1.INTRODUCTION

A million of engineering students are struggling to get placed is an extremely challenging in market

Somewhere between a fifth to a third of the million students graduating out of engineering colleges run the risk of being unemployed.

B.Tech. (Bachelor of Technology) is one of the most preferred undergraduate degree courses among students from the Science stream post-school. The 4-year engineering course is offered by many colleges across India. Once you have completed engineering, what next? What to do after B.Tech.? What are the available jobs after B.Tech.? These questions are faced by many engineering students in the fourth year of B.Tech.

There are many routes you can take after completing your B.Tech. However, with the huge number of career options available, it is natural that you may feel confused about which one to select. Frustrated engineers are taking jobs for which they are ovequalified and, therefore, underpaid. There are so many new jobs being created every year. The number of new engineers is far higher. Most Indian engineering colleges provide no or very bad internship opportunities. Most engineers have no clue what the industry actually is like..

The student fail to understand the importance of trustworthy online websites which are providing jobs and relying on platforms like facebook and instaram for job search which is a major mistake.

To help you decide where to go for the jobs according to your interest, we have prepared a website which will increase your limited options for jobs.

Our project is all about providing the best website for engineering students to search jobs and Internships or to know about the current government vacancies available. People don’t have to waste their time by going through all the jobs for each branch.
We have divide our website according to the interest of the students and according to their branches.

1.1 PROJECT DESCRIPTION

The 'JOBS4U' website is built to overcome the problem of the tiring and time consuming process of job search.

- The job searchers will be able to find employment in their fields.
- Our Website is an easy-to-use tool that shows information without being overly complicated.
- The website will make finding a job quite effective.
- Visitors to our websites will receive an updated jobs list.
- Users will receive assistance in selecting a specific career path.

1.2 METHODOLOGY

This project has been created using web development technology, which includes basic web page design with the working of web scraping. Our main aim is to create an online job portal 'JOB4U' which provides you the latest government and private job opportunities for your field of interest and education, with internships too for the freshers looking for a head start in their career.

For the construction of the frontend, the technologies used are HTML and CSS:

- The framework of our web page is built here using HTML. For creating headings, texts, tables, icons, and images, among other things.
- The web page's appearance, including its sizes, layouts, colors , and fonts, is created using CSS in this instance.

Further the project is continued by building a web scraping tool with Python. By incorporating it into a Django web app, we'll be extending the functionality of our scheduled web scraper.

We have used web scraping for:

- Job listings: Details regarding job openings are collected from different websites and then listed in one place so that it is easily accessible to the user.

We have used the following modules and libraries:

- Python 3.7+
- Requests — For web requests
- Beautiful Soup 4 — HTML parsing tool
- A text editor (we have used Visual Studio Code)

When we run the code for web scraping, a request is sent to the URL that we have mentioned. As a response to the request, the server sends the data and allows to read the HTML or XML page. The code then, parses the HTML or XML page, finds the data and extracts it.

The libraries used in python for scraping the data are:

- Bs4 - It provides simple methods to navigate, search, and modify parse trees.
- Render - It serves as a central place to store the metadata data about image tiles and how they should be transformed.
- Beautiful-Soup - A python library that is used for pulling data out of HTML and XML files. It provides simple methods for navigating, searching and modifying a parse tree in HTML,XML files
- Request - The requests module allows you to send HTTP requests using Python. The HTTP request returns a Response Object with all the response data (content, encoding, status, etc).

And then web scraping is connected to frontend using the Django framework which includes:

- ASGI - The Asynchronous Server Gateway Interface is a calling convention for web servers to forward requests to asynchronous-capable Python programming language frameworks, and applications. It describes a common interface between a Python web application and the web server.
- An HTTP Response - instance wraps the HTTP response from the server. It provides access to the request headers and the entity body.
- Django shortcuts module is a collection of helper functions that are generally used in view function/classes.
- WSGI – The Web Server Gateway interface is a simple calling convention for web servers to forward requests to web applications or frameworks written in the python programming language.

