

HOW AI CAN HELP DEVELOPERS CODE FASTER AND SMARTER

Featuring:

- Analysis of AI tools
- Tips to improve efficiency
- Experts insights



Table Of Contents

02	Introduction	20	AI Tools for Auditing Smart Contracts
03	What is AI?	23	How to choose the right AI tools for your development process
04	How can AI be used to enhance the development process?	24	Tips to help you integrate AI into your development workflow
04	What are the benefits of using AI-powered tools for developers?	25	The Future of AI in development
05	Types and examples of AI-powered tools for developers	25	Sure, here are some helpful free resources for developers to learn how to use AI:
06	Code Completion Tools	26	Resources for developers to find articles on integrating AI tools:
09	Bug Detection and Code Analysis Tools	27	Explore repositories related to specific AI tools by searching for the tool name
12	Code Refactoring Tools	28	Survey with Developers in the Web3 Space
14	Testing Tools	29	Conclusion
17	Automated Code Review Tools	30	techFiesta

Introduction

Artificial intelligence (AI) is rapidly changing the world, and the field of development is no exception. In the past, developers were responsible for every aspect of the development process, from writing the code to testing and debugging it. However, Artificial Intelligence is quickly changing all of that. AI-powered tools are now available that can help developers code faster, smarter, and more efficiently.

It's not that developers are being replaced by robots—rather, AI-powered tools are making them more productive and more effective, enabling them to produce higher-quality software at lower cost.

An example of an Artificial Intelligence tool that has had a significant impact on the development process is GitHub Copilot. In June 2022, it was reported that GitHub Copilot was already generating 27% of developers' code and in February 2023, an average of 46% of code was built using GitHub Copilot across all programming languages, and 61% among developers using Java.

This indicates that the artificial intelligence-powered tool, GitHub Copilot has a positive impact on developers' productivity and satisfaction. In fact, GitHub Blog reported that 90% of developers using GitHub Copilot completed their tasks faster, with 73% being able to stay in the flow and conserve mental energy. 75% of developers who use GitHub Copilot also shared that they feel more fulfilled and able to focus on more satisfying work.

In this guide, we will explore how artificial intelligence can be used to enhance the development process. We will discuss the benefits of using AI-powered tools, and we will provide examples of how these tools can be used to improve the quality of software.

We will also discuss the future of artificial intelligence in development, and we will provide tips on how developers can get started with AI-powered tools.

WHAT IS AI?

Artificial intelligence (AI) is a branch of computer science that deals with the creation of intelligent agents, which are systems that can reason, learn, and act autonomously. AI is frequently applied to the project of developing systems that can perform tasks that are typically considered to require human intelligence, such as reasoning, problem-solving, and language processing.

It has the potential to revolutionize many aspects of our lives, from the way we work to the way we interact with the world around us.



Here are some examples of artificial intelligence systems that are currently in use:

- Self-driving cars use AI to navigate roads and avoid obstacles.
- Virtual assistants like Siri and Alexa use AI to understand our commands and respond to our questions.
- Spam filters use artificial intelligence to identify and block unwanted emails.
- Fraud detection systems use AI to identify and prevent fraudulent transactions.

Artificial intelligence is a young field, but it has the potential to make a significant impact on our world. It has been used in a variety of fields, including healthcare, finance, and transportation. In recent years, it has begun to be used in software development. It now helps developers with a variety of tasks, including code generation, testing, deployment, etc.

HOW CAN AI BE USED TO ENHANCE THE DEVELOPMENT PROCESS?

AI can be used to enhance the development process in a variety of ways. For example, it helps developers:

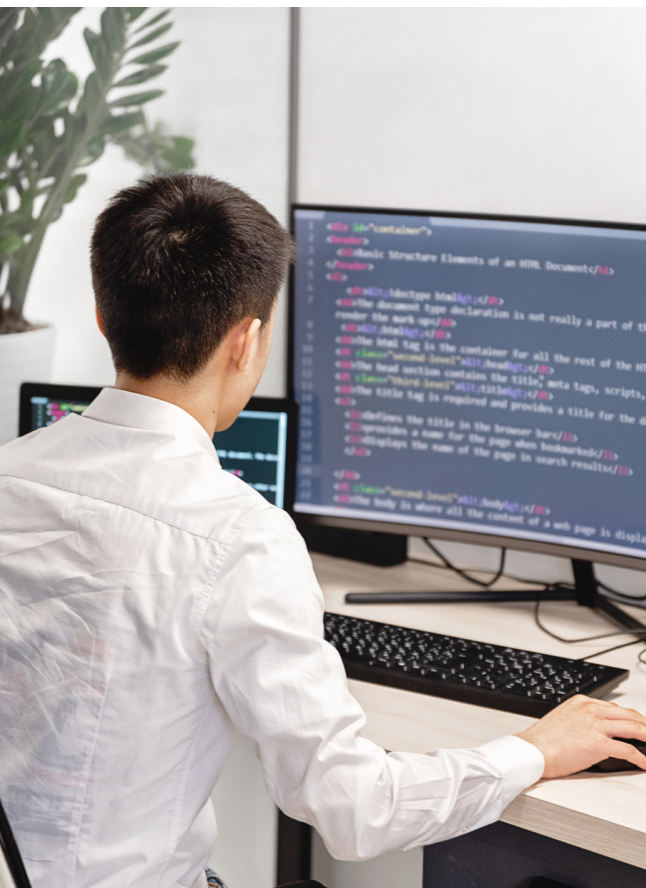
- Write code faster
- Write more secure code
- Write codes efficiently
- Find bugs faster
- Improve the quality of their code
- Automate the testing process
- Deploy their code more quickly
- Monitor their applications in real-time
- Troubleshoot problems more easily



