Generative AI Tools as Enablers for Applications and Website Development Among Non-Programmers: A Practical Study

Akriti Bhanot
Researcher, New Delhi, India

M.S in Business Analytics and Information Systems, University of South Florida

Email: akritib2111@gmail.com

Abstract

In the past, creating applications and websites usually required programming knowledge. Because of this, many people without technical background found it difficult to build their own digital solutions. Without coding skills, turning an idea into a working application can be challenging.

Recent developments in Generative Artificial Intelligence (GenAI) have made it possible for users to create applications and websites using simple natural language instructions instead of writing code from scratch. In this paper we examine how Generative AI tools can help non-programmers create simple applications and websites with very little technical knowledge.

With the help of Generative AI users can generate working code, help fix errors, and provide explanations that make programming easier to understand. During the experiments, simple applications and websites were created using AI-assisted coding tools, and most of the generated code worked with only small modifications.

The paper presents practical examples using Python programs and simple website development to demonstrate how Generative AI tools support non-programmers in building functional applications and websites. The results suggest that Generative AI makes software development easier and more accessible for people without programming experience.

Keywords: Generative AI, Non-Programmers, Web Development, Application Development, Artificial Intelligence.

1. Introduction

Software applications and websites play an essential role in modern society. Businesses, educational institutions, and individuals rely on digital platforms for communication and information sharing. Traditionally, the development of applications and websites has required programming skills and knowledge of multiple technologies such as HTML, CSS, JavaScript, and backend programming languages.

For individuals without technical backgrounds, learning programming can be a difficult and time-consuming process. Many non-programmers have innovative ideas for applications and websites but they

lack the technical knowledge required to implement them. As a result, they must depend on professional developers, which increases development costs and delays project implementation.

Recent advances in Artificial Intelligence have introduced new possibilities for simplifying software development. Generative Artificial Intelligence tools are capable of generating text, images, and programming code based on user instructions. These tools allow users to describe their requirements in natural language and receive fully functional code as output.

Generative AI tools can generate website layouts, application logic, and database structures. In addition, these tools can explain programming concepts and assist users in correcting errors. This makes software development more accessible to individuals without formal programming training.

During the preparation of this study, several Generative AI tools were explored to understand how easily non-programmers could build working applications and websites.

The experiments described in this paper were carried out in a typical development environment without using advanced programming tools.

This study investigates how Generative AI tools enable non-programmers to build applications and websites. The focus of this research is on practical demonstrations of Generative AI-assisted development using simple examples.

---

2. Literature Review

Artificial Intelligence has been widely studied in the context of automation and decision-making systems. Machine learning and deep learning techniques have been applied in areas such as healthcare, finance, and education. Recent developments in Generative Artificial Intelligence have expanded the capabilities of AI systems beyond traditional prediction tasks.

Large language models are now able to generate programming code just by reading plain language descriptions. Researchers are testing out AI-powered programming assistants that help developers write and debug code, offering real-time suggestions that make the development process smoother and more efficient.

Newer research is showing how Generative AI tools can make learning programming easier. Studies find that AI coding assistants are especially helpful for beginners—they make tough concepts simpler and help users finish programming tasks faster. These tools let people experiment with code and learn by trying out real examples.

People are using generative AI tools more and more in web development these days. With AI-generated templates and scripts, you don’t need to be a coding expert to build a working website. It just makes the whole process less complicated than old-school software development.

Still, not a lot of research looks closely at how these AI tools actually help people who don't know how to code create apps or websites. That's the gap this study is trying to fill, by showing real examples of AI-assisted development in action.

---

3. Methodology

This study adopts a practical approach to evaluating the role of Generative AI in enabling non-programmers to build applications and websites. The study focuses on demonstrating how Generative AI tools can generate code based on natural language instructions.

Several development tasks were performed using Generative AI tools. These tasks included:

- Creating a simple Recipe website
- Developing a Weather Update application
- Debugging errors

The Generative AI system was provided with prompts describing the desired functionality. The generated code was then tested and executed to verify correctness.

In this study, the code generation process was performed by entering simple natural language instructions and observing the generated output. It was observed that the tools were able to produce usable code within a few seconds in most cases.

Only minor adjustments were required before the generated programs could be executed successfully.

The experiments were conducted on a standard personal computer using commonly available software tools.

