```
import requests
from bs4 import BeautifulSoup


scrapejobs = requests.get("https://www.timesjobs.com/candidate/job-search.html?searchType=personalizedSearch&from=submit&txtKeywords=Powe
soup = BeautifulSoup(scrapejobs.content,"html.parser")
Jobs = soup.find_all("li",class_="clearfix job-bx wht-shd-bx")
for Job in Jobs:
    JobTitle = Job.find("h2").a.text
    CompanyName = Job.find("h3",class_ = "joblist-comp-name").text.strip()
    KeySkills = Job.find("span", class_ = "srp-skills").text.replace(" ","")
    print(f"Job Title:{JobTitle}")
    print(f"Company Name :{CompanyName}")
    print(f"Key Skills:{KeySkills}")
    print("***********")


# Fetching the Recent jobs
scrapejobs = requests.get("https://www.timesjobs.com/candidate/job-search.html?searchType=personalizedSearch&from=submit&txtKeywords=Powe
soup = BeautifulSoup(scrapejobs.content,"html.parser")
Jobs = soup.find_all("li",class_="clearfix job-bx wht-shd-bx")
for Job in Jobs:
    Time = Job.find("span",class_= "sim-posted").span.text
    if "few" in Time:
        JobTitle = Job.find("h2").a.text
        CompanyName = Job.find("h3",class_ = "joblist-comp-name").text.strip()
        KeySkills = Job.find("span", class_ = "srp-skills").text.replace(" ","")
        print(f"Job Title:{JobTitle}")
        print(f"Company Name :{CompanyName}")
        print(f"Key Skills:{KeySkills}")
        print("***********")


# Adding Fucntionality Search Functionality
scrapejobs = requests.get("https://www.timesjobs.com/candidate/job-search.html?searchType=personalizedSearch&from=submit&txtKeywords=Powe
soup = BeautifulSoup(scrapejobs.content,"html.parser")
print("Input the skills that you are unfamiliar with")
unfamiliar_skills = input("Enter the Skill : ")
print("Loding Jobs according to your Skills")
print("********************")
Jobs = soup.find_all("li",class_="clearfix job-bx wht-shd-bx")
for Job in Jobs:
    Time = Job.find("span",class_= "sim-posted").span.text
    if "few" in Time:
        JobTitle = Job.find("h2").a.text
        CompanyName = Job.find("h3",class_ = "joblist-comp-name").text.strip()
        KeySkills = Job.find("span", class_ = "srp-skills").text.replace(" ","")
        if unfamiliar_skills not in KeySkills:
            print(f"Job Title:{JobTitle}")
            print(f"Company Name :{CompanyName}")
            print(f"Key Skills:{KeySkills}")
            print("***********")


import time
def ScrapeJobs():
    scrapejobs = requests.get("https://www.timesjobs.com/candidate/job-search.html?searchType=personalizedSearch&from=submit&txtKeywords=
    soup = BeautifulSoup(scrapejobs.content,"html.parser")
    print("Input the skills that you are unfamiliar with")
    unfamiliar_skills = input("Enter the Skill : ")
    print("Loding Jobs according to your Skills")
    print("********************")
    Jobs = soup.find_all("li",class_="clearfix job-bx wht-shd-bx")
    for Job in Jobs:
        Time = Job.find("span",class_= "sim-posted").span.text
        if "few" in Time:
            JobTitle = Job.find("h2").a.text
            CompanyName = Job.find("h3",class_ = "joblist-comp-name").text.strip()
            KeySkills = Job.find("span", class_ = "srp-skills").text.replace(" ","")
            if unfamiliar_skills not in KeySkills:
                print(f"Job Title:{JobTitle}")
                print(f"Company Name :{CompanyName}")
                print(f"Key Skills:{KeySkills}")
                print("***********")
if __name__ == "__main__":
    while True:
        ScrapeJobs()
        Waiting_Time = 1
        print(f"Waiting For 6 seconds")
        time.sleep(6*Waiting_Time)
```

```
# Adding Jobs Folder By Creating Path using file system in Python using Path library
import time
from pathlib import Path
def ScrapeJobs():
    scrapejobs = requests.get("https://www.timesjobs.com/candidate/job-search.html?searchType=personalizedSearch&from=submit&txtKeywords=
    soup = BeautifulSoup(scrapejobs.content,"html.parser")
    print("Input the skills that you are unfamiliar with")
    unfamiliar_skills = input("Enter the Skill : ")
    print("Loding Jobs according to your Skills")
    print("********************")
    Jobs = soup.find_all("li",class_="clearfix job-bx wht-shd-bx")
    posts_directory = Path("posts")
    posts_directory.mkdir(parents=True, exist_ok=True)
    for index,Job in enumerate(Jobs):
        Time = Job.find("span",class_= "sim-posted").span.text
        if "few" in Time:
            JobTitle = Job.find("h2").a.text
            CompanyName = Job.find("h3",class_ = "joblist-comp-name").text.strip()
            KeySkills = Job.find("span", class_ = "srp-skills").text.replace(" ","")
            if unfamiliar_skills not in KeySkills:
                with open(posts_directory / f"job{index + 1}.txt", "w") as f:
                    f.write(f"Job Title:{JobTitle}")
                    f.write(f"Company Name :{CompanyName}")
                    f.write(f"Key Skills:{KeySkills}")
                    f.write("***********")
if __name__ == "__main__":
    while True:
        ScrapeJobs()
        Waiting_Time = 1
        print(f"Waiting For 6 seconds")
        time.sleep(6*Waiting_Time)



# adding a functionaity so that the user can put more then one unfamilair skills
import time
from pathlib import Path
def ScrapeJobs():
    scrapejobs = requests.get("https://www.timesjobs.com/candidate/job-search.html?searchType=personalizedSearch&from=submit&txtKeywords=
    soup = BeautifulSoup(scrapejobs.content,"html.parser")
    print("Input the skills that you are unfamiliar with")
    Input_unfamiliar_skills = input("Enter the skills")
    unfamiliar_skills = []
    skills_list = Input_unfamiliar_skills.strip(",")
    for skills in skills_list:
        stripped_skills =skills.strip()
        unfamiliar_skills.append(stripped_skills)
    print("Loding Jobs according to your Skills")
    print("********************")
    Jobs = soup.find_all("li",class_="clearfix job-bx wht-shd-bx")
    posts_directory = Path("posts")
    posts_directory.mkdir(parents=True, exist_ok=True)
    for index,Job in enumerate(Jobs):
        Time = Job.find("span",class_= "sim-posted").span.text
        if "few" in Time:
            JobTitle = Job.find("h2").a.text
            CompanyName = Job.find("h3",class_ = "joblist-comp-name").text.strip()
            KeySkills = Job.find("span", class_ = "srp-skills").text.replace(" ","")
            if unfamiliar_skills not in KeySkills:
                with open(posts_directory / f"job{index + 1}.txt", "w") as f:
                    f.write(f"Job Title:{JobTitle}")
                    f.write(f"Company Name :{CompanyName}")
                    f.write(f"Key Skills:{KeySkills}")
                    f.write("***********")
if __name__ == "__main__":
    while True:
        ScrapeJobs()
        Waiting_Time = 1
        print(f"Waiting For 6 seconds")
        time.sleep(6*Waiting_Time)
```