2. SOFTWARE REQUIREMENTS

2.1.HTML

- To understand "HTML" from front to back, let's look at each word that makes up the abbreviation:
- **Hypertext**: text (often with embeds such as images, too) that is organized in order to connect related items.
- **Markup**: a style guide for typesetting anything to be printed in hardcopy or soft copy format.
- **Language**: a language that a computer system understands and uses to interpret commands.
- HTML determines the structure of web pages. This structure alone is not enough to make a web page look good and interactive. So you'll use assisted technologies such as CSS and JavaScript to make your HTML beautiful and add interactivity, respectively.

2.2.CSS

- CSS stands for Cascading Style Sheets.
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media.
- CSS saves a lot of work. It can control the layout of multiple web pages all at once External stylesheets are stored in CSS files.

2.3. WEB SCRAPING

- Web scraping refers to the extraction of data from a website. This information is collected and then exported into a format that is more useful for the user.

2.4. PYTHON 3

- Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable.
- It supports functional and structured programming methods as well as OOP. It can be used as a scripting language or can be compiled to byte-code for building large applications.

2.5. DJANGO FRAMEWORK

- Django is a free, python-based, open-source web development framework that facilitates clean and rational designing of websites driven by databases.
- It reduces a lot of hassles that a developer has to face during the development of a website. It is used to develop minimal yet smart web applications.

3.IMPLEMENTATION

3.1.FRONTEND

3.1.1 The icon Bar: Icon is a small graphical representation of a program or file. When we click an icon, the associated file or program will be opened. For example, if we were to click on the Facebook icon, it would open Facebook.

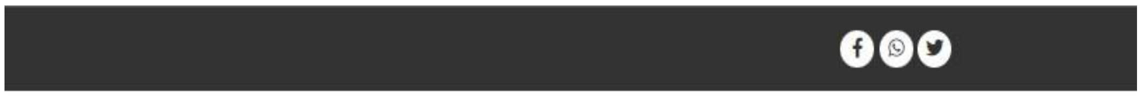


Fig.3.1.The icon Bar

3.1.2 The Logo Bar: A logo is a symbol made up of text and images that identifies a business.

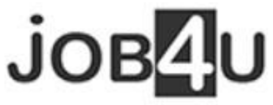


Fig.3.2.The logo Bar

3.1.3 The Poster: Typically, Posters include both textual and graphic elements,although a poster may be either wholly graphical or wholly text. Posters are designed to be catching and informative.

