Habit Tracker – Managing Habits Digitally

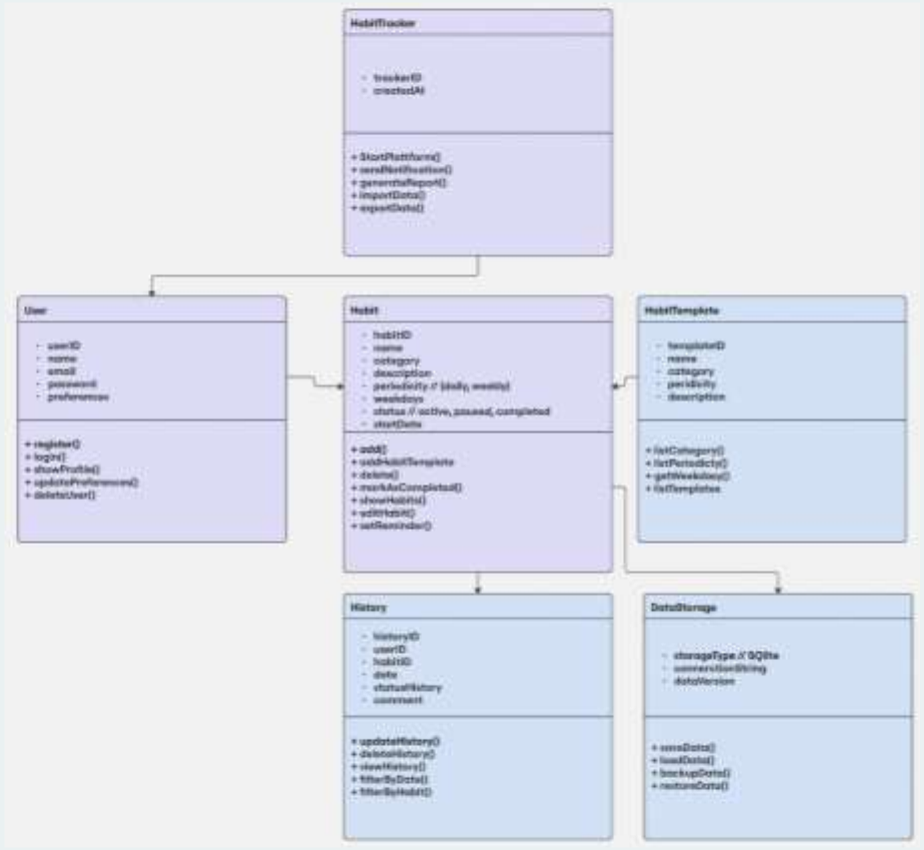
Key Features

- User registration & login
- Custom and template-based habit creation
- Progress analysis and history (to be expanded)
- All data stored offline locally

Architecture & Technologies

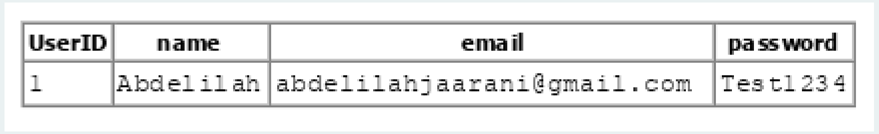
- Python 3 as primary language
- SQLite for secure, local data storage
- Modular code structure: user.py, habit.py, dataStorage.py, etc.

$$\begin{tikzcd}
\text{User} \arrow[r] & \text{CLI} \arrow[r] & \text{Python} \arrow[r] & \text{SQLite}
\end{tikzcd}$$

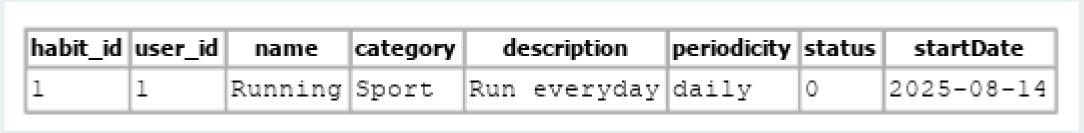


Data Structure & Storage

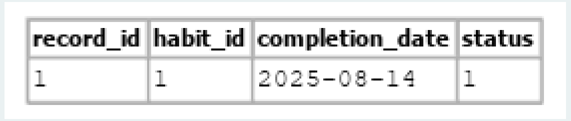
- USER table: UserID, name, email, password
- habits table: HabitID, user_id, name, category, description, periodicity, status, startDate
- “Periodicity” explicitly implemented (daily/weekly)



