



TopFood

Esprimi i tuoi gusti

Dragone Giuseppe
Mat 0124/318

Perché TopFood?

Trovarsi in un locale ed avere a che fare con tantissime voci nel menù ed avere paura di scegliere una voce non proprio nelle corde del locale non è cosa gradita.

Hai intenzione di cenare in un locale e voler provare il suo “cavallo di battaglia” ma vuoi informazioni a riguardo prima di entrare??

Con TopFood si può!



Funzionalità



TopFood permette di andare a stilare una classifica dei piatti che offre un locale sulla base del giudizio degli utenti attraverso una valutazione da 1 a 5.

L'utente si registra, registra il proprio scontrino e effettua la valutazione su ciò che ha ordinato.

Sarà presente l'elenco dei locali, selezionando uno di questi verranno visualizzate le voci del menù che saranno formate dal nome del piatto, dagli ingredienti e dalla media delle valutazioni.

TopFood non è utile solo al cliente ma anche al gestore del locale, in quanto diventa uno strumento attraverso il quale il gestore può prendere coscienza dei problemi sul menu e capire su quali piatti migliorarsi.

Funzionalità

Un' ulteriore possibile funzione è la ricerca di un piatto comune a più locali ed attraverso la localizzazione ricercare quale dei ristoranti prossimi ha il giudizio migliore.

Utilizzabile anche dai locali che servono drink e cocktail.

“

*Non si può pensare bene, amare bene,
dormire bene, se non si ha mangiato
bene.*

”

cit. Virginia Woolf

Lamp

le componenti software con cui è realizzata l'applicazione web sono:

- Linux
- Apache
- MongoDB
- Python

Nel dettaglio: SO Ubuntu LTS 16.04, server apache 2, database nosql mongodb v 3.4.10, Python con microframework Flask e pymongo, ospitato su Cloud Hosting AWS

Flask

```
from flask import Flask,render_template,url_for,request,session,redirect
import os
from pymongo import MongoClient
from bson.objectid import ObjectId
from media import media_rank
import bcrypt

def connect():
    connection = MongoClient("ds163016.mlab.com",63016)
    handle = connection["my_mongo1"]
    handle.authenticate("peppe","peppe10")
    return handle

app=Flask(__name__)
app.secret_key='onepiece'

mongo = connect()
```

importiamo tutte le estensioni che ci servono

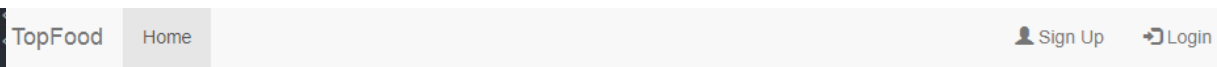
connessione al database mongodb

nome per riferimento appFlask

collegamento al database

Templates - Layout

Head

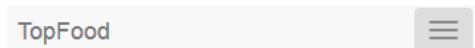


Esprimi i tuoi gusti

Cerca nella tua città i locali aderenti

Ricerca Ristorante

Ricerca in base al piatto desiderato



Home

flavio

Log Out

Esprimi i tuoi gusti

Cerca nella tua città i locali aderenti

Ricerca Ristorante

Ricerca in base al piatto desiderato

Flask-Login

```
@app.route('/')
def home():
    if 'username' in session:
        return render_template('home.html',user=session['username'])
    return render_template('home.html')

@app.route('/login')
def login():
    return render_template('login.html')

@app.route('/login_exe',methods=['POST'])
def login_exe():
    utenti=mongo.utenti
    login_user=utenti.find_one({'username':request.form['username']})
    if login_user:
        if bcrypt.hashpw(request.form['pass'].encode('utf-8'), login_user['password'].encode('utf-8')) == login_user['password'].encode('utf-8'):
            session['username'] = request.form['username']
            return redirect(url_for('home'))
    return render_template('login.html',fail=True)

@app.route('/logout')
def logout():
    session.pop('username',None)
    return redirect(url_for('login'))
```

Gestione del login\logout e sessione utente, credenziali ricevute tramite form, username e password, password criptata e poi confrontata

Templates- Login

```
{%extends "layout.html" %}
{% block content %}

<div class="container">
  {% if fail %}
    <div class="alert alert-danger">
      <strong>Attenzione!</strong> Credenziali Non Valide.
    </div>
  {% endif %}
  <div class="row">
    <div class="col-md-4 col-md-offset-4">
      <form method="POST" action="{{url_for('login_exe')}}">
        <div class="form-group">
          <label for="Utente">Username</label>
          <input type="text" name="username" class="form-control" placeholder="Username" required>
        </div>
        <div class="form-group">
          <label for="Password">Password</label>
          <input type="password" name="pass" class="form-control" placeholder="Password" required>
        </div>
        <button type="submit" class="btn btn-primary btn-block">Login</button>
      </form>
      <br>
      <div class="">
        Non hai un account? <a href="{{url_for('register')}}">Registrati</a>
      </div>
    </div>
  </div>
</div>

{% endblock %}
```

Username

Password

Login

Non hai un account? [Registrati](#)

Attenzione! Credenziali Non Valide.

