**NEPTUNOMAX**

**Proyecto de capacitación**

**Creadores:Tomas Molina y Mauro Olmos**

**Presentación:**

**NeptunoMax** es una página web en la cual se trata de que puedas ver películas,y te garantizamos películas de buena calidad espectaculares,y no sólo películas si no también series de todo tipo, y puedes ver los trailer de último momento para que te enteres lo que va a pasar o saber que se trata.

**PASO A PASO:**

En la página inicial podemos encontrar el navbar que está arriba y a la izquierda podemos encontrar el logo o icono de la página que la hicimos nosotros con ayuda de la ia(Inteligencia Artificial) al medio podemos ver el nombre de nuestra página y el la derecha veremos los tres puntos el la cual si nosotros le damos clic se abrirá una pestaña o ventanita a la derecha que podemos ver que dice películas(se verá todas las películas que ahi),series(por el momento no es disponible pero pronto estará),las categorías que podemos encontrar acción , terror, comedia, infantil, anime, doramas, novelas,etc… y más abajo estará para iniciar sesión.

Siguiendo en la página inicial podemos ver unas fotos de las películas en cual se mueve automáticamente cada 4 segundos las fotos y aparecerá una nueva con la descripción de la película,más abajo podemos ver las cartas que está compuesta por una portada de la película que podemos ver.

Al entrar a una película podemos ver la portada y la descripción de la película abajo esta el URL donde podemos ver el video y reproducirlo sin ningún tipo de problema y con todo tipo de películas o serie.

**INICIAR SESIÓN**:

cuando le das iniciar sesión te aparecerá la pestaña del login en la cual está compuesta por el formato del lenguaje con django y MySQL te decterara si hiciste una cuenta o si ya la habías iniciado y si no tienes la cuenta no va a pasar nada solamente te va a sugerir que te registres primero .si ya tienes te va a aparecer el inicio de la página y podrás ver todo el contenido

**REGISTRO DE USUARIO:**

cuando entramos veremos casi el mismo formato que el iniciar sesión,tienes que completar el registro o las casillas completa con toda tu información y datos como ,el nombre de usuario,nombre real, apellido, dni, email,   contraseña y todo se guardara en la base de dato de django.

**LENGUAJE**:

Usamos para la base de datos el MYSQL junto con Python que en la cual nos deja registrar las personas registras y las que ya se registraron, **devproject/settings.py:** acá utilizamos para la configuración de la base de datos, configuración de las rutas y URL, configuración de las aplicaciones y la configuración de plantillas, etc...

Ejemplo:

INSTALLED\_APPS **=** [

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',

    'myapp', #add myapp

]

DATABASES **=** {

    'default': {

        'ENGINE': 'django.db.backends.mysql',

        'NAME': 'django',

        'USER': 'root',

        'PASSWORD': '',

        'HOST': 'localhost',

        'PORT': '3306'

    }

}

En models.py: acá lo que hicimos fue poner los name a cada input que los registra a los usuarios. Ejemplo:

**from** django.db **import** models

# Create your models here.

**class** Employee(models.Model):

    name **=** models.CharField(max\_length**=**100)

    email **=** models.EmailField()

    contact **=** models.CharField(max\_length**=**15)

**class** Meta:

        db\_table **=** "tblemployee"

FORMS: Acá pudimos crear los Forms para que se conectara con el models.py ejemplo.

**from** django **import** forms

**from** myapp.models **import** Employee

**class** EmployeeForm(forms.ModelForm):

**class** Meta:

        model **=** Employee

        fields **=** ['name', 'contact', 'email']

        widgets **=** { 'name': forms.TextInput(attrs**=**{ 'class': 'form-control' }),

            'email': forms.EmailInput(attrs**=**{ 'class': 'form-control' }),

            'contact': forms.TextInput(attrs**=**{ 'class': 'form-control' }),

      }

El views: nos sirvió para añadir cosas en el html de forma más compleja pero mejor para poner una base. Ejemplo

def my\_view(request):

# View logic goes here

return HttpResponse("Hello, World!")

Templates: acá básicamente pusimos todo lo que tiene que ver para poner los html los css o boostrap que esta en el html, ejemplo:

//myapp/templates/base.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>{% block title %}{% endblock %}</title>

{% load **static** %}

<link rel="stylesheet" type="text/css" href="<https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css>"/>

<link rel = "stylesheet" type = "text/css" href = "<https://cdn.datatables.net/1.12.0/css/jquery.dataTables.min.css>"/>

<meta charset="UTF-8" name="viewport" content="width=device-width, initial-scale=1"/>

</head>

<body>

<div **class**="container">

 <div **class**="row">

    <h4>Django Mysql CRUD (Create Read Update Delete) with Datatables</h4>

    {% block content %}{% endblock %}

 </div>

</div>

<script src="<https://cdn.jsdelivr.net/npm/jquery@3.6.0/dist/jquery.min.js>"></script>

<script src="<https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js>"></script>

<script src = "<https://cdn.datatables.net/1.12.0/js/jquery.dataTables.min.js>"></script>

<script>

$(document).ready(**function**() {

    $('#bootstrapdatatable').DataTable({

      "aLengthMenu": [[3, 5, 10, 25, -1], [3, 5, 10, 25, "All"]],

        "iDisplayLength": 3

       }

    );

} );

</script>

</body>

</html>

URL: Al configurar correctamente las URLs en tu proyecto de Django, puedes crear una aplicación web bien estructurada y fácilmente navegable. Los patrones de URL definen cómo los usuarios pueden acceder a varias(la parte más fundamental que no se puede olvidar) ejemplo.

**//**devproject**/**urls.py

**from** django.contrib **import** admin

**from** django.urls **import** path

**from** myapp **import** views

urlpatterns **=** [

    path('admin/', admin.site.urls),

    path('', views.index, name='index'),

    path('addnew',views.addnew, name**=**'addnew'),

    path('edit/<int:id>', views.edit, name**=**'edit'),

    path('update/<int:id>', views.update, name**=**'update'),

    path('delete/<int:id>', views.destroy, name**=**'destroy'),

]

Usamos Html y css para la página y utilizamos boostrap con los tipos de diseño que nos encanto y en eso usamos el Frontend

**METODOLOGIA DE SCRUM:**

**Master: Tomas molina y Mauro Olmos ambos pusimos en marcha la idea innovadora de la página de películas**

**Manager:Tomas molina vio que todo este en marcha**

**Los developers:Mauro olmos(front) y Tomas molina (backend)**

###### *El trabajo es para capacitación*