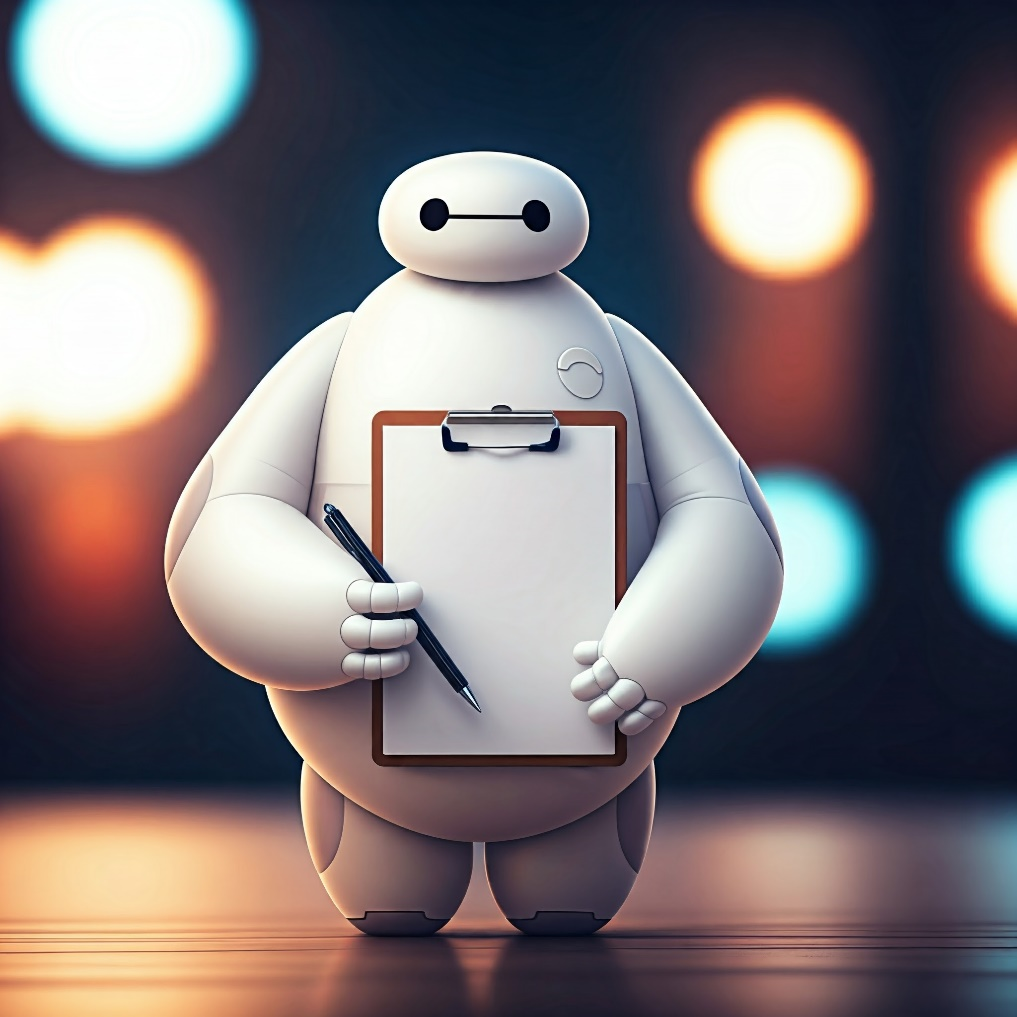
Deckblatt



Inhalt

[Einleitung 3](#_Toc183609741)

[Projekt Title 3](#_Toc183609742)

[Kurzbeschreibung 3](#_Toc183609743)

[Projekt Planung 3](#_Toc183609744)

[Einleitung in die Planung 3](#_Toc183609745)

[Ist Soll Zustand 3](#_Toc183609746)

[Zeitplanung 3](#_Toc183609747)

[Vergleich von Alternativen 4](#_Toc183609748)

[Projekt Durchführung 4](#_Toc183609749)

[Dokumentation der Durchführung 4](#_Toc183609750)

[Beschreibung der Durchgeführten Arbeitsschritte 4](#_Toc183609751)

[Technische Details 5](#_Toc183609752)

[Herausforderungen und Lösungen 5](#_Toc183609753)

[Abschluss 5](#_Toc183609754)

[Ergebnisbeschreibung 5](#_Toc183609755)

[Funktionstest 6](#_Toc183609756)

[Kundeneinweisung 6](#_Toc183609757)

[Fazit 6](#_Toc183609758)

[Reflektion über den Verlauf des Projektes 6](#_Toc183609759)

[Persönliche Erkenntnis und Lernerfahrung 6](#_Toc183609760)

# Einleitung

## Projekt Title

Web-Tool für verständliche und fehlerfreie Texte

## Kurzbeschreibung

In meinem Projekt entwickle ich ein benutzerfreundliches Webtool, mithilfe von Vue 3 und Quasar, das Texte annimmt und über die Google Gemini API auf Rechtschreib und Satzbaufehler überprüft. Die API korrigiert den Text automatisch, ohne den persönlichen Schreibstil des Nutzers zu verändern.

