

# Projektspezifikation

## „Hotel mit Zimmerreservierungen“

### I. Allgemein

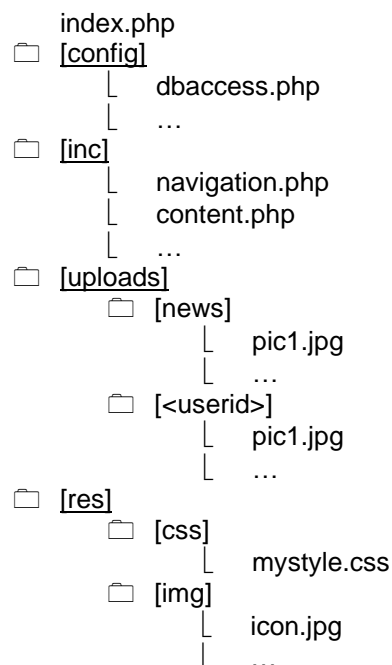
- a. Zum Abschluss der LV Webtechnologien muss ein Web-Projekt umgesetzt werden. Die Gewichtung der einzelnen Aufgaben und die Punkteverteilung finden Sie in der zugehörigen Bewertungsmatrix.
- b. Verwenden Sie für die Umsetzung Teile Ihrer Übungen wieder.
- c. Die Bearbeitung erfolgt in 2er Teams. Einzelarbeiten sind nur in Ausnahmefällen zulässig.
  - i. **Blau markierte Aufgaben sind nur von 2er Teams zu bearbeiten, Einzelarbeiten lassen diese aus! (Punkte für blau markierte Aufgaben werden nicht zu der Gesamtpunktezahl für Einzelprojekte addiert)**
- d. Das Abnahmegespräch erfolgt dann im Team (bzw. im Ausnahmefall einzeln). Jedes Teammitglied muss das Projekt (Funktionsumfang **und** Code) kennen und Fragen dazu beantworten können.
  - i. Das Abnahmegespräch orientiert sich an einer Bewertungsmatrix. Diese steht vorab zur Verfügung und dient u.a. der Darstellung der Gewichtung der einzelnen Aufgaben sowie zur Kontrolle der Vollständigkeit.
- e. **Für manche in grün markierte Tasks, gibt es Extrapunkte zu erreichen. Diese Aufgaben sind optional.**
- f. Die Angabe lässt gewisse Gestaltungs- und Implementierungsfreiheiten zu. Folgende **Basis-Features** müssen abgebildet werden:
  - i. Registrierung
  - ii. Zimmerreservierung
  - iii. Buchungsverwaltung durch Administrator\*innen

- g. Die Umsetzung des Projekts wird **sämtliche LV-Inhalte** abdecken:
  - i. HTML5, CSS3, Bootstrap
  - ii. PHP
  - iii. Fileupload
  - iv. MySQLi + Prepared Statements
  - v. Authentifizierung
- h. Achten Sie auf eine **gute User Experience** innerhalb der Anwendung (keine Whitepages, keine Fehlerzustände, klare Abläufe, guter Gesamteindruck, ...)

## II. Projektstruktur und Grundlayout

- a. Erstellen Sie eine Seite **index.php**, welches das **Grundlayout** enthält und die Hauptbereiche der Hotelverwaltung abdeckt sowie CSS-Files einbindet. Die Inhalte aller Bereiche werden mittels ***include*** eingebunden. (Header, Navigation, Hauptinhaltsbereich, ...).
- b. Die Projektstruktur der Hotelverwaltung soll die **Komponenten** klar **unterteilen** (siehe Beispielstruktur).

### Beispielstruktur:



- c. Legen Sie eine neue Datenbank für das Projekt an.
- d. Legen Sie ein ***dbaccess.php*** File an, welches **zentral** die Zugangsdaten auf die Datenbank beinhaltet.

- e. Achten Sie beim Implementieren darauf, dass **Sicherheitslücken** möglichst **geschlossen** werden.

### III. Frameworks und Libraries

- a. Es bleibt Ihnen überlassen, ob Sie für die Implementierung diverse Frameworks oder Bibliotheken verwenden, wie z.B. Bootstrap, YAML, ...
- b. Die Verwendung von Fertiglösungen (wie CMS o.ä.) ist nicht erlaubt!

