Dokumentation für Projekt Learning Journal

Mischok Academy

17.09.2024 - 02.10.2024

Karl Straße 12

86150

Augsburg



Projekt vom

Jasmin Kozlica

Nikola Greguric

Joe

[1.Ausgang Situation   2](#_Toc1)

[2.Software Architektur  2](#_Toc2)

[1. Übersicht der Datenbank  2](#_Toc3)

[2. Datenbankstruktur  3](#_Toc4)

[3. Tabellen im Detail 3](#_Toc5)

[**Users** 3](#_Toc6)

[**JournalEntries** 3](#_Toc7)

[**Roles** 3](#_Toc8)

[4. Beziehungen zwischen den Tabellen  4](#_Toc9)

[5. Datenbankverwaltung mit Flyway  4](#_Toc10)

[6. Sicherheitsaspekte  4](#_Toc11)

[7. Fazit  4](#_Toc12)

[1. Übersicht des Frontends  5](#_Toc13)

[2. Struktur des Frontends  6](#_Toc14)

[3. Funktionalitäten des Frontends  6](#_Toc15)

[4. Design des Frontends  7](#_Toc16)

[5. Herausforderungen bei der Entwicklung des Frontends  7](#_Toc17)

[6. Fazit  7](#_Toc18)

## 1.Ausgang Situation

Nach den Sommerferien hatten wir zwei Wochen theoretischen Unterricht zum Thema Softwarearchitektur, inklusive eines Crashkurses in HTML und CSS. Ziel dieses Unterrichts war es, uns die theoretischen Grundlagen einer Webanwendung näherzubringen. Wir haben dabei grundlegende Konzepte für dieses Projekt gelernt, wie zum Beispiel, was das Spring Framework ist und wie es theoretisch angewendet werden kann. Außerdem haben wir uns mit der Datenbank-Migration mittels Flyway und der Nutzung von JPA-Queries vertraut gemacht.

Beim Projekt "Learning Journal" sollten wir basierend auf dem zuvor erstellten Pflichtenheft das Learning Journal entwickeln. Das Problem dabei war, dass wir zum Zeitpunkt der Erstellung des Pflichtenhefts noch zu wenig Verständnis für Softwarearchitektur hatten. Im Pflichtenheft war ursprünglich vorgesehen, viele Funktionen hardcodiert umzusetzen, was zwar möglich gewesen wäre, aber sehr viel Zeit in Anspruch genommen hätte. Dank Spring konnten wir jedoch einen Großteil der Funktionen automatisieren und effizienter implementieren.

## 2.Software Architektur

Das Projekt "Learning Journal" besteht aus drei Hauptkomponenten: Einer H2-Datenbank, die wir lokal eingerichtet haben, einem Backend, und einer Schnittstelle dazwischen. Um die Verbindung zwischen der Datenbank und dem Backend zu ermöglichen, haben wir Flyway für die Datenbank-Migration und -Versionierung verwendet.Flyway ist eine Open-Source-Datenbankmigrationslösung, die SQL- oder  Java-basierte Datenbankmigrationen vereinfacht. Flyway verwendet eine  Befehlszeilenschnittstelle zur Ausführung grundlegender  Migrationsbefehle und unterstützt Datenbanken wie Oracle, SQL Server,  MySQL, PostgreSQL und andere.

#### 1. Übersicht der Datenbank

Die Datenbank für das Projekt „Learning Journal“ basiert auf dem H2-Datenbankmanagementsystem. H2 ist eine leichtgewichtige, relationale Datenbank, die lokal in der Entwicklungsumgebung aufgesetzt wird. Sie bietet die Möglichkeit, SQL-Abfragen effizient zu testen und bietet eine In-Memory-Option, um die Leistung bei der Entwicklung zu steigern. Für die Verwaltung von Schema-Migrationen und Datenbankversionierung nutzen wir Flyway, ein bekanntes Tool zur Durchführung von kontrollierten Änderungen an der Datenbank.

Die Hauptaufgabe der Datenbank besteht darin, die Daten des Learning Journals zu speichern und zu verwalten. Dabei handelt es sich vor allem um Benutzerdaten, Einträge im Learning Journal und dazugehörige Metadaten wie das Erstellungsdatum und die zugehörigen Rollen der Benutzer.

