

# **AI-Powered Code Reviewer: A MERN Stack-Based Intelligent Code Review Application**

*A thesis submitted in partial fulfillment of the requirements for the award of the  
degree of*

**Master of Computer Applications**

by

**Amit Gupta**

**23419MCA065**



**Department of Computer Science**

**Institute of Science**

**Banaras Hindu University, Varanasi – 221005**

**MAY-2025**

## BONAFIDE CERTIFICATE

This is to certify that the project report entitled "*AI-Powered Code Reviewer: A MERN Stack-Based Intelligent Code Review Application*", submitted by *Amit Gupta (23419MCA065)* to the Department of Computer Science, Institute of Science, Banaras Hindu University, Varanasi – 221005, is a bonafide record of the work carried out under the academic supervision of *Dr. Vandana Kushwaha* during the period from *January 2025* to *March 2025*.

Signature of the Supervisor  
**Dr. Vandana Kushwaha**  
*Assistant Professor*  
*Department of Computer Science*  
*Banaras Hindu University*

Signature of the Coordinator  
Dr. Manoj Kumar Singh  
Professor  
*Department of Computer Science,*  
Banaras Hindu University

Place: Varanasi  
Date: 23 May 2025

## CANDIDATE'S DECLARATION

I Amit Gupta, hereby certify that the work, which is being presented in the project report, entitled **AI-Powered Code Reviewer: A MERN Stack-Based Intelligent Code Review Application**, in partial fulfillment of the requirements for the award of the degree of **Master of Computer Applications** and submitted to the institution, is an authentic record of my own work carried out during my internship at **Smart Swift Innovation** from January 2025 to March 2025 under the supervision of **Mr. Pawan Kumar**, and with academic guidance from **Dr. Vandana Kushwaha**. I also cited the reference about the text(s) /figure(s) /table(s) /equation(s) from where they have been taken.

The matter presented in this report has not been submitted elsewhere for the award of any other degree or diploma from any Institution.

Date:29-05-2025

Signature of the Candidate

This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge.

Date:29-05-2025

Signature of the Supervisor

The Viva-Voce examination of Amit Gupta, M.C.A. Student has been held on 29-05-2025.

Signature of The External Examiner  
of the Department

Signature of Head

## ABSTRACT

This major-project report presents the development of an intelligent web application titled **AI-Powered Code Reviewer**, designed to automatically analyze and provide feedback on code snippets using artificial intelligence. The application is built using the MERN stack (MongoDB, Express.js, React.js, Node.js) and integrates Google's AI model (Gemini 2.0 Flash) to enhance code review capabilities. The system enables developers to receive AI-driven code insights, including bug detection, code optimization suggestions, and best practice recommendations.

One of the core objectives of the project was to bridge the gap between manual code review practices and modern AI-driven solutions. Manual reviews often suffer from inconsistency, subjectivity, and delay, particularly when development teams scale. By introducing automation, this system significantly accelerates the review cycle while maintaining accuracy and consistency.

The application architecture follows a modular and scalable approach, allowing seamless interaction between frontend and backend components, and facilitating the integration of third-party AI services. The AI model is prompted with instructions to mimic an expert reviewer and provide structured, human-readable outputs. The system was tested with various code inputs across multiple scenarios to validate its reliability and performance.

Overall, this project showcases how artificial intelligence can be leveraged in real-world software engineering workflows. It not only aids in maintaining code quality but also supports developers in learning and improving through continuous feedback.

**Keywords:** MERN Stack, AI Integration, Google Gemini, Code Analysis, Full-stack Development, Code Review, AI-Powered Applications

## ACKNOWLEDGEMENT

I express my sincere gratitude to my industry mentor **Mr. Pawan Kumar**, at **Smart Swift Innovation**, for providing me the opportunity to work on this exciting project and for his invaluable guidance, support, and encouragement throughout the internship.

I am equally thankful to my academic supervisor **Dr. Vandana Kushwaha**, for her continuous support and insightful suggestions during the course of this project.

