

PROJEKTBERICHT

Aufgabenstellung zum Kurs: DLBITPEWP01-01 – Projekt:
Einstieg in die Web-Programmierung

INHALTSVERZEICHNIS

1.	Aufgabenstellung	2
1.1.	Aufgabenstellung 1: Konzeption und Umsetzung eines einfachen Webforums	2
1.2.	Aufgabenstellung 2: Konzeption und Umsetzung eines einfachen Onlineshops	3
1.3.	Aufgabenstellung 3: Konzeption und Umsetzung eines einfachen Bewerbermanagementportals	4
2.	Zusatzinformationen zur Bewertung des Projektberichts	5
3.	Betreuungsprozess	6

1. AUFGABENSTELLUNG

Für den Projektbericht stehen folgende Aufgabenstellungen zur Auswahl. Bitte entscheide Dich für eine davon, die Du in Deinem Projektbericht bearbeiten möchtest.

Hinweis zum Urheberrecht und zur Plagiatsprüfung:

Es wird darauf hingewiesen, dass der IU Internationale Hochschule GmbH das Urheberrecht der Prüfungsaufgaben/Aufgabenstellungen obliegt. Einer Veröffentlichung der Aufgabenstellungen auf Drittplattformen wird ausdrücklich widersprochen. Im Falle einer Zu widerhandlung stehen der Hochschule u.a. Unterlassungsansprüche zu. Zudem weisen wir darauf hin, dass jede eingereichte schriftliche Ausarbeitung mittels einer Plagiatssoftware überprüft wird. Wir empfehlen daher auch, keinesfalls ausgearbeitete Lösungen zu teilen, da dies den Verdacht eines Plagiates begründen kann.

Hinweis auf Datenanonymisierung:

Wir weisen darauf hin, dass grundsätzlich eine Anonymisierung von personenbezogenen Daten erfolgen soll, sodass kein Rückschluss auf eine bestimmte oder bestimmbare Person möglich ist. Dies gilt ebenso für unternehmensbezogene Daten.

1.1. Aufgabenstellung 1: Konzeption und Umsetzung eines einfachen Webforums

Hintergrund: Online-Communities profitieren von schlanken, leicht bedienbaren Foren, in denen Themen und Beiträge strukturiert erfasst und gefunden werden können. Für einen funktionsfähigen Minimalumfang sind Registrierung, Login/Logout, die Anzeige von Themen und Beiträgen sowie einfache Verwaltungsfunktionen erforderlich. Bereits in der frühen Konzeption sind Datenschutz, Sicherheit, Skalierung und der passende Technologie-Stack (PHP oder Java, Datenbankwahl, Serverumgebung) zu berücksichtigen.

Aufgabenstellung: Entwickle ein einfaches Webforum. Gehe dabei in folgender, aufeinander aufbauender Reihenfolge vor:

- Überlege Dir zunächst Zielgruppe(n), Rollen sowie die Zugriffsregeln, beispielsweise ob Gäste nur lesen dürfen oder ob ausschließlich registrierte Nutzer:innen Zugriff erhalten und ob Moderator:innen bzw. Administrator:innen vorgesehen sind.
- Konkretisiere anschließend den Funktionsumfang des Forums, indem Du festlegst, dass Registrierung, Login/Logout sowie das Erstellen, Anzeigen, Bearbeiten und Löschen von Themen und Beiträgen möglich sein soll.
- Skizziere die Grundlagen zu Datenschutz und Sicherheit, indem Du etwa sicheres Passwort-Hashing, Session-Management, Eingabevalidierung sowie Schutz vor XSS, CSRF und SQL-Injection einplanst (**Diese müssen aber nicht implementiert werden!**).
- Erstelle im nächsten Schritt UI-Skizzen oder Mockups für z. B. Intro-/Login-/Registrierungsseiten, Forum-Übersicht und-Detailansicht.
- Entwirf schließlich ein grobes Datenmodell mit den wesentlichen Entitäten wie Nutzer:innen, Themen, Beiträge und Rollen beziehungsweise Berechtigungen.
- Lege die Programmiersprache und die Technologien fest, z. B. **HTML, PHP oder Java**, und begründe diese Wahl aus fachlicher und technischer Sicht. Trifft eine Entscheidung zur Datenhaltung, indem Du beispielsweise SQLite als einfachen Einstieg auswählst und die Abwägungen im Hinblick auf Nutzerzahl, Betrieb, Wartung und Skalierbarkeit erläutert sowie Alternativen wie MySQL oder PostgreSQL einordnest.
- Plane die Server- und Testumgebung, indem Du entscheidest, ob Du z. B. XAMPP Packages, eine lokale Linux-VM über VirtualBox oder das IU Virtual Lab verwendest, und beschreibe die dafür benötigten Komponenten.

