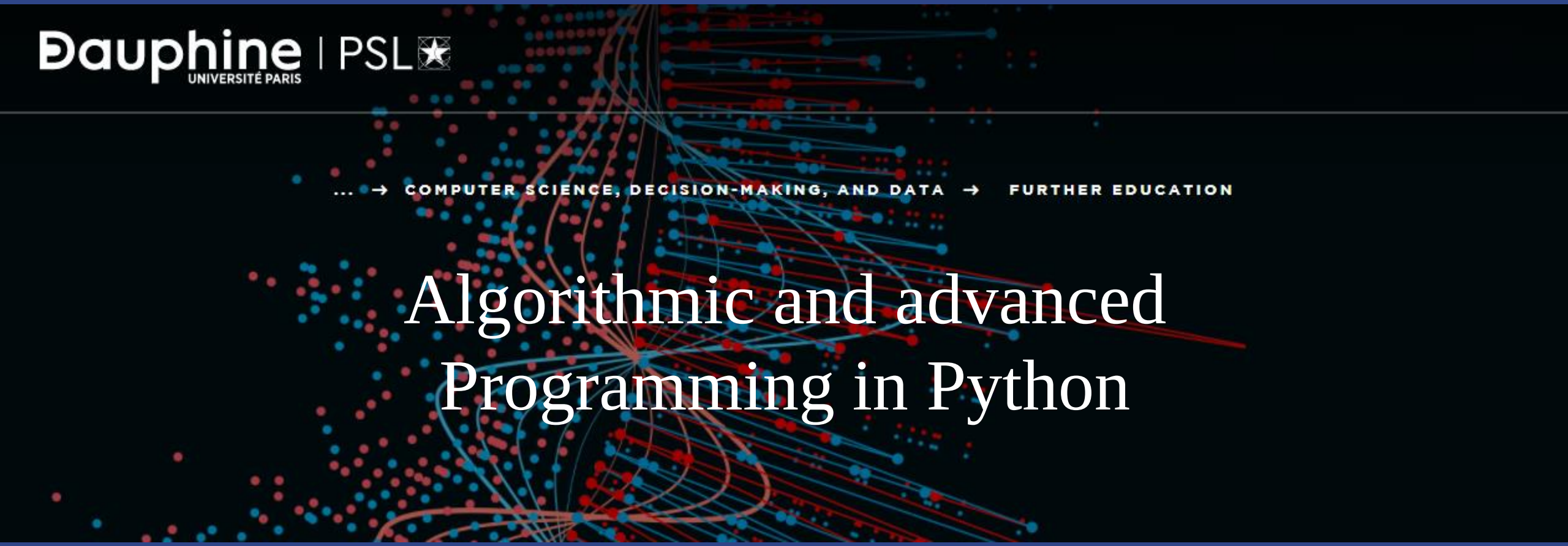


Algorithmic and advanced Programming in Python

Mid term evaluation instructions

Reminder of the objective of this course

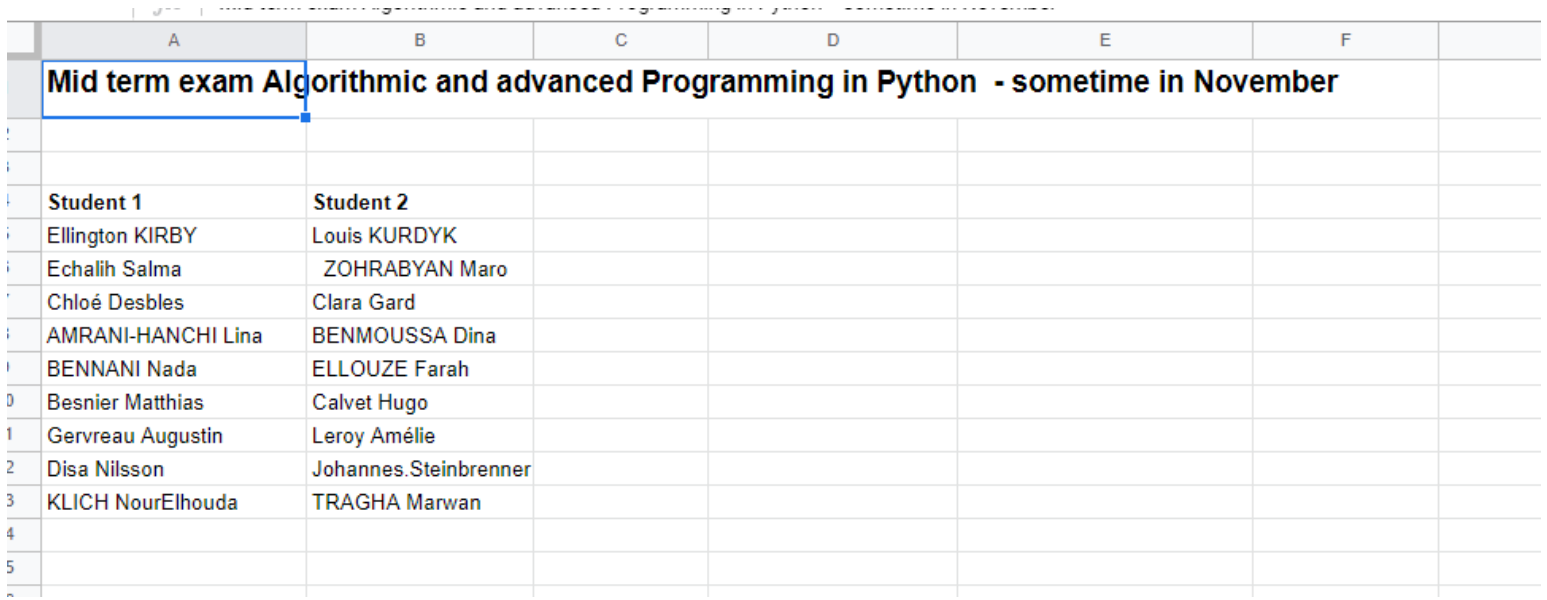
- People often learn about data structures out of context
- But in this course you will learn foundational concepts by building a real application with python and Flask

- To learn the ins and outs of the essential data structure, experiencing in practice has proved to be a much more powerful way to learn data structures

- The mid term evaluation is precisely to start playing with the data structure

Instructions for registering

- Form a group of two people and register on
- https://docs.google.com/spreadsheets/d/1zGRyi8vDB0-88_wBJ0grygla9gAl8jNewzGXC1_hI-A/edit?usp=sharing



	A	B	C	D	E	F
	Mid term exam Algorithmic and advanced Programming in Python - sometime in November					
4	**Student 1**	**Student 2**				
	Ellington KIRBY	Louis KURDYK				
	Echalih Salma	ZOHRABYAN Maro				
	Chloé Desbles	Clara Gard				
	AMRANI-HANCHI Lina	BENMOUSSA Dina				
	BENNANI Nada	ELLOUZE Farah				
0	Besnier Matthias	Calvet Hugo				
1	Gervreau Augustin	Leroy Amélie				
2	Disa Nilsson	Johannes.Steinbrenner				
3	KLICH NourElhouda	TRAGHA Marwan				
4						
5						

