**Online Job Application Protocol**

END-TERM REPORT

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

By:

|  |  |  |  |
| --- | --- | --- | --- |
| *S.no.* | *Name* | *Roll No.* | *Registration no.* |
| *1.* | *Tamal Saha* | *A140* | *11617846* |
| *2.* | *Royce Elijha* | *02* | *11912988* |
|  |  |  |  |
|  |  |  |  |

**Courses Code: INT213**

**School of Computer Science and Engineering**

Lovely Professional University

Phagwara, Punjab (India)

**Objective**

The objective of this python project “Online job application protocol” is to develop an online portal where recruiters can post job requirements, they can search for candidates. Candidates can search for job openings and apply.

**Users of this protocol :**

1. HR (employer)
2. Job seeker (Student or employee)
3. Admin

**Introduction**

1. Login and Sign up for HR, not for Job seekers because there are a lot of websites where they would be applying, so logging in and remembering all usernames and passwords would be really hectic and why would we want that? So it is better to ask about their information when they actually apply for some company. But HR would have access to details of the job seekers, for which authorization is required.
2. Listing of available jobs.
3. Resume upload and apply to an organization.
4. List of all candidates to the HR.

**Data to be stored**

**For HR:**

1. Name of the organization.
2. Position
3. Job description
4. Salary
5. Location
6. Experience

**For Job seekers:**

1. Name
2. DOB
3. Gender
4. Mobile
5. Email
6. Resume
7. Organization name (the one on which he is applying)

**Project Prerequisites**

We will use the following technologies:

* **Python** (Python is an interpreted, high-level and general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace.)
* **HTML** (Hypertext Markup Language is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets and scripting languages such as JavaScript. )
* **CSS** (Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.)
* **Bootstrap** (Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.)
* **Django framework** (Django is a Python-based free and open-source web framework that follows the model-template-views architectural pattern.)

Now Here I am shearing some codes and Screen shoot of the project :

To install the Django framework we have to type the following command on terminal.

**pip install Django**

For storing the data we have to connect the server of a database, which name tamalmodels.py will handle.

. **tamalmodels.py :**

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

user=models.OneToOneField(User,**null**=**True**,on\_delete=models.CASCADE)

name=models.CharField(max\_length=200,**null**=**True**)

position=models.CharField(max\_length=200,**null**=**True**)

description=models.CharField(max\_length=2000,**null**=**True**)

salary=models.IntegerField(**null**=**True**)

experience=models.IntegerField(**null**=**True**)

Location=models.CharField(max\_length=2000,**null**=**True**)

**def** \_\_str\_\_(self):

**return** self.name

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

category=(

('Male','male'),

('Female','female'),

('Other','other'),

)

name=models.CharField(max\_length=200,**null**=**True**)

dob=models.DateField(**null**=**True**)

gender= models.CharField(max\_length=200,**null**=**True**,choices=category)

mobile= models.CharField(max\_length=200,**null**=**True**)

email= models.CharField(max\_length=200,**null**=**True**)

resume=models.FileField(**null**=**True**)

company=models.ManyToManyField(Company,blank=**True**)

**def** \_\_str\_\_(self):

**return** self.name

### tamalforms.py :

from django.forms import ModelForm

from .models import \*

class ApplyForm(ModelForm):

class Meta:

model=Candidates

fields="\_\_all\_\_"

### tamaladmin.py

from django.contrib import admin

from .models import \*

# Register your models here.

admin.site.register(Company)

admin.site.register(Candidates)

### tamalsettings.py

STATIC\_URL = '/static/'

MEDIA\_URL='/media/'

MEDIA\_ROOT=os.path.join(BASE\_DIR,'media/')

### tamalurls.py

from django.contrib import admin

from django.urls import path,include

from django.conf import settings

from django.conf.urls.static import static

urlpatterns = [

path('admin/', admin.site.urls),

path('',include('JobPortal.urls'))

] + static(settings.MEDIA\_URL,document\_root=settings.MEDIA\_ROOT)

