**AI for CI/CD**

A few of them include, Jenkins: an open-source automation server known for its CI/CD capabilities.

IBM Watson Studio, WS, and Watson Machine Learning, WML: offers DevOps automation for ML workflows.

Codefresh: utilizes AI for intelligent caching and predictive scaling, enhancing the development process.

GitLab CI/CD: seamlessly integrates with version control for streamlined CI/CD management.

PagerDuty AIOps: leverages intelligence and automation to aid engineering teams and incident response.

Harness: uses AI to revolutionize software release processes through automation and analysis.

Snyk: incorporates AI and ML for automated security testing and vulnerability management.

Dynatrace's Davis AI: an automated engine assisting in managing complex IT environments.

**AI Tools for Security in Software**

*Secure coding platforms*:

Qwiet AI Pre-Zero platform that integrates security into CI/CD pipelines and development tools, providing rapid feedback to identify and address high-priority code vulnerabilities.

Snyk Code tool that quickly analyzes code, offering actionable insights and static analysis results to assist engineers in addressing code issues efficiently within their workflows.

GitHub advanced security simplifies vulnerability detection and offers autofix suggestions for code scanning alerts, aiding developers in resolving vulnerabilities without leaving their workflow.

Veracode fix suggests remediations for security flaws in code and open-source dependencies, leveraging Veracode's proprietary data sets.

Endor Labs introduces DroidGPT. A chatbot helping developers select secure and updated open-source components for their projects.

Microsoft Security Copilot assists security teams in investigating and responding to security incidents.

BurpGPT is a burp suite plugin that uses GPT to analyze HTTP requests and responses.

EscalateGPT identifies privilege escalation vulnerabilities using GPT in misconfigured identity access and management IAM policies for AWS.

*Platforms to detect threat*:

Sophos Intercept X. It's an endpoint protection solution that employs deep learning technology to identify and respond to threats.

Symantec Endpoint Security. It's an AI-powered tool that features machine learning. It identifies vulnerabilities and takes preventative measures before they result in damaging attacks.

Splunk User Behavior Analytics, which focuses on user behavior. It also establishes a baseline for different user activities and can recognize potential breaches by detecting differences in user behavior.

Vectra Threat Detection and Response tool utilizes behavioral analytics to identify and stop threats across different environments. It uses hundreds of metadata elements to detect attackers and minimize false positives.

IBM QRadar advisor with Watson, which includes a QRadar advisor with Watson and automate security operations center operations. Watson analyzes anomalies and offers root cause analysis without human input. Over time, QRadar Advisor provides increasingly helpful advice on improving security, helping companies stay protected from emerging threats

**AI Code Review Tools**

DeepCode

GitHub Copilot

PullRequest

CodeScene

CodeClimate

Synk

**Ai Debugging Tools**

DeepCode

Sentry

DeepScan

Testim

Mabl

Codacy

XRebel

**AI documentation Tools**

NaturalDocs

Mintlify

Doxygen

DocuWriter.ai

**AI Tools incorporated in Training and Education**

Quizgecko

Teachable

7Tabs

Kajabi

**Ethical Consideration in AI**

Bias and discrimination (Moderation API was used to warn or block certain types of unsafe content)

Intellectual property rights (Looka is an AI tool that can be used for logo creation)

Privacy and data protection (Deep Nostalgia is a privacy-conscious generative AI tool which allows users to animate historical family photographs)

Misinformation and manipulation (Legal Robot, an AI tool that simplifies legal language. By translating complex legal terms into plain language, this tool aims to enhance legal understanding for both professionals and laypeople)

Accountability and transparency (Murf, a text-to-speech engine that allows users to create synthetic vocal recordings. It promotes accountability and transparency by clearly indicating that the output is generated by AI and providing options for selecting different voices and dialects)