Instructions for preparing the mid term eval

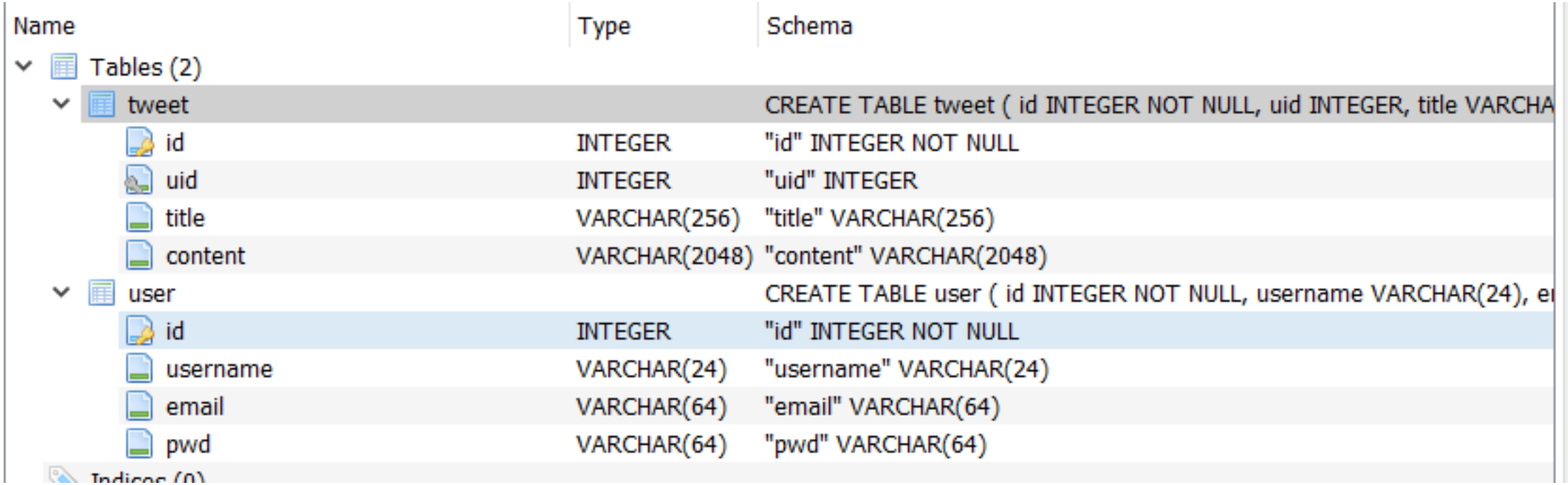
- The goal of this project is to recreate twitter and play with data structure

- You will have to work on both
  - The backend that will take care of storing
    - The users
    - The tweets

  - The front end with
    - A loggin section (loggin + register)
    - A part to display tweets

Backend

- Using flask, create an application that has the following data structure:
- Two tables:



Name	Type	Schema
Tables (2)		
tweet		CREATE TABLE tweet (id INTEGER NOT NULL, uid INTEGER, title VARCHA
id	INTEGER	"id" INTEGER NOT NULL
uid	INTEGER	"uid" INTEGER
title	VARCHAR(256)	"title" VARCHAR(256)
content	VARCHAR(2048)	"content" VARCHAR(2048)
user		CREATE TABLE user (id INTEGER NOT NULL, username VARCHAR(24), e
id	INTEGER	"id" INTEGER NOT NULL
username	VARCHAR(24)	"username" VARCHAR(24)
email	VARCHAR(64)	"email" VARCHAR(64)
pwd	VARCHAR(64)	"pwd" VARCHAR(64)
Indices (0)		

Tweet table

- Id: a unique identifier of tweets (primary key)
- Uid: a foreign key giving the unique identifier of the tweet's user
- Title: the title of the tweet
- Content: tweet content

- Because of tweet should be fairly brief, we will impose the following constraints:
  - Title: No more than 256 characters
  - Content: No more than 2048 characters

User table

- Id primary key
- Username (not more than 24 characters)
- Email
- password

Flask application for user

1. Create a class for User
2. Create function that get a user based on its uid
3. Add a user (insert equivalent in the database)
4. Remove a user (delete equivalent in the database)
5. Create a single route that can
   - Adding a user
   - Deleting a user
   - Get a user

```
@app.route("/api/users", methods=["GET", "POST", "DELETE"])
def users():
        ....
```

Data structure question

1. If I want from a user name to get his/her email and not to rely on the database, what data structure should I use to make it really fast, given that I have preloaded already all users in memory?

Connexion between users

- Create a table for storing the friendship relations between users?
- Which data structure should I use to have all relations in memory?
- How do I answer very rapidly if someone is a connexion of a given user?

Tweets

- Create a class object for the tweets

class Tweet(db.Model) ….

