

Pflichtenheft für das KI-Tutor-Web-Projekt

1. Projektübersicht:

Ziel des Projekts:

Entwicklung einer interaktiven Website, die einen Tutor über die OpenAI-API integriert. Die Website soll Schülern ermöglichen, in Teams zu arbeiten, sich selbst zu organisieren und eine funktionale Webanwendung zu entwickeln.

Technische Anforderungen:

- Frontend: HTML, CSS, JavaScript.
- Backend: Node.js, OpenAI-API-Integration.
- Versionsverwaltung: GitHub, Feature Branching (GitHub Flow).
- Projektmanagement: Trello mit User Stories.
- **Teamgröße:** 4 Schüler pro Team.

2. Seitenübersicht und Anforderungen

2.1. Startseite (Landingpage)

Pflicht:

- **Header:**
- Logo (links).
- Navigation mit Links zu „Tutor“, „Wie funktioniert’s?“, „Über uns“, „Kontakt“.
- **Hero-Sektion:**
- Überschrift: „Dein persönlicher KI-Tutor für alle Fragen“.
- Unterüberschrift: Kurze Erklärung, was der Tutor bietet, z. B. „Erhalte in Sekunden Antworten auf deine Fragen zu Mathematik, Informatik und mehr“.
- Call-to-Action (CTA)-Button: „Jetzt starten“ (führt zur Tutor-Seite).
- Animiertes Element (JavaScript): z. B. Partikelwolke oder animierte Sprechblasen, die auf Mausbewegungen reagieren.

- **Vorteile:**
- Drei Bereiche mit Icons und kurzen Beschreibungen (z. B. „Schnelle Antworten“, „Lerne jederzeit“, „Individuelle Unterstützung“).
- **Footer:**
- Links zu „Datenschutz“, „Nutzungsbedingungen“, und einem kurzen Projektbeschreibungstext.

Optional:

- Testimonials: Feedback von fiktiven Nutzern.
- Video- oder Bildanimation zur Nutzung des Tutors.

2.2. Tutor-Seite

Pflicht:

- **Header:** Konsistent mit der Startseite.
- **Hauptbereich:**
- Eingabefeld für Fragen und „Senden“-Button.
- Anzeige der Konversation mit dem Tutor (Sprechblasenstil).
- Scrollbare Anzeige für den Verlauf der Konversation.
- Hinweise zur Nutzung des Tutors (z. B. „Stellen Sie spezifische Fragen, um bessere Antworten zu erhalten“).
- **Footer:** Konsistent mit der Startseite.

Optional:

- Anpassbare Themenbereiche: Auswahlmenü, um den Fachbereich des Tutors zu ändern (z. B. „Mathematik“, „Programmierung“).
- Ladeanimation während der Verarbeitung der Anfrage.

2.3. Wie funktioniert's? (Erklärungsseite)

Pflicht:

- **Header:** Konsistent mit der Startseite.
- **Hauptbereich:**
- Schritt-für-Schritt-Anleitung zur Nutzung des Tutors (mindestens 3 Schritte).
- Je Schritt eine einfache Grafik oder Illustration.
- FAQ-Bereich mit mindestens 3 häufig gestellten Fragen (z. B. „Wie stelle ich Fragen an den Tutor?“).
- **Footer:** Konsistent mit der Startseite.

Optional:

- Video-Tutorial zur Nutzung des Tutors.
- Interaktive Demo, die zeigt, wie man Fragen stellt.

2.4. Über uns (Teamseite)

Pflicht:

- **Header:** Konsistent mit der Startseite.
- **Hauptbereich:**
- Vorstellung der Teammitglieder (je ein Kasten mit Bild, Name, Rolle im Projekt, Kurzbeschreibung).
- Projektbeschreibung: Ziele und Lernziele des Projekts.

Footer: Konsistent mit der Startseite.

- **Optional:**
- Teamfoto.
- Links zu den GitHub-Profilen der Teammitglieder.

2.5. Kontaktseite

Pflicht:

- **Header:** Konsistent mit der Startseite.
- **Hauptbereich:**
- Kontaktformular mit Feldern für Name, E-Mail, Betreff und Nachricht.
- „Senden“-Button.
- Bereich für alternative Kontaktinformationen (z. B. E-Mail-Adresse des Projekts).
- **Footer:** Konsistent mit der Startseite.

Optional:

- Social Media-Links (fiktive Links zu Plattformen).
- Map-Integration (z. B. Standort der Schule).

3. Designanforderungen:

- **Design-Stil:** Modern und minimalistisch, mit klaren Linien und einer freundlichen Farbpalette (z. B. Blau- und Grautöne mit Akzenten in Grün oder Orange).
- **Responsive Design:** Die Webseite muss auf verschiedenen Geräten (Desktop, Tablet, Smartphone) gut nutzbar sein.
- **Icons und Illustrationen:** Verwendung von SVG-Grafiken für schnelle Ladezeiten.
- **Schriftarten:** Verwendung von gut lesbaren Webfonts (z. B. Roboto, Open Sans).

4. Technische Anforderungen:

Versionskontrolle:

- Nutzung von GitHub mit GitHub Flow:
- Jeder Schüler arbeitet in einem eigenen Branch.
- Änderungen werden regelmäßig gepusht und durch Pull-Requests in den Main-Branch gemergt.
- Feature Branches werden für die Entwicklung neuer Funktionen verwendet.

API-Integration:

- Einbindung der OpenAI-API in das Node.js-Backend zur Verarbeitung der Fragen und Rückgabe der Antworten.

Projektdokumentation:

- Dokumentation der API-Integration und der einzelnen Seiten.
- README-Datei mit Installationsanweisungen und Nutzungshinweisen auf GitHub.

5. Projektdokumentation und Abgabe:

Projektdokumentation:

- Jede Gruppe dokumentiert ihre Arbeitsschritte im Trello-Board.
- User Stories: Aus den Anforderungen abgeleitete User Stories (z. B. „Als Nutzer möchte ich eine Frage stellen können, damit ich eine Antwort vom Tutor erhalte“).

- Sprintplanung im Trello-Board: Aufgaben werden in To-Do, In Progress und Done unterteilt.

Abgabe:

- Ein vollständig funktionierendes GitHub-Repository.
- Dokumentation der API-Integration, Nutzung der GitHub Flow-Prozesse und der Aufgabenaufteilung im Team.