In [1]:
```python
import pandas as pd
import os
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
import missingno as msno
```

Data Preprocessing

In [2]:
```python
df=pd.read_csv("fake.csv")
```

In [3]:
```python
df.head()
```

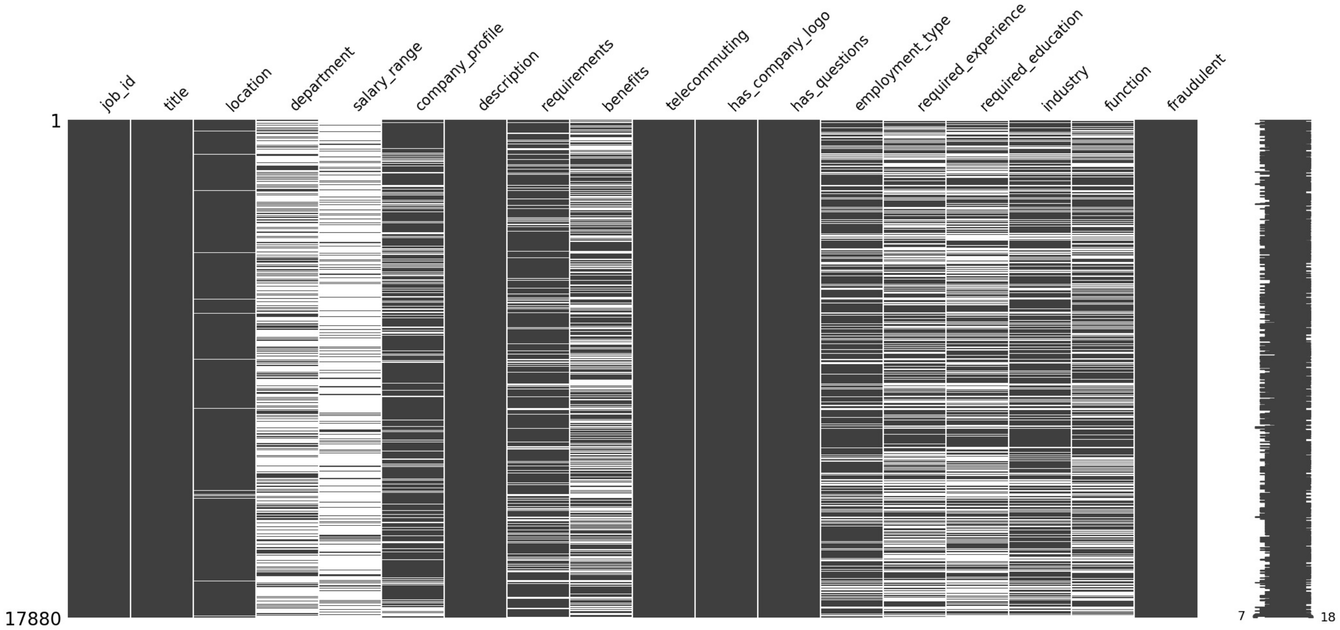
Out[3]:

	job_id	title	location	department	salary_range	company_profile	description	requirements	benefits	tel
0	1	Marketing Intern	US, NY, New York	Marketing	NaN	We're Food52, and we've created a groundbreaki...	Food52, a fast-growing, James Beard Award-winn...	Experience with content management systems a m...	NaN	
1	2	Customer Service - Cloud Video Production	NZ, , Auckland	Success	NaN	90 Seconds, the worlds Cloud Video Production ...	Organised - Focused - Vibrant - Awesome!Do you...	What we expect from you:Your key responsibilit...	What you will get from usThrough being part of...	
2	3	Commissioning Machinery Assistant (CMA)	US, IA, Wever	NaN	NaN	Valor Services provides Workforce Solutions th...	Our client, located in Houston, is actively se...	Implement pre-commissioning and commissioning ...	NaN	
3	4	Account Executive - Washington DC	US, DC, Washington	Sales	NaN	Our passion for improving quality of life thro...	THE COMPANY: ESRI – Environmental Systems Rese...	EDUCATION: Bachelor's or Master's in GIS, busi...	Our culture is anything but corporate—we have ...	
4	5	Bill Review Manager	US, FL, Fort Worth	NaN	NaN	SpotSource Solutions LLC is a Global Human Cap...	JOB TITLE: Itemization Review ManagerLOCATION:...	QUALIFICATIONS:RN license in the State of Texa...	Full Benefits Offered	

