Modul 151 - Datenbanken in Webauftritte einbinden

20.06.206 – Felix Saaro

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Datum** | **Autor** | **Änderung** |
| V 0.1 | 20.06.2016 | Felix Saaro | Erstellung |
| V0.2 | 27.06.2016 | Felix Saaro | Erweiterung |
| V1.0 | 28.06.2016 | Felix Saaro | Fertigstellung |
|  |  |  |  |

Inhalt

[Vorwort 3](#_Toc454874536)

[Vorgaben 3](#_Toc454874537)

[Beschreibung 3](#_Toc454874538)

[Anforderungen 3](#_Toc454874539)

[Abgrenzung 4](#_Toc454874540)

[Ordnerstruktur 4](#_Toc454874541)

[Mockup 5](#_Toc454874542)

[Standard 5](#_Toc454874543)

[UseCase Diagramm 9](#_Toc454874544)

[Planung 9](#_Toc454874545)

[Datenbank 9](#_Toc454874546)

[Installation 11](#_Toc454874547)

[Anforderungen 11](#_Toc454874548)

[Konfiguration 11](#_Toc454874549)

[Login Daten 11](#_Toc454874550)

[Datenbankinstallation 11](#_Toc454874551)

[Test Daten 11](#_Toc454874552)

[Testplan 12](#_Toc454874553)

[Testfälle 12](#_Toc454874554)

[Abweichungen 13](#_Toc454874555)

[Fazit 13](#_Toc454874556)

[Abbildungsverzeichnis 14](#_Toc454874557)

# Vorwort

Das Ziel dieses Projektes ist es, eine funktionsfähige Applikation mit PHP umzusetzen. Die Applikation muss auf dem MVC Prinzip basieren und die Daten werden in einer Datenbank gespeichert.

# Vorgaben

## Beschreibung

Das Ziel ist es eine Applikation zu erstellen, in welcher jeder Benutzer Texte oder Bilder posten kann. Alle geposteten Texte und Bilder werden auf einer Timeline angezeigt. Jeder Benutzer besitz ein persönliches Profil, auf dem Profil sind alle Posts des ausgewählten Benutzers sichtbar.

Die Benutzer können von einem Administratorprofil bearbeitet und gelöscht werden. Um ein neuen Benutzer zu erstellen muss dieser sich registrieren. Benutzer können durch einen Administrator gesperrt werden. Der Administrator kann alle Posts löschen und bearbeiten.

## Anforderungen

|  |  |
| --- | --- |
| Anforderung | Status |
| Benutzer kann sich registrieren | DONE |
| Benutzer kann Account löschen | DONE |
| Benutzer kann Account bearbeiten | DONE |
| Benutzer kann Artikel erstellen | DONE |
| Benutzer kann Artikel mit Bild erstellen | DONE |
| Benutzer kann Artikel bearbeiten | DONE |
| Benutzer sieht Artikel von allen anderen Benutzer | DONE |
| Administrator kann alle Artikel sehen | DONE |
| Administrator kann alle Artikel löschen | DONE |
| Administrator kann alle Artikel bearbeiten | DONE |
| Administrator kann Benutzer löschen | DONE |
| Administrator kann Benutzer bearbeiten | DONE |
| Artikel enthält Titel | DONE |
| Artikel enthält Text und oder Bild | DONE |
| Artikel enthält Datum | DONE |
| Artikel enthält Benutzernamen des Erstellers | DONE |

## Abgrenzung

Als Erweiterung würde ich die Applikation mit der Framework ReactJs umsetzen dies beinhaltet folgende Punkte:

* Dynamisches Laden der Seiten (one-page-application)
* Einfachere Handhabung mit Elementen auf der Seite (Artikel laden automatisch wenn ein neuer Artikel in der Datenbank ist)

Folgende Funktionen sind als Ausbauschritte zu betrachten:

* Punkte verteile (Jeder Benutzer kann pro Artikel +1 oder -1 Punkt verteilen)
* Jeder Benutzer kann die Profile von den anderen Benutzern sehen
* Verwendung von AJAX

## Ordnerstruktur

Die Ordnerstruktur sieht wie folgt aus:

/ThisIsMe

| /app

| | /controller

| | /model

| | /view

| /lib

| /public

| | /css

| | /fonts

| | /img

| | | /upload

| | | /icons

| | | /bg

| | /js

| /template

| /testdaten

## Mockup

## Standard



Abbildung : Standard Layout

Alle nachfolgenden Mockups sind innerhalb des Container.



