**Ein Bild, das Schrift, Screenshot, Kreis, Text enthält.

Automatisch generierte Beschreibung**

Abschlussprüfung Winter 2025

Fachinformatiker Anwendungsentwicklung  
Dokumentation zur betrieblichen Projektarbeit

Projekt:

**Entwicklung eines digitalen Anwesenheitstools für Get-Togethers**

Abgabetermin: 25.11.2024

**Prüfungsbewerber:**  
Konrad Reyhe  
Alfred-Kästner-Straße 84  
04275 Leipzig

**Ausbildungsbetrieb:**  
4K Analytics GmbH  
Markt 8  
04109 Leipzig

Ein Bild, das Grafiken, Logo, Kreis, Schrift enthält.

Automatisch generierte Beschreibung

Inhaltsverzeichnis

[Abkürzungsverzeichnis 1](#_Toc179296301)

[Abbildungsverzeichnis 1](#_Toc179296302)

[1.Einleitung 1](#_Toc179296303)

[1.1 Projektbeschreibung 1](#_Toc179296304)

[1.2 Projektziel 1](#_Toc179296305)

[1.3 Projektumfeld 2](#_Toc179296306)

[1.4 Projektbegründung 2](#_Toc179296307)

[1.5 Projektschnittstellen 2](#_Toc179296308)

[1.6 Projektabgrenzung 2](#_Toc179296309)

[2. Projektplanung 2](#_Toc179296310)

[2.1 Projektphasen 2](#_Toc179296311)

[2.2 Ressourcenplanung 3](#_Toc179296312)

[2.3 Entwicklungsprozess 3](#_Toc179296313)

[3. Analysephase 3](#_Toc179296314)

[3.1 Ist-Analyse 3](#_Toc179296315)

[3.2 Wirtschaftlichkeitsanalyse 3](#_Toc179296316)

[4. Entwurfsphase 3](#_Toc179296317)

[4.1 Zielplattform 3](#_Toc179296318)

[4.2 Architekturdesign 4](#_Toc179296319)

[4.3 Entwurf der Benutzeroberfläche 5](#_Toc179296320)

[4.4 Datenmodell 6](#_Toc179296321)

[5. Implementierungsphase 6](#_Toc179296322)

[5.1 Iterationsplanung 6](#_Toc179296323)

[5.2 Implementierung der Datenstrukturen 6](#_Toc179296324)

[5.3 Implementierung der Geschäftslogik 7](#_Toc179296325)

[5.4 Tests und Qualitätssicherung 9](#_Toc179296326)

[6. Abnahme- und Einführungsphase 12](#_Toc179296327)

[6.1 Abnahme durch den Fachbereich 12](#_Toc179296328)

[7. Dokumentation 12](#_Toc179296329)

[7.1 Entwicklerdokumentation 12](#_Toc179296330)

[7.2 Benutzerhandbuch 12](#_Toc179296331)

[7.2.1 Einleitung 12](#_Toc179296332)

[7.2.2 Starten der Anwendung 12](#_Toc179296333)

[7.2.3 Anmeldung als Mitarbeiter 13](#_Toc179296334)

[7.2.4 Anmeldung als Gast 13](#_Toc179296335)

[7.2.5 Beenden des Get-Togethers und Export der Daten 13](#_Toc179296336)

[8. Fazit 13](#_Toc179296337)

[8.1 Soll-/Ist-Vergleich 13](#_Toc179296338)

[8.2 Lessons Learned 14](#_Toc179296339)

[10. Quellenverzeichnis 14](#_Toc179296340)

[11. Anhang 14](#_Toc179296341)

# Abkürzungsverzeichnis

|  |  |
| --- | --- |
| **Abkürzung** | **Bedeutung** |
| API | Application Programming Interface |
| CSV | Comma-Separated Values |
| CRUD | |  | | --- | |  |  |  | | --- | | Create, Read, Update, Delete | |
| HTTP | Hypertext Transfer Protocol |
| GUI | Graphical User Interface |

# Abbildungsverzeichnis

[Abbildung 1: Projektphasen 2](#_Toc179295085)

[Abbildung 2: Systemarchitektur des Anwesenheitstools 4](#_Toc179295086)

[Abbildung 3: Benutzeroberfläche der Anwendung 5](#_Toc179295087)

[Abbildung 4: Datenmodell der Anwendung 6](#_Toc179295088)

