

**GFOS** Innovationsaward 2024

## **TrackIn - Zeiterfassung**

von Cuma Erarslan, Fahri Emin Dogan  
Heinz-Nixdorf Berufskolleg, HI22A und HI21A



# **Dokumentation**

**Thema: Entwicklung einer responsiven Java- und JavaScript-basierten  
Anwendung**



## Inhaltsverzeichnis

1.0 Einführung.....	5
1.1    Mindestanforderungen .....	5
1.2    Erlaubte Technologien .....	5
1.3    Leistungsumfang .....	6
2.0 Projektplanung.....	7
2.1 Kommunikation .....	7
2.2 GitHub .....	7
2.3 Aufgabenverteilung .....	7
2.4 Code Richtlinien .....	8
2.5 Entwicklungsumgebung.....	8
3.0 Verwendete Technologien und Planung .....	9
3.1 Anwendungen.....	9
3.1.1 Bootstrap Studio .....	9
3.1.2 Microsoft Azure .....	9
3.3 Maven-Konfiguration und Bibliotheken .....	10
3.4 Verwendete JavaScript-Bibliotheken und Frameworks .....	10
3.4.1 Js-Year-Calendar .....	10
3.4.2 Chart.js .....	11
3.4.3 Bootstrap .....	11
3.5 Planung .....	11
3.5.1 Ziele.....	11
3.5.2 Wunschziele .....	12
4.0 Benutzeroberfläche und Webseite-Features .....	13
4.1 Anwendungsmöglichkeiten .....	13
4.1.1 Gerätekompatibilität .....	14
4.2 Registrierung.....	14
4.2.1 Registrierungsprozess:.....	15
4.2.2 Passwortvergessen-Funktionalität: .....	15
4.2.3 Meldungen.....	15
4.3 Anmeldung.....	16
4.4 Seitenleiste und Navigationsleiste.....	17
4.4.1 Seitenleiste .....	18
4.4.2 Navigationsleiste.....	19



4.4.3 Fußbereich .....	19
4.4.4 Impressum .....	20
4.4.5 Datenschutz .....	20
4.4.6 Error 404-Seite .....	21
4.5 Hauptseite .....	22
4.5.1 Statistiken .....	22
4.5.2 Diagramme .....	23
4.5.3 Benutzerdefinierte Einträge .....	24
4.6 Kalender .....	26
4.6.1 Dropdown-Menü .....	27
4.6.2 Statusvergabe .....	27
4.7 Mitteilungen .....	29
4.8 Profil .....	30
4.9 Verwaltung .....	31
4.9.1 Tabellenübersicht .....	31
4.9.2 Mitarbeiter Hinzufügen .....	33
4.9.3 Mitarbeiter Bearbeiten .....	34
4.9.4 Mitarbeiter Entfernen .....	34
4.9.5 Kalender .....	34
5.0 Datenbank / Datenbankarchitektur .....	36
5.1 Datenbankhosting .....	36
5.1.1 Warum Microsoft Azure? .....	36
5.2 Sicherheitsmaßnahmen .....	36
5.3 Datenbankzugriffsschicht mit Hibernate und SQL-Query .....	37
5.4 Datenbankstruktur .....	37
5.4.1 Mitarbeiter .....	38
5.4.2 Logindata .....	39
5.4.3 Token .....	40
5.4.4 Days .....	41
5.4.5 Entries .....	42
5.4.6 Message .....	43
6.0 Quellcode Übersicht .....	44
6.1 Projekt Struktur .....	44
6.2 Hibernate .....	45
6.2.1 Hibernate-Konfiguration .....	45
6.2.2 Entitätsklassen .....	45



6.2.3 Datenbankoperation mit Hibernate .....	46
6.2.4 Beziehung zwischen Entitäten .....	46
6.3.1 Sessions .....	47
6.3.2 Filter .....	47
6.3.3 Servlets .....	48
6.3.4 Hilfsklassen .....	50
6.4 Hintergrundprozesse .....	52
6.5 JSP-Seiten und Web-Konfiguration .....	54
6.5.1 Web-Konfiguration .....	54
6.5.2 Java Server Pages .....	54
6.6 Javascript .....	56
7.0 Installation .....	59
8.0 Schlusswort .....	63



## 1.0 Einführung

Die Kernidee unserer Aufgabenstellung fokussierte sich auf die Entwicklung eines webbasierten und Responsiven Zeiterfassungssystems auf **Java- und Javascript-Basis**. Dieses Projekt wurde erfolgreich von einem zweiköpfigen Team umgesetzt, bestehend aus Schülern des Heinz-Nixdorf-Berufskollegs in Essen.

Die technische Expertise wurde durch ausführliche Recherche und **“Learning-by-Doing”** erworben. Unsere Dokumentation bietet Einblicke in den Entwicklungsprozess, die Herausforderungen und die erlangten Erkenntnisse während der Umsetzung dieses webbasierten Systems.

Unser Zuständiger Lehrer war **Herrn Weber** um unsere Organisatorischen Belange zu kümmern wie z.b. die Anmeldung Für Fragen im Zusammenhang mit dem Wettbewerb haben wir uns an Frau Kirsten Kipper gewandt.

### 1.1 Mindestanforderungen

Die von GFOS festgelegte **Aufgabenstellung**<sup>1</sup> umfasst wesentliche Anforderungen an das Programm, die implementiert werden müssen, darunter:

- Benutzeranmeldung, Benutzerabmeldung und -authentifizierung
- Sitzungs- und Timeouthandling
- Anzeige und Bearbeitung des Anwesenheitsstatus
- Anpassungsmöglichkeit von Stammdaten
- Stammdatenanpassung
- Responsive Visualisierung der Daten
- Meldungshandling
- Datenverarbeitung und -validierung

**Optionale Funktionen** können implementiert werden.

### 1.2 Erlaubte Technologien

Für die Umsetzung sind Folgende **Technologien**<sup>1</sup> erlaubt:

- Java Enterprise Edition (inkl. GlassFish-Server o.ä)
- HTML, CSS & JavaScript (inkl. JQuery und ähnlicher Bibliotheken)
- Frameworks zur Visualisierung
- MySQL oder Java DB

Nicht erlaubt sind unter anderem:

- PHP
- C#
- Python
- Perl

---

<sup>1</sup>

[https://www.gfos.com/uploads/pdfs/Innovationsaward/GFOS\\_Innovationsaward\\_2024\\_Aufgabenstellung.pdf](https://www.gfos.com/uploads/pdfs/Innovationsaward/GFOS_Innovationsaward_2024_Aufgabenstellung.pdf)



## 1.3 Leistungsumfang

Um den festgelegten Anforderungen an den Einreichungsumfang gerecht zu werden, müssen verschiedene Aspekte berücksichtigt werden:

1. **Komplettes Softwareprojekt:** Die Einreichung sollte den gesamten Quellcode, die vollständige Anwendung und Exporte der Datenbank umfassen.
2. **Dokumentation zur Realisierung und Installation:**
  - *Benutzerhandbuch:* Enthalten sollte die Dokumentation mindestens ein umfassendes Benutzerhandbuch.
  - *Beschreibung der Anwendungsarchitektur:* Eine detaillierte Erläuterung der Anwendungsarchitektur ist erforderlich, einschließlich einer kurzen Darstellung der implementierten Klassen und Skripte.
  - *Installationsanleitung:* Die Dokumentation sollte eine klare Anleitung zur Installation der Anwendungen beinhalten.
3. **Präsentation der Ergebnisse mittels Screencasts:** Die Präsentation sollte einen detaillierten Einblick in die Funktionsweise der Anwendung bieten. Durch die Demonstration der laufenden Anwendung im Screencast, gegebenenfalls kommentiert<sup>1</sup>, wird eine anschauliche Vorstellung der entwickelten Lösung ermöglicht.

---

<sup>1</sup> Aufgrund der unzureichenden Mikrofonqualität wurde der Screencast mithilfe einer Text-to-Speech-KI kommentiert.



## 2.0 Projektplanung

In Anbetracht der Dynamik unseres Projektverlaufs möchten wir einen Einblick in unsere Herangehensweise an die Projektplanung geben. Von Anfang an war klar, dass wir durch **praxisorientiertes Lernen** wertvolle Erkenntnisse gewinnen würden. Dennoch haben wir uns zu Beginn nicht gescheut, uns intensiv zu beraten und eine initiale Planung zu erstellen. Diese Überlegungen am Anfang bildeten das Fundament für unseren weiteren Vorgehensansatz. Zusätzlich möchten wir betonen, dass das **GitHub Student Pack**<sup>1</sup> eine maßgebliche Rolle spielte, indem es uns mit wichtigen Werkzeugen ausstattete und so unsere Arbeit effizient unterstützte.

## 2.1 Kommunikation

Für effektive Kommunikation und Zusammenarbeit haben wir **Discord** als zentrale Plattform genutzt. Hierbei trafen wir uns nicht nur virtuell, sondern auch persönlich, um unseren Fortschritt zu besprechen, **Ideen auszutauschen** und gemeinsam am Projekt zu arbeiten. Dieser hybride Ansatz ermöglichte es uns, den aktuellen Stand des Projekts zu klären, Informationen zu dokumentieren und Dateien, einschließlich Bilder, effizient auszutauschen. Diese vielseitige Kommunikationsstrategie trug wesentlich zu einem reibungslosen Arbeitsprozess und einem effizienten Informationsaustausch im Team bei.

## 2.2 GitHub

Für eine effiziente Zusammenarbeit nutzten wir die Plattform **GitHub** als zentrales Werkzeug für die Verwaltung unseres Quellcodes. Zwar beschränkten wir uns zu Beginn auf einen einzigen „Branch“, da uns die genaue Funktionsweise von GitHub noch nicht vollständig vertraut war, jedoch erwies sich diese vereinfachte Struktur als ausreichend für unsere Bedürfnisse. Während des gesamten Projekts haben wir kontinuierlich und konsequent unsere Änderungen in Form von „Commits“ festgehalten. Dieser Ansatz ermöglichte es jedem Teammitglied, stets auf dem neuesten Stand zu sein, indem es die jeweils aktuellen Codeänderungen auf seinen eigenen Arbeitsplatz übernehmen konnte. Diese transparente Vorgehensweise gewährleistete nicht nur einen reibungslosen Entwicklungsprozess, sondern erleichterte auch die Nachverfolgung und Behebung von etwaigen Konflikten im Code.

## 2.3 Aufgabenverteilung

In unserem Team bestehend aus Cuma und Emin, war ursprünglich auch Fabian geplant, der die Rolle des Fullstack-Entwicklers übernehmen sollte. Leider mussten wir uns von Fabian trennen, da er seinen Verpflichtungen nicht nachkommen konnte. Infolgedessen haben Cuma und ich, Emin, improvisiert und die Verantwortung für die Fullstack-Entwicklung auf Cuma übertragen. Cuma hat dann eigenständig an der Programmierung, der Dokumentation und dem Screencast gearbeitet, während auch ich an der Dokumentation und dem Screencast mitgewirkt habe.

---

<sup>1</sup> <https://education.github.com/pack>



### 2.4 Code Richtlinien

Für eine konsistente und gut **lesbare Codebasis** haben wir klare Richtlinien und Stilvorgaben definiert. Zusätzlich haben wir eigene **Recherchen**<sup>1</sup> durchgeführt, um zeitgemäße Best Practices zu berücksichtigen. Während wir uns größtenteils daran gehalten haben, müssen wir ehrlicherweise zugeben, dass gegen Ende nicht immer alle Regeln eingehalten wurden. Einige der definierten Regeln umfassen:

- Verwendung von „camelCase“ für Variablennamen.
- Vermeidung von Variablennamen mit einem einzelnen Buchstaben wie Beispielsweise „X“.
- Benennung von Funktionen entsprechend ihrer Aufgabe, um deren Zweck klar zu kommunizieren.
- Verfassen von Code in einer selbsterklärenden Art und Weise um seine Funktionalität ohne zusätzliche Erklärungen verständlich zu machen.
- Kommentierung nur dann, wenn der Code nicht intuitiv verständlich ist, um eine klare Dokumentation sicherzustellen.

### 2.5 Entwicklungsumgebung

Für die Entwicklung unserer Anwendung haben wir die IntelliJ IDEA als unsere **Hauptentwicklungsumgebung** verwendet. Diese leistungsfähige IDE wurde uns kostenlos durch das GitHub Student Pack zur Verfügung gestellt. Die Entscheidung für IntelliJ basierte nicht nur auf der Kostenfreiheit, sondern auch auf den umfassenden Funktionen, die sie bietet. Insbesondere die nahtlose Integration von Datenbanken erleichterte die Entwicklung und den Umgang mit der Datenbank erheblich.

---

<sup>1</sup> Z.b. <https://www.youtube.com/watch?v=-J3wNP6u5YU>





## 3.0 Verwendete Technologien und Planung

Im Rahmen unseres Projekts haben wir eine Vielzahl von Technologien eingesetzt, um eine robuste und effiziente Lösung zu entwickeln. Die Auswahl der richtigen **Werkzeuge** spielte eine entscheidende Rolle in der Gestaltung und Umsetzung unseres Projekts. Im folgenden Abschnitt bieten wir einen detaillierten Einblick in die von uns verwendeten Technologien, Bibliotheken und Sprachen, die maßgeblich zum Erfolg unserer Anwendung beigetragen haben.

### 3.1 Anwendungen

Alle von uns genutzten Anwendungen sind kostenfrei erhältlich oder Teil des kostenlosen **GitHub Student Developer Packs**.

Anwendung	Version	Grund	Link
<b>IntelliJ Ultimate</b> <sup>1</sup>	2023.3	Entwicklungsumgebung	<a href="#">Link</a>
<b>Bootstrap Studio</b> <sup>1</sup>	5.3	Frontend Design mit Bootstrap	<a href="#">Link</a>
<b>Microsoft Azure MySQL Server</b> <sup>1</sup>	8.0	Cloud Hosting der MySQL Datenbank	<a href="#">Link</a>
<b>MySQL Workbench</b>	8.0.33	Modellierung und Zugriff auf die Datenbank	<a href="#">Link</a>
<b>Java SDK 17 (Temurin)</b>	17.0.10	Java-Entwicklungskit	<a href="#">Link</a>
<b>Google Chrome</b>	N/A	Dev-Tools zum Debuggen und Testen	<a href="#">Link</a>
<b>Microsoft Word</b>	2401	Dokumentation	<a href="#">Link</a>

#### 3.1.1 Bootstrap Studio

Wir haben Bootstrap Studio als Teil unserer Entwicklungswerkzeuge verwendet, da es eine benutzerfreundliche Plattform bietet, die es uns ermöglicht, schnell und einfach responsive Webdesigns zu erstellen. Die integrierte Vorschau-Funktion ermöglichte es uns, Änderungen in Echtzeit zu sehen und das Aussehen unserer Anwendung zu optimieren. Darüber hinaus bot uns Bootstrap Studio die Möglichkeit, unsere Fähigkeiten im Umgang mit dem Bootstrap-Framework zu verbessern, indem es uns eine Vielzahl von vorgefertigten Komponenten und Vorlagen zur Verfügung stellte. Die umfangreiche Dokumentation unterstützte uns dabei, neue Funktionen zu erlernen und Probleme effizient zu lösen. Insgesamt war Bootstrap Studio eine wertvolle Ressource, die es uns ermöglichte, hochwertige und benutzerfreundliche Frontend-Designs für unsere Anwendung zu erstellen.

#### 3.1.2 Microsoft Azure

Für die Datenbankinfrastruktur unserer Anwendung haben wir Microsoft Azure als Hosting Plattform für unseren MySQL-Server genutzt. Diese Entscheidung wurde durch das großzügige Angebot des Microsoft Azure Studentenguthabens beeinflusst, das uns einen Startbetrag von 100 US-Dollar gewährte. Durch die Nutzung von Microsoft Azure konnten wir unsere MySQL-Datenbank sicher und effizient hosten, und das Studentenguthaben ermöglichte uns, die Hosting-Kosten im Rahmen zu halten. Die Integration von Microsoft Azure bot uns eine zuverlässige Cloud-Umgebung für unsere Datenbank, wodurch eine reibungslose Funktionalität und Skalierbarkeit unserer Anwendung gewährleistet wurde.

## 3.2 Sprachen

---

<sup>1</sup> Teil des GitHub Student Developer Packs



In diesem Abschnitt beleuchten wir die Programmiersprachen, die in unserem Projekt eine zentrale Rolle spielen. Diese Sprachen bilden das Grundgerüst unserer Implementierung und tragen maßgeblich zur Umsetzung der definierten Anforderungen bei.

Sprache	Version	Verwendungszweck
Java	17	Hauptprogrammiersprache für das Backend
JavaScript	N/A	Verwendet für das Frontend und als Schnittstelle für das Backend (Java)
Java Server Pages (JSP)	3.1.1	Eingesetzt für das Frontend und als Schnittstelle für das Backend (Java)
HTML	HTML5	Verwendet im Frontend
CSS	CSS3	Verwendet im Frontend
XML	1.0	Dient als Konfiguration für Maven, Hibernate und andere Komponenten

### 3.3 Maven-Konfiguration und Bibliotheken

Für unser Projekt nutzen wir Maven als Werkzeug zur Strukturierung und Automatisierung des Build-Prozesses. Hier sind die Hauptabhängigkeiten, die in unserem Projekt verwendet werden.

Bibliothek/Plugin	Version	Verwendungszweck
Gson	2.10.1	JSON-Konvertierung
Jakarta Servlet API	6.0.0	Servlet Funktionalität
Jakarta Servlet JSP API	3.1.1	JavaServer Pages (JSP) Funktionalität
Jakarta Servlet JSTL API	3.0.0	JSP Standard Tag Library (JSTL)
Hibernate Core	6.4.0 Final	ORM-Framework für Java
Hibernate C3P0	6.4.0 Final	C3P0-Integration für Hibernate
MySQL Connector/J	8.2.0	MySQL JDBC Treiber
Eclipse Jetty Server	12.0.4	Webserver und Servlet-Container
JBCrypt	0.4	Bibliothek für Sicheres Verschlüsseln von Passwörtern

### 3.4 Verwendete JavaScript-Bibliotheken und Frameworks

Für die Frontend-Entwicklung haben wir verschiedene JavaScript-Bibliotheken eingesetzt, um eine reibungslose und ansprechende Benutzeroberfläche zu gestalten. Diese Bibliotheken bieten uns Funktionalitäten, die die Darstellung und Interaktion mit dem Frontend verbessern. Im Folgenden werden die spezifischen Bibliotheken und ihre Verwendungszwecke näher erläutert.

Bibliothek/Framework	Version	Verwendungszweck	Link
Js-Year-Calendar	V1.0	Visualisierung und Interaktion des Kalenders	<a href="#">Link</a>
Chart.js	V2.8.0	Erstellung von Graphen und Diagramme	<a href="#">Link</a>
Bootstrap	V5.3.2	Erstellung von responsiven Frontend-Design	<a href="#">Link</a>

#### 3.4.1 Js-Year-Calendar

Für die Visualisierung und Interaktion des Kalenders in unserer Anwendung haben wir die Bibliothek "js-year-calendar" in der Version 1.0 verwendet. Diese Bibliothek bietet zahlreiche Funktionen, darunter die Fähigkeit, das gesamte Jahr auf einen Blick darzustellen, die Möglichkeit, einen Bereich auszuwählen und somit mehrere Tage gleichzeitig zu markieren.



Die Responsivität der Bibliothek gewährleistet zudem eine optimale Anpassung an verschiedene Bildschirmgrößen. Die Entscheidung für "js-year-calendar" basierte auf der Vielseitigkeit und den benutzerfreundlichen Funktionen, die sie für die Darstellung und Interaktion mit dem Kalender in unserer Anwendung bereitstellt.

### 3.4.2 Chart.js

Für die Erstellung von Graphen und Diagrammen in unserer Anwendung haben wir das Framework "Chart.js" in der Version 2.8.0 integriert. "Chart.js" bietet eine benutzerfreundliche API für die einfache Erstellung verschiedener Arten von Diagrammen, einschließlich Linien-, Balken- und Kreisdiagrammen. Die umfassende Dokumentation und die breite Unterstützung für verschiedene Chart-Typen erleichtern die Integration von ansprechenden visuellen Elementen in unsere Anwendung. Durch die Verwendung von "Chart.js" konnten wir Daten effektiv visualisieren und benutzerfreundlich präsentieren.

### 3.4.3 Bootstrap

Für das Frontend-Design unserer Anwendung haben wir Bootstrap als zentrales Framework verwendet. Bootstrap ermöglichte es uns, das Benutzerinterface mit minimalem Aufwand zu gestalten, da es eine umfangreiche Sammlung von vorgefertigten Komponenten und Stilen bereitstellt. Die Entscheidung für Bootstrap basierte auf mehreren Überlegungen.

Erstens bietet Bootstrap eine einfache und intuitive Bedienung, was besonders wichtig war, da wir während des Projekts die Lernkurve so gering wie möglich halten wollten. Durch die Verwendung von Bootstrap konnten wir schnell responsive und ästhetisch ansprechende Benutzeroberflächen erstellen, ohne tiefgreifende Kenntnisse im Bereich Frontend-Design zu benötigen.

Zusätzlich haben wir von verschiedenen nützlichen Bootstrap-Komponenten profitiert, wie Modals, Alerts und DataTables, um interaktive Pop-up-Fenster, wichtige Benachrichtigungen und tabellarische Datenübersichten zu integrieren. Diese Elemente haben dazu beigetragen, eine ansprechende und funktionale Benutzeroberfläche zu realisieren.

## 3.5 Planung

Unsere Planung für das Projekt begann mit der Festlegung unserer grundlegendsten **Ziele**. Zu Beginn definierten wir die essenziellen Anforderungen und Strukturpunkte, die das Fundament für unsere Anwendung bilden sollten. Während dieses Prozesses entwickelten sich viele weitere Aspekte und Feinheiten organisch im Laufe der Zeit. Die flexible Ausrichtung unserer Planung ermöglichte es uns, auf neue Erkenntnisse und Herausforderungen während der Entwicklung angemessen zu reagieren. Diese dynamische Herangehensweise in unserer Planung spiegelte wider, dass die Feinabstimmung und Optimierung unserer Ziele im Verlauf des Projekts stattfanden und uns erlaubten, sich verändernde Anforderungen aufzugreifen.

### 3.5.1 Ziele

Neben den verpflichtenden Mindestanforderungen, wie sie im Abschnitt [Mindestanforderungen](#) beschrieben sind, haben wir zu Beginn des Projekts erste Schritte und Ziele festgelegt, um den Entwicklungsprozess zu strukturieren und voranzutreiben. Hier sind die Hauptziele in Stichpunkten:

- Festlegen der benötigten Frameworks und Bibliotheken
- Inspirieren lassen durch bestehende Designs
- Analyse bereits vorhandener Seiten zur Erfahrungssammlung
- Einrichten eines GitHub-Repositories für die Versionskontrolle



- Suche nach einer geeigneten Cloud-Datenbanklösung
- Erstellen eines grundlegenden Gerüsts für die Anwendung
- Erste Datenbank erstellen

### 3.5.2 Wunschziele

Unter "Wunschziele" fassen wir optionale Erweiterungen und Ziele zusammen, die während des Entwicklungsprozesses als mögliche Verbesserungen identifiziert wurden. Diese Ziele sind nicht verpflichtend, sondern bieten die Möglichkeit, den Funktionsumfang der Anwendung nach Bedarf zu erweitern.

- Registrierung
- Tabellarische Darstellung der Benutzer
- Dunkles Erscheinungsbild (Dark Mode)
- Option für automatische Anmeldung
- Sicherheitsmaßnahmen
- Optimierung der Anwendungsleistung
- Mitteilungen für Benutzer und Administratoren
- Aktualisierbare und Löschbare Einträge
- Anzeige von Kalenderdaten über mehrere Jahre
- Durchdachte Anwendungslogik
- Möglichkeit für Benutzer, Profilbilder hochzuladen
- Echtzeitinformation über die Anwesenheit von Benutzern
- Adminzugriff auf Benutzerkalender
- Überblick über Arbeitsstundenentwicklung
- Funktion zur Suche nach Mitarbeitern



### 4.0 Benutzeroberfläche und Webseite-Features

Im folgenden Abschnitt 4.0 werden die Benutzeroberfläche und die herausragenden Features der Webseite detailliert beleuchtet. Dieser Überblick konzentriert sich auf die visuelle Gestaltung sowie die Funktionalitäten der Plattform, ohne dabei auf den zugrundeliegenden Programmcode einzugehen. Für weitere technische Einblicke und Implementierungsdetails verweisen wir auf Abschnitt 6.0 Quellcode Übersicht.