### IV. Abgabemodalitäten

- a. Das Projekt ist **vollständig** als zip/rar File in Moodle hochzuladen (maximal Dateigröße 20MB), inklusive einer Kopie der MySQL-Datenbank (sql-Anweisungen) und einem Hinweis auf die Teammitglieder im Dateinamen, bspw:
  - i. **WEB\_Hotel\_Nachname\_Nachname.zip**
    - Halten Sie sich unbedingt an die Nomenklatur, um die korrekte Zuweisung und somit Bewertung von gemeinsam abgegeben Projekten sicherzustellen
    - Verkleinern Sie Bilddateien/Videos ggf., um die Größe des Projekts für den Upload zu reduzieren
  - b. **Kommentieren** Sie komplexere Abläufe/Funktionen im Code
  - c. Falls weitere Konfigurationsschritte zur Inbetriebnahme des Projekts nötig sind, legen Sie ein **README.txt**-File bei (z.B. Zugangsdaten zur Datenbank).

### Semester Arbeitsplan

Um Ihnen die Bearbeitung des Projekts sowie die Aufteilung des Workloads über das Semester zu erleichtern, wurden alle Tasks entsprechend den wöchentlichen Themen in die Bausteine 1-13 aufgeteilt.

Baustein 1	Thema	← Jeder Baustein berücksichtigt die aktuellen Inhalte des jeweiligen Blocks. Die Bausteine sind entsprechend beschriftet mit Nummer und Thema.
------------	-------	--

Beachten Sie, dass die meisten Oberflächen (Eingabemasken, Anzeigen, ...) am Ende an eine Datenbank anzubinden sind. Berücksichtigen Sie daher beim Umsetzen, dass einige Projektteile stetig erweitert werden.

## Grundidee

**Registrierte User\*innen** der Webseite können Zimmer reservieren und Ihre Reservierungsdetails einsehen.

**Administrator\*innen** der Website haben die Möglichkeit **News-Beiträge** zu posten, welche in einem eigenen Bereich der Webseite zu sehen sind.

**Anonyme User\*innen** können die **News-Beiträge** der **Administrator\*innen** (auch ohne eingeloggt zu sein) auf der Webseite sehen.

Schwerpunkte sollen sein:

- **User-Rollen:**
  - anonym (nicht eingeloggt)
  - registrierter User
  - Administrator\*in
- **Datenbank:** Grundlegende Verwaltung aller Daten in einer Datenbank (User-Rollen, News-Beiträge, Reservierungen, etc.) → Aufbau einer geeigneten Datenbankstruktur, Zugriff mittels PHP.
- **Posten von News-Beiträgen:** Administrator\*innen können News-Beiträge [inkl. Bildupload](#) auf der Startseite posten.
- **Registrierung:** Anonyme User\*innen können sich auf der Webseite registrieren.
- **Zimmer reservieren:** Registrierte User\*innen können Zimmer reservieren und folgende Leistungen optional mit Aufpreis dazu-buchen:
  - Frühstück
  - Parkplatz
  - Mitnahme von Haustieren

Die Höhe der jeweiligen Aufpreise kann vom Projektteam frei (jedoch sinnvoll) selbst gewählt werden.

Für die Reservierung der einzelnen Zimmer ist auch deren zeitliche Verfügbarkeit zu berücksichtigen!

- weitere Bereiche der Webseite:
  - **Hilfe** (Anleitung bzw. FAQs)
  - **Impressum** (Standard-Impressum), **Namen und Bilder des Projektteams**.

Die gesamte Website muss **responsive** sein und somit für Smartphones, Tablets und Desktop-Clients verwendbar sein. Testen Sie dies mit den jeweiligen Developer Tools Ihres Browsers.

Für eine außergewöhnlich gute **Usability**, Grundzüge der **Accessibility** und **Suchmaschinenoptimierung** werden Zusatzpunkte gewährt.

