

Arbeitsauftrag LAP Vorbereitung

Applikationsentwicklung - Coding

Arbeitsauftrag Nr.: LAP-002

Lehrjahr: 4. Lehrjahr

Arbeitsumfang: 16 Stunden

Aufgabenstellung

Thema: Arbeitszeiterfassung

Sie sollen eine **Webanwendung zur Arbeitszeiterfassung** entwickeln, die es einem Benutzer ermöglicht, seine Arbeitszeiten zu erfassen und auszuwerten.

Technische Rahmenbedingungen:

- **Framework:** MVC oder MVVM Framework
- **Programmiersprache:** Wahl zwischen PHP, C#, Java oder Python
- **Datenbank:** MariaDB oder Microsoft SQL Server
- **Architektur:** Webanwendung mit klarer Trennung von Darstellung und Geschäftslogik
- **Abgabe:** GitHub Repository, das mit dem Trainer geteilt wird

Funktionale Anforderungen

1. Grundfunktionen der Arbeitszeiterfassung

1.1 Kategorien von Zeiterfassungen:

Der Benutzer soll zwischen folgenden drei Kategorien unterscheiden können:

- **Arbeitstag**
- **Urlaub**
- **Krankenstand**

1.2 Erfassung von Arbeitstagen:

Bei der Erfassung von Arbeitstagen sind folgende Daten zu erfassen:

- **Datum** des Arbeitstages
- **Arbeitsbeginn** (Startzeit)
- **Arbeitsende** (Endzeit)
- **Mittagspause** (Dauer in Minuten)
- **Automatische Berechnung** der Arbeitszeit
- **Soll-Arbeitszeit:** 8 Stunden pro Tag
- **Automatische Berechnung** von Überstunden/Minusstunden

1.3 Erfassung von Urlaub und Krankenstand:

Für Urlaubs- und Krankenstandstage genügt die Erfassung von:

- **Datum**
- **Kategorie** (Urlaub oder Krankenstand)
- **Optional:** Bemerkung/Notiz

2. Übersicht und Filterung

2.1 Übersicht aller Einträge:

- Tabellarische Darstellung aller erfassten Zeiteinträge
- Anzeige von Datum, Kategorie, Arbeitszeit (bei Arbeitstagen), Überstunden/Minusstunden

2.2 Filterfunktionen:

Die Übersicht soll gefiltert werden können nach:

- **Kategorie:** Arbeitstag, Urlaub, Krankenstand
- **Monat:** Auswahl eines bestimmten Monats
- **Kombinierte Filter:** z.B. alle Krankenstände im März

3. Auswertungen und Statistiken

3.1 Saldo-Anzeige:

- **Gesamtsaldo** der Plus-/Minusstunden
- Klar erkennbare Darstellung, ob Überstunden oder Minusstunden vorliegen

3.2 Statistiken:

Top 5 Monate mit höchsten Abwesenheiten:

- Ranking der 5 Monate mit den meisten Urlaubs- und Krankenstandstagen
- Anzeige der Anzahl der Abwesenheitstage pro Monat

3.3 Visualisierung:

Kreisdiagramm zur Verteilung der Arbeitstage:

- Anteil Arbeitstage/Urlaubstage/Krankenstandstage (in %)

Technische Anforderungen

6. Benutzeroberfläche

6.1 Navigation:

Erstellen Sie eine übersichtliche Navigation mit folgenden Bereichen:

- **Dashboard/Übersicht**
- **Neue Zeiterfassung**
- **Alle Einträge**
- **Statistiken**

6.2 Formulare:

- **Eingabeformular** für neue Zeiterfassungen mit serverseitiger Validierung
- **Suchformular** für Filterung

6.3 Listen und Tabellen:

- Übersichtliche Darstellung der erfassten Daten
- Sortierung nach Datum (neueste zuerst)
- Paginierung bei vielen Einträgen (optional)

Abgabe und Dokumentation

GitHub Repository:

Das Projekt ist in einem **GitHub Repository** zu erstellen und zu verwalten:

README.md muss enthalten:

- **Projektbeschreibung**
- **Verwendete Technologien**
- **Datenbankschema**

Repository teilen:

Das Repository ist mit dem Ausbilder zu teilen:

- Repository auf **public** setzen ODER
- Ausbilder als **Collaborator** hinzufügen

Zusätzliche Hinweise

Testdaten:

- Erstellen Sie **Beispieldaten** für verschiedene Monate
- Mindestens **15 Einträge** mit unterschiedlichen Kategorien
- **Realistische Arbeitszeiten** (7-9 Stunden pro Arbeitstag)

Validierung:

- **Serverseitige Validierung:** Überprüfung aller Eingaben
- **Datumsprüfungen:** Zukünftige Daten sollten nicht erlaubt sein

Erlaubte Hilfsmittel:

- **Offizielle Dokumentationen** der verwendeten Technologien
- **Stack Overflow** und andere Entwicklerportale
- **GitHub** als Referenz für ähnliche Projekte
- **IDE-Funktionen** wie IntelliSense und Debugger

Nicht erlaubt:

- X Vollständige Code-Kopien aus bestehenden Projekten
- X KI-Tools für komplette Code-Generierung