Das Tool richtet sich vor allem an Menschen mit Lese-Rechtschreib-Schwäche, aber auch an Schüler und Studierende sowie an alle, die sich bei Rechtschreibung und Grammatik unsicher fühlen. Mein Ziel ist es, eine einfache Möglichkeit zu schaffen, Texte schnell fehlerfrei und gut lesbar zu machen, ohne dass Nutzer aufwendige Nachbearbeitungen vornehmen müssen.

Dieses Tool soll den Nutzern Zeit sparen und ihnen helfen, die Lesbarkeit und Qualität ihrer Texte zu verbessern. Da sie sich nicht mehr um Grammatik und Rechtschreibung kümmern müssen, können sie sich auf den eigentlichen Inhalt ihrer Texte konzentrieren. In der Zukunft ließe sich das Tool noch erweitern, zum Beispiel um zusätzliche Sprachen oder Funktionen, um eine größere Zielgruppe zu erreichen. Auch eine mobile Version wäre denkbar, die Korrekturen direkt auf Smartphones ermöglicht.

# Projekt Planung

## Einleitung in die Planung

In der Schule machen wir wir ein Schulprojekt wo wir und ein Projekt aussuchen durften, die Anforderrungen waren das es in Python geschrieben ist und wir mindestes eine API anfrage stellen.

## Ist Soll Zustand

Die detaillierte Ist-Soll-Analyse befinden sich in den separaten Dokumenten „IstSollZustand.docx“. Die Datei ist Bestandteil der Projektunterlagen.

## Zeitplanung

**Vorbereitung**

* Ist- und Soll-Analyse: **2 Stunden**
* Projektplanung und Erstellung eines Anwendungsfalldiagramms: **3 Stunden**

**Durchführung**

* Installation und Einrichtung aller Nötigen Programme und Frameworks **3 Stunden**
* Entwicklung der Benutzeroberfläche: **6 Stunden**
* API-Integration und Korrektur-Logik: **10 Stunden**
* Implementierung der Anzeige der Korrekturvorschläge: **4 Stunden**
* Usability-Tests und Feedback-Implementierung: **2 Stunden**

**Abschluss**

* Soll-Ist-Vergleich: **1 Stunden**
* Projektdokumentation: **12 Stunden**
* Letzte Anpassungen und Optimierungen: **3 Stunden**

**Gesamt: 46 Stunden**

## Vergleich von Alternativen

Im Internett gibt es bereits Rechtschreib Tool alerdings muss man immer manuel auswählen wie etwas geschrieben wird, da ist es deutslich einfacher wenn man einfach seinen Fertigen Text wieder bekommt und direkt alle Fehler behoben sind. Gemini ist eine führende Text KI daher bietet es sich an diese zu verwenden um texte berichtigen zu lassen.

# Projekt Durchführung

## Dokumentation der Durchführung

### Beschreibung der Durchgeführten Arbeitsschritte

Ich habe mich im Internett informiert welche KI geeignet ist um mein Projekt durchzuführen da habe ich zwei große Herausgesucht einmal Google Gemini und ChatGPT, aufgrund der Kosten und das Google Gemini kostenlos ist habe ich mich für Google Gemini entschieden und mir dann die Doku angeschaut um zu Prüfen ob eine Intigration überhaubt technisch möglich ist.

Danach habe ich überlegungen angestellt wie ich das Projekt umsezten kann, den Backendserver muss ich mit Python machen dann habe ich mich kurtz Informiert wie ich grafische Oberflächen mit Python baue und wie ich einen Webserver in Python baue. Aufgrund der geringen Zeit habe ich mich für einen Webserver in Python entschieden und das Framework Quasar. Dies ermöglicht es mir vorgefährtigte komponenten zu verwenden die das entwickeln sehr vereinfacht. Dann habe ich mich noch informiert was ich noch alles Brauche damit das Projekt nacher laufen kann. Wie zumbeispiel NodeJS oder npm.

Nachdem ich das gemacht habe habe ich Den Ist soll zustand festgehalten um darzustellen was aktuell gegeben ist und was ich mit dem Projekt vorhabe.

Als meine Recerce abgeschlossen war habe ich ein UseCase Diagramm erstellt um darzustellen wie ich Plane mein Pogramm umzusetzen und mich bei der Entwicklung daran zu orientieren.

