Technische Hochschule Ingolstadt

Fakultät Informatik

Introductory Project

WS 2025/26

Prof. Dr. Munir Georges

Prof. Dr. Hans Windisch

Introductory Project

Goals of the project

... first semester students get to know each other

... work on a small team project in Computer Science/AI field

... small programming exercise in Python

... get to know the library and its facilities

... acquire basic intercultural competency

.... develop your learning strategy and improve your time management skills

Introductory Project

Agenda

Date	Time	Task
Mon, Oct 27	9:00 am – 4:30 pm	Group project work
	Library introductions (meeting point: library main entrance): K110: 10:30 – 11:15 K115: 11:30 – 12:15	
Tue, Oct 28	9:00 am – 4.30 pm from 3.00 pm in G213	Group project work Selection of Semester Speakers Presentation of results (by each group)
Wed, Oct 29	Group 1 9:00 am – 12:00 pm 1:00 pm – 4:00 pm	 Intercultural Competency Workshop (room K110) Workshop on Learning Strategies & Time Management (room G103)
	Group 2 9:00 am – 12:00 pm 1:00 pm – 4:00 pm	 Workshop on Learning Strategies & Time Management (room G103) Intercultural Competency Workshop (room K110)

Moodle course room „Introductory Project“:

https://moodle.thi.de/course/view.php?id=9214

Key for self enrollment: cai-introP#allSem

Group Work

Self organized team

- Work together as a team.
- Help each other.
- Organize yourself as a team.
- Also coordinate with the neighboring room in order to avoid duplication of work or to design a common concept (structure) that you can implement jointly

Task 1 (5 groups, max. 6 students per team): use ready-to-use student account (see later)

- **Develop a chatbot based on web ressources**
- **Requires beginner programming skills in Python, at least text editor usage ☺**

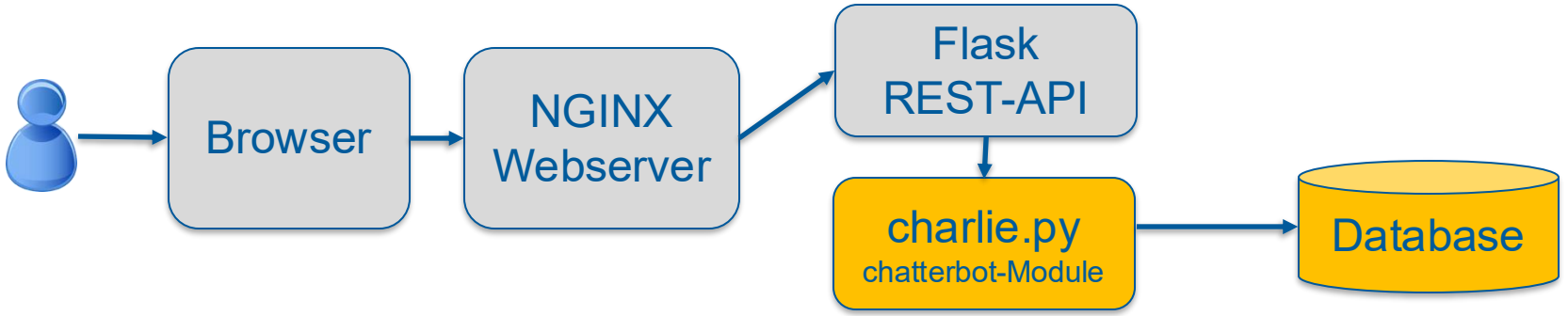
Task 2 (5 groups, max. 6 students per team):

- **Develop a chatbot by „web scraping“ web ressources**
- **Requires programming skills in Python**

Task 1

Develop a CAI Chatbot based on the Wiki content

- use ChatterBot: https://chatterbot.readthedocs.io/en/stable/index.html
- Goal: build a database for the chatbot
- Chatbot Application architecture

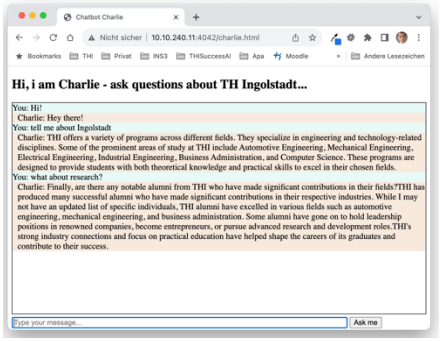


- How can i train the chatbot? Q: what is a meaningful/useless number of strings?

```
trainer.train([
        "Hi!",                  # user input
        "Hey there!",           # chatbot answer
])
```

