

# **Unser Thema ist Mediahub**

## **DIPLOMARBEIT**

verfasst im Rahmen der

**Reife- und Diplomprüfung**

an der

**Höheren Abteilung für IT-Medientechnik**

Eingereicht von:

Markus Tran  
Abdulkerim Cufurovic

Betreuer:

Robert Reder

Leonding, 4. April 2025

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Die Arbeit wurde bisher in gleicher oder ähnlicher Weise keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Die vorliegende Diplomarbeit ist mit dem elektronisch übermittelten Textdokument identisch.

Leonding, 4. April 2025

Markus Tran & Abdulkerim Cufurovic

# Inhaltsverzeichnis

|   |          |
|---|----------|
| <b>1 Einleitung</b>   | <b>1</b> |
| 1.1 Entstehung der Projektidee . . . . .                        | 1        |
| 1.2 Zielsetzung des Projekts . . . . .                          | 1        |
| 1.3 Aufbau der Diplomarbeit . . . . .                           | 1        |
| <b>2 Problemstellungen</b>                                      | <b>2</b> |
| 2.1 Aktuelle Herausforderungen bei der Abo-Verwaltung . . . . . | 2        |
| 2.2 Technische Problemfelder . . . . .                          | 2        |
| 2.3 Anforderungen an die geplante Lösung . . . . .              | 2        |
| <b>3 Technologische Grundlagen</b>                              | <b>3</b> |
| 3.1 Angular . . . . .   | 4        |
| 3.2 Quarkus . . . . .   | 4        |
| 3.3 Keycloak . . . . .  | 4        |
| 3.4 PostgreSQL . . . . .  | 4        |
| 3.5 Docker . . . . .  | 4        |
| 3.6 Tailwind CSS . . . . .                                      | 4        |
| 3.7 GitHub . . . . .  | 4        |
| <b>4 APIs im Projekt</b>  | <b>5</b> |
| 4.1 Grundlagen von APIs . . . . .                               | 5        |
| 4.2 Evaluierte API-Optionen . . . . .                           | 5        |
| 4.3 Movie API – Entscheidungsfaktoren . . . . .                 | 5        |
| 4.4 Begründung der finalen API-Auswahl . . . . .                | 5        |
| 4.5 Implementierungsdetails . . . . .                           | 5        |
| <b>5 Architektur und Implementierung</b>                        | <b>6</b> |
| 5.1 Systemarchitektur . . . . .                                 | 6        |
| 5.2 Datenbankstruktur . . . . .                                 | 6        |

|                              |   |            |
|------------------------------|---|------------|
| 5.3                          | REST-Schnittstellen . . . . .                     | 6          |
| 5.4                          | Frontend-Architektur . . . . .                    | 6          |
| 5.5                          | Sicherheitsarchitektur mit Keycloak . . . . .     | 6          |
| <b>6</b>                     | <b>Projektdurchführung</b>                        | <b>7</b>   |
| 6.1                          | Planung und Arbeitsweise . . . . .                | 7          |
| 6.2                          | Entwicklungsschritte . . . . .                    | 7          |
| 6.3                          | Testverfahren . . . . .                           | 7          |
| 6.4                          | Herausforderungen . . . . .                       | 7          |
| 6.5                          | Lessons Learned . . . . .                         | 7          |
| <b>7</b>                     | <b>Ergebnisse und Ausblick</b>                    | <b>8</b>   |
| 7.1                          | Erfüllung der Anforderungen . . . . .             | 8          |
| 7.2                          | Bewertung der eingesetzten Technologien . . . . . | 8          |
| 7.3                          | Mögliche Erweiterungen . . . . .                  | 8          |
| <b>8</b>                     | <b>Zusammenfassung</b>                            | <b>9</b>   |
| <b>Glossar</b>               |   | <b>III</b> |
| <b>Literaturverzeichnis</b>  |   | <b>IV</b>  |
| <b>Abbildungsverzeichnis</b> |   | <b>V</b>   |
| <b>Tabellenverzeichnis</b>   |   | <b>VI</b>  |
| <b>Quellcodeverzeichnis</b>  |   | <b>VII</b> |

# **1 Einleitung**

## **1.1 Entstehung der Projektidee**

Text ...

## **1.2 Zielsetzung des Projekts**

Text ...

## **1.3 Aufbau der Diplomarbeit**

Text ...

## **2 Problemstellungen**

### **2.1 Aktuelle Herausforderungen bei der Abo-Verwaltung**

Text ...

### **2.2 Technische Problemfelder**

Text ...

### **2.3 Anforderungen an die geplante Lösung**

Text ...



## **3 Technologische Grundlagen**

### **3.1 Angular**

**3.1.1 Einordnung und Funktionsweise**

**3.1.2 Single-Page-Application-Konzept**

**3.1.3 Begründung der Wahl von Angular gegenüber React**

**3.1.4 Komponenten, Data Binding und Services**

### **3.2 Quarkus**

**3.2.1 Grundlagen**

**3.2.2 Vorteile im Kontext dieses Projekts**

**3.2.3 Implementierung der API**

### **3.3 Keycloak**

**3.3.1 Grundlagen**

**3.3.2 Ablauf des Login-Prozesses**

**3.3.3 Rollen- und Benutzerverwaltung**

### **3.4 PostgreSQL**

**3.4.1 Datenhaltung und Struktur**

**3.4.2 Vorteile relationaler Datenbanken**

**3.4.3 Mögliche Alternativen**

# **4 APIs im Projekt**

**4.1 Grundlagen von APIs**

**4.2 Evaluierte API-Optionen**

**4.3 Movie API – Entscheidungsfaktoren**

**4.4 Begründung der finalen API-Auswahl**

**4.5 Implementierungsdetails**

# **5 Architektur und Implementierung**

## **5.1 Systemarchitektur**

## **5.2 Datenbankstruktur**

## **5.3 REST-Schnittstellen**

## **5.4 Frontend-Architektur**

## **5.5 Sicherheitsarchitektur mit Keycloak**

# **6 Projektdurchführung**

**6.1 Planung und Arbeitsweise**

**6.2 Entwicklungsschritte**

**6.3 Testverfahren**

**6.4 Herausforderungen**

**6.5 Lessons Learned**

# **7 Ergebnisse und Ausblick**

**7.1 Erfüllung der Anforderungen**

**7.2 Bewertung der eingesetzten Technologien**

**7.3 Mögliche Erweiterungen**

# **8 Zusammenfassung**

# Glossar

**GUID** Globally Unique Identifier

# **Literaturverzeichnis**

# **Abbildungsverzeichnis**

# **Tabellenverzeichnis**

# **Quellcodeverzeichnis**