

IT-Kolleg Imst

Kolleg/Aufbaulehrgang für Informatik

Aufgabenstellung zur Fachklausur

Software zur Erfassung der Luftqualität
auf den Tiroler Straßen



Klasse: 6AKIF/6AAIF
Datum: Jänner 2024
Gesamtbearbeitungszeit: 5 Stunden = 300 min

Name:

Benutzerkennung:

Bitte speichern Sie Ihre digitalen Ausarbeitungen auf dem **Netzlaufwerk**

Projektkurzbeschreibung

Als Softwareentwickler*in wurden Sie von der Europäischen Union mit der Erstellung von Softwarekomponenten für die Erfassung der Luftqualität auf Tiroler Straßen beauftragt.

Im Detail soll die Software für verschiedene Messstationen entlang der Tiroler Straßen die zugehörigen CO₂-Messungen anzeigen und ggf. manuell nacherfassen können. Zu diesem Zweck wird auch ein Anmeldebereich für privilegierte User:innen angeboten.

Die einzelnen Komponenten werden unter Verwendung von PHP und JavaScript als webgestützte Anwendung umgesetzt. Als Datenbankmanagementsystem kommt MariaDB zum Einsatz.

Teil A: Objektorientierung & Webservices

Teil B: Formularerstellung & http-Sessions

Teil C: Datenbankzugriff & Modularisierung

Vorbemerkungen

Die für die Bearbeitung der Aufgabenstellungen allenfalls notwendigen Dateien, Werkzeuge und weiteren Unterlagen (Dokumentation) werden Ihnen ergänzend zu den Angabeblättern digital zur Verfügung gestellt.

Außer den zur Verfügung gestellten Unterlagen sind keine weiteren Hilfsmittel gestattet.

Erlaubte Unterlagen und Werkzeuge

- DevDocs API Documentation für HTML, Bootstrap, PHP (devdocs.io)
- Software-Werkzeuge am Prüfungs-PC (XAMPP, VSCode, Firefox Portable)

Strukturierung der Lösungen

Alle Teillösungen sind in einem Projekt zu erstellen, wobei die Benennung folgendem Muster entspricht: **Projekt_NachnameVorname**

Abgabe der Lösungen

Am Ende der Prüfung erfolgt die Abgabe des Gesamtprojekts durch das Übertragen des Projektordners mit allen digital erstellten Dokumenten und Quelltextdateien auf das Abgabelaufwerk im **zip-Format**, wobei die Benennung folgendem Muster entspricht: **NachnameVorname**

Bitte **SPEICHERN Sie Ihre Lösungen LAUFEND** und achten Sie bei handschriftlicher Ausarbeitung auf die Lesbarkeit.

Beurteilung

Die Beurteilung erfolgt anhand des nachstehenden Beurteilungsrasters, der pro Prüfungsteil maximal 20 Punkte vorsieht.

Die **Gesamtbeurteilung** erfolgt anhand der Summe der in den drei Prüfungsteilen (A – C) erreichten Punkte:

<i>Beurteilung</i>	<i>Erreichte Punktezahl</i>	<i>Anteil Gesamtpunkteanzahl</i>
Sehr gut	≥ 54	$\geq 90 \%$
Gut	≥ 45	$\geq 80 \%$
Befriedigend	≥ 36	$\geq 65 \%$
Genügend ¹⁾	≥ 30	$\geq 50 \%$
Nicht genügend ¹⁾	< 30	$< 50 \%$

¹⁾ **Mindestanforderungen:** Die Gesamtleistung ist mit „Nicht genügend“ zu beurteilen, wenn **in einen oder mehreren Prüfungsteilen (A – C) weniger als 30 % der maximalen Punktezahl** erreicht wurden. Die Aufgabenteile können unabhängig voneinander bearbeitet werden.

Beurteilungsblatt

Kandidat*in:

Teil A

Wesentliche Bereiche	/ 14 Punkte
Zusätzliche Bereiche	/ 6 Punkte
Summe	/ 20 Punkte

Teil B

Wesentliche Bereiche	/ 14 Punkte
Zusätzliche Bereiche	/ 6 Punkte
Summe	/ 20 Punkte

Teil C

Wesentliche Bereiche	/ 14 Punkte
Zusätzliche Bereiche	/ 6 Punkte
Summe	/ 20 Punkte

