

Projektbericht

Einleitung

In diesem Projekt wollte ich eine fortgeschrittene Task-Management-App erstellen, indem ich Technologien wie HTML und CSS, die ich neu gelernt habe, sowie Python Flask und JavaScript, mit denen ich zuvor keine Erfahrung hatte, nutze. Mein Ziel war es, die Kraft der Künstlichen Intelligenz zu nutzen, um auch mir unbekannte Tools einzusetzen und so eine anspruchsvolle Anwendung zu entwickeln. Gleichzeitig wollte ich durch den gezielten Einsatz von KI eine höhere Effizienz in kürzerer Zeit erreichen.

Die Hauptwerkzeuge, die ich in diesem Projekt eingesetzt habe, sind ChatGPT und GitHub Copilot. Zusätzlich habe ich einige Recherchen über Google durchgeführt. Diese Werkzeuge trugen dazu bei, eine benutzerfreundliche Anwendung zu entwickeln, die dem Projekt Geschwindigkeit und Flexibilität verlieh.

Die Herausforderungen, denen ich bei der Verwendung dieser Werkzeuge begegnete, verdeutlichten zugleich die Notwendigkeit, KI richtig zu steuern. Ich möchte auch meine Erfahrungen über die potenziellen Vorteile und Einschränkungen von KI-gestützten Werkzeugen im Entwicklungsprozess teilen.

Projektbeschreibung

Mein Task-Management-Projekt ist eine Webanwendung, mit der Benutzer ihre Aufgaben effizient verwalten können. Benutzer können neue Aufgaben hinzufügen, bestehende Aufgaben bearbeiten und deren Status aktualisieren, um den Überblick zu behalten. Das Projekt wurde mit modernen Webtechnologien wie HTML, CSS und JavaScript entwickelt und durch das Flask-Framework in Python mit einem Backend integriert.

Die Hauptstruktur des Projekts besteht in der Datei `index.html`, die das allgemeine Seitenlayout der Anwendung darstellt. Diese Seite enthält alle Elemente, mit denen der Benutzer interagieren kann. Benutzer können das „Enter todo“-Formular verwenden, um Aufgaben hinzuzufügen. Zur Gestaltung eines benutzerfreundlichen und modernen Designs wurde die Datei `styles.css` verwendet, in der Farbschema und Stile sorgfältig

ausgewählt wurden, um die Benutzerfreundlichkeit zu verbessern. In der Datei `script.js` wurden mehrere Funktionen entwickelt, die es den Benutzern ermöglichen, mit der Anwendung zu interagieren. Diese Funktionen erleichtern das Hinzufügen, Aktualisieren und Löschen von Aufgaben, sodass Benutzer ihre Aufgaben leicht organisieren können.

Um die Bearbeitung von Aufgaben zu ermöglichen, wurde die Datei `edit.html` erstellt, in der Benutzer Kategorien ändern und Änderungen speichern können.

Auf der Hauptseite befindet sich das Formular „Enter todo“, mit dem Benutzer neue Aufgaben hinzufügen können. Die Aufgaben können in die Kategorien „Personal“, „Work“ oder „Other“ eingeteilt werden. Wenn die Benutzer das Formular ausfüllen und auf „ADD“ klicken, wird die neue Aufgabe hinzugefügt und im Abschnitt „TO DO“ angezeigt.

Aufgabenstatus:

- **TO DO:** Enthält neu hinzugefügte Aufgaben.
- **IN PROGRESS:** Zeigt Aufgaben an, an denen aktiv gearbeitet wird. Aufgaben können durch Anklicken des „task check“-Icons oder durch Ziehen aus dem „TO DO“-Bereich in diesen Abschnitt verschoben werden.
- **COMPLETED:** Zeigt abgeschlossene Aufgaben. Wenn eine Aufgabe abgeschlossen ist, kann der Benutzer das Kontrollkästchen links aktivieren, um die Aufgabe in diesen Abschnitt zu verschieben.

Abgeschlossene Aufgaben werden im Abschnitt „COMPLETED“ angezeigt. Die Aufgaben in diesem Bereich haben eine höhere Deckkraft. Benutzer können abgeschlossene Aufgaben zurück in den „TO DO“- oder „IN PROGRESS“-Bereich verschieben, indem sie das Kontrollkästchen deaktivieren oder auf das „task check“-Icon klicken.