WHAT ARE THE BENEFITS OF USING AI-POWERED TOOLS FOR DEVELOPERS?

There are several benefits to using AI-powered tools for developers. These benefits include:

- Increased productivity: Artificial intelligence-powered tools can help developers code faster and more efficiently. This can free up developers' time so that they can focus on more creative and strategic tasks.



- **Improved quality:** AI-powered tools can help developers write more secure, reliable, and efficient code. This can help to reduce the number of bugs and errors in software, which can save time and money.
 - **Reduced costs:** They can help to reduce the cost of development by automating tasks that would otherwise be done manually. This can free up developers' time and resources so that they can focus on more important tasks.
 - **Increased innovation:** AI-powered tools can help developers to innovate by providing them with new tools and techniques. This can help developers to create new and innovative products and services.
-

TYPES AND EXAMPLES OF AI-POWERED TOOLS FOR DEVELOPERS

Artificial Intelligence tools are changing the way developers write code. These tools can help developers write code faster, improve code quality, and find and fix bugs.

Outlined below are the types of AI-powered tools for developers:

1. Code completion tools
2. Bug Detection and Code analysis tools
3. Code refactoring tools
4. Testing tools
5. Automated Code Review Tools
6. AI Tools for Auditing Smart Contracts

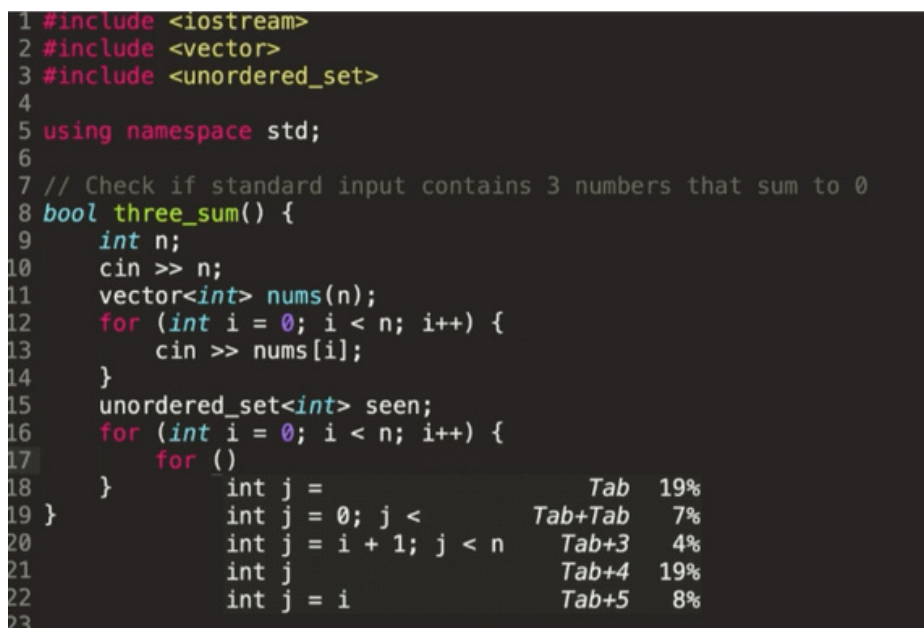
#1 CODE COMPLETION TOOLS

Code completion tools leverage artificial intelligence and machine learning algorithms to assist developers in writing code more quickly and accurately. They analyze existing codebases and offer context-aware suggestions for completing lines of code, functions, or even entire code blocks. This can save programmers a lot of time, as they don't have to remember all the different syntax rules for different programming languages. Code completion tools can be a great way to improve your productivity and accuracy when writing code.

There are several different code completion tools available. Some of the most popular code completion tools include:

- TabNine
- GitHub Copilot

TabNine is a free and open-source code completion tool that uses machine learning to provide accurate and relevant suggestions. It supports a wide range of programming languages, including Python, Java, JavaScript, and C++.

