```
import re
from PyPDF2 import PdfReader
# Define the function to extract data from PDF
def get_data(filename):
    with open(filename, 'rb') as pdf_file:
        read_pdf = PdfReader(pdf_file)
        number_of_pages = len(read_pdf.pages)
        page = read_pdf.pages[0]
        page_content = page.extract_text()
        print(page_content.encode('utf-8'))

        # Example regular expression matching
        prog = re.compile("\s*(Name|name|nick).*")
        result = prog.match("Name: Shreya Meher")

        if result:
            print(result.group(0))

        result = prog.match("University: MIT")

        if result:
            print(result.group(0))

# List of filenames
files_list = ['/content/shreya_meher_Resume.pdf']

# Iterate over the list and call get_data function
for filename in files_list:
    get_data(filename)
```

```
b' \n \n \nSHREYA MEHER  \n           \xef\x80\xaa: shreya .meher017@nmims.edu.in |  \nwww.linkedin.com/in/shreya -meher
Name: Shreya Meher
```

```
!pip install PyPDF2
```

```
Collecting PyPDF2
  Downloading pypdf2-3.0.1-py3-none-any.whl (232 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 232.6/232.6 kB 3.9 MB/s eta 0:00:00
Installing collected packages: PyPDF2
Successfully installed PyPDF2-3.0.1
```

```
#to check if applicant is right to hire or not.
def check_skill_match(job_skills, applicant_skills):
    """
    Check if applicant's skills match with the required job skills.

    Args:
    job_skills (list): List of skills required for the job.
    applicant_skills (list): List of skills possessed by the applicant.

    Returns:
    bool: True if there is a match, False otherwise.
    """
    # Convert both lists to sets for efficient comparison
    job_skills_set = set(job_skills)
    applicant_skills_set = set(applicant_skills)

    # Check if the intersection of the sets is non-empty
    if job_skills_set.intersection(applicant_skills_set):
        return True
    else:
        return False

# Example usage:
job_skills = ["Python", "Machine Learning", "Data Analysis"]
applicant_skills = ["Python", "SQL", "Data Visualization"]

if check_skill_match(job_skills, applicant_skills):
    print("Applicant's skills match with the job requirements.")
else:
    print("Applicant's skills do not match with the job requirements.")
```

```
 Applicant's skills match with the job requirements.
```

# Introduction:

The dataset titled "Employability Classification of Over 70,000 Job Applicants" contains a comprehensive collection of information regarding job applicants and their respective employability scores. The dataset has been compiled to assist organizations and recruiters

in evaluating the suitability of candidates for various employment opportunities. By utilizing machine learning techniques, this dataset aims to provide valuable insights into the factors influencing employability and enhance the efficiency of the hiring process.

```
#importing libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns



df=pd.read_csv("/content/stackoverflow_full.csv")


df
```

| | Unnamed: 0 | Age | Accessibility | EdLevel | Employment | Gender | MentalHealth | MainBranch | YearsCode | YearsCode |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | <35 | No | Master | 1 | Man | No | Dev | 7 | |
| **1** | 1 | <35 | No | Undergraduate | 1 | Man | No | Dev | 12 | |
| **2** | 2 | <35 | No | Master | 1 | Man | No | Dev | 15 | |
| **3** | 3 | <35 | No | Undergraduate | 1 | Man | No | Dev | 9 | |
| **4** | 4 | >35 | No | PhD | 0 | Man | No | NotDev | 40 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **73457** | 73457 | <35 | No | Undergraduate | 1 | Man | No | Dev | 7 | |
| **73458** | 73458 | >35 | No | Undergraduate | 1 | Man | No | Dev | 21 | |
| **73459** | 73459 | <35 | No | Undergraduate | 1 | Man | No | Dev | 4 | |
| **73460** | 73460 | <35 | Yes | Undergraduate | 1 | Man | Yes | Dev | 5 | |
| **73461** | 73461 | <35 | No | Master | 1 | NonBinary | No | Dev | 10 | |