Für jede Aufgabe stehen Symbole zum Bearbeiten und Löschen zur Verfügung, mit denen Benutzer ihre Aufgaben einfach bearbeiten oder löschen können. Durch Drag-and-Drop können Benutzer zudem die Reihenfolge der Aufgaben nach Belieben ändern. Für jede Kategorie wurden unterschiedliche Farben für die „task check“-Icons verwendet, um eine bessere Benutzererfahrung zu gewährleisten. Benutzer können Aufgaben auch nach Kategorien filtern, um nur die gewünschten Aufgaben anzuzeigen.

Am oberen Rand der Seite befindet sich eine Suchleiste, mit der Benutzer nach bestimmten Begriffen innerhalb der Seite suchen können. In der oberen rechten Ecke ist ein Profil-Symbol mit Benutzerinformationen angezeigt. Neben dem Symbol befindet sich der Benutzername, und darunter gibt es ein Dropdown-Menü mit „Profil“,

„Einstellungen“ und „Abmelden“. Benutzer können über das Profil-Symbol auf ihre Einstellungen zugreifen und sich abmelden. Zudem können sie zwischen einem Dunkel- und einem Hellmodus wechseln, je nach ihrer Präferenz und Umgebung.

Wenn Benutzer Aufgaben bearbeiten möchten, werden sie auf die Seite `edit.html` weitergeleitet. Auf dieser Seite finden sie:

- **Bearbeitungsformular:** Hier können Benutzer die Details der Aufgabe (Titel, Beschreibung, Kategorie usw.) ändern.
- **Speichern-Schaltfläche:** Benutzer können ihre Änderungen speichern, um danach zur Hauptseite zurückzukehren.
- **Pomodoro-Timer:** Benutzer können diesen Timer zur Fokussierung auf ihre Aufgaben verwenden. Der Timer bietet die Optionen „Start“, „Stop“ und „Reset“. Diese Seite unterstützt ebenfalls den Dunkelmodus.

Im Backend des Projekts wurde das Flask-Framework in Python verwendet. Die Datei `app.py` stellt die Verbindung zwischen der Webanwendung und der Datenbank her. So werden alle hinzugefügten und bearbeiteten Aufgaben der Benutzer in einer SQL-Datenbank gespeichert und aktualisiert.

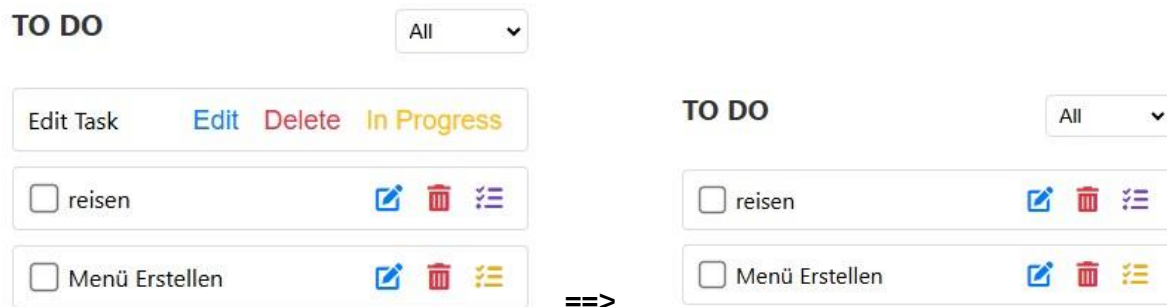
Benutzer können ihre Aufgaben zur Datenbank hinzufügen, sodass alle Seiten synchron arbeiten. Änderungen auf jeder Seite werden direkt in der Datenbank gespeichert.

Flask verwaltet die HTTP-Anfragen, die für die Aktionen der Benutzer erforderlich sind, wie das Hinzufügen, Bearbeiten und Löschen von Aufgaben. Die Benutzerdaten werden dynamisch verwaltet, sodass die Anwendung stets aktuell und benutzerfreundlich ist.