[Abbildung 5: Codebeispiel zur Speicherung der Anwesenheitsdaten 7](#_Toc179295089)

[Abbildung 6: Codebeispiel zur Exportfunktion der Anwesenheitsliste 8](#_Toc179295090)

[Abbildung 7: Testergebnisse 11](#_Toc179295091)

# 1.Einleitung

## 1.1 Projektbeschreibung

Im Rahmen des Projekts „Entwicklung eines digitalen Anwesenheitstools für Get-Togethers“ wurde ein System entwickelt, das die Anwesenheit von Mitarbeitern und Gästen bei Firmenveranstaltungen digital erfasst. Die Lösung ersetzt die handschriftlich geführten Listen und ermöglicht den Export der Daten in eine CSV-Datei, um sie für die Buchhaltung weiterzuverarbeiten. Die Anwendung läuft auf einem Tablet im Kiosk-Modus, um unbefugte Zugriffe zu verhindern.

## 1.2 Projektziel

Das Ziel des Projekts ist die Bereitstellung einer benutzerfreundlichen und sicheren Lösung, die die manuelle Erfassung der Anwesenheit durch ein digitales Tool ersetzt. Der Fokus liegt auf der Effizienzsteigerung und der Reduzierung von Fehlern bei der Erfassung und Verarbeitung der Daten.

## 1.3 Projektumfeld

Das Tool wird in der 4K Analytics GmbH eingesetzt, die regelmäßig Firmen-Get-Togethers organisiert. Bislang erfolgte die Anwesenheitserfassung manuell, was zu Problemen wie unleserlicher Schrift und Verzögerungen bei der Datenverarbeitung führte.

## 1.4 Projektbegründung

Durch die Digitalisierung des Prozesses wird nicht nur die Effizienz erhöht, sondern auch die Genauigkeit der Datenverarbeitung verbessert. Zudem werden potenzielle Fehlerquellen, wie der Verlust von Listen, vermieden.

## 1.5 Projektschnittstellen

Die Anwendung arbeitet vollständig lokal und benötigt keine externen Schnittstellen. Die Daten werden lokal gespeichert und können nach Bedarf exportiert werden.

## 1.6 Projektabgrenzung

Das Projekt umfasst ausschließlich die Erfassung und den Export der Anwesenheitsdaten. Eine Integration in andere Systeme wie das ERP oder Buchhaltungssystem ist nicht Teil des Projekts.

# 2. Projektplanung

## 2.1 Projektphasen

|  |  |
| --- | --- |
| **Phase** | **Dauer (Stunden)** |
| Analysephase | 9 |
| Entwurfsphase | 10 |
| Zeit- und Ablaufplanung | 4 |
| Ressourcen- und Kostenplanung | 4 |
| Design und Entwicklung | 30 |
| Tests und Qualitätssicherung | 10 |
| Projektdokumentation | 7 |
| **Gesamt** | **80 Stunden** |

Abbildung 1: Projektphasen

## 2.2 Ressourcenplanung

Die für das Projekt benötigten Ressourcen umfassen:

* **Hardware**: Tablet für die Anwendung, Laptop für die Entwicklung
* **Software**: Python, Flask, Streamlit, CSV-Editor
* **Personal**: Entwickler (Fachinformatiker), Tester, Projektleiter

## 2.3 Entwicklungsprozess

Die Entwicklung erfolgte agil, wobei regelmäßiges Feedback vom Fachbereich eingeholt wurde. Dies stellte sicher, dass die Anforderungen vollständig erfüllt wurden und das Design den Wünschen entsprach.

# 3. Analysephase

## 3.1 Ist-Analyse

Derzeit erfolgt die Erfassung der Anwesenheit handschriftlich, was zu Verzögerungen und Fehlern bei der Datenverarbeitung führt. Ein digitales Tool soll diesen Prozess automatisieren und effizienter gestalten.

## 3.2 Wirtschaftlichkeitsanalyse

Durch die Einführung der digitalen Lösung wird Zeit eingespart und der Prozess der Datenverarbeitung optimiert.

Die Amortisation erfolgt innerhalb eines Jahres durch die Reduktion der manuellen Arbeit.

# 4. Entwurfsphase

## 4.1 Zielplattform

Die Anwendung wurde für Tablets entwickelt, die im Kiosk-Modus betrieben werden, um den Zugriff auf andere Apps zu unterbinden.