The study involved three practical experiments:

Example 1 – Website Development Using Cursor AI

In this experiment, the AI-assisted development environment **Cursor AI** was used to generate a functional recipe website based on natural language instructions. The objective of this experiment was to evaluate whether a non-programmer could develop a structured website using Generative AI tools.

Prompt Provided:

The following prompt was provided in Cursor AI:

Create a food recipe website using HTML and CSS. The website should include:

A homepage title "Delicious Food Recipes"
Navigation menu (Home, Recipes, Contact)
Three recipes with ingredients and preparation steps

Simple styling using CSS

Clean layout suitable for beginners.

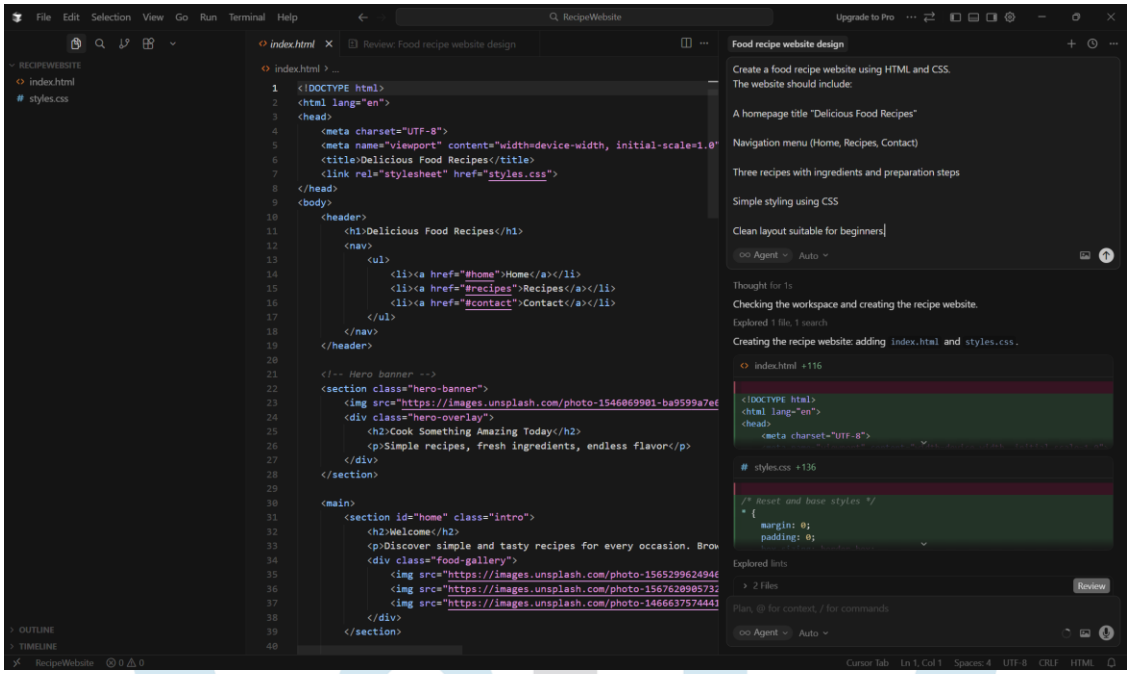

Figure 1: *Code generation using Cursor AI based on natural language instructions.*

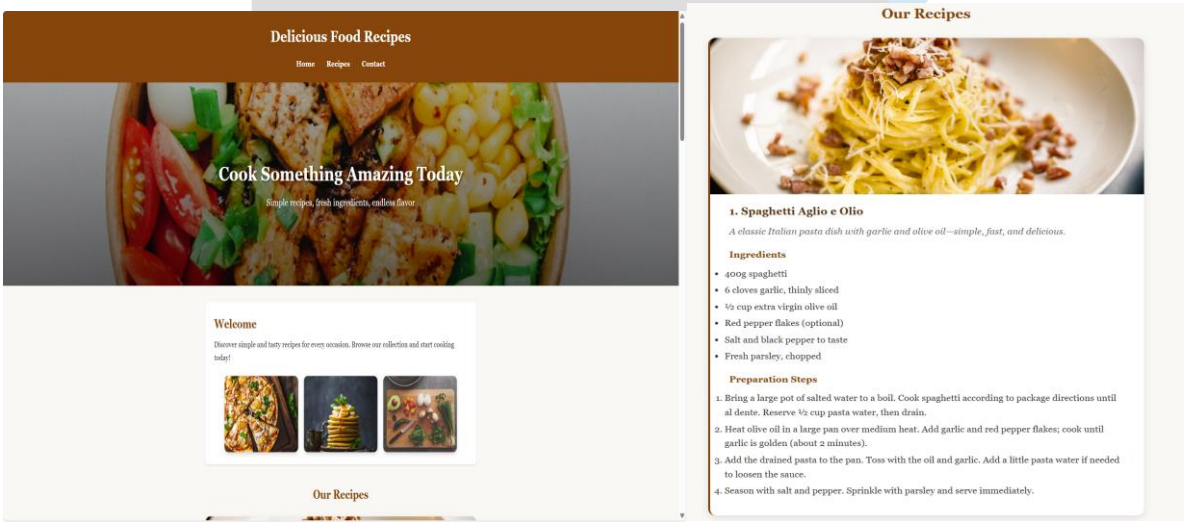

Figure 2: *Recipe website generated using Cursor AI.*

The generated code produced a structured recipe website that included navigation sections and formatted content. The website displayed correctly in a web browser and required only minor adjustments. The experiment demonstrates that Cursor AI can enable non-programmers to develop functional websites using natural language instructions.

During the implementation of the recipe website, the generated code was mostly correct and required only small formatting changes.

In practice, the website could be created within a short time even without prior web development experience.

The navigation structure generated by Cursor AI was clear and easy to modify.

Experiment 2 – Weather Application Development Using GitHub Copilot

In this experiment, **GitHub Copilot** was used to assist in developing a simple weather prediction application using the Python-based Streamlit framework. The objective of this experiment was to evaluate how Generative AI tools integrated within development environments can support non-programmers in building interactive software applications.

GitHub Copilot provides automated code suggestions based on user-written comments and partial code. These suggestions allow users with limited programming experience to develop functional applications without writing code from scratch. In this experiment, GitHub Copilot was used to generate code for a simple weather application that accepts city input and displays predicted weather conditions.

The application was developed using the Python programming language and the Streamlit library, which provides a simple framework for creating interactive web-based applications.

Prompt Used in GitHub Copilot

The following comment prompt was written to guide GitHub Copilot:

Create a simple weather application in python that asks the user to enter a city and displays temperature and weather conditions.

GitHub Copilot generated code suggestions based on this prompt, which were accepted and modified slightly where necessary.

