Projektspezifikation „Hotelverwaltung“

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. Gäste- & Personalverwaltung durch Administrator*innen
    ii. Melden von Schäden/Problemen durch Gäste (inkl. Bildupload) via „Tickets“
    iii. Antworten & Statusänderungen der gemeldeten Tickets durch Personal (Servicetechniker*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, 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:

```
index.php
[config]
    dbaccess.php
    …
[inc]
    navigation.php
    content.php
    …
[uploads]
    [news]
        pic1.jpg
        …
    [<userid>]
        pic1.jpg
        …
[res]
    [css]
        mystyle.css
    [img]
        icon.jpg
        …
```

c. Legen Sie eine neue Datenbank für die Hotelverwaltung 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. **Hotelverwaltung_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 der Hotelverwaltung nötig sind, legen Sie ein **README.txt**-File bei (z.B. Zugangsdaten zur Datenbank).

Grundidee

Administrator*innen der Website haben die Möglichkeit **News-Beiträge** zu posten, welche auf der **Startseite** zu sehen sind. Weiters legen sie neue Accounts für Gäste an, wenn diese im Hotel einchecken.

Gäste können die **News-Beiträge** der **Administrator*innen** (auch ohne eingeloggt zu sein) auf der Starseite der Webseite sehen. **Nach dem Einloggen** steht den Gästen **eine weitere Funktionalität**, das Anlegen von **Service-Tickets**, zur Verfügung, um z.B. Schäden/technische Gebrechen, oder andere Probleme zu melden.

Neben den Administrator*innen und den Gästen gibt es eine dritte User-Gruppe: **Die Servicetechniker*innen**. Diese haben die Möglichkeit, Tickets der Gäste einzusehen und darauf zu reagieren (z.B. den Status eines Tickets auf „erfolgreich geschlossen“ zu ändern).

Schwerpunkte sollen sein:

- **User**: Userverwaltung mit 4 Zuständen:
    - anonym (nicht eingeloggt)
    - Gast (registrierter User)
    - Administrator*in
    - Servicetechniker*in
- **Datenbank**: Grundlegende Verwaltung aller Daten in einer Datenbank (Usertypen, News-Beiträge, Tickets, …) → Aufbau einer geeigneten Datenbankstruktur, Zugriff mittels PHP.
- **Posten von News-Beiträgen**: Administrator*innen können News-Beiträge inkl. Bildupload auf der Startseite posten.
- **Anlegen von User-Accounts für Gäste**: Administrator*innen können User-Accounts für Gäste und Servicetechniker*innen anlegen.
- **Anlegen von Tickets inkl. Bildupload**: Gäste können Tickets anlegen, welche von Servicetechniker*innen gelesen und bearbeitet werden können.
- **Bearbeiten von Tickets**: Servicetechniker*innen können Status von Tickets ändern (z.B. auf „erfolgreich geschlossen).
- weitere Bereiche der Webseite:
    - **Hilfe** (Benutzeranleitung)

- **Impressum** (Standard-Impressum), **Namen und Bilder der Hotelverwaltung.**

Die gesamte Website muss **responsive** sein und somit für Smartphones, Tablets und Desktop-Rechner verwendbar sein.

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

Userverwaltung

Es gibt 4 verschiedene Usertypen:

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

Alle Usertypen sehen:

- Hilfe und Impressum
- News-Beiträge der Administrator*innen

Was kann der **anonyme User**:

- News-Beiträge auf der Startseite ansehen
- Hilfe-Seite aufrufen und lesen
- Impressum aufrufen und lesen

Was kann der **registrierte und eingeloggte Gast** zusätzlich zum anonymen User:

- Neue Tickets anlegen (inkl. Bildupload)
- Angelegte Tickets einsehen

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

- Neue Gäste (z.B. beim Einchecken) anlegen
- Gäste (z.B. beim Auschecken) deaktivieren
- News-Beiträge (inkl. Bildupload) erstellen und löschen
- Userverwaltung (Übersicht bestehender und deaktivieren einzelner User)
- Servicetechniker*innen mit entsprechenden Rechten anlegen und deaktivieren

Was kann der/die **Servicetechniker*in** zusätzlich zum anonymen User:

- Tickets der Gäste einsehen
- Status der Tickets ändern

Aufbau der Datenbank

Datenbankmodell:

Überlegen Sie sich ein **geeignetes Datenbankmodell**

Hinweise: Sie müssen User und ihre Rolle (Gast, Admin, Servicetechniker*in) speichern, News-Beiträge verwalten (etwaige Bilder selbst liegen im Filesystem des Webservers, die Datenbank verwaltet nur Referenzen auf die Bilder), wer Ersteller*n eines News-Beitrags.

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

1) Hilfe / Impressum

a. Erstellen Sie eine Hilfe-Seite, auf der erklärt wird, wie die Website zu verwenden ist (kurze Benutzeranleitung).
b. Fügen Sie ein Impressum ein (→ recherchieren Sie, was aktuell ein Impressum enthalten muss).
c. Das Impressum enthält zusätzlich Bilder und Namen der Hotelverwaltung (=am Projekt beteiligte Studierende).
d. Sowohl Hilfe als auch Impressum müssen jederzeit einfach erreichbar sein.