A screenshot of a code editor showing C++ code. The code includes headers for <iostream>, <vector>, and <unordered_set>, and uses the std namespace. A function 'three_sum' is defined, which takes an integer 'n' and returns a boolean. It reads 'n' from standard input, creates a vector 'nums' of size 'n', and reads 'n' integers into it. It then uses an unordered_set 'seen' to find pairs of numbers that sum to zero. The code is partially completed, and TabNine suggestions are shown on the right side of the editor. The suggestions are: 'int j =' (19%), 'int j = 0; j <' (7%), 'int j = i + 1; j < n' (4%), 'int j' (19%), and 'int j = i' (8%). The suggestions are preceded by 'Tab' or 'Tab+' indicating the number of tabs to insert.

```
1 #include <iostream>
2 #include <vector>
3 #include <unordered_set>
4
5 using namespace std;
6
7 // Check if standard input contains 3 numbers that sum to 0
8 bool three_sum() {
9     int n;
10    cin >> n;
11    vector<int> nums(n);
12    for (int i = 0; i < n; i++) {
13        cin >> nums[i];
14    }
15    unordered_set<int> seen;
16    for (int i = 0; i < n; i++) {
17        for ()
18    }
19 }
20
21     int j =
22     int j = 0; j <
23     int j = i + 1; j < n
24     int j
25     int j = i
```

Suggestion	Percentage
Tab	19%
Tab+Tab	7%
Tab+3	4%
Tab+4	19%
Tab+5	8%

A screenshot showcasing TabNine in use

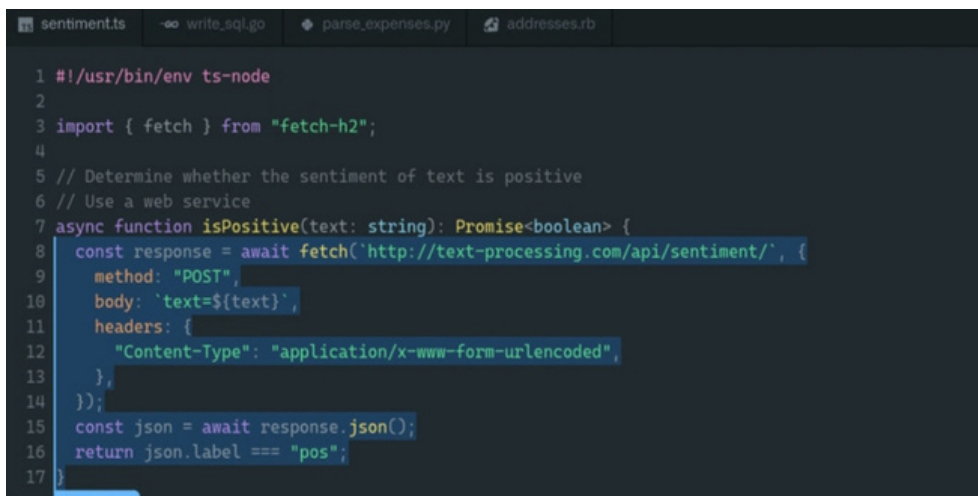
Its key features are:

- Multi-language support
- IDE integrations
- Local and cloud-based processing
- Automatic code refactoring

Website: <https://www.tabnine.com/>

VSCode Plugin: bit.ly/3QTVc8c

GitHub Copilot is a paid code completion tool that is powered by OpenAI's GPT-3 language model. It can generate code, suggest edits, and complete functions. GitHub Copilot is currently in beta, but it has already been used to generate millions of lines of code.



```
1 #!/usr/bin/env ts-node
2
3 import { fetch } from "fetch-h2";
4
5 // Determine whether the sentiment of text is positive
6 // Use a web service
7 async function isPositive(text: string): Promise<boolean> {
8   const response = await fetch('http://text-processing.com/api/sentiment/', {
9     method: "POST",
10    body: `text=${text}`,
11    headers: {
12      "Content-Type": "application/x-www-form-urlencoded",
13    },
14  });
15  const json = await response.json();
16  return json.label === "pos";
17 }
```

GitHub Copilot in use

Key Features:

- Multi-language support
- IDE integrations
- Context-aware suggestions
- Automatic documentation generation

Website: github.com/features/copilot

VSCode Plugin: bit.ly/3R1mWg2

Tip: The best code completion tool for you will depend on your specific needs and preferences. If you are looking for a free and open-source tool, then TabNine is a good option. If you are looking for a paid tool that can generate code, then GitHub Copilot is a good option.

#2 BUG DETECTION AND CODE ANALYSIS TOOLS

These tools help developers find potential bugs, security vulnerabilities, and other errors in their code. Developers automate the detection of potential issues using these tools and they also provide actionable insights. Code analysis tools can be a valuable tool for any developer, as they can help to improve the quality of code and prevent errors.

