

**DOMAĆI ZADATAK: 6-7**

Web sistemi I tehnologije

|  |  |
| --- | --- |
| Ime i prezime: | **Nikola Dimic** |
| Broj indeksa | **S46/19** |
| Datum izrade | **16.12.2020** |



Za domaci zadatak broj 6-7 potrebno je da student napravi raspored casova kao na skolskom sajtu, tako sto ce koristi python, modul flask i jquery. Podatke o rasporedu smo dobili u obliku sql dokumenta.

Glavni cilj ovog zadataka je da mozemo da upotrebimo live pretragu u tabeli i iz druge tabele kada kliknemo na neku stavku, da nam u prvom tabeli prikaze sve potrebne detalje u vezi tog elementa na koji smo kliknuli.

# Rešenje domaćeg zadatka:

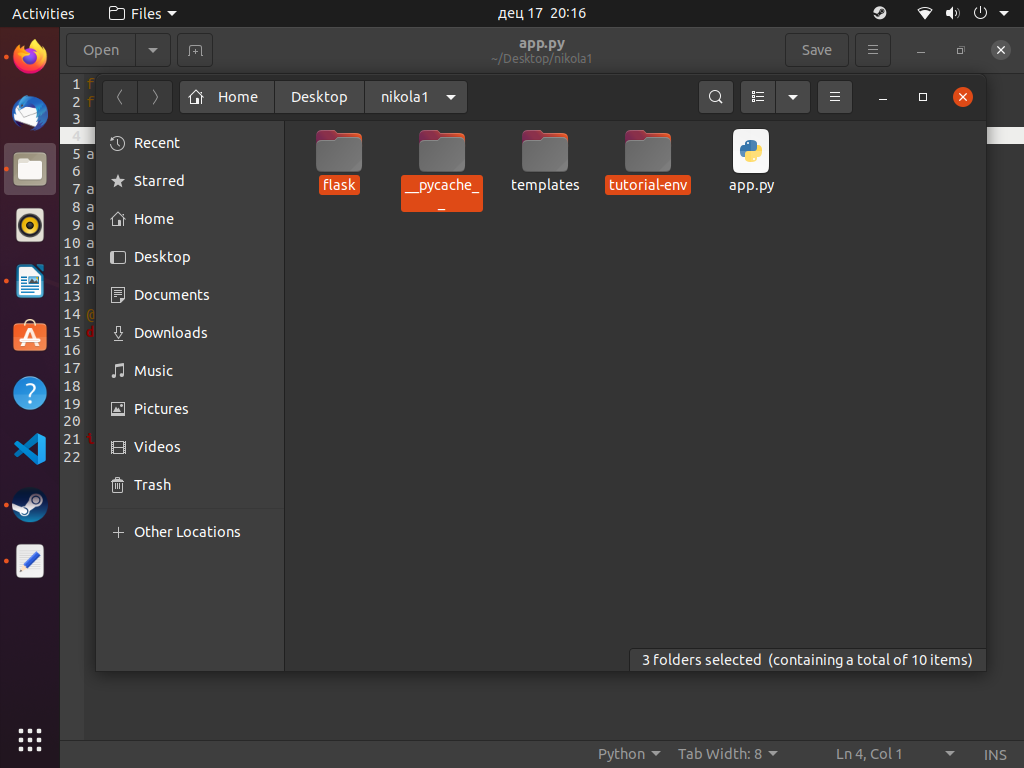
Za rešavanje prvog dela domaćeg zadatka korišćeno je gotovo rešenje “Lightbox” (autor: Lokesh Dhakar). Preko CSS-a napravljen je okvir koji je podeljen na 5 jednakih kolona i dva jednaka reda. Iznad njega je dodat tekst i naslov.

Prvo sta sam napravio, uradio sam html (index.html) stranicu preko bootstrapa I datatables-a, napravio sam tabelu sa potrebnim stavkama (id, nastavnik, vreme, grupa, dan.. ) I ispod toga jos jednu mini tabelu sa dve kolone (nastavnik I vreme ). Samim tim sto sam radio preko bootstrapa I datatables-a, imao sam gotovu jquery biblioteku koju sam uveo u svoj fajl I dobio live pretragu za tabelu, ali u tom momentu nismo mogli pretrazivati jer jos uvek nisam uvezao podatke iz sql dokumenta.

Nakon toga instaliram sve sto je potrebno preko linuxovog terminala I visual studio code-ovog terminala.   
- flask

- flask\_mysqldb  
- sql  
- apache  
- phpmyadmin

Prvi deo kada sam instalirao sve to, bilo je potrebno aktivirati flask, kako bi on uvezao svoje fajlove koji su nam potrebni.



Nakon toga sam u novom otvorenom fajlu app.py otkucao code koji se povzeuje sa mojoj stranicom index.html

from flask import Flask, render\_template, json, request, redirect

app = Flask(\_\_name\_\_)

@app.route("/")

def index():

return render\_template('index.html')

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)

Nakon povezivanja stranica, potrebno je bilo da sposjim svoju app.py sa bazom koju sam importovao na phpmyadmin, na localhost

app.config['MYSQL\_HOST'] = 'localhost'

app.config['MYSQL\_USER'] = 'root'

app.config['MYSQL\_PASSWORD'] = 'Nikola!123'

app.config['MYSQL\_DB'] = 'raspored'

app.config['MYSQL\_CURSORCLASS'] = 'DictCursor'

mysql = MySQL(app)

Ovim kodom smo se povezali sa bazom podataka (raspored.sql)

Sledeci cilj je bio uvesti iz baze podatke u nase tabele.

@app.route("/")

def index():

cur = mysql.connection.cursor()

cur.execute("SELECT \* FROM raspored")

raspored = cur.fetchall()

return render\_template('index.html', raspored=raspored)

A u index.html stranici smo otkucali kod koji ce povezati kod sa tabelom I uneti tacno podatke gde smo zeleli:

{% for row in raspored %}

<tr>

<th>{{row.id}}</th>

<th>{{row.predmet}}</th>

<th>{{row.tip}}</th>

<th>{{row.nastavnik}}</th>

<th>{{row.grupe}}</th>

<th>{{row.dan}}</th>

<th>{{row.termin}}</th>

<th>{{row.vreme}}</th>

</tr>

{% endfor %}

Isto tako smo uradili I sa drugom tabelom.

Poslednji deo zadatka, klikom na element u drugoj tabeli treba da nam da rezultate u prvom. To smo uradili preko jquery-ja. Tako sto kliknemo na element, on ce nam klikom odradjivati funkciju koja ce raditi kao da smo to uneli u search polje koje imamo na vrhu I raditi pretragu u prvoj tabeli.

{% for row in raspored %}

<tr>

<th><span **onclick="onFilter(this)"**>{{row.nastavnik}}</span></th>

<th><span **onclick="onFilter(this)**">{{row.vreme}}</span></th>

</tr>

{% endfor %}

<script>

function onFilter(value) {

console.log(value.textContent);

var table = $('#example').DataTable();

table.search( value.textContent ).draw();

}

</script>

A izgled koji je zahtevao zadatak njega sam izabrao u datatables-u, a boje koje su mi trebale, samo sam uradio u html-u preko style-a.