**Mój set - PA**

**1. Create a page that lists all shows’: title, rating of a show and third season’s overview - if show doesn’t have third season, don’t include that show in this view**

**2. Add a button on top of the page with text “Actor of the day”. Onclick it should display an alert with randomly selected name of the actor from the list created in previous point. It should be displayed with 3 seconds of delay.**

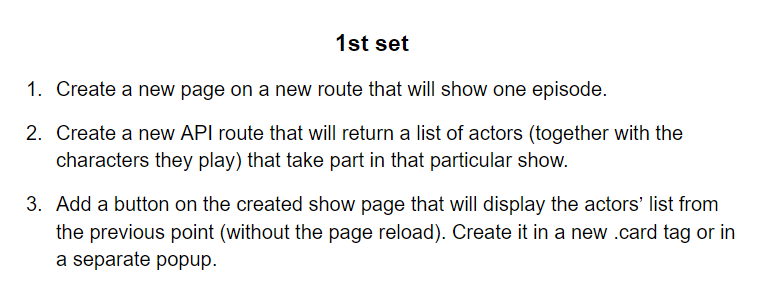
**3. Add a link to the actor name that will open a new page with a form that will enable to change the actor name. After submitting the form the name should be changed in the database.**

**@app.route('/shows3season', methods=['GET'])**

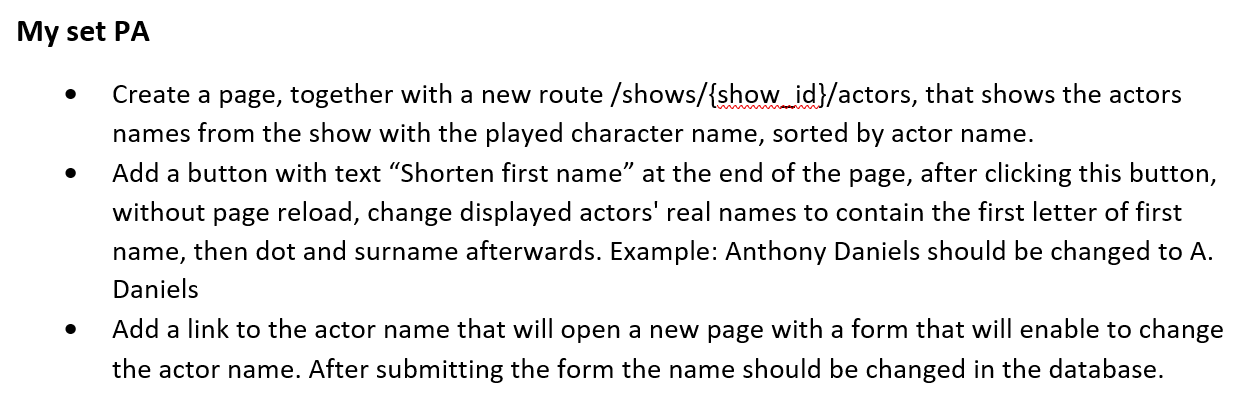
**@app.route('/actorofaday', methods=['GET', 'POST'])**