#### 2. Datenbankstruktur

Die Datenbank besteht aus mehreren Tabellen, die miteinander in Beziehung stehen. Die wichtigsten Tabellen sind:

* **Users**: Speichert alle Informationen zu den Benutzern, wie Benutzernamen, E-Mail-Adressen, Passwörter (gehasht) und die zugehörige Benutzerrolle (z.B. Administrator oder Nutzer).
* **JournalEntries**: Diese Tabelle speichert die Einträge im Learning Journal. Jeder Eintrag enthält Felder wie Titel, Inhalt, Erstellungsdatum, sowie einen Fremdschlüssel zu dem Benutzer, der den Eintrag erstellt hat.
* **Roles**: Diese Tabelle speichert die verschiedenen Benutzerrollen (z.B. Admin, User) und dient als Verknüpfung zur **Users**-Tabelle, um Berechtigungen festzulegen.

#### 3. Tabellen im Detail

##### **Users**

|  |  |  |
| --- | --- | --- |
| **Feldname** | **Datentyp** | **Beschreibung** |
| id | Long | Eindeutige Benutzer-ID (Primärschlüssel) |
| username | String | Benutzername |
| password | String | Gehashte Version des Benutzerpassworts |
| email | String | E-Mail-Adresse des Benutzers |
| role | Enum | Rolle des Benutzers (z.B. Admin, User) |

Die Tabelle **Users** speichert alle Benutzerinformationen. Die Passwörter werden dabei sicher mit **BCrypt** gehasht, um den Sicherheitsanforderungen gerecht zu werden. Die Rolle des Benutzers wird als ENUM gespeichert, was bedeutet, dass die Werte in der Tabelle nur vordefinierte Rollen beinhalten können.

##### **JournalEntries**

|  |  |  |
| --- | --- | --- |
| **Feldname** | **Datentyp** | **Beschreibung** |
| id | Long | Eindeutige ID des Journal-Eintrags (Primärschlüssel) |
| title | String | Titel des Journal-Eintrags |
| content | String | Inhalt des Journal-Eintrags |
| created\_at | Timestamp | Erstellungsdatum des Eintrags |
| user\_id | Long | Verweis auf den Benutzer, der den Eintrag erstellt hat (Fremdschlüssel) |

Die Tabelle **JournalEntries** speichert alle Journaleinträge, wobei jeder Eintrag einem spezifischen Benutzer zugeordnet ist. Der Fremdschlüssel **user\_id** bezieht sich auf die **Users**-Tabelle, wodurch eine Zuordnung der Journaleinträge zu den jeweiligen Benutzern ermöglicht wird.

##### **Roles**

|  |  |  |
| --- | --- | --- |
| **Feldname** | **Datentyp** | **Beschreibung** |
| id | Long | Eindeutige Rollen-ID (Primärschlüssel) |
| role\_name | String | Name der Benutzerrolle (z.B. Admin, User) |

Die Tabelle **Roles** entält vordefinierte Rollen, die den Benutzern zugeordnet werden. Über die Tabelle **Users** wird jedem Benutzer eine spezifische Rolle zugewiesen, die dessen Zugriffsrchte in der Anwendung definiert.

#### 4. Beziehungen zwischen den Tabellen

Die wichtigsten Beziehungen in unserer Datenbank sind:

* **Users** ↔ **JournalEntries**: Eine **One-to-Many**-Beziehung, bei der ein Benutzer mehrere Journal-Einträge haben kann, aber jeder Eintrag nur einem Benutzer zugeordnet ist.
* **Users** ↔ **Roles**: Eine **Many-to-One**-Beziehung, bei der ein Benutzer nur eine Rolle haben kann, aber mehrere Benutzer die gleiche Rolle teilen können.

Diese Beziehungen sorgen dafür, dass die Datenbank effizient und normalisiert ist, sodass Datenredundanzen vermieden werden und eine klare Zuordnung zwischen den Datensätzen gewährleistet wird.

#### 5. Datenbankverwaltung mit Flyway

Für die Verwaltung von Datenbankmigrationen setzen wir **Flyway** ein. Flyway ermöglicht es uns, Änderungen an der Datenbankstruktur schrittweise und kontrolliert durchzuführen. Bei jeder Änderung, sei es das Hinzufügen einer neuen Tabelle oder das Anpassen von Feldern, wird eine neue Migrationsdatei erstellt und über Flyway in die Datenbank eingespielt. Dadurch können wir sicherstellen, dass die Datenbank immer in einem konsistenten Zustand ist und alle Änderungen nachvollziehbar dokumentiert werden.