Herausforderungen

1. Wenn ich mehrere Anweisungen gleichzeitig eingefügt habe, ist die Struktur meines Codes durcheinandergeraten. Zum Beispiel haben sich beim gleichzeitigen Ändern des "Headers" und Hinzufügen einer Kategorienoption zu meinen Aufgaben Layout-Probleme ergeben. Ich habe gelernt, dass es effizienter ist, diese Schritte nacheinander auszuführen. Zuerst habe ich also den Header angepasst und anschließend die Kategorienoption hinzugefügt. Auf diese Weise habe ich ein funktionaleres und ästhetisch ansprechenderes Design erreicht.
2. Der Code, den GPT generiert, ist nicht immer korrekt. Um dieses Problem zu lösen, muss ich ihn jedes Mal überprüfen und so lange Anpassungen anfordern, bis das gewünschte Ergebnis erreicht ist. Außerdem muss ich meine Anfragen oft in

verschiedenen Formulierungen stellen. Zum Beispiel wollte ich, dass die Bearbeitung einer Aufgabe direkt auf derselben Seite erfolgt, wenn ich auf das Bearbeitungssymbol klicke. Trotz wiederholter Anfragen neigte GPT jedoch dazu, immer wieder zusätzliche Schaltflächen hinzuzufügen. Auch nach mehreren Versuchen wurde das Problem nicht wie gewünscht gelöst. Letztendlich habe ich eine separate Seite, edit.html, erstellt, auf der die Änderungen vorgenommen werden können, und so das Problem gelöst.



3. Um die gewünschten Ergebnisse zu erhalten, muss ich oft detaillierte Informationen angeben. Um GPT dazu zu bringen, Änderungen und Erweiterungen für unsere App vorzunehmen, ist es erforderlich, die neuen Anpassungen mit den entsprechenden englischen und terminologischen Bezeichnungen anzufordern. Dafür benötige ich zumindest grundlegende Kenntnisse zum Thema. Zum Beispiel musste ich wissen, dass der Abstand zwischen Elementen über den „margin“-Wert definiert wird, die Positionierung mit dem Parameter „position“ erfolgt und passende Icons beispielsweise über Seiten wie Google Icons importiert werden können.

Indem ich klare Anweisungen wie „margin-top: 10px für die Sandwichbar“ oder „profilicon color: green“ verwendet habe, konnte ich präzisere Anforderungen an GPT stellen. Andernfalls wurden Änderungen und Ergänzungen von GPT oft zufällig bestimmt, was den Erfolg der Ergebnisse beeinträchtigte. Bei fehlenden Kenntnissen habe ich ChatGPT sowie Google für Recherchen genutzt und so die Probleme gelöst.

4. Für ein großes Projekt ist es wichtig, Schritt für Schritt vorzugehen. Obwohl mein Projekt derzeit noch kein sehr großes Projekt ist, habe ich dennoch auf diese Regel geachtet. Zuerst habe ich das Grundgerüst der Seite mit HTML erstellt und anschließend dafür gesorgt, dass Details, Styles und Funktionen reibungslos funktionieren.

Zum Beispiel wollte ich, nachdem mein Projekt eine fortgeschrittene Stufe erreicht hatte, das Design meines Checkbox-Styles in ein animiertes Icon umwandeln. Nach dieser Änderung kam es jedoch zu Konflikten mit meinen anderen Styles, was dazu führte, dass sich die anderen Elemente auf der Seite verschoben haben. Die Korrektur

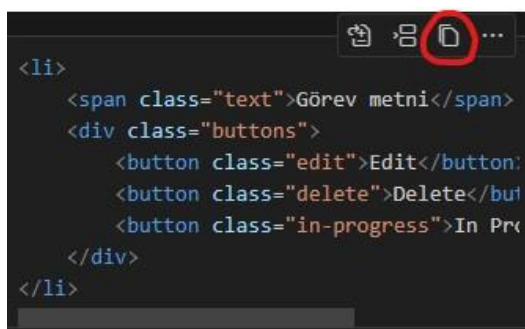
dieser Probleme war sehr zeitaufwendig, und ich musste das Projekt löschen und neu schreiben. Hätte ich diesen Schritt bereits am Anfang des Projekts gemacht, während ich noch das Grundgerüst erstellt habe, wäre dieses Problem wahrscheinlich gar nicht erst aufgetreten.

Daher habe ich danach jede Änderung in der richtigen Reihenfolge und systematisch vorgenommen, und es traten keine weiteren Probleme dieser Art auf oder ließen sich schwerer beheben.