Dadurch war meine Planung abgeschlossen und ich habe angefangen einen Simplen Webserver zu bauen der eine anfrage an Google Gemini Stellt und eine Antwort erwartet. Das das Backend die anfrage ab Google Stellt ist daher empfählenswert da ich einen Key benötige und dieser Key sollte nicht einfach einsehbar sein. Im Web kann man mithilfe der Entwicklertool den Key herausfinden was bei dem backendserver nicht geht. Außerdem gibt es ein extra Package von google für Python was es mir erleichter die Google Gemini api anzusprechen und mein ergebnis zu erhalten. Als mir das geplückt ist habe ich Das Web gebaut dafür habe ich den Webserver so angepasst das wenn die URL angesurft wird meine Generierte Index.html ausgegeben wird. Danach musste ich noch die Public folder des Projektes freigeben damit Images und das Favicon geladen werden können und die anfrage nicht vom Server geblockt wird.

Als eine einfache Weboberläche in meinem python Server lief habe ich das Design der Seite gebaut, dafür habe ich die Komponeten von Quasar verwenden wie beispielsweise den Loader oder die Buttons und eingabefelder. Als das Design fertig war habe ich die anfrage an meinen Server gebaut um diese dort an Google Gemini zu schicken und die Antwort wieder auf der Seite auszugeben. Dann habe ich ausprobiert was der oplimalste Prompt sein kann um gute Antworten von meinem Model zu bekommen. Und ich habe auch die Model ausprobiert da die verschiedenen Models unterschiedliche Antworten geben. Ich habe mich für Gemini 1.5 Pro entschieden da es das Akutuell best entwickelste Model von Google ist und die Antworten im algemeinem am besten waren.

### Technische Details

Eine Python Version wird benötigt das Projekt wurde mit der Version 3.12 entwickelt. Das Package googel-ai-generativelanguage in der Version 0.6.10, flask und flask\_cors in der jewals neusten Version. Für das Frontend wird eine NodeJS auf der Version 20 und npm installation benötigt.

### Erklärung einzelner verwendeter Packages oder Frameworks

#### Quasar

Quasar ist ein Framework das dem Entwickler bei der Benutzeroberfläche (UI) hilft und verschiedene Komponenten bereitstellt um eine bessere und schönere UI zu erstellen. Es bietet eine einfache Möglichkeit ein Response Design zu entwerfen und eine Riesige Komponentenbiblioteck. Quasar basiert auf vue

#### Vue

Vue.js ist ein JavaScript-Framework, das den Aufbau von Benutzeroberflächen erheblich erleichtert. Es ermöglicht Entwicklern, sogenannte Vue-Komponenten zu erstellen, die später in einem Build-Prozess zu einer vollständigen HTML-Seite zusammengefügt werden.

Ein großer Vorteil von Vue.js ist die Unterstützung von nützlichem Code, wie beispielsweise:

* v-if: Ermöglicht das dynamische Ein- oder Ausblenden von HTML-Elementen basierend auf Bedingungen.
* v-for: Erlaubt das einfache Erstellen von Schleifen im HTML, um z. B. Listen von Elementen dynamisch zu generieren.

Eine zentrale Rolle in Vue.js spielen Referenzen und Reaktivität. Mithilfe dieser Mechanismen können Entwickler flexibel auf Änderungen im Datenmodell reagieren. Dadurch ist es möglich, ohne großen Aufwand Komponenten ein- oder auszublenden, Benutzeraktionen zu verarbeiten oder die Benutzeroberfläche dynamisch zu aktualisieren.

Zusätzlich bietet Vue.js eine klare Struktur für die Entwicklung moderner Webanwendungen. Durch die Modularität der Komponenten bleibt der Code übersichtlich, und der Entwicklungsprozess wird effizienter. Für komplexere Projekte stellt Vue erweiterbare Tools wie den Vue Router (zur Navigation in Single-Page-Anwendungen) und Vuex (für Statusmanagement) bereit.

#### Node.js

Node.js basiert auf der V8-Engine von Google Chrome und zeichnet sich durch seine Geschwindigkeit und Flexibilität aus. Es unterstützt asynchrone Programmierung, wodurch mehrere Prozesse parallel ausgeführt werden können, ohne den Server zu blockieren. Dies macht es ideal für Anwendungen, die auf hohe Leistung und Skalierbarkeit ausgelegt sind, wie beispielsweise Webserver oder APIs.

Ein weiteres Merkmal von Node.js ist, dass es nicht nur für serverseitige Anwendungen verwendet werden kann, sondern auch zur Erstellung von Entwicklungswerkzeugen und Build-Systemen, was es für Entwickler äußerst vielseitig macht.