I also extend my heartfelt thanks to the team members and staff at **Smart Swift Innovation**, as well as the faculty of the **Department of Computer Science, BHU**, for their continuous encouragement and help during this internship journey.

Last but not least, I am grateful to my parents and friends for their unwavering encouragement, patience, and belief in me throughout the internship period. Their constant support provided me the strength and motivation to accomplish this project successfully.

This report is a culmination of guidance, learning, and experiences gained during my internship, and I thank everyone who contributed to its successful completion

## Table of Contents

Title	Page No
BONAFIDE CERTIFICATE.....	2
CANDIDATE'S DECLARATION.....	3
ABSTRACT.....	4
ACKNOWLEDGEMENT.....	5
LIST OF FIGURES.....	7
LIST OF ABBREVIATIONS.....	8
1. Introduction.....	9
2. Problem Statement.....	10
3. Scope.....	11-12
4. Objectives.....	13
5. Literature Review.....	14
6. Tools and Technologies.....	15
7. System Architecture.....	16
8. Project Design and Development.....	17
9. Backend Integration.....	18
10. AI Service Integration.....	19
11. Frontend Development.....	20
12. Data Flow and API Design.....	21
13. Testing and Debugging.....	22
14. Results and Screenshots.....	23
15. Challenges Faced.....	24
16. Internship Experience.....	25
17. Skills Acquired During Internship.....	26
18. Conclusion.....	27
19. Future Scope.....	28
20. References.....	29
21. Appendix.....	30

## List of Figures

<b>Figure Number</b>	<b>Title</b>	<b>Page Number</b>
<b>Figure 1</b>	<b>System Architecture Diagram (Client-Server Flow)</b>	<b>23</b>
<b>Figure 2</b>	<b>Code Editor Interface Screenshot</b>	<b>23</b>
<b>Figure 3</b>	<b>AI Feedback Output Example</b>	<b>23</b>
<b>Figure 4</b>	<b>Postman API Test Results</b>	<b>23</b>

## List of Abbreviations

Abbreviation	Full Form	Description
<b>AI</b>	Artificial Intelligence	Technology enabling machines to mimic human-like code analysis
<b>API</b>	Application Programming Interface	Protocol for backend-AI communication
<b>BHU</b>	Banaras Hindu University	Institution where research was conducted
<b>CORS</b>	Cross-Origin Resource Sharing	Middleware for secure frontend-backend connectivity
<b>JSON</b>	JavaScript Object Notation	Data interchange format for APIs
<b>LLM</b>	Large Language Model	AI model type (e.g., Gemini 2.0 Flash)
<b>MERN</b>	MongoDB, Express.js, React.js, Node.js	Full-stack framework used in the project
<b>NoSQL</b>	Non-relational Database	Database type (MongoDB) for scalability
<b>UI</b>	User Interface	React.js-based frontend components
<b>REST</b>	Representational State Transfer	Architectural style for backend APIs



## **1. Introduction**

Software development is increasingly complex and fast-paced. Developers often need rapid feedback on their code, not only for syntax correctness but also for quality assurance and maintainability. Manual code reviews are time-consuming, and automation can play a key role in scaling this process.

The “AI-Powered Code Reviewer” application is built to address this challenge. It acts as a smart assistant that can analyze, interpret, and offer improvements on source code submitted by developers. This tool leverages recent advances in AI to transform how code reviews are conducted.

By combining artificial intelligence with modern web development, the project serves as an innovative example of how technology can support developers in producing higher quality software efficiently.

## **2. Problem Statement**

Manual code reviews, while effective, have inherent limitations such as bias, fatigue, and inconsistency. As teams scale, maintaining code quality becomes increasingly difficult. Additionally, beginners and solo developers lack access to senior-level feedback.

### **Challenges:**

- Lack of consistent and timely code feedback.
- Difficulty in identifying bugs and poor practices early.
- Resource-intensive manual review processes.
- Inaccessibility to expert-level suggestions for all developers.

The goal of this project is to build a tool that addresses these challenges using AI models to review and analyze code in a structured and intelligent manner.

