

Web Technologien FS23 MEP Abschlussprojekt

Der Leistungsnachweis im Modul WEBT ist in Form eines bewerteten Abschlussprojekts zu erbringen. **Dies ist eine Einzelarbeit, d.h., Sie müssen dieses Abschlussprojekt selbstständig erarbeiten.** Sie haben einen Zeitraum von mehreren Wochen zur Verfügung, um diese Projektarbeit zu erarbeiten. Die Projektarbeit ist einzureichen bis spätestens am:

Freitag, 16. Juni 2023 um 23:59 Uhr.

1. Aufgabenstellung

Erstellen Sie eine Webapplikation, welche die untenstehenden Rahmenbedingungen und Anforderungen erfüllt. Der grundlegende Aufbau der Webapplikation ist vorgegeben und besteht in einem clientseitigen Teil (statisches HTML-Dokument mit Formular) sowie einem serverseitigen Teil (PHP-Skript, welches einen Effekt hat und ein HTML-Dokument erstellt). Abbildung 1 zeigt schematisch den Aufbau und die Interaktion innerhalb der Webapplikation. Die Webapplikation muss ein eigenes, einheitliches Thema haben. Die Umsetzung aller Anforderungen ist nach diesem Thema auszurichten. Der eigentliche Inhalt und das Design der Webapplikation ist Ihnen überlassen, muss aber zu dem gewählten Thema passen.

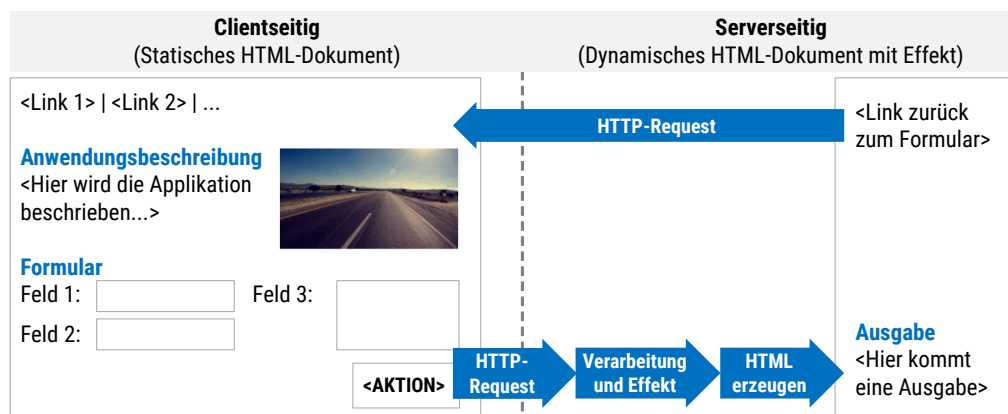


Abbildung 1: Schematische Darstellung des Aufbaus und der Interaktion innerhalb der Webapplikation.

2. Rahmenbedingungen

- Sie dürfen nur die im WEBT-Modul besprochenen Technologien (HTML, CSS, JavaScript, W3.CSS, Vue.JS, PHP, SQL, XML, JSON, AJAX) verwenden. Die Verwendung anderer Programmiersprachen, zusätzlicher Funktionsbibliotheken und Frameworks ist nicht gestattet¹. **Teile der Webapplikation, welche mit unzulässigen Technologien erstellt wurden, werden von der Bewertung ausgenommen.**
- Sie wählen ein eigenes Thema für die Webapplikation. Dieses muss konsistent über die ganze Webapplikation hinweg angewendet werden. Sie dürfen bei der Wahl des Themas kreativ sein. **ABER:** Das Thema darf keines der Themen sein, welche wir bereits in den Vorlesungsbeispielen oder Übungen behandelt haben, also kein **Kontaktformular**, Idealgewicht, Zufallszahl, Pizzashop (andere Shops sind erlaubt), Währungsumrechnung, Fahrplan, etc. **Teile der Webapplikation, welche nicht dem Thema entsprechen, erhalten nicht die volle Punktzahl.**
- Sie müssen folgende Limiten einhalten:

Kategorie	Beschreibung
HTML	Maximal eine HTML-Datei mit maximal 200 Zeilen (Kommentare werden nicht gezählt) und höchstens einem HTML-Element pro Zeile.

¹ Erlaubt sind die objektorientierten PHP-APIs für MySQL-Zugriffe (sowohl mysqli und PDO) sowie die Javascript fetch-API (anstelle der XMLHttpRequest-API).

