

LA projekt 3 dokumentation

Wetter - WebSockets



30. April 2024

Herrmann, dal corso, hartmann

Berufsfachschule BBBaden

Inhalt

[Projektantrag 2](#_Toc165386323)

[Informieren 3](#_Toc165386324)

[Anforderungen 3](#_Toc165386325)

[Testfälle 3](#_Toc165386326)

[Planen 4](#_Toc165386327)

[ToDo-Liste 4](#_Toc165386328)

[Entscheiden 5](#_Toc165386329)

[Kontrollieren 5](#_Toc165386330)

[Testfallprotokoll 5](#_Toc165386331)

[Fazit 5](#_Toc165386332)

[Auswerten 5](#_Toc165386333)

[Reflexion 5](#_Toc165386334)

[Quellen 5](#_Toc165386335)

# Projektantrag

|  |  |  |
| --- | --- | --- |
| **Projektantrag zu LA\_ILA3\_0140** | | |
| **Projektteam (1. Zeile = Projektleitung)** | | |
| ***Name*** | ***Vorname*** | ***Klasse*** |
| **Dal Corso** | **Luca** | **Im21c** |
| Hartmann | Dominik | Im21c |
| Herrmann | Michael | Im21c |
| **Projektidee / Produkt** | | |
| *Beschreiben Sie, was Sie erstellen werden. Sie können, aber müssen hier noch keine Technologien festlegen. Der Umfang des Projekts sollte klar werden. Beschreiben Sie die Funktionen allgemein.*  Eine kleine Wetterapplikation, die in echt Zeit verschieden Daten von einer gesuchten Stadt zeigen, wie zum Beispiel die Temperatur oder die Feuchtigkeit in der Stadt. | | |
| **Betroffene Informatik-Module** | | |
| 294/295 | | |
| **Infrastruktur** | | |
| *Wie organisieren Sie Ihre Zusammenarbeit? Wo ist die Ablage für den Code? Wo wird die Dokumentation abgelegt?*  Code und Dokumentation wird auf Github abgelegt. Zusammenarbeit über Discord | | |
| **Projektmethode** | | |
| *Benennen Sie die Projektmethode, die Sie verwenden werden. Dies muss sich in der Projektdokumentation wiedererkennen lassen.*  Wir werden die Projektmethode IPERKA verwenden. | | |
| **Grobplanung** | | |
| *Beschreiben Sie hier in groben Zügen 5 bis 10 Arbeitspakete (ca. 1 Arbeitspaket je Person und Arbeitstag). Definieren Sie 2 bis 4 Meilensteine (mit Termin).*  Kann nach Orten suchen Wird in Echtzeit updatet Wetterdaten werden auf einer Webseite angezeigt Api Wetterdaten abspeichern von den Städten  Abgespeicherten Daten mit Websockets versenden.  26.3.2024 Wetterdaten werden auf der Webseite angezeigt  02.4.2024 Projekt Fertig | | |
| **Persönliche Ziele** | | |
| Dal Corso, Luca  Ziel 1: Die Daten in Echtzeit zu akutalisieren Ziel 2: Das man nach verschieden Orten die Richtigen Daten erscheinen Ziel 3: Das ich ein Teil des Projektes mit Websockets erledigen kann. | | |
| Herrmann, Michael  Ziel 1: WebSockets in einem konkreten Beispiel umsetzen Ziel 2: Auf einer Website-Session Daten abspeichern Ziel 3: Ein Node.JS Backend erstellen | | |
| Hartmann, Dominik  Ziel 1: Die Wetterdaten der jeweiligen Stadt richtig ausgeben Ziel 2: WebSockets in dem Programm einbauen Ziel 3: Die Wetterdaten von der API aufrufen und verwenden | | |

# Informieren

## Anforderungen

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Muss / Kann?** | **Funktional? Qualität? Rand?** | **Beschreibung** |
| 1 | Muss | **Funktional** | Der Benutzer soll nach einer Stadt für das Wetter suchen |
| 2 | Muss | **Qualität** | Die Wetterdaten werden mithilfe eines WebSockets von unserem Backend gesendet |
| 3 | Muss | **Funktional** | Das WebSocket soll jede Minute alle Wetterdaten aktualisieren |
| 4 | Kann | **Qualität** | Listen von Städten können im Backend abgespeichert werden |
| 5 | Kann | **Funktional** | Benutzer können angelegt werden mit Username und Password |
| 6 | Kann | **Qualität** | Die Listen werden in einer Datenbank abgespeichert |
| 8 | Muss | **Randbedingungen** | Das Backend ist mit Node.js erstellt |
| 9 | Muss | **Randbedingungen** | Es wird die NPM Library ws für WebSockets verwendet |