5. Obwohl ich ChatGPT gebeten habe, bestimmte Fehler zu korrigieren, wurden die gewünschten Änderungen manchmal nicht übernommen. In solchen Fällen ist es notwendig, auf andere GPT-Modelle oder Google-Recherchen zurückzugreifen. Zum Beispiel wollte ich die Größe der Suchleiste (searchbar) anpassen, indem ich Breiten- und Höhenwerte angegeben habe, aber diese Änderungen wurden im Browser nicht übernommen. Schließlich habe ich nach einer Lösung gesucht und herausgefunden, dass ich dieses Problem mit der Verwendung von "!important" lösen kann. Auf diese Weise konnte ich die Suchleiste auf die gewünschte Größe einstellen.

```
.search-bar input {  
  padding: 5px 10px;  
  font-size: 14px;  
  border: 1px solid #ddd;  
  border-radius: 5px;  
  width: 100px !important;  
  height: 10px !important;  
}
```

6. Manchmal funktioniert der "Apply"-Button bei neuen Änderungen nicht. Wenn das der Fall ist, werden die Änderungen in der App nicht angezeigt. In solchen Fällen sollte man die Änderung manuell durch Kopieren/Einfügen übernehmen. Dieses Problem trat bei vielen Änderungen auf, und ich konnte es fast immer lösen, indem ich den neuen Code per "Copy/Paste" eingefügt habe.



```
<li>  
  <span class="text">Görev metni</span>  
  <div class="buttons">  
    <button class="edit">Edit</button>  
    <button class="delete">Delete</button>  
    <button class="in-progress">In Progress</button>  
  </div>  
</li>
```

7. Man sollte nicht jeden von GPT erstellten Code akzeptieren; vor jeder Änderung sollte man mit "Show Change" die Änderung überprüfen. Außerdem sollte man, bevor man den Code von GPT anwendet, gegebenenfalls eine Sicherung des Codes anfertigen. So kann man, falls es zu irreversiblen Fehlern im Projekt kommt, mit den Backup-Dateien weiterarbeiten.

Ich habe auch festgestellt, dass es sehr vorteilhaft ist, nicht nur vor Änderungen, sondern auch nach bestimmten Fortschritten eine Sicherung anzulegen. Dadurch konnte ich jederzeit einfach auf eine frühere Version zurückkehren und mit effektiveren Designs an meinem Projekt weiterarbeiten.

```
header h1 {  
  margin: 0;  
  font-size: 30px;  
  position: absolute;  
  left: 50%;  
  top: 50%;  
  font-size: 24px;  
  position: absolute;  
  left: 50%;  
  transform: translateX(-50%);  
}
```

8. GPT gibt oft die Warnung "Sorry, the response hit the length limit. Please rephrase your prompt." aus, weil es eine Beschränkung für die Zeilenanzahl gibt, wenn es neuen Code schreibt. Ich habe diese Situation mehrmals erlebt und festgestellt, dass die gleichen Lösungen mich nicht zum Ziel führen. Ich habe es gelöst, indem ich gesagt habe: "Vervollständige mir die fehlenden Teile." Manchmal habe ich auch erreicht, dass mir nur die benötigten Zeilen geschrieben werden, um die Beschränkung zu umgehen. Gelegentlich bin ich auch zu einem anderen Chatfenster gewechselt und habe meine Anfrage von vorne wiederholt. Eine andere Möglichkeit, um besser mit diesem Problem umzugehen, könnte sein, den zu ändernder Teil zu kopieren und die Anfrage entsprechend zu stellen.

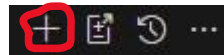
⊗ Sorry, the response hit the length limit. Please rephrase your prompt.



9. Vor dem Anwenden neuer Änderungen sollte die Zeilenanzahl der Seite überprüft werden. Bei einer übermäßigen Verringerung oder Erhöhung sollte man misstrauisch werden und mit GPT den aktuellen Stand der Seiten und mögliche Mängel überprüfen. Zum Beispiel habe ich festgestellt, dass meine CSS-Datei 300 Zeilen hat, als ich Änderungen am Design vornehmen wollte. Wenn ich danach sehe, dass diese Zeilenanzahl auf 280 gesunken ist, weiß ich, dass etwas nicht stimmt und dass GPT möglicherweise einige notwendige Codezeilen weggelassen hat. In diesem Fall würde ich die Anwendung des "apply"-Befehls vermeiden und die Änderungen erneut anfordern.

10. Wenn die Chat-Seite von GPT zu viele Nachrichten enthält, kann es zu einer Verlangsamung der Leistung kommen, und die Antworten werden langsamer. Um dieses Problem zu lösen, habe ich zuerst ein Backup des letzten Standes meines Codes gemacht und dann eine neue Chat-Seite geöffnet. Außerdem habe ich festgestellt, dass es eine gute Lösung ist, zu einer neuen Seite zu wechseln, wenn die aktuelle Code-Seite

fehlerhafte Codes ausgibt oder sich ständig wiederholt.