Some of the most popular code analysis tools include:

- DeepCode
- SonarLint

DeepCode: DeepCode is a code review tool that can help developers improve the quality of their code. DeepCode uses machine learning to analyze code repositories and identify potential security vulnerabilities, bug risks, and performance issues. By learning from a massive dataset of open-source code and applying advanced semantic analysis techniques, DeepCode can provide developers with precise and context-aware recommendations to improve code quality. It supports a wide range of programming languages, including Java, Python, JavaScript, and C++.



Snyk Code

Secure code with a developer-friendly experience.

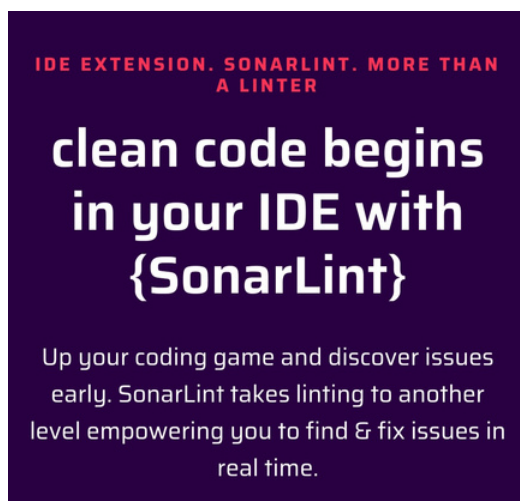
DeepCode

Key Features:

- Multi-language support
- IDE integration
- Continuous analysis
- Customizable rules

Website: <https://www.deepcode.ai/>

SonarLint: SonarLint is a tool that helps developers write better code. It does this by analyzing code in real-time and highlighting potential problems, such as bugs, vulnerabilities, and code smells. SonarLint integrates directly into popular IDEs, so developers can get instant feedback as they write code. This helps developers catch problems early in the development process, which can save time and money.



SonarLint

Key Features of SonarLint include:

- IDE integration
- Customizable rules
- Integration with SonarQube and SonarCloud
- Multi-language support

Website: <https://www.sonarlint.org/>

Tip: Use code analysis tools early and often in your development process. The earlier you find potential bugs and security vulnerabilities, the easier and cheaper it will be to fix them. It also helps you to catch potential problems before they cause major issues.

#3 CODE REFACTORING TOOLS

Code refactoring is the process of improving the internal structure of existing code without changing its external behavior. It can make code more readable, maintainable, and efficient. AI-powered automatic code refactoring tools can help developers identify areas of code that can be improved and suggest changes that enhance code quality.

Some popular automatic code refactoring tools include Sourcery and Codota.

Sourcery is a tool that uses artificial intelligence to help developers write better code. It does this by analyzing code and suggesting changes that can improve its structure, readability, and efficiency. These changes can be made in real-time, while the developer is still working on the code. This helps developers to write cleaner, more maintainable code with less effort.



Sourcery

Key Features:

- Multi-language support
- IDE integration
- Context-aware refactorings
- Performance optimizations

Website: <https://sourcery.ai/>

Codota is a code completion and refactoring tool that uses machine learning to suggest code completions and refactorings. Codota is available in a variety of programming languages, including Java, Python, and JavaScript.

It predicts what code you are likely to write next, and then suggests it to you. This can save you a lot of time and effort, as you don't have to type out the same code over and over again. Codota also helps you to write more idiomatic code, by suggesting the most common ways of writing certain functions or classes. This can help to make your code more readable and maintainable.



Key Features:

- Multi-language support
- Code linting
- Code documentation

Get the plugins here. <https://plugins.jetbrains.com/plugin/7638-codota-ai-autocomplete-for-java-and-javascript/versions#tabs>

Tip: Both Sourcery and Codota are powerful tools that can help developers improve the quality of their code. However, it is important to note that these tools are not perfect and should not be used without careful consideration. It is always important to review any changes suggested by these tools before adding them to your code.

#4 TESTING TOOLS

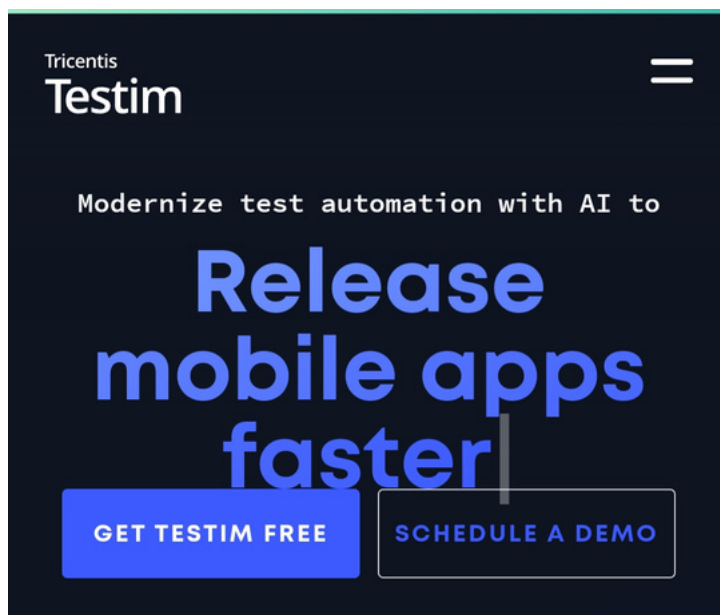
AI testing tools are used by developers to automate software testing tasks. These tools can be used to find bugs, verify the functionality of the software, and improve the overall quality of the software.