Gesamtsumme	/ 60 Punkte
--------------------	--------------------

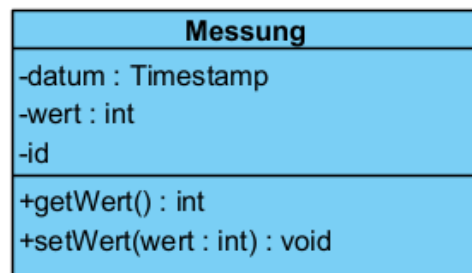
Beurteilung

Teil A: Objektorientierung & Webservices

A.1 Objektorientierte Programmierung

Wesentliche Bereiche

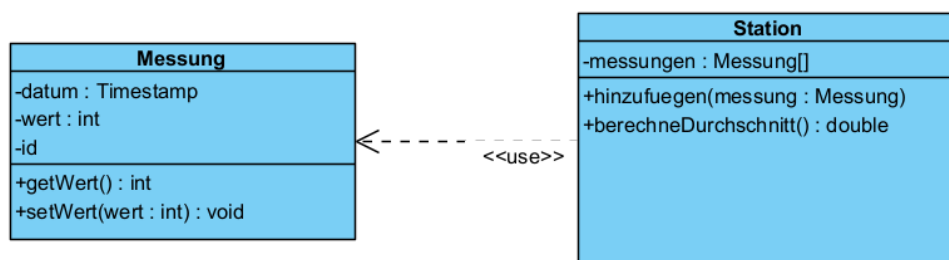
Erstellen Sie in der Datei a1_Messung.php die Klasse Messung in PHP. Achten Sie dabei auf Einhaltung wichtiger **Objektorientierter Konzepte** wie z.B. Kapselung, Verwendung von Konstruktoren, etc..



Erstellen Sie im Skript a1_messungstest.php ein **Array** mit mindestens drei Messungen inklusive unterschiedlichen Zeitstempeln und Messwerten. Berechnen Sie anschließend den **Durchschnittswert** aller Messungen im Array (Mittelwert) und geben Sie diesen aus.

Zusätzliche Bereiche

Erstellen Sie in der Datei a1_Station.php die Klasse Messstation in PHP. Die Methode `berechneDurchschnitt()` liefert den Mittelwert aller Messungen dieser Messstation zurück.



Erzeugen Sie im Skript index.php ein Klassenraum-Objekt und fügen Sie mindestens fünf Messungen hinzu. Berechnen Sie anschließend den Durchschnittswert der Messungen mit Hilfe der Methode `berechneDurchschnitt()` und geben Sie diesen aus.

A.2 Schnittstellen/API konsumieren

Wesentliche Bereiche

Implementieren Sie mittels AJAX (JavaScript oder jQuery) das asynchrone Laden von Messwerten über die gegebene Schnittstelle. Erstellen Sie dazu eine HTML-Datei mit dem Namen `a2_messwerte.html` mit einem Button zum Aufruf der Schnittstelle und stellen Sie die Messwerte und deren Eigenschaften in übersichtlicher Tabellenform dar.

URL: <http://localhost:8080/<PROJEKTVERZEICHNIS>/api.php>

Mockup:



ID	Datum	CO2-Wert
1	2026-11-22 11:27	899
2	2024-08-25 14:10	1230

JSON-Datenformat:

```
[
  {
    "id": 1,
    "datum": "2021-12-26 00:12:16",
    "wert": 2413
  },
  {
    "id": 2,
    "datum": "2022-04-15 12:40:35",
    "wert": 1735
  }, ...
]
```

Zusätzliche Bereiche

Fügen Sie Ihrer HTML-Seite ein Suchfeld hinzu und implementieren Sie mittels JavaScript oder jQuery eine Filterfunktion um die Messwerte (nach dem Laden und vor dem Anzeigen) zu durchsuchen. Nur das Datum soll durchsuchbar sein.

Mockup:

✕ 🔍

Daten laden

ID	Datum	CO2-Wert
1	2026-11-22 11:27	899,98
2	2026-11-22 14:10	897,11
:	:	:

Teil B: Formularerstellung & http-Sessions

Erstellen Sie ein Formular „**b_anmeldung.php**“ mit folgenden Feldern (siehe Abbildung 1):

Es werden keine Datenbankzugriffe durchgeführt, um den Login zu überprüfen. Prüfen Sie lediglich, ob der Benutzer „admin@euparlament.gov“ als Emailadresse und das Passwort „testitest“ angegeben wurde.

Falls die Eingabedaten nicht übereinstimmen, wird eine Fehlermeldung „**Falsche Benutzer- oder Passwortangaben!**“ ausgegeben.

Stimmen jedoch die eingegebenen Benutzerdaten, wird dies in einer Session vermerkt und auf das folgende Formular weitergeleitet.

 A mobile app wireframe for a login page. At the top is a header image with the text 'EU for Tyrol'. Below it is the title 'Anmelden'. There are two input fields: 'Email' with the value 'admin@euparlament.gov' and 'Passwort' with masked characters '.....'. A red button labeled 'Anmelden' is at the bottom.

Abbildung 1: Wireframe "anmeldung.php"

Erstellen Sie das Formular „**b_formular.php**“ mit folgenden Feldern (siehe Abbildung 2):

Felder in der Tabelle Stationen:

- s_id (INT PRIMARY autoincrement)
- s_bezeichnung (varchar 50)

Felder in der Tabelle Messungen:

- m_id (key autoincrement)
- m_zeitpunkt (timestamp)
- m_messwert (int)
- s_id (INT FK)

Beachten Sie, dass die Auswahl der Schule automatisch aus der Datenbank geladen wird und im Auswahlfeld zur Verfügung steht. Somit sind Fehleingaben ausgeschlossen.