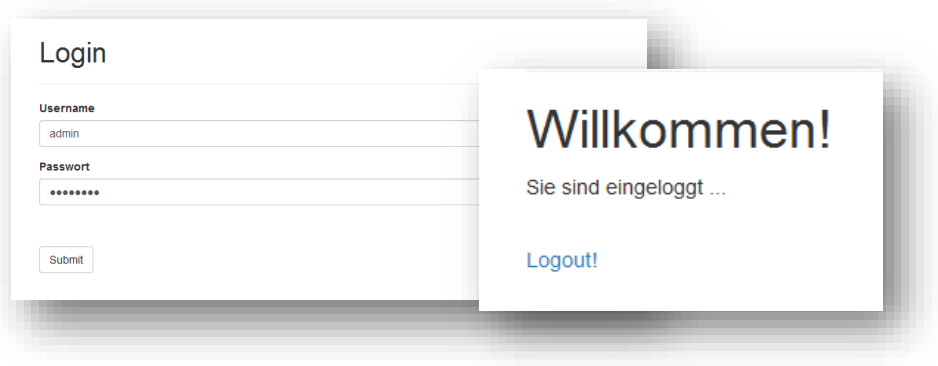
2) Gast-Registrierung durch Administrator*in

a. Beim Anlegen eines neuen User-Accounts für einen Hotelgast sind (zumindest) folgende Daten anzugeben:
    i. Anrede
    ii. Vorname
    iii. Nachname
    iv. E-Mail-Adresse
    v. Benutzername
    vi. Passwort
b. Überprüfen Sie alle eingegebenen Daten sowohl clientseitig (mit speziellen HTML5 Tags voreinschränken) als auch serverseitig auf Vollständigkeit und Richtigkeit. Der Username muss eindeutig sein (auch in der Datenbank). Erst wenn alle Daten validiert wurden, wird der neue User in der Datenbank (DB) angelegt. Das Passwort wird verschlüsselt in der DB abgelegt.
c. Legen Sie manuell in der DB einen eigenen User an, der als Administrator*in gekennzeichnet wird. Die Rechte des/der Administrator*in unterscheiden sich von den anderen Usertypen.

3) User-Login

a. Erstellen Sie ein Login-Formular mit Eingabefeldern für Usernamen und Passwort.
b. Überprüfen Sie die Werte gegen die Datenbank. Nur wenn es genau einen User 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. Eingeloggte User sehen einen Logout-Button.

c. Auf Basis des eingeloggten Users (Administrator*in, Hotelgast, Servicetechniker*in) werden Features der Webseite freigeschaltet oder deaktiviert. (Als Beispiel folgender Screenshot)



4) Profilverwaltung

a. Eingeloggte User können die eigenen Profildaten einsehen und bearbeiten (Stammdaten bearbeiten, Passwort neu setzen, …)
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.

5) News-Beiträge von Administrator*innen

Die Beitragsverwaltung besteht aus folgenden Funktionen:

a. **Beitragsanzeige**: Anzeige der Beiträge auf der Startseite übersichtlich und voneinander getrennt. 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.
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 (uploads/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.
d. Zu jedem Beitrag soll das Datum der Veröffentlichung angezeigt werden.

6) Userverwaltung für AdministratorInnen

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

a. Anzeige einer Liste aller User
b. Profilbearbeitung aller User (falls Stammdaten fehlerhaft erfasst wurden)
c. Ändern der Passwörter von Hotelgästen (z.B. falls sie es vergessen haben)
d. Verändern des Userstatus: aktiv, inaktiv (z.B. nach dem Auschecken)
e. Inaktive User können sich nicht mehr einloggen, bleiben aber in der Datenbank erhalten.
f. Servicetechniker*innen mit entsprechenden Rechten anlegen und deaktivieren.
g. Anzeigen einer Liste aller Tickets (Titel, Gast und Status)

7) Tickets

a. Eingeloggte Gäste haben folgende Möglichkeiten:
    i. neue Tickets anlegen
    ii. eine Liste aller ihrer Tickets inkl. deren Status einsehen
    iii. der Status kann folgende drei Werte haben: „offen“, „erfolgreich geschlossen“, oder „erfolglos geschlossen“
    iv. geschlossene Tickets erneut öffnen (z.B. wenn das Problem nicht/nicht vollständig behoben wurde)
b. Das Formular für neue Tickets muss mindestens aus folgenden Komponenten bestehen:
    i. Titel
    ii. Beschreibung
    iii. Bildupload

iv. Absenden-Button

c. Beim Anlegen von Tickets soll das aktuelle Datum sowie die Uhrzeit gespeichert werden.

8) Ticketverwaltung für Servicetechniker*innen

a. Servicetechniker*innen haben die Möglichkeit, alle angelegten Tickets in einer übersichtlichen Liste (Titel, Status, Datum) anzuzeigen und können diese nach allen Statuswerten filtern.
b. Sie können einzelne Tickets in der Liste auswählen, um Beschreibung und hochgeladene Bilder anzuzeigen.
c. Sie können auf Tickets antworten (z.B. dass die Schäden zu einer bestimmten Uhrzeit behoben werden.)
d. Sie können den Status einzelner Tickets ändern (z.B. von „offen“ auf „erfolgreich geschlossen“).

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. *<u>Vergessen Sie nicht, den Code/die Oberfläche zu testen!</u>*
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 Files an, für noch zu implementierende Komponenten*
8. *Entwerfen Sie ein erstes DB-Modell (am Papier), das alle benötigten Daten aufnehmen und abbilden kann. Einzelne Tabellen können bereits ein Anhaltspunkt für den Entwurf zukünftiger PHP-Funktionen sein (Anlegen von Gästen, Erstellen von News-Beiträgen, Tickets, …)*
   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. *<u>Vergessen Sie nicht, den Code zu Kommentieren und zu testen!</u>*
   e. *Entwerfen Sie jene Funktionalitäten, die Sie zum Abbilden von Daten (z.B. User, News-Beiträge, Tickets, …) 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 Tickets 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. *<u>Vergessen Sie nicht, den Code zu Kommentieren und zu testen!</u>*
j. *Ergänzen Sie etwaige Zusatzfeatures für Extrapunkte*