



**Westfälische
Hochschule**

Gelsenkirchen Bocholt Recklinghausen

Prof. Dr.-Ing. Martin Schulten

Modulprüfung Internetanwendungen

Bearbeitungszeit:	30.06.23 09:00 Uhr – 16.07.23 23:59 Uhr
Erlaubte Hilfsmittel:	alle außer Hilfe Dritter
Erreichbare Punkte:	100
Mindestpunktzahl für Bestehen:	50

Hinweise:

- Die erreichbare Punktzahl ist bei den Aufgaben in eckigen Klammern angegeben.
- Lesen Sie die Aufgabenstellungen genau durch und achten Sie auf alle Vorgaben.
- Bewertet wird der Erfüllungsgrad der Aufgaben, die schriftlichen Lösungen sowie die Codequalität. Es wird Wert auf die Anwendungsarchitektur sowie kurzen, eleganten und sauber strukturierten Code gelegt. Nutzen Sie die Möglichkeiten, die Java bietet. Unsaubere Strukturierung und suboptimale „Bastellösungen“ führen zu Punktabzügen.
- Gruppenarbeit ist unzulässig.
- Die Abgabe erfolgt in einer ZIP-Datei.
 - Diese muss Unterverzeichnisse für jede Aufgabe enthalten. Details zur Abgabe stehen unter jeder Aufgabe.
 - Außerdem muss die ZIP-Datei die Eigenständigkeitserklärung (siehe nächste Seite) enthalten.

Eigenständigkeitserklärung

Diese Erklärung ist mit Namen, Datum und eigenständiger Unterschrift versehen zusammen mit den Aufgabenlösungen digital abzugeben.

Hiermit versichere ich, dass ich die Prüfung „Internetanwendungen“ selbstständig, ohne Hilfe Dritter und ohne unerlaubte Hilfsmittel abgelegt habe.

Name: _____

Datum: _____

Unterschrift: _____

Abzugeben ist:

- Dieses Blatt vollständig ausgefüllt als PDF-Dokument oder Grafik zusammen mit den Aufgabenlösungen

Konzeption und Implementierung eines Webshops (Java-Webanwendung)

Es soll ein Java-Webshop konzipiert und programmiert werden.

Dies sind die grundlegenden funktionalen Anforderungen an den Shop:

- Öffentliche Artikelliste
 - o Artikel sind mit Bezeichnung und jeweiligem Lagerbestand aufgelistet.
 - o 5 Artikel sind fest implementiert mit anfänglichem Lagerbestand.
 - o Jeder Anwender (auch ein nicht eingeloggter Anwender) kann Artikel einzeln in Warenkorb legen, sofern jeweiliger Lagerbestand > 0 .
 - o Wenn Artikel in den Warenkorb gelegt werden, wird der Lagerbestand reduziert.
 - o Kauf mehrerer auch gleicher Artikel ist möglich.
- Warenkorb
 - o Jeder Anwender kann seinen Warenkorb einsehen.
 - o Jeder Anwender kann Artikelposten aus seinem Warenkorb entfernen. Eine Bearbeitungsmöglichkeit der Artikelanzahl ist im Warenkorb nicht vorgesehen. Der Lagerbestand wird entsprechend aktualisiert.
 - o Ein eingeloggter Anwender kann Artikel bestellen. Damit wird der Warenkorb geleert. Die Bestellfunktion fordert zum Login auf, falls der Anwender noch nicht eingeloggt war.
 - o Bei Logout mit gefülltem Warenkorb wird der Warenkorb gelöscht und die Artikel werden zurück ins Lager gebucht.
- Login und Logout mit Benutzername und Passwort
 - o Es gibt zwei fest implementierte Anwenderkonten zum Testen (user1/pw1, user2/pw2).
 - o Es gibt eine Login- und eine Logout-Funktion.

Hinweise:

- Als Applikationsserver benutzen Sie bitte Apache Tomcat 10.x.
- Es soll bei der Entwicklung keine Datenbank eingesetzt werden. Die Daten sind jeweils nur zur Laufzeit gültig.
- Achten Sie darauf, welchen Kontext die Daten jeweils haben müssen.
- Die Unterpunkte, in denen programmiert werden muss [c) bis e)] bauen aufeinander auf. Sie können die Lösungen getrennt abgeben oder alternativ das Endergebnis dieser drei Unteraufgaben in einer Abgabe für c) bis e) einreichen.
- Die Unteraufgaben folgen auf der nächsten Seite.

a) [5 Punkte] Welches **Entwurfsmuster** nutzen Sie für die Implementierung der Anwendung? Bitte geben Sie den Namen des Musters an und skizzieren Sie das Muster (alle Blöcke und Pfeile sowie die wichtigsten Beschriftungen).

Abgabe: Dokumentation

b) [10 Punkte] Überlegen Sie sich, wie Sie die Anwendung strukturieren möchten (**Module**). Benennen Sie jedes Modul und wählen Sie eine Technik, in der Sie es implementieren wollen. Ordnen Sie es einem Teil des Entwurfsmusters (siehe a)) zu.

Beispiel:

- Loginseite, JSP, [spezifischer Teil des Entwurfsmusters]
- Benutzerdaten, ...

Abgabe: Dokumentation

c) [60 Punkte] **Programmieren** Sie die Anwendung entsprechend der obigen Planung.

Abgabe: WAR-Datei, Quelltexte

d) [10 Punkte] **Erweiterung** der Implementierung: Zusätzlich zu den beiden festen Benutzerkonten soll es eine Registrierungsmöglichkeit für neue Nutzer geben.

Abgabe: WAR-Datei, Quelltexte

e) [10 Punkte] **Erweiterung** der Implementierung: Zusätzlich zur initial festen Artikelliste soll es exklusiv für den Benutzer user1 möglich sein, die Liste zu pflegen (Artikel hinzufügen, Artikel löschen, Lagerbestand ändern).

Abgabe: WAR-Datei, Quelltexte

f) [5 Punkte] Nur Planung, keine Implementierung: Nun sollen die Daten der Artikelliste fiktiv in einer Datenbank gespeichert werden. Die Implementierung soll möglichst datenbankunabhängig mit einem **ORM-Framework** erfolgen.

Beschreiben Sie im Detail, was Sie tun müssten, um ein Persistenz-Framework zu integrieren.

Abgabe: Dokumentation