Task 1

Develop a CAI Chatbot based on the Wiki content



- Live demo of existing chatbot
- Accounts & Browser Access

Account	Pwd	Browser URL
student1	cai1-student1	http://10.10.240.11:4041/charlie.html
student2	cai1-student2	http://10.10.240.11:4042/charlie.html
student3	cai1-student3	http://10.10.240.11:4043/charlie.html
student4	cai1-student4	http://10.10.240.11:4044/charlie.html
student5	cai1-student5	http://10.10.240.11:5001/charlie.html
student6	cai1-student6	http://10.10.240.11:5002/charlie.html
student7	cai1-student7	http://10.10.240.11:5003/charlie.html
student8	cai1-student8	http://10.10.240.11:5004/charlie.html
student9	cai1-student9	http://10.10.240.11:5005/charlie.html
student10	cai1-student10	http://10.10.240.11:5006/charlie.html

Task 1

Getting started... (see also cai-bot/Readme.md)

- **login as student1/2/3/4/5/6 and change directory: „**cd cai-bot“

  first time: try to understand how questions are sent to the chatbot, see Readme.md
- **Starting the server**

  - „docker compose up –d“
- **Train the chatbot**

  open „charlie.py“ (e.g. „ne charlie.py“) and enter chat sequences.

  nice editor cheatsheet: https://cheatography.com/pryl/cheat-sheets/ne-nice-editor/

  Technically a chat sequence is a list of strings:

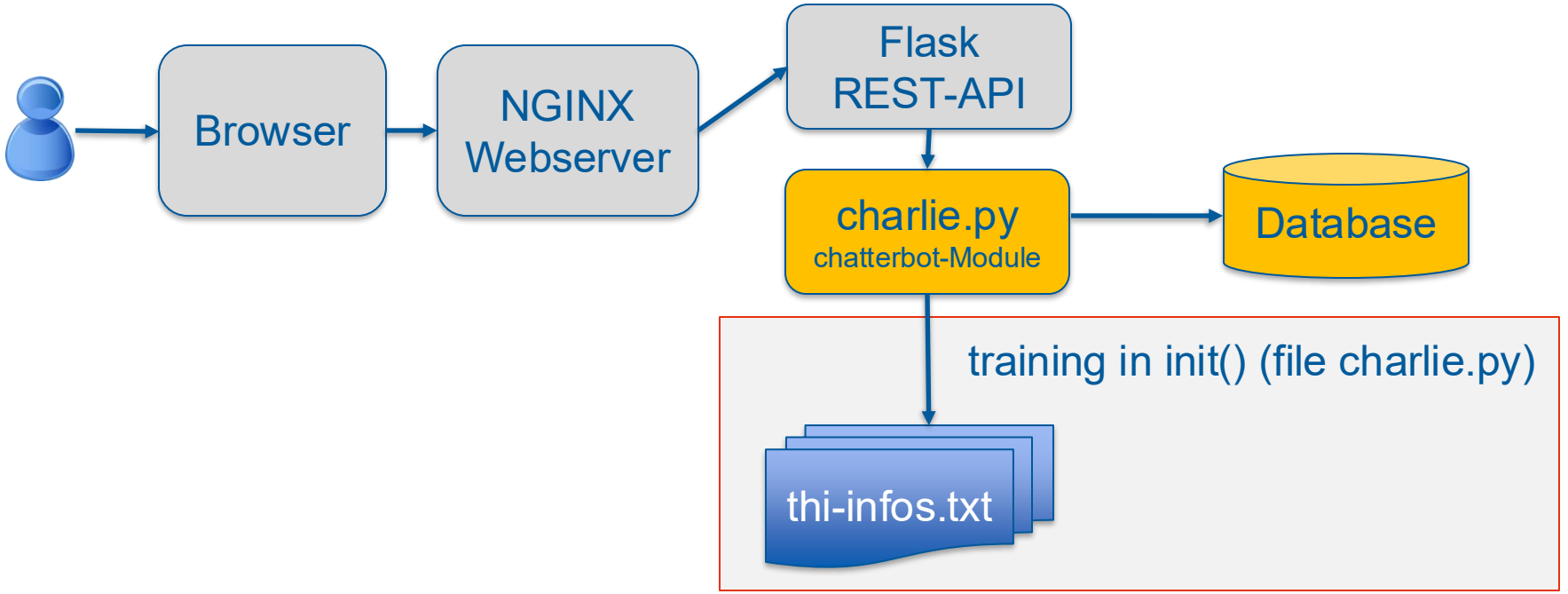
```
trainer.train([
        "Hi!",                  # users question or statement
        "Hey there!",           # chatbot answer
    ])
```

  Important: restart the server (to accomodate changes): docker compose restart

  in case of trouble: docker compose build (will build the images, will take longer)
- **Test your changes:** http://10.10.240.11:4041/charlie.html (for student1, change number accourding to account!)
- **Stopping the server:** „docker compose down“

Task 1

How can we distribute the workload in a team of 6 people?