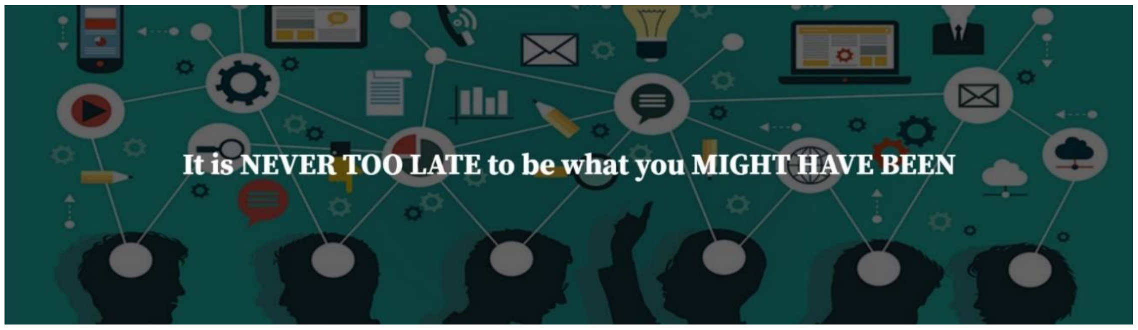


Fig.3.3.The Poster

3.1.4 Jobs Based on Branches: Here you will get jobs according to your branches.

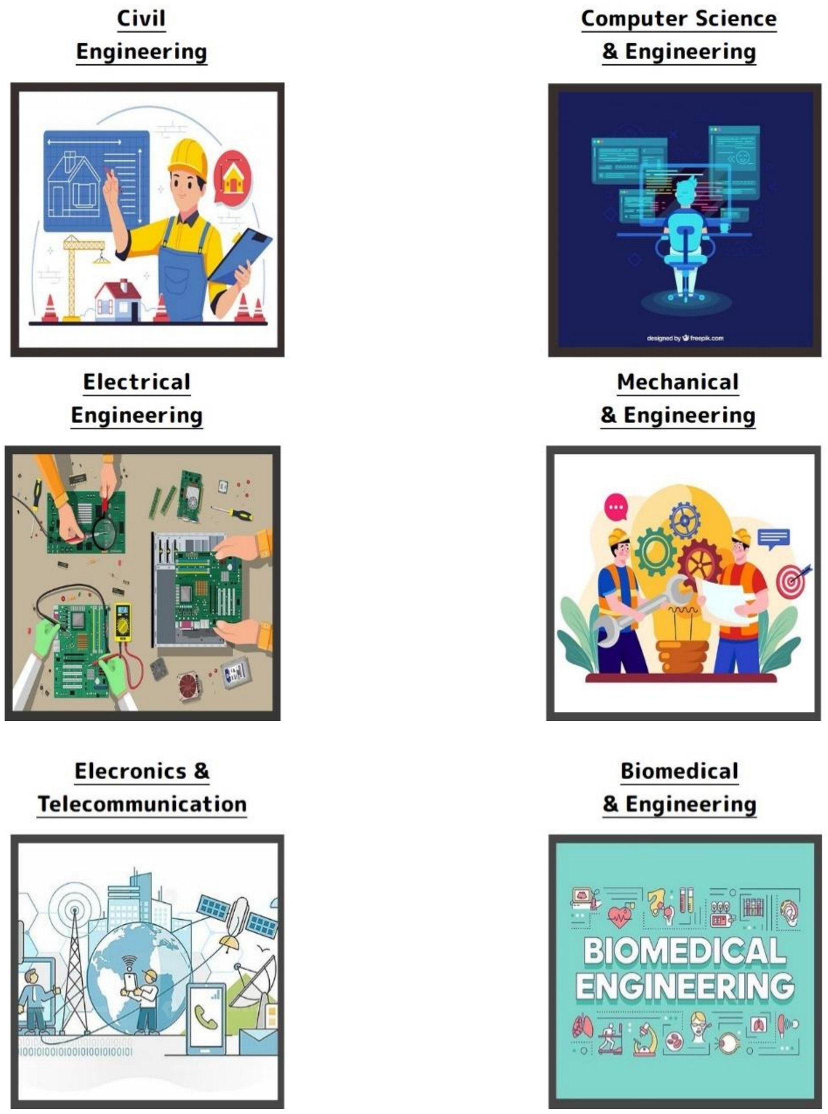
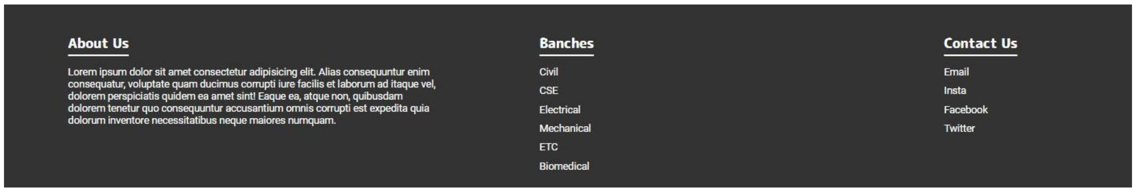


Fig.3.4.Jobs Based on Branches

3.1.5 The Footer: A Website's footer is an area located at the bottom of every page on a website, below the main body content. The term "footer" comes from the print world, in which the "footer" is a consistent design element that is seen across all pages of a document.



About Us

Lorem ipsum dolor sit amet consectetur adipisicing elit. Alias consequuntur enim consequatur, voluptate quam ducimus corrupti iure facilis et laborum ad itaque vel, dolorem perspiciatis quidem ea amet sint! Eaque ea, atque non, quibusdam dolorem tenetur quo consequuntur accusantium omnis corrupti est expedita quia dolorum inventore necessitatibus neque maiores numquam.

Banches

- Civil
- CSE
- Electrical
- Mechanical
- ETC
- Biomedical

Contact Us

- Email
- Insta
- Facebook
- Twitter

Fig.3.5.The Footer

3.1.6 The Copyright: Copyright refers to the legal right of the owner pf intellectual property. In simpler terms, copyright is right to copy . This means that the originl creators of products and anyone they give authorization to are the only ones with the exclusive right to reproduce the work.