**JobPortal/urls.py:**

from django.urls import path

from .views import \*

urlpatterns = [

path('',home,name='home'),

path('login/',loginUser,name='login'),

path('logout/',logoutUser,name='logout'),

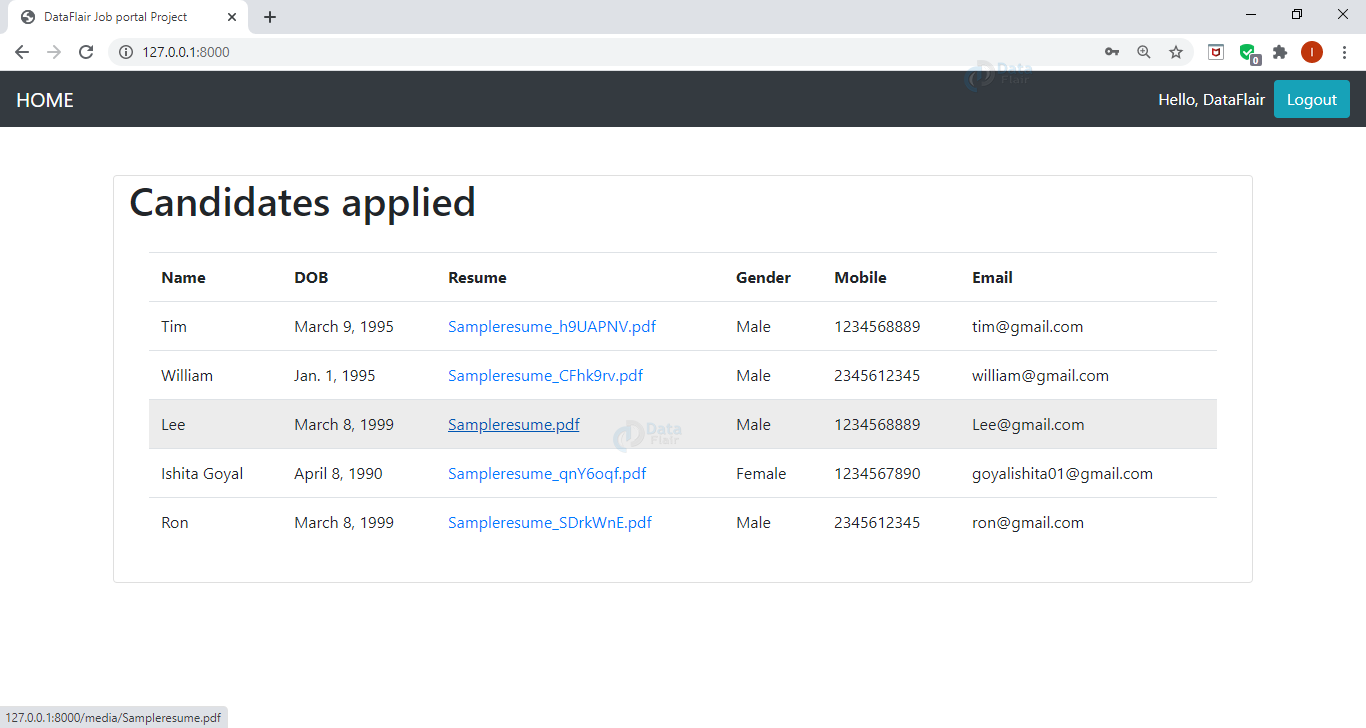
path('register/',registerUser,name='register'),

path('apply/',applyPage,name='apply'),

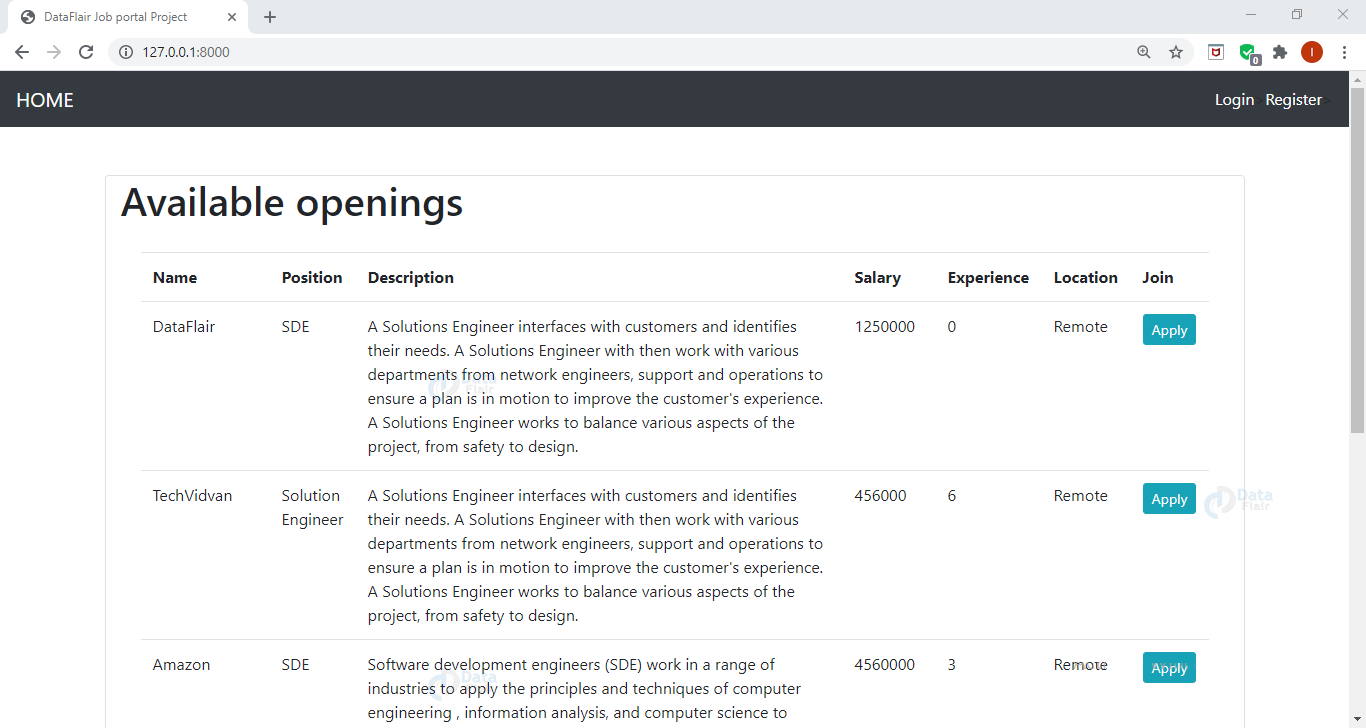
]

**Screenshots:**

**Home Page :**

****

**Jobseeker Home Page :**

****

**References**

* [**https://www.w3schools.com/css/**](https://www.w3schools.com/css/)
* [**https://www.w3schools.com/html/default.asp**](https://www.w3schools.com/html/default.asp)
* [**https://www.w3schools.com/python/default.asp**](https://www.w3schools.com/python/default.asp)
* [**https://www.w3schools.com/bootstrap/bootstrap\_ver.asp**](https://www.w3schools.com/bootstrap/bootstrap_ver.asp)
* [**https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction**](https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction)