



Technische Berufsschule Zürich TBZ  
Höhere Fachschule

## Semesterarbeit Einreichungsformular

### Klasse ITCNE23 – 3. Semesterarbeit

Im Folgenden beschreibt der Studierende das geplante Thema seiner Zertifikatsarbeit. Der Studierende orientiert sich dabei an dem Bewertungsraster. Der begleitende Dozent entscheidet, ob es sich dabei um ein geeignetes Thema handelt und fügt seine Kommentare und Überlegungen hinzu.

Bitte die *kursiven Textteile* durch die konkreten Angaben ersetzen.

#### Name und Vorname des Studierenden

Perparim Thaqi

#### Titel der Semesterarbeit

*Phishing Login Page für interne Tests*

#### Zu behandelnde Themenfelder / Module (bitte ankreuzen)

Pkt.	Themenfeld / Modul	
1.	Projektmanagement	Pflicht
2.	Relationale Datenbanken	<input type="checkbox"/>
3.	NoSQL, Key/Value, Time Series	<input type="checkbox"/>
4.	Microservices mit Python	X
5.	Datensicherheit / Datenschutz	<input type="checkbox"/>
6.	Netzwerk	<input type="checkbox"/>
7.	Optional: Thema aus Semester 2	<input type="checkbox"/>

## Kurzbeschreibung des Zertifikatsarbeit Themas (1 bis max. 2 Seiten)

### Problemstellung / Ausgangslage / Potential der Semesterarbeit

#### DOITNOW!

Ein Start-up namens " DOITNOW!", das eine einfache ToDo-Listen-Anwendung entwickeln möchte.

**Problemstellung / Ausgangslage:** «DOITNOW!» hat erkannt, dass viele vorhandene ToDo-Listen-Anwendungen entweder überladen sind. Sie möchten eine minimalistische Anwendung anbieten, die es Benutzern ermöglicht, ihre Aufgaben effizient zu verwalten und abzuschließen.

**Potential der Semesterarbeit:** Die Entwicklung einer Microservices-Architektur ermöglicht es «DOITNOW!», flexibel auf zukünftige Anforderungen zu reagieren und die Anwendung bei Bedarf zu erweitern. Durch die Verwendung von Docker können die verschiedenen Komponenten der Anwendung isoliert und effizient bereitgestellt werden. Die Verwendung von Flask API für das Backend und React für das Frontend ermöglicht eine saubere Trennung von Backend- und Frontend-Logik, was die Wartbarkeit und Skalierbarkeit verbessert. Auch soll eine CI/CD Pipeline über GitLab erstellt werden, welche ein einfaches Entwickeln und Deployen der Applikation ermöglichen soll.

#### SWOT

##### Stärken:

- Einfache Benutzeroberfläche: Die minimalistische Benutzeroberfläche macht die Anwendung benutzerfreundlich und leicht verständlich.
- Flexibilität durch Microservices: Die Verwendung einer Microservices-Architektur ermöglicht eine flexible Skalierung und Erweiterung der Anwendung.
- Effiziente Datenverwaltung: Die Verwendung von SQLAlchemy für die Datenbankverwaltung ermöglicht eine effiziente Verwaltung und Abfrage von Aufgaben.
- Gute Integration von Frontend und Backend: Die klare Trennung von Frontend und Backend ermöglicht eine saubere und wartbare Codebasis.

##### Schwächen:

- Begrenzte Funktionalität: Da es sich um eine minimalistische ToDo-Listen-Anwendung handelt, könnten einige Benutzer zusätzliche Funktionen vermissen, die in komplexeren Anwendungen verfügbar sind.
- Abhängigkeit von Drittanbietern: Die Verwendung von AWS für die Infrastruktur und Docker für die Containerisierung macht die Anwendung teilweise abhängig von externen Diensten, was potenziell zu Ausfallzeiten führen könnte.