Username

Password

Login

Non hai un account? [Registrati](#)

Flask - Ricerca

```
@app.route('/local',methods=['POST'])
def local():
    city_s=mongo.locali.find_one({'city':request.form['city']})
    city=mongo.locali.find({'city':request.form['city']})

    if city_s is None:
        if 'username' in session:
            return render_template('home.html',user=session['username'],serch=True)
        return render_template('home.html',serch=True)

    if 'username' in session:
        return render_template('locali_geo.html',user=session['username'],locali=city)
    return render_template('locali_geo.html',locali=city)
```

```
@app.route('/ricerca piatto',methods=['POST'])
def s_piatto():
    menu_s=mongo.menu.find_one({'nome_p':request.form['piatto']})
    menu_x=mongo.menu.find({'nome_p':request.form['piatto']})

    if menu_s is None:
        if 'username' in session:
            return render_template('home.html',user=session['username'],serchF=True)
        return render_template('home.html',serchF=True)

    if 'username' in session:
        return render_template('locali_geo.html',user=session['username'],locali=menu_x,serch_p=True)
    return render_template('locali_geo.html',locali=menu_x,serch_p=True)
```

Ricerca Tramite città o piatto desiderato, verifica se l'utente ha effettuato l'accesso

Templates- Ricerca/ Locali

```
{% extends "layout.html" %}
{% block content %}
```

Esprimi i tuoi gusti

Cerca nella tua città i locali aderenti

Ricerca Ristorante

Ricerca in base al piatto desiderato

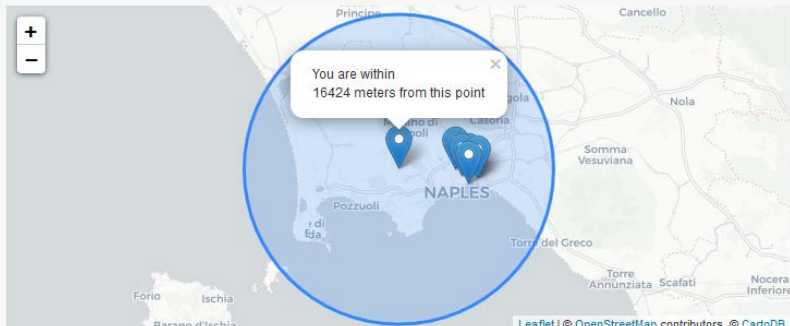
```
<label for="SerchCity">Ricerca Ristorante</label>
<input type="text" name="city" class="form-control" placeholder="Inserisci città">
</div>
</form>
{%if serchF%
<div class="
<strong>Warning!!Nessun locale nella città
</div>
{%endif%
<form class="
<div class="
<label for="
<input type="
</div>
</form>
</div>
{% endblock %}
```

Warning!!Nessun locale nella città

Ricerca Ristorante

```
{% extends "layout.html" %}
{% block content %}
```

Locali



La Masardona

De figliole

Pizzeria da Michele

Sorbillo

Stanita

```
<h1>Locali
<div id="m
<script
var ma
var la
map.ad
map.le
functi
va
L.
L.
}
map.on(
</script>
<div cla
{% for
{%if
{%
{% e
<s
</
{% end
</div>
{% endblock %}
```

Flask - Sig In/ menù

```
@app.route('/register',methods=['POST','GET'])
def register():
    if request.method == 'POST':
        utenti=mongo.utenti
        gia_utente=utenti.find_one({'$or':[{'username':request.form['username']},{'email':request.form['email']}]})

        if gia_utente is None:
            hashpass = bcrypt.hashpw(request.form['pass'].encode('utf-8'), bcrypt.gensalt())
            utenti.insert({'username' : request.form['username'], 'password' : hashpass, 'email':request.form['email']})
            session['username'] = request.form['username']
            return redirect(url_for('home'))

        return render_template('register.html',fail=True)

    return render_template('register.html')
```

Riceve i dati tramite form
e verifica l'univocità
dell'username

effettua criptazione password

restituisce i vari risultati

```
@app.route('/menu/<path:nome>/<path:locale>',methods=['GET'])
def menu(nome,locale):
    menu_l=mongo.menu.find({'locale':ObjectId(locale)})
    locale=mongo.locali.find({'$or':[{'_id':ObjectId(locale)},{'locale':ObjectId(locale)}]})
    if 'username' in session:
        return render_template('menu.html',menu=menu_l,nome=nome,user=session['username'],locale=locale)

    return render_template('menu.html',menu=menu_l,nome=nome,locale=locale)
```