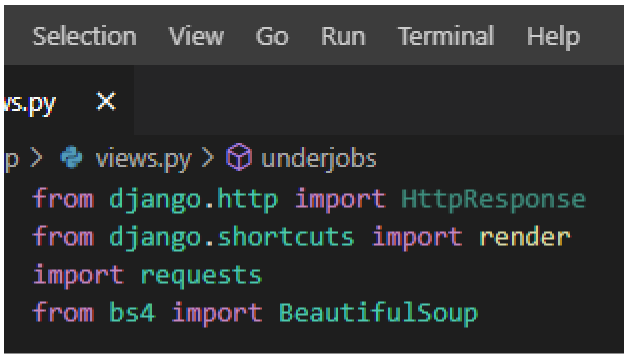
Copyright© job4u

Fig.3.6.The Copyright

3.2 BACKEND

The following fundamental procedures must be followed in order to extract data with Python web scraping:

3.2.1 Import the Libraries required for Web Scraping:



```
from django.http import HttpResponse
from django.shortcuts import render
import requests
from bs4 import BeautifulSoup
```

Fig 3.7.Importing the Libraries for Web Scraping

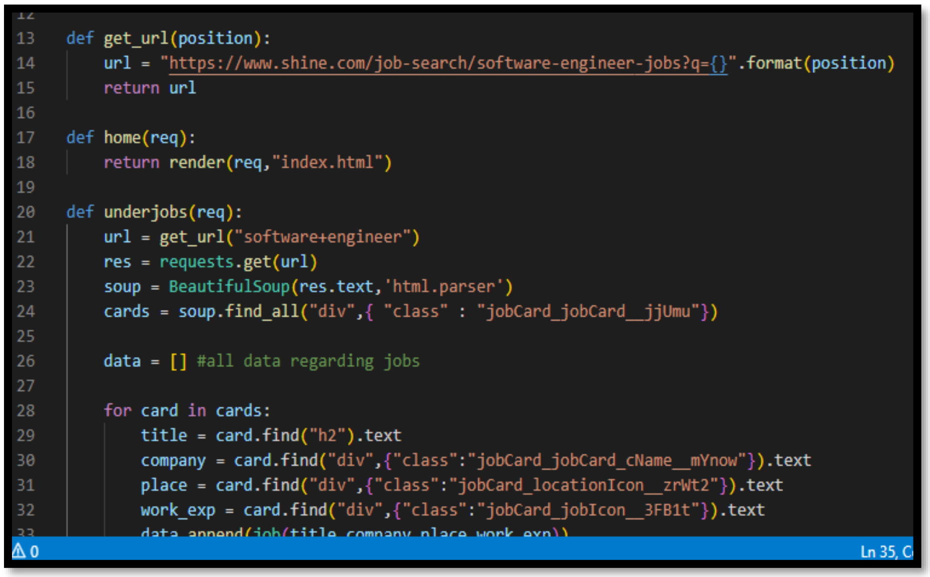
3.2.2 Find the URL that you want to scrape:

-Here we are scraping the jobs from shine.com

-The Required URL is: https://www.shine.com/job-search/software-engineer-jobs?q={}

-Moving to more depth we are scraping all the private jobs of software engineers.

-Thus we will be getting all the jobs listed in the software engineering from: url = get_url("software+engineer")

