

**Ministerul Educației, Culturii și Cercetării al Republicii
Moldova**

Universitatea Tehnică a Moldovei

Facultatea Calculatoare Informatică și Microelectronică

Departamentul Inginerie Software și Automatică

Disciplina: Tehnologii Web

Lucrarea de Laborator nr.2

Tema: Crearea unei pagini Web in Visual Studio

A efectuat: Balaur Dorina gr. TI-184

A verificat: Cristian Rusu

Chișinău 2020

Scopul lucrării:

Adaugarea paginii facute in laboratorul 1 pe platforma ASP.Net in Visual Studio.

//Luand in considerare ca cunosc mai bine alta platforma si ne-ati permis sa o utilizam, am ales sa fac proiectul in Django PyCharm.

Django este un soft cadru pentru dezvoltarea aplicațiilor web gratuit și cu sursă deschisă, scris în Python, care urmează modelul arhitectural Model-View-Controller.

Scopul principal al acestui soft cadru pentru dezvoltarea aplicațiilor web este de a facilita crearea de website-uri complexe, fundate pe baze de date. Django pune accent pe reutilizarea codului, pe modularitate, dezvoltare rapidă a site-urilor web, ghidându-se după principiul "nu te repeta" (*Don't repeat yourself - DRY*). Django este codat de la un capăt la altul în Python, chiar și fișierele de configurare și modelele de date sunt implementate în acest limbaj de programare. Django oferă și un panou administrativ, care, deși vine preinstalat, este opțional, prin intermediul acestuia se pot crea, citi, actualiza și șterge cu ușurință informații din baza de date. Acest panou de administrare este generat dinamic prin introspecție (prin analizarea tabelor din baza de date) și poate fi ușor configurat prin modelele administrative de date.

Realizarea lucrării:

1. Crearea unui proiect nou in PyCharm

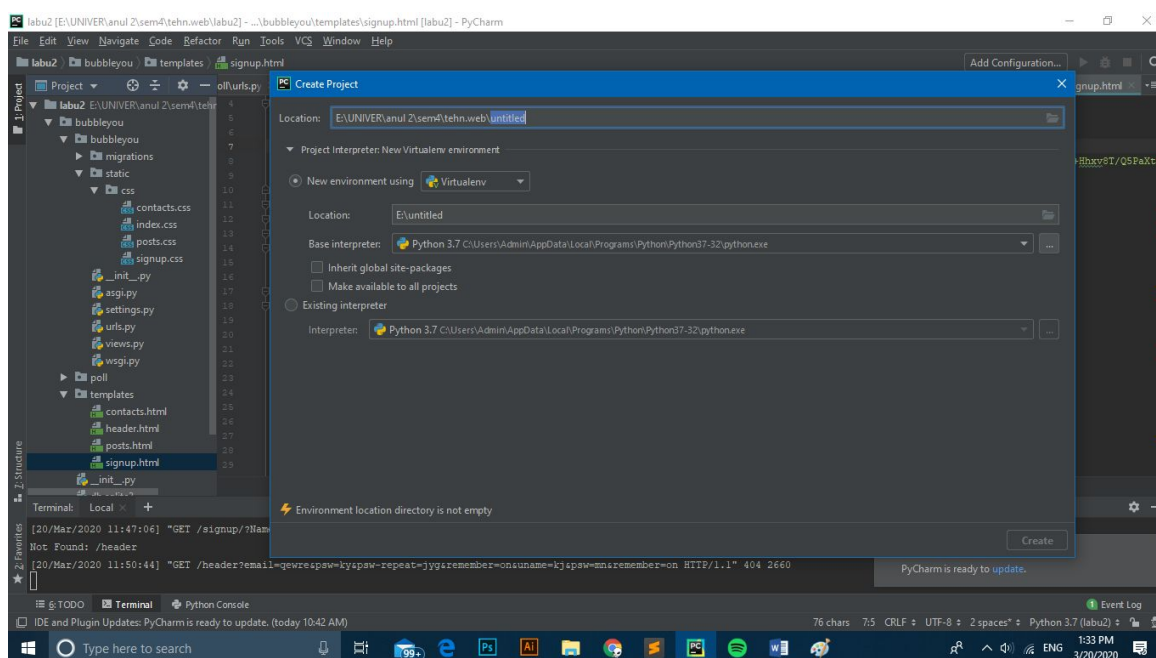
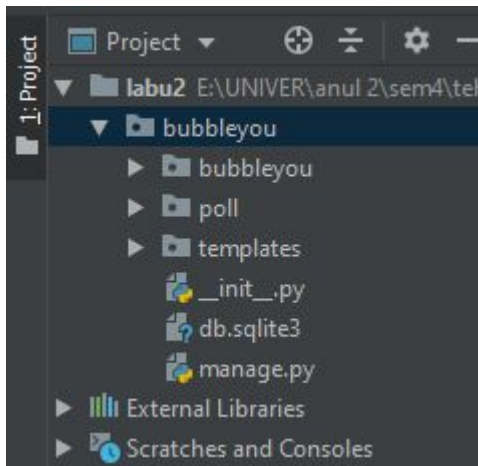