- Create a function to add a tweet, delete a tweet, get the user tweet

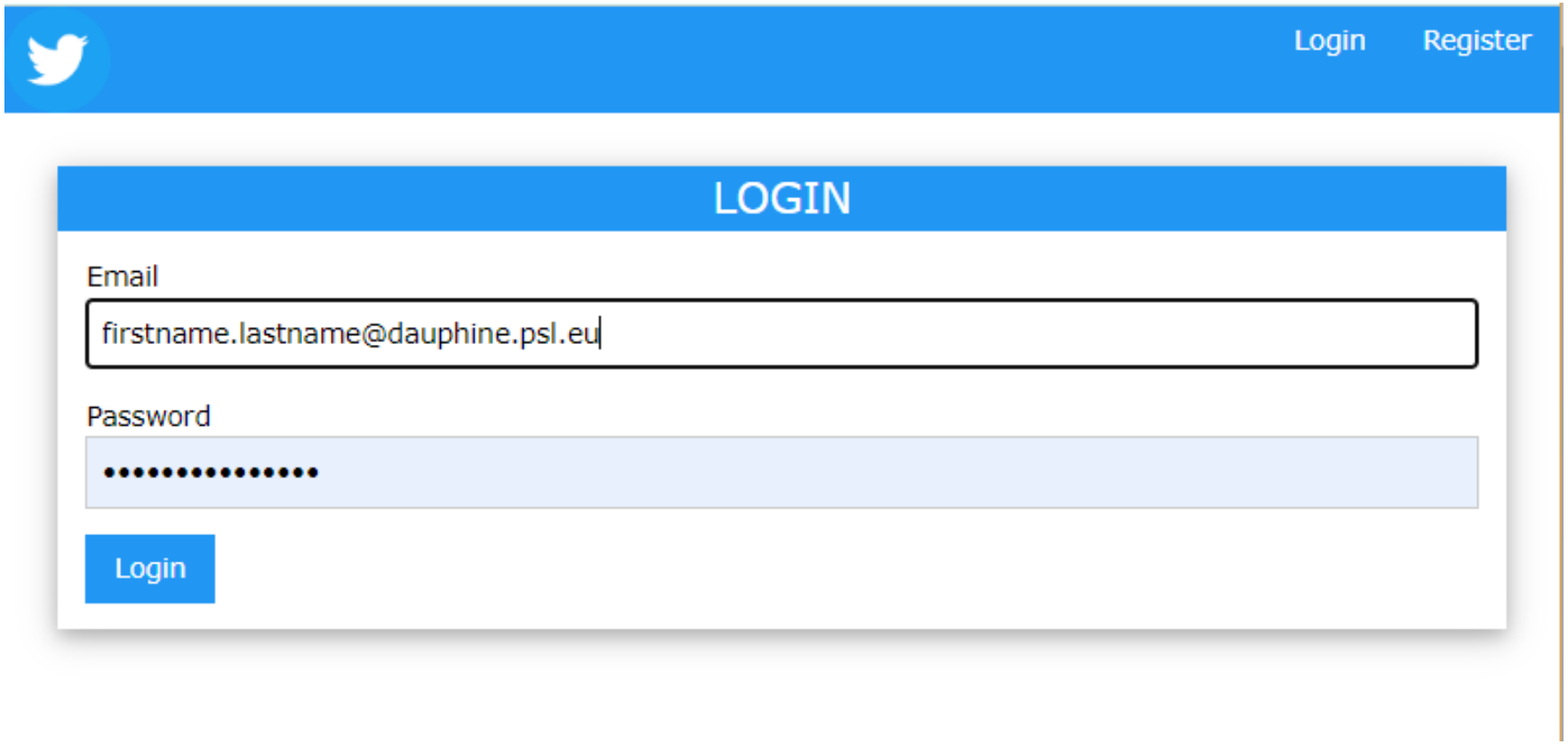
- Create an app route for
  - Getting tweets
  - Adding tweets
  - Deleting tweets

Tweets

- Create a function that provides in the fastest way all the tweets that contains a given word?
- Explain what data structure you choose?

