

Prüfungsleitfaden

Summative Prüfungen WBD5100



1. Praktische Projektarbeit Custom CMS „Online-Shop“

Format:	Praktische Projektarbeit
Abgabemodalitäten:	Abnahme durch Fachbereichsleitung oder Supervisor
Einzel-/Gruppenarbeit:	Einzelarbeit
Typ:	Praktisch
Gewichtung:	60%
Abgabeart:	Online-Upload

1.1. Überblick und Aufgabenstellung

Ein Start-Up Unternehmen beauftragt euch mit der Umsetzung eines Online-Shops mittels PHP & MySQL. Das Unternehmen will auch bezüglich des Designs der Site ein Höchstmaß an Identifikation mit der Marke bzw. dem angebotenen Produkt/der Dienstleistung. Das Projekt sollte mehrere Unterseiten beinhalten (mindestens vier), von denen einige öffentlich zugänglich und andere nur für registrierte Nutzer erreichbar sind.

Zusätzlich soll jeder registrierte Nutzer (z.B. Admin) die Möglichkeit haben, Inhalte zu editieren und neue hinzuzufügen. Die Seite sollte über eine direkte Datenbankanbindung verfügen, durch die Userdaten, Seiteninhalte etc. gespeichert werden.

1.2 Pflichtvorgaben:

- Erstelle ein **responsives** Layout, dessen Design sich mit der Marke/Produkt/Dienstleistung identifiziert
- HTML muss semantisch korrekt und W3C valide sein
- es dürfen keine CSS- oder sonstige Frameworks verwendet werden
- Erstelle eine Datenbank, die alle relevanten Daten beinhaltet (z.B. Nutzerdaten, Artikel, Seiteninhalte etc.)
- Folgende Seiten müssen vorhanden sein:
 - ♦ Für unregistrierte User:
 - » **Produktübersicht**
 - » **Produktdetailseite:** dort ist es dem User möglich, Informationen über das Produkt zu holen und gegebenenfalls das Produkt mit der gewünschten Anzahl in den Warenkorb zu legen
 - » **Warenkorb:** User haben hier die Möglichkeit, ausgewählte Produkte zu bearbeiten. D.h. Produkte können hier wieder entfernt werden bzw. die Anzahl pro Produkt kann geändert werden.

- » **Login-Formular mit Validierung (mindestens serverseitig)**, über welches sich registrierte Nutzer einloggen
- » **Registrierungs-Formular mit Validierung (mindestens serverseitig)**, über das sich unregistrierte Nutzer anmelden können. Das Formular sollte mindestens Benutzername, Passwort, E-Mail-Adresse, Name, Vorname und Adresse als Felder zum Ausfüllen voraussetzen, die korrekt validiert werden)
- ♦ Für registrierte und eingeloggte Nutzer (Standarduser)
 - » **Bestellübersicht:** kann nur erreicht werden, wenn der User bereits registriert bzw. eingeloggt ist. Hier soll auf einen Blick ersichtlich sein, was bestellt worden ist, in welcher Anzahl und wohin geliefert wird, wobei als Lieferadresse die vom User beim registrieren angegebene Adresse angezeigt werden soll
 - » Sobald ein Benutzer eingeloggt ist, soll er seine User-Daten in der Datenbank editieren können.
- ♦ Für Admin User
 - » **Produkt-CMS:** Möglichkeit Produkte anzulegen und zu bearbeiten, folgende Parameter müssen mindestens administrierbar sein: Bezeichnung, Beschreibung, Foto (Upload), Preis
 - » **Übersicht aller Bestellungen:** der Admin soll hier die Möglichkeit haben, einzelne Bestellungen zu bearbeiten bzw. zu stornieren.
 - » **Übersicht aller User:** der Admin soll hier die Möglichkeit haben, einzelne User zu bearbeiten bzw. zu löschen. (evtl. „softdelete“)

Nutze grundlegende Verschlüsselung beim Transfer von Daten von und zur Datenbank. Außerdem erstelle ein README.pdf, README.md oder README.txt, in denen du das Design deiner Datenbank und das PHP-Konzept auf der Webseite detailliert darstellst. Bitte auch die URLs, falls das Projekt online abrufbar ist und Zugänge in die Readme-Datei schreiben. Datenbank-Dump der abgegebenen Datei hinzufügen nicht vergessen, da das Projekt ansonsten nicht bewertet werden kann.

1.3 Optional:

- Dem User wird im Warenkorb bzw. vorm Bestellabschluß ermöglicht eine alternative Lieferadresse anzugeben.
- Live-Preisberechnung in Javascript. Ändert sich etwas im Warenkorb, wird die Preisanzeige live angepasst.
- Man kann im Produkt-CMS mehrere Fotos pro Produkt hochladen, die in der Produktdetailseite als bspw. Slider angezeigt werden
- Kategorisierung von Produkten (Damit man bspw. zwischen einer CD und einem Merchartikel differenzieren kann)

- Limitierung von Produkten – Es können nur so viele Produkte bestellt werden, wie im Backend festgelegt.
- clientseitige Formularvalidierung
- alternativ kann das Projekt auch mit dem MVC Framework, welches im Unterricht verwendet wurde mit OOP umgesetzt werden

1.3 Bewertungskriterien

Programmierprinzipien

- Ist der PHP-Code vielseitig und durchwegs fehlerfrei?
- Ist Datensicherheit durch die Datenverschlüsselung gewährleistet?
- Wurde der Code nachvollziehbar kommentiert?

Datenbankstruktur und Datenmanipulation

- Ist die Datenbankstruktur durchdacht und zweckdienlich?
- Wurde die Datenmanipulation sinnvoll umgesetzt und entspricht sie grundlegenden Sicherheitsansprüchen?

Softwarearchitektur

- Erfüllt die Ordnerstruktur die Anforderungen an nachvollziehbarer Softwarearchitektur?

Gesamtprojekt

- Wurde der Umfang eingehalten?
- Wurde aus den relevanten Komponenten eine gut benutzbares und technisch sauber umgesetztes Projekt erstellt?
- Wurden alle notwendigen Dateien der Abgabe hinzugefügt und mitgegeben.

1.4 Lernziele

Nach erfolgreicher Absolvierung dieser Aufgabe ist man in der Lage:

- Komplexe Webanwendungen zu erstellen und strategisch an Sicherheitsprobleme heranzugehen
- Funktionen serverseitig umzusetzen
- Daten von einem Server zu empfangen und an diesen zu senden;
- Ein sinnvolles Datenbank-Design zu entwickeln und Daten mittels SQL-Queries zu manipulieren;
- Weitergehende Funktionalitäten durch die Verbindung von PHP mit HTML, CSS und JS umzusetzen.



2. Präsentation : Custom CMS

Format:	Präsentation
Abgabemodalitäten:	Präsentation, Online-Upload
Einzel-/Gruppenarbeit:	Einzelarbeit
Typ:	Praktisch und schriftlich
Gewichtung:	20%

2.1 Überblick und Aufgabenstellung

Die Studenten sollen in einer kurzen Präsentation einen Überblick über das Ergebnis der Summativen Projektarbeit “Custom CMS” geben.

Das Projekt soll innerhalb von 5-10 Minuten vorgestellt werden.

Ein Fokus der Präsentation soll auf den wichtigsten Bestandteilen und Funktionen der Webseite liegen: Frontend und Backend, Datenspeicherung und -abruf vom Webserver.

Gleichmaßen sollen auch die eigene Herangehensweise, die aufgetretenen Probleme und ihre Lösungen demonstriert und kritisch beleuchtet werden.

Diese Interaktionselemente können zum Beispiel sein:

- Ein- und Ausblenden und Ein- und Ausklappen von Bestandteilen der Webseite (z.B. Textabsätze, Info-Boxen, Navigation etc.);
- Hover-Effekte und -Animationen, Cursor-Änderungen, Bildertausch;
- Animationen von Schrift- und Hintergrundfarben und Hintergrundgrafiken.

Zudem soll die Webseite um eine Auswahl an userfreundlichen Funktionen ergänzt werden. Diese können zum Beispiel sein:

- Responsives Ändern und Anpassen von Bestandteilen (z.B. Hintergrundbild, Icons) der Webseite entsprechend der ausgelesenen Fenstergröße;
- Einbinden von ausgewählten Elementen aus Javascript-Libraries, um das Nutzererlebnis zu steigern.

2.2 Vorgaben

- Dauer der Präsentation 5-10 Minuten, exklusive anschließender Fragen und Diskussion;
- Präsentation eines vollständig selbst erstellten CMS.

2.3 Lernziele

Nach erfolgreicher Ausführung des Assignments sollte man dazu in der Lage sein:

- Einen kurzen und prägnanten Überblick über ein eigenes Projekt zu geben;
- Ein CMS mit seinen Bestandteilen und Funktionalitäten in Frontend und Backend verständlich und nachvollziehbar zu erklären;
- Die eigenen Ideen, Vorgehensweisen und Wege zur Problemlösung zu kommunizieren und kritisch zu bewerten;

2.5 Bewertungskriterien

- Wurde ein prägnanter Überblick über das Projekt gegeben
- Ist ein roter Faden erkennbar
- Ist der Vortragende vorbereitet und spricht deutlich und klar
- Ist die Präsentation übersichtlich und Informativ strukturiert?
- Gesamteindruck



3. Praktische Prüfung : PHP & MySQL

Format:	Praktische Prüfung
Abgabemodalitäten:	Prüfung im Klassenverband
Einzel-/Gruppenarbeit:	Einzelarbeit
Typ:	Praktisch
Gewichtung:	20%

3.1 Überblick und Aufgabenstellung

Eine kleine vorgegebene Webseite soll so überarbeitet und erweitert werden, dass sie voll funktionsfähig ist. Dazu müssen zum einen die Datenbankstruktur vervollständigt und weitere spezielle Funktionalitäten und Features ergänzt werden. Der Code soll gut strukturiert und sinnvoll kommentiert sein. Im Unterricht vermittelte Design-Muster in Bezug auf das Datenbank-Management sollen berücksichtigt werden.

Eine genaue Aufgabenstellung mit konkret aufgelisteten und beschriebenen Aufgabenbestandteilen wird zu Prüfungsbeginn ausgegeben.

3.2 Vorgaben

- 90 Minuten im Klassenverband.
- Umsetzung aller Aufgabenbestandteile

3.3 Lernziele

- Daten mit Hilfe von PHP- und MySQL-Techniken abzurufen, zu speichern und zu manipulieren;
- PHP mittels OOP und MVC in Syntax und Logik fehlerfrei zu programmieren.