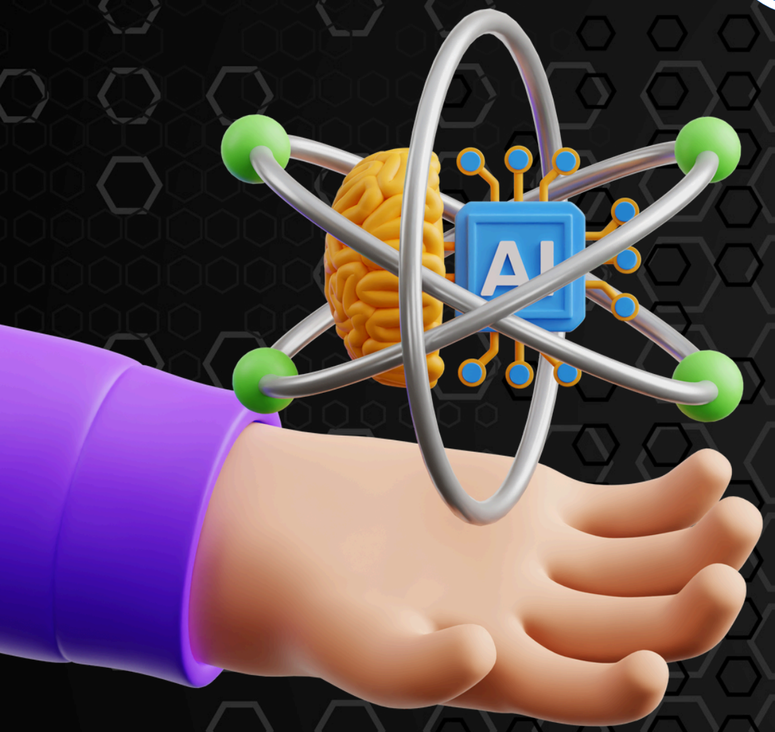
STYLE & BEST PRACTICES

- Coding style compliance: The code does not follow a consistent coding style, with inconsistent indentation and spacing.
- Naming conventions: The variable names are descriptive, but could be improved by following a consistent naming convention (e.g., camelCase or PascalCase).
- Best practices adherence: The code does not follow best practices for error handling, caching, or DOM manipulation.
- Documentation quality: There is no documentation provided for the functions or variables.

BUG DETECTION

- Syntax errors: None detected
- Logic errors: The `cacheUserData` function does not have a mechanism to clean up cached data, which

Backend & AI Technologies



Groq API

This API enables rapid processing of code input, ensuring that developers receive analysis results promptly.

BLACKBOX.AI

This integration ensures that the AI provides informed suggestions based on a wealth of programming knowledge.

Llama 3-8B-8192 Model

This high-performance model enables precise detection of bugs and adherence to coding best practices.

Works With Any Programming Language

No Matter What Code You Write

Popular Languages We Support:

- Python
- JavaScript
- C/C++
- Java
- And 16 more languages!

Why This Matters

Most tools only work with 1-2 languages Our tool works with almost everything

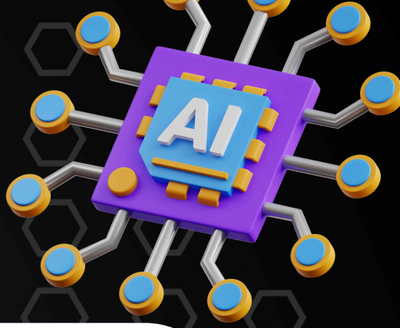
What Makes our Solution Unique

Better Technology

- Get results in under 10 seconds
- Check security, speed, and quality all at once
- Easy to understand results

Better Value

- Costs 10x less than big company tools
- Set up in 5 minutes (not weeks)
- Works with tools teams already use
- Always getting smarter with new AI updates



Business Value

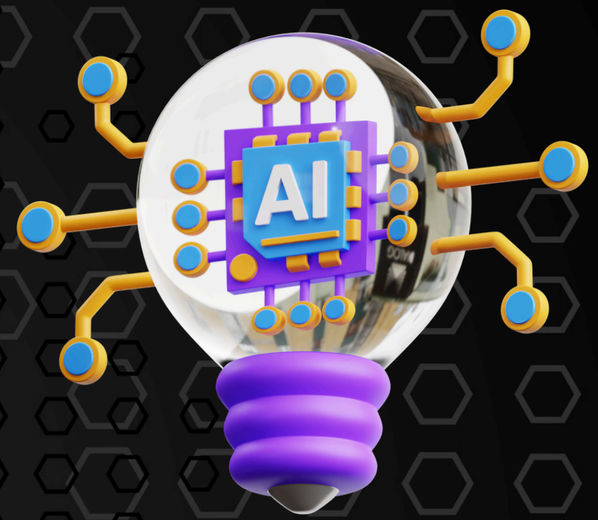
For Development Teams

- **Reduce Review Time:** Automate the initial code review process
- **Improve Code Quality:** Consistent quality standards enforcement
- **Knowledge Transfer:** Educational feedback for junior developers
- **Security Enhancement:** Early detection of security vulnerabilities

For Organizations

- **Cost Reduction:** Decrease manual review overhead
- **Risk Mitigation:** Proactive security vulnerability detection
- **Productivity Boost:** Faster development cycles
- **Quality Assurance:** Standardized code quality across projects

Future Enhancements



01 **Multi-file Analysis:** Support for entire project analysis

02 **Custom Rule Engine:** Organization-specific coding standards

03 **Integration APIs:** CI/CD pipeline integration

04 **Historical Tracking:** Code quality trends over time

Recognition of Key Contribution

01

BLACKBOX.AI

Special thanks to BLACKBOX.AI for providing advanced coding assistance capabilities, which enrich CodeSentinel AI's overall functionality and effectiveness.

02

Groq

Gratitude is extended to Groq for its fast inference infrastructure, which is critical for maintaining the responsiveness of the CodeSentinel AI application.

03

Meta

Recognition is also due to Meta for developing the innovative Llama 3 model, which serves as a cornerstone for CodeSentinel AI's code analysis capabilities.

04

Hugging Face

Hugging Face is acknowledged for hosting the deployment platform that provides users with access to CodeSentinel AI and its range of features.

05

Gradio

Lastly, appreciation goes to Gradio for providing the web interface framework that enhances user experience and simplifies interaction with CodeSentinel AI. Built with ❤ using AI-powered technologies

Team
QuantX

Thank You!

https://github.com/isana25/CodeSentinel