PHP	Maximal eine PHP-Datei mit maximal 200 Zeilen (Kommentare werden nicht gezählt) und höchstens einer Anweisung pro Zeile.
JavaScript	Mehrere JS-Dateien erlaubt. In der Summe über alle Dateien maximal 200 Zeilen (Kommentare werden nicht gezählt) mit höchstens einer Anweisung pro Zeile.
CSS	Mehrere CSS-Dateien erlaubt. In der Summe über alle Dateien maximal 200 Zeilen (Kommentare werden nicht gezählt) mit höchstens einer Regel/Property pro Zeile.

3. Anforderungen an die Webapplikation

Bereich	Beschreibung
HTML-Dokument (Client)	<p>Ein einzelnes HTML-Dokument, welches folgende drei Abschnitte hat:</p> <ul style="list-style-type: none"> • Navigation: Die Navigation muss ein direktes Anspringen der Abschnitte Information und Formular innerhalb desselben HTML-Dokuments erlauben. • Information: Ein Abschnitt mit Informationen zur Webapplikation und wie der Benutzer die Applikation zu bedienen hat. Dieser Abschnitt muss mindestens ein Paragraphelement und ein Bildelement enthalten. • Formular: Ein Formular mit mindestens drei Inputelementen, welche jeweils unterschiedlichen Inhalt (*) erwarten, den entsprechenden Labels und einer Submit-Aktion. Die Submit-Aktion des Formulars muss das serverseitige Skript aufrufen und die Inhalte aller Inputelemente an das Skript übermitteln. <p>(*) Unterschiedlich sind z.B.: Text (Name, Strasse, usw.), kontinuierlicher Wert (z.B. Körpergrösse), diskreter Wert (z.B., verschiedene Optionen), Datum, Farbwert, usw.</p> <p>Zusätzlich muss das HTML-Dokument folgende Anforderungen erfüllen:</p> <ul style="list-style-type: none"> • Nur gültiges HTML verwendet. • Verwendung passender (semantischer) HTML-Elemente (<header>, <nav>, <section>, etc.). • Verwendung dokumenten-relativer URLs überall dort, wo dies möglich ist. • Erfüllung des WCAG-Erfolgskriterium 1.1.1. • JavaScript-Code und CSS-Regeln müssen jeweils in separaten Dateien stehen (nicht im HTML-Dokument). JavaScript-Code in HTML-Attributen ist erlaubt.
PHP-Skript (Server)	<p>Serverseitig muss die Webapplikation folgende Funktionalität implementieren:</p> <ul style="list-style-type: none"> • Erzeugen einer HTTP-Response mit Inhalt von Typ HTML unter Einbezug der Inhalte aller vom HTML-Formular gesendeten HTTP-Request-Parametern. • HTTP-Response muss einen Hyperlink zurück zum HTML-Dokument beinhalten. • Das serverseitige Skript muss weiter einen Effekt haben. Dieser Effekt muss zum HTML-Formular passen. Beispiele: eine Registrierung erzeugt einen Tabelleneintrag in der Datenbank, eine Berechnung führt diese durch und gibt das Resultat in der Response aus, usw. • Das serverseitige Skript muss ein selbstdefiniertes Cookie (kein Sessioncookie) setzen, welches zum Thema passt und vom serverseitigen Skript bei einem erneuten Abruf wieder ausgewertet wird. Das Cookie muss eine sichtbare Änderung im serverseitig erstellten HTML bewirken.
Design / Layout (Client)	<p>Das statische HTML-Dokument muss über ein responsives Layout für die drei Abschnitte Navigation, Information und Formular verfügen.</p> <ul style="list-style-type: none"> • Das responsive Layout muss für die zwei Geräteklassen Mobile und Desktop jeweils unterschiedliche und für die entsprechende Geräteklasse geeignete Darstellungen verwenden. • Die Implementation des responsiven Layouts muss unter der Verwendung von W3.CSS erfolgen. • W3.CSS muss mittels einer absoluten URL direkt vom W3.CSS-Anbieter (w3schools.com) eingebunden werden.

