Habit Tracker

User Manual

Table of Contents

1. Overview 2
1.1 Introduction 2
1.2 Installation 2
1.3 User creation 2
1.4 Login 3
2. Program usage 3
2.1 Home 3
2.2 Managing Habits 3
2.2.1 Create a habit 3
2.2.2 Check off a habit 4
2.2.3 Delete a habit 4
2.3 Analysing Habits 4
2.3.1 Habits with same periodicity 4
2.3.2 Longest run streak 4
2.3.3 Longest streak of a habit 4
2.4 Logout 4
3. Test suite 4
3.1Test suite introduction 4
3.2 Back-end test 4
3.3 Front-end test 5
3.3.1 Front-end test requirements 5
4. Additional content 5
4.1 Data 5
4.2 Web driver 5

1. Overview

1.1 Introduction

‘Habit Tracker ‘ is a web application, which allows a user to manage and analyze their self-defined habits with a specific periodicity. Since ‘Habit Tracker’ is a web application, the program can be controlled through a GUI (Graphical User Interface) on a web browser. The web browser exposes the functionality of the program through HTML and CSS. The data is stored locally on a .json file, which is named automatically after the respective username.

1.2 Installation

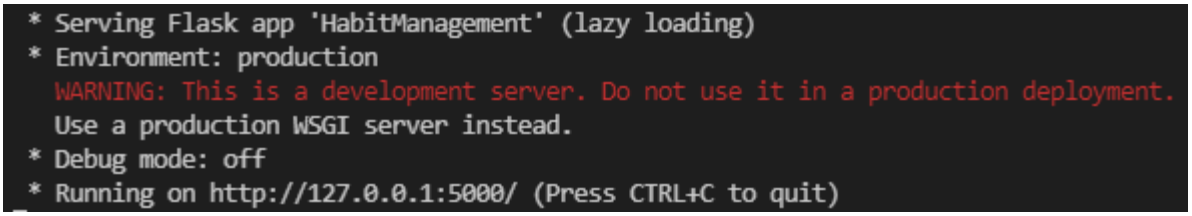
The requirements for ‘Habit Tracker’ are:

- Python Version 3.9.0
- A webbrowser (e.g., Google Chrome, Internet Explorer)

To install and use the application the following steps must be fulfilled:

1. Download the application from Github and extract the contents
2. If you do not have ‘pipenv’ already installed. Open a code command prompt or terminal in the extracted folder and install ‘pipenv’ with the command ‘pip install pipenv’
3. Then execute the command ‘pipenv install’. This installs all the dependencies needed for the program.
4. Execute the command ‘pipenv shell’ to open a new virtual environment
5. Change the directory to ‘Program’ with the command ‘cd Program’
6. Execute the command ‘py bridge.py’ to run the application

To access the webpage, use the link, which is generated in the terminal after running the application.



1.3 User creation

To create a new user, use the button ‘Register’ in the navbar. This redirects you to the ‘register’ page, which allows you to fill in your form. If the entered form is compliant with the registration requirements press the ‘Submit’ button to create the user profile. This profile can then be accessed again through ‘Login’.

1.4 Login

To access an already existing user profile, use the button ‘Login’ in the navbar. This redirects you to the ‘login’ page’, which allows you to fill in your form. Enter the username and the password of one of your profiles and press the ‘Login’ button, to log in. Per default, an already existing user profile is in the Program package. The content of the user data can be inspected by opening the .json file.

The log in information for the default user is:

- Username: ‘user’
- Password: ‘1234’
- Email-address: ‘user@hotmail.com’

2. Program usage

2.1 Home

After login, you are directed to the homepage, which showcases a ‘Habit Tracker’ image. From the Homepage all the functionality of the Habit Tracker is accessible.

2.2 Managing Habits

Pressing the ‘Habits’ button on top of the navbar allows you to access the Habit Management page. On this page, all your habits can be managed.

The following actions can be performed on habits:

- Create
- Check
- Delete

It also shows a complete list of all your habits with their respective statistics:

- Task
- Periodicity
- Last checked
- Current streak

2.2.1 Create a habit

To create a habit, fill out the form directly under the heading of ‘Create a Habit’. You have to define a task and set a periodicity. The periodicity is in days and defines how many days you have left to check off a habit.

2.2.2 Check off a habit

Under the heading ‘List of your habits’ various statistics of a given habit can be viewed. In addition to the task and periodicity, the date when a habit was last checked, and the current streak of a habit are also shown. To check a habit off, press the ‘Check’ button in the respective row of the habit. Checking off a habit means that a task has been fulfilled. If a task has been fulfilled at least once in a given periodicity, the streak of the habit increase. If a task is however checked off outside of a given periodicity, the streak interrupts. A streak can only be increased once per day.

2.2.3 Delete a habit

The deletion of a habit functions in the same manner as checking a habit off. To delete a habit, press the ‘Delete’ button in the respective row of a habit. The deletion removes the habit completely and the habit can not be restored.

2.3 Analysing Habits

Pressing the ‘Analytics’ button on top of the navbar allows you to access the Analytics page. On this page, all your habits can be inspected.

2.3.1 Habits with same periodicity

This function returns all the habits, which share the same periodicity. To do that you must enter a periodicity in the form and press the Enter key on your keyboard.

2.3.2 Longest run streak

This function returns the habit or habits with the overall longest run streak. The table is shown by default and no user input is necessary.

2.3.3 Longest streak of a habit

This function returns the longest-run streak of a given habit. To do that you must enter the task of a habit in the form and press the Enter key on your keyboard.

2.4 Logout

When you want to exit the application simply press the ‘Logout’ button in the navbar. This redirects you back to the login page. Another way to exit the application is to simply close the web browser.

3. Test suite

3.1 Test suite introduction

Since ‘Habit-Tracker’ is a web application, the test suite of this application is split into two different parts back-end tests and front-end tests. To run the test applications the same steps must be fulfilled like running the main program. Instead of step 6 however, the commands ‘py test_back_end.py’ and ‘py test_front_end.py’ must be executed.

3.2 Back-end test

The back-end tests, test that the views of the habit tracker are working as expected. The testing is done via the built-in testing framework ‘unittest’. The tests check if a successful connection can be

established to the different views. This is shown via the status code 200. The tests also confirm if a user can not access views, which they do not have the authorization for. A user needs to be logged in first. Accessing views of the program, which are not ‘login’ and ‘registration’, should not be possible. If a user can not access a page of the habit tracker main program, this is shown via the status code 302.

3.3 Front-end test

The front-end tests, test that the functions of the habit tracker are working as expected. The testing is done via the built-in testing framework ‘unittest’ and the third-party framework ‘selenium’. ‘Selenium’ is a framework, which automates web applications for testing purposes. The tests simulate a user and check if the functionality of all features is working as intended.

3.3.1 Front-end test requirements

For the front-end test to work, the following steps must be fulfilled:

1. Execute the main program like the instructions in chapter 1.2
2. Open a second terminal
3. Execute the command ‘pipenv shell’ to open a new virtual environment
4. Change the directory to ‘Program’ with the command ‘cd Program’
5. Execute the command py.test_front_end.py to run the tests

For the front-end test to work, the server must be up and running otherwise, a connection can not be established.

4. Additional content

4.1 Data

The ‘Data’ folder contains all the created accounts of a user. The filename is generated automatically and consists of the username and .json. The name of the files should therefore not be changed. User information can be further inspected by viewing the files.

4.2 Web driver

For the front-end test to work, a web driver is required. The web driver acts independently of the standard web browser and is needed by selenium. Per default the chrome web driver is used.