

Trophäen-Verwaltungs-App | API-Backend in Laravel (PHP-Framework)

21.03.2022 – 11.04.2022



Kandidat

Schneider Elias

Betrieb (=Durchführungsort)

twofold

Thurgauerstrasse 54, PLZ 8050

T 0774200577 (am besten erreichbar)

G 0774200577

M elias.schneider@twofold.swiss

BerufsbildnerIn/ Lehrfirma

Hofer Domenik

Asperger AG

Thurgauerstrasse 54, 8050 / Zürich

T 0445334238 (am besten erreichbar)

G 0445334220

M domenik.hofer@twofold.swiss

Verantwortliche Fachkraft

Hofer Domenik

Asperger AG

Thurgauerstrasse 54, 8050 / Zürich

T 0445334238 (am besten erreichbar)

G 0445334220

M domenik.hofer@twofold.swiss

Hauptexperte

Gut Oriol

T 0774427354 (am besten erreichbar)

G

M oriol00@bluewin.ch

Arbeitsbereiche

-
-
-
-
-
-
-
-
-

Trophäen-Verwaltungs-App | API-Backend in Laravel (PHP-Framework)

21.03.2022 – 11.04.2022

Trophäen-Verwaltungs-App | API-Backend in Laravel (PHP-Framework)

21.03.2022 – 11.04.2022

Ausgangslage

Wir bei der Asperger AG bieten Lernenden auf dem Autismus-Spektrum die Möglichkeit in einem geschützten Rahmen eine Lehre zu machen und fit für den ersten Arbeitsmarkt zu werden. Durch unser spezielles Setting und die Finanzierung durch die IV dürfen wir keine gewinnbringenden, externen Aufträge annehmen, weswegen wir auf interne Projekte bauen. Nebst Informatik bieten wir auch Lehren als Interactive Media Designer (IMD) an. Einer unserer sehr sportlichen IMD-Lernenden hatte sich das Konzept und das zugehörige Screendesign für eine Trophäen-Verwaltungs-App ausgedacht. Da dieser Lernende regelmässig mit seinem Sportclub an Wettkämpfen in der ganzen Schweiz teilnimmt, wollte er eine App schaffen, in welcher man die gewonnenen Trophäen und Medaillen speichern, verwalten und sich darüber auch austauschen kann. Mit dieser IPA soll jetzt das API-Backend für dieses Projekt gestartet werden.

Detaillierte Aufgabenstellung

Es soll ein API-Backend aufgebaut werden für eine Trophäen-App in welcher gewonnene Trophäen gespeichert werden können. Diese Trophäen können zudem zu Kategorien zugewiesen werden.

- Es ist ein API-Backend mit dem Laravel Framework in PHP umzusetzen
- Die Daten sollen in einer geeigneten Struktur in einer Datenbank abgelegt werden
- Die Datenbank-Struktur wird mithilfe einer Migration automatisch aufgebaut
- Die einzelnen Datenbanken werden mit Model-Klassen in Laravel abgebildet
- Für die verschiedenen Inhalte werden Routen bereitgestellt
- Die Routen werden mit den üblichen Verben angesprochen (Create = POST, Read = GET, Update = PUT, Delete = DELETE)
- Die API-Routen sollen in einer geeigneten Form dokumentiert werden für die künftigen Frontend-Entwickler (Swagger o.ä.)
- Die Kommunikation mit der API läuft über JSON
- Anfragen werden in Request-Klassen validiert und aufbereitet
- Antworten werden in Ressourcen-Klassen aufbereitet und zurückgesendet
- Fehler in den Eingaben werden dem Nutzer zurückgeschickt
- Die Routen und Inhalte sind geschützt, sodass nur eingeloggte User Zugriff auf die eigenen Inhalte haben
- Die Routen sollen händisch getestet und in einem Testprotokoll aufgeführt werden
- Das Projekt soll lokal laufen und muss noch nicht veröffentlicht werden
- Das Projekt muss sauber, nachvollziehbar und leicht erweiterbar sein

Endpoints:

Users:

- User sollen sich registrieren und einloggen können (ohne Mail-Validierung)

Categories:

Trophäen-Verwaltungs-App | API-Backend in Laravel (PHP-Framework)