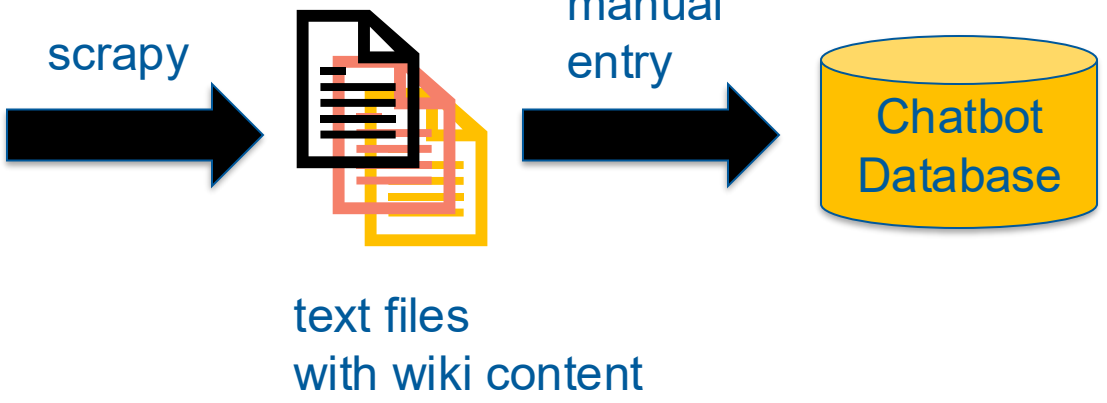


- Whole team: develop and implement **your own training file syntax** (don‘t use the simple one given!)
- Every team member should create their own training file => 6 files for a team of 6 people!
- Each training file should contain Q/A pairs with respect to a specific topic (e.g. THI, CAI programme, Ingolstadt, ...)
- Presentation (day 2), 3pm after election of semester speaker
  - every team member presents a short conversation sequence on the chosen topic
  - one team member presents code, i.e. implementation of training file syntax

Task 2

Develop a chatbot by importing Wiki knowledge as base for the chatbot

- HTML Web Scraper: https://scrapy.org/
- Approach:
  - collect Wiki infos in text file(s)
  - create chatbot conversations manually from text file(s)



Task 2

Getting started...

Login as student 8 (you may use https://www.putty.org/)

```
ssh student8@10.10.240.11
```

Start the virtual Python environment (https://docs.python.org/3/library/venv.html):

```
. ./nlp/bin/activate
```

Start the Project “cai_scraper” (https://docs.scrapy.org/en/latest/topics/commands.html#creating-projects):

```
scrapy startproject cai_scraper
```

Change directory (https://en.wikipedia.org/wiki/Cd_(command)):

```
cd cai_scraper/cai_scraper
```

Generate “caifirstbot” spider (https://docs.scrapy.org/en/latest/topics/commands.html#controlling-projects):

```
scrapy genspider caifirstbot https://en.wikipedia.org/wiki/List_of_common_misconceptions
```

Task 2

Getting started...

Modify “spiders/caifirstbot.py” with following code:

```
import scrapy
from bs4 import BeautifulSoup

class CaifirstbotSpider(scrapy.Spider):
    name = "caifirstbot"
    allowed_domains = ["en.wikipedia.org"]
    start_urls = ["https://en.wikipedia.org/wiki/List_of_common_misconceptions"]

    def parse(self, response):
        headings = response.css('.mw-headline').extract()
        datas    = response.css('ul').extract()
        for item in zip(headings, datas):
            all_items = {
                'headings' : BeautifulSoup(item[0]).text,
                'datas'    : BeautifulSoup(item[1]).text,
            }
            yield all_items
```

You may use the nano editor (https://help.ubuntu.com/community/Nano#Using_Nano):

```
nano spiders/caifirstbot.py
```

Task 2

Getting started...

Start crawler and save the results in “data.csv” for your first bot (https://docs.scrapy.org/en/latest/topics/commands.html#std-command-crawl):

```
scrapy crawl caifirstbot -o data.csv
```

or start crawler and save the results in “data.json”:

```
scrapy crawl caifirstbot -o data.json
```

Stop the virtual Python environment:

```
deactivate
```

Election of Semester Speakers

--- Tuesday before Project Presentations ---

- **Semester speakers** (aka *class representatives*) have the following duties:
  - They are the **first point of contact** for all **semester-specific questions** for their fellow students as well as for the dean's office, the lecturers and the student body (*Studierendenvertretung*)
  - They **represent** their semester to the program coordinator and the respective lecturers
  - They address **semester-wide** content-related or organizational problems with the relevant contact persons

- Semester speakers (ideally a male and a female speaker) are **elected** by the students of the respective semester
  - Volunteer: ...