Wir haben für unsere Webseite mehrere Seiten entwickelt, von denen jede spezifische Funktion und Aufgaben erfüllt. Hier sind die wichtigsten Seiten und ihre jeweiligen Aufgaben:

- **Dashboard:** Hauptseite mit benutzerdefinierten Einträgen, Statistiken und mehr.
- **Login:** Anmeldeseite für bereits registrierte Benutzer.
- **Register:** Seite zur Erstellung eines neuen Benutzerkontos.
- **Managment:** Verwaltungsseite für die Überwachung und Übersicht der Mitarbeiter sowie ihrer Stundenkonten.
- **Calendar:** Kalenderseite zum Einsehen, Bearbeiten und Hinzufügen von vorgefertigten Anwesenheitsstatus.
- **Profile:** Profilseite zum Ändern von E-Mail, Passwort und Profilbild.
- **Messages:** Mitteilungsseite zum Einsicht nehmen aller Benachrichtigungen der letzten 7 Tage.
- **Impressum:** Seite mit Impressumsinformationen.
- **Datenschutz:** Seite mit Datenschutzinformationen.
- **404:** Fehlerseite für nicht gefundene Ressourcen.

Diese Struktur gewährleistet eine klare und effiziente Navigation für die Nutzer, um ihre gewünschten Funktionen schnell zu erreichen.

### 4.1 Anwendungsmöglichkeiten

Nach Rücksprache<sup>1</sup> mit **Frau Dr. Kerstin Kipper** wird hervorgehoben, dass unsere Zielgruppe Unternehmen sind. Die Anwendung wurde gezielt konzipiert, um eine breite Palette von Firmenbedürfnissen zu erfüllen. Besonderes Augenmerk legten wir dabei auf die Flexibilität der Anwendung, um unterschiedlichen Arbeitszeitmodellen gerecht zu werden. Die Idee, die Anwendung flexibel zu gestalten, war ein zentraler Bestandteil unserer Entwicklungsphilosophie. Hierbei bieten wir zwei Hauptoptionen an:

#### 1. Festgelegte Arbeitszeiten:

- Für Unternehmen mit standardisierten Arbeitszeiten, beispielsweise 8 Stunden Arbeit und 1 Stunde Pause, bietet unsere Anwendung eine passende Lösung.

#### 2. Unabhängige Arbeitszeiten:

---

<sup>1</sup> E-Mail-Kommunikation



- Unternehmen mit flexiblen Arbeitszeitmodellen, die sich beispielsweise an Stundenkonten orientieren, finden in unserer Anwendung die erforderliche Anpassungsfähigkeit.

### 4.1.1 Gerätekompatibilität

Bei der Erstellung unserer Webseite lag unser Fokus darauf, eine optimale Funktionalität auf allen mobilen Geräten, **einschließlich Tablets und Smartphones**, sicherzustellen. Die Nutzung von Bootstrap als responsivem Framework ermöglichte eine konsistente Darstellung und Benutzererfahrung über verschiedene Bildschirmgrößen hinweg. Jede Seite wurde sorgfältig angepasst, um eine reibungslose Nutzung der Webseite auf einer Vielzahl von Geräten zu gewährleisten. Diese Anpassungen stellen sicher, dass die Webseite benutzerfreundlich und funktionsreich ist, unabhängig vom verwendeten mobilen Gerät. Im Verlauf der Dokumentation werden wir konkrete Beispiele für die Anpassungen und ihre Auswirkungen durch Bilder veranschaulichen.

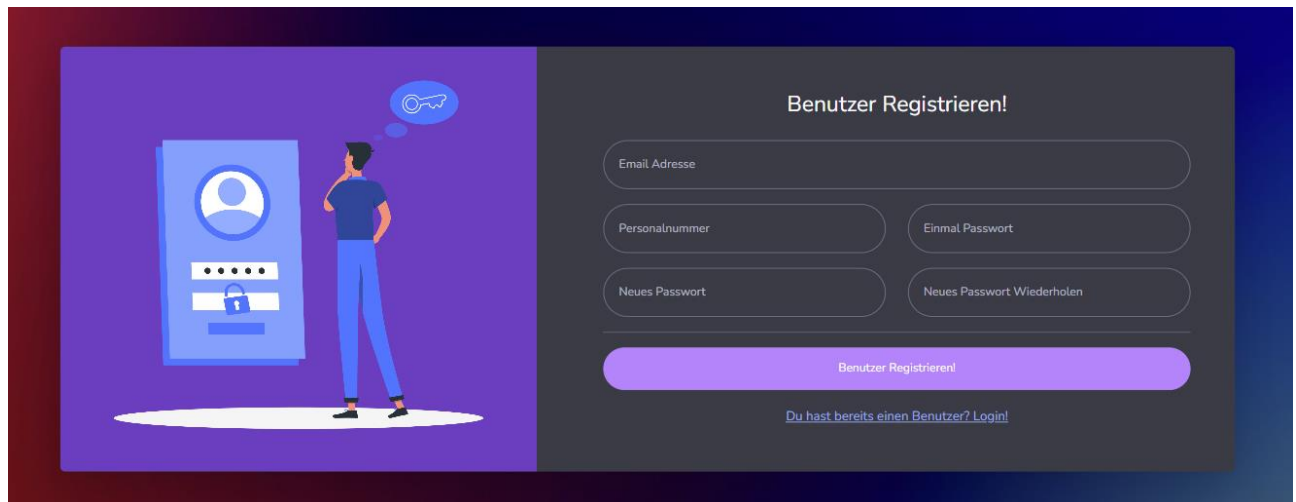
## 4.2 Registrierung

Die Registrierungsseite unserer Anwendung bietet eine klare und benutzerfreundliche Oberfläche. Nutzer haben hier die Möglichkeit, sich für die Nutzung der Plattform zu



registrieren. Es ist jedoch wichtig zu betonen, dass der Registrierungsprozess eine besondere Vorgehensweise erfordert.

So sieht unsere Seite aus:



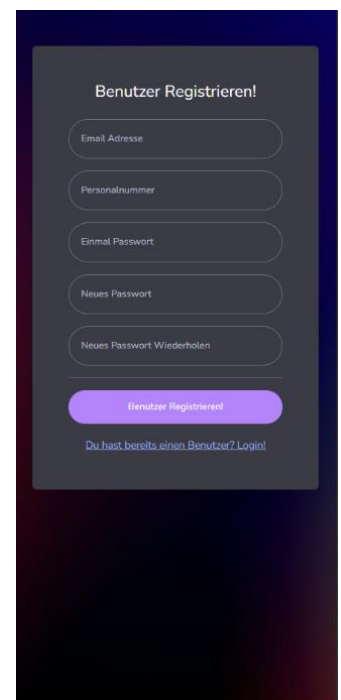
Desktop | Chrome

## 4.2.1 Registrierungsprozess:

Um sich zu registrieren, muss ein Benutzer von einem Administrator in der Datenbank eingetragen werden. Der Administrator weist dem Mitarbeiter eine Personalnummer zu und erstellt ein einmaliges Passwort (One-Time-Password). Nach dieser Zuweisung gibt der Benutzer seine persönliche E-Mail-Adresse und ein individuelles Passwort ein, um den Registrierungsprozess abzuschließen. Dieser spezielle Prozess gewährleistet nicht nur die Sicherheit der Anmeldeinformationen, sondern ermöglicht es dem Administrator auch, die Benutzerkonten im Detail zu verwalten.

## 4.2.2 Passwortvergessen-Funktionalität:

Sollte ein registrierter Benutzer seine Anmeldeinformationen vergessen, einschließlich der E-Mail-Adresse, hat der Administrator die Möglichkeit, ein neues Einmalpasswort zu generieren. Dieser Schutzmechanismus ermöglicht es dem Benutzer, seinen Account reibungslos wiederherzustellen, ohne dabei wichtige Daten wie sein Stundenkonto oder Kalenderdaten zu verlieren. Es ist jedoch zu beachten, dass im Falle des Vergessens der E-Mail-Adresse die erneute Registrierung erforderlich ist, um einen nahtlosen Zugang zur Plattform zu gewährleisten.

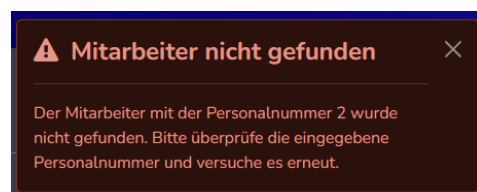


iPhone 14 Pro Max | Chrome

## 4.2.3 Meldungen

Auf der Registrierungsseite sind verschiedene Statusmeldungen implementiert, um den Benutzer über den Ausgang seiner Aktionen zu informieren:

1. Erfolgreiche Registrierung:





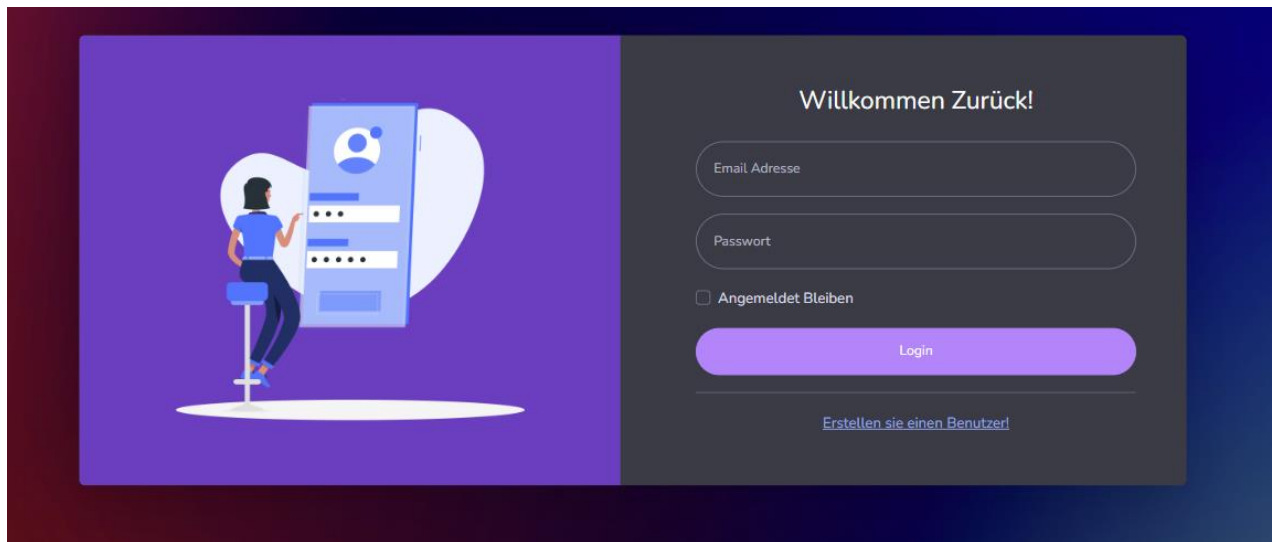
- Nach erfolgreicher Registrierung wird dem Benutzer eine positive Rückmeldung angezeigt.
2. Falsches One-Time-Password (OTP):
    - Wenn das eingegebene einmalige Passwort nicht korrekt ist, erhält der Benutzer eine Meldung über das falsche OTP.
  3. Passwort und Wiederholung stimmen nicht überein:
    - Bei Unstimmigkeiten zwischen dem eingegebenen Passwort und der Wiederholung wird dem Benutzer eine entsprechende Fehlermeldung angezeigt.
  4. Mitarbeiter nicht gefunden:
    - Falls der Mitarbeiter nicht in der Datenbank gefunden wird, erhält der Benutzer eine Benachrichtigung darüber.
  5. Gültiger Account vorhanden:
    - Wenn bereits ein gültiger Account vorhanden ist, wird der Benutzer darüber informiert.

Diese Statusmeldungen dienen dazu, dem Benutzer klare Rückmeldungen über den Verlauf seiner Registrierungsaktionen auf der Webseite zu geben.

### 4.3 Anmeldung

So sieht unsere Anmeldeseite aus:



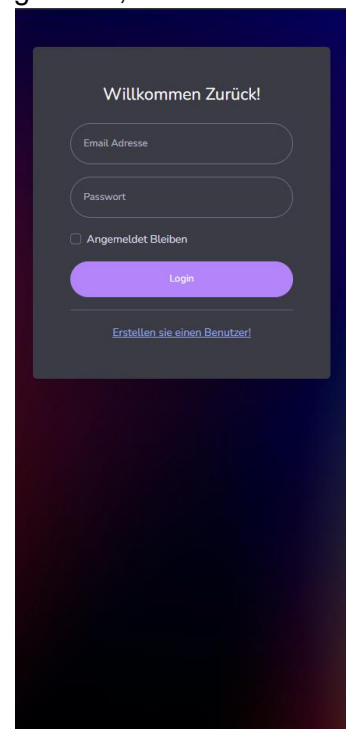


Desktop | Chrome

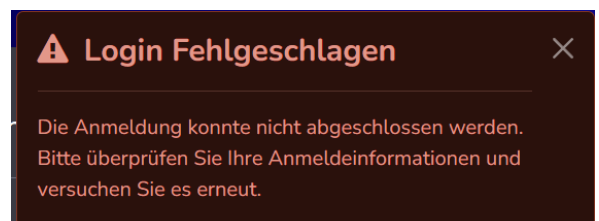
Die Anmeldeseite ist die erste Seite, auf die Benutzer zugreifen, wenn sie nicht die Option "Angemeldet bleiben" aktiviert haben. Hier haben Benutzer auch die Möglichkeit, auf den Link "Erstellen Sie einen Benutzer!" zu klicken, um zur Registrierungsseite zu gelangen, falls sie noch keinen Benutzer haben.

Um sich anzumelden, geben Benutzer einfach die bei der Registrierung festgelegte E-Mail und ihr Passwort ein. Optional besteht die Möglichkeit, das Kästchen "Angemeldet bleiben" anzuklicken, um automatisch 14 Tage lang angemeldet zu bleiben. Beachten Sie dabei, dass für diese Funktion Cookies aktiviert sein müssen. Diese bequeme Option sorgt dafür, dass Benutzer nicht jedes Mal ihre Anmeldeinformationen eingeben müssen.

Im Falle von falschen Anmeldeinformationen, sei es ein falsches Passwort oder eine ungültige E-Mail, wird dem Benutzer eine klare Fehlermeldung angezeigt.



iPhone 14 Pro Max | Chrome



## 4.4 Seitenleiste und Navigationsleiste

Die Seitenleiste und Navigationsleiste sind zentrale Elemente unserer Webseite, die auf jeder Seite vorhanden sind, um eine effiziente Navigation zu gewährleisten. Im folgenden



Abschnitt werden wir detailliert erläutern, welche Funktionen und Optionen diese Elemente bieten, um den Benutzern eine klare Orientierung und einfache Bedienbarkeit zu ermöglichen.

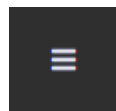
### 4.4.1 Seitenleiste

Die Seitenleiste stellt eine Vielzahl von Funktionen bereit, um die Navigation auf unserer Webseite effizient und benutzerfreundlich zu gestalten. So sieht die Seitenleiste aus:

1. **Ein- und Ausklappen:** Die Seitenleiste kann je nach Bedarf ein- oder ausgeklappt werden und ermöglicht so eine flexible Navigation.
2. **Seitenübersicht:** Die verschiedenen Seiten der Webseite sind aufgelistet, wobei die aktuelle Seite durch Hervorhebung deutlich erkennbar ist.
3. **Ausloggen:** Benutzer haben die Möglichkeit, sich über die Seitenleiste aus ihrem Konto auszuloggen.
4. **Dark Mode und Light Mode:** Die Seitenleiste bietet eine Schaltfläche, mit der Benutzer zwischen dem dunklen und hellen Modus umschalten können, um ihre bevorzugte Anzeigeeoption auszuwählen.

Es ist zu beachten, dass der Reiter Mitarbeiter nur für Administratoren angezeigt wird.

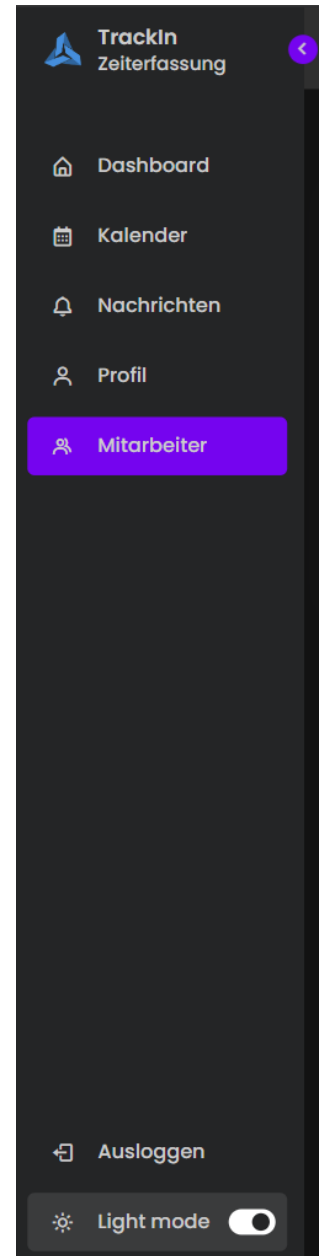
Auf mobilen Geräten wird der eingeklappte Modus der Seitenleiste standardmäßig ausgeblendet. Stattdessen steht den Benutzern oben in der Navigationsleiste ein Burger-Menü-Icon mit drei horizontalen Strichen zur Verfügung. Durch Anklicken dieses Symbols wird die Seitenleiste ausgefahren, sodass die Benutzer auch auf kleineren Bildschirmen bequem zwischen den verschiedenen Seiten navigieren können.



iPhone 14 Pro Max | Chrome |  
Burger-Menü-Icon



Desktop | eingeklappt



Desktop | Chrome | ausgeklappt



## 4.4.2 Navigationsleiste

So sieht die Navigationsleiste unserer Webseite aus, wobei das Erscheinungsbild je nach Bildschirmgröße variiert.



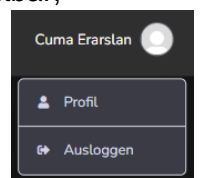
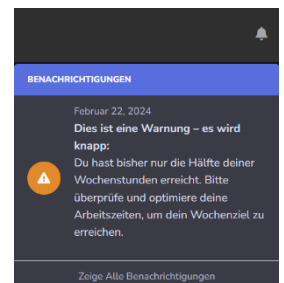
Desktop | Chrome



iPhone 14 Pro Max | Chrome

Die Funktionen der Navigationsleiste kann in 3 Bereiche zusammengefasst werden:

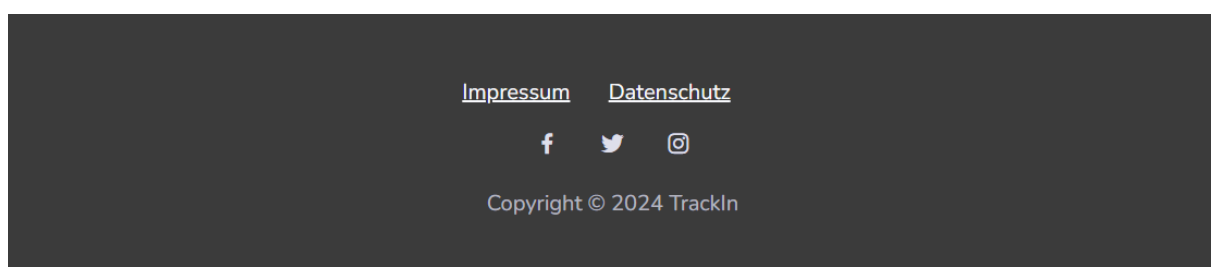
1. **Burger-Menü-Icon:** Auf kleinen Bildschirmen sichtbar und ermöglicht das Einklappen der Seitenleiste.
2. **Glocken-Symbol:** Das Glocken-Symbol zeigt die **Anzahl** der ungelesenen Benachrichtigungen durch eine rot umrandete Zahl an. Bei einem Klick auf das Symbol öffnet sich ein Dropdown-Menü.
  - **Dropdown-Menü:** das Dropdown-Menü zeigt die letzten **drei** Benachrichtigungen an. Durch die Schaltfläche "Alle Benachrichtigungen anzeigen" kann der Benutzer zur Nachrichtenseite gelangen, auf der alle Benachrichtigungen angezeigt werden.
3. **Benutzerbereich:** Nur Auf größeren Bildschirmen ist der vollständige Name sichtbar, enthält den vollen Namen des Benutzers und sein Profilbild.
  - **Dropdown-Menü:** Öffnet sich beim Klicken auf den Benutzerbereich und bietet Optionen wie auf die Profilseite wechseln und Ausloggen an.



## 4.4.3 Fußbereich

Der Footer, auch als Fußbereich bezeichnet, ist ein Abschnitt am unteren Rand unserer Webseite. Er enthält folgendes:

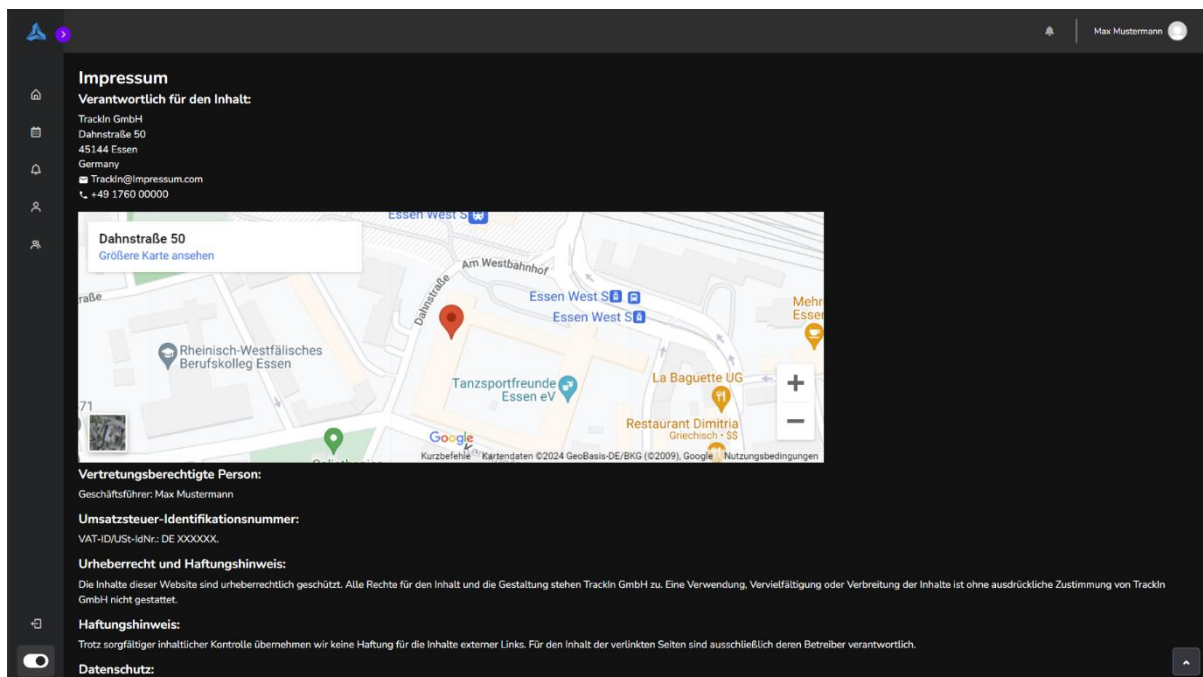
1. **Impressum:** Ein Link, der die Nutzer zu unserer Impressumsseite führt.
2. **Datenschutz:** Ein Link, der die Nutzer zur Datenschutzseite leitet.
3. **Copyright:** Das Copyright-Jahr und der Verweis auf die Urheberrechte.
4. **Soziale Medien:** Symbole für Twitter, Facebook und Instagram.