Front end

- Create a navigation bar with the following items

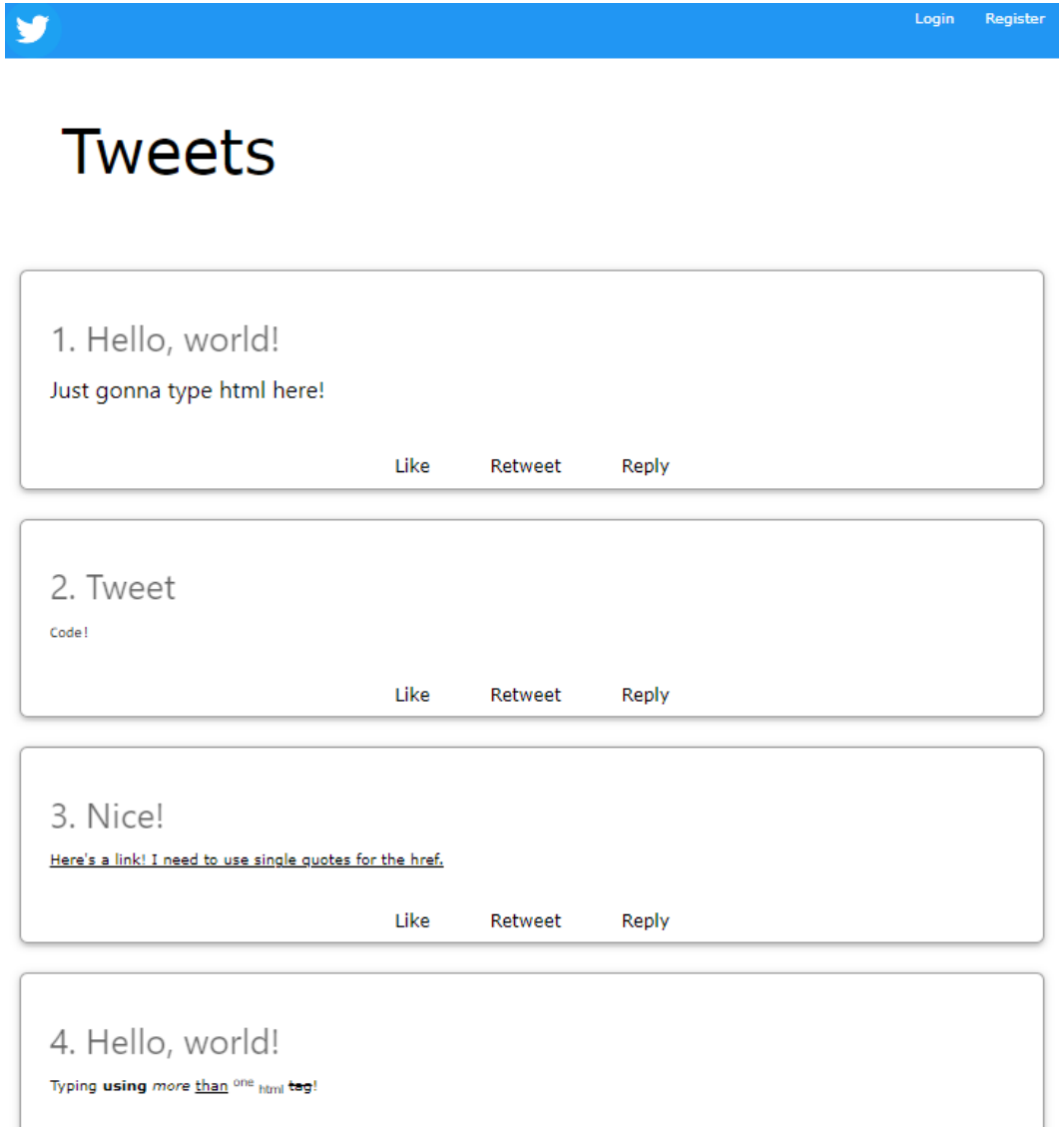


Connexion between front and back

- Make a connexon between front and back end using request

tweet

- Show tweets like this



Do a powerpoint presentation of your work

- Present your work
- Give some details about data structure
- Demonstrate the website

Instructions for the presentation

- Take 3 minutes to make a demo of the website on your laptop
- Explain your data structure decision in 2 mn **without any slide**!
- Leave 4 minutes for Question and answers
- Your instructor will warn you after 5 and 8 minutes
- Presentation stop after 9 minutes

- **Send us after the presentation within a day your final code.**

Some tips and advices

- Do not start at the last minute!
- If you have technical problems, liaise with the rest of the class and let us know who managed to help you!
- Work as a group and not individually!
- Test before the presentation that everything runs well on your computer to avoid blank presentation in the due day!