**Dokumentation:  
GeoSurfer**

Inhaltsverzeichnis:

[Projektantrag fürs IMS-Lernatelier: 3](#_Toc148367987)

[Meine persönlichen Ziele in diesem Projekt 3](#_Toc148367988)

[Betroffene Informatik-Module 3](#_Toc148367989)

[Projektbeschrieb 3](#_Toc148367990)

[Projektteam 3](#_Toc148367991)

[Projektmethode 3](#_Toc148367992)

[Informationsphase: 3](#_Toc148367993)

[Quellen: 3](#_Toc148367994)

[Anforderungen: 4](#_Toc148367995)

[Planen: 4](#_Toc148367996)

[Entscheiden: 5](#_Toc148367997)

[Realisieren: 5](#_Toc148367998)

[Kontrollieren: 5](#_Toc148367999)

[Testprotokoll: 6](#_Toc148368000)

[Auswerten: 6](#_Toc148368001)

[Fazit: 6](#_Toc148368002)

# Projektantrag fürs IMS-Lernatelier:

Name, Vorname: Neiger, Finn

Datum: 15.08.2023

## Meine persönlichen Ziele in diesem Projekt

*Responsives Design: Modernes, schlichtes Design sicherstellen, das auf verschiedenen Geräten reaktionsschnell ist.*

*React SPA: Entwicklung einer reaktiven Single Page Application (SPA) mit nahtloser Navigation zwischen Funktionen und Informationen.*

*API-Nutzung: Fehlerfreie Integration der Countries Cities API für präzise Datenanzeige.*

*GitHub-Repository: Hochladen des gesamten Projektcodes auf GitHub mit klarer Historie und organisierter Struktur.*

*Benutzerdokumentation: Erstellung einer intuitiven Benutzerdokumentation für das einfache Verstehen des Projektablaufs.*

*Qualitätssicherung: Umfassende Tests, Fehleridentifizierung und -behebung vor der endgültigen Veröffentlichung.*

## Betroffene Informatik-Module

*Modul 324: DevOps-Prozesse mit Tools unterstützen*

## Projektbeschrieb

Geo Surfer ist eine React-Applikation, welche die [Countries Cities API](https://rapidapi.com/natkapral/api/countries-cities/) verwendet. Über diese Applikation sollten die Benutzer die verschiedensten Informationen über Städte und Länder abrufen und lesen können, welche sich in der Nähe befinden. Zu den ausgegebenen Informationen gehören Infos über die Zeitzone, Sprache, Währung, Population, Breite, Länge und etc. Die Applikation sollte responsive sein und React benutzen. Das Design der App soll modern und eher schlicht gehalten sein, um übersichtlich zu bleiben. Die ganze Applikation sollte als eine SPA (Single Page Application) funktionieren. Das ganze Projekte sollten dann ordnungsgemäss auf GitHub hochgeladen werden.

## Projektteam

*Teammitglieder: Vin Appenzeller, Finn Neiger*

*Ablage Code & Dokumentation:* [*GitHub*](https://github.com/Kappa-X/Geo-Surfer)

## Projektmethode

*Wir benutzen für unser Projekt die IPERKA Projektmethode, welche wir auch schon in vorherigen Lernatelier Projekten verwendet haben.*

# Informationsphase:

## Quellen:

GitHub: <https://github.com/Kappa-X/Geo-Surfer>

Einführung in ReactJs: <https://www.youtube.com/watch?v=qJqjcxLvEwg&t=366s>

Quelle für Bild auf Deckblatt: <https://pixabay.com/de/illustrations/erde-welt-planet-globus-1303628/>

## Anforderungen:

|  |  |  |
| --- | --- | --- |
| Nr. | Anforderung | Minimal oder nice to have |
| 1 | Responsives Design: Modernes, schlichtes Design sicherstellen, das auf verschiedenen Geräten reaktionsschnell ist | minimal |
| 2 | React SPA: Entwicklung einer reaktiven Single Page Application (SPA) mit nahtloser Navigation zwischen Funktionen und Informationen. | minimal |
| 3 | API-Nutzung: Fehlerfreie Integration der Countries Cities API für präzise Datenanzeige. | minimal |
| 4 | GitHub-Repository: Hochladen des gesamten Projektcodes auf GitHub mit klarer Historie und organisierter Struktur. | minimal |
| 5 | Qualitätssicherung: Umfassende Tests, Fehleridentifizierung und -behebung vor der endgültigen Veröffentlichung. | minimal |
| 6 | Benutzerdokumentation: Erstellung einer intuitiven Benutzerdokumentation für das einfache Verstehen des Projektablaufs. | Nice to have |

