

Web-Engineering I

Organisation:

Die Übungen sollen in 2er-Teams erfolgen. Wenn nicht anders möglich, können sie auch allein durchgeführt werden. Mehr als zwei Team-Mitglieder werden nicht zugelassen.

Leistungsbewertung:

Wer die LV MME1 (Übungs- und VL-Teil) bestehen will, muss folgende Bedingungen erfüllen (ein teilweises Bestehen ist ausgeschlossen).

1. Alle Übungsaufgaben müssen vom gesamten Team rechtzeitig abgeben und vorgeführt werden.
2. Alle Übungsaufgaben müssen mit einer 4.0 oder besser bewertet worden sein.
3. Die Abschlussklausur muss bestanden sein.

Wenn die LV bestanden wurde, resultiert die Note der LV aus der Summe aller Punkte (Punkte der Übungsaufgaben 1 bis 4 und der anrechenbaren Punkte der Klausur) anhand der folgenden Tabelle:

Note:	4.0	3.7	3.3	3.0	2.7	2.3	2.0	1.7	1.3	1.0
Ab Pkte.	4	8	12	16	20	24	28	34	40	46

Falls freiwillig ein guter Vortrag zur Ergänzung der Vorlesung gehalten wurde, wird die so ermittelte Note um eine Stufe verbessert (Beispiel: 1.7 → 1.3).

Übung:

Vorbereitung:

Wählen Sie ein klassische Anwendungsszenario (z. B. Termin-, Aufgaben-, Musik-, Video-, Location-, Routen-Verwaltung), das Sie mit einer Webanwendung verwalten wollen.

Sie dürfen die Komplexität so beschränken, dass alle vorkommenden persistent zu haltenden Daten in einer Datenbanktabelle zusammengefasst werden können.

Es sollen mindestens folgende Formate in der Datenbank gespeichert werden:
Geo-Koordinaten, String, URL, Email, Datum, Uhrzeit.

Spezifizieren Sie diese Datenbanktabelle.

1. Aufgabe: Client-Seite mit HTML5 und CSS ohne Framework

Erstellen Sie ein passendes Web-Interface für die naheliegenden Datenbank-Operationen (new, delete, update, find, show all).

Integrieren Sie Header, Footer, mind. 2 Spalten, mind. eine Tabelle zur späteren Anzeige aller Datensätze, mind. ein Bild.

Da erst später die Verbindung zum Server und der Datenbank realisiert wird, kann die Tabelle vorerst nur mit

statischen Daten (im HTML) gefüllt werden.

Verwenden Sie HTML5-Formulare und die damit gegebenen Möglichkeiten, Eingaben ohne JavaScript zu erzwingen und zu prüfen.

2. Aufgabe: Zusätzlich JavaScript ohne Framework und ohne Bibliothek (wie z. B. JQuery)

2.1:

Implementieren Sie mit JavaScript die Überprüfung der Benutzereingaben im Eingabeformular. Erzwingen Sie dabei mindestens eine spezielle, nicht triviale Anforderung.

Beispielsweise, dass eine Email-Adresse mit @beuth-hochschule.de endet, oder bestimmte Geokoordinaten in der Umgebung von Berlin liegen.

Wichtig: Das gesamte User Interface sollte möglichst professionell, komfortabel und informativ auf Eingabefehler reagieren.

2.2:

Implementieren Sie einen Testdatengenerator für das Eingabe-Formular. Bei Tastendruck oder einem gezielten Mausklick sollten die Eingabe-Felder mit immer wieder neuen repräsentativen Eingaben gefüllt werden. Es wäre super, wenn damit das Testen und das Demonstrieren erleichtert würde.

Für eine sehr gute Bewertung (> 8 Punkte) muss noch die folgende Aufgabe gelöst werden:

2.3:

Programmieren Sie eine weitere nicht triviale Funktionalität mit JavaScript, die zu Ihrer Anwendung passt. Die Einbindung von Google Maps (OpenStreetMap, ...) ist hierfür eine naheliegende Option.

3. Aufgabe: Client-Seite mit Framework(s)

3.1:

Ersetzen Sie (teilweise?) das für Aufgabe 1 und 2 erstellte Interface, durch eine responsive Version. Verwenden Sie dazu ein leistungsfähiges Framework (wie z. B. Bootstrap oder Foundation) und beliebig viele Bibliotheken (wie z. B. JQuery mit Plugins).