They can help to reduce the time and effort required to test software, and they help find bugs that would not be found with traditional testing methods.

There are several different testing tools available on the market. Some of the most popular testing tools include:

- Testim
- Diffblue

Testim is an AI-powered test automation platform that helps developers create, execute, and maintain end-to-end tests for mobile and web applications. Testim uses machine learning to learn the behavior of applications and generate test cases automatically. This can save developers a significant amount of time and effort, and it can also help to improve the quality of their tests.



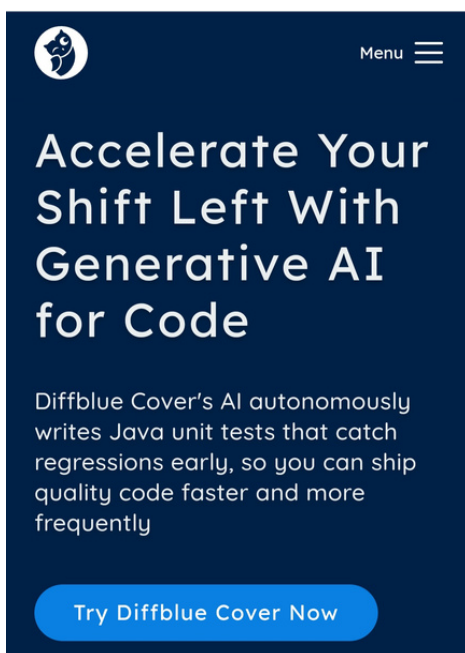
Testim

Key Features:

- Automatic test generation
- AI-powered test case selection
- Parallel test execution
- Test reporting

Website: <https://www.testim.io/>

Diffblue is an automated testing tool that uses machine learning to generate unit tests for Java applications. It can create test cases that cover a wide range of scenarios and edge cases, improving test coverage and reducing the risk of undetected issues.



Diffblue

Key Features:

- Automatic test generation
- IDE integration
- Test suite optimization
- Test coverage analysis

Website: <https://www.diffblue.com/>

Tip: AI testing tools learn from the data that they are given. Make sure to train the tool properly so that it can find the bugs that you are looking for.

AI testing tools are not a replacement for human testing. Use the tool in conjunction with other testing methods to get the best results.

#5 AUTOMATED CODE REVIEW TOOLS

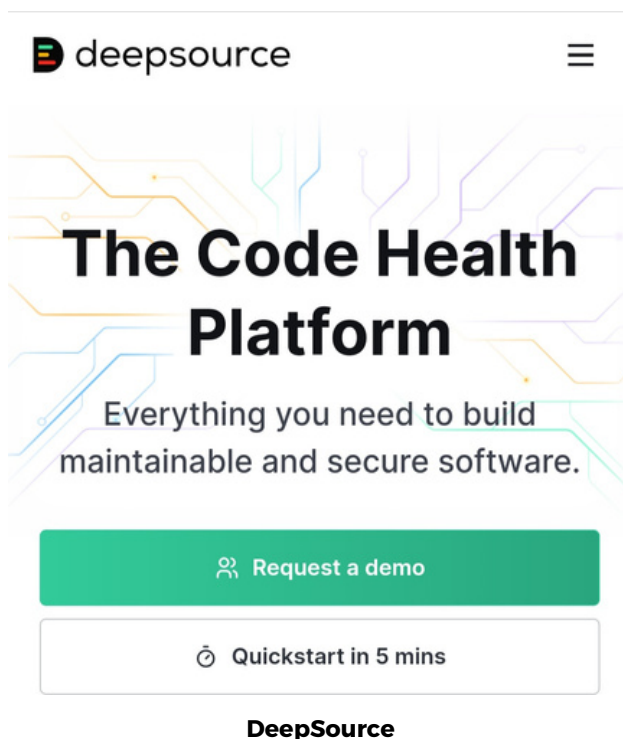
These tools use artificial intelligence and machine learning to identify potential issues in code, such as errors, security vulnerabilities, and coding standards violations. This can help developers to improve the quality of their code and reduce the risk of defects making it into production.

Code review assistance tools are becoming increasingly popular as they can save developers a significant amount of time and effort. They can also help to improve the quality of code by identifying potential issues that may not be caught by human reviewers.