fig.1 Crearea unui proiect

2.Instalarea Django

3.Introducem codul de mai jos in terminal in PyCharm pentru a crea proiectul pe baza de Django.

```
$ django-admin startproject mysite
```



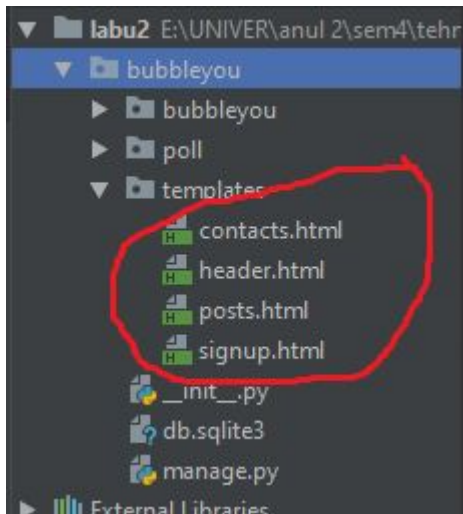
```
$ python manage.py runserver
```

4.Introducem in terminal codul pentru a porni serverul de dezvoltare a proiectului si vom primi ceva de genul in raspuns:

```
March 18, 2020 - 15:50:53  
Django version 3.0, using settings 'mysite.settings'  
Starting development server at http://127.0.0.1:8000/  
Quit the server with CONTROL-C.
```

Prin accesarea linkului ne va trimite pe site-ul creat.

5. In fisierul cu numele site-ului creat un New->Directory si il numim "Templates". Acolo deja cream file-uri .html pentru pagina noastra.



6. In acelasi mod ca la pasul precedent cream un nou Directory cu numele “static” si adaugam in acesta fisierele .css necesare.

7. Pentru a lega fisierele html cu cele css introducem codul urmatoare sub urmatoarea forma:

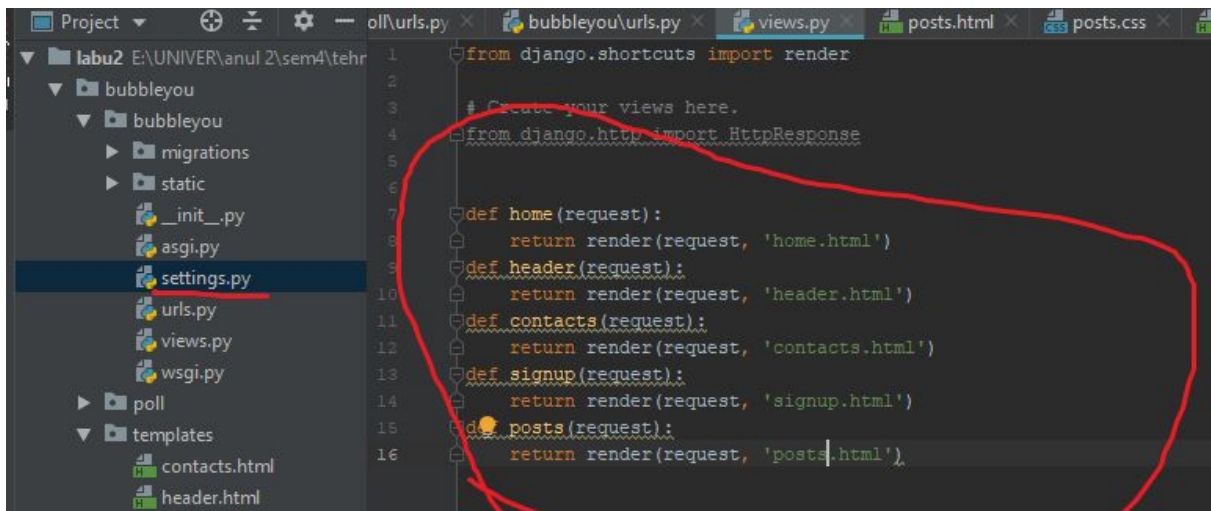
```
<!DOCTYPE html>
{% load static %}
<html>
<head>
  <title>signup</title>

  <link rel="stylesheet" type="text/css" href="{% static 'css/signup.css' %}">
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integr
```