#### 6. Sicherheitsaspekte

Da die Datenbank sensible Benutzerinformationen wie E-Mail-Adressen und Passwörter speichert, haben wir besondere Sicherheitsmaßnahmen getroffen:

* **Passwort-Hashing**: Alle Benutzerpasswörter werden mit dem **BCrypt**-Algorithmus gehasht, um sie vor unbefugtem Zugriff zu schützen.
* **Benutzerrollen**: Über die **Roles**-Tabelle und das **Spring Security**-Framework stellen wir sicher, dass nur autorisierte Benutzer Zugriff auf bestimmte Funktionen haben.

#### 7. Fazit

Die Datenbank des Projekts „Learning Journal“ ist schlank und effizient strukturiert, um alle notwendigen Daten für Benutzer und Journaleinträge zu speichern. Durch die Nutzung von H2 als Entwicklungsdatenbank, Flyway für Migrationen und Spring Data JPA für die Interaktion mit der Datenbank, haben wir eine flexible und sichere Lösung geschaffen, die einfach erweiterbar ist.

Die zweite Komponente des Projekts ist ein Java-basiertes Backend, das als Verbindung zwischen der Datenbank und dem Frontend fungiert. Das Backend stellt das zentrale Nervensystem unseres Projekts dar, das alle erforderlichen Aufgaben ausführt, insbesondere die CRUD-Anfragen (Create, Read, Update, Delete). Es verarbeitet diese Anfragen, gibt die Ergebnisse an das Frontend weiter und speichert die Daten in der Datenbank für zukünftige Anwendungen.

Für den backend haben wir uns auf Spring Framework entschieden.Das Spring Framework bietet ein umfassendes Programmier- und Konfigurationsmodell für moderne, Java-basierte Unternehmensanwendungen - auf jeder Art von Bereitstellungsplattform.

Ein zentrales Element von Spring ist die infrastrukturelle Unterstützung auf Anwendungsebene: Spring konzentriert sich auf die "Verkabelung" von Unternehmensanwendungen, sodass sich die Teams auf die geschäftliche Logik auf Anwendungsebene konzentrieren können, ohne unnötige Bindungen an spezifische Bereitstellungsumgebungen.

Das Backend des Projekts "Learning Journal" bildet das zentrale Nervensystem der Anwendung. Es ist verantwortlich für die Logik und den Datenzugriff, die es ermöglichen, Benutzeranfragen zu verarbeiten und die entsprechenden Informationen bereitzustellen. Das Backend wurde mit dem Spring Framework entwickelt, welches eine Vielzahl von Funktionen bietet, die die Entwicklung von Webanwendungen erleichtern.

Architektur und Struktur: Das Backend folgt einer schichtbasierten Architektur, die eine klare Trennung zwischen der Präsentations-, Geschäfts- und Datenzugriffsschicht ermöglicht. Diese Struktur verbessert die Wartbarkeit und Testbarkeit der Anwendung.

RESTful API: Wir haben eine RESTful API implementiert, um eine effiziente Kommunikation zwischen dem Frontend und dem Backend zu gewährleisten. Diese API stellt Endpunkte bereit, die es dem Frontend ermöglichen, auf Daten zuzugreifen, neue Einträge zu erstellen, bestehende Einträge zu aktualisieren und zu löschen.

Datenzugriff mit JPA: Zur Interaktion mit der H2-Datenbank verwenden wir Java Persistence API (JPA). JPA ermöglicht es uns, Datenbankoperationen mithilfe von Entitäten und Repositories durchzuführen, was die Implementierung von Datenbankabfragen erheblich vereinfacht.

Flyway für Migrationen: Um die Datenbankstruktur zu verwalten und Versionierungen durchzuführen, haben wir Flyway eingesetzt. Flyway ermöglicht es uns, Datenbankmigrationen einfach zu organisieren und sicherzustellen, dass die Datenbank immer auf dem neuesten Stand ist, ohne dass manuelle Anpassungen erforderlich sind.

Sicherheit: Im Backend haben wir auch Sicherheitsmechanismen implementiert, um den Zugriff auf sensible Daten zu schützen. Dazu gehören Authentifizierung und Autorisierung der Benutzer, was sicherstellt, dass nur berechtigte Benutzer Zugriff auf bestimmte Funktionen und Daten haben.