In [4]:
```python
df.shape
```

Out[4]:
```
(17880, 18)
```

In [5]:
```python
msno.matrix(df)
plt.show()
```



Lot of Null Values

In [6]:
```python
#filling null values, 'Not Applicable' and 'Unspecified' with 'Not Specified'
df.fillna('Not Specified', inplace=True)
df = df.replace(['Not Applicable','Unspecified'],'Not Specified')
```

In [7]:
```
df = df.drop(columns = ['job_id'])
```

In [8]:
```
#Label counts for each attribute
labelcountlist = []
for x in df.columns:
        labelcountlist.append((len(df[x].unique())))
labelcount = pd.DataFrame({'Attribute': df.columns, 'Count': labelcountlist})
print(labelcount)
```

```
              Attribute  Count
0                 title  11231
1              location   3106
2            department   1338
3          salary_range    875
4       company_profile   1710
5           description  14802
6          requirements  11969
7              benefits   6206
8         telecommuting      2
9      has_company_logo      2
10        has_questions      2
11      employment_type      6
12  required_experience      7
13   required_education     13
14             industry    132
15             function     38
16           fraudulent      2
```

In [9]:
```
df.head(20)
```

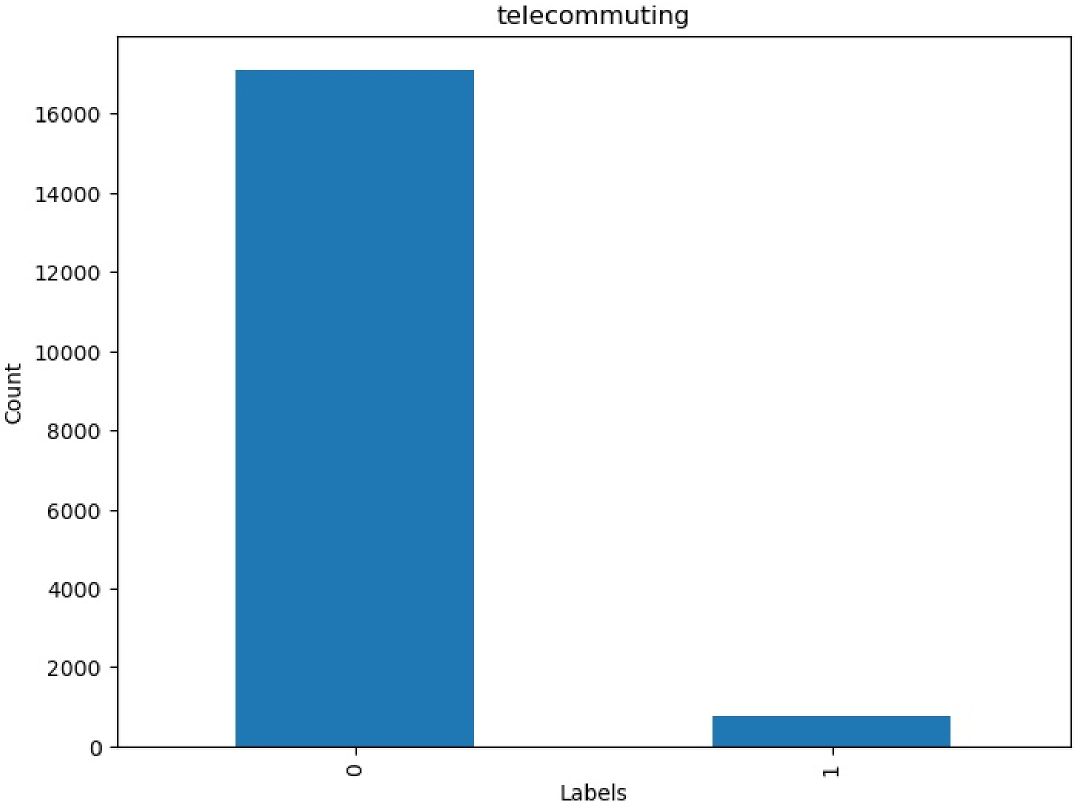
Out[9]:

