

Dokumentation

Projekt KI-Tutor

Teammitglieder:

- Dieminger Elia
- Nestelberger Florian
- Riener Tobias
- Schippany Tobias

Hauptsächliche Dokumentation:

- Trello Board
- Text Dokument

Teammitglieder Einteilung:

HTML:

- Tobias Schippany
- Elia Dieminger

CSS:

- Florian Nestelberger

Node.js & OpenAI-API-Integration

- Tobias Riener

Projektmanagement

- Elia Dieminger

Dokumentation der Aufgaben:

HTML:

Die HTML-Struktur der „Startseite“ mit Navbar und Footer sowie die „Tutor-Seite“ wurde von Tobias Schippany konzipiert und erstellt. Elia Dieminger erstellte die „Über uns“- und „Kontakt“-Seite.

Die Startseite, die als Eyecatcher dienen soll, bietet einen Überblick darüber, welche Aufgaben der KI-Tutor übernehmen kann. Der letzte Absatz lädt zudem dazu ein, die KI kostenlos auszuprobieren.

Die Navbar dient zur einfachen Navigation zwischen den verschiedenen Seiten und wird zusammen mit dem Footer auf allen Unterseiten angezeigt. Der Footer ermöglicht den Zugriff auf das Impressum, die Datenschutzerklärung und die Nutzungsbedingungen.

Die Unterseite „KI-Tutor“ ist die wichtigste Seite der Webseite; hier können die Nutzer ihre Fragen von der KI beantworten lassen.

Informationen über das Entwicklungsteam finden sich auf der Unterseite „Über uns“, die auch kurze Beschreibungen zu den Teammitgliedern enthält.

Die „Kontakt“-Seite dient dazu, Fragen zu stellen oder Kontakt mit uns aufzunehmen. Hier können Nutzer uns Nachrichten senden, uns auf Instagram und Facebook finden oder unseren Standort erkunden.

CSS:

Der CSS-Code dient dazu, den Inhalt der Webseite ansprechend und benutzerfreundlich darzustellen. Mithilfe von CSS wird die Darstellung auch automatisch an die Bildschirmgröße und Systemfarbe des Nutzergeräts angepasst, um die Inhalte optimal anzuzeigen.

Node.js & OpenAI-API-Integration:

Die Webseite läuft auf einem Node.js-Server, der HTML, CSS und JavaScript bereitstellt. Der Wechsel zwischen den verschiedenen Unterseiten wird über das main-Element im HTML-Code realisiert; um die Seiten zu wechseln, wird dieses Element mithilfe von JavaScript ein- bzw. ausgeblendet.

Der Chatverlauf wird ebenfalls über JavaScript im Local Storage des Browsers gespeichert.

Es wurden 21 KI-Tutoren implementiert, die über ein „Selection“-Element ausgewählt werden können. Die definierten Beschreibungen der Tutoren und die Eingabe des Nutzers werden an die Chat-GPT-API übergeben. Diese liefert einen String als Antwort zurück, der auf der Webseite angezeigt wird.

Die OpenAI-API wurde mittels dem Standardbeispiel der Dokumentation implementiert. Es wurden nur kleine Änderungen vorgenommen, um für unseren Use-Case zu funktionieren.

GitHub Flow-Prozess

Branch-Struktur im GitHub-Repository

Im GitHub-Repository unseres Projekts wurde eine klare Branch-Struktur implementiert, um die Entwicklung effizient zu organisieren. Der Hauptentwicklungszweig war der „development“-Branch, von dem aus für jede Teilaufgabe spezifische Subbranches erstellt wurden.

Struktur der Branches:

HTML-Struktur:

- Ein eigener Branch namens „html-structure“ wurde erstellt, um die Grundstruktur der HTML-Dateien zu entwickeln.

- Für spezifische Unterseiten wie „Über uns“ und „Kontakt“ wurden jeweils separate Subbranches innerhalb von „html-structure“ angelegt. Nach der Fertigstellung der Seiten wurden diese in den „html-structure“-Branch gemergt.

Integration der OpenAI-API:

- Die Implementierung der OpenAI-API erfolgte in einem separaten Branch namens „openai-api“.
- Dafür wurden die entsprechenden HTML- und CSS-Dateien mithilfe des pull-Befehls in diesen Branch kopiert, was eine erfolgreiche Integration und anschließende Tests ermöglichte.
- In diesem Branch wurden auch der Server, der auf Node.js basiert, sowie sämtliche EventListener (z. B. für die automatische Anpassung an Systemfarben) entwickelt.

JavaScript-Funktionalität:

- Die vollständige Programmierung zur Navigation zwischen den Unterseiten wurde im „js“-Branch umgesetzt. Dies wurde durch dynamische Manipulation der Inhalte innerhalb einer einzigen HTML-Datei mittels JavaScript realisiert.

CSS-Styling:

- Für das Design wurde ein separater Branch namens „css-style“ erstellt, welcher ebenfalls ein Subbranch von „development“ ist.
- In diesem Branch wurde die komplette „style.css“ entwickelt. Um Konsistenz zu gewährleisten, wurden die entsprechenden HTML- und JavaScript-Dateien ebenfalls durch pull integriert.