Insgesamt gewährleistet das Backend des "Learning Journal" Projekts eine robuste und flexible Grundlage für die Anwendung, die sowohl die Benutzerfreundlichkeit als auch die Sicherheit berücksichtigt.

Die dritte Komponente des Projekts "Learning Journal" ist das Frontend, welches aus HTML und CSS besteht. Im HTML-Teil sind alle funktionalen Elemente der Seiten umgesetzt, während wir CSS für das Design und Layout verwenden. Da die Entwicklung des Backends sehr viel Zeit in Anspruch genommen hat und die Frontend-Entwickler nicht durchgehend verfügbar waren, haben wir es leider nicht geschafft, das Frontend vollständig abzuschließen.Im Frontend sind alle Funktionalitäten vorhanden, allerdings ist es optisch nicht besonders ansprechend gestaltet.

#### 1. Übersicht des Frontends

Das Frontend des Projekts "Learning Journal" ist der Teil der Anwendung, mit dem die Benutzer direkt interagieren. Es dient als Schnittstelle zwischen den Benutzern und den Backend-Services. Das Frontend wurde unter Verwendung von **HTML** (HyperText Markup Language) und **CSS** (Cascading Style Sheets) entwickelt. HTML wurde verwendet, um die Struktur und Inhalte der Webseiten zu definieren, während CSS für das Layout und Design verantwortlich ist, um die Benutzeroberfläche optisch ansprechend zu gestalten.

Das Ziel des Frontends ist es, den Benutzern eine benutzerfreundliche und intuitive Plattform zu bieten, auf der sie ihre Einträge im Learning Journal erstellen, anzeigen, bearbeiten und löschen können. Gleichzeitig sorgt das Frontend dafür, dass die Kommunikation mit dem Backend über HTTP-Anfragen nahtlos abläuft.

#### 2. Struktur des Frontends

Das Frontend besteht aus verschiedenen HTML-Seiten, die jeweils spezifische Funktionalitäten bereitstellen. Die wichtigsten Seiten sind:

* **Login-Seite**: Auf dieser Seite können sich die Benutzer mit ihren Zugangsdaten anmelden. Sie enthält Eingabefelder für den Benutzernamen und das Passwort sowie ein Formular, das die Anmeldeinformationen an das Backend sendet.
* **Registrierungsseite**: Diese Seite ermöglicht es neuen Benutzern, sich zu registrieren. Hier werden Felder für Benutzernamen, E-Mail-Adresse und Passwort angeboten. Die Daten werden ebenfalls an das Backend gesendet, das die Registrierung verarbeitet.
* **Dashboard-Seite**: Diese Seite wird angezeigt, nachdem sich ein Benutzer erfolgreich angemeldet hat. Sie zeigt eine Übersicht der bisherigen Einträge im Learning Journal an und ermöglicht den Benutzern, neue Einträge hinzuzufügen oder vorhandene Einträge zu bearbeiten bzw. zu löschen.
* **Eintragsseite**: Diese Seite wird verwendet, um neue Journaleinträge zu erstellen oder bestehende Einträge zu bearbeiten. Sie enthält ein Formular mit Feldern für den Titel und den Inhalt des Eintrags.

Die Navigation zwischen den Seiten erfolgt mithilfe von Hyperlinks, die jeweils auf verschiedene HTML-Dateien verweisen.

#### 3. Funktionalitäten des Frontends

Das Frontend bietet eine Reihe von Funktionalitäten, die für die Interaktion mit den Benutzern und das Backend von zentraler Bedeutung sind:

* **Benutzerregistrierung und -anmeldung**: Über die entsprechenden HTML-Formulare können Benutzer sich registrieren und anmelden. Die Formulardaten werden per **POST**-Anfragen an das Backend gesendet. Dabei werden Fehler wie ungültige Anmeldeinformationen oder ein bereits existierender Benutzername über entsprechende Fehlermeldungen im Frontend angezeigt.
* **Erstellung von Journaleinträgen**: Benutzer können über ein Formular neue Einträge in ihr Learning Journal hinzufügen. Diese Einträge werden dann an das Backend übermittelt, wo sie in der Datenbank gespeichert werden.
* **Bearbeitung und Löschung von Einträgen**: Vorhandene Einträge können vom Benutzer bearbeitet oder gelöscht werden. Die Änderungen oder die Löschung werden ebenfalls über **PUT**- oder **DELETE**-Anfragen an das Backend gesendet.
* **Fehlermeldungen und Validierungen**: Das Frontend enthält grundlegende Validierungsmechanismen, um sicherzustellen, dass alle erforderlichen Felder ausgefüllt sind. Falls der Benutzer z.B. beim Erstellen eines Eintrags das Titel- oder Inhaltsfeld leer lässt, wird eine Fehlermeldung angezeigt.