- Skizziere die Software-Architektur, lege außerdem Kriterien für die Performance, Wartbarkeit und Sicherheit fest.
- Entwickle auf Basis deiner Festlegungen, Skizzen und Technologien einen MVP des Webforums.

1.2. Aufgabenstellung 2: Konzeption und Umsetzung eines einfachen Onlineshops

Hintergrund: Ein schlanker Onlineshop umfasst eine Startseite mit Angeboten, eine flexible Kategoriestruktur, Produktlisten und -details, einen Warenkorb, den Check-out sowie ein einfaches Management für Kategorien und Artikel. Zahlungen werden zweckmäßig über etablierte Zahlungsdienstleister wie Stripe oder PayPal abgewickelt (**Diese müssen aber nicht implementiert werden!**). Neben der technischen Architektur sind Datenschutz und regulatorische Anforderungen wie DSGVO, PCI-DSS / PSD2 zu berücksichtigen.

Aufgabenstellung: Entwickle einen funktionsfähigen Onlineshop und arbeite die folgenden Schritte in der angegebenen Reihenfolge ab:

- Lege die Zielgruppe(n), die Rollen und die zentralen Use Cases fest, indem Du z. B. für Kund:innen das Stöbern, das Anzeigen von Produktdetails, die Nutzung des Warenkorbs und den Check-out sowie für Administrator:innen die Pflege von Kategorien und Produkten definierst.
- Präzisiere den Funktionsumfang des Shops, indem Du eine Startseite, eine Kategorienübersicht, Produktlisten und -detailseiten, einen Warenkorb, die Registrierung und Anmeldung sowie den Check-out festlegst.
- Plane die Zahlungsabwicklung durch die Einbindung eines Zahlungsanbieters wie Stripe oder PayPal und berücksichtige die Compliance-Aspekte, insbesondere DSGVO, PCI-DSS und PSD2, mit einer kurzen, schlüssigen Begründung.
- Erstelle UI-Skizzen oder Mockups, die die unterschiedlichen Seiten und Bereiche visualisieren.
- Entwirf ein grobes Datenmodell mit Entitäten wie Nutzer:innen, Kategorien, Produkte, Bestellungen, Bestellpositionen und Zahlungen.
- Lege die Programmiersprache und die Technologien fest, z. B. **HTML, PHP oder Java**, und begründe diese Wahl aus fachlicher und technischer Sicht. Triff eine Entscheidung zur Datenhaltung, indem Du beispielsweise SQLite als einfachen Einstieg auswählst und die Abwägungen im Hinblick auf Nutzerzahl, Betrieb, Wartung und Skalierbarkeit erläuterst sowie Alternativen wie MySQL oder PostgreSQL einordnest.
- Plane die Server- und Testumgebung, indem Du entscheidest, ob Du z. B. XAMPP Packages, eine lokale Linux-VM über VirtualBox oder das IU Virtual Lab verwendest, beschreibe die dafür benötigten Komponenten.
- Skizziere die Software-Architektur, lege außerdem Kriterien für die Performance, Wartbarkeit und Sicherheit fest.
- Entwickle auf Basis Deiner Festlegungen, Skizzen und Technologien einen MVP des Webshops.