##### Chancen:

- Marktwachstum für Produktivitäts-Apps: Mit der steigenden Nachfrage nach Produktivitäts-Apps könnten ToDo-Listen-Anwendungen eine wachsende Benutzerbasis anziehen.

- Möglichkeit zur Integration zusätzlicher Funktionen: Durch kontinuierliche Iteration und Feedback von Benutzern besteht die Möglichkeit, die Anwendung um zusätzliche Funktionen zu erweitern, um den Anforderungen der Benutzer besser gerecht zu werden.

### **Bedrohungen:**

- Wettbewerb von etablierten Anbietern: Es gibt bereits eine Vielzahl von ToDo-Listen-Anwendungen auf dem Markt, die mit DOITNOW! konkurrieren könnten.
- Datenschutz- und Sicherheitsbedenken: Da die Anwendung personenbezogene Daten der Benutzer speichert, besteht die Gefahr von Datenschutzverletzungen oder Sicherheitsproblemen, die das Vertrauen der Benutzer beeinträchtigen könnten. Um dies zu gewährleisten, sollen bestimmte Daten verschlüsselt gespeichert werden.
- Die Anwendung bietet ausserdem ein sicheres Login, um die Daten nur Authentifizierten und Autorisierten Benutzern zur Verfügung zu stellen.

### **Zielsetzung der Semesterarbeit**

Entwicklung einer Microservices-Architektur unter Verwendung von AWS, Docker, Flask API für das Backend, React für das Frontend und SQLAlchemy für die Datenbankverwaltung. Das Frontend soll eine minimalistische Benutzeroberfläche bieten, auf der Benutzer Aufgaben hinzufügen, bearbeiten und abhaken können. Es wird eine Listenansicht mit Aufgaben und eine Möglichkeit zum Hinzufügen neuer Aufgaben geben. Die Anwendung soll intuitiv zu bedienen sein und eine reibungslose Benutzererfahrung bieten.

### **Terminplan mit den wesentlichen Arbeitsschritten**

- 06.05.24-10.05.24: Abgabe und Besprechung Einreichungsformular
- 27.05.24: Einzelbesprechung Zwischenstand
- 05.07.24: Abgabe der Semesterarbeit
- 08.07.24: Schlusspräsentation

### **Sachmittel / Rahmenbedingungen**

- Büroräumlichkeiten, TBZ
- Flask/APIFlask Dokumentation
- Dozenten als Unterstützung

### **Vorgaben, Methoden und Werkzeuge**

- AWS
- APIFlask
- Python
- GitLab
- Office Suite
- Projektmethode: KanBan
- Terminplanung/Projektmanagement: GANTT

## Risiken

- Da ich kein Fullstack Entwickler bin, besteht die chance, dass das Frontend nicht einwandfrei funktioniert.
- Da das Projekt bereits am 05.07.2024 fertiggestellt werden muss, besteht ein gewisser Zeitdruck.

## Entscheid des begleitenden Dozenten

Bitte ankreuzen

- Genehmigt ☐
- Zu verbessern ☐
- Abgelehnt ☐

## Begleitender Dozent

Ort und Datum: \_\_\_\_\_

Name & Unterschrift: \_\_\_\_\_

## Beurteilung des Antrages

Kriterium	Kommentare	Erfüllt
<b>Anforderungen an die Form (Strukturierung) der Semesterarbeit</b>		
Problemstellung		<input type="checkbox"/>
Ziele (mindestens drei!)		<input type="checkbox"/>
Risiken bezogen auf Zertifikatsarbeit		<input type="checkbox"/>
Themenfelder (mindestens zwei!)		<input type="checkbox"/>
<b>Anforderungen an Qualität der Semesterarbeit</b>		
Machbar		<input type="checkbox"/>
Praxisnah		<input type="checkbox"/>
Herausfordernd		<input type="checkbox"/>
Lehrgangsbezug		<input type="checkbox"/>

Damit die Semesterarbeit angenommen wird, müssen alle Kriterien erfüllt sein.

Die Kommunikation wird über den jeweiligen Teams-Kanal geführt.