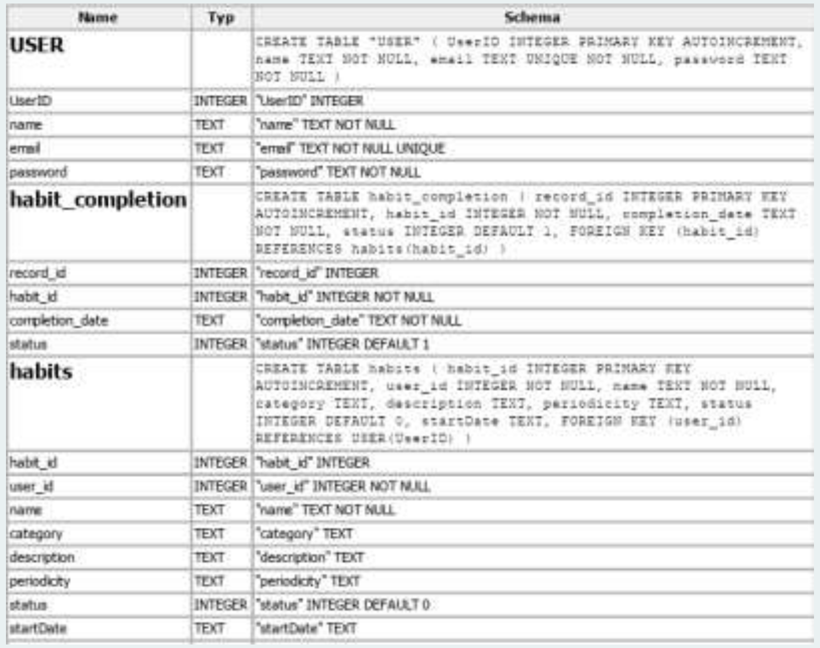
UserID	name	email	password
1	Abdelilah	abdelilahjaarani@gmail.com	Test1234



habit_id	user_id	name	category	description	periodicity	status	startDate
1	1	Running	Sport	Run everyday	daily	0	2025-08-14



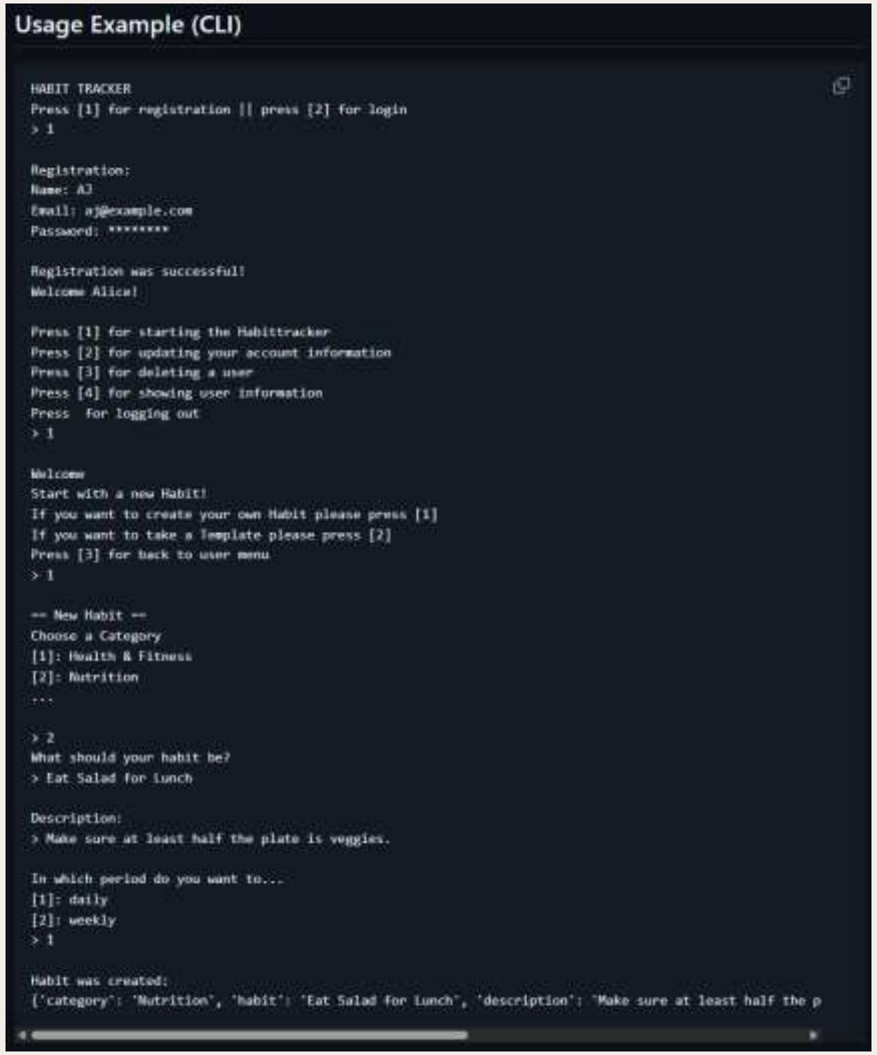
record_id	habit_id	completion_date	status
1	1	2025-08-14	1