1.3. Aufgabenstellung 3: Konzeption und Umsetzung eines einfachen Bewerbermanagementsystems

Hintergrund: Ein schlankes Bewerbermanagementsystem (ATS) umfasst eine Karriereseite mit Stellenübersicht und -details, ein bewerberfreundliches Formular zur Einreichung von Unterlagen, eine transparente Bewerbungs-Pipeline mit Statusverfolgung sowie einen einfachen Verwaltungsbereich für Stellen und Kandidatenprofile. Optionale Integrationen in externe Dienste wie Kalender, E-Mail-Provider oder Jobbörsen können vorgesehen werden (**Diese müssen aber nicht implementiert werden!**). Neben der technischen Architektur sind Datenschutz und regulatorische Anforderungen wie DSGVO, Aufbewahrungs- und Löschfristen, Einwilligungen zur Datenverarbeitung sowie Antidiskriminierungsvorgaben (z. B. AGG) zu berücksichtigen.

Aufgabenstellung: Entwickle ein funktionsfähiges Bewerbermanagementsystem (ATS) und arbeite die folgenden Schritte in der angegebenen Reihenfolge ab:

- Lege die Zielgruppe(n), die Rollen und die zentralen Use Cases fest, indem Du z. B. für Bewerber:innen das Finden von Stellen, das Anzeigen von Stellendetails, das Ausfüllen des Bewerbungsformulars definierst.
- Für Recruiter:innen soll das Anlegen und Verwalten von Stellen, das Sichten und Bewerten von Bewerbungen, das Verschieben entlang einer Pipeline und die Dokumentation von Notizen möglich sein.
- Präzisiere den Funktionsumfang des Systems, indem Du eine Karrierestartseite, eine Stellenliste, Stellendetailseiten, ein Bewerbungsformular mit Upload von Anhängen (z. B. PDF-Lebenslauf) sowie einen Adminbereich für Stellen- und Kandidatenmanagement mit Statuswechseln festlegst.
- Plane auch die rechtlichen Aspekte sowie die relevanten Compliance-Anforderungen (insbesondere DSGVO, Einwilligungen, Löschkonzept, Antidiskriminierung) mit einer kurzen, schlüssigen Begründung ein.
- Erstelle UI-Skizzen oder Mockups, die die unterschiedlichen Seiten und Bereiche visualisieren.
- Entwirf ein grobes Datenmodell mit Entitäten wie z. B. Nutzer:innen (Bewerber:innen, Recruiter:innen, Admin), Stellen, Bewerbungen, Pipeline-Status, Anhängen/Dokumenten etc.
- Lege die Programmiersprache und die Technologien fest, z. B. **HTML, PHP oder Java**, und begründe diese Wahl aus fachlicher und technischer Sicht. Triff eine Entscheidung zur Datenhaltung, indem Du beispielsweise SQLite als einfachen Einstieg auswählst und die Abwägungen im Hinblick auf Nutzerzahl, Betrieb, Wartung und Skalierbarkeit erläuterst sowie Alternativen wie MySQL oder PostgreSQL einordnest.
- Plane die Server- und Testumgebung, indem Du entscheidest, ob Du z. B. XAMPP Packages, eine lokale Linux-VM über VirtualBox oder das IU Virtual Lab verwendest, beschreibe die dafür benötigten Komponenten.
- Skizziere die Software-Architektur, lege außerdem Kriterien für die Performance, Wartbarkeit und Sicherheit fest.
- Entwickle auf Basis Deiner Festlegungen, Skizzen und Technologien einen MVP des ATS.

2. ZUSATZINFORMATIONEN ZUR BEWERTUNG DES PROJEKTBERICHTS

Bei der Konzeption und Erstellung des Projektberichts sollten die im Prüfungsleitfaden aufgeführten Bewertungskriterien und Erläuterungen berücksichtigt werden.

Transfer: Im Vordergrund steht die praktische Anwendung von Web-Engineering-Grundlagen zur Konzeption und Umsetzung einer funktionsfähigen Webanwendung. Beschreibe nachvollziehbar, wie du Anforderungen erhoben und priorisiert hast, wie daraus ein Datenmodell, eine Software-Architektur und ein Betriebskonzept abgeleitet wurden und wie Du zentrale Aspekte wie Sicherheit, Performance und Wartbarkeit praktisch umgesetzt hast. Begründe Deine Technologieentscheidungen (z. B. Sprache, Framework, Datenbank, Server-/Testumgebung) und erläutere, welche Alternativen Du erwogen, verworfen oder für spätere Ausbaustufen vorgesehen hast. Eine rein theoretische Darstellung steht nicht im Fokus; entscheidend ist die begründete, praktische Anwendung.