3

2

1

Abbildung : Login Formular

|  |  |
| --- | --- |
| 1 | Eingabe des Benutzernamens |
| 2 | Eingabe des Passworts |
| 3 | Login Daten Überprüfung starten |



7

5

6

3

2

4

1

Abbildung : Registration Formular

|  |  |
| --- | --- |
| 1 | Eingabe des Benutzernamens |
| 2 | Eingabe Vorname |
| 3 | Eingabe Nachname |
| 4 | Eingabe Email |
| 5 | Eingabe Passwort |
| 6 | Passwort wiederholen |
| 7 | Formular abschicken |

4

2

1



7

6

5

3

Abbildung : Artikel

|  |  |
| --- | --- |
| 1 | Profilbild des Benutzers, welcher den Artikel erstellt hat |
| 2 | Titel des Artikels |
| 3 | Name des Benutzers, welcher den Artikel erstellt hat |
| 4 | Bearbeiten Button (Kann nur vom Benutzer, welcher den Artikel erstellt hat, benutzt werden oder einem Administrator) |
| 5 | Datum an welchem der Artikel erstellt worden ist |
| 6 | Text des Artikels |
| 7 | Bild des Artikels |



4

5

3

2

1

Abbildung : Artikel bearbeiten

|  |  |
| --- | --- |
| 1 | Titel des Artikels |
| 2 | Bild des Artikels |
| 3 | Texts des Artikels |
| 4 | Artikel speichern |
| 5 | Artikel löschen |



4

3

1

2

Abbildung : Benutzer Info

|  |  |
| --- | --- |
| 1 | Profilbild des Benutzers |
| 2 | Benutzername |
| 3 | Vorname und Name |
| 4 | Benutzerbeschreibung |



1

8

7

6

5

4

3

2

Abbildung : Benutzer Info bearbeiten

|  |  |
| --- | --- |
| 1 | Profilbild des Benutzers |
| 2 | Benutzername |
| 3 | Vorname |
| 4 | Name |
| 5 | Email |
| 6 | Passwort (Um das Passwort zu ändern einfach den Wert ändern) |
| 7 | Bild (Fileupload) |
| 8 | Beschreibung (Verwendung von TinyMCE) |

## UseCase Diagramm



Abbildung : UserCase Diagramm

# Planung

## Datenbank

In der Datenbank werden die Benutzer und die Artikel gespeichert. Jeder Artikel hat 1 Benutzer und ein Benutzer kann auch mehrere Artikel besetzen.

Die Datenbank sieht wie folgt aus:



Abbildung : Datenbank

# Installation

## Anforderungen

|  |  |
| --- | --- |
| Framework / System | Version |
| XAMPP | 3.3.2 |
| * Apache | 2.4.17 |
| * PHP | 5.6.21 |
| * MySQL | 10.1.13-MariaDB |

## Konfiguration

|  |  |  |
| --- | --- | --- |
| File | Variable | Wert |
| Index.php | BASEURL (Linie 18) | Wo ist die Applikation aufrufbar? (z.B. http://localhost:81/thisisme) |

## Login Daten

|  |  |  |
| --- | --- | --- |
| Benutzer | Passwort | Berechtigung |
| Admin | admin | Administrator |
| User | user | Standard |
| fabioardito | Fabio | Standard |
| devinmenzi | Devin | Standard |
| janled | Jan | Standard |
| joelklingler | Joel | Standard |

## Datenbankinstallation

Die Datenbank wird automatisch beim Seitenaufruf installiert.

1 public function dbConnect(){

2 //Datenbankverbindung wird eingerichtet

3 @$this->\_con = new mysqli(MYSQL\_HOST, MYSQL\_USER, MYSQL\_PW, MYSQL\_DB);

4 //Datenbankverbindung wird hergestellt. Tritt ein Error auf so wird die

5 //Datenbank installiert

6 if($this->\_con->connect\_errno){

7 $this->dbInstall();

8 }

9 }

## Test Daten

Die Testdaten befinden sich im Ordner Testdaten.

