

## **Beurteilung Projektarbeit**

Das Thema und deren Inhalt werden durch das Team bestimmt. Die Applikation muss eine klare 3- Tier Architektur aufweisen.

- Presentation Layer muss gut strukturiert und mit CSS Definitionen bestimmt werden. Überprüfung von User Eingaben muss implementiert werden.
- Business Layer zeigt sinnvolle Strukturen auf und wird nach den Standards der Programmierung (Modular, Strukturiert, etc.) implementiert und dokumentiert.
- Data Layer muss eine Datenbank aufweisen, die nach den Relationalen Richtlinien definiert und implementiert wird.
- Sicherheits-relevante Aspekte werden berücksichtigt.

## Beurteilungskriterien (insgesamt 25 Punkte möglich):

Kriterien	Beschreibung der geforderten Inhalte	ten Inhalte Punkte (1 – 5)		
Analyse / Design	Funktionale Anforderung transparent Die Applikation ist nach einer 3-Tier Archi- tektur aufgebaut; GUI - Design			
Implementation / Funktion	Die Applikation beinhaltet folgende Punkte: Datenbestand, Formularüberprü- fung, Usability, Login für Benutzer-Ac- count, Registrierung.			
Tests / Sicherheits- aspekte	Sicherheitskritische Aspekte sind mit ent- sprechenden Testfällen belegt Korrektes Session-Handling. SQL-Injection prevention.			
Dokumentation / Arbeitsplanung	Die Analyse, Planung und Realisierung der Web-Appl			
Ergebnisse	Arbeitsteilung im Team, Schwierigkeits- grad der Aufgabe, Besonderes			

<u></u>	1	$\nu$	//
	ı	DΣ	//
	L		/ -

Kriterien	Fragen	Check	
Analyse/ Design	<ul> <li>Ausgangsbeschreibung Problemstellung ist in der Analyse weiter verfeinert, z.B. UseCase Diagramme</li> <li>Das Aussehen der Bildschirmmasken wurde schon vor der Implementation skizzenhaft festgelegt. Z.B. mit Mockups</li> <li>Aufgaben bis zur Eingabe Überlegungen transparent</li> <li>Benutzerprofil (Überlegungen)</li> <li>Das Design zeigt, wie die Aufgabenstellung umgesetzt wird. Direktory-Struktur, Unterteilung in Schichten</li> <li>Dialogspezifikationen, modal, nicht modal zur sicheren Benutzerführung</li> </ul>		
Implemtaion/Funktionen	<ul> <li>Registrierung Login Logout Funktionen sind vorhangen</li> <li>Look &amp; Feel mit geeigneten Mittel gestaltet (CSS, responisvDesign)</li> <li>Code ist strukturiert, Aufteilung wer hat was gemacht</li> <li>DesignPattern, z.B. MVC wird korrekt angewendet</li> <li>Korrekte Formularauswertung client- severseitig</li> <li>Implementation ist vollständig und entspricht dem Design</li> <li>Wichtige Stellen des Codes sind beschrieben</li> <li>Es ist klar wer welchen Code geschrieben hat</li> <li>History: Autor; Änderungen, Namensgebung Klassen, File</li> <li>Kopierte Teile sind mit Quellenangaben vorhanden</li> <li>Code wurde eigenständig erstellt</li> </ul>		
Test/Sicherheitsas- pekte	<ul> <li>Testfälle sind sehr detailliert und ausführlich beschrieben</li> <li>Testszenarios UseCases, sowohl positiv und negativ</li> <li>Benutzerfeedback vorhanden</li> <li>Welche neuen Erkenntnisse wurden durch den Test gewonnen</li> </ul>		
Dokumentation/Arbeitsplanung	<ul> <li>Dokumentation umfasst: Pflichtenheft, Analyse, Tests, Fazit</li> <li>Vorlagen mit einheitlichem ansprechendem Layout</li> <li>Planungsdokument (Meilensteine, Aufgabenteilung, usw.)</li> </ul>		
Ergebnisse	<ul> <li>Aufgabenstellung ist vollständig umgesetzt worden</li> <li>Welche Punkte zeigen die Zusammenarbeit, Kooperation</li> <li>Leistung kann kritisch hinterfragt werden</li> <li>Selbsteinschätzung ist realistisch und transparent</li> <li>Eigener Lernprozess kann transparent gemacht werden</li> <li>Aufgabenstellung niveaugerecht eingeschätzt</li> </ul>		