21.03.2022 – 11.04.2022

- Kategorien sollen CRUD Möglichkeiten haben
- Kategorien beinhalten einen Namen + eine Farbe (Bsp: Tennis, Blau oder Fussball, Rot)

Trophies:

- Trophäen sollen CRUD Möglichkeiten haben
- Trophäen beinhalten:
 - Typ (Trophäe/Medallie)
 - Titel
 - Platzierung
 - Datum
 - Kategorie
 - Ort
 - Gegner
 - Spielstand
 - Preisgeld
 - Name des Clubs
 - Bild
- Trophäen können sortiert und gefiltert werden
- Sortierungen: Datum, Platzierung
- Filterung: Typ, Platzierung, Kategorien

Erwartet wird ein funktionierendes API-Backend, aufgebaut in Laravel, mit welchem bspw. über Postman kommuniziert werden kann. Es soll eine funktionierende Registrierung eines Nutzers möglich sein, mit anschliessendem Login. Der eingeloggte Nutzer sollte dann die Möglichkeit haben Kategorien zu erstellen, bearbeiten und zu löschen. Zudem soll der Nutzer die Möglichkeit haben, Trophäen zu erstellen, diese einer Kategorie zuzuweisen und mit allen Inhalten zu speichern, bearbeiten und zu löschen. Auf der Übersichtsseite sieht der Nutzer alle seine eingetragenen Trophäen und kann diese sortieren und filtern.

Mittel und Methoden

XAMPP, Laravel, PHP, Postman

Vorkenntnisse

Laravel, PHP

Mehrfach verwendet und seit über 1 Jahr bekannt

Trophäen-Verwaltungs-App | API-Backend in Laravel (PHP-Framework)

21.03.2022 – 11.04.2022

Vorarbeiten

Keine

Neue Lerninhalte

Keine

Arbeiten in den letzten 6 Monaten

Externes Praktikum, hat dort mit Laravel gearbeitet

Individuelle Kriterien

Auf den folgenden Seiten werden die individuellen Kriterien aufgeführt, welche durch die verantwortliche Fachkraft für diese IPA festgelegt wurden.

Individuelle Kriterien

Leitfrage 1	Plausibilisierung der Benutzer-Eingaben Werden die Eingaben des Benutzers überprüft?
Gütestufe 3	Alle Eingabefelder werden überprüft. Es ist eindeutig gekennzeichnet, welche Felder Pflichtfelder sind. Für den Benutzer ist ersichtlich, welche Wertebereiche zulässig sind. Findet die Plausibilisierung eine Fehleingabe, so wird der Benutzer mit konkreten Hinweisen geführt.
Gütestufe 2	Plausibilisierung findet statt, Feedback an Benutzer ist mangelhaft/nicht eindeutig/unvollständig. Nur korrekte Daten werden übermittelt.
Gütestufe 1	Eingaben werden plausibilisiert, aber bei Fehlern oder fehlenden Eingaben sind die bisher gemachten Eingaben verloren oder die fehlerhaften Eingaben werden trotzdem übermittelt. Oder: es werden nicht alle Eingaben ueberprueft, welche ueberprueft werden sollten.
Gütestufe 0	Es findet keine Plausibilisierung statt.

Notizen

Individuelle Kriterien

Leitfrage 2	Implementierung von Lösungen (Programmieren) Ist der Kandidat in der Lage die vorgeschlagenen Lösungen zu implementieren?
Gütestufe 3	Der Code entspricht den Erwartungen und kann bedenkenlos verwendet werden. Die Sprachmittel wurden richtig eingesetzt.
Gütestufe 2	Der Code weist einzelne Mängel auf. Das Resultat muss vor dem Einsatz überarbeitet werden. Die Sprachmittel sind nicht immer richtig gewählt.
Gütestufe 1	Der Code weist klare Mängel auf. Das Resultat muss vor dem Einsatz gründlich überarbeitet werden. Die Sprachmittel sind nicht richtig gewählt. Oder: der Kandidat versteht den Zweck der Sprachelemente nicht.
Gütestufe 0	Der Code ist deutlich unter den Erwartungen und kann nicht wirklich gebraucht werden.