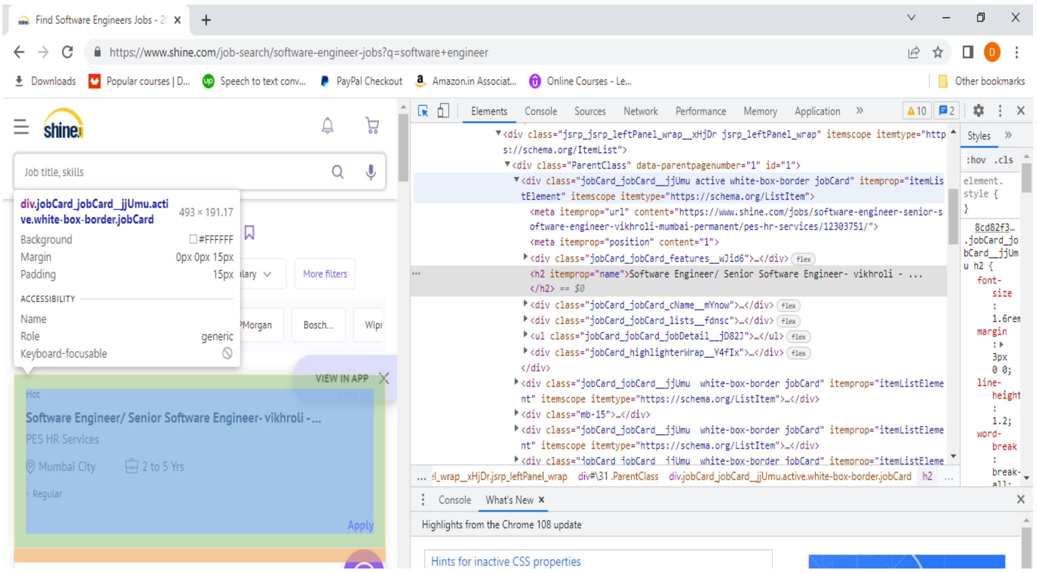


```
def get_url(position):
    url = "https://www.shine.com/job-search/software-engineer-jobs?q={}".format(position)
    return url

def home(req):
    return render(req,"index.html")

def underjobs(req):
    url = get_url("software+engineer")
    res = requests.get(url)
    soup = BeautifulSoup(res.text,'html.parser')
    cards = soup.find_all("div",{ "class" : "jobCard_jobCard__jjUmu"})

    data = [] #all data regarding jobs

    for card in cards:
        title = card.find("h2").text
        company = card.find("div",{"class":"jobCard_jobCard_cName__mYnow"}).text
        place = card.find("div",{"class":"jobCard_locationIcon__zrWt2"}).text
        work_exp = card.find("div",{"class":"jobCard_jobIcon__3FB1t"}).text
        data.append(job(title,company,place,work_exp))
```

Fig 3.8.Getting the URL

3.2.3 Inspecting the Page:

-Now by selecting any one of the jobs inside software engineer, click inspect.



```
▼<div class="jsrp_jsrp_leftPanel_wrap__xHjDr jsrp_leftPanel_wrap" itemscope itemtype="https://schema.org/ItemList">
  ▼<div class="ParentClass" data-parentpagenumber="1" id="1">
    ▼<div class="jobCard_jobCard__jjUmu active white-box-border jobCard" itemprop="itemListElement" itemscope itemtype="https://schema.org/ListItem">
       <meta itemprop="url" content="https://www.shine.com/jobs/software-engineer-senior-software-engineer-vikhroli-mumbai-permanent/pes-hr-services/12303751/">
       <meta itemprop="position" content="1">
      ▶<div class="jobCard_jobCard_features__wJid6">…</div> flex
       <h2 itemprop="name">Software Engineer/ Senior Software Engineer- vikhroli - ...
       </h2> == $0
      ▶<div class="jobCard_jobCard_cName__mYnow">…</div> flex
      ▶<div class="jobCard_jobCard_lists__fdnsc">…</div> flex
      ▶<ul class="jobCard_jobCard_jobDetail__jD82J">…</ul> flex
      ▶<div class="jobCard_highlighterWrap__Y4fIx">…</div> flex
     </div>
    ▶<div class="jobCard_jobCard__jjUmu  white-box-border jobCard" itemprop="itemListElement" itemscope itemtype="https://schema.org/ListItem">…</div>
    ▶<div class="mb-15">…</div>
    ▶<div class="jobCard_jobCard__jjUmu  white-box-border jobCard" itemprop="itemListElement" itemscope itemtype="https://schema.org/ListItem">…</div>
    ▶<div class="jobCard_jobCard__jjUmu  white-box-border jobCard" itemprop="itemListEleme
```

div.jobCard_jobCard__jjUmu.active.white-box-border.jobCard 493 × 191.17

Background	#FFFFFF
Margin	0px 0px 15px
Padding	15px
ACCESSIBILITY	
Name	
Role	generic
Keyboard-focusable	

Fig 3.9.Inserting Page

3.2.4 Find the data you want to extract:

- Here we have extracted:
  - Title
  - Company
  - Work Experience
  - Place
  - Salary

3.2.5 Create an empty list to collect and store all the data above:

- Here we have created: - data = [] #to store all the jobs.