Beim Klick auf den Button „**Speichern**“ sollen die Daten in der DB-Tabelle gespeichert werden (siehe Teil C).

 A mobile app wireframe for a measurement form. The title is 'Messung manuell erfassen'. It contains three input fields: 'Messstation' with a dropdown menu showing 'Auswählen', 'Zeitpunkt der Messung' with a date/time picker showing '11/06/2023 6:34', and 'Messwert CO2 in ppm' with a numeric input showing '12'. A red button labeled 'Speichern' is at the bottom. At the very bottom, it says 'Angemeldet als admin@euparlament.gov'.

Abbildung 2: Wireframe "formular.php"

Wesentliche Bereiche

1. Erstellen Sie die Formulare wie oben angegeben. Das Feld „**Messstation**“ ist ein Auswahlfeld das schreibgeschützt ist.
2. Die Felder „**Messstation, Zeitpunkt der Messung, Messwert**“ sind Pflichtfelder. Es muss geprüft werden, ob passende Daten eingegeben wurden. Prüfen Sie bitte client-seitig.

Zusätzliche Bereiche

3. Verwenden Sie die **Session** und zeigen Sie im Formular „**b_formular.php**“ den aktuellen angemeldeten Benutzer (Email) an.
4. Stylen Sie die Formulare mit Hilfe von Bootstrap, sodass diese den Wireframes nahekomen und auch für Mobilgeräte funktionieren.

Teil C: Datenbankzugriff & Modularisierung

Gegeben ist eine Datenbank zur Speicherung der erfassten Daten (database.sql) mit einer Tabelle Stationen und einer Tabelle Messungen, welche in Abbildung 3 und Abbildung 4 beschrieben sind. Die beiden Tabellen sind über eine Fremdschlüsselbeziehung miteinander verbunden.

#	Name	Typ	Kollation	Attribute	Null	Standard	Kommentare	Extra
<input type="checkbox"/> 1	s_id	int(11)			Nein	kein(e)		AUTO_INCREMENT
<input type="checkbox"/> 2	s_bezeichnung	varchar(50)	utf8mb4_general_ci		Nein	kein(e)		

Abbildung 3: Tabelle Stationen

#	Name	Typ	Kollation	Attribute	Null	Standard	Kommentare	Extra
1	m_id	int(11)			Nein	kein(e)		AUTO_INCREMENT
2	m_zeitpunkt	timestamp			Nein	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP
3	m_messwert	int(11)			Nein	kein(e)		
4	s_id	int(11)			Nein	kein(e)		

Abbildung 4: Tabelle Messungen

Wesentliche Bereiche

1. Importieren Sie die Datenbank über phpMyAdmin
2. Implementieren Sie jeweils eine Methode
 - a. zum Abrufen aller Stationen in c_station.php
 - b. zum Speichern der Daten aus dem Formular von Teil B in c_messung.php
3. Implementieren der Darstellung zum Anzeigen aller Klassenräume laut Abbildung 5: Wireframe „Liste aller Messstationen“

Wichtig: Der Datenbankzugriff muss SQL-Injection vermeiden.

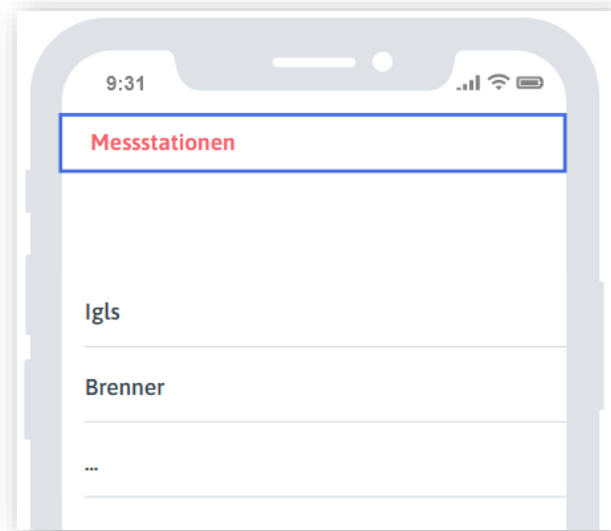


Abbildung 5: Wireframe „Liste aller Messstationen“

Zusätzliche Bereiche

Refaktorisieren Sie obige Implementierung entsprechend den Vorgaben, die sich aus **Active Record** und dem Grundkonzept des **objektrelationalen Mappings** ergeben.

Verwenden Sie dazu das bereitgestellte Interface (DatabaseObject.php).

Es sollte auch die Darstellung zum Anzeigen aller Messwerte entsprechend angepasst werden (siehe Abbildung 6).

Der Verbindungsaufbau zur Datenbank muss in einer eigenen Klasse gekapselt werden.



Abbildung 6: Wireframe „Liste aller Messwerte“