

ORGANIZADOR DE TAREAS

...

INTRODUCCIÓN:

A menudo los estudiantes sufren de estrés y pasan por muchos problemas en sus cursos debido a su mala organización y el tiempo que le dedican a cosas innecesarias, entonces para eso he desarrollado una aplicación web en donde cualquier usuario pueda ver sus tareas o notas que cumplir y no tener que estar guardandolas en su cabeza.

PARTES IMPORTANTES DEL CÓDIGO:

En esta parte del código estoy creando la base de datos y mi esquema va a contener la tabla que en este caso lo declaré como clase ya que almaceno todos los datos y de ahí los voy a utilizar.

```
3
4 class Users(db.Model):
5
6     id= db.Column(db.Integer,primary_key=True)
7     username=db.Column(db.String(30),unique=True,nullable=False)
8     email=db.Column(db.String(50),unique=True,nullable=False)
9     password=db.Column(db.String(80),nullable=False)
10
11 class Tareas(db.Model):
12     username=db.Column(db.String(30),unique=True,nullable=False,primary_key=True)
13     tarea_1=db.Column(db.String(50),unique=True)
14     tarea_2=db.Column(db.String(50),unique=True)
15     tarea_3=db.Column(db.String(50),unique=True)
16     tarea_4=db.Column(db.String(50),unique=True)
17     tarea_5=db.Column(db.String(50),unique=True)
18     tarea_6=db.Column(db.String(50),unique=True)
19     tarea_7=db.Column(db.String(50),unique=True)
20     tarea_8=db.Column(db.String(50),unique=True)
21     tarea_9=db.Column(db.String(50),unique=True)
22     tarea_10=db.Column(db.String(50),unique=True)
23
```

Esta parte del código recibe una POST de parte de registrarse.html con un form con los datos username,email y password. Entonces si recibe este va a preguntar si los datos estan vacios y te mandara un mensaje emergente en caso sea cierto,sino te registrara los datos y te cifrara la contraseña.

```
@app.route('/sign_up',methods=['GET','POST'])
def sign_up():
    session.pop(["username",None])
    if request.method == "POST":
        if request.form["username"]=="" or request.form["email"]=="" or request.form["password"]=="":
            flash('Los campos no deben estar vacios para que te registres')
        else:
            hashed_pw = generate_password_hash(request.form["password"],method="sha256")#sha256 es un metodo de cifrado
            new_user=Users(username=request.form["username"],email=request.form["email"],password=hashed_pw)
            new_user_tar=Tareas(username=request.form["username"],tarea_1=None,tarea_2=None,tarea_3=None,tarea_4=None,tarea_5=None,
            tarea_6=None,tarea_7=None,tarea_8=None,tarea_9=None,tarea_10=None)
            db.session.add(new_user)
            db.session.add(new_user_tar)
            db.session.commit()
            flash('REGISTRADO CORRECTAMENTE')

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

En el login lo primero que hago es hacer un pop a la session para que cuando entres a la ruta login la sesión anterior no esté guardada. luego recojo los datos del form que me envia el login.html y hago una query que me pase los datos de acuerdo al username que recibo formulario.

```
@app.route('/login',methods=['GET','POST'])
def login():
    session.pop("username",None)
    if request.method=="POST":
        user=Users.query.filter_by(username=request.form["username"]).first()
        if user and check_password_hash(user.password,request.form["password"]):
            session['username']=user.username
            return redirect(url_for('tareass_app'))
        flash('CREDENCIALES INVALIDAS')

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

Esta ruta sirve para crear las tareas lo que hago aquí es eliminar la fila en donde estaba el usuario y la renovava con nuevas tareas insertadas dependiendo de qué tarea sea

```
@app.route('/tareas_app/Crear_nueva_tarea',methods=['GET','POST'])
def Crear_nueva_tarea():
    if "username" in session:
        nueva_tarea=True
        user=Tareas.query.filter_by(username=session["username"]).first()
        if request.method=='POST':
            if(user.tarea_1!=""):
                new_user=Tareas(username=session["username"],tarea_1=request.form["description"],tarea_2=user.tarea_2,tarea_3=user.tarea_3,tarea_4=user.tarea_4,
                tarea_5=user.tarea_5,tarea_6=user.tarea_6,tarea_7=user.tarea_7,tarea_8=user.tarea_8,tarea_9=user.tarea_1,tarea_10=user.tarea_1)
                db.session.add(new_user)
                db.session.delete(user)
                db.session.commit()
            elif(user.tarea_2!=""):
                new_user=Tareas(username=session["username"],tarea_1=user.tarea_1,tarea_2=request.form["description"],tarea_3=user.tarea_3,tarea_4=user.tarea_4,
                tarea_5=user.tarea_1,tarea_6=user.tarea_1,tarea_7=user.tarea_1,tarea_8=user.tarea_1,tarea_9=user.tarea_1,tarea_10=user.tarea_1)
                db.session.add(new_user)
                db.session.delete(user)
                db.session.commit()
            elif(user.tarea_3!=""):
                new_user=Tareas(username=session["username"],tarea_1=user.tarea_1,tarea_2=user.tarea_2,tarea_3=request.form["description"],tarea_4=user.tarea_4,
                tarea_5=user.tarea_5,tarea_6=user.tarea_6,tarea_7=user.tarea_7,tarea_8=user.tarea_8,tarea_9=user.tarea_9,tarea_10=user.tarea_10)
                db.session.add(new_user)
                db.session.delete(user)
                db.session.commit()
            elif(user.tarea_4!=""):
                new_user=Tareas(username=session["username"],tarea_1=user.tarea_1,tarea_2=user.tarea_2,tarea_3=user.tarea_3,tarea_4=request.form["description"],
                tarea_5=user.tarea_4,tarea_6=user.tarea_6,tarea_7=user.tarea_7,tarea_8=user.tarea_8,tarea_9=user.tarea_9,tarea_10=user.tarea_10)
```