**@app.route('/shows/<int:show\_id>/actors', methods=['GET'])**

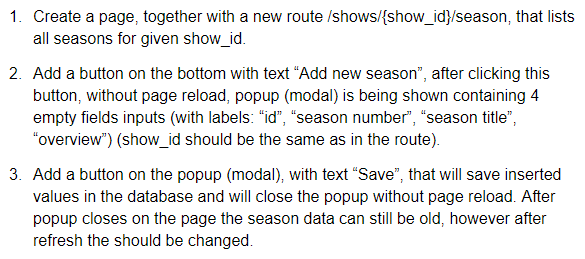
# new route to one show

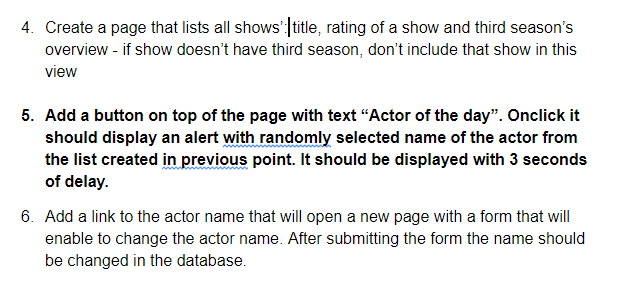


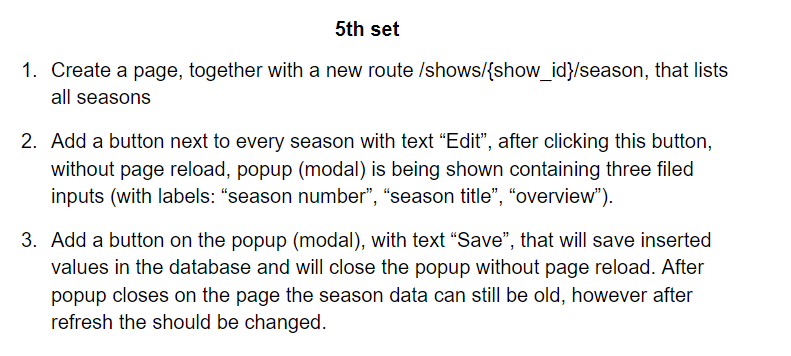
# editing actor's name



# list all show seasons







**--question 1 - show one EPISODE**

select shows.id, shows.title, episodes.title as Episode

from shows

left join seasons on shows.id = seasons.show\_id

left join episodes on seasons.id = episodes.season\_id

where shows.id = %(id)s

order by random()

limit 1;

**-- question 2 -part!!!**

select show\_characters.character\_name as role, actors.name as real\_name

from shows

left join show\_characters on shows.id = show\_characters.show\_id

left join actors on show\_characters.actor\_id = actors.id

where shows.id = %(id)s; --dodać fetchall=True do funkcji w queries!!!!!!!!!!!!!!!!!

**--question 1 -- first 20 actors from db ordered by name**

select name from actors

order by name

limit 20;

**-- question 2show actors shows he/she played**

select actors.name, shows.title

from shows

left join show\_characters on shows.id = show\_characters.show\_id

left join actors on show\_characters.actor\_id = actors.id

where actors.id %(id)s;

**-- question 3 link to the series details by name or by id be careful**

select shows.title as Show\_Title, seasons.season\_number, seasons.title, shows.runtime, shows.rating, shows.year, seasons.overview

from shows

left join seasons on shows.id = seasons.show\_id

-- where shows.title LIKE 'Gilmore Girls';

-- where shows.id = %(id)s;

**question 1**

select shows.title , season\_number, seasons.overview from shows

left join seasons

on shows.id = seasons.show\_id

order by seasons.id;

Question 1:

def get\_characters\_from\_show(id):

return data\_manager.execute\_select("""

SELECT sc.id as c\_id, a.id as a\_id, character\_name as cn, name

FROM actors a

JOIN show\_characters sc on a.id = sc.actor\_id

WHERE show\_id = %(id)s

ORDER BY cn;

""", {"id": id})

**Qestion 2**

let ren1 = document.getElementById('btn1');

let ren2 = document.getElementById('btn2');

ren1.addEventListener('click', rename1);

ren2.addEventListener('click', rename2);

// let acc = document.querySelector('.nn');

let names = document.querySelectorAll('td');

// actors name

function rename1() {

let i;

for (i = 0; i < names.length; i = i + 2){

console.log(names[i].innerText);

let a = names[i].innerText;

let result = a.split(' ');

// console.log(names[i].innerText = result[0][0]+". "+result[1]);

console.log(names[i].innerHTML = result[0][0]+". "+result[1]);

}

}

// real name

function rename2() {

let i;

for (i = 1; i < names.length; i = i + 2){

console.log(names[i].innerText);

let a = names[i].innerText;

let result = a.split(' ');

// console.log(names[i].innerText = result[0][0]+". "+result[1]);

console.log(names[i].innerHTML = result[0][0]+". "+result[1]);

}

}

########################################

Question 3:

def update\_character(c\_id, new\_name):

data\_manager.execute\_dml\_statement(sql.SQL("""

UPDATE show\_characters

SET character\_name = %(new\_name)s where show\_characters.id = %(c\_id)s;

"""), {"c\_id": c\_id, "new\_name": new\_name})