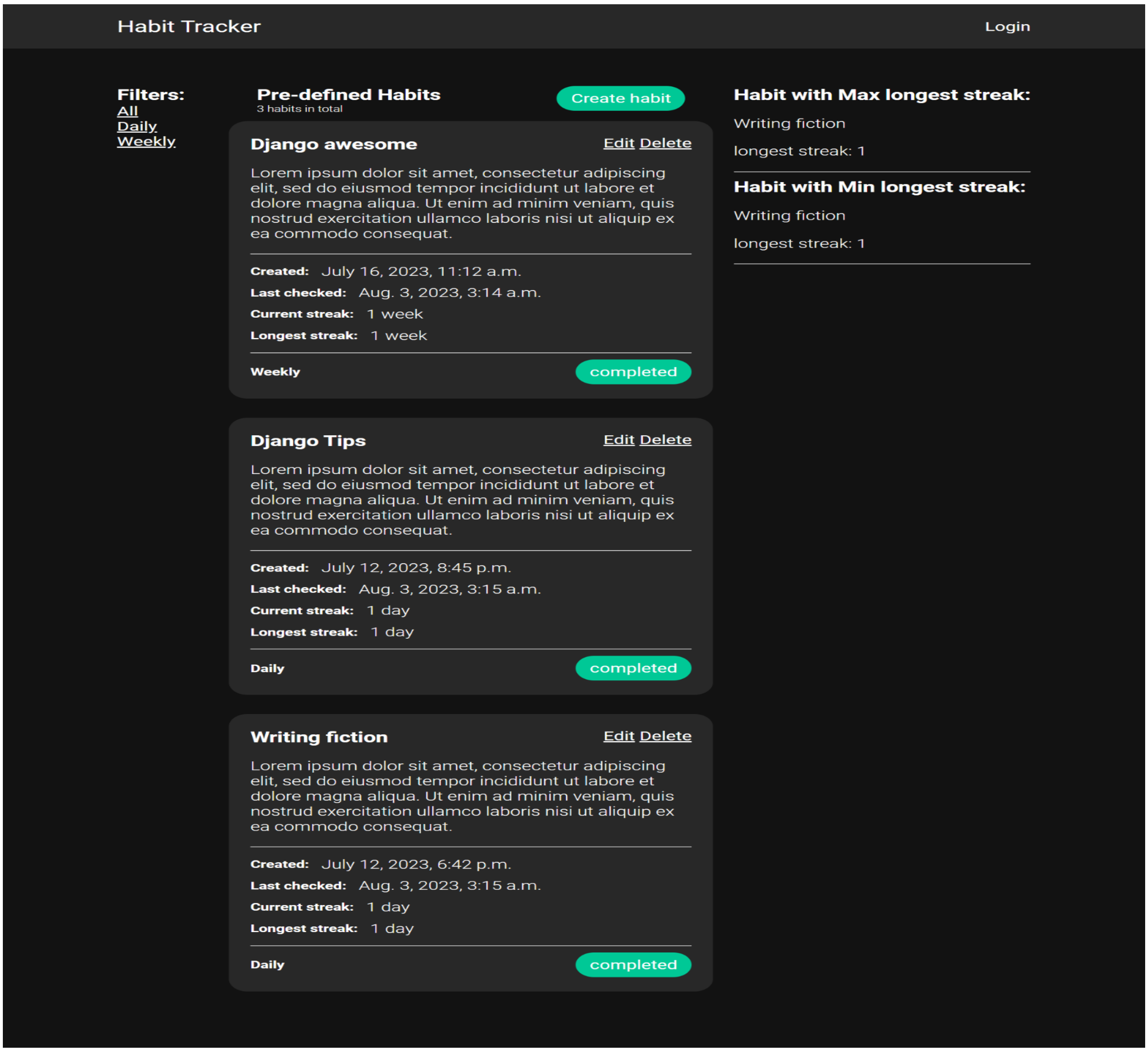
Conception of the Habit Tracker website:

The Habit Tracker website will be built using the Django framework and the data will be stored in a sqlite3 database.

