

Arbeitszeittracker Web-App

Eine interaktive **Arbeitszeittracker**-Web-App, die gesetzeskonforme Tageserfassungen mit mehreren Arbeits- und Pausenblöcken unterstützt. Die Anwendung läuft komplett im Browser – keine Server-Installation notwendig.



Features

- **Monate & Jahresüberblick:** Startet im Jahresüberblick des aktuellen Jahres und bietet Summen-Widgets für Zeitkonten sowie Warn-Overlays bei Auffälligkeiten. Ein Wechsel in den Bereich „Monate“ ist jederzeit möglich.
- **Mehrarbeit-Monitoring:** Ein eigener Menüpunkt „Mehrarbeit“ visualisiert den kumulierten Wochenverlauf des Arbeitszeitkontos als Chart.js-Liniendiagramm inklusive Vorjahressaldo und zeigt die in den Einstellungen hinterlegten Grenzen als gestrichelte Referenzlinien.
- **Setup-Assistent:** Ein modaler Wizard fragt beim ersten Start (oder auf Wunsch erneut) zunächst die Sprache sowie zentrale Parameter ab – Bundesland, Sollarbeitszeit, Vorjahressaldo, Resturlaub und den Projektstunden-Schalter. Die Sprachauswahl aktualisiert alle Labels unmittelbar beim Wechsel, die Eingaben werden in den Einstellungen gespiegelt und automatisch in Export/Import übernommen; die Aktivierung von Beispieldaten erfolgt ausschließlich über die Einstellungen.
- **Getrennte Einstellungsbereiche:** Die Ansicht „Einstellungen“ teilt dauerhafte Basiseinstellungen (Arbeits-/Pausenvorgaben, Wochenenden, Mehrarbeitsgrenzen, Bundesland, Urlaubstage, Intro-, Beispiel- und Projekt-Toggles) von jahresbezogenen Überträgen (Vorjahressaldo, Resturlaub) mit eigenem Jahreswähler ab, der mit Monats-, Jahres- und Mehrarbeitsansicht synchronisiert ist und beim Start automatisch auf das effektive Jahr gestellt wird.
- **Segmentierte Tageserfassung:** Mehrere Arbeits-, Reise-, Pausen- und Privatblöcke pro Tag erfassen – die Blöcke bleiben in der Eingabereihenfolge, und bei Bedarf sorgt der Button „**Neu sortieren**“ für eine einmalige Chronologisierung. Arbeitsorte, Notizen und Projektauswertungen bleiben erhalten.
- **Lückenlose Zeitachsen:** Die Anwendung ergänzt keine Pausen mehr automatisch. Stattdessen analysiert „**Konflikte prüfen**“ erkannte Lücken, schlägt passende Pausen oder Kürzungen vor und setzt Änderungen nur nach ausdrücklicher Bestätigung um.
- **Typische Arbeitstage abgedeckt:** Der Assistent berücksichtigt Mittagspausen nach 6 Stunden, zusätzliche 15-Minuten-Pausen nach 9 Stunden sowie die

gesetzlichen Höchstwerte von 10 Stunden pro Tag und 48 Stunden pro Woche und bietet passende Splits und Kürzungen an – der Anwender entscheidet je Vorschlag.

- **Compliance-Hinweise & Warnungen:** Gesetzliche Mindestpausen (inkl. Mindestlänge von 15 Minuten), Pausenlage vor 6/9 Stunden, die Tages- (10 Stunden) und Wochenhöchstgrenze (48 Stunden) sowie die 11-stündige Ruhezeit werden geprüft. Abweichungen erscheinen als Hinweise unter den Aktionsbuttons und können über den Konflikt-Dialog gezielt gelöst werden; verbleibende Verstöße blockieren das Speichern und erzeugen Warn-Icons in Kalenderansichten. Kritische Meldungen werden farbig hervorgehoben, die Warnfarbe lässt sich im Theme-Editor individuell anpassen.
- **Projektstundenabgleich & Visualisierung:** Validierung, dass Projektzeiten die dokumentierten Arbeitsstunden nicht überschreiten, plus gestapeltes Chart.js-Balkendiagramm mit Monatsachse und Projektstunden pro Monat.
- **Projektarchivierung statt Löschung:** Projekte mit bereits verbuchten Stunden lassen sich nicht mehr löschen; ein Lösversuch setzt den Status automatisch auf inaktiv, damit Historien erhalten bleiben.
- **Farbcodierte Segmente & Themes:** Arbeit, Dienstfahrten, Pausen und Privatblöcke erhalten eigene Hintergrundfarben, die im Menüpunkt „Themes“ angepasst, dupliziert sowie als JSON exportiert/importiert werden können.
- **Eigene Home-Office-Palette:** Arbeit und Home Office lassen sich in den Theme-Karten getrennt einfärben und mit individuellen Icons versehen.
- **Icon-Steuerung pro Theme:** Über den Theme-Schalter „Icons anzeigen“ lassen sich alle Emoji-Badges der Monats- und Jahresübersicht je Theme aktivieren oder vollständig ausblenden; die Einstellung wird in Presets, JSON-Exporten und Excel übernommen.
- **Skalierung & Schriftgrößen je Theme:** Themes enthalten nun getrennte Regler für Icon-Skalierung (Monat/Jahr) und die Schriftgröße der Tageszahlen. Standardwerte (100 %, 1,15 rem bzw. 0,95 rem) werden beim Export nach JSON/Excel mitgespeichert und bei Importen sofort angewendet.
- **Responsives Vollbild-Layout:** Navigation, Topbar-Kennzahlen und Monats-Tabs spannen sich über die gesamte Seitenbreite, passen sich an große Screens an und brechen auf kleineren Viewports übersichtlich um.
- **Mobile Navigation & Projektkarten:** Unter 768 px blendet ein Burger-Button die Navigation auf Wunsch ein, Topbar-Steuerelemente stapeln sich automatisch