## Testfälle

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Testfall Nr.** | **Anf.- Nr.** | **Vorrausetzung** | **Eingabe** | **Ausgabe** |
| 1.1 | 1 | Programm gestartet | 1 Aarau  3 Plus Klicken | 2 Wetterdaten von Abspeichern (Plus Button)  4 Daten abgespeichert. |
| 2.1 | 3 | Programm gestartet  Und Stadt ausgewählt | 1 eine Minute gewartet | 2 Wetterdaten der Stadt updated. |
| 4.1 | 5 | Programm gestartet | 1 Klickt auf Registrieren  3 Username  und Password ein | 2 Input Fenster  4 Daten abgespeichert  5 Kann nach Stadt suchen |
| 4.2 | 5 | Programm gestartet  Und Testfall 4.1 | 1 Klickt auf Login  3 Username  Und Password ein | 2 Input Fenster  4 Kann nach Stadt suchen |

# Planen

## ToDo-Liste

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Herrmann** | **Hartmann** | **Dal Corso** |
| 12.03.2024 |  |  |  |
| Lektion 1 | Projektantrag | Projektantrag | Projektantrag |
| Lektion 2 | Projektantrag | Projektantrag | Projektantrag |
| Lektion 3 | Anforderung | Anforderung | Anforderung |
| Lektion 4 | Testfall | Testfall | Testfall |
| Lektion 5 | Planen | Planen | Planen |
| 19.03.2024 |  |  |  |
| Lektion 1 | Informieren | Informieren | Informieren |
| Lektion 2 | Informieren | Informieren | Informieren |
| Lektion 3 | Backend | GUI erstellen | Pair-Programming |
| Lektion 4 | Backend | GUI erstellen | Pair-Programming |
| Lektion 5 | Backend | GUI erstellen | Pair-Programming |
| 26.03.2024 |  |  |  |
| Lektion 1 | Backend mit Frontend verbinden | Pair-Programming | Pair-Programming |
| Lektion 2 | Logik implementieren | Logik implementieren | Logik implementieren |
| Lektion 3 | Logik implementieren | Logik implementieren | Logik implementieren |
| Lektion 4 | Logik implementieren | Logik implementieren | Logik implementieren |
| Lektion 5 | Hilfestellung | Dokumentation | Projektbericht |
| 02.04.2024 |  |  |  |
| Lektion 1 | Pair-Programming | Pair-Programming | Daten speichern |
| Lektion 2 | Pair-Programming | Pair-Programming | Daten speichern |
| Lektion 3 | Besprechung weitergehen des Projektes | Besprechung weitergehen des Projektes | Besprechung weitergehen des Projektes |
| Lektion 4 | GUI-Design | Testen | Dokumentation |
| Lektion 5 | GUI-Design | Dokumentation | Pair-Programming |
| 30.04.2024 |  |  |  |
| Lektion 1 | Portfolio schreiben | Portfolio schreiben | Portfolio schreiben |
| Lektion 2 | Portfolio schreiben | Portfolio schreiben | Portfolio schreiben |
| Lektion 3 | Portfolio schreiben | Portfolio schreiben | Portfolio schreiben |
| Lektion 4 | Portfolio schreiben | Portfolio schreiben | Portfolio schreiben |
| Lektion 5 | Portfolio schreiben | Portfolio schreiben | Portfolio schreiben |

# Entscheiden

Wir haben uns dafür entschieden, dass wir kein Login erstellen.

Wir haben uns dafür entschieden, dass wir unsere Webseite hellblau machen, da blau gut zum Wetter passt. Deshalb haben wir uns anschliessend entschieden, dass wir ein wolkiger Hintergrund für unsere Web-App verwenden.

# Kontrollieren

## Testfallprotokoll

|  |  |  |  |
| --- | --- | --- | --- |
| **Testfall-Nr.** | **Bemerkung** | **Bestanden (OK/NOK)** | **Person** |
| 1.1 | - | OK | Dal Corso |
| 2.1 | - | OK | Dal Corso |
| 4.1 | Kann-Anforderung haben wir nicht umgesetzt | NOK | Dal Corso |
| 4.2 | Kann-Anforderung haben wir nicht umgesetzt | NOK | Dal Corso |

## Fazit

Unser Produkt funktioniert. Unsere Kann-Anforderungen haben wir zum Teil umgesetzt, aber das Login nicht.

# Auswerten

## Reflexion

Siehe Portfolio

# Quellen

* <https://www.youtube.com/watch?v=FduLSXEHLng>
* <https://stackoverflow.com/questions/10211145/getting-current-date-and-time-in-javascript>
* <https://open-meteo.com/>
* <https://open-meteo.com/en/docs/geocoding-api#name=z%C3%BCrich>