

# Arbeitszeiterfassung mit SQLite-Datenbank

---







## Überblick

---

Diese professionelle Arbeitszeiterfassung nutzt eine **SQLite-Datenbank** für sichere und zuverlässige Datenspeicherung. Die Anwendung läuft lokal auf Ihrem Computer und bietet alle Funktionen der ursprünglichen Excel-Tabelle in einer modernen Web-Oberfläche.

## Vorteile der SQLite-Version

---

-  **Sichere Datenspeicherung** - Keine Datenverluste durch Browser-Cache
-  **Professionelle Datenbank** - Strukturierte, konsistente Daten
-  **Bessere Performance** - Schnellere Abfragen und Berechnungen
-  **Backup-fähig** - Einfache Sicherung der `zeiterfassung.db` Datei
-  **Mehrbenutzerfähig** - Vorbereitet für Netzwerk-Zugriff
-  **Skalierbar** - Kann später erweitert werden

## Systemvoraussetzungen

---

### Windows:

- **Python 3.7+** (Download: <https://python.org>)
- Webbrowser (Chrome, Firefox, Edge, Safari)

### macOS:

- **Python 3.7+** (meist vorinstalliert oder via Homebrew)

- Webbrowser (Chrome, Firefox, Safari)

## Linux:

- Python 3.7+ und pip3
- Webbrowser (Chrome, Firefox)



## Installation & Start

---

### Einfacher Start:

1. **Ordner entpacken** an gewünschten Ort
2. **Doppelklick** auf das entsprechende Startscript:
3. **Windows:** `start.bat`
4. **macOS/Linux:** `start.sh`
5. **Fertig!** Browser öffnet automatisch

### Manueller Start:

```
# Terminal/Kommandozeile öffnen
cd pfad/zum/zeiterfassung-ordner

# Abhängigkeiten installieren (einmalig)
pip install flask flask-cors

# Server starten
python server.py

# Browser öffnen: http://localhost:5000
```



## Dateien im Paket

---

```
zeiterfassung-sqlite/
├── server.py          # Python-Backend mit SQLite
├── index.html         # Web-Frontend
├── app.js             # JavaScript-Logik
├── start.bat          # Windows-Startscript
├── start.sh           # macOS/Linux-Startscript
├── README.md          # Diese Anleitung
└── zeiterfassung.db   # SQLite-Datenbank (wird automatisch erstellt)
```

## 1. Dashboard

- Übersicht aller Mitarbeiter
- Aktuelle Statistiken
- Schnellzugriff auf wichtige Daten

## 2. Zeiterfassung

- **Mitarbeiter auswählen** aus Dropdown
- **Monat/Jahr** einstellen
- **Kalenderansicht** wie in Excel
- **Klick auf Tag** öffnet Eingabedialog

## 3. Eingabedialog (vergrößert)

- **Arbeitszeit:** Start/Ende/Pause
- **Urlaub/Krank:** Überschreibt andere Einträge
- **Duftreisen:** Bis 18h / Ab 18h (mehrfach möglich)
- **Provision:** Wird angezeigt (später automatisch berechnet)
- **Notizen:** Zusätzliche Informationen

## 4. Besonderheiten

- **Samstag ist Arbeitstag** (Einzelhandel)
- **Duftreisen unabhängig** von Arbeitszeit
- **Urlaub/Krank** macht andere Einträge irrelevant
- **Automatische Speicherung** in SQLite-Datenbank



## Erweiterte Funktionen

---

### Datenbank-Backup:

```
# Einfach die Datei kopieren  
cp zeiterfassung.db zeiterfassung_backup_2025-06-23.db
```

### Datenbank-Wiederherstellung:

```
# Backup-Datei zurückkopieren  
cp zeiterfassung_backup_2025-06-23.db zeiterfassung.db
```

### Netzwerk-Zugriff (optional):

In `server.py` ändern:

```
app.run(host='0.0.0.0', port=5000, debug=False)
```

Dann von anderen PCs erreichbar unter: `http://[IP-ADRESSE]:5000`

## 🔧 Problemlösung

---

### "Python nicht gefunden"

- Python von <https://python.org> installieren
- Bei Installation "Add to PATH" aktivieren

### "Port bereits belegt"

- In `server.py` Port ändern: `port=5001`
- In `app.js` `API_BASE_URL` anpassen: `http://localhost:5001/api`

### "Keine Verbindung zum Server"

- Prüfen ob `server.py` läuft

- Browser-Cache leeren (Strg+F5)
- Firewall/Antivirus prüfen

## Datenbank-Probleme

- `zeiterfassung.db` löschen (wird neu erstellt)
- Backup einspielen falls vorhanden

## Support

---

Bei Problemen: 1. **Startscript verwenden** (automatische Fehlererkennung) 2. **Terminal-Ausgabe prüfen** (Fehlermeldungen) 3. **Browser-Konsole öffnen** (F12 → Console) 4. **Backup der Datenbank erstellen** vor Änderungen

## Migration von localStorage

---

Falls Sie die alte localStorage-Version verwendet haben: 1. Alte Daten im Browser exportieren (falls möglich) 2. Neue SQLite-Version starten 3. Daten manuell neu eingeben (sicherer)

## Nächste Schritte

---

Die Anwendung ist bereit für: - **Automatische Provisionsberechnung** basierend auf Umsatz - **PDF-Export** für Lohnbuchhaltung - **Erweiterte Berichte** und Statistiken - **Netzwerk-Deployment** für mehrere Benutzer

---

**Die Arbeitszeiterfassung ist jetzt professionell und sicher mit SQLite-Datenbank!**

