Projektdokumentation Fitness Under Control

Elia Reutlinger, Gianluca Frongia & Valentino Rusconi

Inhaltsverzeichnis

1. Inhalt des Projektes

Das Projekt soll eine Anwendung ergeben, welche die Möglichkeit bildet Trainings-Pläne mit unterschiedlichen Übungen zu definieren und speichern. Zusätzlich können Pläne von anderen Benutzern durchsucht und favorisiert werden. Somit hat man eine Sammlung aller fürs Training benötigten Einheiten an einem zentralen und überschaubaren Ort.

Die Anwendung baut auf einem Modul-System auf, welches einen dynamischen Aufbau der Seite ermöglicht und Funktionen gruppiert in einzelne Module einteilt.

2. Registrierung und Login-Prozess

Die Registrierung sowie der Login werden über den index.php geregelt. Dabei werden bereits 2 Module (login.php & register.php) angewendet, in welchen die Logik und einige UI-Elemente implementiert sind.

register.php:
Hier kann sich ein Benutzer ein Profil generieren, wobei Grunddaten vom Nutzer abgefragt werden. Diese Daten tragen zur vollständigkeit der dynamisch erstellten Seiten innerhalb der Anwendung bei. Passwörter werden speziell behandelt (Siehe #3).

login.php:
Hier kann sich der Benutzer mit einem vorhandenen Profil durch seinen selbst definierten Benutzernamen und das Passwort anmelden. Dabei wird das Passwort serverseitig mit dem Eintrag in der Datenbank abgeglichen.

3. Datensicherheit sowie Script- und SQL-Injection

Datensicherheit:
Für Zugriffe auf die Datenbank wurde ein eigener User mit beschränkten Rechten erstellt. Dieser kann nur Einträge bearbeiten, löschen und erstellen.
Passwörter werden nie im Klartext in der Datenbank abgespeichert. Wir verwenden ein Crypting Verfahren mit Hash und Salt, was das Passwort in der Datenbank unkenntlich macht, jedoch der Anwendung erlaubt, es abzurufen.

Script- und SQL-Injection:
Damit sich User keinen Zugriff auf nicht autorisierte Daten verschaffen können, wenden wir stets nur Prepared-Statements an. Zusätzlich wird die Usereingabe Clientseitig sowie Serverseitig vor einer Datenbankabfrage geprüft.