## User-Rollenverwaltung

Es gibt 3 verschiedene Rollen (Usertypen):

- anonyme User\*in (nicht eingeloggt)
- registrierter und eingeloggter User\*in
- Administrator\*in

Alle Usertypen sehen:

- Startseite
- Hilfe/FAQs
- Impressum
- News-Beiträge der Administrator\*innen

Was kann der/die **anonyme User\*in**?

- Startseite ansehen
- News-Beiträge ansehen
- Hilfe-Seite aufrufen und lesen
- Impressum aufrufen und lesen
- Registrierung durchführen

Was kann der/die **registrierte und eingeloggte User\*in** zusätzlich zum/r anonymen User\*in?

- Zimmerreservierung durchführen
- Reservierungsdetails einsehen´
- Profilverwaltung (Stammdaten ändern)

Was kann der/die **Administrator\*in** zusätzlich zum anonymen User\*in?

- News-Beiträge (**inkl. Bildupload**) erstellen **und löschen**
- Reservierungsverwaltung (Übersicht über alle Reservierungen)
- Userverwaltung (Übersicht über registrierte und deaktivieren einzelner User\*innen)

## Aufbau der Datenbank

### Datenbankmodell:

Überlegen Sie sich ein **geeignetes Datenbankmodell**

Hinweise: Sie müssen User\*innen und ihre Rolle (registriert bzw. Admin) speichern, News-Beiträge verwalten (etwaige Bilder selbst liegen im Filesystem des Webserver/Projektordners, die Datenbank verwaltet nur Referenzen auf die Bilder).

### Datenbankzugriff mittels PHP:

Der Datenbankzugriff hat über **mysqli** zu erfolgen.

Die Zugangsdaten sollen in einer **zentralen Datei** gespeichert werden, damit sie an „zentraler“ Stelle leicht zu ändern sind.

Explizite Vorkehrungen **gegen SQL Injections** werden mit Extrapunkten belohnt.

## Beschreibung der einzelnen Komponenten

Baustein 1	HTML Basics	<p><b>1) <u>Hilfe / Impressum</u></b></p> <ul style="list-style-type: none"><li>a. Erstellen Sie eine Hilfe-Seite, auf der erklärt wird, wie die Website zu verwenden ist (kurze Benutzeranleitung bzw. FAQs), die die wichtigsten Fragen zur Benutzung beantworten.</li><li>b. Fügen Sie ein Impressum ein (→ recherchieren Sie, was aktuell ein Impressum enthalten muss).</li><li>c. Das Impressum enthält zusätzlich Bilder und Namen der Hotelverwaltung (=am Projekt beteiligte Studierende).</li><li>d. Sowohl Hilfe als auch Impressum müssen jederzeit einfach erreichbar sein.</li></ul>
Baustein 2	HTML Forms	<p><b>2) <u>Registrierung (wird erweitert in Baustein 7 und 10)</u></b></p> <ul style="list-style-type: none"><li>a. Erstellen Sie ein Registrierungsformular - bei der Registrierung sind (zumindest) folgende Daten anzugeben:<ul style="list-style-type: none"><li>i. Anrede</li><li>ii. Vorname</li><li>iii. Nachname</li><li>iv. E-Mail-Adresse</li><li>v. Username</li><li>vi. Passwort (muss 2x eingegeben werden)</li></ul></li><li>b. Überprüfen Sie alle eingegebenen Daten clientseitig (mit speziellen HTML5 Tags voreinschränken).</li></ul> <p><b>3) <u>User-Login (wird erweitert in Baustein 7, 8 und 10)</u></b></p> <ul style="list-style-type: none"><li>a. Erstellen Sie ein Login-Formular mit Eingabefeldern für Usernamen und Passwort.</li></ul>
Bausteine 3-5	CSS/Responsive Design	<p><b>4) <u>CSS</u></b></p> <ul style="list-style-type: none"><li>a. Verwenden Sie Bootstrap für das Basis-Layout Ihres Projekts sowie entsprechende Komponenten.</li><li>b. Ergänzen Sie Ihr Projekt um eigene Style-Angaben und Bilder, sodass ein stimmiges Design passend zum Thema entsteht.</li><li>c. Lassen Sie sich dabei von Beispiel-Bootstrap-Templates inspirieren.</li><li>d. Erstellen Sie bereits Templates für alle folgenden Ansichten Ihres Projekts.</li><li>e. Testen Sie Ihr Design für verschiedene Endgeräte/Auflösungen → es soll responsive sein und die Website entsprechend konsumierbar.</li></ul>