## 4.4.4 Impressum

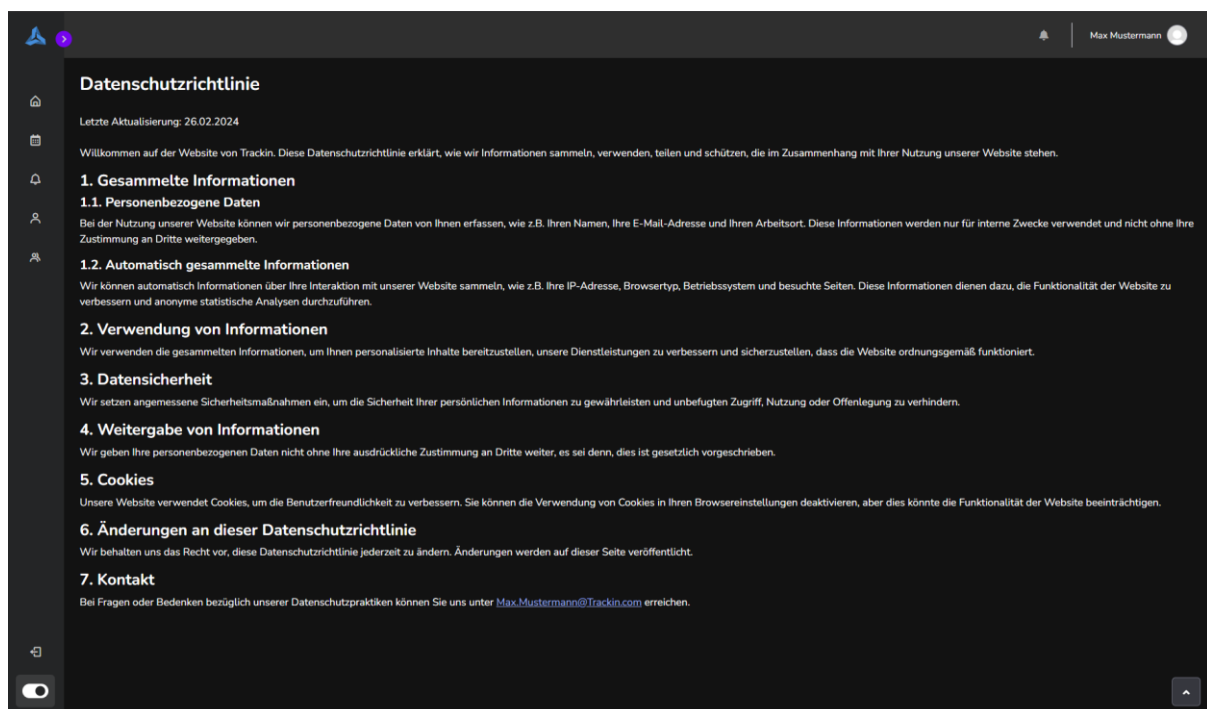
Die folgenden Informationen dienen der transparenten Darstellung der **rechtlichen** Rahmenbedingungen unserer Website. So sieht unser Impressum aus:



Hier finden Sie Angaben zu Verantwortlichkeiten, Urheberrecht, Haftungshinweisen, Datenschutz und weiteren relevanten Aspekten im Rahmen unseres Impressums. Zusätzlich haben wir Google Maps in Form eines „iframes“ in unsere Website eingebunden, um Ihnen eine interaktive Standortanzeige zu ermöglichen.

## 4.4.5 Datenschutz

Hier geben wir Ihnen einen Überblick über die Datenschutzpraktiken von TrackIn. So sehen unsere Datenschutzrichtlinien aus:





### 4.4.6 Error 404-Seite

Für den Fall, dass Ressourcen nicht gefunden werden, haben wir eine spezielle 404-Seite erstellt.

Diese Seite ist nicht nur informativ, sondern auch animiert, um eine ansprechende Benutzererfahrung zu bieten. Sollten Sie auf den Link zum Dashboard klicken, werden Sie direkt zum Dashboard weitergeleitet, um nahtlos mit Ihrer Navigation fortzufahren.

# 404

Seite nicht Gefunden!

It looks like you found a glitch in the matrix...  
[zum Dashboard](#)



## 4.5 Hauptseite

Die **Hauptseite** unserer Webseite, auch als **Dashboard** bezeichnet, bietet eine zentrale Anlaufstelle für Benutzer, um verschiedene Funktionen zu nutzen und Informationen einzusehen. Hierzu gehören unter anderem die Einsicht in das persönliche Stundenkonto, die Betrachtung von statistischen Auswertungen in Form von Diagrammen sowie die Möglichkeit zur Erstellung und Löschung von Arbeitszeiten. Das Dashboard wurde entwickelt, um den Nutzern einen übersichtlichen Zugriff auf wichtige Daten und Funktionen zu ermöglichen, um ihre Arbeitszeiten effektiv zu verwalten und relevante Informationen schnell abzurufen. So sieht unsere Hauptseite aus:

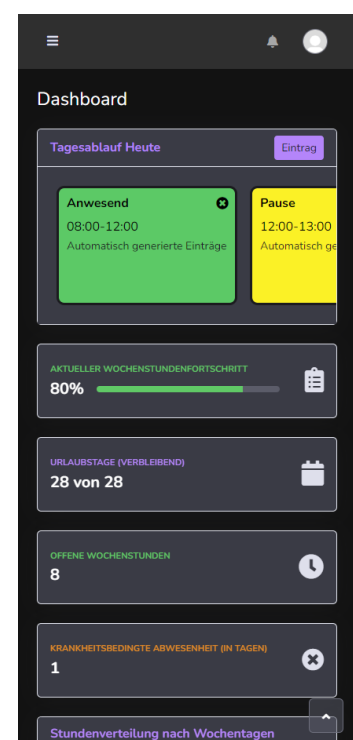


Desktop | Chrome

### 4.5.1 Statistiken

Die Statistik-Funktion auf unserer Webseite bietet eine detaillierte Übersicht über verschiedene relevante Kennzahlen. Diese Informationen werden durch die Daten im Kalender sowie die Einträge im Dashboard berechnet. Hierbei werden vier verschiedene Statistikfelder präsentiert, die Einblicke in den aktuellen Arbeitsstand und weitere wichtige Aspekte ermöglichen.

1. **Aktueller Wochenstunden Fortschritt:** Dieses Feld zeigt eine Prozentzahl und einen Balken, um den aktuellen Wochenfortschritt der im wöchentlichen Stundenkonto abgearbeiteten Stunden zu visualisieren
2. **Verbleibende Urlaubstage:** Die verbleibenden Urlaubstage werden durch die im Kalender eingetragenen Urlaubstage berechnet und zeigen die noch verfügbaren Urlaubstage an.



iPhone 14 Pro Max | Chrome



3. **Offene Wochenstunden:** Hier wird angezeigt, wie viele Stunden noch benötigt werden, um das Stundensoll zu erfüllen. Diese Information bietet eine Orientierung für die noch zu erbringende Arbeitszeit.
4. **Krankheitsbedingte Abwesenheit:** Dieses Feld gibt an, wie viele Tage dieses Jahr der Mitarbeiter aufgrund von Krankheit nicht anwesend war. Dabei wird ein normaler Arbeitstag mit 8 Stunden als ein Tag gewertet.

Diese Statistikfelder dienen dazu, den Benutzern eine klare Übersicht über ihren aktuellen Arbeitsstand und relevante Kennzahlen zu geben, um ihre Arbeitszeiten effektiv zu planen und zu verwalten.

### 4.5.2 Diagramme

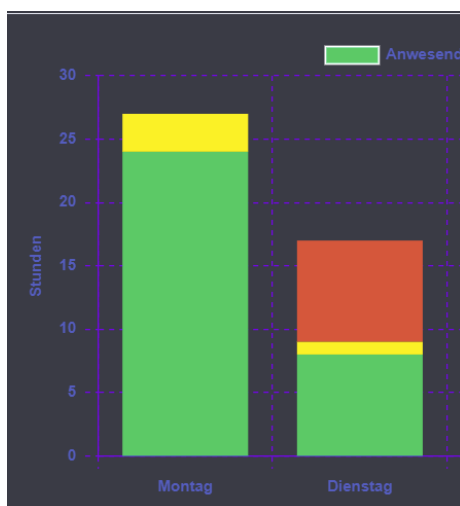
Die Diagramm-Funktion auf unserer Webseite bietet eine visuelle Darstellung der Arbeitszeiteinträge des gesamten Jahres und ermöglicht eine leicht verständliche Analyse. Wir haben zwei Arten von Diagrammen integriert, um unterschiedliche Aspekte der Arbeitszeit effektiv zu visualisieren.

#### 1. Balken-Diagramm:

- Das Balken-Diagramm teilt die Einträge von Pausen, Krankmeldungen, Anwesenheiten und Abwesenheiten auf die einzelnen Wochentage auf.
- Durch einen Klick in der Legende auf den jeweiligen Text kann der Benutzer die Daten einer bestimmten Art aus- oder einblenden und dabei das Diagramm erneut skalieren.
- Beim Hovern über die Abschnitte werden die entsprechenden Stunden in einem Pop-Over angezeigt.

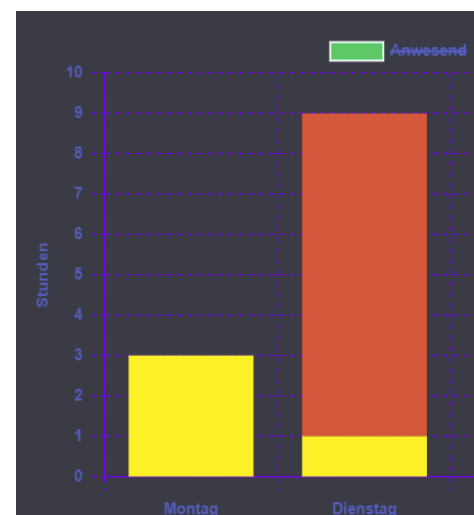
Beispiel:

Mit den „Anwesenheits-Daten“



Desktop | Chrome

Ohne die „Anwesenheits-Daten“

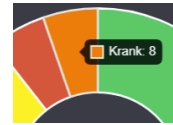


Desktop | Chrome

#### 2. Donut-Diagramm:



- Das Donut-Diagramm fasst die zuvor genannten vier Arten von Einträgen zusammen und visualisiert sie auf eine übersichtliche Weise.
- Beim Hovern über die Abschnitte werden die entsprechenden Stunden in einem Pop-Over angezeigt.
- Es bietet einen schnellen Überblick über die Verteilung der verschiedenen Arten von Arbeitszeiteinträgen.



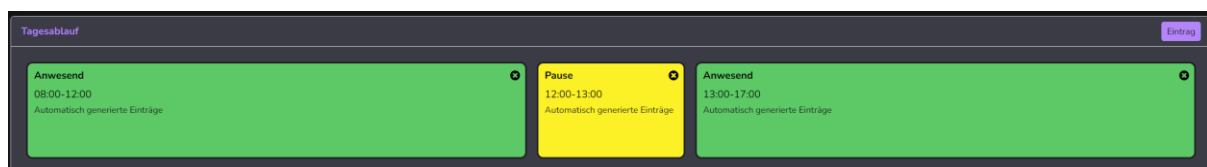
Diese Diagramme dienen dazu, die Arbeitszeitdaten in einer leicht verständlichen Form zu präsentieren, sodass die Benutzer schnell Muster und Trends erkennen können. Die Interaktivität in Form von Ein- und Ausblenden von Daten sowie die detaillierte Anzeige beim Hovern, bieten eine zusätzliche Ebene der Benutzerfreundlichkeit.

### 4.5.3 Benutzerdefinierte Einträge

Kommen wir zu einer der Hauptfunktionen, das Feld Tagesablauf ermöglicht es dem Nutzer, seine Arbeitszeiteinträge individuell anzupassen und zu verwalten. Im Feld werden die heutigen Einträge visuell dargestellt, einschließlich der Zeitspanne, des Status (z. B. Anwesend, Pause) und einer Beschreibung.

Neben den vorgefertigten Einträgen, die im Kalender erstellt werden, können Benutzer auch im Dashboard benutzerdefinierte Einträge erstellen und löschen. Es sei darauf hingewiesen, dass die vorgefertigten Einträge aus dem Kalender automatisch generierte Einträge in der Beschreibung enthalten. Benutzer haben auch die Möglichkeit, diese vorgefertigten Einträge im Dashboard anzuzeigen, zu überschreiben oder zu löschen, wenn dies erforderlich ist. Diese Flexibilität bietet den Benutzern eine effiziente Möglichkeit, ihre Arbeitszeiteinträge genau nach ihren Bedürfnissen anzupassen und zu verwalten.

Die Einträge werden chronologisch von links nach rechts angezeigt. Auf kleinen Bildschirmgrößen können Benutzer horizontal scrollen, um alle Einträge vollständig anzuzeigen.



Desktop | Chrome

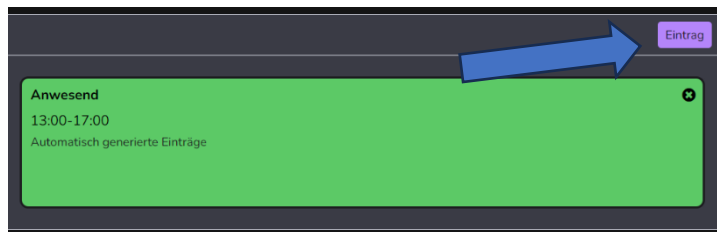
Zum Löschen eines Eintrags drückt der Benutzer einfach auf das X-Symbol im jeweiligen Eintragsfeld.







Zum Hinzufügen eines Eintrags drückt der Benutzer auf den "Eintrag" Knopf, um das entsprechende Fenster zu öffnen.

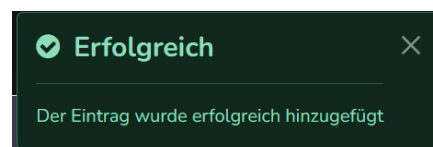


Der Benutzer hat nun die Möglichkeit, einen neuen Eintrag zu erstellen, wobei er die folgenden Felder ausfüllen kann:

1. **Status:** Der Benutzer kann den Status des Eintrags auswählen, wobei vier Optionen zur Verfügung stehen: Anwesend, Abwesend, Pause und Krank.
2. **Uhrzeit:** Es gibt zwei Uhrzeitfelder, die die Eingabe nur in 5-Minuten-Schritten zulassen. Beispielsweise ist die Eingabe von 13:03 nicht möglich, aber 13:05 ist erlaubt.
3. **Notizen:** Der Benutzer kann bei Bedarf Notizen zu dem erstellten Eintrag hinzufügen.

Nachdem der Mitarbeiter auf "Speichern" gedrückt hat, wird eine entsprechende Meldung angezeigt:

1. **Erfolgsmeldung:** Falls die Eingabe erfolgreich war, wird dem Benutzer eine Erfolgsmeldung angezeigt.
2. **Fehlermeldung - Sonntag:** Falls der Benutzer versucht, einen Eintrag für den Sonntag zu erstellen, wird eine Fehlermeldung angezeigt, da an diesem Tag keine Einträge möglich sind.
3. **Fehlermeldung - Überschneidung:** Wenn der Eintrag mit einem bereits vorhandenen Eintrag kollidiert, wird eine Fehlermeldung angezeigt, die auf die Überschneidung hinweist.





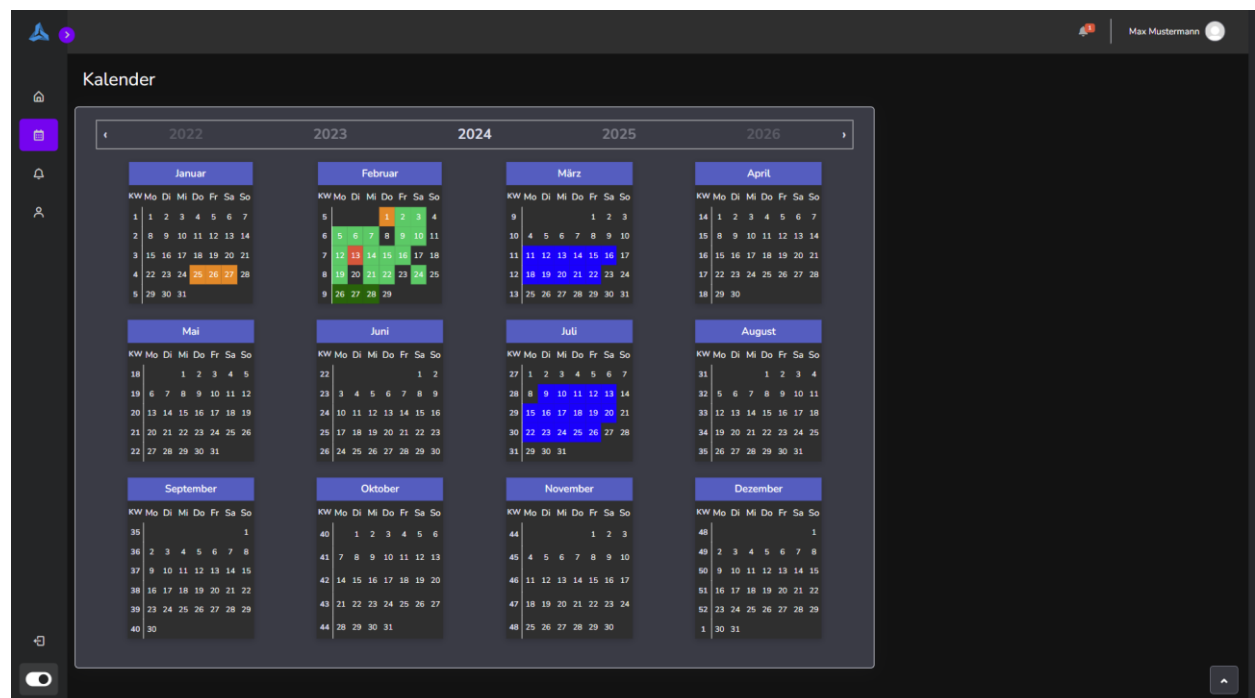
## 4.6 Kalender

Die Kalenderseite ist eine zentrale Funktion unserer Webseite, die es Benutzern ermöglicht, einzelne Tage oder mehrere Tage mit einem bestimmten Status zu versehen. Diese Statuszuweisungen generieren wiederum Einträge, die wichtige Informationen im Stundenkonto und im Tagesablauf festhalten. Die vordefinierten Tagesabläufe sind wie folgt:

1. **Anwesend:** 8 Stunden Anwesenheit, was zu einer Gutschrift im Stundenkonto führt. Eine Stunde Pause ist inbegriffen, jedoch nicht bezahlt.
2. **Urlaub:** Ein Tag Urlaub entspricht einer Gutschrift von 8 Stunden im Stundenkonto. Urlaubstage werden bezahlt.
3. **Krank:** Ein Krankheitstag entspricht einer Belastung von 8 Stunden im Stundenkonto. Krankheitstage werden bezahlt.
4. **Abwesend:** 8 Stunden Abwesenheit ohne Auswirkungen auf das Stundenkonto.
5. **Dienstreise:** 10 Stunden werden bezahlt.

Auf der Kalenderseite haben Benutzer die Möglichkeit, Tage zu bearbeiten oder zu löschen. Zusätzlich können sie auch in andere Jahre zurück- oder vorausblicken, um zukünftige Planungen wie Dienstreisen im Voraus zu berücksichtigen.

So sieht unsere Kalender Seite aus:



Desktop | Chrome

- Diese Farbgebung ermöglicht es Benutzern, auf einen Blick den Status eines bestimmten Tages im Kalender zu erkennen und vereinfacht so die schnelle Identifikation. Zusätzlich gibt es ein Pop Over für die eingetragenen Tage, das beim Hovern über das Element zeigt, welchen Status der Tag hat, wie viele Stunden man anwesend





war, wie viele Stunden krank, und die eingetragene Beschreibung. Dies ist dynamisch und berücksichtigt auch Benutzerdefinierte Einträge, die auf der Hauptseite erstellt wurden. Beachten Sie, dass Änderungen an den Einträgen auf der Hauptseite Auswirkungen auf den Status im Kalender haben können.

- Blau: Urlaub
- Hellgrün: Anwesend
- Dunkelgrün: Dienstreise
- Orange: Krank
- Rot: Abwesend

Diese Farbgebung ermöglicht es Benutzern, auf einen Blick den Status eines bestimmten Tages im Kalender zu erkennen und vereinfacht so die schnelle Identifikation.

## 4.6.1 Dropdown-Menü

Um die Benutzerfreundlichkeit zu erhöhen, haben wir ein Dropdown-Menü in die Kalenderseite integriert. Dieses Menü ermöglicht es Benutzern, durch Rechtsklicken auf einen bestimmten Tag oder durch längeres Drücken auf Touchscreens zusätzliche Optionen auszuwählen. Die beiden Hauptoptionen sind:

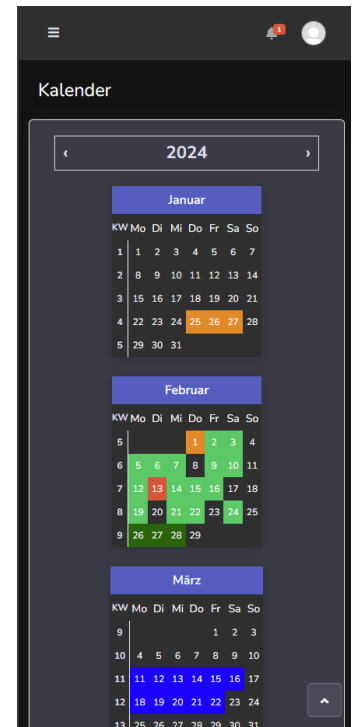
1. **Aktualisieren:** Durch Auswahl dieser Option öffnet sich ein Fenster, das es dem Benutzer ermöglicht, den ausgewählten Tag zu bearbeiten. Dies beinhaltet das Überschreiben von Informationen wie dem Status oder der Beschreibung des Tages. Beachten Sie dabei, dass beim Aktualisieren eine Überprüfung auf mögliche Fehler erfolgt, zum Beispiel, wenn zu wenige Urlaubstage verfügbar sind.
2. **Löschen:** Wenn der Benutzer einen Tag löscht, werden nicht nur die Tagesinformationen entfernt, sondern auch alle zugehörigen Einträge im Stundenkonto und im Tagesablauf gelöscht.

Dieses Dropdown-Menü bietet eine effiziente Möglichkeit, einzelne Tage schnell zu aktualisieren oder zu entfernen, ohne zusätzliche Schritte durchlaufen zu müssen.

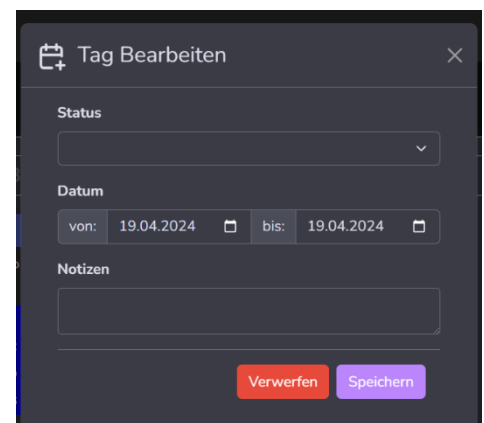
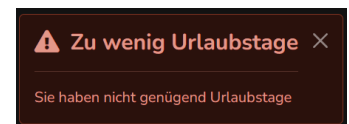
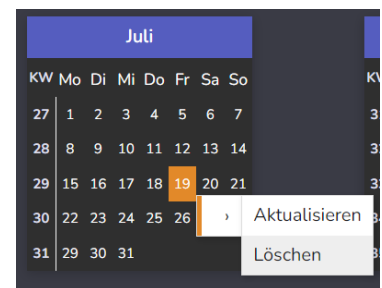
## 4.6.2 Statusvergabe

Die Statusvergabe auf der Kalenderseite erfolgt durch verschiedene Interaktionsmöglichkeiten, um die Benutzerfreundlichkeit zu gewährleisten.

