

Moderne Webtechnologien I (WT 1) Praktikum 3

WS 2020/21

Prof. Dr. C. Köhn
Daniela Böing

7. Oktober 2020

Website: <https://www.hochschule-bochum.de/fbe/einrichtungen-im-fachbereich/labor-fuer-medienkommunikation-internet-und-robotik/>

Hochschule Bochum
Bochum University
of Applied Sciences



Inhaltsverzeichnis

1	Praktikum 3	2
1.1	Ausgangslage	2
1.2	Vorgehen	2

1 Praktikum 3

1.1 Ausgangslage

Aktuell wird in Ihrem Unternehmen ein administratives Backend aufgebaut. Eine entsprechend zuständige Abteilung hat bereits die zugehörigen HTML-, CSS- und JS-Dateien bereitgestellt, eine andere hat sich im Rahmen der Vagrant-Technologie um eine Testdatenbank gekümmert.

Eine Datenbankverbindungsklasse, die die initiale Datenbankverbindung herstellt, wird zur Verfügung gestellt und soll erweitert werden.

Die Datenbank enthält die Tabelle `Product`, die wie folgt aussieht:

- `product_id` INT PRIMARY KEY
- `product_name` VARCHAR(60) NOT NULL
- `unit_price` DECIMAL(4,2) NOT NULL

Ihr Chef stellt Ihnen einen Prototypen zur Verfügung und bittet Sie um Folgendes:

Die Produkte sollen aus der Datenbank gelesen und auf der Website so angezeigt werden, dass der Preis geändert und anschließend verändert in der Datenbank gespeichert werden können.

Entsprechend den Unternehmensstandard möchte Ihr Chef, dass Sie als Programmiersprache PHP verwenden.

1.2 Vorgehen

Vorbereitend ist Folgendes nötig:

- Laden Sie sich die Vorlage sowie das MySQL-Dump herunter.
- Starten Sie die Ihren Webserver wie gewohnt (Vagrant, Docker, XAMPP, ...).
- Laden Sie das MySQL-Dump wie gewohnt in Ihre Arbeitsumgebung. Bei Problemen melden Sie sich bei uns.

Erweitern Sie dann die Vorlage wie folgt:

1. Erzeugen Sie eine Klasse `Product` (in gleichnamiger Datei), die die Eigenschaften der Tabelle beinhaltet.
2. Schreiben Sie in der Datenbankklasse eine Methode `read_products`, die die Produkte mittels PDO aus der Datenbank ausliest und als Array aus `Product`-Objekten zurückgibt. Zur Hilfe folgendes SQL-Statement:

```
1 SELECT product_id, product_name, unit_price FROM Product;
```

3. Rufen Sie die Methode auf der Seite auf und geben Sie alle Objekte mit ihren Werten aus.

4. Nutzen Sie nun die Einträge des Arrays, um die Datenbankeinträge formatiert wie in der Vorlage auszugeben.
5. Schreiben Sie die jeweilige Produkt-ID in das versteckte Eingabefeld, damit diese nachher genutzt werden kann. Nutzen Sie diese auch für das `form`-Attribut sowie für die ID des `form`-Tags, damit die Zeile eindeutig identifiziert wird.
6. Ergänzen Sie anschließend in der Datenbankklasse eine Methode `update_product` zum Updaten eines Produkts mit übergebener Produkt-ID und Produktpreis. Es sollen Prepared Statements mit PDO verwendet werden. Zur Hilfe folgendes SQL-Statement (`<id>` und `<unit_price>` müssen entsprechend ersetzt werden):

```
1 UPDATE Product SET unit_price = <unit_price> WHERE product_id = <id>;
```

7. Die Auswertung des Formulars (Eingabefelder und Update-Button) sollen durch die `index.php` passieren. Nach dem Update sollen die aktualisierten Werte angezeigt werden.
8. Sorgen Sie dafür, dass keine Fehlermeldungen auf der Seite angezeigt werden (auch nicht beim ersten Aufruf der Seite ohne Formular).

Wichtig: Wenn Sie fertig sind, laden Sie bitte Ihre Dateien in Moodle bei der entsprechenden Aufgabe hoch. Die Aufgabe wird nur im Zeitraum der jeweiligen Versuche freigeschaltet sein. Mindestens 50% der Anforderungen, also 4 Punkte, müssen korrekt umgesetzt werden. Die Dateien dürfen nur einmalig hochgeladen werden.