11. Wenn Änderungen manuell an den Codes vorgenommen werden, muss dies GPT mitgeteilt werden. Andernfalls erkennt es nach jedem „Apply“-Vorgang die manuellen Änderungen nicht und akzeptiert stattdessen seine eigenen letzten Änderungen als Standard. Um mit dieser Situation umzugehen, habe ich die vorgenommenen Änderungen GPT vorgestellt.



Die Vorteile von GPT

1. Einige Codes, die ich mit GPT erstellt habe, haben das Aussehen meiner Seite leicht auf eine bessere Gestaltung angehoben. Zum Beispiel habe ich meine Seite mit dem Befehl „Gestalte das Header-Design benutzerfreundlich und modern“ aktualisiert und dabei erfolgreiche Ergebnisse erzielt. Wenn dieser Vorgang jedoch nicht zu einem erfolgreichen Ergebnis führt, kann es erforderlich sein, einige manuelle Änderungen zur Korrektur vorzunehmen. Die Möglichkeit, dass es zu einer Störung kommt, sollte immer berücksichtigt werden. **Code-Lokalisierung:** GPT erleichterte die Suche nach spezifischen Codeblöcken und ermöglichte so eine präzisere Bearbeitung.

2. Wenn eine Änderung an einem bestimmten Teil der Seite vorgenommen werden soll, kann der entsprechende Bereich über die Entwicklerseite der Website gefunden werden. Alternativ kann auch GPT einfach gefragt werden (z. B. „Wo befindet sich der CSS-Codeblock für den Header?“). Dies ermöglicht ein präziseres Vorgehen und spart Zeit. Dies war eine der Methoden, die ich häufig verwendet habe. (UNTEN STEHT ES)



The screenshot shows a GitHub Copilot chat window. At the top, the user has entered the query "wo ist header style" (where is the header style) in a text input field. Below the input, the Copilot interface shows "Generating..." and "Used 1 reference". The reference is listed as "# styles.css:1-539 static". The Copilot response states: "The header styles are located in the # styles.css file. Here is the relevant section for the header styles:". Below this text, a red 'X' mark is visible. The relevant CSS code is displayed in a code block:

```
header {
  background-color: #fff; /* Header a:
  color: #000; /* Yazılar siyah */
  padding: 20px;
  text-align: center;
  position: relative;
}
```

3. Je größer das Projekt wird, desto schwieriger wird die Verwaltung, und es besteht ein Bedarf an einer gut strukturierten und ordentlichen Architektur. In diesem Fall kann Unterstützung von GPT und ChatGPT in Anspruch genommen werden, um die Seite zu verwalten. Als ich meine Edit-Seite (edit.html) entworfen habe und dabei die Dateien styles.css und script.js gemeinsam mit meiner Hauptseite verwendet habe, stellte ich fest, dass eine Änderung auf einer Seite die andere Seite beeinträchtigte.

Um solche Situationen besser zu bewältigen, fragte ich ChatGPT nach den am häufigsten verwendeten Stilen und Seitenstrukturen auf dem Markt und realisierte mein Projekt entsprechend. Ich habe gelernt, dass ich die CSS- und JS-Codes für die Seiten in meiner App direkt in die HTML-Datei als Inline-Codes integrieren kann. So stellte ich sicher, dass meine Änderungen die Struktur der anderen Seite nicht beeinträchtigen und ich leichter herausfinden kann, aus welcher Seite die Fehler stammen.

```
project-root/  
├─ index.html  
├─ edit.html  
├─ styles.css  
├─ script.js  
├─ app.py  
└─ README.md
```

4. Ich habe meine Website am Ende des zweiten Tages in eine Task Management Website umgewandelt und sie entsprechend strukturiert. Dabei war mir GPT eine große Hilfe. Denn zwischen meinem alten Projekt und dem neuen, das ich erstellen wollte, gab es Unterschiede.

Durch schrittweise Anfragen konnte ich einige Strukturen auf der Seite mit GPT in die neue Form überführen, und schließlich habe ich mein Projekt, das ursprünglich als TODO APP begann, in die jetzige Form als Task Management App umgewandelt.