1. **Einzelner Tag auswählen (Desktop und Mobile):** Benutzer können einen einzelnen Tag auswählen, indem sie daraufklicken oder antippen. Dadurch wird ein Bearbeitungsfenster geöffnet, das es dem Benutzer ermöglicht, den Status des ausgewählten Tages aus einer Liste von fünf Optionen zu wählen.



iPhone 14 Pro Max | Chrome





2. **Bereichsauswahl (Desktop):** Desktop-Benutzer können zusätzlich durch Linksklick beginnen, das Mauspad zu halten und über mehrere Tage zu ziehen, um einen Bereich auszuwählen. Auch hier öffnet sich ein Bearbeitungsfenster, in dem der Benutzer den Status für den ausgewählten Bereich festlegen kann.

**Bearbeitungsfenster öffnen:** Nach Auswahl eines einzelnen Tags oder eines Bereichs öffnet sich ein Bearbeitungsfenster. In diesem Fenster kann der Benutzer den Status für die ausgewählten Tage festlegen. Das Fenster zeigt das Datum des ausgewählten Tages oder des Bereichs an und bietet die Möglichkeit, zwischen fünf verschiedenen Statusoptionen zu wählen. Zusätzlich gibt es ein Notizfeld, das es dem Benutzer ermöglicht, bei Bedarf weitere Informationen oder Anmerkungen hinzuzufügen.

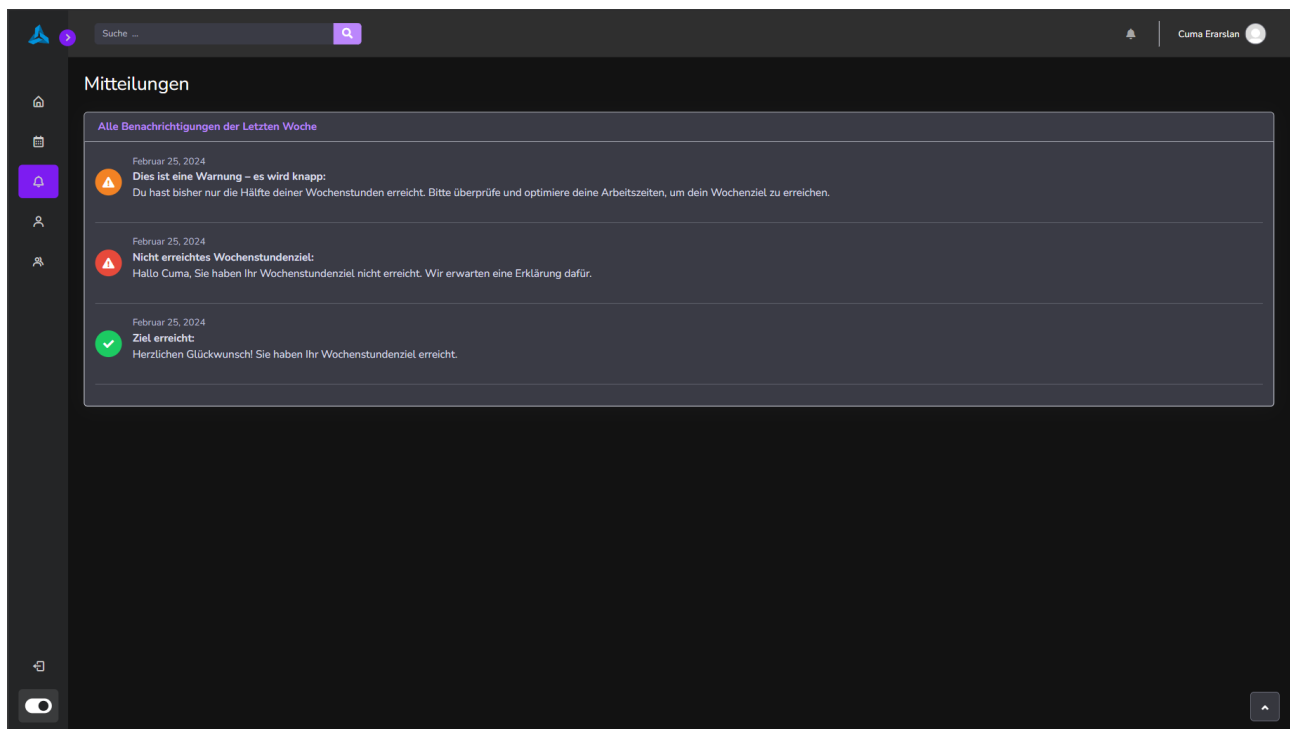
Nach dem Ausfüllen und Absenden des Bearbeitungsfensters durch den "Speichern"-Button, gibt es drei mögliche Meldungen:

1. **Erfolgsmeldung:** Eine Bestätigung wird angezeigt, wenn die Änderungen erfolgreich übernommen wurden.
2. **Fehlermeldung - Sonntag:** Falls der Benutzer versucht, den Status für einen Sonntag zu ändern, wird eine entsprechende Fehlermeldung angezeigt, da an Sonntagen keine Einträge gemacht werden können.
3. **Fehlermeldung - Nicht genug Urlaubstage:** Falls der Benutzer versucht, mehr Urlaubstage einzutragen, als verfügbar sind, wird eine Fehlermeldung angezeigt, um darauf hinzuweisen, dass nicht genügend Urlaubstage vorhanden sind.



## 4.7 Mitteilungen

Auf der Mitteilungsseite erhalten die Benutzer einen Überblick über alle Benachrichtigungen der letzten 7 Tage. Die Seite zeigt verschiedene Arten von Mitteilungen, von denen es insgesamt fünf gibt. Für normale Benutzer stehen drei Mitteilungen zur Verfügung, während Administratoren zusätzlich zwei Mitteilungen für jeden Mitarbeiter erhalten können. Beachten Sie, dass die Benachrichtigungen nach einer Woche automatisch gelöscht werden, und jede Minute neue Benachrichtigungen generiert werden können. So sieht die Seite mit den Mitteilungen für die Mitarbeiter aus:



Die drei Mitteilungen, die jeder Mitarbeiter erhalten kann, sind:

Desktop | Chrome

1. **Ziel erreicht:** Diese Mitteilung wird angezeigt, wenn der Nutzer sein wöchentliches Stundenkonto erfüllt hat.
2. **Warnung – es wird knapp:** Diese Mitteilung wird dem Nutzer ab Donnerstag gesendet, wenn er weniger als die Hälfte seiner Stunden für die Woche offen hat. Es dient als Warnung.
3. **Nicht erreichtes Wochenstundenziel:** Der Nutzer erhält diese Mitteilung am Sonntag, wenn er seine Wochenstunden nicht abgearbeitet hat. In dieser Mitteilung wird er aufgefordert, mit seinem Vorgesetzten zu sprechen.

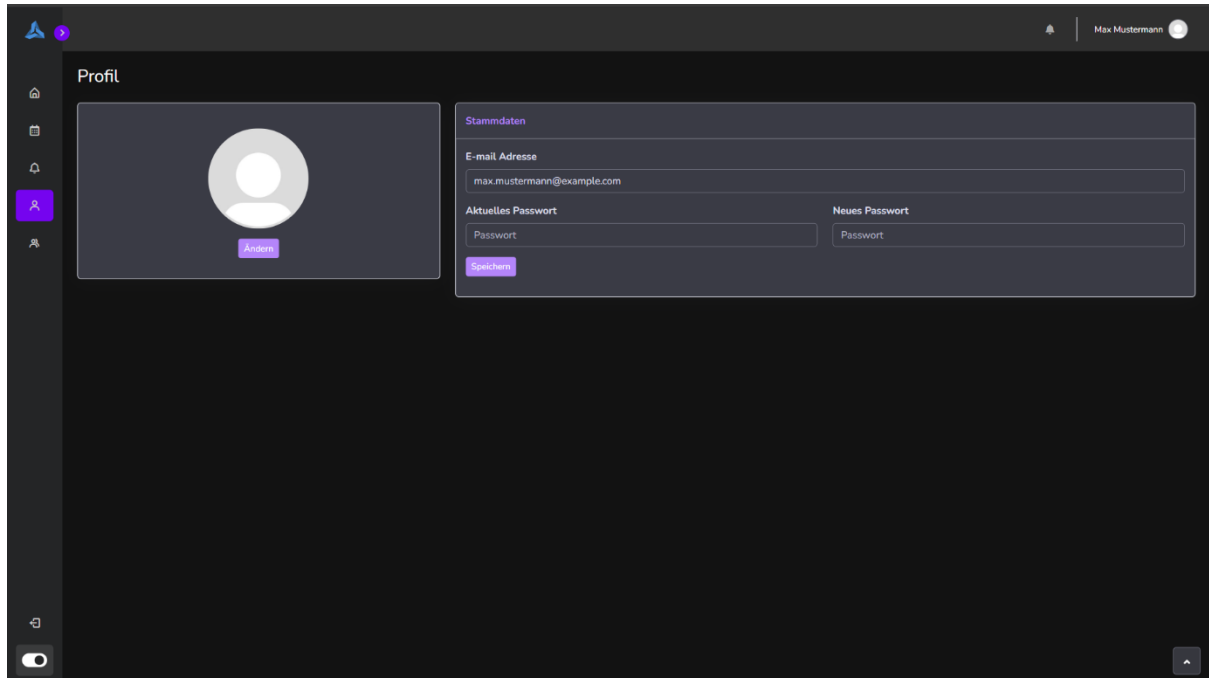
Die beiden zusätzlichen Mitteilungen für Administratoren sind die gleichen wie die Mitteilungen unter Punkt 2 und Punkt 3, die an jeden Mitarbeiter gehen, um den Administrator aufmerksam zu machen.





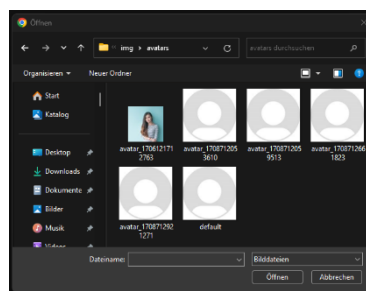
## 4.8 Profil

Auf der Profilseite haben Benutzer die Möglichkeit, ihre E-Mail-Adresse, ihr Passwort und ihr Profilbild zu ändern. So sieht die Profilseite aus:



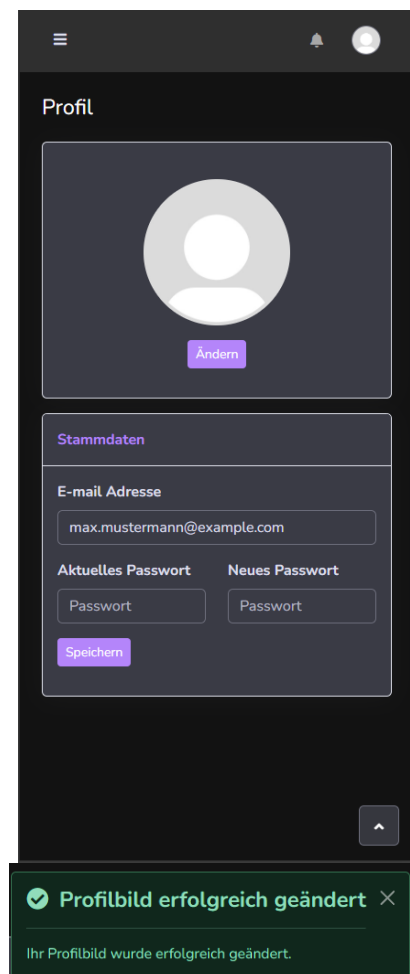
Desktop | Chrome

Um das Profilbild zu ändern, muss der Benutzer auf den "Ändern"-Button klicken, woraufhin sich ein Fenster öffnet, in dem das gewünschte Bild ausgewählt werden kann. Die unterstützten Datentypen für das Profilbild können je nach verwendetem Browser variieren, wobei wir jedoch die Verwendung von .jpeg und .png empfehlen.



Für die Änderung des Passworts oder der E-Mail muss der Benutzer zunächst sein altes Passwort in das entsprechende Feld eingeben und anschließend das neue Passwort oder die neue E-Mail eingeben. Nach dem Drücken der "Speichern"-Schaltfläche werden dem Benutzer drei verschiedene Meldungen angezeigt:

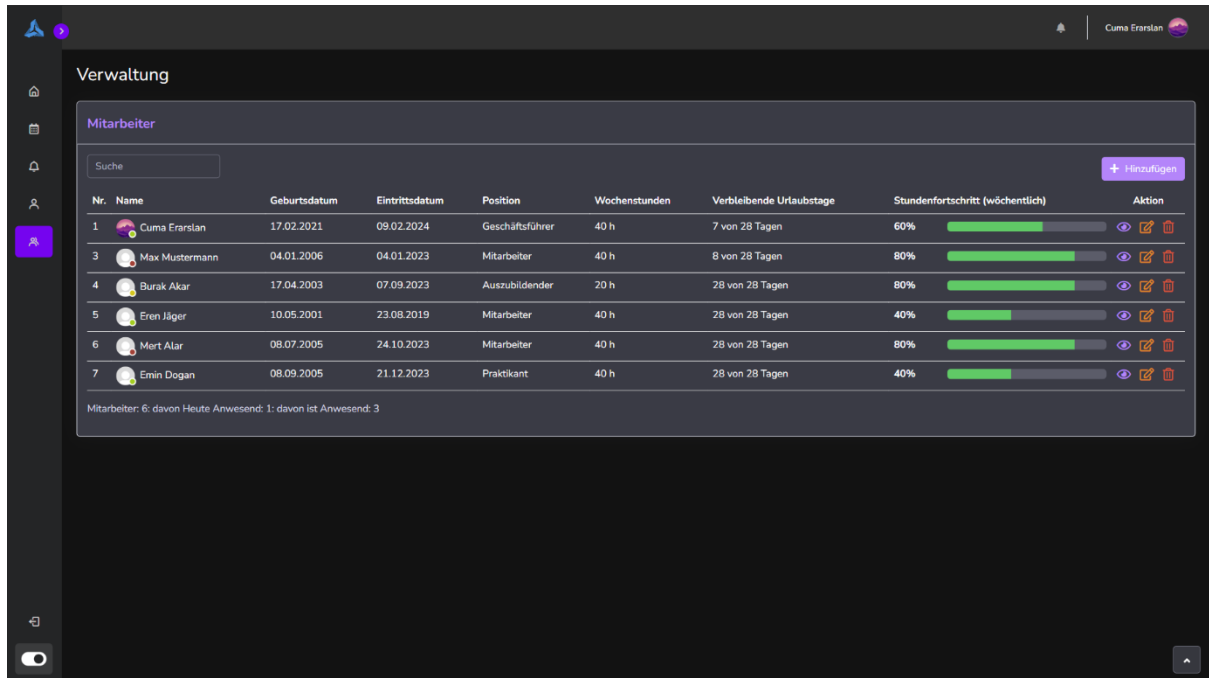
1. Stammdaten erfolgreich geändert
2. Stammdaten nicht geändert – Ihr aktuelles Passwort ist nicht korrekt.
3. Profilbild erfolgreich geändert – Ihr Profilbild wurde erfolgreich aktualisiert.





## 4.9 Verwaltung

Die Verwaltungsseite ist ein exklusiver Bereich, der ausschließlich für Benutzer mit Administratorrechten zugänglich ist. Hier haben Administratoren die Möglichkeit, umfassende Informationen über alle Mitarbeiter einzusehen, zu verwalten und gegebenenfalls Anpassungen vorzunehmen. So sieht die Seite aus:

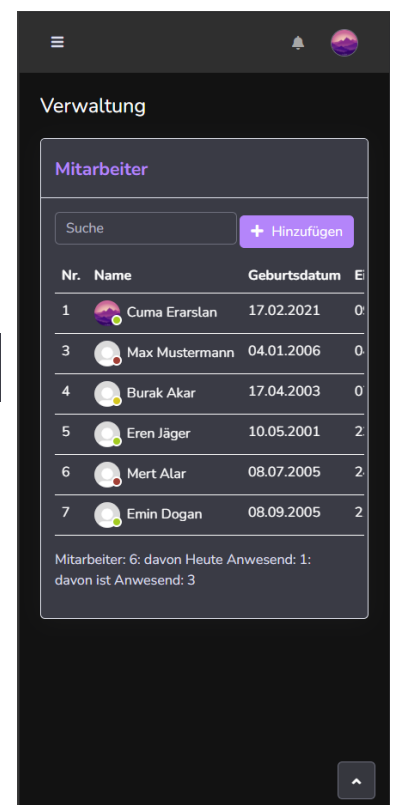


Desktop | Chrome

### 4.9.1 Tabellenübersicht

Die Tabelle auf der Verwaltungsseite bietet eine umfassende Übersicht über die Mitarbeiterdaten. Hier sind die verschiedenen Spalten und ihre Bedeutungen:

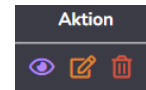
1. **Nr. (Personalnummer):** Die eindeutige Identifikationsnummer jedes Mitarbeiters.
2. **Profilbild mit Statuspunkt:** Das Profilbild jedes Mitarbeiters, das durch einen farbigen Statuspunkt ergänzt wird.
  - Grüner Punkt für gerade anwesend
  - Gelber Punkt für heute anwesend
  - roter Punkt für heute nicht anwesend.
3. **Name:** Der Vorname und Nachname des Mitarbeiters.
4. **Geburtsdatum:** Das Geburtsdatum des Mitarbeiters.
5. **Einstellungsdatum:** Das Datum, an dem der Mitarbeiter eingestellt wurde.
6. **Position:** Die Position des Mitarbeiters im Unternehmen.



iPhone 14 Pro Max | Chrome



7. **Wochenstunden:** Das festgelegte Wochenstundensoll des Mitarbeiters (z.B., 40 Stunden Vollzeit, 20 Stunden Teilzeit).
8. **Verbleibende Urlaubstage:** Die Anzahl der noch verfügbaren Urlaubstage für den Mitarbeiter.
9. **Stundenfortschritt:** Eine Fortschrittsleiste, die den prozentualen Fortschritt der erreichten Wochenstunden anzeigt. Aktualisiert sich alle 30 Sekunden.
10. **Aktionen:** Eine Spalte mit Symbolen für verschiedene Aktionen, darunter Kalender einsehen, Stammdaten bearbeiten und Mitarbeiter löschen.



Mitarbeiter								
Suche				+ Hinzufügen				
Nr.	Name	Geburtsdatum	Eintrittsdatum	Position	Wochenstunden	Verbleibende Urlaubstage	Stundenfortschritt (wöchentlich)	Aktion
1	Cuma Erarslan	17.02.2021	09.02.2024	Geschäftsführer	40 h	7 von 28 Tagen	60%	
3	Max Mustermann	04.01.2006	04.01.2023	Mitarbeiter	40 h	8 von 28 Tagen	80%	
4	Burak Akar	17.04.2003	07.09.2023	Auszubildender	20 h	28 von 28 Tagen	80%	
5	Eren Jäger	10.05.2001	23.08.2019	Mitarbeiter	40 h	28 von 28 Tagen	40%	
6	Mert Alar	08.07.2005	24.10.2023	Mitarbeiter	40 h	28 von 28 Tagen	80%	
7	Emin Dogan	08.09.2005	21.12.2023	Praktikant	40 h	28 von 28 Tagen	40%	

Mitarbeiter: 6 davon Heute Anwesend: 1 davon ist Anwesend: 3

Desktop | Chrome

- **Zusammenfassung unterhalb der Tabelle:** Die Gesamtzahlen für alle Mitarbeiter, heute anwesende Mitarbeiter und gerade anwesende Mitarbeiter sind unterhalb der Tabelle aufgeführt, um eine schnelle Übersicht über die Anwesenheitssituation zu bieten.
- **Suchfunktion oben links:** Ein Suchfeld ermöglicht Administratoren eine gezielte Suche nach Mitarbeitern anhand von Kriterien wie Name, Personalnummer oder Position. Diese Funktion erleichtert die Navigation und Verwaltung großer Mitarbeiterdatenbanken.

Mitarbeiter: 6 davon Heute Anwesend: 1 davon ist Anwesend: 3

Im Unteren Beispiel wird die Suchfunktion demonstriert:

Mitarbeiter								
Mit				+ Hinzufügen				
Nr.	Name	Geburtsdatum	Eintrittsdatum	Position	Wochenstunden	Verbleibende Urlaubstage	Stundenfortschritt (wöchentlich)	Aktion
3	Max Mustermann	04.01.2006	04.01.2023	Mitarbeiter	40 h	8 von 28 Tagen	80%	
5	Eren Jäger	10.05.2001	23.08.2019	Mitarbeiter	40 h	28 von 28 Tagen	40%	
10	Mert Alar	08.01.2003	30.08.2023	Mitarbeiter	40 h	28 von 28 Tagen	0%	

Desktop | Chrome





## 4.9.2 Mitarbeiter Hinzufügen

Nach dem Klicken auf das Plus-Symbol öffnet sich ein Fenster, in dem Administratoren die Daten des neuen Mitarbeiters eingeben können. Alle Felder müssen ausgefüllt werden. Hier sind die spezifischen Schritte:

- **Personalnummer wählen:**
  - Beim Drücken auf das Feld "Personalnummer" öffnet sich ein Menü mit allen noch unbelegten Personalnummern. Der Administrator kann entweder eine Nummer aus der Liste auswählen oder eine neue Nummer eingeben.
- **Eingabe der Stammdaten:**
  - Der Administrator trägt die Stammdaten des Mitarbeiters ein, darunter Vorname, Nachname, Geburtsdatum, Einstellungsdatum, Position und die gewünschten Wochenstunden.
- **Administratorrechte auswählen:**
  - Der Administrator kann festlegen, ob der neue Mitarbeiter Administratorrechte haben soll, indem er das entsprechende Optionsfeld auswählt. Beachten Sie, dass Administratorrechte den Zugriff auf die Verwaltungsseite ermöglichen.
- **Festlegen des Einmalpassworts:**
  - Ein einmaliges Passwort muss festgelegt werden, dass der Mitarbeiter bei der Registrierung wieder Ändern kann.
- **Speichern und Rückmeldung:**
  - Nachdem alle Felder ausgefüllt sind, kann der Administrator auf "Speichern" klicken. Die erfolgreiche Hinzufügung wird durch eine Erfolgsmeldung bestätigt, und die Tabelle wird automatisch um den neu hinzugefügten Mitarbeiter ergänzt.



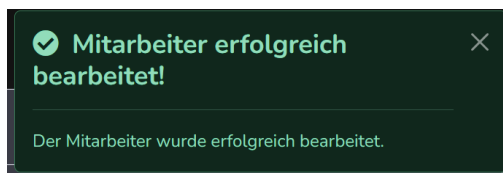
Diese Methode erleichtert den Prozess des Hinzufügens neuer Mitarbeiter und stellt sicher, dass alle erforderlichen Informationen korrekt eingetragen werden.



## 4.9.3 Mitarbeiter Bearbeiten

Die Verwaltungsseite bietet Administratoren auch die Möglichkeit, die Stammdaten der Mitarbeiter zu Bearbeiten. Folgende Schritte müssen für diesen Vorgang unternommen werden:

- Klicken Sie auf das orangefarbene Bearbeiten-Symbol des jeweiligen Mitarbeiters
- Ein Fenster öffnet sich mit den bestehenden Mitarbeitern
- Die Personalnummer ist ausgegraut und kann nicht geändert werden.
- Um die Anmeldeinformationen zurückzusetzen, geben Sie ein neues Passwort in das Feld ein.
- Nach dem Absenden des Formulars erhalten sie eine Erfolgsmeldung



**Mitarbeiter Bearbeiten**

Vorname \* Cuma Name \* Erarlan

Personalnummer \* 1 Administrator \* ☒ Ja ☐ Nein

Geburtsdatum \* 17.02.2021 Einstellungsdatum \* 09.02.2024

Position \* Geschäftsführung Wochenstunden \* Vollzeit (40h)

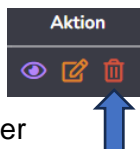
Anmeldeinformationen zurücksetzen

Neues Einmal Passwort

Verwerfen Speichern

## 4.9.4 Mitarbeiter Entfernen

Die Funktion "Mitarbeiter entfernen" ermöglicht es Administratoren, Mitarbeiter aus dem System zu löschen. Durch Klicken auf das Mülleimer-Icon neben dem jeweiligen Mitarbeiter öffnet sich ein Fenster, in dem die Löschaktion bestätigt werden muss. Nach der Bestätigung wird der Mitarbeiter mit seinem Profilbild aus der Datenbank entfernt.



Achtung!

Möchten sie den Mitarbeiter wirklich löschen?

