ÜbunGszettel Fullstack-Development mit Spring Boot

# Aufgabe 1: Dokumentation

Begleite deine komplette Tätigkeit über ein Github oder GitLab-Repository. Je Aufgabenstellung in diesem Tutorial legst du ein separates GIT-Repository an. Nach jeder deiner Arbeitsphasen führst du ein Commit/Push durch. Es soll ein gutes Bild davon entstehen, wie oft und woran du an diesem Übungszettel gearbeitet hast.

Du dokumentierst deine Arbeit außerdem in einem separaten Ordner „Dokumentation“ und unter Verwendung einer Word-Datei oder - noch besser - einer Markdown-Datei (gerne auch mit Screenshots) auf der verwendeten Git-Plattform (siehe [Mastering Markdown · GitHub Guides](https://guides.github.com/features/mastering-markdown/)). Schreibe regelmäßig einen *kurzen* Bericht, der folgende Punkte enthalten kann:

* Selbsterklärung neuer Spring Boot Features
* Erklärung des Einsatzes neuer Features
* Aufgetretene Probleme und ihre Lösung
* Besondere Lernereignisse, die deinen Lernfortschritt / deinen Kompetenzzuwachs besonders positiv beeinflussten
* Etc.

# Aufgabe 2: SprinG Boot Fullstack User-Management (2,5 Stunden) (10 STUNDEN gebrauch)

Erledige das folgende Tutorial, in dem du lernst, wie man mit Spring Boot eine Fullstack-Applikation zum Management von Benutzern mit einer etablierten Standardarchitektur erstellt.

[Spring Boot CRUD Tutorial with IntelliJ IDEA, MySQL, JPA, Hibernate, Thymeleaf and Bootstrap - YouTube](https://www.youtube.com/watch?v=u8a25mQcMOI)

# Aufgabe 3: Spring Boot Fullstack Department-Management (4 Stunden) (12 stundeN)

Erledige das folgende Tutorial, in dem du lernst, wie man mit Spring Boot eine REST-API entwickelt, globales Exceptions-Handling implementiert UND automatisiert testet. Teile dieses Tutorials sind eine gute Wiederholung des vorhergehenden.

[Spring Boot Tutorial | Full In-depth Course - YouTube](https://www.youtube.com/watch?v=c3gKseNAs9w)

# Aufgabe 4: Spring Data JPA Vertiefung (3 Stunden)

Erledige das folgende Tutorial und lerne im Detail, wie man mit Spring Data JPA effizient und effektiv Daten in einer Spring Boot Applikation in eine Datenbank speichern kann.

[Spring Data JPA Tutorial | Full In-depth Course - YouTube](https://www.youtube.com/watch?v=XszpXoII9Sg)

Konzentriere dich dabei insbesondere auf die ManyToOne-Beziehung.

# Aufgabe 5: Weitere Entität hinzufügen (3 Stunden)

Erweitere eines der ersten beiden durchgemachten Spring-Boot-Tutorials (Aufgabe 2 oder Aufgabe 3) mit mindestens einer weiteren Domänenklasse, die zu einer der bestehenden Domänenklassen eine sinnvolle 1:n bzw. n:1 Beziehung aufbaut. Implementiere alle Layers (Domäne, Service / Business, Repo, REST- oder Thymeleaf-MVC-Controller) entsprechend. Globales Exceptionhandling, JUNIT-Tests sind obligatorisch. Bereite dich auf eine Präsentation deiner Lösung vor.