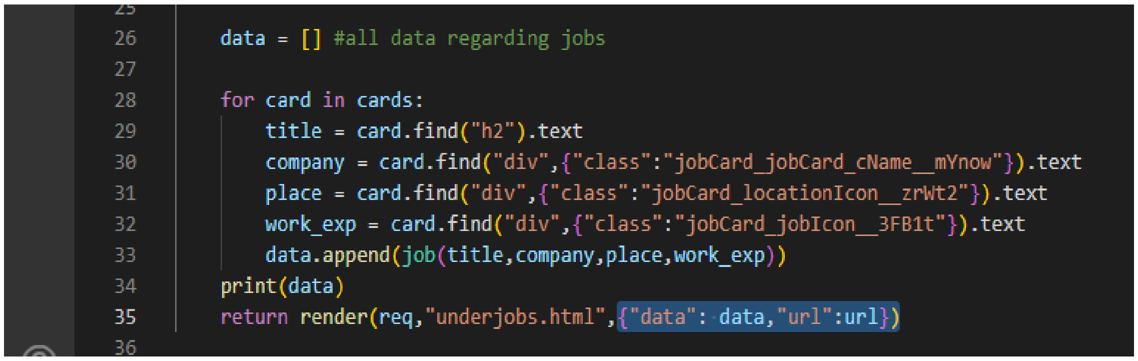
3.2.6 Start a for loop to iterate the jobs:

-To get all the jobs from the website of the desired page start a for loop.

-Run the code and start extracting the data by calling their class as it will be same for all the jobs.

3.2.7 Store the data in the required format:

- {“data”: data, “url”: url})



```
    data = [] #all data regarding jobs

    for card in cards:
        title = card.find("h2").text
        company = card.find("div",{"class":"jobCard_jobCard_cName__mYnow"}).text
        place = card.find("div",{"class":"jobCard_locationIcon__zrWt2"}).text
        work_exp = card.find("div",{"class":"jobCard_jobIcon__3FB1t"}).text
        data.append(job(title,company,place,work_exp))
    print(data)
    return render(req,"underjobs.html",{"data": data,"url":url})
```

Fig3.10.Extracting Data

3.3 CONNECTING FRONTEND TO BACKEND

The following fundamental procedures must be followed in order to display the extracted data with Python in the frontend:

3.3.1 Installing Django

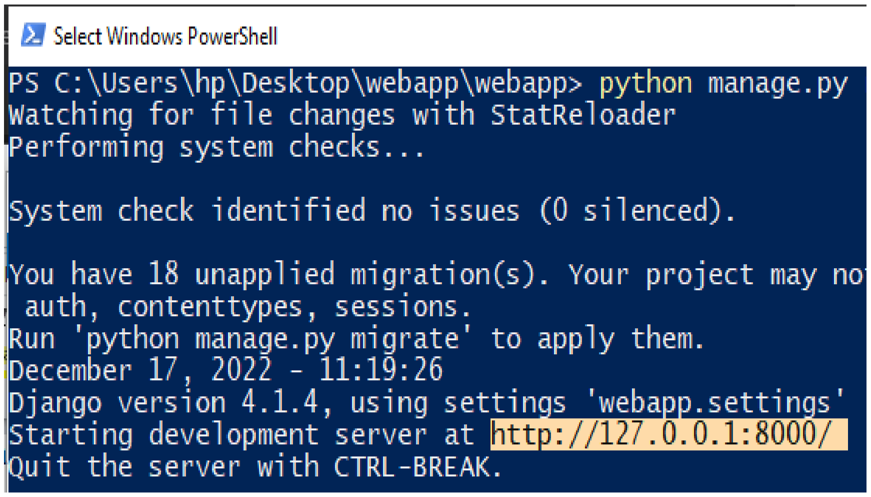
-To install django we use: pip install django.

- It automatically creates a very important file named manage.py

3.3.2 Running the Django:

-Open the Command Prompt or the power-shell and write: Python manage.py runserver

-We will get the address of the server in which our project will run.