5. GPT und ChatGPT haben mir während des Programmierprozesses bei der Erstellung vieler meiner Codes geholfen. Oft habe ich ihm gesagt, welchen Code ich erstellen wollte, und er hat ihn für mich generiert. Das hat mir in erheblichem Maße Zeit gespart.

Darüber hinaus haben mir die Aspekte, auf die ich achten sollte, sowie die Erfahrungen, die ich mit den Problemen gemacht habe, die ich begegnet bin, geholfen, maximale Effizienz zu erzielen.

6. KI hat mir sehr dabei, meine Fähigkeiten zu entwickeln und neue Fähigkeiten zu erlernen. Es hat mir klare und präzise Informationen über Themen und Codes gegeben,

die ich nicht kannte. Obwohl ich keine Kenntnisse in JavaScript und Python-Flask habe, hat es viele JS- und Flask-Codes korrekt für mich erstellt.

Darüber hinaus hat es mir auch Codes gegen böswillige Angriffe wie HTML Injection problemlos erstellt. So hat es eine solide Grundlage für die Projekte geschaffen, die ich in Zukunft erstellen werde, und meine Praxis bereichert sowie mir ermöglicht, unterschiedliche Perspektiven zu entwickeln.

7. Eine weitere Möglichkeit, wie KI mir geholfen hat, war, mein Design benutzerfreundlich zu gestalten und Designalternativen anzubieten, die die Bedürfnisse der Nutzer erfüllen. Als Beispiel habe ich Dropdown-Menüelemente mit den von GPT vorgeschlagenen Elementen erstellt.

Bei der Anfrage an die KI habe ich darum gebeten, Vorschläge zu machen, die zu meinem Konzept passen und die Anforderungen der Nutzer erfüllen. Darüber hinaus habe ich sie gebeten, in meinem Design viele benutzerfreundliche Änderungen vorzunehmen, und das war für mein Projekt sehr effektiv.

8. Eine weitere Möglichkeit, wie KI mir geholfen hat, war, mein Design benutzerfreundlich zu gestalten und Designalternativen anzubieten, die die Bedürfnisse der Nutzer erfüllen. Als Beispiel habe ich Dropdown-Menüelemente mit den von GPT vorgeschlagenen Elementen erstellt.

Bei der Anfrage an die KI habe ich darum gebeten, Vorschläge zu machen, die zu meinem Konzept passen und die Anforderungen der Nutzer erfüllen. Darüber hinaus habe ich sie gebeten, in meinem Design viele benutzerfreundliche Änderungen vorzunehmen, und das war für mein Projekt sehr effektiv.

Fazit

ChatGPT und GitHub Copilot haben sowohl beim Schreiben von Code als auch beim Überwinden bestimmter Herausforderungen im Designprozess erheblich zur Beschleunigung beigetragen. Sie haben den Prozess erheblich beschleunigt, indem sie die Grundlage der Code-Struktur schnell erstellt, in jedem Schritt alternative Lösungen angeboten und notwendige Vorschläge zur Vermeidung bestimmter Fehler gegeben haben.

Insbesondere konnte ich bei den neuen Technologien, die ich gelernt habe, direkt über ChatGPT Antworten auf meine Fragen erhalten, ohne umfangreiche Dokumentationen durchsuchen zu müssen, und so mit meinem Projekt fortfahren. GitHub Copilot hat durch die automatische Erledigung wiederkehrender Aufgaben Zeit gespart und Fehler minimiert.

Dank der Geschwindigkeit und Unterstützung, die diese Werkzeuge bieten, hatte ich die Möglichkeit, mein Projekt in kürzerer Zeit abzuschließen. Darüber hinaus hat es meine Fähigkeiten in den Bereichen Problemlösung, kreatives Design und kritische Technologiebewertung weiterentwickelt.

Gleichzeitig haben mir die Herausforderungen und Einschränkungen beigebracht, dass ich in jeder Phase vorsichtig handeln muss. Abschließend möchte ich Folgendes betonen: Es ist sicher, dass sich künstliche Intelligenzwerkzeuge in Zukunft weiterentwickeln und nicht nur im Softwarebereich, sondern in allen Bereichen und Berufen hilfreich sein werden. Deshalb ist es sehr wichtig, dass jeder weiß, wie man diese Werkzeuge richtig einsetzt. Denn diejenigen, die diese Werkzeuge gut nutzen, werden im Vergleich zu denen, die sie nicht nutzen, erfolgreichere Ergebnisse erzielen und in ihren Tätigkeiten mehr hervorstechen.