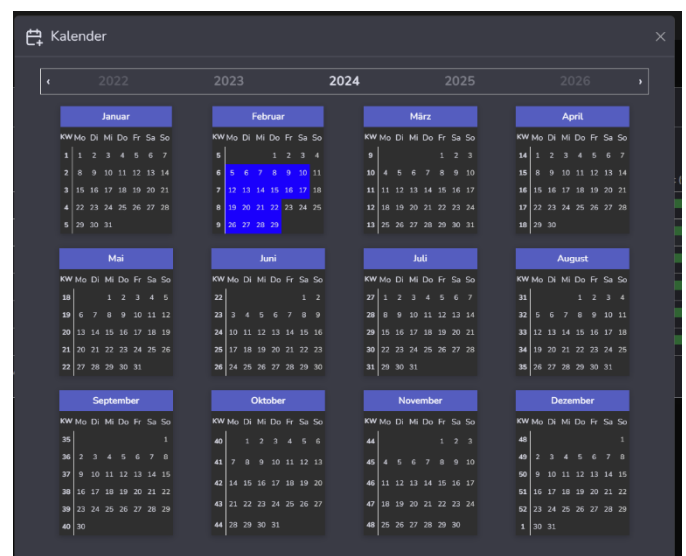
Abbrechen Bestätigen

## 4.9.5 Kalender

Die Funktion "Kalender einsehen und bearbeiten" ermöglicht es Administratoren, die Kalenderdaten eines bestimmten Mitarbeiters einzusehen und zu bearbeiten. Durch Klicken auf das Augen-Icon neben dem Mitarbeiter öffnet sich der Kalender des Mitarbeiters. Hier kann der Administrator die eingetragenen Arbeitszeiten und Statusänderungen sehen.



Die Kalenderseite zeigt farblich markierte Tage an, die den Status des Mitarbeiters an dem Tag widerspiegeln, wie z. B. Anwesend (grün), Krank (orange) oder Urlaub (blau). Durch Hovern über einen Tag erhält der Administrator weitere Details





wie die Anzahl der Stunden, die der Mitarbeiter an diesem Tag gearbeitet hat, sowie eventuelle Beschreibungen.

Darüber hinaus kann der Administrator durch Rechtsklicken auf einen Tag bestimmte Aktionen durchführen, wie das Löschen eines Tages. Es besteht jedoch keine Möglichkeit, neue Tage hinzuzufügen oder bestehende Einträge zu bearbeiten. Für solche Anpassungen sollte der Administrator die Stammdaten des Mitarbeiters bearbeiten oder das Passwort zurücksetzen und sich als der betreffende Mitarbeiter einloggen.

Februar								
KW	Mo	Di	Mi	Do	Fr	Sa	So	KW
5				1	2	3	4	9
6	5	6	7	8	9	10	11	10
7	12	13	14	15	16	17	18	11
8	19	20	21	22	23	24		
9	26	27	28	29				13

Löschen



## 5.0 Datenbank / Datenbankarchitektur

Die Datenbank und ihre Architektur spielen eine entscheidende Rolle in der Funktionalität und Zuverlässigkeit unserer Anwendung. In diesem Abschnitt werden wir einen detaillierten Einblick in die Struktur unserer Datenbank geben, die Hosting-Entscheidung und die Sicherheitsmaßnahmen.

Wir haben uns für die Nutzung von MySQL als Datenbankmanagementsystem entschieden, da es bereits im Rahmen unserer schulischen Ausbildung und des Java Summercamps Verwendung fand. Diese vorherige Erfahrung erleichtert die Entwicklung und Wartung der Datenbankstruktur erheblich.

### 5.1 Datenbankhosting

Wir haben uns für Microsoft Azure als Hosting-Plattform für unsere Datenbank entschieden. Dies bietet mehrere Vorteile, darunter die Möglichkeit, als Schüler kostenloses Guthaben von 100 US-Dollar zu erhalten. Durch die Nutzung von Azure können wir sicherstellen, dass beide, Entwickler und Benutzer, stets auf die aktuelle Datenbank zugreifen können. Dies ermöglicht eine reibungslose und konsistente Erfahrung für alle Beteiligten.

#### 5.1.1 Warum Microsoft Azure?

Die Auswahl von Microsoft Azure als Hosting-Plattform bietet uns folgende Vorteile:

- **Kostenloses Guthaben:** Azure for Students ermöglicht es Schülern, 100 US-Dollar Guthaben kostenlos zu nutzen.
- **Zuverlässigkeit:** Microsoft Azure ist bekannt für seine Zuverlässigkeit und bietet eine hohe Verfügbarkeit.
- **Sicherheit:** Um die Sicherheit unserer Daten zu gewährleisten, setzt Azure fortschrittliche Sicherheitsmaßnahmen ein. Dazu gehören beispielsweise IP-basierte Zugriffskontrollen, die dazu dienen, den Datenzugriff auf autorisierte Quellen zu beschränken und unerwünschte Zugriffe zu blockieren. Diese Sicherheitsvorkehrungen tragen zur Integrität und Vertraulichkeit unserer Daten bei.

Die Nutzung von Azure ermöglicht es uns, eine leistungsstarke Datenbankinfrastruktur zu nutzen, während gleichzeitig Ressourcen für andere Aspekte unserer Anwendung bereitgestellt werden können.

### 5.2 Sicherheitsmaßnahmen

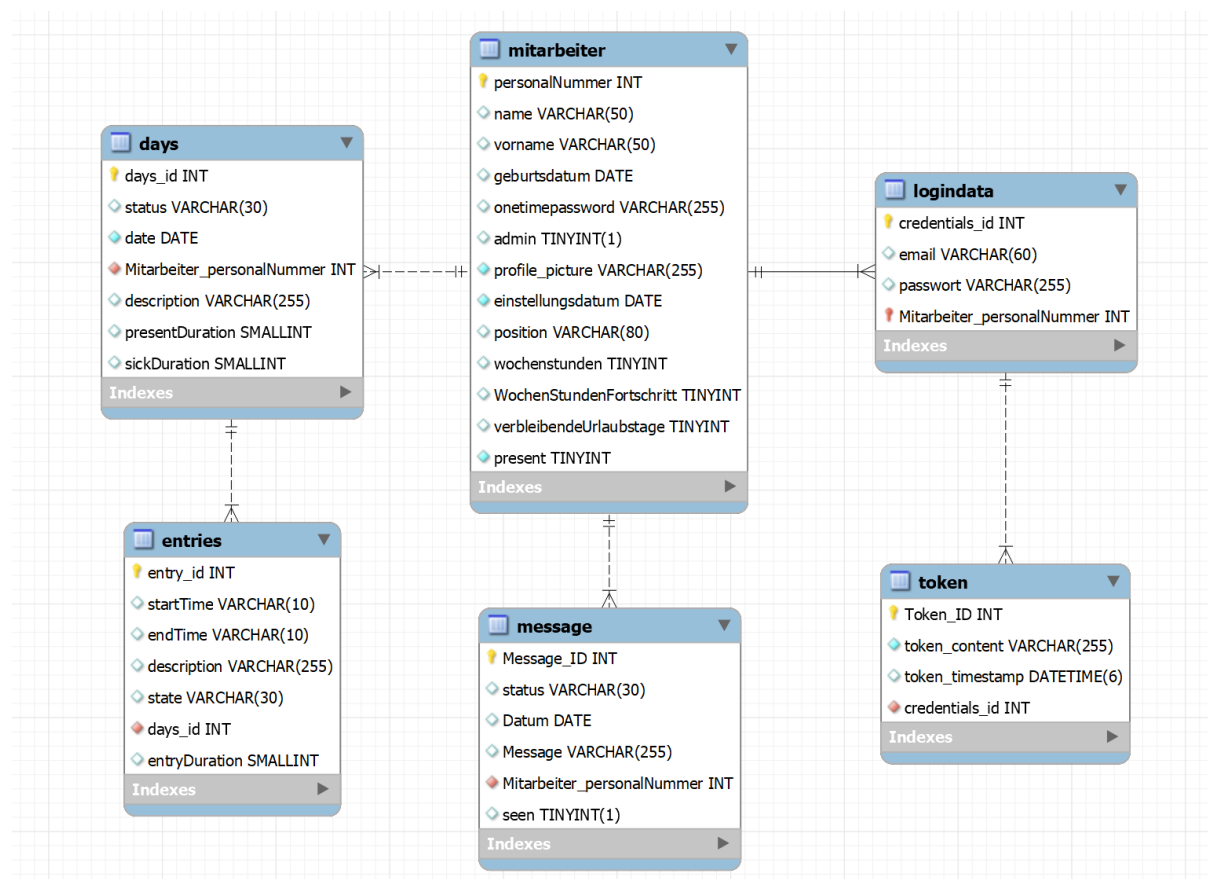
Die Sicherheit unserer Daten hat oberste Priorität. Zur Gewährleistung eines sicheren Datenaustauschs nutzen wir SSL (Secure Sockets Layer). Die SSL-Zertifikate sind im PEM-Format (Privacy Enhanced Mail), was eine zusätzliche Ebene der Sicherheit bietet. Durch die Verwendung von SSL werden Datenverschlüsselung und sichere Übertragungen gewährleistet, um die Vertraulichkeit und Integrität unserer Daten in der Datenbank zu schützen. Um den Zugriff auf unsere Datenbank zu beschränken und die Sicherheit zu erhöhen, haben wir eine IP-Beschränkung implementiert. Derzeit sind nur bestimmte IP-Adressen zugelassen. Bei der Veröffentlichung des Programms wird diese Beschränkung aufgehoben. Zusätzlich zu den beschriebenen Sicherheitsmaßnahmen wird regelmäßig ein lokales Backup der Datenbank erstellt, um jederzeit auf eine sichere Version zurückgreifen zu können.

## 5.3 Datenbankzugriffsschicht mit Hibernate und SQL-Query

Die Datenbankzugriffsschicht bildet die Schnittstelle zwischen der Anwendungslogik und der Datenbank. Hierbei setzen wir hauptsächlich auf Hibernate, ein leistungsstarkes ORM-Framework (Object-Relational Mapping), dass die Objekte unserer Anwendung direkt mit der Datenbank verbindet. Hibernate erleichtert die Datenbankinteraktion, indem es den Datenbankzugriff auf Objektebene ermöglicht. Zusätzlich verwenden wir auch herkömmliche SQL-Queries, um spezifische Anforderungen abzudecken, die nicht direkt von Hibernate abgedeckt werden. Diese hybride Herangehensweise bietet Flexibilität und Effizienz bei der Datenbankinteraktion in unserer Anwendung.

## 5.4 Datenbankstruktur

Die Datenbank bildet das Herzstück unserer Anwendung und ist entscheidend für die Speicherung und Organisation von Informationen. In diesem Abschnitt werden wir einen detaillierten Blick auf die Struktur unserer Datenbank werfen. Hierbei werden wir die verschiedenen Tabellen, ihre Attribute und die zugrunde liegende Architektur erläutern. Dies ermöglicht nicht nur ein besseres Verständnis der Datenbankfunktionalitäten, sondern auch Einblicke in die Gestaltung und Verwaltung unserer Daten. Hier haben wir eine Datenbankmodel unserer Datenbank mit allen Tabellen, Attributen und Beziehungen:



Das Datenbank-Skript, welches die Datenbank und ihre Tabellen erstellt, befindet sich im Verzeichnis "Datenbank" unter dem Hauptverzeichnis des Projekts. Der vollständige Pfad lautet: **Datenbank/Database.md**



## 5.4.1 Mitarbeiter

Die Tabelle "Mitarbeiter" fungiert als Haupttabelle in der Datenbankstruktur und ist dafür zuständig, die Informationen der Mitarbeiter der jeweiligen Firma zu speichern. Sie enthält sowohl persönliche Informationen zum Mitarbeiter als auch Daten zu seinem Stundenkonto. Die Datensätze in dieser Tabelle werden vorwiegend über die Verwaltungsseite erstellt und gelöscht.

Die Mitarbeiter Tabelle hat folgende Attribute:

1. **Personalnummer (Primärschlüssel):** Eindeutige Identifikationsnummer für jeden Mitarbeiter.
2. **Name:** Nachname des Mitarbeiters.
3. **Vorname:** Vorname des Mitarbeiters.
4. **Geburtsdatum:** Geburtsdatum des Mitarbeiters.
5. **OneTimePassword:** Bei der Registrierung genutztes Einmal Passwort des Mitarbeiters.
6. **Admin (Default 0):** Kennzeichnet, ob der Mitarbeiter Administratorrechte hat (0 für Nein, 1 für Ja).
7. **Profile Picture (Default ../resources/img/avatars/default.jpeg):** Relativer Pfad des Profilbildes wird auf der Profilseite festgelegt.
8. **Einstellungsdatum (Default 1900-01-01):** Einstellungsdatum des Mitarbeiters.
9. **Position (Default Keine Position):** Position des Mitarbeiters in der Firma.
10. **Wochenstunden (Default 1):** Anzahl der Wochenstunden, die der Mitarbeiter arbeiten soll.
11. **Wochenstunden Fortschritt (Default 0):** Fortschritt in Bezug auf die erwarteten Arbeitsstunden.
12. **Verbleibende Urlaubstage (Default 28):** Anzahl der noch verfügbaren Urlaubstage für den Mitarbeiter.
13. **Present (Default 0):** Anwesenheitsstatus des Mitarbeiters, 0 Für Abwesend, 1 Für Anwesend, 2 Für heute Anwesend

mitarbeiter	
name	varchar(50)
vorname	varchar(50)
geburtsdatum	date
onetimepassword	varchar(255)
admin	tinyint(1)
profile_picture	varchar(255)
einstellungsdatum	date
position	varchar(80)
wochenstunden	tinyint
WochenStundenFortschritt	tinyint
verbleibendeUrlaubstage	tinyint
present	tinyint
personalNumber	int

Die "Personalnummer" dient als Primärschlüssel und muss manuell zugewiesen werden. Datensätze können über die Verwaltungsseite erstellt, gelöscht sowie bearbeitet werden.

Beispiel:

personalNumber	name	vorname	geburtsdatum	onetimepassword	admin	profile_picture	einstellungsdatum	position	Wochenstunden	WochenStundenFortschritt	verbleibend...	present	
1	2	Ernastan	Cuma	2003-05-14	\$2a\$12\$Y8SVk5Z1..VB..	1	../resources/img/ava...	2023-10-12	Geschäftsführer	40	3	28	2
2	3	Mustermann	Max	2006-01-04	\$2a\$12\$92EPxLIGMK97S...	1	../resources/img/ava...	2023-01-04	Mitarbeiter	40	32	0	0
3	10	Alan	Hert	2003-01-08	\$2a\$12\$iv3TehTERFNq...	0	../resources/img/ava...	2023-08-30	Mitarbeiter	40	0	28	0



## 5.4.2 Logindata

Die Tabelle "Logindata" enthält die Anmeldeinformationen der Mitarbeiter. Diese Informationen werden bei der Registrierung erstellt und können in der Verwaltungsseite gelöscht werden. Hier sind die Attribute der Tabelle:

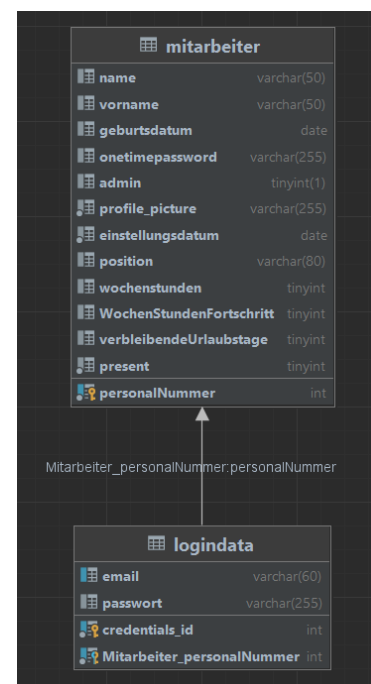
1. **Credentials id (Primärschlüssel):** Eindeutige Identifikationsnummer für jeden Datensatz (Auto-Increment).
1. **E-Mail (Unique Index):** E-Mail-Adresse des Mitarbeiters, die für die Anmeldung verwendet wird.
2. **Passwort:** Passwort des Mitarbeiters, das für die Anmeldung verschlüsselt gespeichert wird.
3. **Personalnummer (Fremdschlüssel und Primärschlüssel):** Diese Nummer dient als Primärschlüssel und gleichzeitig als Fremdschlüssel der auf die "Personalnummer" in der Tabelle "Mitarbeiter" verweist.

logindata	
email	varchar(60)
passwort	varchar(255)
credentials_id	int
Mitarbeiter_personalNumber	int

Der Unique Index für die E-Mail in der Tabelle "Logindata" gewährleistet, dass jede E-Mail-Adresse nur einmal in der Tabelle vorhanden sein kann. Das bedeutet, dass keine zwei Mitarbeiter denselben E-Mail-Account für die Anmeldung verwenden können.

Beziehung:

Die Tabelle "Logindata" hat eine 1-zu-1-Beziehung zur Tabelle "Mitarbeiter". Jeder Datensatz in "Logindata" ist eindeutig einer bestimmten Person zugeordnet und verweist auf deren Anmeldedaten. Diese Verbindung ermöglicht eine präzise Zuordnung der Anmeldedaten zu den entsprechenden Mitarbeiterprofilen in der Datenbank.



Beispiel:

credentials_id	email	passwort	Mitarbeiter...
1	57 cumaerarslan11@gmail.com	\$2a\$12\$1rDawR65zDYQBR2sCK/HyeazDqURee7ed.j05/L8eCqAg.Mt75XW.	2





## 5.4.3 Token

Die Tabelle "Token" speichert Informationen über die Benutzersitzungen, wenn die Option "Angemeldet bleiben" ausgewählt wurde. Der Token wird nur erstellt und gespeichert, wenn die Funktion "Angemeldet bleiben" aktiviert ist. Die Tokens werden in Cookies gespeichert und haben eine Gültigkeitsdauer von 14 Tagen. Nach diesem Zeitraum werden sowohl die Cookies als auch der entsprechende Datensatz in der "Token"-Tabelle gelöscht. Darüber hinaus erfolgt beim Ausloggen des Benutzers auch die Löschung des Tokens und des zugehörigen Cookies. Hier sind die Attribute der Tabelle:

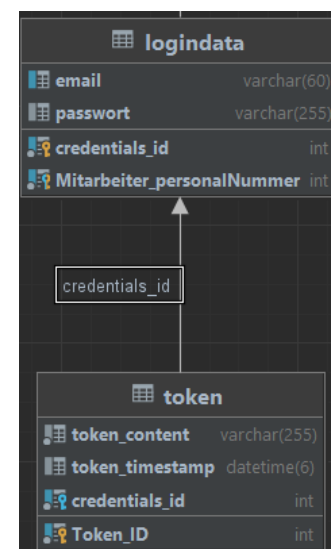
2. **Token id (Primärschlüssel):** Eindeutige Identifikationsnummer für jeden Datensatz (Auto-Increment).
1. **Token Timestamp:** Erfasst den genauen Zeitpunkt der Token-Erstellung.
2. **Token Content:** Wird aus einer UUID generiert, diese wird auch im Cookie gespeichert.
3. **Credentials id (Fremdschlüssel):** Dieser Fremdschlüssel verweist auf einen „Logindata“ Datensatz in der Tabelle „Logindata“

token	
token_content	varchar(255)
token_timestamp	datetime(6)
credentials_id	int
Token_ID	int

Der „Token Timestamp“ hat eine Uhrzeit Präzision von bis zu 6 Dezimalstellen nach dem Komma.

Beziehung:

Die Beziehung zwischen der Tabelle "Token" und der Tabelle "Logindata" ist als 1-zu-N (eins zu viele) definiert, da jeder Datensatz in der Tabelle "Logindata" mehrere zugehörige Datensätze in der Tabelle "Token" haben kann. Dies spiegelt die Situation wider, dass ein Benutzer sich auf verschiedenen Geräten anmelden und auf jedem Gerät angemeldet bleiben kann, wobei für jedes Gerät ein eigenes Token erstellt wird.



Beispiel:

Token_ID	token_content	token_timestamp	credentials_id
1	68 be231f0e-f4b1-4431-96a0-65279c4a51dd	2024-02-26 18:38:51.878000	57





## 5.4.4 Days

Die Tabelle "Days" spielt eine zentrale Rolle bei der Verwaltung von Tagesdatensätzen. Sie ermöglicht die Erstellung, Bearbeitung und Löschung von Datensätzen, die hauptsächlich im Kalender genutzt werden. Die Attribute dieser Tabelle dienen nicht nur dazu, den Status des Tages, das Datum und die Personalnummer des Mitarbeiters zu speichern, sondern sie werden auch für die Generierung von Statistiken und die Berechnung des Wochenstundenfortschritts verwendet.

Hier sind die Attribute der Tabelle:

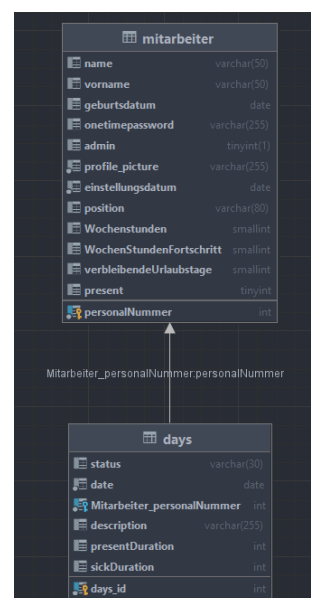
3. **Days id (Primärschlüssel):** Eindeutige Identifikationsnummer (Auto-Increment).
1. **Status:** Der Status des Tages.
2. **Date:** Das Datum des Tages.
3. **Description:** Die Beschreibung des Tages.
4. **Present Duration (Default 0):** Gibt die Stunden an, die als Anwesenheit betrachtet werden.
5. **Sick Duration (Default 0):** Gibt die Stunden an, die als krank betrachtet werden.
6. **Personalnummer (Fremdschlüssel):** Diese Nummer dient als Fremdschlüssel für die Tabelle Mitarbeiter und verweist auf einen Mitarbeiter.

days	
status	varchar(30)
date	date
Mitarbeiter_personalNumber	int
description	varchar(255)
presentDuration	int
sickDuration	int
days_id	int

Der Status des Tages berechnet sich aus den Einträgen. Sick Duration und Present Duration berechnen sich aus den Einträgen. Description kann Null sein.

Beziehung:

Die Tabelle „Days“ hat eine 1-zu-N-Beziehung zur Tabelle „Mitarbeiter“. Ein Mitarbeiter kann mehrere Tage im Kalender haben, während ein Tag immer genau einem Mitarbeiter zugeordnet ist.



Beispiel:

days_id	status	date	Mitarbeiter_personalNumber	description	presentDuration	sickDuration
1	295 Urlaub	2024-02-05	3		0	0
2	296 Urlaub	2024-02-06	3		0	0
3	297 Urlaub	2024-02-07	3		0	0



## 5.4.5 Entries

Die Tabelle „Entries“ ist für die Speicherung von Einträgen verantwortlich, die hauptsächlich auf der Hauptseite erstellt und gelöscht werden, aber auch im Kalender beim Bearbeiten, Löschen oder Erstellen von Tagen. Die Daten in dieser Tabelle werden in „Days“ zusammengefasst. Die Einträge des aktuellen Tages werden auf der Hauptseite angezeigt.

Hier sind die Attribute:

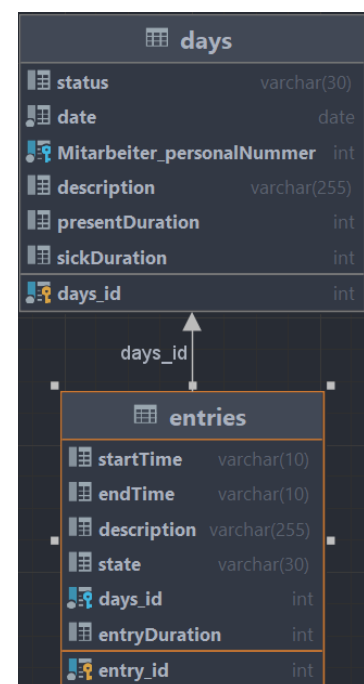
4. **Entry id (Primärschlüssel):** Eindeutige Identifikationsnummer (Auto-Increment).
5. **Start Time:** Start Uhrzeit des Eintrags.
6. **End Time:** End Uhrzeit des Eintrags.
7. **Description:** Beschreibung des Eintrags.
8. **State:** Status des Eintrages.
9. **Entry Duration:** Gesamtzeitspanne des Eintrages in Minuten.
10. **Days id (Fremdschlüssel):** Verweist auf einen Tag in der Tabelle Days

entries	
startTime	varchar(10)
endTime	varchar(10)
description	varchar(255)
state	varchar(30)
days_id	int
entryDuration	int
entry_id	int

„Start Time“ und „End Time“ wird im Format HH:MM gespeichert. Die Entry Duration berechnet sich aus dem Start und End Time Attribut.