8. in fisierul settings.py adaugam urmatorul cod:

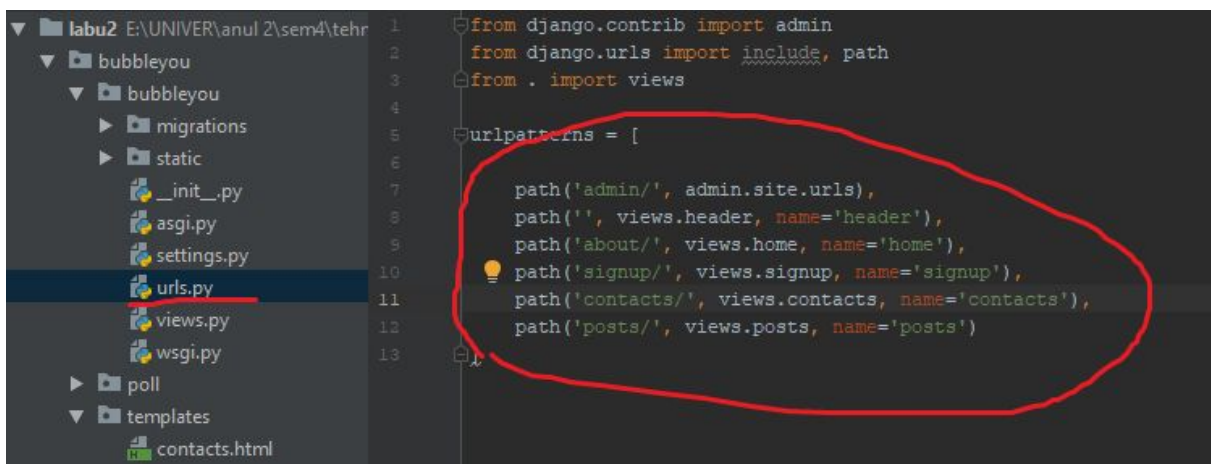
```
19 # Quick-start development settings - unsuitable for production
20 # See https://docs.djangoproject.com/en/3.0/howto/deployment/checklist/
21
22 # SECURITY WARNING: keep the secret key used in production secret!
23 SECRET_KEY = 'a@0dnhdgv7ly6wobx=ed_^hwdl-c^^say#lnlwt402%60e$*!r'
24
25 # SECURITY WARNING: don't run with debug turned on in production!
26 DEBUG = True
27
28 ALLOWED_HOSTS = []
29
30 SITE_ROOT = os.path.dirname(os.path.realpath(__file__))
31 STATICFILES_DIRS = (
32     os.path.join(SITE_ROOT, 'static/'),
33 )
34
35 # Application definition
```

