PROJECT: TECH TALENT RECRUITING WITH REGEX

As a Data Analyst at a leading global HR consultancy, your mission is to delve into an extensive database of resumes to identify suitable candidates for tech-focused roles. This task involves using regular expressions to extract key data points and applying data preprocessing techniques to organize this information effectively.

Dataset Summary

`resumes.csv`

Column	Data Type	Description
`ID`	float	Unique identifier for each resume.
`Resume_str`	object	Full text of the resume, rich with details for analysis.
`Category`	object	Job category of the resume, indicating the field of expertise.

Let's Get Started!

Embark on this analytical journey to harness advanced data analysis techniques for real-world HR challenges. This project is your chance to impact the hiring process by ensuring that tech talent finds their ideal job. Let's begin this exciting journey!

```python
import pandas as pd
import re

# Load the resume dataset from a CSV file into a DataFrame
resumes = pd.read_csv('resumes.csv')
resumes.sample(3)
```

index	ID	Resume_str	Cate
258	24083609	INFORMATION TECHNOLOGY SPECIALIST (INFOSEC) Summary Retired Information Assurance Systems Security Certification Specialist responsible for man...	INFO
1087	15620421	CENTER SALES Summary Results-oriented customer service manager with diverse background in management and customer service. Dedicated to providin...	SALE
311	10641230	IT MANAGEMENT Career Overview Detail-oriented professional with extensive Information Technology experience in hardware and software troubleshootin...	INFO

Rows: 3 Expand

```python
import pandas as pd
import re

# Load data
resumes = pd.read_csv('resumes.csv')

# Define a function to extract required info from resume string
def parse_resume(row):
    resume = row['Resume_str']

    # Extract job title (assumes first 5 words as a placeholder)
    job_title_match = ' '.join(resume.split()[:5])

    # Extract tech skills
    tech_skills_found = re.findall(r'\b(Python|SQL|R|Excel)\b', resume, re.IGNORECASE)
    tech_skills = list(set([skill.capitalize() for skill in tech_skills_found]))

    # Extract education
    edu_match = re.search(r'\b(PhD|Master|Bachelor)\b', resume, re.IGNORECASE)
    education = edu_match.group(0).capitalize() if edu_match else None

    return pd.Series({
        'id': row['ID'],
        'job_title': job_title_match,
        'tech_skills': ', '.join(tech_skills) if tech_skills else None,
        'education': education
    })

# Apply parsing function
candidates_df = resumes.apply(parse_resume, axis=1)

# Drop rows with missing or empty values
candidates_df.replace('', pd.NA, inplace=True)
candidates_df.dropna(subset=['id', 'job_title', 'tech_skills', 'education'], inplace=True)

# Preview the clean DataFrame
print(candidates_df.head())
```

```
            id  ... education
2   33176873.0  ...    Master
4   17812897.0  ...  Bachelor
8   11847784.0  ...  Bachelor
9   32896934.0  ...  Bachelor
16  93002334.0  ...  Bachelor

[5 rows x 4 columns]
```