Baustein 6	PHP Basics	<p><b>5) <u>Modularer Aufbau</u></b></p> <ul style="list-style-type: none"><li>a. Teilen Sie Ihre Projektstruktur derart auf, sodass einzelne Seitenabschnitte/Komponenten in eigenen Files untergebracht sind. Fügen Sie diese wieder via „includes“ zusammen.</li><li>b. Behalten Sie diesen modularen Ansatz für die Implementierung der weiteren Projektteile bei.</li></ul>
Baustein 7	Formularauswertung mit PHP	<p><b>6) <u>Registrierung (Fortsetzung, wird erweitert in Baustein 10)</u></b></p> <ul style="list-style-type: none"><li>a. Überprüfen Sie alle eingegebenen Daten auch serverseitig auf Vollständigkeit und Richtigkeit. Der Username muss eindeutig sein (auch in der Datenbank, siehe Block 11).</li><li>b. Das Passwort muss zweimal eingegeben werden, um Tippfehler zu vermeiden.</li></ul> <p><b>7) <u>User-Login (wird erweitert in Baustein 8 und 10)</u></b></p> <ul style="list-style-type: none"><li>a. Begrüßen Sie den/die „eingeloggt“ User*in auf der Website. Prüfen Sie die eingegebenen Daten vorerst gegen statische Werte (hardcoded).</li></ul>
Baustein 8	Cookies und Sessions	<p><b>8) <u>Profilverwaltung (User-Login Fortsetzung)</u></b></p> <ul style="list-style-type: none"><li>a. Merken Sie sich die Login-Informationen in der Session. Eingeloggte User*innen können die eigenen Profildaten einsehen und bearbeiten (Stammdaten bearbeiten, Passwort neu setzen, ...). Verwenden Sie vorerst statische Daten zur Prüfung des Logins, da die DB-Anbindung erst ab Baustein 10 folgt.</li><li>b. Achten Sie darauf, dass sensible Informationen wie das Passwort nicht vollständig angezeigt werden und beim Ändern des Passwortes auch das alte Passwort eingegeben und überprüft werden muss.</li></ul>

### 9) Zimmer reservieren (wird erweitert ab Baustein 10)

- a. Eingeloggte User\*innen haben folgende Möglichkeiten (vorerst mit statischen Daten):
  - i. neue Zimmerreservierung anlegen
  - ii. eine Liste aller eigener Reservierungen einsehen (vorerst statische Werte)
  - iii. Reservierungen haben einen Status „neu“, „bestätigt“, oder „storniert“
- b. Für neue Reservierungen müssen mindestens folgende Features implementiert sein:
  - i. Reservierungszeitraum (An- und Abreisedatum, Abreisedatum darf nicht  $\leq$  Anreisedatum sein)
  - ii. mit / ohne Frühstück
  - iii. mit / ohne Parkplatz
  - iv. Mitnahme von Haustieren (individuelle Ausgestaltung möglich)

### 10) News-Beiträge von Administrator\*innen (Erweiterung ab Baustein 10)

Die Beitragsverwaltung besteht aus folgenden Funktionen:

- a. **Beitragsanzeige:** Beiträge werden öffentlich zugänglich in einem eigenen Bereich der Webseite (nicht auf der Startseite) übersichtlich angezeigt. Zusammengehörige Elemente wie Bild und Text sind gut sichtbar und weisen eine klare Abgrenzung zu anderen News-Beiträgen auf. Die neuesten Beiträge sollen immer ganz oben angezeigt werden. Verwenden Sie vorerst statische Daten für die Darstellung.
- b. **File-Upload:** Hochladen von Bildern, die **serverseitig verkleinert**, in den Beiträgen angezeigt werden. Dabei ist zu beachten, dass nur Bilddateien für den Upload erlaubt sein sollen, welche in einem eigenen Verzeichnis (z.B. /news) gespeichert werden sollen. **Bilder werden als Thumbnails konstanter Größe dargestellt. Achten Sie darauf, dass die Bilder, tatsächlich nur in Thumbnail-Größe vom Server zum Browser übertragen werden. Es empfiehlt sich die Thumbnails als fertige Bilder am Server in einem eigenen Verzeichnis vorzubereiten (automatische Verkleinerung mittels PHP gleich nach dem Upload).**
- c. Die Darstellung soll übersichtlich sein. Ein News-Beitrag ist klar als solcher erkennbar und kann, sofern hochgeladen, auch Bilder enthalten.