9. In fisierul views adaugam urmatorul cod pentru a crea legaturile cu link-urile



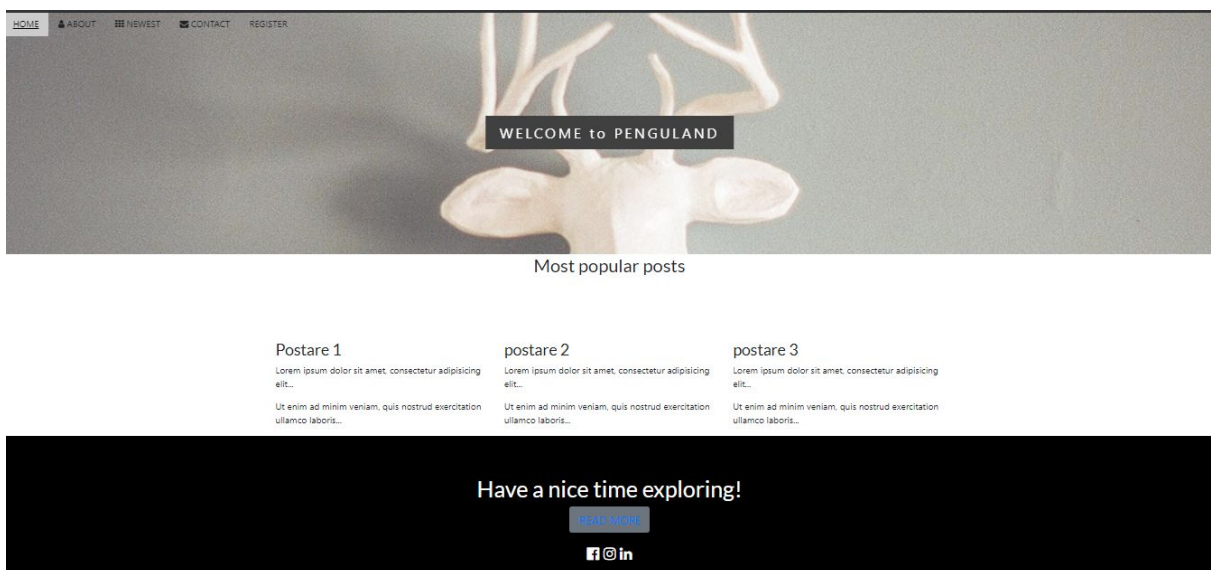
```
1 from django.shortcuts import render
2
3 # Create your views here.
4 from django.http import HttpResponse
5
6
7 def home(request):
8     return render(request, 'home.html')
9
10 def header(request):
11     return render(request, 'header.html')
12
13 def contacts(request):
14     return render(request, 'contacts.html')
15
16 def signup(request):
17     return render(request, 'signup.html')
18
19 def posts(request):
20     return render(request, 'posts.html')
```

10. In fisierul urls.py adaugam pentru a fi directionate linkurile



```
1 from django.contrib import admin
2 from django.urls import include, path
3 from . import views
4
5 urlpatterns = [
6
7     path('admin/', admin.site.urls),
8     path('', views.header, name='header'),
9     path('about/', views.home, name='home'),
10    path('signup/', views.signup, name='signup'),
11    path('contacts/', views.contacts, name='contacts'),
12    path('posts/', views.posts, name='posts')
13]
```

11. In final rulam pasul 4 pentru a vizualiza site-ul.



HOMEABOUTNEWESTCONTACTREGISTER

Contact Me

Chicago, US

Phone: +00 151515

Email: mail@mail.com

Lets get in touch. Send me a message

Name

Email

Subject

Message

SEND MESSAGE

Sign Up

Please fill in this form to create an account.

Email

Enter Email

Password

Enter Password

Repeat Password

Repeat Password

☒ Remember me

By creating an account you agree to our [Terms & Privacy](#).

Cancel

Sign Up

Log In

Username

Enter Username

Password

Enter Password

Login


☒ Remember me


Forgot [password?](#)

HOMEABOUTNEWESTCONTACTREGISTER

TITLE HEADING

Title description, May 2, 2019





pengu-sama

Just me, myself and I, exploring the universe of unknownment. I have a heart of love and a interest of lorem ipsum and mauris neque quam blog. I want to share my world with you.

Popular Posts

Concluzie:

In final am realizat mutarea unui site creat in Sublime pe o platforma Django cu ajutorul PyCharm. Am insusit crearea fisierelor in proiect si cum au loc legaturile pentru a putea accesa si directiona linkurile de la o pagina la alta. Deasemenea am facut cunostinta cum se combina limbajul Python cu HTML si CSS si diferentele care trebuiesc adaugate pentru ca site-ul sa ruleze.