#### 4. Design des Frontends

Das Design des Frontends wurde mit CSS gestaltet, um eine benutzerfreundliche und übersichtliche Benutzeroberfläche zu schaffen. Da das Projekt aufgrund der Backend-Entwicklung zeitlich eingeschränkt war, liegt der Fokus des Designs mehr auf Funktionalität als auf visueller Perfektion.

**Layout und Struktur:** Das Layout der Seiten ist simpel und funktional gehalten. Die Seiten sind in verschiedene Abschnitte unterteilt:

* Ein **Header**, der die Navigationslinks und gegebenenfalls den Logout-Button enthält.
* Ein **Hauptbereich**, in dem die Formulare und Listen von Journaleinträgen angezeigt werden.
* Ein **Footer**, der allgemeine Informationen zur Anwendung enthält.

**Verwendung von CSS:** CSS wird verwendet, um das Erscheinungsbild der Seiten zu definieren, einschließlich der Farben, Schriftarten, Abstände und Positionierungen der Elemente. Folgende grundlegende CSS-Eigenschaften wurden verwendet:

* **Farbschema**: Das Projekt verwendet ein einfaches Farbschema mit kontrastierenden Farben, um die Lesbarkeit und Benutzerfreundlichkeit zu gewährleisten.
* **Schriftarten**: Standard-Schriftarten wurden verwendet, um eine klare und leicht lesbare Darstellung der Texte sicherzustellen.
* **Responsive Design**: Aufgrund des Fokus auf Funktionalität wurde ein grundlegendes responsive Design umgesetzt, das sicherstellt, dass die Seiten auch auf mobilen Geräten oder kleineren Bildschirmen nutzbar sind.

**Verbesserungspotenzial:** Während alle Funktionalitäten des Frontends implementiert wurden, bleibt das Design eher minimalistisch. Das Frontend erfüllt zwar seinen Zweck, könnte jedoch durch weitere gestalterische Verbesserungen deutlich ansprechender gestaltet werden. Hier könnten zusätzliche CSS-Effekte, modernere Layout-Techniken wie **Flexbox** oder **Grid** sowie optimierte Schriftarten und Farbschemata zum Einsatz kommen.

#### 5. Herausforderungen bei der Entwicklung des Frontends

Die Entwicklung des Frontends stieß auf einige Herausforderungen, insbesondere aufgrund der begrenzten Verfügbarkeit der Frontend-Entwickler und des Zeitaufwands, der für das Backend benötigt wurde. Daher wurde das Design nicht so weit ausgearbeitet, wie ursprünglich geplant.

**Zusammenarbeit mit dem Backend**: Ein wesentlicher Punkt war die Integration des Frontends mit dem Backend. Da die API-Endpoints vom Backend bereitgestellt wurden, musste sichergestellt werden, dass die Daten korrekt an das Backend übermittelt und die entsprechenden Antworten verarbeitet werden. Dies führte zu einigen Verzögerungen, da die Schnittstellen zwischen Frontend und Backend sorgfältig abgestimmt werden mussten.

**Zeiteffizienz**: Aufgrund der engen Zeitvorgaben wurden einige geplante Design-Verbesserungen zurückgestellt. Trotz dieser Einschränkungen konnte das Frontend jedoch alle erforderlichen Funktionalitäten bereitstellen.

#### 6. Fazit

Das Frontend des Projekts "Learning Journal" bildet die Benutzeroberfläche der Anwendung und ermöglicht die Interaktion mit den Backend-Diensten. Es wurde mit HTML und CSS erstellt und bietet eine funktionale Basis für die Nutzung der wichtigsten Features. Obwohl das Design derzeit minimalistisch ist, könnten zukünftige Verbesserungen am Layout und an der Ästhetik vorgenommen werden, um das Benutzererlebnis weiter zu optimieren.

Dokumentation :

-https://www.nobleprog.de/cc/flyway

-Spring.io