## 4.2 Architekturdesign

Die Anwendung basiert auf einer Microservices-Architektur, bei der Streamlit für das Frontend und Flask für das Backend verwendet wird. Die Daten, die über die Benutzeroberfläche eingegeben werden, werden per API-Anfrage an das Backend gesendet. Das Backend speichert die erfassten Daten in einer CSV-Datei, die bei Bedarf exportiert werden kann. Diese Architektur gewährleistet eine klare Trennung von Benutzeroberfläche und Logik sowie eine einfache Erweiterbarkeit der Anwendung.

In **Abbildung 1** wird die Architektur des Systems dargestellt, bei der die Kommunikation zwischen Frontend, Backend und der CSV-Datenbank visualisiert wird.

Ein Bild, das Text, Screenshot, Diagramm, Reihe enthält.

Automatisch generierte Beschreibung

Abbildung 2: Systemarchitektur des Anwesenheitstools

## 4.3 Entwurf der Benutzeroberfläche



Abbildung 3: Benutzeroberfläche der Anwendung

## 4.4 Datenmodell

Abbildung 4: Datenmodell der Anwendung

# 5. Implementierungsphase

## 5.1 Iterationsplanung

* **Iteration 1**: Grundlegende Benutzeroberfläche
* **Iteration 2**: Backend-Logik und Datenexport
* **Iteration 3**: Tests und Qualitätssicherung

## 5.2 Implementierung der Datenstrukturen

Die Daten werden in CSV-Dateien gespeichert, die sich leicht exportieren lassen.

## 5.3 Implementierung der Geschäftslogik

Codebeispiel zur Speicherung der Anwesenheitsdaten

Ein Bild, das Text, Screenshot enthält.

Automatisch generierte Beschreibung

Abbildung 5: Codebeispiel zur Speicherung der Anwesenheitsdaten

**Erklärung des Codes:**

* **Flask route /add**: Diese Route wird verwendet, um Anwesenheitsdaten über eine POST-Anfrage hinzuzufügen.
* **csv.writer**: Hier wird csv.writer verwendet, um die übermittelten Namen und die Firmenzugehörigkeit in der CSV-Datei **attendance.csv** zu speichern. Diese Datei enthält alle erfassten Anwesenheiten.
* **Fehlerbehandlung**: Falls während des Schreibvorgangs ein Fehler auftritt, wird eine Fehlermeldung zurückgegeben. Dies sichert die Integrität des Prozesses.

