Lernatelier: Projektdokumentation

Martic, Tanaskovic

# Informieren

## Ihr Projekt

Eine Webapplikation, in welcher man das heutige Datum mit einem frei eingegebenem vergleicht.

## Quellen

<https://learn.microsoft.com/en-us/aspnet/core/fundamentals/static-files?view=aspnetcore-7.0>

<https://www.tutorialsteacher.com/core/aspnet-core-wwwroot>

1.3. Anforderungen

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Muss / Kann?** | **Funktional? Qualität? Rand?** | **Beschreibung** |
|  | Muss | Rand | In Visual Studio implementiert |
|  | Muss | Funktional | Der Nutzer kann ein Datum eingeben |
|  | Muss | Funktional | Der Nutzer bekommt die Differenz vom eingegebenen Datum im Vergleich zum Heutigem zurück. |
|  | Muss | Funktional | Die Rechnung wird mithilfe einer API ausgeführt |
|  | Muss | Funktional | Es gibt eine kleine Webseite als Frontend |

1.4 Testfälle

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nummer** | **Anforderung** | **Voraussetzung** | **Eingabe** | **Erwartete Ausgabe** |
| 1.1 | 2 | Programm gestartet | Eingabe eines Datums im Textfeld | Eingegebenes Datum wird angezeigt |
| 2.1 | 3 | " | Eingabe eines Datums im Textfeld Klick auf Calculate | Die Differenz vom Heutigem Datum und dem eingegebenen Datum wird angezeigt. |
| 3.1 | 4 | Programm gestartet  Postman benutzen | Neue Get Request erstellen mit dieser URL https://localhost:7075/Date/datedifference?inputDate=BeliebigesDatum | Die Differenz vom Heutigem Datum und dem eingegebenen Datum wird ausgegeben. |
| 4.1 | 5 | Projekt offen | Programm starten | Index.html öffnet sich |

1. Planen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nummer** | **Frist** | **Beschreibung** | **Zeit (geplant)** | **Zuständiger** |
| **1.1** | **31.05.2023** | **Projektdokumentation schreiben** | **4h** | **Igor & David** |
| **2.1** | **07.06.2023** | **API erstellen im Swagger** | **1h** | **David & Igor** |
| **3.1** | **07.06.2023** | **Öffnen eines Htmls in einem C# projekt** | **2h** | **Igor & David** |
| **4.1** | **07.06.2023** | **JavaScript Funktionalität hinzufügen** | **1h** | **David & Igor** |
| **5.1** | **07.06.2023** | **Realisieren nachtragen** | **15min** | **Igor & David** |
| **6.1** | **14.06.2023** | **Ausschmücken des CSS und HTMLs** | **2h (1h pro Person)** | **Igor & David** |
| **7.1** | **14.06.2023** | **Testen und Reflexion schreiben** | **30min** | **Igor & David** |
| **8.1** | **14.06.2023** | **Portfolio schreiben** | **1h** | **Igor & David** |

Kommentar: Pair-Programming für das gesamte Projekt. Erst genannte Person programmiert, die zweite überprüft.

1. Realisieren

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nummer** | **Frist** | **Beschreibung** | **Zeit (geplant)** | **Zeit (effektiv)** |
| **1.1** | **31.05.2023** | **Projektdokumentation schreiben** | **4h** | 4h |
| **2.1** | **07.06.2023** | **API erstellen im Swagger** | **1h** | 1h |
| **3.1** | **07.06.2023** | **Öffnen eines Htmls in einem C# projekt** | **2h** | 2h |
| **4.1** | **07.06.2023** | **JavaScript Funktionalität hinzufügen** | **1h** | 1h |
| **5.1** | **07.06.2023** | **Realisieren nachtragen** | **15min** | 15min |
| **6.1** | **14.06.2023** | **Ausschmücken des CSS und HTMLs** | **2h (1h pro Person)** | 2h |
| **7.1** | **14.06.2023** | **Testen und Auswerten schreiben** | **30min** | 30min |
| **8.1** | **14.06.2023** | **Portfolio schreiben** | **1h** | 1h |

1. Kontrollieren

## Testprotokoll

|  |  |  |  |
| --- | --- | --- | --- |
| **Testumgebung:** | **Laptop, Postman, Google Chrome** | | |
| **Nummer** | **Datum** | **Resultat** | **Durchgeführt** |
| 1.1 | **14.06.2023** | **OK** | David |
| 2.1 | **14.06.2023** | **OK** | David |
| 3.1 | **14.06.2023** | **OK** | Igor |
| 4.1 | **14.06.2023** | **OK** | Igor |

Fazit

Alle Testfälle verliefen wie geplant. Das Projekt ist demnach abgabebereit.

1. Auswerten

Unsere Vorgehensweise war diesmal gut organisiert. Wir haben die Funktionalitäten die wichtig sind zuerst implementiert und danach alles, was optional ist hinzugefügt (Design, usw.).

Da das unser letztes Projekt war und dieses ein recht kleines war, haben wir teilweise getrödelt. Ideal ist das nicht, denn wir müssen als Informatiker zu jeder Zeit konzentriert an unsere Aufgaben herangehen. Bei jedem künftigen Projekt ist es vom Vorteil, sich kurz zu sammeln und Gas zu geben. Wenn man danach noch Restzeit hat dann kann man sich in dieser Entspannen.