Examples of code review assistance tools:

- DeepSource
- Amazon CodeGuru

DeepSource is a tool that helps developers review and improve their code. It uses machine learning to analyze code for potential issues, such as bugs, security vulnerabilities, and code smells. DeepSource then provides developers with real-time feedback and actionable insights to help them fix these issues. This can help to streamline the code review process and improve overall code quality.



Key Features:

- Multi-language support
- Continuous Analysis
- Customizable rules
- Integration with popular tools

Website: <https://deepsource.com/>

Amazon CodeGuru is a code review service that uses machine learning to help developers improve the quality, performance, and security of their code. CodeGuru analyzes code for potential issues, such as bugs, security vulnerabilities, and performance bottlenecks. It then provides developers with real-time feedback and actionable recommendations to help them fix these issues.



Amazon CodeGuru

Key Features:

- IDE integration
- Automated code reviews
- Performance profiling
- Language support

Website: <https://aws.amazon.com/codeguru/>

Tip: It is important to note that automated code review tools are not perfect. They can miss some problems, and they can also generate false positives. It is important to use multiple automated code review tools to get a complete picture of the quality of the code. It is also essential to have a human review of the results of the automated code review tools to catch any problems that the tools missed.

AI TOOLS FOR AUDITING SMART CONTRACTS

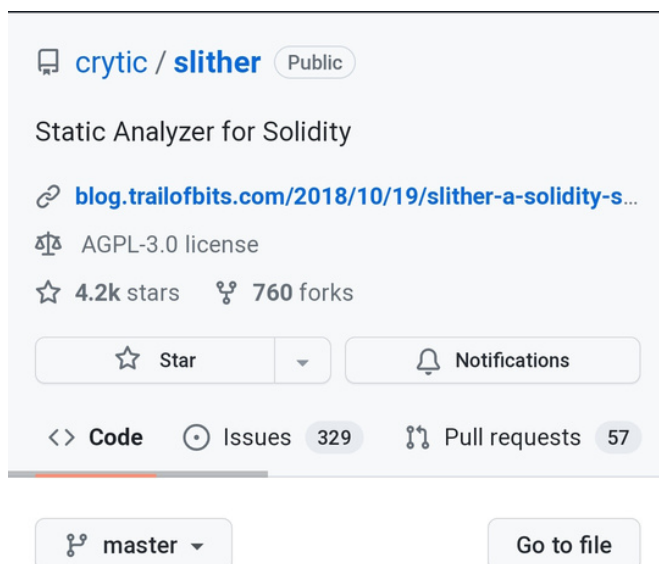
These are a type of software that uses AI to find security vulnerabilities in smart contracts. These tools can be used to find a variety of vulnerabilities, including reentrancy attacks, time-of-check-to-time-of-use (TOCTOU) vulnerabilities, and integer overflow vulnerabilities.

They work by analyzing the code of the smart contract and looking for patterns that are associated with known vulnerabilities. These tools can also be used to generate reports that detail the findings of the audit. There are many different AI tools for auditing smart contracts. Some of the most popular tools for auditing smart contracts include:

- Slither
- MythX

Slither is an open-source static analysis tool for smart contracts written in Solidity. It is developed by Trail of Bits, a security research firm. Slither uses a variety of AI techniques to find security vulnerabilities in smart contracts, including control flow analysis, data flow analysis, and symbolic execution. Slither can be used to find a wide variety of security vulnerabilities, including reentrancy attacks, time-of-check-to-time-of-use (TOCTOU) vulnerabilities, and integer overflow vulnerabilities. Slither is available for free on GitHub.





Slither

Key Features of Slither:

- Open-source
- Available for free on Github
- Can find a wide variety of security vulnerabilities

Website: <https://github.com/trailofbits/slither>

MythX is a cloud-based smart contract auditing service that uses artificial intelligence to find security vulnerabilities. It is developed by ConsenSys, a blockchain technology company. MythX integrates with a variety of development tools, including Remix, Truffle, and Solidity Browser. MythX can be used to find a variety of security vulnerabilities, and integer overflow vulnerabilities. MythX is available for free for open-source projects. Paid plans are available for private projects.

Smart contract security service for Ethereum

MythX™ by **ConsensSys Software Inc™** is the premier security analysis service for Ethereum smart contracts. Our mission is to ensure development teams avoid costly errors and make Ethereum a more secure and trustworthy platform.

MythX

Key Features

- IDE integrations
- Cloud-based
- Integrate security into the smart contract development lifecycle.

Website: <https://mythx.io/>

Tip: Use multiple AI tools to get a more complete picture of the security of your smart contract. No single tool is perfect, and they can all miss some vulnerabilities. By using multiple tools, you can increase your chances of finding all of the vulnerabilities in your smart contract.

HOW TO CHOOSE THE RIGHT AI TOOLS FOR YOUR DEVELOPMENT PROCESS

AI-powered tools can be a valuable asset for developers. By choosing the right tools, you can improve your development process and get more done.