	<ul style="list-style-type: none"> • Verwenden Sie zusätzlich mindestens drei eigene CSS-Regeln, welche bei der Darstellung zur Anwendung kommen und fließende Übergänge mittels CSS-Transition implementieren. • Skalierung des Viewports muss mit folgendem Metatag ausgeschaltet sein: <meta name="viewport" content="width=device-width, initial-scale=1.0">
Canvas (Client)	<p>Verwendung eines selbstprogrammierten Canvas an einer passenden Stelle.</p> <ul style="list-style-type: none"> • Auf diesem clientseitigen Canvas müssen mindestens zehn Zeichenoperationen (lineTo, arc, fillRect, etc.) ausgeführt werden. • Der Canvas muss sich während der Anzeige des HTML-Dokuments dynamisch verändern können. Dies, indem die Berechnung auf einen Wert aufbaut, welcher entweder aufgrund einer Benutzereingabe oder der Zeit verändert wird.
Validierung (Client und Server)	<p>Die Webapplikation muss alle Eingaben des Formulars sowohl clientseitig als auch serverseitig sinnvoll validieren, bevor diese verarbeitet werden:</p> <ul style="list-style-type: none"> • Die serverseitige Validierung muss derjenigen der clientseitigen Validierung entsprechen und umgekehrt (**). • Die clientseitige Validierung muss das Absenden eines falsch ausgefüllten Dokuments verhindern. • Clientseitig muss die Webapplikation für alle Eingabefehler eine separate, passende Fehlermeldung direkt im Formular anzeigen. Alle vorhandenen Fehler müssen gleichzeitig angezeigt werden. Fehlermeldungen müssen als DOM-Modifikation implementiert werden und NICHT mit einem Dialog (z.B. alert). <p>(**) Eine Ausnahme bilden serverseitige Validierungen, welche mit Daten arbeiten, welche nur serverseitig vorhanden sind (z.B. in einer Datenbank). Diese Validierungen müssen nicht clientseitig implementiert werden.</p>
Zusatztechnik (Client oder Server)	<p>Implementieren Sie zusätzlich exakt eine Technik aus der folgenden Liste:</p> <ol style="list-style-type: none"> a) Reaktive Eingabevalidierung des Formulars mittels Vue.js. Die gesamten Formulardaten müssen dazu in einem einzelnen Objekt gespeichert werden. Fehlerhafte Eingaben müssen direkt während der Eingabe und nicht erst beim Submit angezeigt werden. Ein noch nicht bearbeitetes Inputfeld soll noch keine Fehlermeldung anzeigen, auch wenn die Eingabe nicht korrekt ist. b) Serverseitige Persistenz mit MySQL: Serverseitige Modifikation und Abfrage von Daten von jeweils mindestens einer Zeile aus einer Tabelle mit mindestens drei Spalten. Die Daten einer Abfrage müssen mittels where gefiltert werden und als Teil der HTTP-Response ausgegeben werden. Das dazugehörige und in phpMyAdmin ausführbare Datenbankschema muss in schema.sql mitgeliefert werden. Verwendung von Prepared-Statements ist zwingend. c) Webservices: Clientseitige Abfrage von Daten aus einem JSON- oder XML-Dokument. Diese müssen von einem externen Webservice abgefragt werden. Von diesem Dokument müssen mindestens drei verschiedene Informationen dem Thema entsprechend in das HTML-Dokument eingefügt werden. <p>Achtung: Sie erhalten keine Zusatzpunkte bei der Implementation von mehr als einer Zusatztechnik.</p>
Abgabeformat	<p>Abgabe als Archiv im ZIP-Format mit Namen: WEBT_FS23_<NACHNAME>_<VORNAME>.zip. (Beispiel: WEBT_FS23_BAETTIG_MARTIN.zip.)</p> <p>Die Zip-Datei muss exakt folgende Verzeichnisstruktur aufweisen:</p> <ul style="list-style-type: none"> • HTML-Dokument (Datei: Name beliebig) • PHP-Skript (Datei: Name beliebig). • schema.sql (Datei: nur falls SQL-Datenbank benötigt). • img (Verzeichnis für alle Bilder). • css (Verzeichnis für alle CSS-Dateien). • js (Verzeichnis für alle Javascript-Dateien). • var (Verzeichnis für alle anderen Dateien).

Hinweis: Im Verzeichnis var können Sie Dateien ablegen, welche die im Abschnitt *Rahmenbedingungen* angegebenen Limiten übersteigen. Diese Dateien werden bei der Bewertung **nicht** beachtet.

4. Abgabe

Die Abgabe erfolgt im ILIAS unter: https://elearning.hslu.ch/ilias/goto.php?target=fold_5754620. Benutzen Sie den Abgabelink Ihrer Durchführung.

Folgende Punkte sind bei der Abgabe zu beachten:

- Abgaben sind final: Mit der Abgabe auf ILIAS geben Sie Ihre Arbeit zur Bewertung frei.
- Abzug bei verspäteter Abgabe: **pro Tag eine halbe Note**. Verspätete Abgaben sind noch möglich bis **maximal zwei Tage nach Abgabefrist**. Anschliessend werden keine Abgaben mehr akzeptiert.