# Testplan

## Testfälle

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Testfall | Erwartetes Resultat | Ergebnis |
| 1 | Benutzer kann sich einloggen mit Benutzername ‘User’ und Passwort ‘user’ | Login erfolgreich | OK |
| 2 | Administrator kann sich einloggen mit Benutzername ‘Admin’ und Passwort ‘admin’ | Login erfolgreich | OK |
| 3 | Benutzer geht auf die Startseite | Sieht alle Einträge aller Benutzer | OK |
| 4 | Benutzer geht auf die Startseite | Der Benutzer sieht das Bearbeitungsicon nicht, wenn der Artikel von einem anderen Benutzer ist | OK |
| 5 | Administrator geht auf die Startseite | Sieht alle Einträge aller Benutzer | OK |
| 6 | Administrator geht auf die Startseite | Der Administrator sieht alle Bearbeitungsicon bei allen Artikeln | OK |
| 7 | Administrator klickt auf das Bearbeitungsicon eines Artikels | Der Administrator sieht das Bearbeitungsfenster und kann den Artikel bearbeiten | OK |
| 8 | Benutzer klickt auf das Bearbeitungsicon eines Artikels, welcher von ihm erstellt wurde | Der Benutzer sieht das Bearbeitungsfenster und kann den Artikel bearbeiten | OK |
| 9 | Benutzer klickt in der Navigation auf seinen Benutzernamen | Der Benutzer sieht seinen Account | OK |
| 10 | Benutzer klickt auf seinem Profil auf bearbeiten | Der Benutzer kann seinen Account bearbeiten | OK |
| 11 | Administrator klickt auf das Navigationsitem ‘Settings’ | Der Administrator sieht alle Benutzer | OK |
| 12 | Administrator klickt auf editieren eines Benutzers | Der Administrator wird auf die Bearbeitungsseite des Benutzers weitergeleitet | OK |
| 13 | Administrator klickt auf löschen | Der Administrator kann den gewünschten Account löschen | OK |
| 14 | Der Benutzer klickt auf löschen in seinen Einstellungen | Der Benutzer wird auf die Startseite weitergeleitet und ausgeloggt | OK |
| 12 | Der Benutzer kann auf seiner persönlichen Seite einen neuen Artikel erstellen | Neuer Artikel wird erstellt und veröffentlicht | OK |
| 13 | Benutzer kann seinen Eintrag bearbeiten | Eintrag wird bearbeitet | NICHT OK |

## Abweichungen

Folgende Abweichungen sind aus den Tests hervorgegangen:

|  |  |
| --- | --- |
| Test ID | Abweichung |
| 13 | Wird der Artikel gespeichert ohne ein neues Bild hinzuzufügen wird kein Bild gespeichert. |

# Fazit

Gerne hätte ich das Projekt mit ReactJs umgesetzt, doch der Aufwand ist zu gross und würde nicht in die vorgegebene Zeit passen.

Des Weiteren hätte ich gerne noch mehr Zeit in die Sicherheit investiert, dass nicht jeder Benutzer, wenn er eingeloggt ist, einfach eine URL aufrufen kann und ein Eintrag wird gelöscht. Auch die Verschlüsselung der Passwörter wäre gut gewesen.

In einem nächsten Projekt würde ich wieder eine Abgrenzung erstellen und darauf schauen das ich die wichtigsten Punkte habe. Allerdings würde ich bisschen mehr Zeit in die Planung zu investieren, damit ich genauer weiss was ich machen muss.

# Abbildungsverzeichnis

[Abbildung 1: Standard Layout 5](#_Toc454874341)

[Abbildung 2: Login Formular 5](#_Toc454874342)

[Abbildung 3: Registration Formular 6](#_Toc454874343)

[Abbildung 4: Artikel 6](#_Toc454874344)

[Abbildung 5: Artikel bearbeiten 7](#_Toc454874345)

[Abbildung 6: Benutzer Info 7](#_Toc454874346)

[Abbildung 7: Benutzer Info bearbeiten 8](#_Toc454874347)

[Abbildung 8: UserCase Diagramm 9](#_Toc454874348)

[Abbildung 9: Datenbank 10](#_Toc454874349)