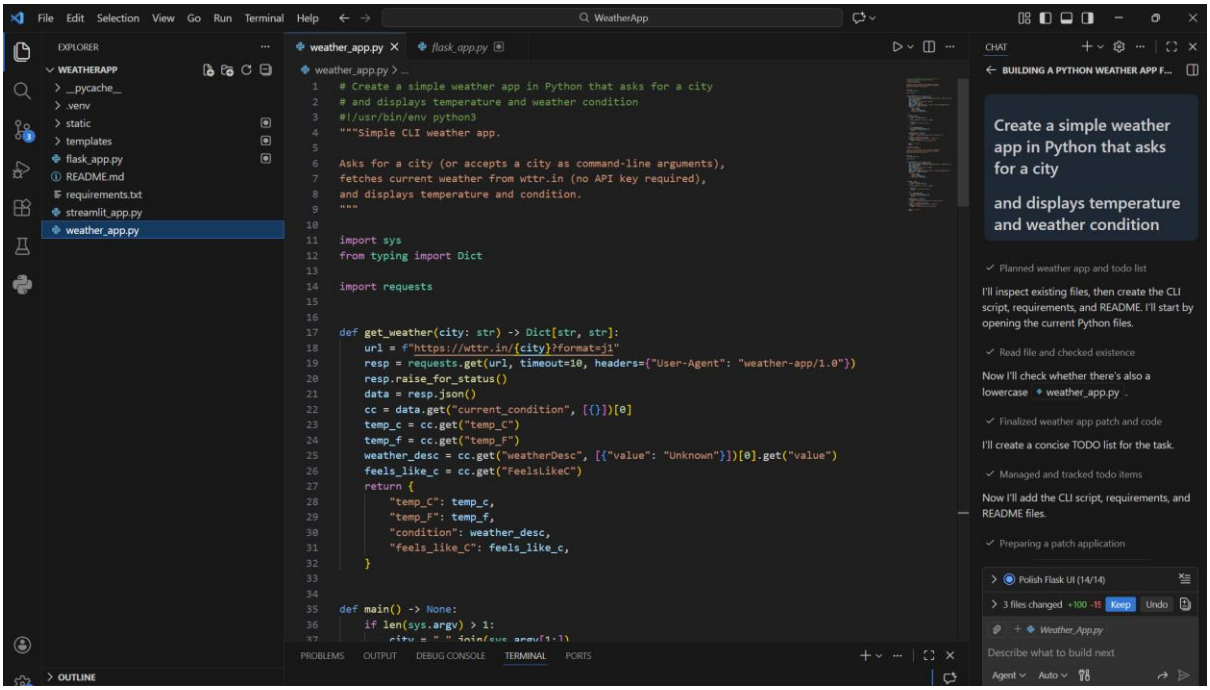

Figure 3: *Code suggestions generated using GitHub Copilot for developing a Streamlit weather application.*

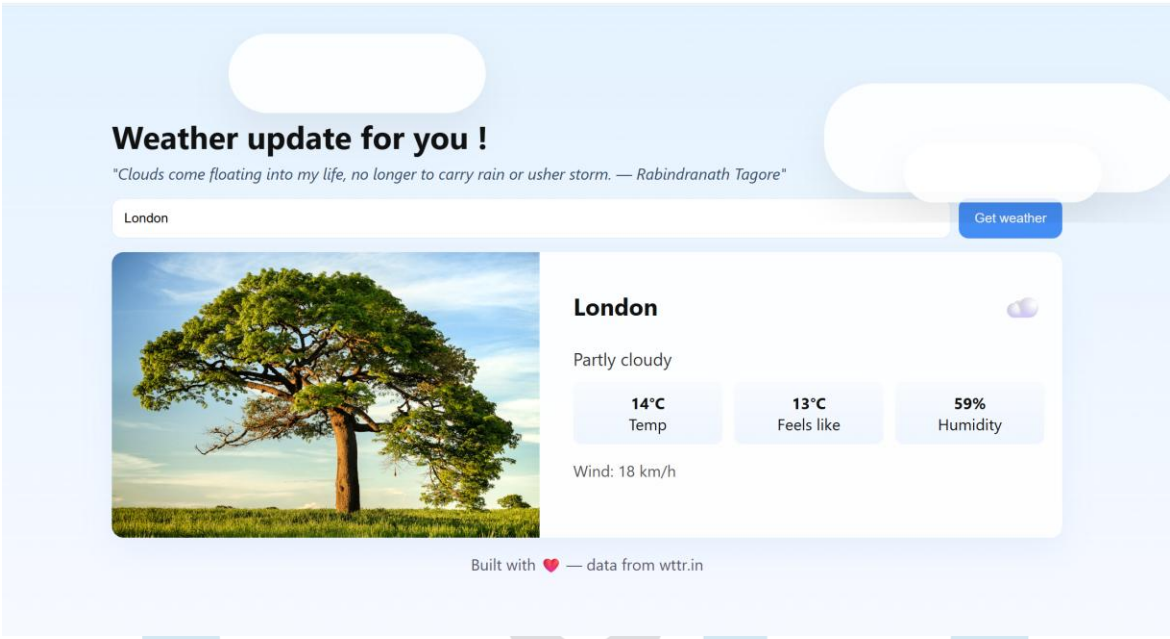

Figure 4: *Streamlit weather prediction application generated with the assistance of GitHub Copilot.*

The Streamlit weather application developed with the assistance of GitHub Copilot executed successfully and produced the expected output. The application allowed users to enter the city and displayed predicted weather conditions through a simple interactive interface.

GitHub Copilot generated useful code suggestions that required only minor modifications before execution. The results demonstrate that Generative AI tools can assist non-programmers in developing basic interactive applications with minimal programming knowledge.

During the development of the weather application, GitHub Copilot provided useful suggestions that helped complete the program quickly. Some suggestions required small corrections before the application ran properly.

The Streamlit interface was easy to configure and did not require advanced programming knowledge.

Experiment 3 – Debugging Support

Generative AI tools were used to identify and correct errors in Python code.

