

Web Engineering

Projekt-Kickoff

Adrian Herzog

Übersicht

Ziel: *Entwickeln einer «realistischen» Web-App,
basierend auf eigener Idee.*

Inspiration: <https://flaviocopes.com/sample-app-ideas/>

teilweise deutlich
zu einfach

Modus: Einzelarbeit

Umfang: ca. 30 Stunden

Abgabe-Format: Spring-Boot-Applikation über GitHub Classroom

Zeitlicher Ablauf

heute	GitHub Classroom Einladung annehmen (kommt per Teams)
spätestens 3.11.	Applikations-Vorschlag per E-Mail an mich (ca. 5 Sätze). Ich bestätige den Vorschlag oder schlage Anpassungen vor.
02.12.	Arbeit am Projekt (Unterstützung vor Ort)
09.12.	Arbeit am Projekt (Unterstützung per E-Mail)
16.12.	Arbeit am Projekt (Unterstützung per E-Mail oder Teams)
16.12. um 23:59	Spätester Zeitpunkt für Abgabe (via GitHub) <i>Letzter Push zählt. Stellen Sie sicher, dass in dieser Version alles funktioniert (auch die Tests).</i> <i>Last-Minute-Änderungen sind keine gute Idee!</i>

Verwendung von externem Code

Erlaubt:

- Kopieren von kleinen Codestücken von Stack Overflow, Tutorials, usw.
- Verwenden von Drittbibliotheken, *aber nur auf der Server-Seite!*
- Diskutieren mit anderen Teams, Beratung durch Andere, Copilot und Chatbots

Wichtig: Kopierter oder generierter Code muss klar markiert sein

→ **Kommentar in Code (oder Commit-Message), inkl. «EXTERNAL» und Quelle**

(Ausnahme: single line code completion muss nicht deklariert werden)

Nicht erlaubt:

- JS-Bibliotheken/-Frameworks, CSS-Frameworks (Bootstrap o.ä.)
- Kopieren von Code aus den Übungen oder Musterlösungen
(Ausnahme: kleine Dinge, die man «nicht anders machen kann»,
z. B. Einstellungen in application.properties)

Bewertung

Siehe Bewertungsraster in Excel (Bewertung Projekt.xlsx)

55 Punkte möglich

- Funktionalität: 30 Punkte
- Engineering: 20 Punkte
- Individuelle Zusatzleistung: 5 Punkte

Notenskala

- Bis 50 Punkte: 1 + Punkte / 10
- > 50 Punkte: Note 6
- Betrugsversuch: Note 1

Beispiel-Web-App

Falls keine andere Idee: ***Frage-und-Antwort-App***

3 Entitäten: *Fragen*, *Antworten*, *Tags* (Themen)

- Eine Frage kann mehrere Tags enthalten (N:M)
- Zu einer Frage kann es mehrere Antworten geben (1:N)

CRUD-Unterstützung für Fragen, Erstellen & Löschen von Antworten.

Vordefinierte Tags in DB (werden beim ersten Starten reingeladen)

Engineering-Aspekte:

- Sinnvolle Unit-Tests, Integration-Tests, E2E-Tests
- Valides HTML, semantische Elemente
- Sauberer Code & saubere Git-History

Git & Hilfe

Zusätzliche Anforderungen:

- Die gesamte Entwicklung muss über das persönliche Git Repository in GitHub Classroom laufen
- Pushe deinen neusten Entwicklungsstand regelmässig auf GitHub

Commit early, commit often!

Hilfe erhalten:

- 20 Minuten selber versuchen, Problem zu lösen (Google, Stack Overflow, Chatbots)
- Wenn möglich 20 Minuten mit Mitstudierenden versuchen, Problem zu lösen
- E-Mail an mich, mit Problem-Beschreibung (inkl. Compiler-Fehler oder vollständigem Stack Trace) und *Links zu Code auf GitHub!*