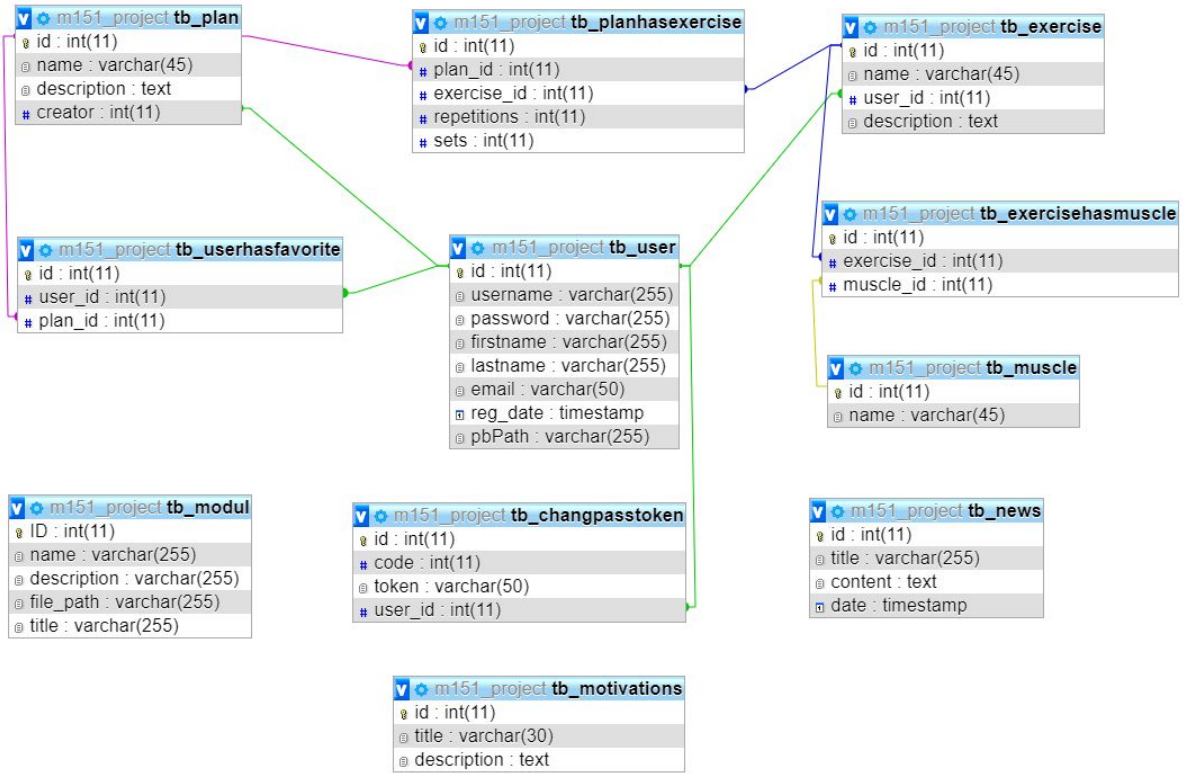
4. Session-Handling

Ein User kann sich nur mit einer aktiven Session innerhalb der Anwendung bewegen. Diese wird beim Login gestartet, und kann durch den “Abmelden” Link in der Navigation beendet/zerstört werden. Um die Sicherheit des Session so gut wie möglich zu gewährleisten, wird deren ID bei jeder Serveranfrage erneuert.

5. Client- und Serverseitige Validierung

Wie bereits in #3 angesprochen, werden die Usereingaben in unserer Anwendung einerseits Clientseitig sowie auch Serverseitig überprüft. Dies bildet einen vertraulichen Prüfungs-Mechanismus, welcher unter Anderem Script- und SQL-Injection verhindert, aber auch die Korrektheit der Daten in der Datenbank garantiert. Clientseitig wird diese Überprüfung durch JQuery und Javascript gelöst, wobei Serverseitig PHP zum Einsatz kommt.

6. Datenbankschema



7. Verbindung zur Datenbank

Die Verbindung zur Datenbank wird in einer separaten Datei definiert, und kann bei Bedarf von jedem Modul abgerufen werden. Dabei wird die Verbindung jeweils geöffnet, jedoch nicht durch einen spezifischen Befehl wieder geschlossen. Darauf konnten wir verzichten, da alle Datenbankabfragen durch PHP geregelt werden. PHP hat die Eigenschaft, nach vollendung seines Scripts die Datenbankverbindung von alleine zu schliessen. Dabei entsteht kein Sicherheitsrisiko, da das ganze Serverseitig abläuft.

8. Behandlung von zusätzlichen Daten

Zusätzliche Daten, welche der User in der Anwendung bearbeitet, löscht oder erstellt, werden durch Ajax-Aufrufe nach der Clientseitigen Validierung an den Server gesendet. Durch die übergebenen Variablen an die zugehörige .php Datei weiss das PHP-Script, was mit den Daten anzufangen ist. Nach der Serverseitigen Validierung werden SQL-Queries ausgeführt, welche die Änderung direkt in der Datenbank bearbeiten.

9. Usability

Der Benutzer wird hauptsächlich anhand von unterschiedlichen Animationen durch die Anwendung geführt. Er hat eine Übersicht über alle Module und kann diese durch nicht mehr als einen Mausklick aufrufen. Um innerhalb eines Moduls eine Funktion auszuführen, werden (abgesehen von möglichen Textfeld-Eingaben) niemals mehr als mindestens 5 Mausklicks benötigt. Dies soll die Usability sowie die Performance der Anwendung garantieren, wobei durch das dynamische hinzufügen von neuen Objekten unnötige Seitenaufrufe erspart bleiben.

10. Errorhandling

Das Errorhandling wurde so simpel wie möglich, jedoch auch mit ausreichend Informationen für den Benutzer gestaltet. Jedes Modul enthält von Grund auf 1-2 UI-Objekte, um Fehler anzuzeigen. Dabei werden diese Objekte erst eingeblendet, wenn durch die Clientseitige oder auch Serverseitige Validierung ein Fehler aufgetreten ist. Je nach Funktionalität bleiben die Fehlermeldungen ersichtlich oder blenden sich nach einigen Sekunden wieder aus.

11. Validierung

Der gesamte Projektcode(HTML/CSS) wurde an Hand W3School erfolgreich validiert.