Gut wäre es, wenn Ihr Resultat professioneller wirken würde, als die Version von Aufgabe 1. Beispielsweise könnte mit einem vergleichsweise geringen Aufwand eine Single Page Application mit einem modalen Dialog entstehen.

3.2:

Betten Sie einen speziellen Font ein, der normalerweise nicht zur Verfügung steht.

4. Aufgabe: Client-Seite mit AJAX ↔ Server-Seite mit Apache, MySQL und PHP

Installieren Sie – falls Sie es noch nicht gemacht haben - auf Ihrem Entwicklungsrechner einen Webserver (mit der XAMPP- oder MAMP-Distribution).

Erstellen Sie die geplante zu Ihrer Anwendung passende relationale Datenbank.

Verbinden Sie Ihren Client aus Aufgabe 3 mit dem Server und der Datenbank unter Verwendung von AJAX, PHP und JSON.

Die Übung ist bestanden, wenn es mit den genannten Techniken gelingt, eine bidirektionale Kommunikation zwischen Client und Datenbank herzustellen.

Für eine gute Note wird erwartet, dass alle CRUD-Operationen (incl. Anzeige in Tabelle und Suchfunktionalität) implementiert sind.

Eine sehr gute Bewertung setzt zusätzlich eine serverseitige Eingabeüberprüfung voraus. Außerdem sollten auch serverseitige Fehler- und Erfolgeereignisse clientseitig sinnvoll dargestellt werden.

Hinweise:

Jede der 4 Übungen wird anhand der folgenden Tabelle mit Punkten bewertet.

Note:	4.0	3.7	3.3	3.0	2.7	2.3	2.0	1.7	1.3	1.0
Punkte:	1	2	3	4	5	6	7	8	9	10

Die Übungsaufgaben 1 bis 3 dürfen nach der pünktlichen Abgabe nachgebessert und zu den folgenden drei Übungsterminen zur Verbesserung der Bewertung erneut vorgeführt werden. Die vierte Übungsaufgabe darf vorzeitig abgegeben werden um eine nachträgliche Verbesserung zu ermöglichen.

Bei der Abgabe müssen alle Team-Mitglieder anwesend sein.

Alle Team-Mitglieder müssen alle abgegebenen Aufgaben vorführen und erklären können. Ansonsten ist ein Bestehen unmöglich.

Die Aufgaben wurden absichtlich so formuliert, dass sie sich auf vielfältige Art realisieren lassen.

Es wird vorausgesetzt, dass die Teilnehmer selbst daran interessiert sind, möglichst viel zu lernen und Freude daran zu haben kreativ zu sein.

Deshalb werden triviale Minimallösungen, auch wenn sie formal den hier formulierten Kriterien entsprechen, deutlich schlechter bewertet als Lösungen, die erkennen lassen, dass viel probiert und viel gelernt wurde.

Der Schwerpunkt der LV liegt auf der Client-Seite. Deshalb reicht serverseitig eine einfache, aber gut strukturierte PHP-Lösung völlig aus.

Serverseitig Frameworks oder PHP-Alternativen sollen nicht verwendet werden. Sie werden u. U. Gegenstand von MME2 sein.

Nutzen Sie die Chance, regelmäßig mit dem Dozenten zu kommunizieren und sich Feedback geben zu lassen. Dann werden die (gar nicht so großen Anforderungen) bald klar sein.

Klausur

Die Klausur behandelt nur Inhalte, die während der Vorlesung vorgetragen wurden oder im Skript stehen. Sie ist relativ leicht und soll primär gewährleisten, dass ALLE Teilnehmer wirklich mitarbeiten.

In der Klausur können maximal 20 Klausurpunkte erreicht werden. Sie wird anhand der folgenden Tabelle bewertet:

Klausurpunkte:	<10	10	11	12	13	14	15	16	17	18	19	20
Note:	5.0	4.0	4.0	3.7	3.3	3.0	2.7	2.3	2.0	1.7	1.3	1.0
Anrechenbare Punkte:	irrelevant	0	1	2	3	4	5	6	7	8	9	10

Die anrechenbaren Punkte gehen zu 20% in die Berechnung der Gesamtnote ein.