Codebeispiel zur Exportfunktion der Anwesenheitsliste

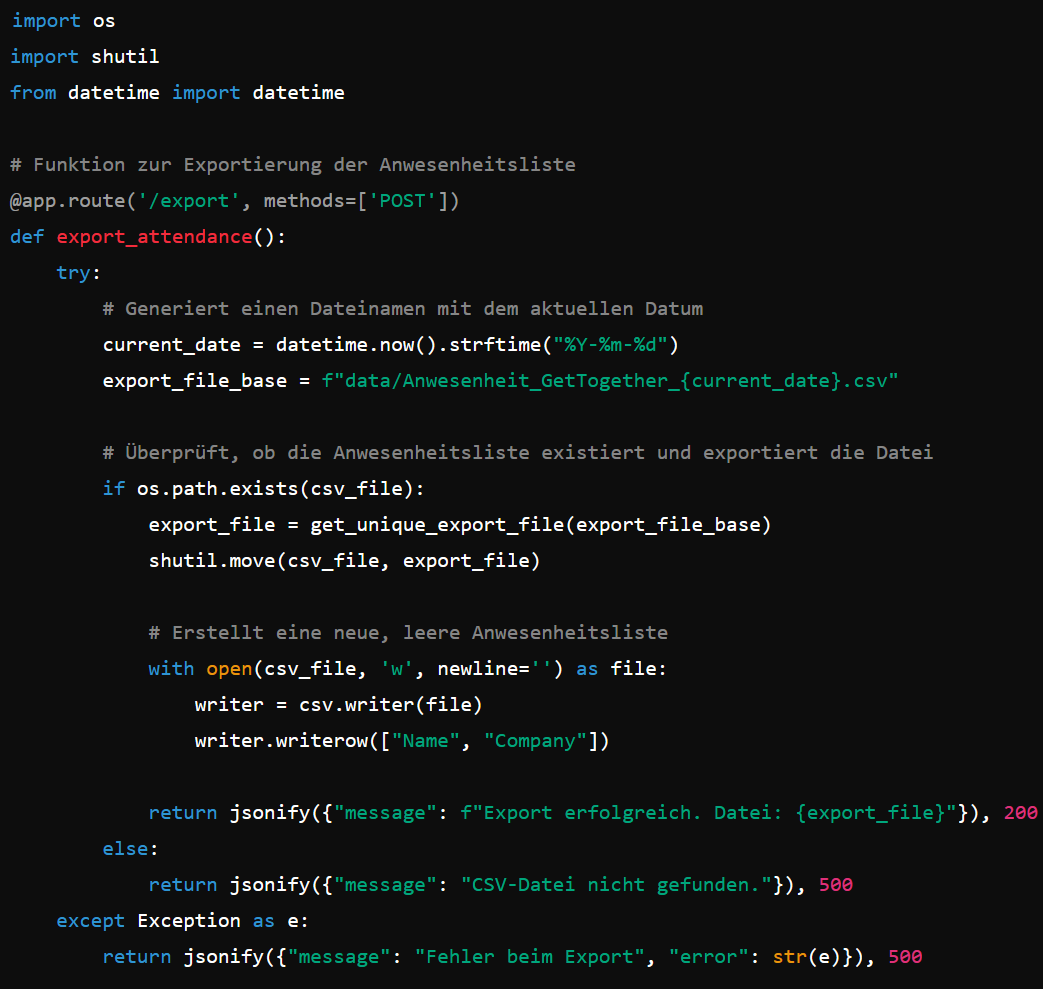


Abbildung 6: Codebeispiel zur Exportfunktion der Anwesenheitsliste

**Erklärung des Codes:**

* **Export der Anwesenheitsliste**: Diese Funktion verschiebt die Datei **attendance.csv** und benennt sie in **Anwesenheit\_GetTogether\_[Datum].csv** um. Der Name enthält das Datum, um Verwechslungen mit früheren Exporten zu vermeiden.
* **shutil.move**: Verwendet, um die Datei zu verschieben und umzubenennen. Die ursprüngliche Anwesenheitsliste wird nach dem Export gelöscht, und eine neue, leere Datei wird erstellt, um zukünftige Daten aufzunehmen.
* **Fehlerbehandlung**: Falls beim Export ein Fehler auftritt (z. B. wenn die Datei nicht existiert), wird eine Fehlermeldung zurückgegeben.

## 5.4 Tests und Qualitätssicherung

Das Ziel dieses Kapitels ist es, die wichtigsten Tests detailliert zu beschreiben, die zur Sicherstellung der Funktionalität und Qualität deiner Anwendung durchgeführt wurden.

**Funktionstests:**

Diese Tests konzentrieren sich darauf, die verschiedenen Funktionen des Anwesenheitstools zu überprüfen. Hier geht es darum zu zeigen, dass die wichtigsten Features der Anwendung korrekt implementiert wurden und wie sie auf bestimmte Eingaben reagieren.

1. **Speichern der Anwesenheit**:
   * **Ziel**: Testen, ob die Eingaben der Benutzer korrekt gespeichert werden.
   * **Beschreibung**: Hier wurde überprüft, ob ein Mitarbeiter nach Auswahl der Firma und des Namens erfolgreich als „anwesend“ markiert wird und die Daten korrekt in die CSV-Datei geschrieben werden.
   * **Durchführung**: Es wurde getestet, ob die Anwesendheitsdaten in verschiedenen Szenarien korrekt erfasst werden, z.B. bei der Anmeldung eines Gastes oder eines Mitarbeiters.
   * **Ergebnis**: Der Test zeigte, dass die Daten in allen Fällen korrekt erfasst wurden, und keine Daten verloren gingen.
   * **Beispielhafter Testfall**:
     + **Eingabe**: Firma „4K Analytics GmbH“, Mitarbeiter „Anna Müller“.
     + **Erwartetes Ergebnis**: Der Name „Anna Müller“ und die Firma „4K Analytics GmbH“ werden erfolgreich in die CSV-Datei geschrieben.
2. **Export der Anwesenheitsliste**:
   * **Ziel**: Testen, ob die gespeicherten Anwesenheitsdaten korrekt in einer CSV-Datei exportiert werden.
   * **Beschreibung**: Hier wurde überprüft, ob beim Beenden des Get-Togethers die Daten ordnungsgemäß exportiert und in einer Datei mit korrektem Namen (z.B. „Anwesenheit\_GetTogether\_2024-11-25.csv“) gespeichert werden.
   * **Durchführung**: Es wurde getestet, ob die exportierte Datei korrekt benannt wird, und ob die darin enthaltenen Daten vollständig und korrekt sind.
   * **Ergebnis**: Der Test zeigte, dass die Datei erfolgreich exportiert wurde. Probleme traten beim Dateinamen auf, der durch eine zusätzliche Prüfung im Code behoben wurde.