Here are a few things to consider when choosing a tool for your development needs:

- **Your specific needs:** What are your biggest challenges in development? What are your goals for improving your development process?
- **The features of the tool:** What features are important to you? Do you need a tool that can help you with code completion, testing, or debugging?
- **The price of the tool:** AI-powered tools can range in price from free to hundreds of dollars per month. It's important to find a tool that fits your budget.
- **The reviews of the tool:** Read reviews of AI-powered tools to see what other developers have said about them. This can help you to get a better idea of how well the tool works and whether it's right for you.
- **Consider the size of your team:** If you're working with a small team, you may not need a tool that can handle a large number of users. However, if you're working on a large team, you'll need a tool that can scale to meet your needs.
- **Think about the future:** When choosing a tool, it's important to think about the future. Make sure to choose a tool that is constantly being updated and improved.



- Don't be afraid to experiment: There are many different AI-powered tools available. Don't be afraid to experiment with different tools to find the ones that work best for you.

Once you've considered these factors, you can start to narrow down your choices.

TIPS TO HELP YOU INTEGRATE AI INTO YOUR DEVELOPMENT WORKFLOW

- Start small: Don't try to integrate too many tools at once. Start with one or two tools that you think will have the biggest impact on your workflow.
- Get help from experts. If you're not sure how to integrate these tools into your workflow, there are many resources available to help you. There are books, articles, and online courses that can teach you the basics of artificial intelligence and how to use the tools in your development process. At [EkoLance](#), we are committed to helping professionals in the web3 space succeed in their careers. That's why we have started a mentorship program where aspiring web3 professionals can be mentored by industry experts for free. Apply [here](#).
- Be patient. It takes time to learn how to use AI tools effectively. Don't expect to be an expert overnight. Just keep practicing and you'll eventually get the hang of it.

THE FUTURE OF AI IN DEVELOPMENT

Artificial Intelligence (AI) has made its way into the realm of software development, offering a range of powerful tools that can help developers code smarter, not harder.

In this guide, we have explored a variety of AI-driven tools that can assist developers in different aspects of the development process, such as code completion, bug detection, code quality analysis, code refactoring, automated review tools, testing tools, etc.

The rise of artificial intelligence in the development landscape is a game-changer for developers, enabling them to work smarter. By embracing and integrating these AI tools into their development processes, developers can not only save time and effort but also improve the quality, reliability, and maintainability of their applications. As AI continues to advance, we can expect even more sophisticated and powerful tools to emerge, further revolutionizing the way developers approach their work and pushing the boundaries of what's possible in the realm of software development.



SURE, HERE ARE SOME HELPFUL FREE RESOURCES FOR DEVELOPERS TO LEARN HOW TO USE AI:

- Coursera: Coursera offers a variety of online courses on artificial intelligence, including introductory courses, courses on specific topics, and courses on how to use AI in different industries.
- edX: edX is another online learning platform that offers a variety of AI courses.

- Udacity: Udacity offers a variety of nano degrees on artificial intelligence, which are immersive online programs that teach you the skills you need to use AI in your career.
- Google AI: Google AI offers a variety of resources for developers, including tutorials, blog posts, and research papers.
- Microsoft: Microsoft offers a variety of AI resources for developers, including tutorials, blog posts, and SDKs.
- IBM: IBM offers a variety of resources for developers on artificial intelligence, including tutorials, blog posts, and Watson services.

RESOURCES FOR DEVELOPERS TO FIND ARTICLES ON INTEGRATING AI TOOLS:

- Medium AI: <https://medium.com/topic/ai>
- Towards AI: <https://medium.com/towards-artificial-intelligence>
- Towards Data Science: <https://towardsdatascience.com/>
- AI category: <https://towardsdatascience.com/tagged/artificial-intelligence>
- GitHub Search: <https://github.com/search>

EXPLORE REPOSITORIES RELATED TO SPECIFIC AI TOOLS BY SEARCHING FOR THE TOOL NAME

- TensorFlow: <https://www.tensorflow.org/>
- PyTorch: <https://pytorch.org/>
- scikit-learn: <https://scikit-learn.org/>
- OpenAI GPT-3: <https://beta.openai.com/>
- Microsoft Azure AI: <https://azure.microsoft.com/en-us/overview/ai-platform/>

Online AI Communities: Join online communities and forums dedicated to AI development. Platforms like [AI Stack Exchange](#), [Kaggle](#), and [AI Village on Reddit](#) provide spaces for developers to ask questions, share insights, and collaborate on AI-related projects.

AI Research Papers and Publications: Stay updated with the latest advancements in AI research by exploring academic journals, conferences, and research papers. Platforms like [arXiv](#) and [Google Scholar](#) provide access to a vast collection of AI-related publications.

AI Development Tools and IDEs: Explore specialized AI development tools and integrated development environments (IDEs) designed to streamline the AI development process. Examples include [Jupyter Notebook](#), [PyCharm](#), and [IBM Watson Studio](#).

