

# Final Report:

## Evaluation of Various Software Development AI Assistants

**Luna Sugiyama**

Graduate School of Information Science and Technology  
Information and Communication Engineering  
lunagracesugiyama@g.ecc.u-tokyo.ac.jp

July 31, 2025

I have recently been interested in Claude Code, and Gemini CLI which are AI assistant tools to help software development [1]. However, due to the pricing I have not been able to use them extensively. I thought this would be a good opportunity to explore these tools. In this report I will choose these services and evaluate them on various tasks and aspects to see their strengths and weaknesses.

### 1. About Each Service

#### 1.1. Claude Code

Claude Code is a code-centric variant of the Claude family of AI models developed by Anthropic. It is optimized for software development and programming-related tasks, such as code generation, debugging, explanation, and refactoring. Built on Anthropic's principles of Constitutional AI, Claude Code emphasizes helpfulness, honesty, and harmlessness while engaging in code-focused dialogues. Claude Code can understand complex software engineering contexts and generate or edit code in a variety of programming languages including Python, JavaScript, TypeScript, and more. It supports in-depth reasoning about existing codebases and integrates well into workflows via the Claude API or the Claude web interface. Key features of Claude Code include:

- **Code Generation:** Generates new code snippets based on natural language
- **Debugging:** Identifies and fixes bugs in existing code
- **Code Explanation:** Provides explanations for code snippets, including complex algorithms
- **Refactoring:** Suggests improvements to existing code for better performance or readability
- **Multi-language Support:** Works with multiple programming languages
- **Contextual Understanding:** Maintains context across multiple interactions

##### 1.1.1 Environment Setup

...

### 2. Tasks

#### 2.1. Task 1: Code Generation

The task for code generation is:

Build a responsive map of a shop with several floors and can search which product is where.



Table 1: Comparison of Claude Code and Gemini CLI for Code Generation Task

Criteria	Claude Code	Gemini CLI
Command Caution	Did not ask about permission to run commands	Asked whether it could run 'npx', 'mkdir', etc.
Response Speed	Fast (under 5 min)	Slow (about 15 minutes)
File Generation	Automatically generated 'index.html', 'style.css', 'app.js', 'data.js', and 'README.md'	Generated initial React project but required 'App.js' and JSON correction
Search Logic	Used simple string matching over all floor data	Similar string matching, but only applied to the current floor
Execution Outcome	Simple working web app with working search feature	React page; minimum page with no map-like component
Error Presence	No errors in generated code	JSON error and missing configuration in app setup
Overall Score	High	Moderate (required debugging)

### 2.1.3 Comparison

### 2.2. Task 2: Debugging

### 2.3. Task 3: Code Explanation

## 3. Evaluation Summary

Describe your approach, data collection methods, assumptions, tools used, etc.

## 4. Results / Findings

Present key results, findings, or analysis. You can include figures, tables, and equations where appropriate.

## 5. Discussion

Interpret your results, compare with existing work if relevant, and explain their implications.

## 6. Conclusion

Summarize the main points, state limitations, and propose future work or recommendations.

## References

[1] Kumar Madhukar. A comprehensive guide to vibe coding tools, 2025-03. Accessed: 2025-07-24.