Name	Typ	Schema
USER		CREATE TABLE "USER" (UserID INTEGER PRIMARY KEY AUTOINCREMENT, name TEXT NOT NULL, email TEXT UNIQUE NOT NULL, password TEXT NOT NULL)
UserID	INTEGER	"UserID" INTEGER
name	TEXT	"name" TEXT NOT NULL
email	TEXT	"email" TEXT NOT NULL UNIQUE
password	TEXT	"password" TEXT NOT NULL
habit_completion		CREATE TABLE habit_completion (record_id INTEGER PRIMARY KEY AUTOINCREMENT, habit_id INTEGER NOT NULL, completion_date TEXT NOT NULL, status INTEGER DEFAULT 1, FOREIGN KEY (habit_id) REFERENCES habits(habit_id))
record_id	INTEGER	"record_id" INTEGER
habit_id	INTEGER	"habit_id" INTEGER NOT NULL
completion_date	TEXT	"completion_date" TEXT NOT NULL
status	INTEGER	"status" INTEGER DEFAULT 1
habits		CREATE TABLE habits (habit_id INTEGER PRIMARY KEY AUTOINCREMENT, user_id INTEGER NOT NULL, name TEXT NOT NULL, category TEXT, description TEXT, periodicity TEXT, status INTEGER DEFAULT 0, startDate TEXT, FOREIGN KEY (user_id) REFERENCES USER(UserID))
habit_id	INTEGER	"habit_id" INTEGER
user_id	INTEGER	"user_id" INTEGER NOT NULL
name	TEXT	"name" TEXT NOT NULL
category	TEXT	"category" TEXT
description	TEXT	"description" TEXT
periodicity	TEXT	"periodicity" TEXT
status	INTEGER	"status" INTEGER DEFAULT 0
startDate	TEXT	"startDate" TEXT

User Interaction & Tools

1. Step-by-step CLI menus guide the user
2. Choose category, description, interval, weekday
3. Predefined templates simplify the experience



Usage Example (CLI)

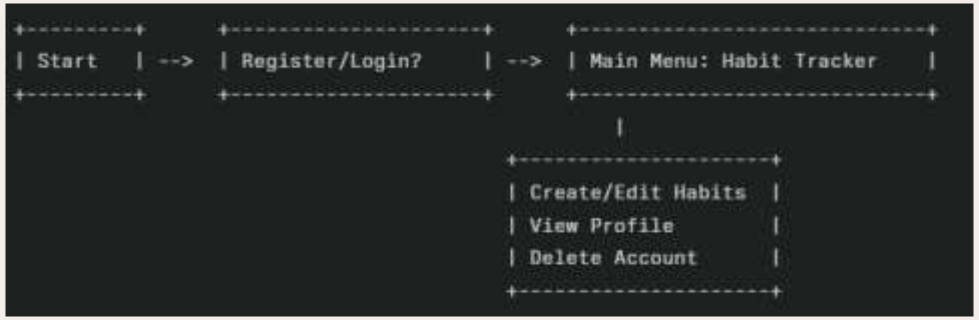
```
HABIT TRACKER
Press [1] for registration || press [2] for login
> 1

Registration:
Name: AJ
Email: aj@example.com
Password: ********

Registration was successful!
Welcome Alice!

Press [1] for starting the Habittracker
Press [2] for updating your account information
Press [3] for deleting a user
Press [4] for showing user information
Press  for logging out
> 1

Welcome
Start with a new Habit!
If you want to create your own Habit please press [1]
If you want to take a Template please press [2]
Press [3] for back to user menu
> 1

== New Habit ==
Choose a Category
[1]: Health & Fitness
[2]: Nutrition
...

> 2
What should your habit be?
> Eat Salad for Lunch

Description:
> Make sure at least half the plate is veggies.

In which period do you want to...
[1]: daily
[2]: weekly
> 1

Habit was created:
{'category': 'Nutrition', 'habit': 'Eat Salad for Lunch', 'description': 'Make sure at least half the p
```

