

Disclaimer: Don't open app in dark mode

StudyBuddy

StudyBuddy ist eine intelligente Lernmanagement-Anwendung, entwickelt zur Unterstützung von Studierenden bei der Organisation ihres Lernalltags. Durch den Einsatz von **Künstlicher Intelligenz** und **maschinellern Lernen** hilft die App, personalisierte Lernempfehlungen zu erstellen und Prokrastination frühzeitig zu erkennen.

Installation & Ausführung

Installation

1. ZIP-Datei entpacken
2. Ins Projektverzeichnis navigieren
3. Abhängigkeiten installieren:

```
pip install -r requirements.txt
```

4. In Light Modus wechseln (3 Punkte -> Settings -> wähle App Darstellung light), falls Browser Dark Modus als Default hat.

Starten der Anwendung

```
streamlit run main.py
```

Die Anwendung ist dann erreichbar unter:

[<http://localhost:8501>]

Funktionen

- **Personalisiertes Dashboard**
Übersicht über Lernaktivitäten und bevorstehende Fristen

- **Lerntyp-Analyse**
Ermittlung des individuellen Lerntyps für optimierte Lernstrategien
 - **Kalender-Integration**
Synchronisation mit Google Calendar für effizientes Zeitmanagement
 - **Kursverwaltung**
Zentrale Verwaltung aller Kurse
 - **KI-gestützte Lernempfehlungen**
Dynamische Lernpläne basierend auf Kursinhalten und Lerntyp
 - **Prokrastinations-Risiko-Analyse**
ML-gestützte Erkennung individueller Prokrastinationsmuster
-

Technische Architektur

Frontend

- **Streamlit** – Interaktive Benutzeroberfläche

Backend

- **Python** – Programmiersprache
- **SQLite / Azure SQL** – Datenbankmanagement lokal oder in der Cloud
- **scikit-learn** – Machine Learning Framework
- **OpenAI API** – Generierung personalisierter Lernempfehlungen

Externe APIs

- **Google Calendar API** – Kalender-Synchronisation
- **OpenAI API** – KI-gestützte Lernpläne
- **HSG API** – Kursdaten und Kursmerkbblätter

Datenbank

Standardmäßig nutzt die Anwendung **SQLite** als lokale Datenbank. Eine Umschaltung auf die Cloud basierte Lösung **Azure SQL** ist in der `database_manager.py` Datei möglich. Die Konfiguration erfolgt über die Datei `.env`.

Ordnerstruktur

```
CS-Project/
├── .streamlit/                # Streamlit-Konfiguration
├── Features/                  # Kernfunktionalitäten
│   ├── .env                  # Umgebungsvariablen
│   ├── .streamlit/           # Streamlit-Konfiguration
│   ├── calendar_study.py     # Kalender-Integration
│   ├── courses.py            # Kursverwaltung
│   ├── dashboard.py          # Dashboard
│   ├── dashboard_charts.py    # Visualisierungen
│   ├── database_manager.py    # Datenbank-Management
│   ├── db.py                  # Datenbank-Initialisierung
│   ├── google_calendar_sync.py # Google Calendar-Sync
│   ├── learning_suggestions.py # KI-Lernempfehlungen
│   ├── learning_tips.py       # Lerntyp-Tipps
│   ├── learning_type.py       # Lerntyp-Analyse
│   ├── local_database.db      # Lokale SQLite-Datenbank
│   ├── procrastination_data.csv # Trainingsdaten ML
│   ├── procrastination_risk.py # Prokrastinations-Analyse
│   ├── procrastination_risk_model.joblib # Trainiertes Modell
├── main.py                    # Anwendungseinstieg
├── Procrastination_ML_Notebook_Realistic_App_Features.ipynb # ML-Dokumentation
└── requirements.txt           # Abhängigkeiten
```

Prokrastinations-Risiko-Modell

Grundlage:

Ein **Random Forest Classifier** analysiert folgende Merkmale:

- Studienjahr
- Sozioökonomischer Hintergrund
- Prüfungsvorbereitung
- Wöchentliche Lernstunden
- Notendurchschnitt
- Zeitmanagement
- Digitale Ablenkungen (z.B. Smartphone-Nutzung)
- Konzentrationsfähigkeit

Risikostufen:

- **0 - Niedrig:** Effektives Zeitmanagement, geringe Prokrastinationsneigung
- **1 - Mittel:** Gelegentliche Prokrastination mit moderaten Auswirkungen

- **2 - Hoch:** Häufige Prokrastination mit starker Leistungsbeeinträchtigung

Die vollständige Modellentwicklung ist im Jupyter Notebook dokumentiert:

`Procrastination_ML_Notebook_Realistic_App_Features.ipynb`

Hinweis zu Azure

Die Anwendung ist für **Azure SQL** vorbereitet, aktuell jedoch auf **SQLite** umgestellt. Die Reaktivierung von Azure ist jederzeit möglich, indem die **.env-Datei** entsprechend konfiguriert wird.

Hinweis zu Datenschutz

Die Anwendung speichert **Benutzerdaten lokal**. Bei Nutzung externer Dienste (z.B. Google Calendar oder OpenAI) werden **nur notwendige Daten übertragen**. API-Schlüssel und Zugangsdaten sind in der `.env` - Datei hinterlegt und **nicht im Code sichtbar**.