### **3. Scope**

The AI-Powered Code Reviewer project encompasses the following boundaries and capabilities:

#### **Included Functionalities**

##### **1.Code Analysis**

- Supports JavaScript and Python (with extensibility to other languages)
- Detects syntax errors, logical bugs, and anti-patterns

##### **2. AI-Powered Feedback**

- Provides optimization suggestions via Gemini 2.0 Flash
- Explains best practices in natural language

##### **3.Technical Scope**

- MERN stack implementation (React.js frontend + Node.js/Express.js backend)
- Google Gemini API integration for AI processing
- Stateless architecture (no mandatory database dependency)

#### **Exclusions**

##### **1.Limitations**

- No support for compiled languages (e.g., C++, Java) in current version
- Cannot execute or debug code (static analysis only)

## **2.Non-Functional Boundaries**

- Deployment and CI/CD pipelines are out of scope
- Advanced user authentication (e.g., OAuth) not implemented

### **Target Users**

- Individual developers seeking quick code reviews
- EduTech platforms for programming training
- Small teams needing lightweight review automation

## 4. Objectives

The core objectives of this project include:

- Develop a MERN stack-based web application to submit and review code.
- Integrate Google's Gemini AI model for natural language code analysis.
- Provide feedback on syntax errors, optimization, and coding standards.
- Display suggestions in a structured and user-friendly format.
- Enable users to interact with the system in real-time.

## 5. Literature Review

The application of AI in code analysis is not new. Tools like DeepCode, CodeGuru (Amazon), and GitHub Copilot have pioneered this domain. These tools use machine learning and large language models to detect bugs, suggest improvements, and even generate code.

Our approach is distinct in integrating open-access AI models (Gemini) within a customizable MERN framework. The literature supports the idea that LLMs (Large Language Models) can be trained or prompted to act as software engineering assistants, capable of understanding context and semantics within code snippets.

## 6. Tools and Technologies

### 5.1 Frontend

**React.js:** JavaScript library for building user interfaces.

**Prism.js:** Syntax highlighting.

**React Simple Code Editor:** In-browser code editing.

### 5.2 Backend

**Node.js:** JavaScript runtime for server-side logic.

**Express.js:** Web framework for building REST APIs.

### 5.3 AI Integration

**Google Gemini 2.0 Flash API:** Cloud-based AI model.

### 5.4 Others

**Postman:** API testing and debugging.

**MongoDB (optional):** NoSQL database (scalability focus).

**dotenv & cors:** Middleware utilities.

## 7. System Architecture

The application follows a “client-server architecture”:

The client (frontend) allows code input and displays feedback.

The server (backend) handles requests and interacts with the AI model.

The AI service processes the code and returns a structured response.

The flow is as follows:

User Input → Frontend → Backend API → Gemini AI → Backend →  
Frontend Output



## **8. Project Design and Development**

The application was divided into the following components:

- Code Editor UI
- Review Button
- Result Display
- API Request Handler
- AI Service Module

These components were developed separately and then integrated.

Emphasis was placed on modularity and reusability of code.

## 9. Backend Integration

The backend server was created using Express.js and Node.js. The main tasks included:

- Creating endpoints for /get-review
- Connecting the AI module with Google Gemini API
- Handling errors and timeouts
- Parsing JSON and managing environmental variables securely

javascript

```
app.post("/api/get-review", async (req, res) => {  
  const { code } = req.body;  
  const result = await aiReview(code);  
  res.send(result);  
});
```

## 10. AI Service Integration

Using the Gemini API required secure API key handling and configuring requests to include prompt-based system instructions:

### **System Instructions Example:**

"You are a code reviewer with expertise in software development. Analyze the code, point out bugs, suggest improvements, and offer best practices."

The AI then returns structured content such as:

- Detected Issues
- Recommended Fixes
- Optimized Version
- Explanation

## 11. Frontend Development

The frontend used React.js for dynamic interaction:

- Code input field with syntax highlighting
- Submit button
- Display panel for AI feedback

Markdown rendering was achieved with react-markdown to show formatted text including:

- Bullet points
- Code blocks
- Bold/italic annotations

## 12. Data Flow and API Design

The key elements of data flow are:

1. Code entered by the user.
2. Code sent via POST request to backend.
3. Backend calls Gemini API and receives structured review.
4. Backend sends result back to frontend.
5. Frontend renders review content using Markdown.