Beziehung:

Die Tabelle "entries" hat eine 1-zu-N-Beziehung zu der Tabelle "days", da ein Tag mehrere Einträge haben kann, wie zum Beispiel einen Anwesenheits- und einen Pauseneintrag an einem Arbeitstag.



Beispiel:

entry_id	startTime	endTime	description	state	days_id	entryDuration
543	08:00	16:00	Automatisch gener...	Urlaub	295	480
544	08:00	16:00	Automatisch gener...	Urlaub	296	480
545	08:00	16:00	Automatisch gener...	Urlaub	297	480



## 5.4.6 Message

Die Tabelle "message" speichert Benachrichtigungen für Mitarbeiter und Administratoren. Diese Benachrichtigungen werden auf der Nachrichtenseite angezeigt sowie die Letzten 3 in der Navigationsleiste. Nach 6 Tagen werden die Benachrichtigungen automatisch gelöscht.

Hier sind die Attribute:

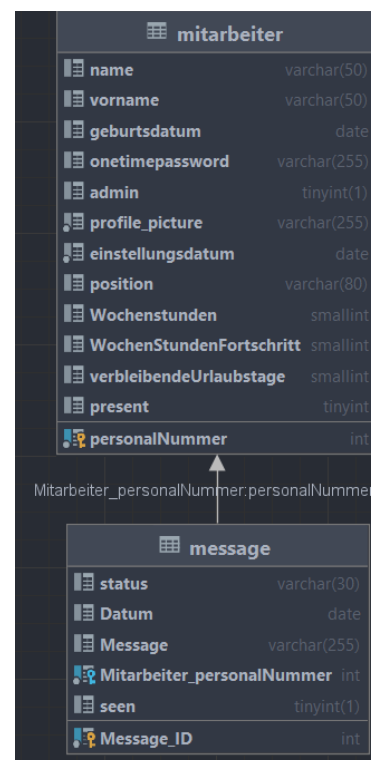
1. **Message\_ID (Primärschlüssel):** Eindeutige ID für jede Nachricht (Auto-Increment).
2. **Status:** Der Status/ Die Art der Nachricht.
3. **Datum:** Das Datum, an dem die Nachricht erstellt wurde.
4. **Message:** Der eigentliche Text der Nachricht.
5. **Seen:** Ein Flag, das anzeigt, ob die Nachricht bereits vom Mitarbeiter gesehen wurde.
6. **Personalnummer (Fremdschlüssel):** Die Personalnummer des Mitarbeiters, für den die Nachricht bestimmt ist.

message	
status	varchar(30)
Datum	date
Message	varchar(255)
Mitarbeiter_personalNumber	int
seen	tinyint(1)
Message_ID	int

Im Attribut Message werden auch die HTML-Formatierungen wie z.b. „<strong>“ gespeichert.

Beziehung:

Die Tabelle „message“ hat eine 1-zu-N-Beziehung zu „mitarbeiter“, Jeder Datensatz in „message“ ist nur einem Datensatz in „mitarbeiter“ zugeordnet. Das Repräsentiert das Szenario, dass ein Mitarbeiter mehrere Benachrichtigungen erhalten kann.



Beispiel:

	Message_ID	status	Datum	Message	Mitarbeiter...	seen
1	118	warning	2024-02-25	<strong>Unvollständige Arbeitsstundenwarnung</s...	3	1
2	119	danger	2024-02-25	<strong>Wochenstundenziel nicht erreicht - Hand...	3	1
3	127	warning	2024-02-25	<strong>Dies ist eine Warnung - es wird knapp: ...	3	1
4	128	danger	2024-02-25	<strong>Nicht erreichtes Wochenstundenziel:</st...	3	1
5	129	success	2024-02-25	<strong>Ziel erreicht:</strong> Herzlichen...	3	1

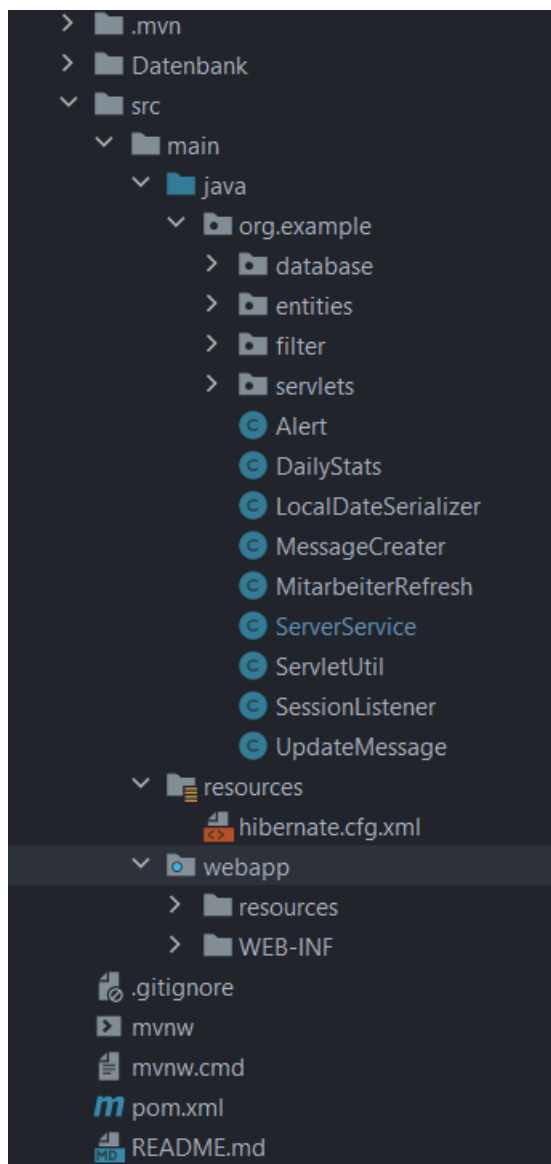


## 6.0 Quellcode Übersicht

In diesem Abschnitt erhalten Sie einen umfassenden Einblick in den Quellcode dieses Projekts. Hier wird die Struktur, die Funktionalitäten und die Interaktionen der verschiedenen Klassen, Skripte und Module detailliert erläutert. Diese Dokumentation bietet Ihnen eine klare Übersicht darüber, wie das Projekt aufgebaut ist und wie die einzelnen Komponenten miteinander interagieren.

### 6.1 Projekt Struktur

In diesem Abschnitt werden die wesentlichen Elemente der Projektstruktur erläutert, die die Grundlage für die Organisation und Verwaltung des Codes bilden. Eine klare Projektstruktur ist entscheidend für die Übersichtlichkeit und Wartbarkeit des Codes. Hier werden die Hauptordner und Dateien im Projektordner "TrackInGfos" vorgestellt. Die Informationen bieten einen Einblick in die Organisation des Projekts und erleichtern das Verständnis der Entwicklungs-Prozesse. Für detaillierte Informationen zu Maven-spezifischen Elementen wie den Dateien "mvnw" und "mvnw.cmd" sowie dem Ordner ".mvn" wird auf die [offizielle Maven-Dokumentation](#) verwiesen. Eine detaillierte Ordnerübersicht wird im Folgenden präsentiert.



IntelliJ Projekt Struktur

- **Datenbank**
  - Enthält das Datenbank-Skript und das SSL-Zertifikat.
- **pom.xml**
  - Konfigurationen und Abhängigkeiten für Maven.
- **src**
  - Hauptordner für den Quellcode des Projekts.
- **src/main**
  - **java:** Java-Quellcode.
  - **resources:** Hibernate-Konfigurationsdatei.
  - **webapp:** Website-Dateien.
- **java/org.example**
  - Enthält Java-Klassen.
- **java/org.example/database**
  - Klassen für die Verwaltung von Transaktionen in der Datenbank.
- **java/org.example/entities**
  - Entitätsklassen für die Datenbanktabellen.
- **java/org.example/servlets**
  - Servlet Klassen für die Anwendung.
- **webapp/resources**
  - Beinhaltet Fonts, Bilder, Icons, CSS, JS sowie JS-Bibliotheken z.b. Bootstrap.
- **webapp/WEB-INF**
  - **jsp:** Enthält JavaServer Pages (JSP)-Dateien, die als HTML-Vorlagen für die Benutzeroberfläche dienen.
  - **web.xml:** Die Konfigurationsdatei für das Webanwendungsprojekt. Hier werden Servlets, Filter und andere Einstellungen definiert.

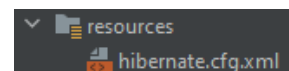


## 6.2 Hibernate

Hibernate ist ein leistungsstarkes **ORM** (Object-Relational Mapping)-Framework, das die Kommunikation zwischen einer Java-Anwendung und einer relationalen Datenbank erleichtert. In diesem Abschnitt wird erläutert, wie Hibernate in das TrackIn-Projekt integriert ist und welche Vorteile es für die Datenbankinteraktion bietet.

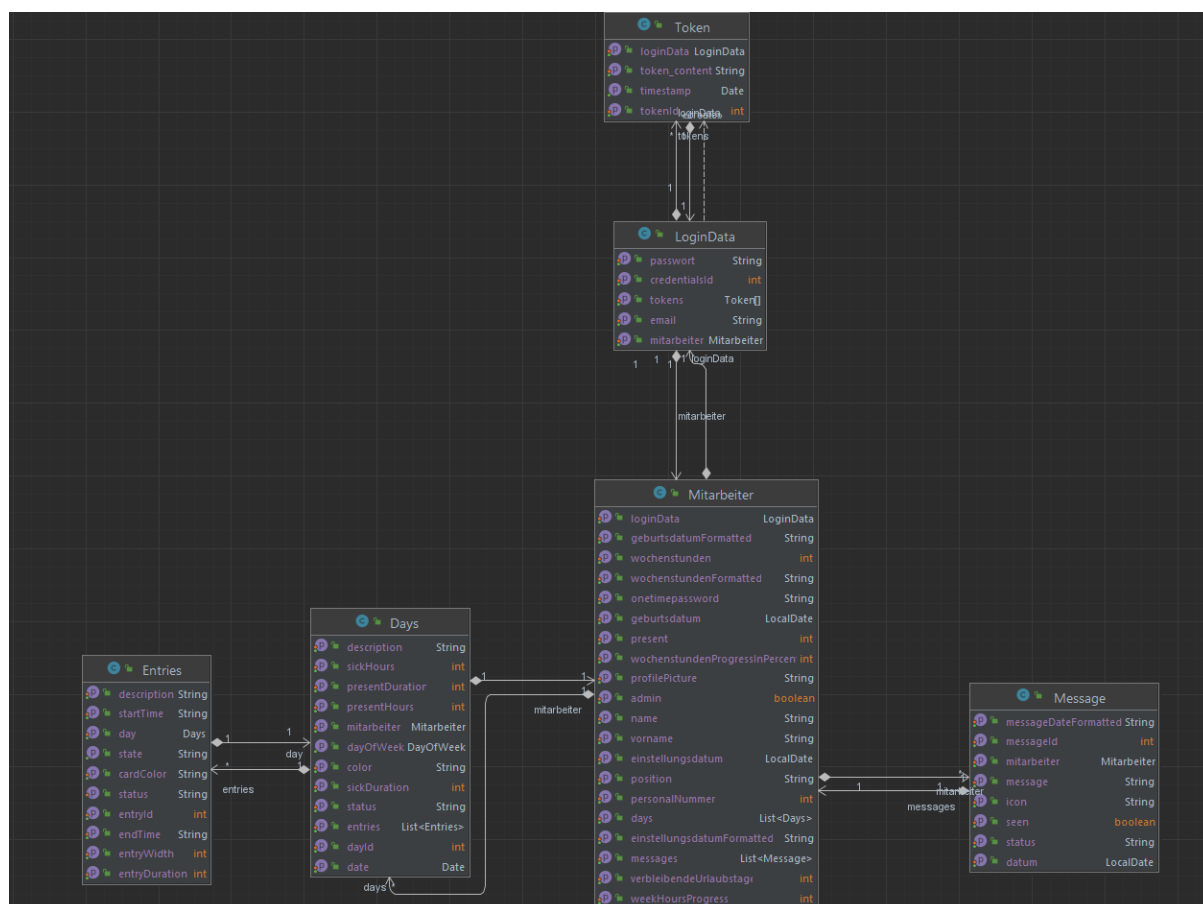
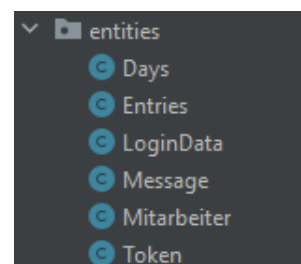
### 6.2.1 Hibernate-Konfiguration

Die Konfigurationsdatei für Hibernate (**hibernate.cfg.xml**) befindet sich im Ordner **resources**. Hier werden Einstellungen wie die JDBC-Verbindungs-URL, der Benutzername, das Passwort und andere relevante Informationen für die Datenbankverbindung festgelegt. Die Konfiguration ermöglicht es Hibernate, die richtigen Datenbanktabellen mit den Java-Entitäten zu verknüpfen.



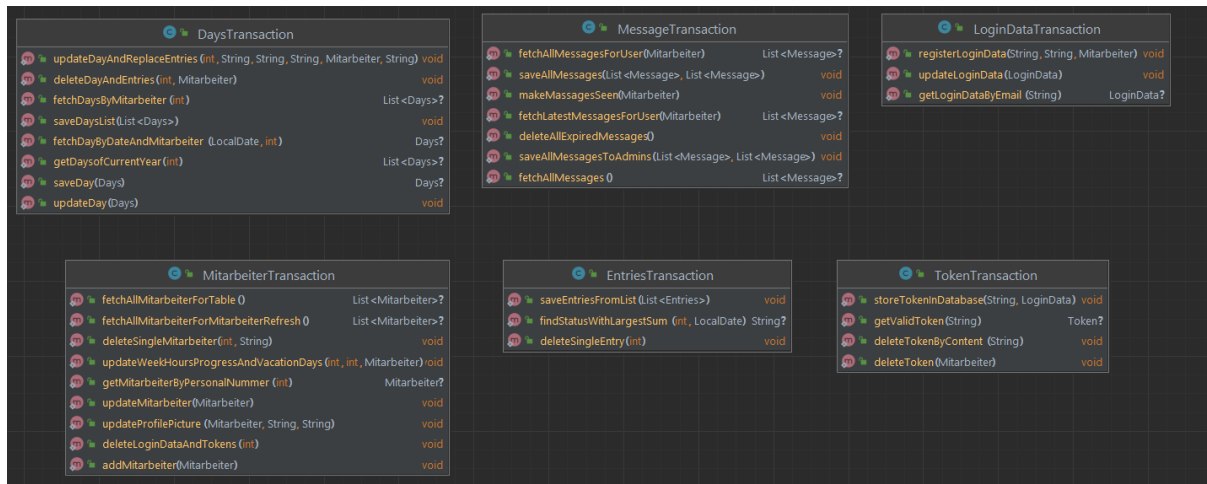
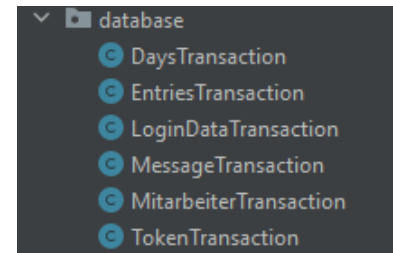
### 6.2.2 Entitätsklassen

Die Java-Klassen im Verzeichnis **java/org.example/database/entities** fungieren als Entitäten für die Datenbank. Jede dieser Klassen repräsentiert eine Tabelle in der Datenbank. Hibernate nutzt Annotations wie **@Entity**, **@Id**, **@Column** und andere, um die Beziehung zwischen den Klassenattributen und den Datenbankspalten festzulegen. Dabei werden auch Getter- und Setter-Methoden in den Klassen für jedes Attribut implementiert, um den Zugriff auf die Attributwerte zu ermöglichen. Zusätzlich wird das **@Transient**-Attribut genutzt, um bestimmte Attribute als nicht persistent zu markieren, wenn sie nicht in der Datenbank gespeichert werden sollen. Hier sind die UML-Diagramme ohne Getter und Setter:



## 6.2.3 Datenbankoperation mit Hibernate

Die Klasse **HibernateUtil** im Verzeichnis **java/org.example/database** stellt Methoden bereit, um eine Hibernate-Session zu erstellen und Datenbankoperationen durchzuführen. Dies umfasst das Hinzufügen, Aktualisieren, Löschen und Abrufen von Datensätzen aus der Datenbank. Hibernate vereinfacht die Datenbankinteraktion, indem es SQL-Anfragen abstrahiert und dem Entwickler eine objektorientierte Schnittstelle bietet. Hier sind die UML-Diagramme der Datenbank Klassen:



## 6.2.4 Beziehung zwischen Entitäten

Hibernate unterstützt das Mapping von Beziehungen zwischen verschiedenen Entitätsklassen. Dies ermöglicht es, komplexe Datenbankstrukturen abzubilden, beispielsweise wenn eine Tabelle auf eine andere verweist. Durch die Nutzung von Hibernate-Annotationen können One-to-One-, One-to-Many- und Many-to-Many-Beziehungen definiert werden.



## 6.3 Verarbeitung von Anfragen

In diesem Abschnitt wird die Verarbeitung von HTTP-Anfragen in der TrackIn-Webanwendung beleuchtet. Erfahren Sie, wie Servlets, Sessions, Filter und etc. in der Anwendung genutzt werden, um auf verschiedene Arten von Anfragen zu reagieren.

### 6.3.1 Sessions

TrackIn verwendet Sessions, um den Zustand der Benutzerinteraktionen über mehrere Anfragen hinweg aufrechtzuerhalten. Eine Session ermöglicht es, Informationen zwischen verschiedenen Seitenaufrufen zu speichern. In TrackIn werden Sessions verwendet, um den angemeldeten Benutzer zu verfolgen und sicherzustellen, dass nur autorisierte Benutzer auf bestimmte Funktionen zugreifen können.

Beim Anmelden eines Benutzers wird eine Sitzung gestartet, und ein eindeutiger Sitzungsidentifikator (Session-ID) wird dem Benutzer zugeordnet. Diese Session-ID wird in Form eines Cookies auf dem Gerät des Benutzers gespeichert. Während der Sitzung können Informationen wie der Name, der Token und andere relevante Daten in der Session gespeichert und abgerufen werden. Wenn der Benutzer sich abmeldet oder die Sitzung endet, werden die entsprechenden Daten aus der Session entfernt.

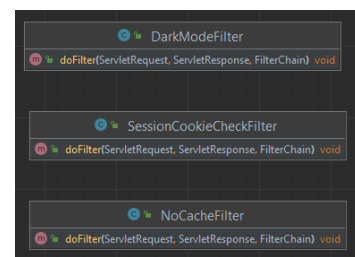
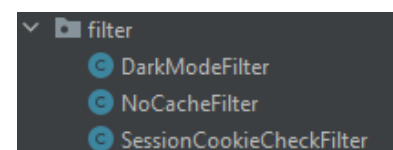
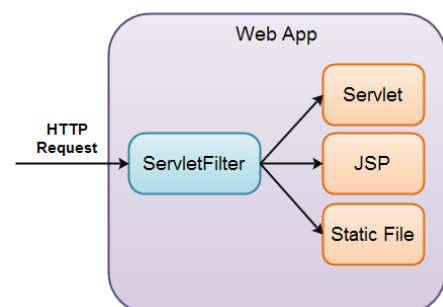
Die Verwendung von Sessions verbessert die Benutzererfahrung und ermöglicht eine sichere und effiziente Kommunikation zwischen dem Benutzer und der Anwendung.

### 6.3.2 Filter

Filter sind spezielle Komponenten in Java Servlets, die verwendet werden, um Anfragen und Antworten während ihres Durchlaufs durch die Filterkette zu modifizieren oder zu überwachen. Sie bieten eine saubere Möglichkeit, gemeinsame Aufgaben wie Authentifizierung, Autorisierung, Protokollierung und Caching zu behandeln, ohne den eigentlichen Servlet-Code zu verunreinigen. Filter können vor oder nach der eigentlichen Verarbeitung einer Anfrage oder Antwort ausgeführt werden, was sie zu einem leistungsstarken Instrument für die Implementierung von sicherheitsrelevanten Funktionen in einer Webanwendung macht.

Im Filter-Abschnitt der TrackIn-Webanwendung werden verschiedene Filter eingesetzt, um die Anfragen zu verarbeiten und bestimmte Funktionalitäten zu implementieren. Hier werden unsere drei Filter kurz vorgestellt:

- **SessionCookieCheckFilter:** Der „SessionCookieCheckFilter“ spielt eine entscheidende Rolle bei der Sicherstellung der Authentizität angemeldeter Mitarbeiter. Durch Überprüfung der Gültigkeit von Sitzungen und Cookies gewährleistet der Filter, dass nur Benutzer mit einer aktiven und gültigen Sitzung auf bestimmte Ressourcen zugreifen können. Besonders hervorzuheben ist die Funktion zur Überprüfung der "Angemeldet bleiben"-Option, die es Benutzern ermöglicht, über einen längeren Zeitraum eingeloggt zu bleiben. Falls keine gültige Sitzung besteht, wird der







Benutzer automatisch zur Anmeldeseite weitergeleitet, wo er sich erneut authentifizieren kann.

- **NoCacheFilter:** Der NoCacheFilter verhindert das Zwischenspeichern (Caching) von Ressourcen im Browser. Dadurch wird sichergestellt, dass Benutzer immer die aktuelle Version der Webseite erhalten und keine veralteten Daten angezeigt werden. Dies ist besonders relevant, um sicherzustellen, dass sensible Informationen nach dem Ausloggen nicht mehr über den Browserverlauf abgerufen werden können.
- **DarkModeFilter:** Der DarkModeFilter ermöglicht es Benutzern, einen Dunkelmodus für die Anwendung zu aktivieren. Dies geschieht durch das Setzen eines Cookies, das den aktuellen Zustand des Dunkelmodus speichert. Der Filter liest dieses Cookie aus und stellt sicher, dass die Webseite im entsprechenden Modus gerendert wird.

### 6.3.3 Servlets

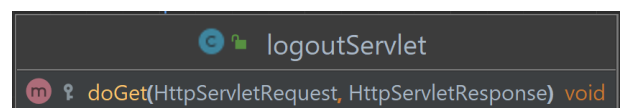
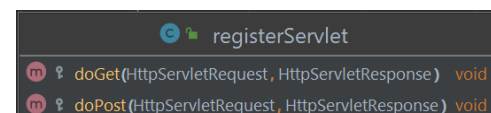
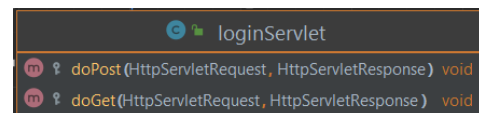
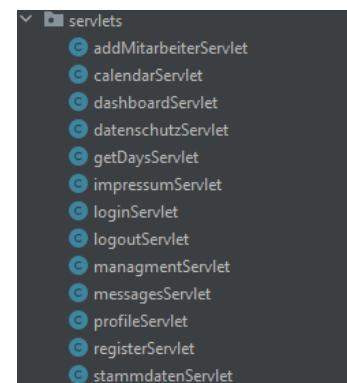
Die Anwendung verwendet **Servlets** als zentrale Steuerungseinheiten für die Verarbeitung von **HTTP-Anfragen**. Jede JSP-Seite ist einem oder mehreren Servlets zugeordnet, die für die Handhabung von Benutzeranfragen und die Verwaltung des Anwendungsflusses verantwortlich sind.