Notizen

Individuelle Kriterien

Leitfrage 3	Codingstyle - lesbarer Code Ist der Code lesbar geschrieben, gut gegliedert und ist die Namensgebung gut gewählt?
Gütestufe 3	Die Namensgebung entspricht den Vorgaben oder ist einfach gut gewählt. Die Struktur des Codes ist ebenfalls gemäss möglicher Richtlinien oder einfach übersichtlich gemacht. Es ist eine gewisse Einheit zu sehen in der Art und Weise, wie der Code strukturiert ist (d.h. es ist überall etwa gleich gemacht).
Gütestufe 2	Die Namensgebung ist ab und zu ungeschickt gewählt, Vorgaben sind teilweise berücksichtigt. Die Codestruktur ist uneinheitlich (so dass Lesbarkeit leidet).
Gütestufe 1	Die Namensgebung ist öfters verwirrend oder unpräzise. Dem Code fehlt es an einigen Stellen an klarer Struktur. Vorgaben sind nicht berücksichtigt.
Gütestufe 0	Die Namensgebung ist verwirrend oder unpräzise. Der Code ist schlecht lesbar.

Notizen

Individuelle Kriterien

Leitfrage 4	Codingstyle - Dokumentation Ist der Quellcode dokumentiert? Hilft der Text, die Funktionalität zu verstehen und nachzuvollziehen?
Gütestufe 3	Der Text erklärt was notwendig ist, um die Funktion/Methode besser zu verstehen. Allfällige Richtlinien sind eingehalten.
Gütestufe 2	Der Text hilft nur zum Teil weiter. Allfällige Richtlinien sind teilweise berücksichtigt.
Gütestufe 1	Der Text hilft selten weiter. Allfällige Richtlinien sind offensichtlich verletzt.
Gütestufe 0	Es ist wenig bis gar nichts dokumentiert.

Notizen

Individuelle Kriterien

Leitfrage 5	Codierung: Fehlerbehandlung
	Fehlerbehandlung: Werden mögliche Fehler mit den entsprechenden Mitteln erkannt und behandelt?
Gütestufe 3	Es wurde eine Strategie zur Fehlerbehandlung eingeführt und im ganzen Code konsistent verwendet.
Gütestufe 2	Die Fehlerbehandlung ist lückenhaft und/oder die Fehler bleiben nach der Erkennung unbehandelt.
Gütestufe 1	Die Fehlerbehandlung wurde oft vergessen. Code streckenweise ohne Fehlertests (wo sie als notwendig erachtet würden).
Gütestufe 0	Fehlerbehandlung nicht oder nur sehr rudimentär vorhanden.

Notizen

Individuelle Kriterien

Leitfrage 6	Durchführen des Normalisierungsprozesses (1NF, 2NF, 3NF) Wurde der Normalisierungsprozess über die modellierte Datenbasis durchgeführt und verifiziert und erfüllt diese die Anforderungen 1NF, 2NF und 3NF?
Gütestufe 3	Alle Anforderungen sind erfüllt und die Datenbasis befindet sich in der 3NF.
Gütestufe 2	Die Anforderungen der 2NF sind korrekt erfüllt, jedoch sind die Anforderungen an die 3NF mangelhaft und unvollständig umgesetzt.
Gütestufe 1	Sowohl die Anforderungen der 2NF als auch der 3NF sind mangelhaft und unvollständig umgesetzt.
Gütestufe 0	Die Datenbasis weist Redundanz auf und befindet sich nicht in der 3NF.

Notizen

Individuelle Kriterien

Leitfrage 7	Vollständiges ERM bzw. Datenmodell Ist das ERM bzw. Datenmodell vollständig dargestellt?
Gütestufe 3	1. Alle Entitäten und Beziehungen sind korrekt dargestellt. 2. Alle Assoziationstypen (1, c, m, mc) sind korrekt eingetragen. 3. Alle Primärschlüssel und Fremdschlüssel sind als solche erkenntlich, also entsprechend markiert bzw. bezeichnet. 4. Alle Attributlisten sind vollständig, die Datentypen aller Attribute sind angegeben.
Gütestufe 2	Drei der obigen Punkte treffen zu.
Gütestufe 1	Zwei der obigen Punkte treffen zu.
Gütestufe 0	Nur einer oder keiner der obigen Punkte trifft zu.

Notizen