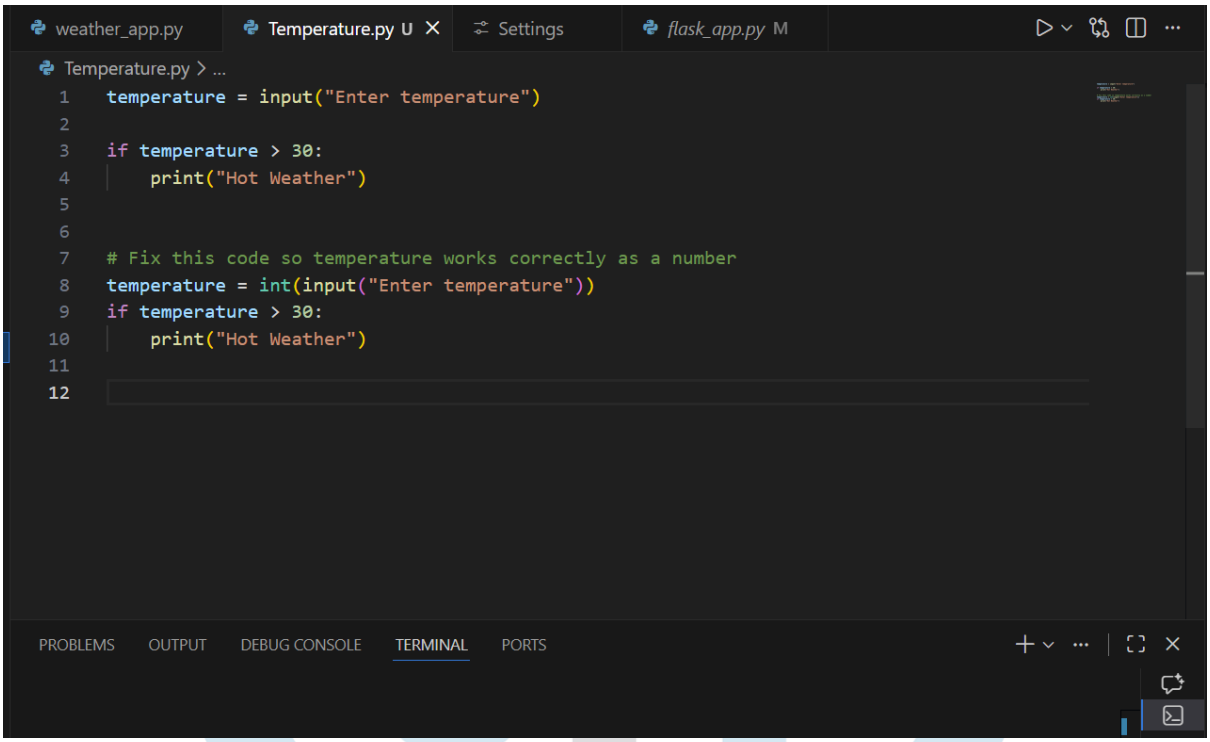

Figure 5: *Error correction suggestions generated using GitHub Copilot.*

GitHub Copilot was used to assist in correcting programming errors. A comment describing the required correction was provided, and GitHub Copilot generated a corrected version of the code. The corrected program executed successfully.

In several cases, the error messages were difficult to understand, but the AI suggestions helped identify the problem quickly.

This made the debugging process easier compared to manually searching for solutions.

---

4. Results and Discussion

The results demonstrate that Generative AI tools significantly simplify the process of application and website development for non-programmers. Users were able to generate functional code by providing simple natural language instructions.

One important observation during the experiments was the reduction in development time when using Generative AI tools. Tasks that would normally require significant effort could be completed more quickly with AI assistance. The generated code was generally understandable even for users with limited programming experience.

In most cases, only minor modifications were needed before the programs worked correctly.

The generated website required minimal modification before use. This suggests that Generative AI tools can provide useful starting points for non-programmers who want to build websites.

The Streamlit Weather application example showed that Generative AI tools can generate executable code with correct syntax. This allows beginners to understand programming logic without writing code from scratch.

Another important observation was the usefulness of Generative AI tools in debugging code. Error messages can be difficult for beginners to interpret. Generative AI tools provide explanations and corrections that improve understanding.

Generative AI tools also reduce development time. Tasks that would normally require hours of learning can be completed within minutes using AI-assisted code generation.

However, some limitations were observed. The generated code occasionally required modification to meet specific requirements. Users must review AI-generated code carefully to ensure correctness.

Overall, the results indicate that Generative AI tools improve accessibility and efficiency in software development for non-programmers.