**Usability-Tests:**

In Usability-Tests geht es darum, zu überprüfen, wie intuitiv und benutzerfreundlich die Anwendung für verschiedene Benutzer ist. Ziel ist es zu sehen, ob Nutzer ohne technisches Vorwissen die Anwendung leicht bedienen können.

**Beschreibung von Usability-Tests:**

1. **Benutzerfreundlichkeit des Anmeldeprozesses**:
   * **Ziel**: Testen, ob die Mitarbeiter und Gäste sich einfach und ohne Probleme anmelden können.
   * **Beschreibung**: Mehrere Benutzer ohne technische Vorkenntnisse wurden gebeten, die Anwendung zu nutzen. Sie sollten sich als Mitarbeiter oder Gast anmelden und ihre Anwesenheit speichern.
   * **Durchführung**: Dabei wurde beobachtet, wie intuitiv die Benutzeroberfläche gestaltet ist und wie leicht die Benutzer ihre Anwesenheit speichern konnten.
   * **Ergebnis**: Das Feedback der Benutzer war größtenteils positiv. Es wurde festgestellt, dass der Dropdown zur Auswahl der Mitarbeiter für Personen mit sehr vielen Mitarbeitern optimiert werden könnte, z.B. durch eine Suchfunktion.
   * **Optimierungen**: Es wurde vorgeschlagen, den Dropdown-Mechanismus zu verbessern, damit Benutzer schneller ihren Namen finden können.

**Export-Test:**

Der Export-Test fokussiert sich darauf, ob die Daten ordnungsgemäß aus der Anwendung exportiert werden können.

**Beschreibung des Export-Tests:**

1. **Exportfunktionalität**:
   * **Ziel**: Testen, ob die Anwesenheitsdaten korrekt exportiert und auf dem Tablet gespeichert werden.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test-ID** | **Funktion** | **Eingabe** | **Erwartetes Ergebnis** | **Tatsächliches Ergebnis** | **Status** |
| 1 | Speichern der Anwesenheit | Name: Anna Müller, Firma: 4K Analytics GmbH | Anwesenheit gespeichert | Anwesenheit gespeichert | Erfolgreich |
| 2 | Export der Daten | - | CSV-Datei „Anwesenheit\_GetTogether\_[Datum].csv“ wird erstellt | Datei korrekt erstellt | Erfolgreich |

* + **Beschreibung**: Es wurde getestet, ob die Exportfunktion bei verschiedenen Veranstaltungen und Zeitpunkten ordnungsgemäß funktioniert. Zusätzlich wurde geprüft, ob bei mehreren Exports am selben Tag automatisch unterschiedliche Dateinamen generiert werden.
  + **Durchführung**: Ein Export wurde durchgeführt, nachdem mehrere Benutzer ihre Anwesenheit registriert hatten. Es wurde überprüft, ob der Dateiname korrekt erstellt wurde und ob die exportierten Daten vollständig sind.
  + **Ergebnis**: Der Test verlief erfolgreich. Der Dateiname wurde korrekt erstellt und die exportierte CSV-Datei enthielt die korrekten Daten.

Abbildung 7: Testergebnisse

Die durchgeführten Tests bestätigten, dass alle Kernfunktionen des Anwesenheitstools einwandfrei funktionieren. Insbesondere der Anmeldeprozess und die Exportfunktion wurden erfolgreich getestet.

# 6. Abnahme- und Einführungsphase

## 6.1 Abnahme durch den Fachbereich

Die Abnahme wurde durch das Personalwesen und die Buchhaltung durchgeführt, wobei die Anwendung auf ihre Funktionalität getestet wurde.