und Projekttabellen werden als Kartenliste ohne horizontales Scrollen dargestellt.

- **Mittagspausen-Defaults:** Einstellbarer Beginn und Dauer der Standard-Mittagspause – wird bei Dummy-Daten sowie Export/Import berücksichtigt.
- **Feiertags-Highlighting:** Feiertage, Urlaubs- und Krankheitstage mit konfigurierbaren Farben hervorheben und – sobald Daten vorliegen – im Tagesdialog-Kopf inklusive Icon und Feiertags-/Ferientext anzeigen.
- **Dark-/Light-Mode:** Umschaltbar über Einstellungen *oder* direkt in der Navbar.
- **Mehrsprachigkeit:** Oberfläche kann per Umschalter zwischen Deutsch und Englisch wechseln.
- **Tutorial-Modus:** Statische, übersetzte Schritt-für-Schritt-Anleitung direkt in der App.
- **Lokale Speicherung & Export:** Datensätze werden im localStorage gespeichert und können inkl. Segmentdetails nach Excel exportiert bzw. importiert werden.
- **Offline verfügbare XLSX-Bibliothek:** Die Anwendung bringt xlsx.full.min.js nun direkt unter lib/xlsx/mit, sodass Import und Export vollständig ohne Internetverbindung funktionieren.
- **Offline verfügbare Ferien- und Feiertagsdaten:** Für alle Bundesländer liegen JSON-Dateien unter data/holidays/<Jahr>/<Bundesland>.json vor (Zeitraum 2024–2030). Die App greift vorrangig auf diese Dateien zurück und benötigt nur bei fehlenden oder veralteten Datensätzen eine Online-Verbindung.
- **Beenden-Hinweis:** Beim Schließen des Browserfensters fragt die App nach einer schnellen lokalen Excel-Sicherung und legt sie direkt im Download-Ordner ab.
- **App-interne Dialoge statt Browser-Popups:** Ein zentraler Dialog-Service (js/modules/dialog-service.js) rendert zugängliche Overlays, verwaltet eine Dialog-Warteschlange samt Fokusfalle und erlaubt automatisierte Standardantworten für Tests. Native Browser-Dialoge bleiben lediglich als Fallback für beforeunload-Prompts aktiv. In Headless-/WebDriver-Läufen erscheinen die Overlays standardmäßig sichtbar; nur explizite Automation-Einstellungen (z. B. dialogService.configureAutomation) beantworten Dialoge automatisch. Für den reinen Datei-Workflow bindet index.html zusätzlich js/dialog-service.legacy.js als klassisches defer-Script ein, sodass der Overlay-Dialog auch ohne ES-Module aktiv bleibt.
- **Release-Fußzeile:** Die Fußzeile zeigt die aktuelle Versionsnummer im Format der Pull-Request-ID (Fallback: PR-DEV).

- **Projektansicht per Schalter:** Über den Toggle „Projektdate erfassen“ in den Einstellungen lassen sich Navigationseintrag, Monatszusammenfassung und Projektabschnitt im Tagesdialog ausblenden – bestehende Projektzuordnungen bleiben erhalten und werden weiterhin exportiert.
- **Schnellzugriff auf Tagesarten:** In der Monats- und Jahres-Topbar öffnen drei Emoji-Schaltflächen (🕒 / 🏖️ / 🤒) schlanke Dialoge für Arbeitstage, Urlaubs- und Krankheitseinträge. Arbeitszeiten übernehmen automatisch Standardpausen und erlauben optional eine Projektzuordnung; nach dem Speichern aktualisieren sich Monat, Jahressummen und Tabs sofort.