---

5. Advantages of Generative AI for Non-Programmers

Generative AI tools provide several advantages for individuals without programming experience.

Reduced Learning Curve

In the past, building software or websites often required months of studying complex programming languages and mastering intricate development tools. Now, with the help of AI-powered platforms, even beginners can create functional applications with minimal effort.

Faster Development

These tools can instantly generate code snippets or even entire programs based on simple instructions, dramatically speeding up the development process.

Improved Understanding

Beyond just writing code, generative AI acts as an intelligent assistant throughout the journey. If you encounter a problem or want to understand the logic behind a piece of code, AI can break down the concepts in straightforward, easy-to-understand language. This makes learning more accessible and less intimidating, empowering more people to participate in software development.

Error Correction

Generative AI tools are adept at identifying mistakes or potential bugs in your code, highlighting errors before they become major issues and offering clear suggestions for how to resolve them.

The impact of these advancements goes beyond mere convenience. Businesses can prototype ideas and launch products faster, educators can offer hands-on experiences without the steep learning curve, and individuals can bring their creative visions to life without relying on expert programmers. As generative AI continues to evolve, it's likely that building technology will become even more intuitive and inclusive, opening doors for a wider range of people to innovate and solve real-world problems through code.

---

6. Limitations

However, despite these advantages, these tools are far from infallible. The code they generate can sometimes contain errors, logical flaws, or inefficiencies that could cause your program to malfunction or perform poorly. Because of this, it's essential to thoroughly review and test any AI-generated code before integrating it into your project. You can't simply trust that the output will work flawlessly; careful oversight is still necessary to ensure reliability and quality.

Additionally, the effectiveness of these AI tools heavily depends on the clarity and precision of your instructions. If you provide a prompt that is vague, incomplete, or ambiguous, the tool may produce code that fails to address your actual needs or completely misses the intended functionality. This means that users must be able to clearly articulate their requirements and think critically about how to communicate their goals to the AI system.

Complex applications still require advanced programming knowledge.

---

7. Conclusion

After conducting these experiments, the conclusion is undeniable: Generative AI tools offer tremendous support to beginners, especially those with no prior programming background. The emergence of generative AI has revolutionized software development, lowering the barriers that once made it a daunting field for newcomers. Where people once needed to spend weeks or months learning the basics before building anything functional, these tools now empower even complete novices to create simple applications and websites simply by describing their ideas in everyday language. Generative AI has the potential to transform software development by enabling individuals without technical backgrounds to participate in application and website creation.

The impact of this shift is profound. Generative AI doesn't just simplify the coding process—it democratizes access to technology creation, allowing a much broader range of people to participate. By translating natural language into working code, these tools open doors for individuals who might never have considered themselves capable of programming. They make experimentation much more approachable, enabling users to try out their ideas and see immediate results, all without requiring a deep understanding of coding syntax or logic.

When users encounter problems or bugs, generative AI can offer explanations and suggest fixes, helping beginners grasp fundamental programming concepts as they go. This hands-on guidance accelerates the learning process, making it less intimidating and more engaging. It's not just about automating tasks—it's about fostering confidence and curiosity, encouraging users to explore and iterate without fear of making mistakes.

As a result, generative AI is transforming the entry points into software development. For students, hobbyists, or professionals from non-technical backgrounds, it represents an unprecedented opportunity to turn ideas into reality. The technology acts as a bridge, connecting creativity with technical execution, and reshaping what it means to learn and create in the digital age. For anyone starting out on their programming journey, these tools are truly a game-changer, leveling the playing field and inspiring a new wave of innovation from voices that might otherwise have gone unheard.

Future work may explore the use of Generative AI tools in developing more complex applications and integrated systems.

---

8. References

[1] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, et al.,
"Language Models are Few-Shot Learners," *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[2] GitHub Inc.,
"Quantifying GitHub Copilot's Impact on Developer Productivity," GitHub Research, 2023.

[3] Cursor AI,
"Cursor AI Documentation," 2024. Available: https://cursor.sh

[4] S. Noy and W. Zhang,
"Experimental Evidence on the Productivity Effects of Generative Artificial Intelligence," Massachusetts Institute of Technology (MIT), 2023.

[5] A. Sahay, A. Indamutsa, and D. Di Ruscio,
"Supporting the Understanding and Comparison of Low-Code Development Platforms," *IEEE International Conference on Software Architecture Companion*, 2020.