**AI tools that can significantly enhance app development**

Lumen5,

Deep Nostalgia

Gen-1

Krisp

Legal Robot

Dall-E-2

Castle

Stable Diffusion 2

Soundraw

Lalal.ai

Clean up pictures

Looka

Fireflies

Murf

**Module 2 Cheatsheet: Generative AI for DevSecOps and Testing**

|  |  |
| --- | --- |
| **Description** | **Tools** |
|  |  |
| **CI/CD using AI** | • Jenkins |
| • IBM Watson Studio |
| • Codefresh |
| • Atlassian |
| • GitLab CI/CD |
| • PagerDuty AIOps |
| • CircleCI |
| • Travis CI |
| • Harness |
| • Snyk |
| • Dynatrace's Davis AI |
| **Software security using AI (secure coding)** | • Qwiet AI preZero |
| • Snyk Code |
| • GitHub Advanced Security |
| • Veracode Fix |
| • Endor Labs |
| • Microsoft Security Copilot |
| • BurpGPT |
| • EscalateGPT |
| **Documentation using AI** | • Doxygen |
| • NaturalDocs |
| **Code reviews using AI** | • DeepCode |
| • CodeClimate |
| **Innovation using AI** | • Lumen5 |
| • Deep Nostalgia |
| • Gen-1, Krisp |
| • Legal Robot |
| • Dall-E 2 |
| • Castle |
| • Stable Diffusion 2 |
| • Soundraw |
| • Lalal.ai |
| • Cleanup.Pictures |
| • Looka |
| • Fireflies |
| • Murf |

Prompts:

|  |  |
| --- | --- |
| **Task** | **Prompts/Links** |
| **Setting up the development environment for Generative AI and ChatGPT** | • Install Visual Studio Code (VS Code) • Link to download - <https://code.visualstudio.com/> |
| • How to get the OpenAI/ChatGPT API key • Follow this Link for OpenAI - <https://platform.openai.com/signup/> |
| **Building a basic chatbot using ChatGPT** | **Download Node.js:**  Visit the official Node.js website - <https://nodejs.org/>  **Installing OpenAI and Running Express** npm install express openai  node server.js  **Prompts**  • What is agile methodology?  • What is the difference between a compiler and an interpreter?  • What is the difference between a stack and a queue?  • What is the difference between a linked list and an array? |
| **Generating table design** | • Could you provide guidance on designing a database table for customers in a company, including fields like name, email, and location? I'm looking to understand the best practices and considerations for structuring such a table effectively. |
| • Can you please give the above in table format? |
| • Could you demonstrate the structure of a database table for customers in a company, outlining the fields like name, email, and location in a clear table format? |
| • Could you advise on creating a separate table to store customer locations, specifically focusing on incorporating fields like state, country, and zip code? |
| • Regarding the database structure, how should I design a separate 'Purchases' table to store information about customer purchases, such as the date of purchase, amount, and more? What would be the best approach to linking this 'Purchases' table to the main customer table for efficient data organization and retrieval? |
| **Generating code with AI tools** | **Code for finding the maximum and minimum elements in a list using Python:**   1. 1 2. 2 3. 3 4. 4 5. 5 6. 6 7. 7 8. 8 9. 9 10. 10 11. 11 12. 12 13. numbers = [5, 2, 9, 1, 7] 14. max\_num = numbers[0] 15. min\_num = numbers[0] 17. for num in numbers[1:]: 18. if num > max\_num: 19. max\_num = num 20. if num < min\_num: 21. min\_num = num 23. print("Maximum element:", max\_num) 24. print("Minimum element:", min\_num)   Copied!  **Code for optimization**   1. 1 2. 2 3. 3 4. 4 5. 5 6. 6 7. def find\_largest(arr): 8. largest = arr[0] 9. for num in arr: 10. if num > largest: 11. largest = num 12. return largest   Copied!  **Prompt for function generation**  Develop a Python function to eliminate duplicates and sort a list of 10 country names.  **Prompt for using the append function**  Create a Python function using ‘append’ to find and return duplicate elements in an integer list. |
| **Creating a Dockerfile using generative AI** | • What is a Dockerfile? |
| • What's the first crucial step when creating a Dockerfile for a Node.js application? |
| • Following the base image selection, set the working directory inside the container to /app. |
| • As selected working directory /app, bring package.json and package-lock.json to the working directory. |
| • Now, how can we install dependencies and copy the remaining application code to the working directory? |
| • The Node.js app needs to be accessible on a specific port, what Dockerfile instruction can we include for this purpose? |
| • How to specify the command to start the Node.js application using CMD? |
| • Combine all the steps to create a complete Dockerfile for a Node.js application. |

**Author(s)**

Ritika Joshi



**© IBM Corporation 2023. All rights reserved.**