01QZP – Ambient Intelligence

Lab 5 – Web in python: first approach to flask

Luigi De Russis, Fulvio Corno

LAB 5 – WEB IN PYTHON: FIRST APPROACH TO FLASK

GOAL

The goal of this lab is to write a web application, based on the Flask framework, to implement a web-based version of the task-management application developed in the fourth laboratory (AmITaskListBot.py).

PRELIMINARY

You should start with the basic structure of a Flask application (app.py file and templates/ and static/ directories). You may do that by creating a new project (PyCharm > File > New Project... > Flask), or by forking+cloning of the Lab5 repository (go to *https://github.com/Aml-2018/python-lab5*, Fork, then on PyCharm > VCS > Checkout from Version Control > Git). Remember to checkout (clone) your forked version of the project, not the one under Aml-2018.

We also suggest to checkout (clone) the Lab4 project (the one you developed last week) in a different window, to copy and re-use all the database access functions.

STEP 1 - INSERT AND "ID" FIELD IN THE DATABASE

Modify the database MySql schema starting from the one used in Lab3/Lab4.

- 1. Add a new integer column called "id" to the task table. This column should be set as *primary key*, *auto-increment*.
- 2. Check and modify the database access functions (especially db_remove_task) in order to use id instead of todo to identify a specific task. In this way we properly use *primary keys* to identify rows in a table.

STEP 2 – TRANSFORM THE TELEGRAM BOT IN A SIMPLE WEB APPLICATION

- 1. Create a web application to display, in its main page (see Figure 1), a list of all existing tasks.
- 2. Add a one-line form to insert a new task, at the bottom of the page. Entering the new task should create it into the database. The form should call the insert task page, with description and urgent parameters.
- 3. Add a "delete" button (or link) besides each task: it will delete the corresponding task. The link should call a delete task page and embed the id parameter¹ of the task to be deleted.

The overall "site map" of the web application is shown in Figure 2.

¹ Suggestion: use url for ('delete page', id task=id), combined with an

[@]app.route('/delete page/<id task>'). See http://flask.pocoo.org/docs/0.12/quickstart/#url-building

01QZP – Ambient Intelligence

Lab 5 – Web in python: first approach to flask

Luigi De Russis, Fulvio Corno

Figure 1 – Index

(*) ⁽¹] 127.0.0.1:5000/index.html

Welcome to the Todo List Manager

Here you can see all existing tasks

- book summer holidays <u>Delete</u>
 buy a new mouse <u>Delete</u>
 call Giovanni for AmI project organization <u>Delete</u>
 find a present for Angelina's birthday <u>Delete</u>
 install Notification Collector from Play Store to help e-Lite research <u>Delete</u>
 organize mega party (last week of April) <u>Delete</u>
 whatsapp Mary for a coffee <u>Delete</u>

Insert a new task:	Enter
Insert a new task:	Enter

Figure 2 – Site Map

dex		<u> </u>	
Welcome to the T	odo List Mai	nager	
Here you can see all existing tasks			
 book summer holidays <u>Delete</u> buy a new mouse <u>Delete</u> call Giovanni for Aml project or find a present for Angelina's birt install Notification Collector fro organize mega party (last week whatsapp Mary for a coffee <u>Delete</u> 	iday <u>Delete</u> 1 Play Store to help e-Lite r f April) <u>Delete</u>	research <u>Delete</u>	
Insert a new task:	Ente	r	
		insert_	task description