# Planen:

|  |  |  |
| --- | --- | --- |
| Arbeitspaket | Datum | Gruppenmitglied |
| Dokumentation Grundaufbau erstellen | 21.08.2023 | Finn Neiger |
| Über ReactJs informieren und Tutorials schauen | 21.08.2023 | Finn Neiger, Vin Appenzeller |
| GitHub einrichten | 28.08.2023 | Finn Neiger |
| Erstellen des Grundgerüst von GeoSurfer mit ReactJs | 28.08.2023 | Vin Appenzeller |
| Einfügen der NavigationBar mit ihren Funktionen | 28.08.2023 | Finn Neiger |
| Erstellen der Start Section | 04.09.2023 | Vin Appenzeller |
| Erstellen der About Section | 04.09.2023 | Finn Neiger |
| Erstellen des Footers | 04.09.2023 | Finn Neiger |
| Hinzufügen der API-Funktionen | 11.09.2023 - 17.10.2023 | Vin Appenzeller, zum Teil Finn Neiger |
| Ausfüllen der Dokumentation | 11.09.2023 | Finn Neiger |
| Start der Testen-Phase | 11.09.2023 | Finn Neiger |
| Beenden der Testen-Phase | 17.10.2023 | Finn Neiger, Vin Appenzeller |
| Portfolio schreiben | 17.10.2023 | Finn Neiger, Vin Appenzeller |

# Entscheiden:

Keine neuen wichtigen Entscheidungen wurden nach dem Stellen des Projektantrags gemacht.

# Realisieren:

Die fertigen Produkte sind auf GitHub aufzufinden.

# Kontrollieren:

|  |  |  |  |
| --- | --- | --- | --- |
| Anforderungs-Nr. | Voraussetzung | Eingabe | Erwartete Ausgabe |
| 1 | Die Anwendung ist auf einem Desktop-Browser geöffnet. | Der Benutzer besucht die Website | Die Website wird modern und ansprechend auf dem Desktop-Browser angezeigt |
| 1 | Die Anwendung ist auf einem Tablet geöffnet. | Der Benutzer besucht die Website auf einem Tablet. | Die Benutzeroberfläche passt sich der Bildschirmgröße an und ist gut lesbar. |
| 1 | Die Anwendung ist auf einem Smartphone geöffnet. | Der Benutzer besucht die Website auf einem Smartphone. | Alle Elemente sind gut sichtbar und leicht navigierbar auf dem Smartphone. |
| 2 | Die Anwendung ist geöffnet. | Die Anwendung ruft Länderdaten von der API ab. | Die Navigation erfolgt nahtlos und ohne Verzögerung. |
| 3 | Die Anwendung ist aktiv ist in einem Browser geöffnet. | Die Anwendung ruft Länderdaten von der API ab. | Die angezeigten Daten entsprechenden Daten in der API, und sie werden präzise und fehlerfrei angezeigt. |
| 3 | Die Anwendung ist aktiv, aber die API ist nicht erreichbar. | Die Anwendung versucht, Daten von der API abzurufen. | Die Anwendung zeigt keine Fehler an und stürzt nicht ab. |
| 4 | Der Projektcode ist lokal vorhanden und bereit für den Upload. | Der Entwickler lädt den Projektcode auf GitHub hoch. | Code kann einfach von GitHub aus abgerufen werden. |
| 5 | Anwendung ist im Browser geöffnet | Die verschiedenen vorherigen Testfälle (ohne 4.1) werden überprüft | Alles funktioniert einwandfrei und kann für die Veröffentlichung freigegeben werden. |

## Testprotokoll:

Testumgebung: Laptop mit Windows 11 und dem OperaGX Browser

|  |  |  |
| --- | --- | --- |
| Testfall Nr. | Durchgeführt von | OK/NOK |
| 1.1 | Finn Neiger | OK |
| 1.2 | Finn Neiger | OK |
| 1.3 | Finn Neiger | OK |
| 2.1 | Finn Neiger | OK |
| 3.1 | Finn Neiger | OK |
| 3.2 | Finn Neiger | OK |
| 4.1 | Finn Neiger | OK |
| 5.1 | Finn Neiger | OK |

# Auswerten:

## Fazit:

Gut gelaufen: Wir haben uns sehr schnell an ReactJS gewöhnt, was uns erlaubt hat nach einer intensiven Informationsphase mit diesem Projekt anzufangen, obwohl wir zuvor noch nie mit ReactJS gearbeitet haben. Sonst funktionierte auch die Zusammenarbeit im Team sehr gut, da wir mit zwei Leuten eine kleine Gruppe waren und uns so besser absprechen konnten.

Schlecht gelaufen: Leider war unsere Zeitplanung eher grosszügig, wodurch wir ein paar Abstriche im Design der Webseite machen musste, um das Projekt in der gewünschten Zeit zu beenden können. Das Problem war vor allem, dass wir das Integrieren der API massiv unterschätzt hatten und deshalb gegen Schluss sehr viel Zeit verloren hatten. Auch war das ganze Projekt für zwei Personen von Anfang an eher zu gross und wir hatten uns überschätzt