The API design ensures:

- Separation of concerns
- Secure token management
- Scalable request handling

## 13. Testing and Debugging

The following tools and strategies were used:

Postman: To test backend API independently.

Console Logging: For backend tracking.

React Developer Tools: To debug component state.

Try-catch blocks: For managing async/await behavior and API failures.

Sample Test Case:

Input:

```
javascript
function sum(a, b) {
  return a + b;
}
```

Output:

- Missing input validation
- Suggestion to check types
- Alternative: `typeof a === 'number' && typeof b === 'number'`

## 14. Results and Screenshots

Screenshots to include:

- Code Editor Interface
- AI Feedback Section
- Postman Test Results
- Console Logs showing successful API interaction

```
1 function sum(a,b) {  
2   return a = b  
3 }
```


Review

### AI Feedback

- Detected Issues:
  - No absence of input validation
- Recommended Fixes
  - Check the types of parameters
- Optimized Version
  - if "a=="b == the API call succeeded."

### Postman Test Results

200 OK 85 ms Response 23.44 K

Status: 200 OK  application/json;  
{  
 "result": "The API call succeeded."  
}

### Console Logs showing successful API interaction

info 16:47.23.864 API received request.

## **15. Challenges Faced**

- API latency with long code inputs
- Formatting issues in Markdown output
- Proper prompting for consistent AI results
- Frontend markdown rendering for multiline content
- Rate limiting with free API usage



## **16. Internship Experience**

During my internship at Smart Swift Innovation, I had the opportunity to work on a real-world project involving advanced web development and artificial intelligence. This experience allowed me to understand the practical aspects of full-stack development, API design, and AI integration. I was actively involved in developing the frontend interface using React.js, setting up backend APIs with Node.js and Express.js, and integrating Google's Gemini AI for intelligent code review. The work environment promoted collaborative learning and problem-solving, which greatly enhanced my development skills and professional growth.

## **17. Skills Acquired During Internship**

Throughout the internship, I acquired a variety of technical and soft skills that are crucial for a professional developer. These include:

- Proficiency in MERN stack (MongoDB, Express.js, React.js, Node.js)
  - API development and integration
  - Understanding of AI services and prompt engineering using Google Gemini API
  - Debugging and performance optimization techniques
  - Version control using Git and GitHub
  - Time management and team collaboration in a fast-paced environment
- These skills have significantly improved my ability to contribute to complex software development projects and prepared me for future roles in the tech industry.

## **18. Conclusion**

The AI-Powered Code Reviewer successfully demonstrates the integration of full-stack development with artificial intelligence to solve a real-world problem in software engineering. It improves productivity, ensures consistency, and democratizes access to expert-level feedback. The modular approach makes it scalable and adaptable for future improvements.

## **19. Future Scope**

- Deploy to cloud with real user login
- Add support for multiple programming languages
- Implement AI-driven code generation
- Add database integration for history tracking
- Create a mobile-friendly version

## 20. References

1. [<https://reactjs.org>](<https://reactjs.org>)
2. [<https://expressjs.com>](<https://expressjs.com>)
3. [<https://ai.google.dev>](<https://ai.google.dev>)
4. [<https://www.postman.com>](<https://www.postman.com>)
5. [<https://prismjs.com>](<https://prismjs.com>)
6. [<https://www.npmjs.com/package/react-markdown>](<https://www.npmjs.com/package/react-markdown>)

## 21. Appendix



# CERTIFICATE

OF INTERNSHIP COMPLETION

Proudly Presented To

**AMIT GUPTA**

has successfully completed the Internship

from **01-Jan-2025** to **30-Mar-2025** in **Web Development** at YHills.



Scan to verify



Powered by Wipro DICE ID



**Aman Kumar**  
Founder & CEO



**Pawan Kr. Rishu**  
Founder & COO