Here is a screenshot of the actual design of the website made using HTML and CSS:



Habit Tracker

Login

Filters:
All
Daily
Weekly

Pre-defined Habits
3 habits in total

Create habit

Django awesome Edit Delete

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Created: July 16, 2023, 11:12 a.m.
Last checked: Aug. 3, 2023, 3:14 a.m.
Current streak: 1 week
Longest streak: 1 week

Weekly completed

Django Tips Edit Delete

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Created: July 12, 2023, 8:45 p.m.
Last checked: Aug. 3, 2023, 3:15 a.m.
Current streak: 1 day
Longest streak: 1 day

Daily completed

Writing fiction Edit Delete

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Created: July 12, 2023, 6:42 p.m.
Last checked: Aug. 3, 2023, 3:15 a.m.
Current streak: 1 day
Longest streak: 1 day

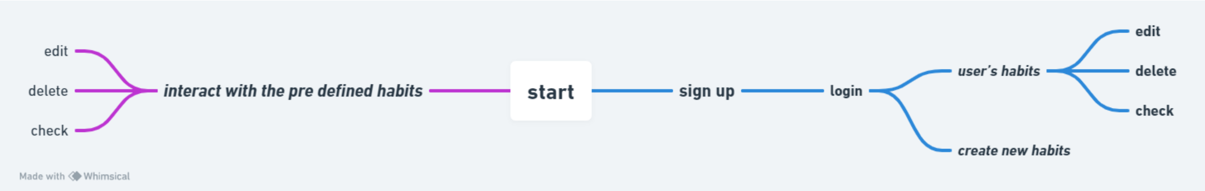
Daily completed

Habit with Max longest streak:
Writing fiction
longest streak: 1

Habit with Min longest streak:
Writing fiction
longest streak: 1

Each website user has to sign up and log in before creating his own habits. He can create, update, delete, check a habit when completed, filter the displayed habits by periodicity, and get a list of all the habits he has created.

User’s flow:



The class Habit will represent a single habit. Each habit will have the following attributes:

user: the user who created the habit.
name: name of the habit.
task: a description of the habit and its required actions.
periodicity: daily or weekly.
created: the date when the habit was created.
last_check: the date when the habit was completed.
streak: the current streak of days or weeks depending on the periodicity.
longest_streak: the longest streak.

The Habit class will also provide methods related to the habit itself, checking habits when completed and unchecking them if not, calculating the streak and longest_streak. The Habit class would therefore encapsulate all the data and behavior related to a single habit.

The Habit class implements streak calculation by maintaining a record of consecutive completion days or weeks for a specific habit. Here's a more detailed explanation of how the streak calculation works:

When a user completes a habit for a day or week, the Habit class updates the last_check attribute with the current date, marking the completion time.

The next time the user interacts with the habit (i.e., checks it), the class compares the current date with the last check date.

If the difference between the current date and the last_check date equals one day (for daily habits) or one week (for weekly habits), the streak is considered unbroken, and it is extended by one day or one week, respectively.

However, if the difference between the dates is more than one day or week, it indicates that the habit was not completed on consecutive days or weeks. In this case, the streak is broken, and the streak counter is reset to 1 to start counting from the current completion.

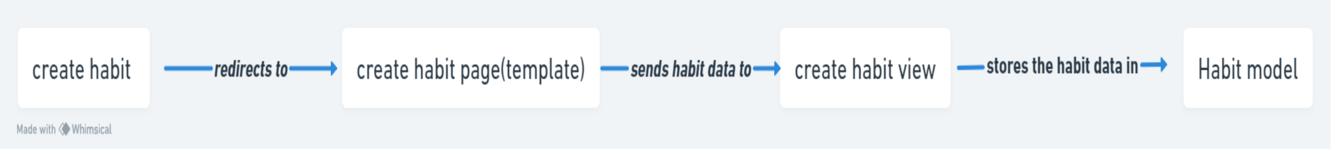
The Habit class then updates the streak attribute with the current streak value, reflecting the consecutive completion count.

Additionally, the longest_streak attribute keeps track of the highest streak achieved so far. If the current streak surpasses the previous longest streak, the value of longest_streak is updated accordingly.

The Habit analysis module will have functions that manage the analysis of all the habits created by the user. For example the filtering of the habits by periodicity, and the calculation of the habits with the max and min longest streaks.

In Django's MVT (model view template) architecture, the Habit class is the model for storing habit instances in the SQLite3 database. Templates, constructed using HTML and CSS, represent the user interface. Views manage the application's logic and communication with the habit database.

Example of Django's MVT architecture in creating a habit:


create habit
redirects to
create habit page(template)
sends habit data to
create habit view
stores the habit data in
Habit model
Made with Whimsical