invia menu del locale con id
corrispondente

Flask - Votazione

def voto: Verifica se l'utente ha già votato quel piatto, aggiorna il valore calcolando la media tramite funzione

```
@app.route('/voto/<path:nome>/<path:locale>/<path:piatto>', methods=['POST'])
def voto(nome, locale, piatto):
    lista_voti=mongo.voti
    gia_votato=lista_voti.find_one({'utente':session['username'], 'nome_p':piatto, 'locale':ObjectId(locale)})
    menu=mongo.menu
    menu_l=mongo.menu.find({'locale':ObjectId(locale)})
    vot_piat=menu.find_one({'locale':ObjectId(locale), 'nome_p':piatto, 'rating':0})
    voto_x=int(request.form['voto'])

    if vot_piat is None and gia_votato is None:
        vol=media_rank(menu, locale, piatto, voto_x)
        lista_voti.insert({'utente':session['username'], 'locale':ObjectId(locale), 'nome_l':nome, 'nome_p':piatto, 'voto':int(request.form['voto'])})
        menu.update({'locale':ObjectId(locale), 'nome_p':piatto}, {'$inc':{'voti':1, 's'+request.form['voto']:1}})
        menu.update({'locale':ObjectId(locale), 'nome_p':piatto}, {'$set':{'rating':vol}})
        return render_template('menu.html', menu=menu_l, nome=nome, user=session['username'])
    elif gia_votato is None:
        lista_voti.insert({'utente':session['username'], 'locale':ObjectId(locale), 'nome_l':nome, 'nome_p':piatto, 'voto':int(request.form['voto'])})
        menu.update({'locale':ObjectId(locale), 'nome_p':piatto}, {'$inc':{'voti':1, 's'+request.form['voto']:1, 'rating':float(request.form['voto'])}})
        return render_template('menu.html', menu=menu_l, nome=nome, user=session['username'])
    else:
        return render_template('menu.html', menu=menu_l, nome=nome, user=session['username'], votato=True)
```

```
def media_rank(menu, locale, piatto, voto_x):

    voti=menu.find({'locale':ObjectId(locale), 'nome_p':piatto}).distinct('voti')
    s1=menu.find({'locale':ObjectId(locale), 'nome_p':piatto}).distinct('s1')
    s2=menu.find({'locale':ObjectId(locale), 'nome_p':piatto}).distinct('s2')
    s3=menu.find({'locale':ObjectId(locale), 'nome_p':piatto}).distinct('s3')
    s4=menu.find({'locale':ObjectId(locale), 'nome_p':piatto}).distinct('s4')
    s5=menu.find({'locale':ObjectId(locale), 'nome_p':piatto}).distinct('s5')

    num=(float(s1[0])+float(s2[0])*2+float(s3[0])*3+float(s4[0])*4+float(s5[0])*5+voto_x)/(float(voti[0])+1)

    rating="{0:.1f}".format(num)

    return rating
```

def media_rank:
Recupera i voti e ne calcola la media, effettua una conversione del tipo e restituisce il valore con un solo valore dopo la virgola

Templates - Sig In

```
{% extends "layout.html" %}
{% block content %}

<div class="container">
  {% if fail %}
  <p>Credenziali Gia Presenti</p>
  {% endif %}
  <div class="row">
    <div class="col-md-4 col-md-offset-4">
      <form action="{{url_for('register')}}" method=POST>
        <div class="form-group">
          <label for="Utente">Username</label>
          <input type="text" name="username" class="form-control" placeholder="Username" required>
        </div>
        <div class="form-group">
          <label for="Password">Password</label>
          <input type="password" name="pass" class="form-control" placeholder="Password" required>
        </div>
        <div class="form-group">
          <label for="email">e-mail</label>
          <input type="email" name="email" class="form-control" placeholder="email" required>
        </div>
        <button type="submit" class="btn btn-primary btn-block">Registra</button>
      </form>
    </div>
  </div>
</div>

{% endblock %}
```

Username

Password

e-mail

Registra

Credenziali Gia Presenti

Username

Password

e-mail



Aggiungi un simbolo "@" nell'indirizzo email. In "derwl.cf" manca un simbolo "@".

Templates- menu

La Masardona

Via Giulio Cesare Capaccio,27

80127

Tel:081281057

Sitoweb: www.facebook.com/lamasardona/

cat: pizzeria italiana vegetariani



Pizza Fritta Completa

ricotta, cicoli, provola, pomodoro, pepe, basilico,

3.8★

6 Totali

vota:

1

invia voto



Veganina

scarole saltate in padella(uvetta,pinoli,peperoncino), olive nere di gaeta, provola,

0★

1 Totali

vota:

1

invia voto

3.8★

6 Totali

Accedi \ Registra

Per votare effettua il Login o iscriviti

invia voto

Info Locali

<div class="container">

Veganina

scarole saltate in padella(uvetta,pinoli,peperoncino), olive nere di gaeta, provola,

3.0★

1 Totali

vota:

1

invia voto

href="#"