73462 rows × 14 columns

```
df.shape
```

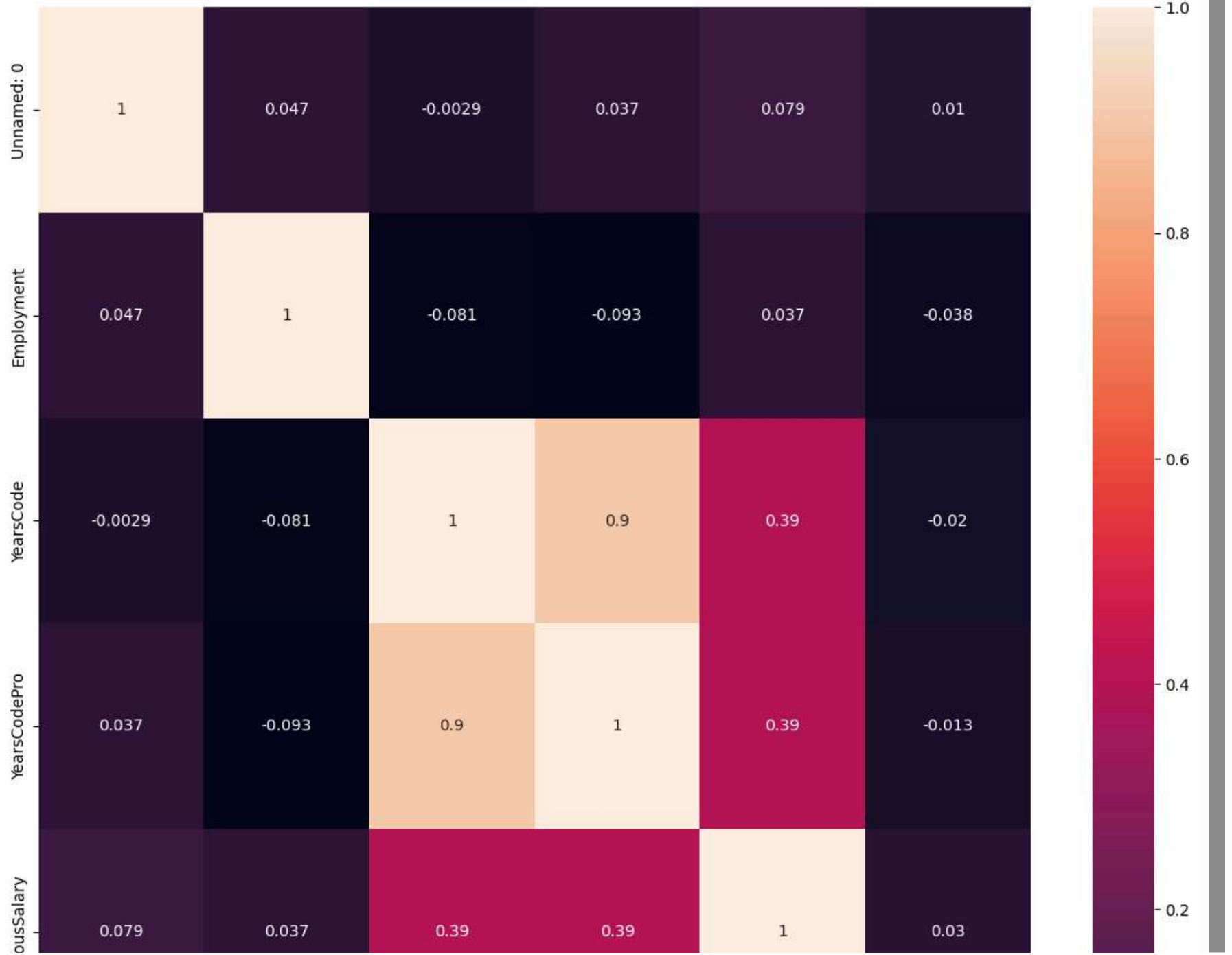
```
(73462, 14)
```

```
df.describe()
```

| | Unnamed: 0 | Employment | YearsCode | YearsCodePro | PreviousSalary | ComputerSkills |
|---|---|---|---|---|---|---|
| **count** | 73462.000000 | 73462.000000 | 73462.000000 | 73462.000000 | 73462.000000 | 73462.000000 |
| **mean** | 36730.500000 | 0.883096 | 14.218902 | 9.098377 | 67750.260611 | 13.428221 |
| **std** | 21206.797075 | 0.321308 | 9.405172 | 7.960201 | 49488.142118 | 7.057835 |
| **min** | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 1.000000 | 0.000000 |
| **25%** | 18365.250000 | 1.000000 | 7.000000 | 3.000000 | 28839.000000 | 8.000000 |
| **50%** | 36730.500000 | 1.000000 | 12.000000 | 7.000000 | 57588.000000 | 13.000000 |
| **75%** | 55095.750000 | 1.000000 | 20.000000 | 12.000000 | 95979.000000 | 17.000000 |
| **max** | 73461.000000 | 1.000000 | 50.000000 | 50.000000 | 224000.000000 | 107.000000 |

```
plt.subplots(figsize=(14,14))
sns.heatmap(df.corr(), annot=True);
```

```
<ipython-input-23-c82e4508676e>:2: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated.
  sns.heatmap(df.corr(), annot=True);
```




```
df.drop(df[df["Age"] == '>35'].index, inplace=True)
df.drop(df[df["Accessibility"] == 'No'].index, inplace=True)
df.drop(df[df["MentalHealth"] == 'No'].index, inplace=True)


df.drop(df[ (df["EdLevel"] != 'Master') & (df["EdLevel"] != 'PhD') ].index, inplace=True)


df.drop(df[df["MainBranch"] == 'NotDev'].index, inplace=True)


df = df.drop(df[df["YearsCodePro"] < 5].index)
df = df.drop(df[df["YearsCode"] <= 10].index)


df.drop(df[df["ComputerSkills"] <= 8].index, inplace=True)


df.shape
```

```
(26, 14)
```