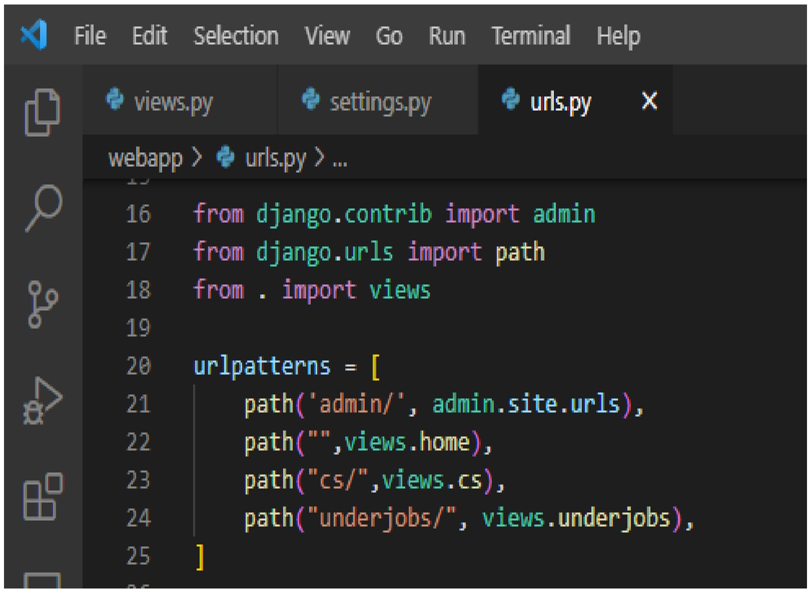


```
PS C:\Users\hp\Desktop\webapp\webapp> python manage.py
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may no
 auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
December 17, 2022 - 11:19:26
Django version 4.1.4, using settings 'webapp.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Fig 3.11.Running the Django

3.3.3 Create a new folder

- For connecting backend to frontend create a new folder which will be declaring the path.
- Here we have created urls.py for declaring the path.

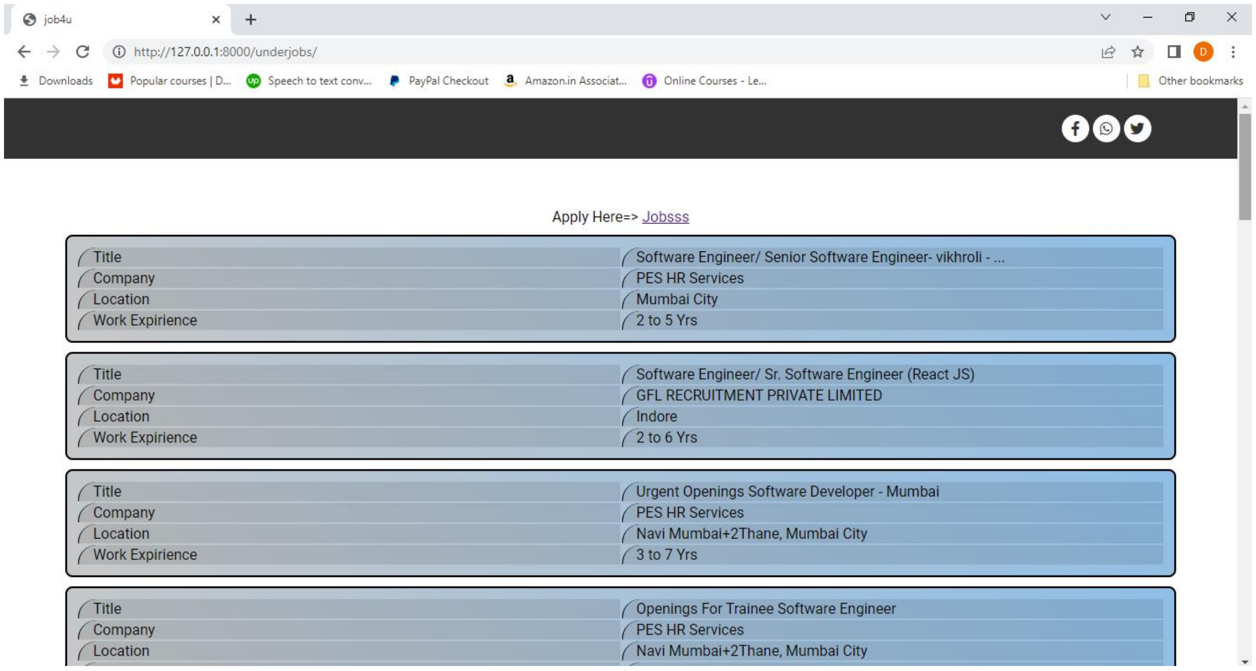
-