#### Struktur und Funktion:

**LoginServlet:** Das LoginServlet ist für den Anmeldeprozess zuständig und führt folgende Aufgaben durch. Es validiert die vom Benutzer bereitgestellten Anmeldeinformationen und erstellt bei erfolgreicher Anmeldung eine Benutzersitzung. Wenn die Option "Angemeldet bleiben" aktiviert ist, wird zusätzlich ein Token erstellt und in einem Cookie gespeichert. Dieses Cookie hat eine Gültigkeitsdauer von 14 Tagen, wodurch der Benutzer automatisch angemeldet bleibt, auch nach dem Schließen des Browsers. Nach erfolgreichem Login leitet das Servlet den Benutzer auf die entsprechende Seite weiter, um die Interaktion mit der Anwendung fortzusetzen.

**RegisterServlet:** Das RegisterServlet ermöglicht neuen Benutzern die Registrierung. Es validiert die eingegebenen Informationen, erstellt einen neuen LoginData Datensatz in der Datenbank und ermöglicht die Anmeldung nach erfolgreichem Abschluss.

**LogoutServlet:** Das LogoutServlet übernimmt die Abmeldung von Benutzern und führt dabei mehrere wichtige Aufgaben aus. Zunächst erfolgt die Löschung des "Angemeldet bleiben"-Cookies, welches den Token enthält. Dies stellt sicher, dass der Benutzer nicht mehr automatisch angemeldet wird. Des Weiteren schließt das Servlet die aktuelle Benutzersitzung, um sicherzustellen, dass keine weiteren Anfragen im Namen des abgemeldeten Benutzers durchgeführt werden können. Schließlich erfolgt die Löschung des Tokens in der Datenbank,







um alle Verknüpfungen mit der abgemeldeten Sitzung zu entfernen. Diese Maßnahmen gewährleisten eine sichere und vollständige Abmeldung des Benutzers.

**DashboardServlet:** Das Dashboard Servlet ist für die Hauptseite der Anwendung verantwortlich. Es übernimmt die Berechnung von Statistiken und verarbeitet Anfragen zum Speichern und Löschen von Einträgen (Entries). Insbesondere erstellt das Dashboard Servlet automatisch einen Datensatz in der Tabelle "Days", wenn ein Eintrag an einem neuen Tag hinzugefügt wird. Umgekehrt wird der entsprechende Datensatz gelöscht, wenn der letzte Eintrag eines Tages entfernt wird.

```
dashboardServlet
doDelete(HttpServletRequest, HttpServletResponse) void
doGet(HttpServletRequest, HttpServletResponse) void
doPost(HttpServletRequest, HttpServletResponse) void
```

**CalendarServlet:** Das CalendarServlet übernimmt die Verwaltung von Kalenderdaten und bietet Funktionen wie das Erstellen, Bearbeiten und Löschen von Tagen und Einträgen im Kalender. Zur Kommunikation zwischen dem Server und dem Kalender im Frontend nutzt das Servlet die Google Gson-Bibliothek. Hier werden auch Validierungen durchgeführt, beispielsweise wird überprüft, ob ausreichend Urlaubstage vorhanden sind. Der Sonntag wird blockiert, und der Wochenstundenfortschritt wird regelmäßig aktualisiert, um die Benutzererfahrung zu optimieren.

```
calendarServlet
doGet(HttpServletRequest, HttpServletResponse) void
doDelete(HttpServletRequest, HttpServletResponse) void
doPost(HttpServletRequest, HttpServletResponse) void
```

**MessageServlet:** Das MessageServlet ist für die Verwaltung von Benachrichtigungen zuständig. Es ruft alle Benachrichtigungen für den jeweiligen Mitarbeiter ab und präsentiert sie auf der Benachrichtigungsseite. Dabei werden alle Nachrichten als "gesehen" markiert. Das Servlet verfügt auch über eine PUT-Methode, die aufgerufen wird, wenn das Glockensymbol in der Navigationsleiste geklickt wird. Diese Methode setzt alle Benachrichtigungen des Mitarbeiters auf "gesehen". Diese Funktionalität wird von verschiedenen Seiten genutzt, um sicherzustellen, dass Benutzer beim Klicken auf das Glockensymbol die Nachrichten als gesehen markieren können.

```
messagesServlet
doPut(HttpServletRequest, HttpServletResponse) void
doGet(HttpServletRequest, HttpServletResponse) void
```

**ProfileServlet:** Das Profilservlet ist für die Verwaltung von Profilbildänderungsanfragen zuständig. Es empfängt Anfragen zum Ändern des Profilbilds, benennt das aktuelle Profilbild um und speichert es im Ordner `webapp/resources/img/avatars/`. Das Umbenennen erfolgt nach dem Muster "avatar\_MomentaneMillisekunden". Das Profilservlet löscht auch das vorherige Profilbild, um den Speicherplatz zu optimieren und sicherzustellen, dass nur das aktuelle Profilbild gespeichert ist.

```
profileServlet
doGet(HttpServletRequest, HttpServletResponse) void
doPost(HttpServletRequest, HttpServletResponse) void
```

**StammdatenServlet:** Das Stammdatenservlet ergänzt das Profilservlet und ist auf die Profilleiste spezialisiert. Es reagiert auf Änderungen der eigenen Benutzerdaten und ermöglicht es den Nutzern, ihre E-Mail und Passwort sicher zu aktualisieren. Das Servlet validiert die eingegebenen Informationen, gleicht das Passwort mit der in der Datenbank gespeicherten Version ab und führt dann die entsprechenden Aktualisierungen durch.

```
stammdatenServlet
doPost(HttpServletRequest, HttpServletResponse) void
```

**ManagmentServlet:** Das Managementservlet spielt eine zentrale Rolle in der Verwaltung der Mitarbeiterdaten. Es übernimmt verschiedene

```
managmentServlet
doGet(HttpServletRequest, HttpServletResponse) void
doDelete(HttpServletRequest, HttpServletResponse) void
doPost(HttpServletRequest, HttpServletResponse) void
```



Aufgaben, darunter die Berechnung der erbrachten Wochenstunden-Prozentsätze für die Fortschrittsanzeige, das Formatieren von Daten für Tabellen, das Zählen der anwesenden Mitarbeiter sowie das Ermitteln der noch verfügbaren Personalnummern. Bei Anfragen zur Aktualisierung von Mitarbeiterdaten führt das Servlet die entsprechenden Änderungen durch, aktualisiert den Mitarbeiter in der Datenbank und löscht gegebenenfalls auch die zugehörigen Logindaten. Auf Anfrage kann das Managementservlet auch einen Mitarbeiter und alle damit verbundenen Daten, einschließlich des Profilbilds, löschen. Es filtert zudem heraus, ob ein Benutzer keinen Adminstatus hat, und leitet ihn gegebenenfalls zurück zur Hauptseite, um den Zugriff auf administrative Funktionen zu beschränken.

### **AddMitarbeiterServlet:** Das

AddMitarbeiterServlet ergänzt das

Managementservlet, indem es die Funktionalität

zum Hinzufügen neuer Mitarbeiter bereitstellt. Bei Anfragen zur Hinzufügung eines neuen Mitarbeiters verarbeitet das Servlet die übermittelten Informationen, verschlüsselt das Passwort und fügt dann einen neuen Mitarbeiter mit den entsprechenden Details in die Datenbank ein.

```
addMitarbeiterServlet  
doPost(HttpServletRequest, HttpServletResponse) void
```

### **GetDaysServlet:** Das GetDaysServlet ist eine Ergänzung zum Managementservlet und erweitert

dessen Funktionalität durch das Abrufen und

Bereitstellen der Kalenderdaten eines Mitarbeiters. Bei Anfragen dieses Servlets werden die relevanten Informationen zu den Arbeitstagen des Mitarbeiters aus der Datenbank abgerufen und mithilfe der Google Gson-Bibliothek in einen String umgewandelt. Dieser String dient als Datenquelle für die JavaScript-Bibliothek, die den Kalender auf der Benutzeroberfläche mit den entsprechenden Informationen füllt. Durch diese Erweiterung ermöglicht das GetDaysServlet eine dynamische und interaktive Anzeige der Kalenderdaten auf der Website.

```
getDaysServlet  
doGet(HttpServletRequest, HttpServletResponse) void
```

### **ImpressumServlet:** Das ImpressumServlet ist

für die Anzeige der Impressumsseite auf der

Website verantwortlich. Bei Anfragen zu dieser Seite lädt das Servlet die entsprechenden Inhalte und stellt sie für die Benutzeranzeige bereit.

```
impressumServlet  
doGet(HttpServletRequest, HttpServletResponse) void
```

### **DatenschutzServlet:** Das DatenschutzServlet

ist für die Anzeige der Datenschutzseite auf der

Website verantwortlich. Bei Anfragen zu dieser

Seite lädt das Servlet die entsprechenden Inhalte und stellt sie für die Benutzeranzeige bereit.

```
datenschutzServlet  
doGet(HttpServletRequest, HttpServletResponse) void
```

## 6.3.4 Hilfsklassen

In diesem Abschnitt werden verschiedene Hilfsklassen vorgestellt, die innerhalb des TrackIn-Projekts verwendet werden, um zusätzliche Funktionalitäten bereitzustellen und repetitive Aufgaben zu automatisieren. Alle diese Klassen befinden sich unter `java/org.example`. Im Folgenden werden fünf dieser Hilfsklassen im Detail erläutert, um einen umfassenden Einblick in ihre Verwendung und Bedeutung zu bieten.

**ServletUtil:** ServletUtil ist eine Hilfsklasse, die innerhalb des TrackIn-Projekts verwendet wird, um verschiedene Funktionalitäten zu unterstützen und repetitive Aufgaben zu automatisieren. Diese Klasse stellt Methoden bereit,

```
ServletUtil  
getCurrentEntriesForDashboard(Mitarbeiter, List<Days>, HttpSession) ies? List<Entries>  
createEntriesForDay(Days, String, String, List<Entries>) List<Entries>  
checkSessionAndRedirect(HttpServletRequest, HttpServletResponse) boolean  
createDefaultEntry(Days, String, String) Entries  
getDailyStatsList(List<Days>, List<Days>, List<Days>, List<Days>, List<Days>) boolean  
capitalizeFirstLetter(String) String  
isInCurrentWeek(LocalDate, LocalDate) boolean  
authenticateUser(String, String, LoginData) boolean  
createDefaultPauseEntry(Days, String) Entries  
calculateDuration(String, String) int
```

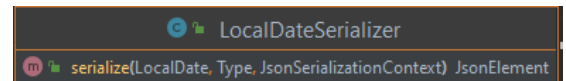


die von verschiedenen Servlets genutzt werden, um den Code zu strukturieren und Wiederverwendbarkeit zu fördern.

**LocalDateSerializer:** Die **LocalDateSerializer**-Klasse wird verwendet, um ein **LocalDate**-Objekt in ein

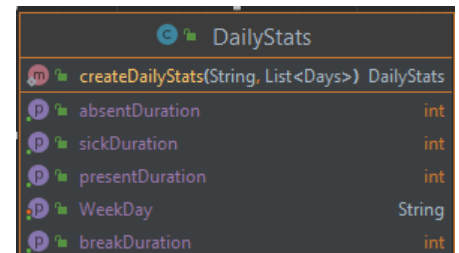
**JsonPrimitive**-Element umzuwandeln. Diese Klasse

wird in **GetDaysServlet** und im **CalendarServlet** verwendet, um Das Datum des Tages in ein JSON-Element umzuwandeln.

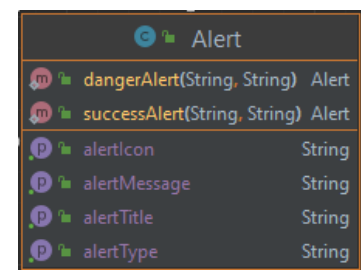


**DailyStats:** Die **DailyStats**-Klasse wird verwendet, um Statistiken für einen bestimmten Wochentag zu erstellen, indem die Gesamtdauer der verschiedenen Eintragstypen (Anwesend, Krank, Abwesend, Pause) summiert wird. Die Methode **createDailyStats** nimmt den Wochentag und die Liste der **Days**-Objekte für diesen Wochentag als Parameter.

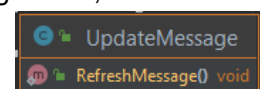
Sie durchläuft die Einträge dieser Tage und berechnet die Gesamtdauer für jeden Eintragstyp. Diese Klasse wird im **dashboardServlet** verwendet, um ein **DailyStats**-Objekt für jeden Wochentag zu erstellen und diese Informationen dann für die Anzeige von des Balken Diagrams auf der Haupt-Seite zu verwenden.



**Alert:** Die **Alert**-Klasse dient der einfachen Erzeugung von Bootstrap-Alerts in einer Java-Anwendung. Mit verschiedenen Eigenschaften wie **alertType**, **alertTitle**, **alertMessage** und **alertIcon** ermöglicht sie die Definition verschiedener Alert-Typen mit entsprechenden visuellen Darstellungen. Die Klasse enthält statische Methoden wie **dangerAlert** und **successAlert**, die Instanzen für spezifische Alert-Typen erstellen. Diese Alerts werden in der Benutzeroberfläche verwendet, um dem Benutzer wichtige Warnungen oder Erfolgsmeldungen anzuzeigen, und bieten eine übersichtliche Möglichkeit, Feedback darzustellen.



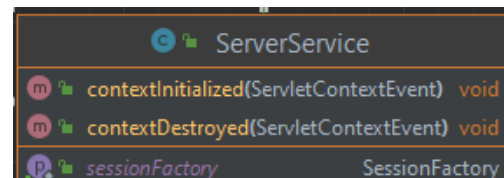
**UpdateMessage:** Die Klasse **UpdateMessage** spielt eine zentrale Rolle bei der Aktualisierung von Benachrichtigungen in der Anwendung. Sie wird aufgerufen, um die neuesten Nachrichten für jeden aktiven Benutzer abzurufen und diese Informationen in den entsprechenden Benutzersitzungen zu aktualisieren. Dabei werden verschiedene Eigenschaften der Benachrichtigungen wie Datum, Status und Symbol festgelegt. Insbesondere wird auch die Anzahl der ungelesenen Nachrichten für jeden Benutzer ermittelt und für die Anzeige in der Benutzeroberfläche vorbereitet. Diese Klasse trägt somit dazu bei, den Benutzern stets aktuelle Informationen über neue Ereignisse oder wichtige Mitteilungen bereitzustellen.



## 6.4 Hintergrundprozesse

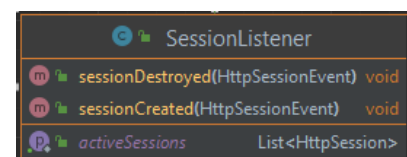
Hintergrundprozesse sind essenzielle Bestandteile der Anwendungslogik, die automatisch beim Starten der Anwendung initialisiert werden und in regelmäßigen Zeitabständen bestimmte Aufgaben ausführen. Diese Prozesse arbeiten im Hintergrund und nehmen aktiv am Lebenszyklus der Anwendung teil. In diesem Abschnitt werden verschiedene Hintergrundprozesse vorgestellt, darunter Listener, die auf bestimmte Ereignisse warten, sowie Runnable-Klassen, die in vordefinierten Intervallen bestimmte Aktionen auslösen. Diese Prozesse spielen eine entscheidende Rolle für die reibungslose Funktionalität und Aktualisierung der Anwendung im laufenden Betrieb.

**ServerService:** Die Klasse **ServerService** fungiert als **ServletContextListener**, der beim Starten der Anwendung alle erforderlichen Ressourcen initialisiert und Hintergrundprozesse einrichtet. Diese Hintergrundprozesse werden in regelmäßigen Zeitabständen ausgeführt, um bestimmte Aufgaben zu erledigen. Der **MitarbeiterRefresh**-Prozess wird alle 30 Sekunden ausgeführt, während der **MessageCreator**-Prozess alle 1 Minute aktiv ist. Beide Prozesse tragen dazu bei, die Aktualität der Mitarbeiterdaten sicherzustellen und Benachrichtigungen zu erstellen.



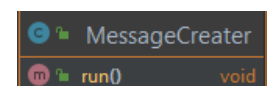
Der **ServletContextListener** stellt sicher, dass diese Prozesse automatisch gestartet werden, wenn die Anwendung bereit ist, und sie werden entsprechend beendet, wenn die Anwendung heruntergefahren wird.

**Sessionlistener:** Die Klasse **SessionListener** implementiert das **HttpSessionListener**-Interface und dient dazu, den Lebenszyklus von HTTP-Sitzungen in der Anwendung zu überwachen. Bei der Erstellung einer neuen Sitzung (Session) wird diese zur Liste der aktiven Sitzungen hinzugefügt, und beim Zerstören einer Sitzung wird sie aus der Liste entfernt. Die Liste (**activeSessions**) ist thread-sicher implementiert, um parallele Zugriffe sicher zu handhaben.



Diese Funktionalität ermöglicht es, eine Liste aller aktiven Sitzungen in der Anwendung zu pflegen. In Kombination mit anderen Komponenten wie dem **ServerService** und den Hintergrundprozessen können so bestimmte Aktionen oder Aktualisierungen für aktive Benutzersitzungen durchgeführt werden.

**MessageCreator:** Die Klasse **MessageCreator** implementiert das **Runnable**-Interface, um einen Hintergrundprozess zu repräsentieren, der in bestimmten zeitlichen Abständen Nachrichten für Mitarbeiter generiert. Dieser Prozess wird vom **ServerService** bei der Initialisierung der Anwendung gestartet.



Im **run**-Methode werden folgende Aufgaben durchgeführt:

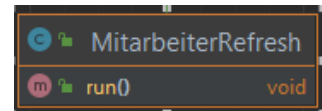
1. Löschen abgelaufener Nachrichten.
2. Abrufen aller Mitarbeiter aus der Datenbank.
3. Generierung von Warnungs- und Erfolgsmeldungen basierend auf dem Fortschritt der Wochenarbeitsstunden der Mitarbeiter und dem aktuellen Wochentag.



4. Speichern der erstellten Nachrichten in der Datenbank.
5. Aktualisierung der Benachrichtigungen für alle aktiven Sitzungen.

Der **MessageCreator** trägt somit dazu bei, Mitarbeiter und Administratoren über den Arbeitsfortschritt zu informieren und notwendige Maßnahmen anzuzeigen.

**MitarbeiterRefresh:** Die Klasse **MitarbeiterRefresh** implementiert das **Runnable**-Interface und wird in regelmäßigen Abständen vom **ServerService** gestartet. Sie ist dafür verantwortlich, den Fortschritt der Wochenarbeitsstunden für jeden Mitarbeiter zu aktualisieren. Hier sind die hervorstechenden Punkte der Implementierung:



1. Abrufen aller Mitarbeiter aus der Datenbank.
2. Iterieren durch die Arbeitstage (**Days**) jedes Mitarbeiters.
3. Für Arbeitstage in der aktuellen Woche wird der Fortschritt der Wochenarbeitsstunden aktualisiert.
4. Überprüfung, ob der Mitarbeiter heute anwesend ist und Festlegen des Anwesenheitsstatus basierend auf den Arbeitszeiteinträgen.
5. Aktualisierung der Datenbank mit den aktualisierten Informationen für jeden Mitarbeiter.

Diese Klasse sorgt dafür, dass der Wochenfortschritt für jeden Mitarbeiter stets aktuell ist, was insbesondere im **managmentServlet** wichtig ist.

## 6.5 JSP-Seiten und Web-Konfiguration

In diesem Abschnitt werfen wir einen Blick auf die grundlegende Konfiguration der Webanwendung sowie auf die Benutzeroberfläche, die durch zehn verschiedene JSP-Seiten repräsentiert wird. Die Seiten werden durch einheitliche Designelemente wie Bootstrap JS und CSS, den Kalender und spezifische Stylesheets wie **datatable.css**, **calendar.css** und **sidebar.css** harmonisch gestaltet. Mit einem klaren Fokus auf Benutzerfreundlichkeit und visuelle Ästhetik bieten diese Seiten eine einheitliche und ansprechende Erfahrung für die Benutzer. Lassen Sie uns nun detaillierter erkunden, wie diese Komponenten miteinander interagieren.

### 6.5.1 Web-Konfiguration

Die **web.xml**-Datei fungiert als zentrale Konfiguration für die Webanwendung. Hier werden wesentliche Einstellungen und Filter festgelegt, um die Anwendungsfunktionalitäten zu steuern.

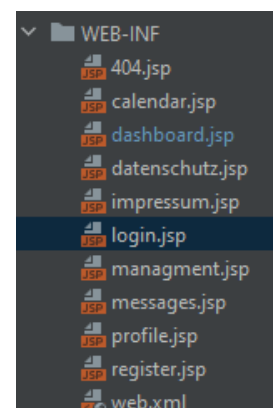
- **Filter-Mappings:** Definieren Filter für verschiedene Aufgaben wie die Überprüfung von Sitzungscookies, das Verhindern von Caching und die Steuerung des Darkmodes.
- **Session-Konfiguration:** Die Sitzungsdauer ist auf 30 Minuten festgelegt, um sicherzustellen, dass Benutzersitzungen angemessen verwaltet werden.
- **Welcome-File-List:** Legt die Standardseite fest, die angezeigt wird, wenn keine spezifische Seite angefordert wird. In unserem Fall ist es **/dashboard**.
- **Listener:** Hintergrundprozesse und Session-Überwachung werden durch Listener aktiviert. Der **ServerService** initialisiert Hintergrundprozesse beim Start, während der **SessionListener** Sitzungen überwacht.
- **Error-Page:** Bei einem 404-Fehler wird eine benutzerdefinierte Seite **/WEB-INF/404.jsp** angezeigt, um dem Benutzer eine hilfreiche Fehlermeldung zu präsentieren.

### 6.5.2 Java Server Pages

Die Java Server Pages (JSP) unserer Anwendung wurden bewusst im Verzeichnis "WEB-INF" platziert, um sicherzustellen, dass sie nicht direkt über die URL aufgerufen werden können. Dies bietet eine zusätzliche Sicherheitsebene, da Benutzer nicht direkt auf JSP-Seiten zugreifen können, sondern nur auf die von den Servlets bereitgestellten Funktionalitäten. Diese strukturierte Anordnung trägt zur Sicherheit und Integrität der Anwendung bei, indem sie einen klaren Trennungsmechanismus zwischen Präsentation und Logik schafft.

Unsere JSP-Seiten sind integraler Bestandteil unserer Webanwendung und bieten eine dynamische Benutzeroberfläche, die auf Java-Code, JavaServer Pages Standard Tag Library (JSTL) und verschiedene Frontend-Technologien basiert. Hier sind einige Schlüsselfunktionen und Techniken, die in unseren JSP-Seiten konsistent angewendet werden:

1. **Strukturiertes Design:** Alle JSP-Seiten folgen einem strukturierten Designansatz, der die Trennung von Logik und Präsentation erleichtert. Java-Code und HTML-Elemente sind sinnvoll organisiert, um die Lesbarkeit und Wartbarkeit zu verbessern.







2. **Effiziente Datenintegration:** JSP-Seiten integrieren Daten dynamisch, indem sie Java-Code in die HTML-Struktur einbinden. Die Verwendung von JSTL-Tags ermöglicht eine saubere Verarbeitung von Schleifen und Bedingungen direkt in der Seite.
3. **Bootstrap-Framework:** Alle Seiten nutzen das Bootstrap-Framework, um ein responsives und ästhetisch ansprechendes Design zu gewährleisten. Dies ermöglicht eine konsistente Darstellung auf verschiedenen Geräten und Bildschirmgrößen.
4. **Verwendung von EL-Ausdrücken:** Expression Language (EL) wird durchgängig verwendet, um auf Servlet-Attribute zuzugreifen und dynamische Daten sicher in die Seiten zu integrieren.
5. **Interaktion durch JavaScript:** JavaScript wird eingesetzt, um interaktive Funktionen bereitzustellen, beispielsweise für die Anzeige von Benachrichtigungen, das Öffnen von Modals oder das Aktualisieren von Seiteninhalten ohne vollständiges Neuladen.