Mehrarbeit überwachen

- Das Wochenlinien-Diagramm zeigt den Startwert aus dem Vorjahr sowie alle Folgeweche als kumulierten Verlauf (positiv wie negativ) und spannt die X-Achse lückenlos über alle Kalenderwochen des aktuellen Jahres, auch wenn einzelne Wochen ohne Buchungen bleiben.
- Einträge, die die hinterlegten Limits überschreiten würden, werden bereits beim Speichern abgeblockt und mit einer verständlichen Fehlermeldung versehen.
- Die Grenzwerte (Maximum/Minimum) lassen sich in den allgemeinen Einstellungen als Stundenwerte festlegen. Sie werden in Excel-Exporten/-Importen mitgeführt und wirken sich sofort auf das Diagramm aus.
- Änderungen an Jahr, Vorjahressaldo oder importierten Daten aktualisieren das Diagramm automatisch.

Schnellerfassung nutzen

1. Öffne die Monats- oder Jahresansicht – in beiden Topbars findest du rechts neben den Import/Export-Aktionen eine runde Emoji-Schaltflächengruppe.
2. Wähle die passende Aktion:
 - 🕒 **Arbeitstag erfassen** – Datum sowie Start/Ende vorausgefüllt, Arbeitsort und optionales Projekt wählbar; Pausen werden automatisch berechnet.
 - 🏖️ **Urlaub eintragen** – Start-/Enddatum wählen; der Wizard erzeugt Einträge für jeden Werktag im Bereich, überspringt Wochenenden, lässt vorhandene Tagesdaten unverändert und hängt bei Kollisionen automatisch zusätzliche Werktage an, bis die angefragte Anzahl erreicht ist.

- 🤒 **Krankheit erfassen** – identisches Verhalten wie beim Urlaub: nur Werktage werden ergänzt, bestehende Einträge bleiben bestehen und blockierte Tage werden durch spätere Werktage ersetzt.
3. Speichere den Dialog. Die History wird strukturiert aktualisiert, laufende Projektauswertungen sowie Monats- und Jahreszahlen werden automatisch neu gerendert.

Hinweis: Wird das Projekt-Feature nachträglich aktiviert oder deaktiviert, synchronisiert der Arbeits-Wizard seine Projektauswahl sofort mit dem aktuellen Datum.

Tipp: Die Wizards respektieren deine Einstellungen aus dem Setup-Assistenten (Sollzeit, Pausenlänge/-beginn, Projektfeature). Passe die Defaults dort an, wenn du andere Vorschlagswerte brauchst.

⚠️ Tageswarnungen & Kacheldesign

- Tageskacheln zeigen die Tagesnummer zentriert, Status-Icons (z. B. Arbeit, Home Office, Urlaub) oberhalb der Zahl und – bei Regelverstößen – ein Warnsymbol direkt über der Zahl; die Icons lassen sich pro Theme komplett deaktivieren.
- Bei erkannten Problemen – etwa fehlenden Pausen, nicht zusammenhängenden Segmenten oder Ruhezeitverletzungen – erscheint ein Warn-Icon inkl. Tooltip mit den Details.
- Die Warnsymbole lassen sich je Theme konfigurieren (Icon: Warnhinweis).

🎨 Theme-Verwaltung & JSON-Workflow

- Themes können gespeichert, dupliziert, als JSON exportiert und wieder importiert werden (Theme speichern, Theme duplizieren, Theme speichern (JSON), Theme laden (JSON)).
- Segmentfarben, Popup-Hintergrund/Rahmen und Tageskachel-Icons werden im Theme gespeichert; beim JSON-Export werden diese Werte außerhalb der Excel-Datei gesichert.
- Die Icon-Auswahl für Schulferien (Icon: Ferien) enthält zusätzlich das Buch-Emoji 📖, damit kuratierte Themes unverändert exportiert/importiert werden können; zudem stehen für Arbeit/Urlaub/Krankheit/Feiertag/GLAZ/Warnung alle kuratierten Emojis einschließlich 🕒 / 🌐 / 🕒 und 🚗 bereit, sodass Exporte keine Fallbacks mehr verwenden.
- In der Navigationsleiste befindet sich ein zusätzliches Theme-Dropdown (Theme), das synchron zum Bereich „Themes“ arbeitet und beim Umschalten sofort Primärfarbe, Buttons sowie Dark-/Light-Modus aktualisiert.