# 7. Dokumentation

## 7.1 Entwicklerdokumentation

* **Backend**: Flask, CSV-Datenbank
* **Frontend**: Streamlit
* **Sicherheitsmaßnahmen**: Verschlüsselung der CSV-Dateien

## 7.2 Benutzerhandbuch

7.2.1 Einleitung  
Das digitale Anwesenheitstool für Get-Togethers wurde entwickelt, um die Anwesenheit von Mitarbeitern und Gästen einfach und effizient zu erfassen. Die Anwendung läuft auf einem Tablet im Kiosk-Modus und bietet eine intuitive Benutzeroberfläche zur schnellen Registrierung und Datenerfassung. Dieses Handbuch führt durch die grundlegenden Funktionen der Anwendung.

### 7.2.2 Starten der Anwendung

1. Schalten Sie das Tablet ein, auf dem die Anwendung installiert ist.
2. Öffnen Sie die Anwendung durch Antippen des Symbols „Anwesenheitstool“.
3. Die Anwendung wird automatisch im Kiosk-Modus gestartet, sodass nur die Nutzung des Tools und keine weiteren Funktionen des Tablets verfügbar sind.

### 7.2.3 Anmeldung als Mitarbeiter

1. Wählen Sie „Mitarbeiter“ auf der Startseite der Anwendung aus.
2. Im Dropdown-Menü „Firma auswählen“ können Sie die entsprechende Firma aus der Liste (z.B. 4K Analytics GmbH, visgato, etc.) auswählen.
3. Wählen Sie anschließend im Dropdown-Menü „Mitarbeiter auswählen“ Ihren Namen.
4. Drücken Sie auf „Anwesenheit speichern“, um Ihre Anwesenheit zu erfassen.
5. Nach erfolgreicher Anmeldung erscheint eine Bestätigungsmeldung.

### 7.2.4 Anmeldung als Gast

1. Wählen Sie „Gast“ auf der Startseite der Anwendung aus.
2. Geben Sie Ihren Namen im Feld „Bitte geben Sie Ihren Namen ein“ ein.
3. Drücken Sie auf „Anwesenheit speichern“, um Ihre Anwesenheit zu erfassen.
4. Nach erfolgreicher Anmeldung erscheint eine Bestätigungsmeldung.

### 7.2.5 Beenden des Get-Togethers und Export der Daten

1. Der Bestätigungsdialog „Sind Sie sicher, dass Sie das GetTogether beenden möchten?“ steht während der gesamten Veranstaltung zur Verfügung.
2. Markieren Sie das Kontrollkästchen „Bestätigen“, um den Export zu bestätigen.
3. Sobald das Kontrollkästchen markiert ist, erscheint der Button „GetTogether beenden und exportieren“. Drücken Sie auf diesen Button, um den Export durchzuführen.
4. Die Anwesenheitsliste wird als CSV-Datei mit dem Namen „Anwesenheit\_GetTogether\_[Datum].csv“ exportiert und auf dem Tablet gespeichert.
5. Nach erfolgreichem Export erscheint eine Bestätigungsmeldung auf dem Bildschirm, die den erfolgreichen Export bestätigt.

# 8. Fazit

## 8.1 Soll-/Ist-Vergleich

Das Projekt wurde innerhalb des Zeitplans abgeschlossen und die geplanten Funktionen wurden erfolgreich umgesetzt.

## 8.2 Lessons Learned

Die enge Zusammenarbeit mit dem Fachbereich und die iterative Entwicklung führten zu einer erfolgreichen Umsetzung der Anforderungen.

# 10. Quellenverzeichnis

- Flask Dokumentation: <https://flask.palletsprojects.com/> (abgerufen am: 07.10.2024)

- Python CSV Modul: <https://docs.python.org/3/library/csv.html> (abgerufen am: 07.10.2024)

- Streamlit Dokumentation: <https://docs.streamlit.io/> (abgerufen am: 07.10.2024)

- shutil Modul Dokumentation: <https://docs.python.org/3/library/shutil.html> (abgerufen am: 07.10.2024)

- requests Modul Dokumentation: https://docs.python-requests.org/en/latest/ (abgerufen am: 07.10.2024)

- Flask RESTful Dokumentation: <https://flask-restful.readthedocs.io/> (abgerufen am: 07.10.2024)

- Python Zeitstempel Dokumentation: <https://docs.python.org/3/library/datetime.html> (abgerufen am: 07.10.2024)

# 11. Anhang

* Mit Auflagen genehmigter **Projektantrag**