#### NPM Paketmanager

npm ist der Paketmanager von Node.js und eine zentrale Komponente der Node.js-Umgebung. Es bietet Zugriff auf eine riesige Bibliothek von Modulen und Tools, die von der Entwickler-Community bereitgestellt werden. Mit npm können Entwickler schnell und einfach Abhängigkeiten für ihre Projekte installieren und verwalten.

Ein wichtiger Vorteil von npm ist die Möglichkeit, eigene Pakete zu erstellen und zu veröffentlichen, um Code wiederverwendbar zu machen und mit anderen zu teilen. Außerdem ermöglicht es npm, Skripte für häufig genutzte Aufgaben im Entwicklungsprozess auszuführen.

### Herausforderungen und Lösungen

Einses der ersten Probleme war das ich eine Fehlerhafte Python version hatte und so meine Pogramme nicht starten konnte. Ich hatte versucht ohne die gesamte die und python neu zu instalieren nur den Interpreter upzudaten, das hat leider nicht funktioniert daher habe ich einmal alles neu aufgesetzt und das hat das Problem geholfen.

Ein Weiteres Problem war das mein Frontend nicht auf seinen Public folder zugreifen konnte und so die Images und andere assets nicht laden konnte da Vue alles in einen Public Ordner baut und der Webserver die Anfragen abgeblockt hat. Um das zu lösen musste ich in Flesh weitere Konfigurationen vorhnemen um die Ordner wirklich public zu machen.

# Abschluss

## Ergebnisbeschreibung

Ich habe einen Webserver der über Flesk eine generierte index.html bereitstellt die ich vorher generiert habe mithilfe eines Frameworkes namens Quasar wo ich die Weboberfläche für mein Pogramm gebaut habe auf der Seite kann der User seinen Text ein textfeld eingeben und dann auf einen Button klicken und der Text wird für ihn geprüft und an mein backend gesendet. Dort wird er weiter an Google Gemini gesendet, und die Antwort wird wieder an mein frontend zurückgegeben. Im Frontend wird es Grafisch dargestellt und dem Nutzer eine Möglichkeit geboten es mit einem Klick zu kopieren.

## Funktionstest

Ich habe die Software auf verschiedenen Geräten Tests unterzogen, ich habe auch versucht Fehlerhafte Texte einzugeben und zu versuchen dadurch einen Fehler zu erzeugen. In meinen Tests sind Pobleme entstanden die ich dann behoben habe, so wie das teilweise Antworten von Gemini zu Fehlerhaften darstellungen gefürt haben die ich behoben habe.

## Kundeneinweisung

Für die Installation benötigt man die Python version 3.12 und die Aktuelle Node und NPM Version. Mit npm installiert man Quasar global oder nur in dem Projekt damit man das Projekt bauen kann. Zuerst muss man aber die Abhängigkeiten installieren das macht man im root Frontend Ordner mit „npm run install“ und danach den Befehl „npm run build“. Wenn das erfolgreich war kann man den Python Server starten dafür startet man die index.py

# Fazit

## Reflektion über den Verlauf des Projektes

Die Entwicklung des Projektes verlief recht gut da ich die Ziele in der vorher festgelegten Zeit Geschaft habe, ich habe viele nützliche Informationen im Internett gefunden wie beispielsweise Stackoverflow wie ich den backendserver gestalten kann und meine Probleme lösen kann. Das Entwickeln des Frontentes war auch sehr einfach da mir das Framework viel Design und Logik Arbeit abgenommen hat. Und ich so die Arbeit an dem Frontend sehr schnell ohne Große Probleme beenden konnte. Beim verbinden von Frontend und Backend sind dann ein paar Probleme aufgetaucht die dadurch enstanden sind das ich einafch noch nicht so viel erfahrung mit Backendservern in Python habe und so ein paar Konfigurationen falsch vorgenommen habe. Aber mithilfe von Foren und Künstlicher inteligenz hatte ich das Problem nach ungefair 1 Stunde behoben.

Was aber nicht so gut war war die Vorbereitung auf das Projekt da ich erst bei der entwicklung festgestellt habe das die Python Verison 1.13 zum Start des Projektes nicht mit dem Package von Google Gemini kompatible war was blöd war da ich bei der Planung darauf hätte achten müssen als ich die Version ausgewält habe.

## Persönliche Erkenntnis und Lernerfahrung

Persönlich habe ich gelernt das ich genauer auf Versionen bei den Packages oder Frameworks achten muss die ich in meinem Projekten verwende da es unnötige Zeit kosten kann wenn man probleme bei der entwicklung hat man einfach nur die falsche version ausgewält hat.