Double-click (or enter) to edit

```
df
```

| | Unnamed: 0 | Age | Accessibility | EdLevel | Employment | Gender | MentalHealth | MainBranch | YearsCode | YearsCodePro |
|---|---|---|---|---|---|---|---|---|---|---|
| **8596** | 8596 | <35 | Yes | Master | 1 | Man | Yes | Dev | 16 | 11 |
| **8681** | 8681 | <35 | Yes | Master | 0 | Man | Yes | Dev | 16 | 7 |
| **8805** | 8805 | <35 | Yes | Master | 0 | Man | Yes | Dev | 14 | 12 |
| **20600** | 20600 | <35 | Yes | Master | 1 | Man | Yes | Dev | 12 | 8 |
| **26894** | 26894 | <35 | Yes | Master | 1 | NonBinary | Yes | Dev | 16 | 5 |
| **30780** | 30780 | <35 | Yes | Master | 1 | Man | Yes | Dev | 11 | 8 |
| **34291** | 34291 | <35 | Yes | Master | 1 | NonBinary | Yes | Dev | 13 | 6 |
| **34597** | 34597 | <35 | Yes | Master | 1 | Man | Yes | Dev | 15 | 6 |
| **37045** | 37045 | <35 | Yes | PhD | 1 | Man | Yes | Dev | 15 | 15 |
| **37670** | 37670 | <35 | Yes | Master | 1 | Man | Yes | Dev | 11 | 7 |
| **39890** | 39890 | <35 | Yes | Master | 0 | Man | Yes | Dev | 15 | 10 |
| **43941** | 43941 | <35 | Yes | Master | 1 | Man | Yes | Dev | 11 | 6 |
| **45004** | 45004 | <35 | Yes | Master | 1 | Man | Yes | Dev | 15 | 11 |
| **45595** | 45595 | <35 | Yes | Master | 1 | Man | Yes | Dev | 17 | 12 |
| **48525** | 48525 | <35 | Yes | Master | 1 | Man | Yes | Dev | 22 | 8 |
| **49828** | 49828 | <35 | Yes | Master | 1 | Woman | Yes | Dev | 12 | 7 |
| **51127** | 51127 | <35 | Yes | Master | 1 | Man | Yes | Dev | 15 | 17 |
| **54379** | 54379 | <35 | Yes | Master | 1 | NonBinary | Yes | Dev | 12 | 7 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **57970** | 57970 | <35 | Yes | Master | 1 | NonBinary | Yes | Dev | 40 | 36 |
| **58658** | 58658 | <35 | Yes | Master | 1 | Man | Yes | Dev | 12 | 12 |
| **59449** | 59449 | <35 | Yes | Master | 1 | NonBinary | Yes | Dev | 16 | 10 |
| **60263** | 60263 | <35 | Yes | Master | 1 | Man | Yes | Dev | 20 | 8 |
| **63231** | 63231 | <35 | Yes | Master | 1 | Man | Yes | Dev | 16 | 9 |
| **67807** | 67807 | <35 | Yes | Master | 1 | Woman | Yes | Dev | 14 | 6 |
| **72177** | 72177 | <35 | Yes | Master | 1 | Man | Yes | Dev | 13 | 9 |
| **73248** | 73248 | <35 | Yes | Master | 1 | Man | Yes | Dev | 12 | 11 |

```
!pip install tkcalendar
```

```
Collecting tkcalendar
  Downloading tkcalendar-1.6.1-py3-none-any.whl (40 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 40.9/40.9 kB 2.1 MB/s eta 0:00:00
Requirement already satisfied: babel in /usr/local/lib/python3.10/dist-packages (from tkcalendar) (2.14.0)
Installing collected packages: tkcalendar
Successfully installed tkcalendar-1.6.1
```

Certainly! Building an expense management system in Python is a practical project that can help you organize and track your financial transactions. Let's create a simple Expense Tracker using Python. In this example, we'll use the Tkinter library for the graphical user interface (GUI) and SQLite for the database. Here are the steps to get started:

```
#Import Necessary Modules and Libraries:
import datetime
import sqlite3
from tkcalendar import DateEntry
from tkinter import *
import tkinter.messagebox as mb
import tkinter.ttk as ttk


import tkinter as tk
# Set GUI window properties
!export DISPLAY=:0
root = tk.Tk()
```

```
---------------------------------------------------------------------------
TclError                                  Traceback (most recent call last)
<ipython-input-30-ed7a9fcf4e48> in <cell line: 4>()
      2 # Set GUI window properties
      3 get_ipython().system('export DISPLAY=:0')
----> 4 root = tk.Tk()

/usr/lib/python3.10/tkinter/__init__.py in __init__(self, screenName, baseName, className, useTk, sync, use)
   2297                 baseName = baseName + ext
   2298         interactive = False
-> 2299         self.tk = _tkinter.create(screenName, baseName, className, interactive, wantobjects, useTk, sync, use)
   2300         if useTk:
   2301             self._loadtk()

TclError: no display name and no $DISPLAY environment variable
```