Bei verspäteten Abgaben oder Problemen mit der Abgabe schreiben Sie eine E-Mail an martin.baetig@hslu.ch. Falls notwendig und möglich, erhalten Sie einen neuen Abgabetermin. Nicht selbstverschuldeten Probleme mit dem Ilias-Zugriff dokumentieren Sie bitte mittels eines Screenshots und fügen diesen als Beweis der E-Mail mit an.

5. Fragen

Sollten Sie Fragen zu dieser Aufgabenstellung haben, stellen Sie diese idealerweise im Plenum während den Präsenzzeiten. Wir beantworten Fragen auch per E-Mail. In beiden Fällen werden allfällige zusätzliche Informationen aus Fairness-Gründen auf der «Fragen und Antworten»-Seite im ILIAS veröffentlicht, damit alle Teilnehmer von diesen Informationen profitieren können.

Bitte beachten Sie, dass wir **ab Freitag, 02.06.2023 23:59 keine Fragen mehr zu der Aufgabenstellung** beantworten. Organisatorische (z.B. wo gebe ich ab?) oder technische Fragen (Hilfe, der XAMPP läuft nicht mehr!) beantworten wir ohne Zeitbeschränkung.

Hinweis: Fragen der Art «erfüllt folgender Code X die Anforderung Y» beantworten wir nicht.

6. Plagiate

Dies ist eine Einzelarbeit, welche Sie **selbstständig erarbeiten müssen**:

- Wir haben viel Erfahrung mit Plagiaten und sehen z.T. Details, welche Sie nicht sehen werden, wenn Sie beispielsweise grössere Teile der Lösungen von Mitstudierenden oder Code aus dem WWW übernehmen.
- Wir vergleichen die Abgaben **automatisiert** mit Abgaben aus diesem und allen vorangehenden Semestern ab FS20 sowie bekannten Codefragmenten aus dem WWW.
- Übernahme von kleineren Programmfragmenten (z.B. aus dem WWW) ist erlaubt, sofern:
 - Das Fragment maximal 15 Zeilen (abzüglich Leerzeilen und Kommentare) umfasst.
 - Und mindestens 5 Zeilen des Fragments auf Ihr Thema angepasst wurden.
- Die Verwendung von KI-Programmgeneratoren (z.B. ChatGPT) ist nicht erlaubt.
- Kleinere Verstösse führen zu Punktabzug, grössere Verstösse zu Nichtbestehen inkl. Aktenvermerk.
- Bei Verdacht, dass die Arbeit oder Teile davon von jemand anderes geschrieben wurden, werden Sie zur Klärung zu einem persönlichen Gespräch aufgeboten.

7. Termine

Datum und Zeit	Termin
In SW12	Auftragsausgabe und Vorbesprechung jeweils zu Beginn der Unterrichtseinheit.
02.06.2023 23:59	Letzte Möglichkeit für Fragen an die Dozierenden mit Bezug zur Aufgabenstellung. Hinweis: Bis zu diesem Datum sollten Sie den grössten Teil Ihrer Webapplikation fertiggestellt haben.
16.06.2023 23:59	Letztmöglicher Termin für die Abgabe des MEP Abschlussprojekts.

Viel Erfolg wünschen Ihre Dozenten

Martin Bättig (Modulverantwortlicher), Silvan Wegmann und David Lichtsteiner

Appendix

A. Bewertung

Die Bewertung erfolgt anhand vordefinierter Kriterien (siehe nachfolgende Tabelle). Pro Bewertungskriterium gibt es ein Maximum an Punkten, welches Sie erreichen können. In der Summe über alle Kriterien können Sie maximal 50 Punkte erreichen. **Um das Modul zu bestehen, müssen Sie mindestens 32 Punkte erzielen.**

Bewertungskriterium	Maximale Punkte
HTML: Navigationsleiste	2
HTML: Informationsabschnitt	2
HTML: Formular	5
HTML: Verwendung	4
PHP: Effekt und HTML-Antwort	4
PHP: Cookie	3
PHP: Hyperlink	1
Darstellung / Layout: Responsives Layout	4
Darstellung / Layout: CSS Transitionen	2
Canvas	5
Validierung: Client	5
Validierung: Server	5
Zusatztechnik	6
Einhaltung des Abgabeformats	1
Einhaltung der Limiten	1
Total	50

B. Beispiel eines in phpMyAdmin ausführbaren schema.sql

Beispiel eines in phpMyAdmin ausführbaren schema.sql (copy und paste in das SQL-Tab und click auf OK):

```
create database tasks;
```

```
use tasks;
```

```
create table tasks(  
    user varchar(200),  
    task varchar(1000)  
);
```