## 6.6 Javascript

In unserem Webprojekt verwenden wir verschiedene JavaScript-Bibliotheken, um die Funktionalität und das Erscheinungsbild unserer Seiten zu verbessern. Hier sind die drei Hauptbibliotheken, die wir nutzen:

### 1. Bootstrap:

- **Verwendung:** Bootstrap ist eine CSS- und JavaScript-Bibliothek, die wir auf jeder Seite unserer Anwendung verwenden.
- **Zweck:** Bootstrap bietet eine Reihe von vorgefertigten UI-Komponenten, Grid-Systemen und JavaScript-Widgets, die eine konsistente und ansprechende Benutzeroberfläche ermöglichen. Wir nutzen Bootstrap, um sicherzustellen, dass unsere Seiten auf verschiedenen Geräten gut aussehen und eine benutzerfreundliche Erfahrung bieten.

### 2. Chart.js:

- **Verwendung:** Chart.js wird hauptsächlich auf der Hauptseite (Dashboard) unserer Anwendung eingesetzt.
- **Zweck:** Diese Bibliothek ermöglicht die einfache Integration von interaktiven Diagrammen und Grafiken. Auf der Hauptseite verwenden wir Chart.js, um das Balkendiagramm für die Stundenverteilung nach Wochentagen und das Donut-Diagramm für die allgemeine Stundenverteilung darzustellen. Dies verleiht dem Dashboard eine visuelle und leicht verständliche Darstellung der Arbeitszeit.

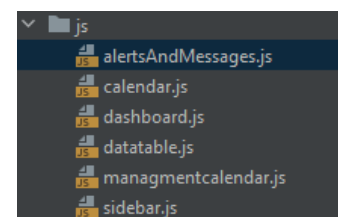
### 3. js-year-calendar:

- **Verwendung:** Die js-year-calendar-Bibliothek wird speziell auf den Seiten "managment.jsp" und "calendar.jsp" eingesetzt.
- **Zweck:** Diese Bibliothek ermöglicht die Implementierung eines interaktiven Jahreskalenders. Auf den Seiten "managment.jsp" und "calendar.jsp" nutzen wir js-year-calendar, um einen benutzerfreundlichen und ansprechenden Kalender darzustellen.

Die Kombination dieser Bibliotheken trägt dazu bei, dass unsere Webanwendung nicht nur funktional, sondern auch visuell ansprechend ist.

im Folgenden werden unsere individuell erstellten JavaScript-Skripte vorgestellt, die speziell für die Verbesserung der Benutzerinteraktion und Funktionalität auf unseren Seiten entwickelt wurden.

**AlertsAndMessage:** Das JavaScript-Skript "AlertsAndMessage" optimiert die Handhabung von Benachrichtigungen und Nachrichten auf unseren Seiten. Es konvertiert ausgewählte HTML-Elemente zu Bootstrap-Alerts und schließt diese automatisch nach 15 Sekunden. Zudem wird der Nachrichtenzähler zurückgesetzt, wenn das Nachrichten-Dropdown angeklickt wird. Die Ajax-Anfrage wird genutzt, um Nachrichten als gelesen zu markieren und verbessert die Benutzerinteraktion auf unseren Seiten. Das „AlertsAndMessage“ Skript wird auf allen Java Server Pages verwendet.







**Calendar:** Dieses JavaScript-Skript wurde für die Anpassung und Funktionalität unseres Kalenders auf der Seite "calendar.jsp" entwickelt. Es integriert das Bootstrap-Modal, um Tage zu bearbeiten oder hinzuzufügen, ermöglicht das Löschen von Tagen und bietet eine interaktive Benutzeroberfläche mit einem Kontextmenü. Darüber hinaus werden spezifische Anpassungen an Datumseingabe und -anzeige vorgenommen.

**Dashboard:** Dieses JavaScript-Skript wurde für die Funktionalität auf der Hauptseite ("dashboard.jsp") entwickelt. Es ermöglicht das Löschen von Einträgen durch Klicken auf das entsprechende Löschsymbol. Das Skript verwendet Fetch-API, um einen DELETE-Request an den Server zu senden und den Eintrag zu löschen. Zudem gibt es eine Interaktion zwischen Start- und Endzeitfeldern, um sicherzustellen, dass die Endzeit nicht vor der Startzeit liegt und umgekehrt.

**DataTable:** Diese JavaScript-Skripte wurden für die Verwaltungsseite ("management.jsp") entwickelt. Hier ist eine kurze Übersicht über die Funktionalitäten:

### 1. Suchleiste für die Tabelle:

- Ein Event-Listener überwacht die Eingabe in der Suchleiste.
- Die Funktion filtert die Tabellenzeilen basierend auf dem Suchwert und blendet unpassende Zeilen aus.

### 2. Löschen-Bestätigung mit Modal:

- Die Skripte steuern das Löschen von Einträgen.
- Durch Klicken auf das Löschsymbol wird das Löschen bestätigt und ein Fetch-Request an den Server gesendet.
- Das zugehörige Tabellenzeilen-Element wird entfernt.

### 3. Bearbeiten-Funktion mit Modal:

- Durch Klicken auf das Bearbeitungssymbol werden die Daten des ausgewählten Eintrags in ein Bearbeitungsmodal geladen.
- Das Modal zeigt Informationen zum Bearbeiten an und ermöglicht die Aktualisierung von Mitarbeiter.
- Es werden Optionen wie Position, Wochenstunden und Admin-Berechtigungen angezeigt.

### 4. Validierungsfunktion:

- Die Funktion **resetIfInvalid(e)** überprüft, ob der eingegebene Wert in einem Datalist-Element gültig ist. Falls nicht, wird das Eingabefeld zurückgesetzt.

**managmentCalendar:** Das JavaScript-Skript für die Verwaltungskalenderseite ("management.jsp") ähnelt stark dem für den allgemeinen Kalender, jedoch liegt der Fokus hier auf der Funktion zum Löschen von Ereignissen. Die Struktur und visuelle Verbesserungen bleiben erhalten, darunter die **deleteEvent(e)**-Funktion. Beim Klick auf das "Augen"-Symbol wird das Modal geöffnet und die Daten für jeden Mitarbeiter neu geladen, um eine aktualisierte Ansicht der Ereignisse anzuzeigen. Dies ermöglicht eine effiziente Verwaltung der Kalenderdaten für jeden Mitarbeiter.



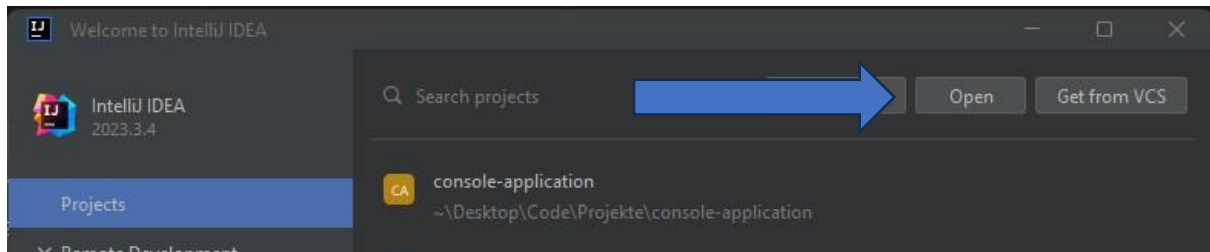
**Sidebar:** Das JavaScript-Skript für die Sidebar-Steuerung und den Dark Mode ist umfassend. Es ermöglicht das Ein- und Ausblenden der Sidebar bei Klicks auf den Toggle-Button und aktualisiert den Dark Mode basierend auf den Benutzervorlieben, die in Cookies gespeichert sind. Der Code initialisiert den Dark Mode entsprechend der gespeicherten Präferenz, und Event-Listener reagieren auf Benutzerinteraktionen, um die Sidebar und den Dark Mode zu aktualisieren.



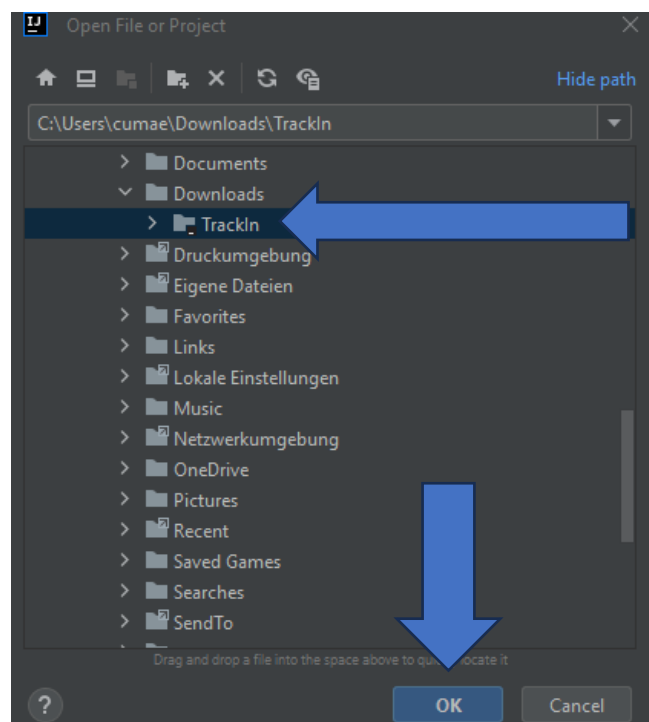
## 7.0 Installation

In diesem Abschnitt erfährst du, wie du unser Projekt erfolgreich installierst. Für die Installation nutzen wir vorzugsweise die Entwicklungsumgebung IntelliJ IDEA Ultimate 2023.3.4. Befolge die folgenden Schritte für eine reibungslose Einrichtung.

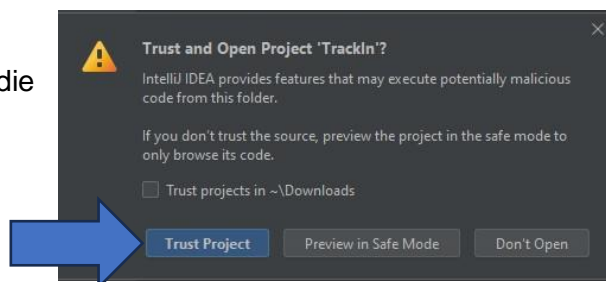
1. Starten sie das Programm IntelliJ IDEA Ultimate 2023.3.4 und Drücken anschließend auf die Schaltfläche "Open".



2. Wählen sie den Projektordner TrackIn aus und Bestätigen sie die Aktion anschließend mit der Schaltfläche "OK".

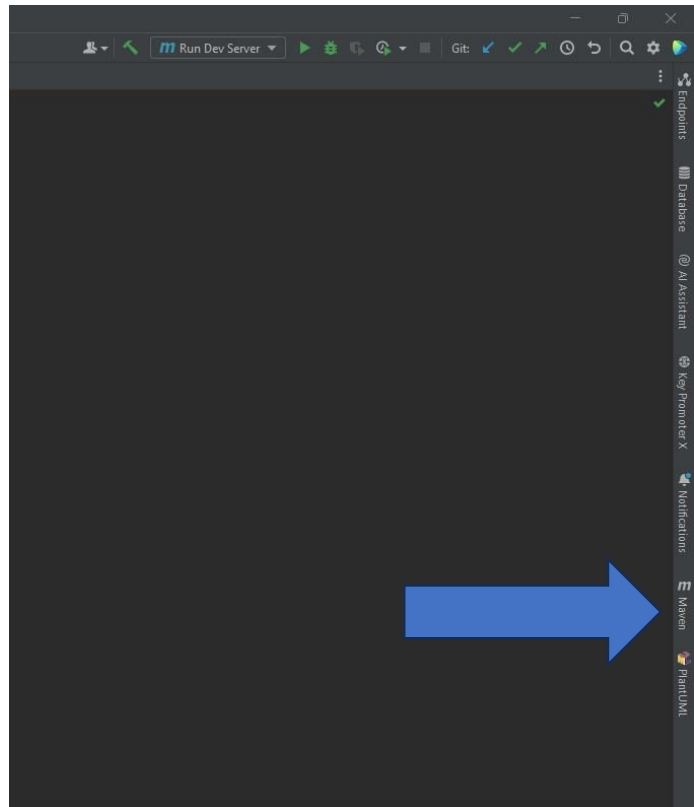


3. Bestätigen Sie, dass Sie dem Projekt vertrauen, indem Sie auf die blaue Schaltfläche 'Trust Project' drücken.

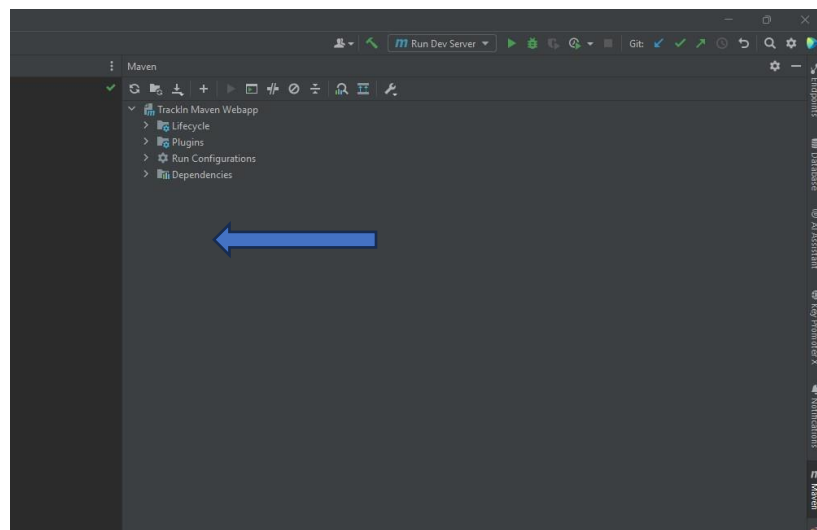




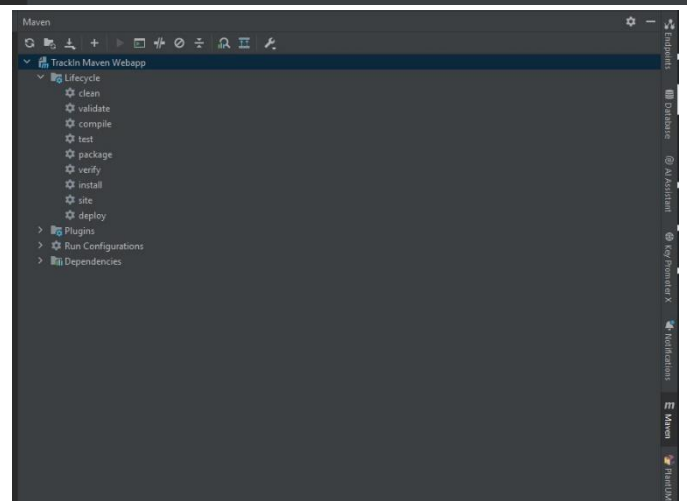
- Bitte warten Sie, bis die Entwicklungsumgebung alle erforderlichen Dateien automatisch heruntergeladen und installiert hat. Falls dieser Vorgang nicht automatisch erfolgt, finden Sie unter dem Reiter 'File -> Project Structure' die Einstellungen, um das SDK herunterzuladen. Andernfalls können Sie es manuell herunterladen (siehe dazu Kapitel [3.0 Verwendete Technologien und Planung](#)). Nach Abschluss des Ladevorgangs öffnen Sie den Reiter 'Maven' am rechten Bildschirmrand.



- Klappen Sie nun das Datei-Symbol mit dem 'm' auf. Anschließend sehen Sie 4 weitere aufklappbare Objekte. Klappen Sie nun das Objekt 'Lifecycle' auf.



- Drücken Sie anschließend mit einem Doppel-Klick auf das Feld 'Install' mit dem Zahnrad-Symbol.

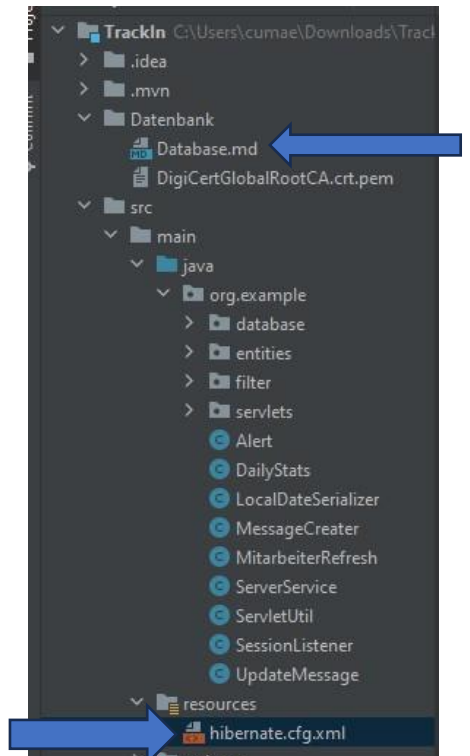




- Warten sie, bis die grüne Meldung "BUILD SUCCESS" in der Run-Console erscheint.

```
[INFO] Installing C:\Users\cumae\Downloads\trackin\target\trackin-1.0-SNAPSHOT.war to C:\Users\cumae\.m2\repository\org\example\trackin\1.0-SNAPSHOT\trackin-1.0-SNAPSHOT.war
[INFO] Installing C:\Users\cumae\Downloads\trackin\pom.xml to C:\Users\cumae\.m2\repository\org\example\trackin\1.0-SNAPSHOT\trackin-1.0-SNAPSHOT.pom
[INFO] BUILD SUCCESS
[INFO] Total time: 9.208 s
[INFO] Finished at: 2024-02-27T21:40:15+01:00
[INFO] Process finished with exit code 0
```

- Falls Sie die Datenbank nicht ändern möchten, springen Sie zu Schritt 11. Wenn Sie aus bestimmten Gründen nicht die bereits konfigurierte Datenbank verwenden möchten, können Sie diese selbst installieren und konfigurieren. Dazu finden Sie unter dem Ordner 'Datenbank' das Installations-Skript.
- Um die Installierte Datenbank mit dem Programm zu verknüpfen, öffnen sie die Hibernate Konfigurationsdatei, diese finden sie unter `src/main/webapp/resources/hibernate.cfg.xml`
- Anschließend geben sie folgende Befehle in ihre Datenbank ein:



```
INSERT INTO trackindatabase.mitarbeiter
(personalNummer, name, vorname, geburtsdatum,
onetimepassword, admin, profile_picture, einstellungsdatum, position,
Wochenstunden, WochenStundenFortschritt, verbleibendeUrlaubstage, present)
VALUES (1, 'Doe', 'John', '2005-06-09',
'$2a$12$9LPu5j5NOjsK6.ns.TnGHuyQjA5XIO1f411hqlbc8AvMSZJHmegYi', 1,
'../resources/img/avatars/default.jpeg', '2023-02-13', 'Geschäftsführer', 40, 20, 25,
0);
```

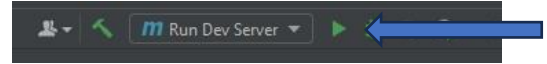
```
INSERT INTO trackindatabase.logindata (credentials_id, email, passwort,
Mitarbeiter_personalNummer) VALUES (66, 'john.doe@mail.com',
'$2a$12$uu8MD6NbJnckh9fUI6tEluPs8SrnbsGnz/x8RuIFYLXi0OwKgxa', 1);
```



11. Ersetzen Sie nun die folgenden drei Zeilen durch die Daten Ihrer Datenbank.

```
<property name="hibernate.connection.url">jdbc:mysql://trackincuma.mysql.database.azure.com:3306/trackindatabase?useSSL=true&requireSSL=true&verifyServerCertificate=false&serverSslCert=TrackIn@fos\Datenbank\DiglCertGlobalRootCA.crt.pem</property>
<property name="hibernate.connection.username">admincuma</property>
<property name="hibernate.connection.password">89#xyp$2zLsiv7*</property>
```

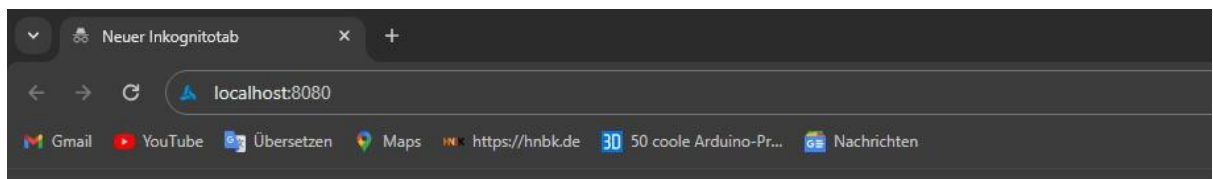
12. Um die Webapplikation zu starten, klicken Sie nun auf den grünen Pfeil rechts neben der Schaltfläche 'Run Dev Server'.



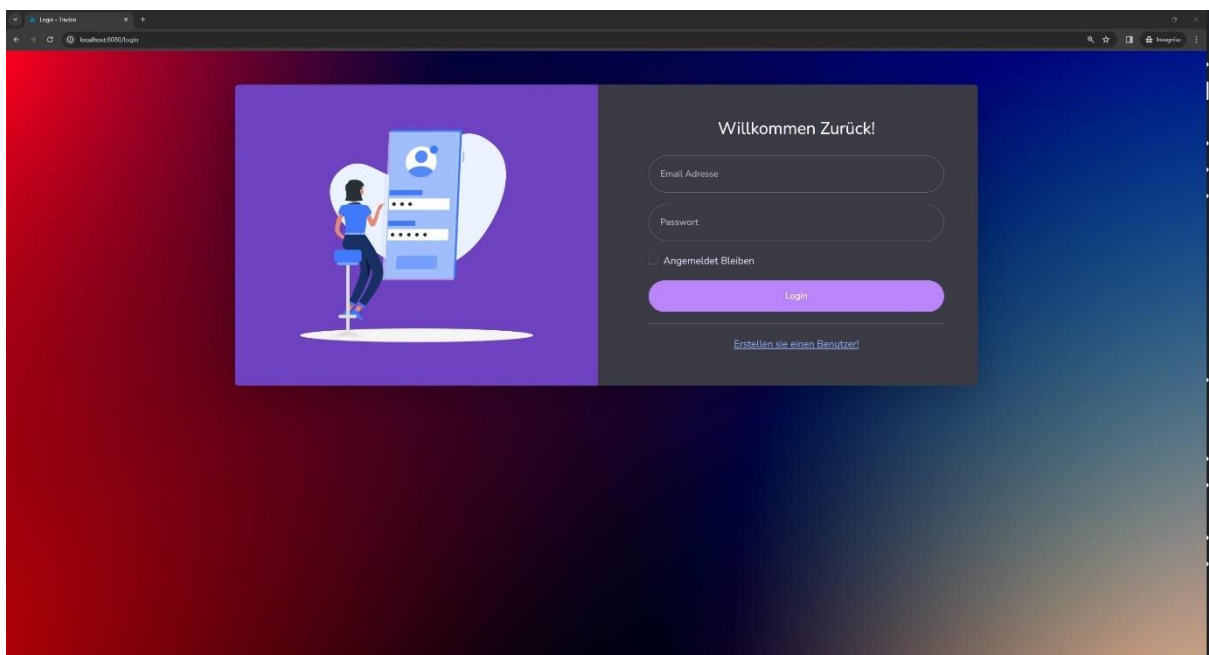
13. Warten Sie nun, bis die folgende Meldung erscheint:

```
[INFO] Started ServerConnector@55731ccd{HTTP/1.1, (http/1.1)}{0.0.0.0:8080}
[INFO] Started Server@2bfb583b{STARTING}[11.0.15,sto=0] @7398ms
[INFO] Scan interval sec = 2
```

14. Anschließend öffnen Sie Ihren Browser, wir empfehlen Google Chrome, und geben Sie folgendes in die URL-Leiste ein: **localhost:8080**.



15. Falls Sie die folgende Seite sehen, haben Sie die Webanwendung erfolgreich installiert. Sie können sich nun mit dem Administrator Konto anmelden  
E-Mail: [john.doe@mail.com](mailto:john.doe@mail.com)  
Passwort: adminroot





### 8.0 Schlusswort

In der Welt der Webentwicklung haben wir durch dieses Projekt eine Menge dazugelernt. Zugegeben, der Start war etwas holprig, und eine bessere Planung hätte uns sicherlich geholfen. Aber hey, Learning by Doing war auch nicht schlecht! Die Praxis hat uns gezeigt, dass nur Theorie manchmal ziemlich langweilig sein kann. Dennoch hatten wir jede Menge Spaß dabei. Alles in allem war es eine großartige Erfahrung, die uns für kommende Webprojekte gut gerüstet hat.