- Beim Speichern eines Themes wird der gewählte Identifier parallel in themeActiveId und im Legacy-Schlüssel selectedTheme hinterlegt, damit ältere Browserdaten weiterhin erkannt und übernommen werden.

Projektanalyse

- Unterhalb der Projektliste visualisiert ein gestapeltes Chart.js-Balkendiagramm die monatlichen Projektstunden des ausgewählten Jahres, zeigt dabei immer alle zwölf Monate dieses Jahres und blendet Buchungen anderer Jahre konsequent aus.
- Tooltips und Achsenlabels formatieren alle Werte konsistent als HH:MM, die Legende erzeugt Chart.js automatisch und passt sich den Projektnamen an.
- Das Diagramm aktualisiert sich automatisch nach Änderungen an Tages- oder Projekteingaben sowie nach einem Wechsel des betrachteten Jahres.

Mittagspausen & Dummy-Daten

- Die Einstellungen enthalten neben der Pausendauer nun auch eine Startzeit der regulären Mittagspause.
- Diese Werte fließen in die Excel-Exporte/-Importe sowie die Dummy-Daten-Erzeugung ein (Standard: 12:00 Uhr, 60 Minuten).
- Der Konflikt-Assistent übernimmt Beginn und Dauer der Mittagspause exakt gemäß Einstellungen, fügt sie als durchgehenden Block ein und ergänzt zusätzliche Pausen nur noch, wenn gesetzliche Mindestzeiten darüber hinaus fehlen.

Modaler Tagesdialog

- Der Tagesdialog blendet den Hintergrund weich („blur“) aus und lässt sich ausschließlich über Löschen, Abbrechen oder OK schließen.
- Spaltenüberschriften für Start, Ende, Arbeitsort und Art sitzen im Tabellenkopf; innerhalb der Zeilen befinden sich Screenreader-Labels.

Dialog-Service & Automatisierung

- index.html enthält einen globalen Dialog-Container (#appDialogHost) mit role="dialog", Fokusfalle, ARIA-Verkettung und Overlay-Hintergrund. Das Styling liegt in css/dialog.css und sorgt für einheitliche Abstände, animierte Überblendung sowie eindeutige Statusfarben für Warnungen/Erfolge.
- js/modules/dialog-service.js exportiert einen Singleton (dialogService), registriert ihn zusätzlich auf window.dialogService und stellt Promise-basierte Methoden alert, confirm, open bereit. Die Warteschlange garantiert, dass immer nur ein Dialog sichtbar ist; nach Bestätigung wandert der Fokus zurück zum

auslösenden Element. Für Browser, die das Modul nicht ausführen (z. B. beim Öffnen via `file://` mit deaktivierter Modulunterstützung), sorgt `js/dialog-service.legacy.js` für dieselbe Registrierung auf Basis eines klassischen Skripts.

- Fachmodule rufen die Dienstfunktionen über `showDialogAlert/showDialogConfirm` aus `js/sections/utils.js` auf. Tests können mit `dialogService.configureAutomation({ confirm: true, alert: true })` automatische Antworten setzen oder via `resetAutomation()` den Originalzustand wiederherstellen.
- Das Exit-Backup (`js/sections/exit-backup.js`) nutzt weiter den nativen `beforeunload`-Confirm (Vorgabe der Browser). Die anschließenden Hinweis-Texte erscheinen jedoch im neuen Dialog, sofern dieser verfügbar ist; andernfalls greift weiterhin `window.alert` als Fallback.

Projektstruktur

- **index.html** Hauptseite mit Navigation (Menüpunkte „Monate“, „Jahresüberblick“, „Projekte“, „Einstellungen“, „Themes“, „Tutorial“), Monatskalender, Modal zur Tagesbearbeitung und Compliance-Widgets.
- **css/** Stylesheets für Modal, Layout und responsive Verhalten.
- **js/sections/** Modular aufgebaute JavaScript-Dateien (z. B. `modal.js`, `calculations.js`, `export.js`) für Erfassung, Validierung, Export und Rendering.
- **js/bootstrap/** Bootstrap-Skripte, die die Initialisierungsfunktionen (`initApp`, `initLanguage`, `initSettings`, `initialRender`) auf `globalThis` registrieren. `js/sections/init.js` ruft `initApp()` beim `DOMContentLoaded`-Event auf und funktioniert dadurch sowohl beim direkten Öffnen der `index.html` als auch in ausgelieferten Builds.
- **tests/bootstrap/** Vitest-Suite für das Bootstrapping; verifiziert Spracheinstellung, Default-Settings, Render-Pipeline und Orchestrierung des Einstiegs. (nur für die Entwicklung verfügbar)