Dokumentation: Beachte die formalen Vorgaben des Projektberichts im Prüfungsleitfaden. Dokumentiere Dein System mit verständlichen Artefakten, z. B. UI-Skizzen/Mockups, einem ER-Diagramm des Datenmodells, einer Architekturübersicht, sowie aussagekräftigen Screenshots oder einer kurzen Bildschirmaufnahme des MVP.

Ressourcen: Nutze verfügbare Zeit und Materialien effizient und benenne die verwendeten Ressourcen transparent, etwa eigene Vorkenntnisse, offizielle Dokumentationen, Tutorials, Bibliotheken/Frameworks, Sandbox-/Testdienste, Tools für Build/Deployment/Tests sowie Rückmeldungen aus Tutorien oder Lerngruppen. Verwende ausschließlich synthetische Daten und anonymisiere reale Bezüge konsequent. Wenn ein Organisationskontext herangezogen wird, nenne nur Rahmendaten (z. B. Branche, Größenordnung der Mitarbeitenden, Anzahl Standorte), ohne Rückschlüsse auf Personen oder Unternehmen zu ermöglichen.

Prozess: Beschreibe Dein Vorgehen entlang der geforderten Schritte von der Anforderungsklärung über Architektur- und Technologieentscheidungen bis zur MVP-Implementierung. Lege dar, wie Du iterativ gearbeitet, frühzeitig Kernflüsse lauffähig gemacht, getestet und Probleme gelöst hast. Reflektiere Dein Vorgehen kritisch: Was hat gut funktioniert, was weniger, welche Risiken sind aufgetreten und wie würdest Du beim nächsten Mal vorgehen?

Kreativität: Stelle eigenständige, begründete Verbesserungen heraus, die den Kernzweck der Anwendung zielgerichtet stärken, z. B. ein durchdachtes Rollen- und Berechtigungskonzept, sinnvolle Suche/Filter und Pagination, klare Navigations- und Feedbackmechanismen, Barrierefreiheitsaspekte, einfache Automatisierungen oder Monitoring/Logging in angemessenem Umfang. Vermeide ungerichteten Funktionszuwachs ohne erkennbaren Mehrwert.

Qualität: Zeige, dass das Ergebnis die beschriebenen Anforderungen zuverlässig erfüllt und im vorgesehenen Kontext betreibbar ist. Belege dies durch Funktions- und, wo sinnvoll, einfache Last- oder Performancetests, sowie eine kurze Usability-Einschätzung zentraler Nutzerflüsse. Weise Sicherheitsmaßnahmen nach, z. B. Eingabevervalidierung, sichere Authentifizierung und Session-Management, Schutz vor XSS/CSRF/SQL-Injection, sichere Datei-Upserts (falls relevant) oder Rechteprüfung. Berücksichtige Datenschutz und Compliance kontextbezogen (insbesondere DSGVO, Einwilligungen, Datenminimierung, Aufbewahrungs- und Löschkonzept; bei Zahlungen die Delegation an Zahlungsdienstleister). Achte zudem auf Codequalität (Struktur, Lesbarkeit, Fehlerbehandlung, Konfigurierbarkeit) und auf Reproduzierbarkeit (klarer Setup-Prozess, Beispielkonfigurationen, lauffähiger MVP).

3. BETREUUNGSPROZESS

Für die Betreuung des Projektberichts stehen grundsätzlich mehrere Kanäle offen. Die jeweilige Inanspruchnahme liegt dabei im eigenen Verantwortungsbereich. Die Tutor:innen stehen für fachliche Rücksprachen zur Themenwahl einerseits sowie für formale und allgemeine Fragen zum wissenschaftlichen Arbeiten andererseits zur Verfügung. Eine Abnahme von Gliederungen, Textteilen oder -entwürfen durch die Tutor:innen ist hierbei jedoch nicht vorgesehen, da die eigenständige Erstellung Teil der zu erbringenden Prüfungsleistung ist und in die Gesamtbewertung einfließt. Es werden jedoch Hinweise zu Gliederungsentwürfen gegeben, um den Einstieg in die Strukturierung einer wissenschaftlichen Arbeit zu erleichtern.