

PROBLEM SET 5

This graded problem set (9 points total) is due Sunday, April 2, 2023.

5.1 - SQL (2PT.)

Django offers a level of abstraction on top of SQL. Not only does it make it easier for us to add, change or delete data in a database, but it also makes our applications a lot safer, most notably preventing SQL injections attacks.

Every function call on our models in Django is converted into a raw SQL statement that is executed. Conversely, we can look at some SQL statements and determine, what the corresponding function call (or model) in Django would be. It is advisable to look at the official documentation, as is linked to for each subtask.

1. The SQL statement below creates a table named “classrooms”. What could a corresponding model in Django look like?

<https://docs.djangoproject.com/en/4.1/topics/db/models/>

```
CREATE TABLE classrooms(  
  id INTEGER PRIMARY KEY AUTOINCREMENT,  
  roomname TEXT NOT NULL,  
  seats INTEGER DEFAULT 50 NOT NULL  
)
```

2. The SQL statements below fills the table “classrooms” with some rows of data. How would this be achieved in Django?

<https://docs.djangoproject.com/en/4.1/topics/db/queries/>

```
INSERT INTO classrooms (roomname, seats) VALUES ('A101', 42);  
INSERT INTO classrooms (roomname) VALUES ('A102');
```

3. The SQL statements below retrieves data from the classrooms table. How would this be achieved in Django?

<https://docs.djangoproject.com/en/4.1/topics/db/queries/#making-queries>

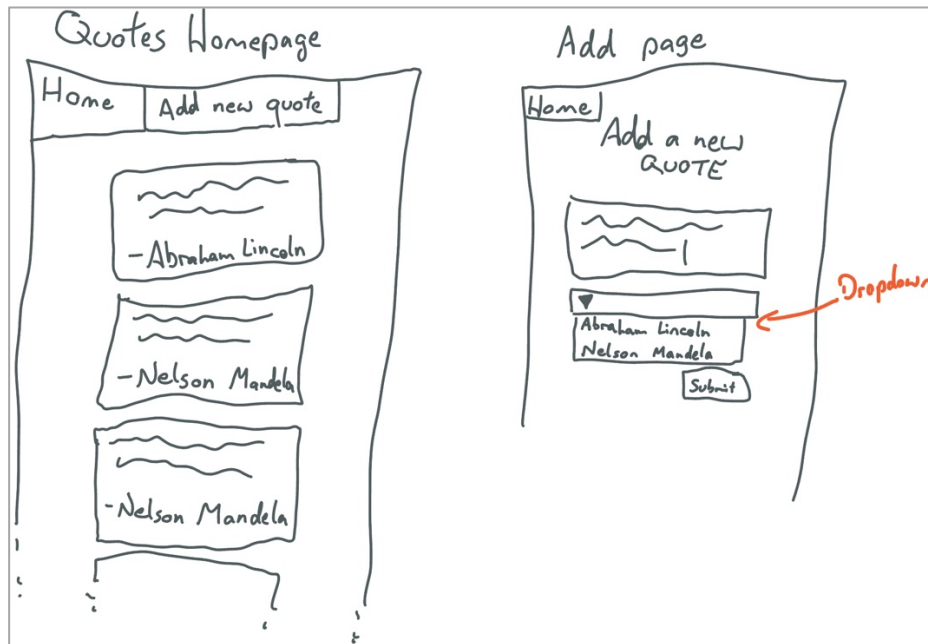
```
SELECT * FROM classrooms WHERE seats > 45;
```

5.2 - MORE SOPHISTICATED JOKES (4P.)

In this exercise, you will extend your jokes website from problem set 4 such that it uses models and a database. If your current project doesn't work well for you, you are free to use this GitHub repository to start off with: <https://github.com/DauphineWeb/BaseJoke>

Your app should change in the following ways:

1. Instead of jokes it now presents users with quotes from well-known people.



(any styling remains optional, though it's highly encouraged)

2. **Two models** should be created,
 - one for the list of people to be quoted, and
 - one for the list of quotes.It should model a one-to-many relationship, as a person may have multiple quotes attributed to them, but a quote cannot be attributed to multiple people.
3. In the admin page, create a few authors.
4. Your website intended to add new jokes should be adapted accordingly such that it now adds quotes to a database instead.
The input field that required a name should be changed to a **dropdown menu** where a user can choose from the list of persons in the database.
(Adding new authors does not have to be supported.)
5. Change the homepage such that it lists all quotes from the database.
6. Update your route that returned a JSON representation of a joke such that it now returns a JSON representation of a quote for a given quote id.

5.3 - GROUP PROJECT (3P.)

Add user authentication to your website.

As a baseline, make sure the following features are implemented:

- A login form for users to authenticate.
- A logout link for users who are logged in.
- A signup form for new users to register.
- If a user is logged in, the login page should be inaccessible.
- Sites that are not intended to be seen by users without an account should be protected, either by redirecting the user to the login page or by showing an error message.

No matter what web framework you have chosen for your project, make sure to follow best practices to securely authenticate users. This means, for example, **not saving a password as plain text** in the database. For Django's built-in authentication system, see here: <https://docs.djangoproject.com/en/4.1/topics/auth/>

You are free to add, i.e., a user settings page for users to change their passwords or personal information. You may also begin modeling the structure of your database the way you see fit for your project.

Make sure to add the db.sqlite3 file to the .gitignore file.

Submit your solutions for 5.1 and 5.2 preferably via OneDrive, alternatively via GitHub, by April 2, 2023, to felix.fritz@dauphine.psl.eu.

For exercise 5.3, I will check by your GitHub repository after the submission date.

Lastly, do not hesitate to contact me if you experience trouble with the problem set.