```python
from django.contrib import admin
from django.urls import path
from . import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path("",views.home),
    path("cs/",views.cs),
    path("underjobs/", views.underjobs),
]
```

Fig3.12.URLs.py

4. RESULT

Since the path was declared in the Fig 3.12. we have our fully completed and connected site, where we can see the listed jobs and can apply one of them according to our intrest.



Apply Here=> Jobsss

Title	Software Engineer/ Senior Software Engineer- vikhroli - ...
Company	PES HR Services
Location	Mumbai City
Work Expirience	2 to 5 Yrs

Title	Software Engineer/ Sr. Software Engineer (React JS)
Company	GFL RECRUITMENT PRIVATE LIMITED
Location	Indore
Work Expirience	2 to 6 Yrs

Title	Urgent Openings Software Developer - Mumbai
Company	PES HR Services
Location	Navi Mumbai+2Thane, Mumbai City
Work Expirience	3 to 7 Yrs

Title	Openings For Trainee Software Engineer
Company	PES HR Services
Location	Navi Mumbai+2Thane, Mumbai City

Fig 3.13.List of Jobs

5. CONCLUSION

Working on this fascinating and difficult project has been a real pleasure for us. An effective way for job seekers to look online for information on open positions is through a job portal. This portal's primary objective is to attempt to develop graduates who are appropriate for the market, and to provide services to those who deserve the job with their skills.

The greater scope of job searching makes it quicker and simpler to discover prospects. Today's changing business environment and expanding employment chances have prompted more people to look for better careers and employers to look for better potentials.

6. FUTURE SCOPE

Additionally, we may add extra features to improve its effectiveness or efficiency and make it more APPEALING

6.1 SCOPE

- Data or File uploader
- Creating a resume
- Individual or Personal Profile

6.2 FUTURE APPLICATION AND IMPLEMENTATION

- In situations where clients already have a completed resume, the ability to upload a resume file is helpful. In that situation, customers don't need to spend more time filling out the questionnaire to create their resume; they can just upload the document.
- The users are required to complete a questionnaire about their job, their education, their talents, their prior experience, their desired income, their location, and their certification.
- All the resources and data a person could require to begin a job search are available in the personal profiles on a job search website.
  Such websites or customer profiles feature job filters, suggestions, and opportunities to:
  - research salaries
  - follow businesses
  - viewing featured content
  - gain additional useful information

7.BIBLIOGRAPHY

[1]"Website used for scraping private software engineer jobs" Shine.com
https://www.shine.com/job-search/software-engineer-jobs?q=software+engineer

[2] "Color Palettes for CSS: Cascading style Sheet" developer.chrome.com
https:// developer.chrome.com /flatuicolors.com/

[3] "Fonts of Text" developer.chrome.com
https:// developer.chrome.com /fonts.google.com/

[4] "PSD to HTML" developer.chrome.com
https://www.youtube.com/watch?v=WHgRcRJJOSI&list=PLU8kFFDEjfkrs7Tdg-sKNzU06bCJkrgj7&index=7

[5] "Social Icons"developer.chrome.com
https://developer.chrome.com/stock.adobe.com/search?k=social+icons&search_type=usertyped

[6] "WebScraping with Python Libraries" freecodecamp.org
https://www.freecodecamp.org/news/web-scraping-python-tutorial-how-to-scrape-data-from-a-website/

[7] "Django Models" geeksforgeeks.org
https://www.geeksforgeeks.org/django-models/

[8] "How to scrape JOB posts from INDEED with PYTHON" youtube.com
https://www.youtube.com/watch?v=eN_3d4JrL_w&feature=youtu.be