Addressing Challenges

Reducing user overwhelm: templates & clear menus

No internet dependency: local database only

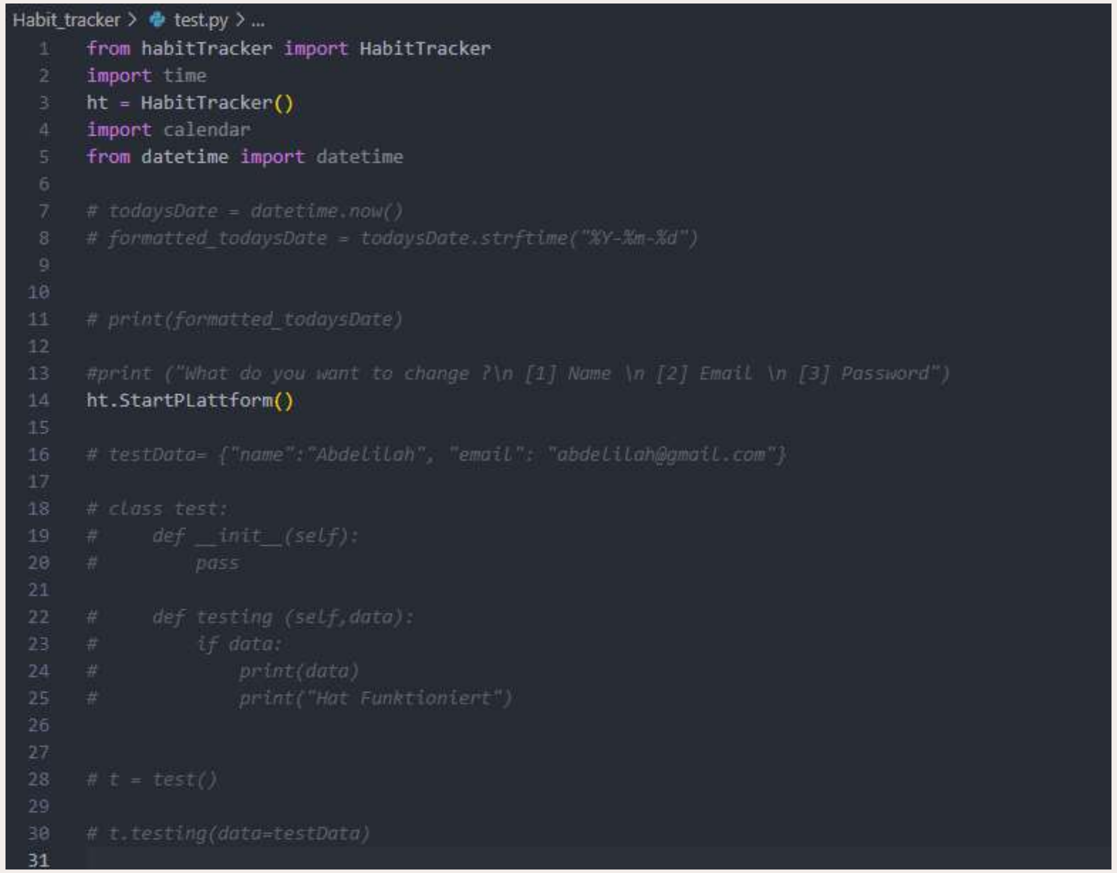
“Coach” idea acknowledged but not the main focus



```
+---------+      +---------------------+      +------------------------------+
| Start   | -->  | Register/Login?     | -->  | Main Menu: Habit Tracker     |
+---------+      +---------------------+      +------------------------------+

                                                  |
                                         +---------------------+
                                         | Create/Edit Habits  |
                                         | View Profile        |
                                         | Delete Account      |
                                         +---------------------+
```

Testing & Validation

1. Initial test script (test.py) for functionality checks
2. Comprehensive validation and error handling planned


Habit_tracker > test.py > ...

```python
from habitTracker import HabitTracker
import time
ht = HabitTracker()
import calendar
from datetime import datetime

# todaysDate = datetime.now()
# formatted_todaysDate = todaysDate.strftime("%Y-%m-%d")


# print(formatted_todaysDate)

#print ("What do you want to change ?\n [1] Name \n [2] Email \n [3] Password")
ht.StartPLattform()

# testData= {"name":"Abdelilah", "email": "abdelilah@gmail.com"}

# class test:
#     def __init__(self):
#         pass

#     def testing (self,data):
#         if data:
#             print(data)
#             print("Hat Funktioniert")


# t = test()

# t.testing(data=testData)

```

Current Status & Future Roadmap

- Known issues: errors in DB queries, incomplete features (edit/delete,analyse,history)
- Roadmap:
  - Fix DB operations
  - Implement habit editing/deleting/history,analyse
  - Improve error handling
  - Add import/export (CSV/JSON)

```
-------|------------------------|--------------------------|--------------------------|-----
```

Aug 11-17th	**Aug 18-21th**	**Aug 22-24th**	**Aug 25th**
MVP Build	*Testing*	*Final Fixes*	*Submission*
CLI Flow	*Bugs*	*Export/Docs*	
DB Setup	*Analysis*	*Polish*	
Templates	*Basic reporting*		