**11) DB-Modell**

- a. Setzen Sie das erarbeitete DB-Modell um und binden Sie Ihr Projekt an

**12) Registrierung - Ergänzungen**

- a. Das Passwort muss zweimal eingegeben werden, um Tippfehler zu vermeiden. Erst wenn alle Daten validiert wurden, wird der/die neue User\*in in der Datenbank (DB) angelegt. Das Passwort wird gehasht in der DB abgelegt.
- b. Legen Sie in der DB manuell ein eigenes Userkonto an, das als Administrator\*in gekennzeichnet wird. Die Rechte des/der Administrator\*in unterscheiden sich von jenen der anderen Usertypen.

**13) User-Login - Ergänzungen**

- a. Überprüfen Sie die Werte gegen die Datenbank. Nur wenn es genau eine\*n User\*in in der DB mit den Daten gibt, gilt der Login als gültig und der Username wird angezeigt. Der Login-Status ist permanent auf der Seite sichtbar durch bspw. den Namen d. User\*in o.ä.. Eingeloggte User\*innen sehen weiters einen Logout-Button.
- b. Auf Basis des/r eingeloggten Users/Userin (registriert oder Administrator\*in) werden Features der Webseite freigeschaltet oder deaktiviert.

**14) News-Beiträge von Administrator\*innen – Ergänzungen**

- a. **Beitragsanzeige:** Beiträge werden aus der DB geladen und angezeigt
- b. **File-Upload:** Speichern Sie den Bildpfad zum jeweiligen DB-Eintrag
- c. Zu jedem Beitrag soll das Datum der Veröffentlichung angezeigt werden.

### **15) Userverwaltung für AdministratorInnen**

Nur AdministratorInnen haben die Möglichkeit andere User zu verwalten. Dazu gehört:

- a. Anzeige einer Liste aller User\*innen
- b. Profilbearbeitung aller User\*innen (falls Stammdaten fehlerhaft erfasst wurden)
- c. Ändern der Passwörter von registrierten User\*innen (z.B. falls sie es vergessen haben)
- d. Verändern des Userstatus: aktiv, inaktiv
- e. Inaktive User\*innen können sich nicht mehr einloggen, bleiben aber in der Datenbank erhalten.
- f. Anzeigen aller Reservierungen und deren Details

### **16) Zimmer reservieren (wird erweitert ab Baustein 11)**

- a. Bilden Sie die Reservierungen in der DB ab und binden Sie die Oberflächen entsprechend an.
  - b. Frühstück, Parkplatz und Haustiere sind mit Aufpreisen zu versehen. Die Höhe der jeweiligen Aufpreise kann vom Projektteam frei (jedoch sinnvoll) selbst gewählt werden.
  - c. Für die Reservierung der einzelnen Zimmer ist auch dessen zeitliche Verfügbarkeit zu berücksichtigen!
  - d. Beim Anlegen von Reservierungen soll das aktuelle Datum sowie die Uhrzeit gespeichert werden, damit Administrator\*innen sehen können, wann welche Reservierungen getätigt wurden.

### **17) Reservierungsverwaltung für Administrator\*innen**

- a. Administrator\*innen haben die Möglichkeit, alle angelegten Reservierungen in einer übersichtlichen Liste anzuzeigen
- b. Sie können einzelne Reservierungen in der Liste auswählen, um deren Details anzuzeigen.
- c. Sie können den Status einzelner Reservierungen ändern (z.B. von „neu“ auf „bestätigt“).
- d. Zudem können sie die Reservierungen nach allen Statuswerten („neu“, „bestätigt“, oder „storniert“) filtern.