AI Experimentation Platforms: Utilize AI experimentation platforms such as [Google Colab](#), [Microsoft Azure Machine Learning](#), and [Amazon SageMaker](#). These platforms provide cloud-based environments and resources to experiment, train models, and deploy AI applications.

SURVEY WITH DEVELOPERS IN THE WEB3 SPACE

To gain further insights into how artificial intelligence has helped developers improve productivity in Web3, we conducted a survey where we asked two professionals to share their experience with using AI in the development process. Here are the key takeaways from our survey:

Samuel, a talented full-stack developer shared that he has experienced a remarkable 20% increase in coding speed by using AI-powered code completion tools. These tools have become his go-to solution, streamlining development and providing insightful suggestions. Samuel highly recommends artificial intelligence tools to all developers, recognizing their ability to boost productivity and code quality. He said they have been instrumental in overcoming complex challenges, such as gas estimations in smart contract development.

Emmanuel, a blockchain developer, says that AI is an integral part of his development process. He advises that it will be essential for developers to know and learn how to use AI tools to debug, create templates, research quickly and get some questions answered, instead of having to go to StackOverflow every single time.

Emmanuel concluded that artificial intelligence will not replace developers but it will replace the developer that can not utilize and leverage it. The future of technology, he said, is bright with the advent of innovative tools and all of these innovations are to aid a developer's development process, and increase productivity and efficiency.

CONCLUSION

AI tools have emerged as powerful allies for developers, enhancing productivity, efficiency, and code quality. With their advanced capabilities and intelligent features, these tools are transforming the way developers approach software development. Throughout this guide, we have explored some notable tools for developers and their key features. By incorporating these tools into their workflow, developers can unlock new possibilities and streamline their development processes.

From AI-powered code completion and suggestion tools like TabNine and GitHub Copilot to AI-driven smart contract audits tools like MythX and Slither, developers now have access to a wide range of tools tailored to their specific needs. These tools offer automated assistance, improve code quality, and expedite development tasks, ultimately empowering developers to focus on innovation and problem-solving.

As the technology landscape continues to evolve, embracing artificial intelligence tools has become increasingly crucial for developers to stay competitive and deliver high-quality software solutions. By leveraging the power of AI, developers can amplify their capabilities, accelerate development cycles, and create robust applications.

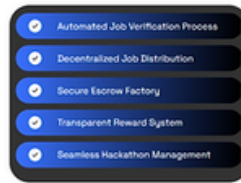
EkoLance will be launching an upcoming smart contract audit training course, aimed at helping developers upskill in this critical domain. This course will provide valuable knowledge and expertise to navigate the complexities of smart contract security and contribute to building secure and reliable blockchain-based applications. Don't miss out. Click here to visit our [website](#) and check out this [page](#) to stay up to date on when the course is launched.

Now, let's take a look at an exciting platform that aims to harness the power of blockchain to fuel innovation: techFiesta, EkoLance's blockchain-powered hackathon platform!

Hack Your Way to Innovation with techFiesta

Create, collaborate and innovate with the best professionals in the tech space.

Join Waitlist



techFiesta

techFiesta offers a revolutionary hackathon experience. With amazing features on the techFiesta platform, EkoLance will transform the hackathon experience for developers and pioneer a new era of collaborative innovation. Outlined below are the benefits developers can expect when participating in a hackathon in techFiesta:

- Get to work with the most innovative and actively building tech companies.
- Enjoy fast and secure payment in stablecoins using blockchain technology developed by HUMAN Protocol, which is audited and secure.
- When developers participate in hackathons, they automatically become part of the EkoLance community where they will frequently get upskilling opportunities with educational workshops, seminars, and events.
- Get noticed by companies sponsoring the events, which may be on the lookout for talented developers to hire or work with on a freelance basis.
- Participating in hackathons on techFiesta opens doors to potential funding opportunities for developers. They can get funded through companies participating in hackathons or investors.
- techFiesta attracts media coverage, providing developers with a platform to showcase their work and gain exposure for their projects, potentially leading to partnerships and wider recognition.
- techFiesta offers developers the opportunity to present their projects or products at exclusive events, providing further opportunities for promotion, networking, and attracting potential investors.

Click [here](#) and sign up for our waiting list today and be one of the first developers to participate in our upcoming hackathons with some of the biggest names in the industry in June 2023.

Follow us on our various social channels:

Twitter: [TechFiesta](#)

Discord: [TechFiesta](#)

Telegram: [TechFiesta](#)

To get notified about other important developments within EkoLance, follow us on our social media handles below:

[LinkedIn](#), [Twitter](#), [Facebook](#) & [Instagram](#)

Join our amazing community on [Discord](#) too.

Thank you for choosing EkoLance as your guide in the exciting world of emerging technologies. We look forward to supporting your growth and success in the future!