### **18) Accessibility und Usability**

- a. Achten Sie bei der Umsetzung auf die Mindestanforderung hinsichtlich Web-Accessibility.
- b. Testen Sie diese Features ggf. mit entsprechenden Browser-Erweiterungen oder Online-Tools.
- c. Achten Sie auf ein gutes Gesamterlebnis beim Konsumieren Ihrer Website und eine schlüssige Navigation.

### **Optionaler Working Plan (Vorschlag)**

*Um Ihnen die Orientierung beim Umsetzen der Anforderungen zu erleichtern, wird folgendes Vorgehen vorgeschlagen (angelehnt an die Themen/Woche innerhalb der LV):*

*Starten Sie zuerst mit den Oberflächen, die Sie benötigen. Beginnen Sie mit einem Bootstrap-Basis Template. Überlegen Sie, welche Seiten Sie benötigen (einzeln, Unterseiten ...). PHP-Funktionen ergänzen Sie dann nacheinander schrittweise, bis zur Datenbank-Anbindung.*

1. *Beginnen Sie mit einem Bootstrap Basislayout, welches responsive ist*
2. *Überlegen Sie, welche Elemente in welcher Form dargestellt werden sollen*
  - a. *Setzen Sie schon vorgefertigte Bootstrap-Komponenten ein*
  - b. *Fügen Sie statische Inhalte hinzu wie bspw. Hilfe/Impressum ...*
  - c. *Verwenden Sie vorerst statische HTML-Platzhalter für dynamische Inhalte*
3. *Ergänzen Sie das individuelle Design*
  - a. *Eigene CSS-Datei*
  - b. *Bilder (img-Verzeichnis in der Projektstruktur)*
  - c. *...*
4. *Vergessen Sie nicht, den Code/die Oberfläche zu testen!*
5. *Erstellen Sie alle Eingabemasken/Uploadformulare*
6. *Beginnen Sie schon jetzt mit dem Kommentieren Ihres Codes und halten Sie offene/zukünftige Punkte fest*
7. *Legen Sie auch leere HTML/PHP-Dateien an, für noch zu implementierende Komponenten*
8. *Entwerfen Sie ein erstes DB-Modell (am Papier oder in einem Dokument), das alle benötigten Daten aufnehmen und abbilden kann. Einzelne*

*Tabellen können bereits ein Anhaltspunkt für den Entwurf zukünftiger PHP-Funktionen sein (Registrierungen von Usern, Erstellen von News-Beiträgen, Reservierungen, ...)*

- a. Sollte das DB-Modell noch unvollständig sein, schärfen Sie dieses sukzessive nach*

*9. Fügen Sie Funktionalität schrittweise hinzu:*

- a. Ergänzen Sie die Auswertung der Eingabemasken (vorerst noch ohne DB-Anbindung), Prüfung auf Vollständigkeit/Validierung, Fehlermeldungen*
- b. Überlegen Sie, wo Sessions und Cookies notwendig sind, und implementieren Sie diese*
- c. Fügen Sie die Upload-Funktionalität an den entsprechenden Stellen hinzu*
- d. Vergessen Sie nicht, den Code zu Kommentieren und zu testen!*
- e. Entwerfen Sie jene Funktionalitäten, die Sie zum Abbilden von Daten (z.B. User, News-Beiträge, Reservierungen, ...) benötigen*
- f. Überlegen Sie, welche Daten Sie von der Oberfläche in die DB-speichern müssen und fügen Sie eine Funktion nach der anderen in der DB-Klasse hinzu*
  - i. Speichern von Usern*
  - ii. Speichern von News-Beiträgen*
  - iii. Speichern von Reservierungen und deren Status*
  - iv. ...*
- g. Wo können sich Änderungen in der DB ergeben? Fügen Sie weiter Funktionen in der DB-Klasse hinzu:*
  - i. Aktualisierungen von Userdaten*
  - ii. ...*
- h. Welche Daten benötigen Sie aus der DB? Ergänzen Sie weitere Funktionen:*
  - i. Userdaten abrufen und an der vorbereiteten Oberfläche anzeigen*
  - ii. ....*
- i. Vergessen Sie nicht, den Code zu Kommentieren und zu testen!*
- j. Ergänzen Sie etwaige Zusatzfeatures für Extrapunkte*