	title	location	department	salary_range	company_profile	description	
0	Marketing Intern	US, NY, New York	Marketing	Not Specified	We're Food52, and we've created a groundbreaki...	Food52, a fast-growing, James Beard Award-winn...	Experience with co
1	Customer Service - Cloud Video Production	NZ, , Auckland	Success	Not Specified	90 Seconds, the worlds Cloud Video Production ...	Organised - Focused - Vibrant - Awesome!Do you...	What we expec
2	Commissioning Machinery Assistant (CMA)	US, IA, Wever	Not Specified	Not Specified	Valor Services provides Workforce Solutions th...	Our client, located in Houston, is actively se...	Implement pre-
3	Account Executive - Washington DC	US, DC, Washington	Sales	Not Specified	Our passion for improving quality of life thro...	THE COMPANY: ESRI – Environmental Systems Rese...	EDUCATION: Bachelor's
4	Bill Review Manager	US, FL, Fort Worth	Not Specified	Not Specified	SpotSource Solutions LLC is a Global Human Cap...	JOB TITLE: Itemization Review ManagerLOCATION:...	QUALIFICATIONS:RN lic
5	Accounting Clerk	US, MD,	Not Specified	Not Specified	Not Specified	Job OverviewApex is an environmental consultin...	
6	Head of Content (m/f)	DE, BE, Berlin	ANDROIDPIT	20000-28000	Founded in 2009, the Fonpit AG rose with its i...	Your Responsibilities: Manage the English-spea...	Your Know-How:
7	Lead Guest Service Specialist	US, CA, San Francisco	Not Specified	Not Specified	Airenvy's mission is to provide lucrative yet ...	Who is Airenvy?Hey there! We are seasoned entr...	Experience with CRM
8	HP BSM SME	US, FL, Pensacola	Not Specified	Not Specified	Solutions3 is a woman-owned small business who...	Implementation/Configuration/Testing/Training ...	MUST BE A US CITIZE
9	Customer Service Associate - Part Time	US, AZ, Phoenix	Not Specified	Not Specified	Novitex Enterprise Solutions, formerly Pitney ...	The Customer Service Associate will be based i...	Minimum Requiren
10	ASP.net Developer Job opportunity at United St...	US, NJ, Jersey City	Not Specified	100000-120000	Not Specified	Position : #URL_86fd830a95a64e2b30ceed829e63fd...	#URL_86fd830a95a64e2b
11	Talent Sourcer (6 months fixed-term contract)	GB, LND, London	HR	Not Specified	Want to build a 21st century financial service...	TransferWise is the clever new way to move mon...	We're looking for someon
12	Applications Developer, Digital	US, CT, Stamford	Not Specified	Not Specified	Novitex Enterprise Solutions, formerly Pitney ...	The Applications Developer, Digital will devel...	Requirements:4 – 5 y
13	Installers	US, FL, Orlando	Not Specified	Not Specified	Growing event production company providing sta...	Event Industry Installers Needed!! (Orlando, F...	Valid driver's licens
14	Account Executive - Sydney	AU, NSW, Sydney	Sales	Not Specified	Adthena is the UK's leading competitive intell...	Are you interested in a satisfying and financi...	You'll need to be smart a
15	VP of Sales - Vault Dragon	SG, 01, Singapore	Sales	120000-150000	Jungle Ventures is the leading Singapore based...	About Vault Dragon Vault Dragon is Dropbox for...	Key Superpowers3-5 yea
16	Hands-On QA Leader	IL, , Tel Aviv, Israel	R&D	Not Specified	At HoneyBook we're re-imagining the events ind...	We are looking for a Hands-On QA Leader for ou...	Previous experience in
17	Southend-on-Sea Traineeships Under NAS 16-18 Y...	GB, SOS, Southend-on-Sea	Not Specified	Not Specified	Established on the principles that full time e...	Government funding is only available for 16-18...	16-18 year olds only
18	Visual Designer	US, NY, New York	Not Specified	Not Specified	Kettle is an independent digital agency based ...	Kettle is hiring a Visual Designer!Job Locatio...	
19	Process Controls Engineer - DCS PLC MS Office ...	US, PA, USA Northeast	Not Specified	Not Specified	We Provide Full Time Permanent Positions for m...	Experienced Process Controls Engineer is requi...	Must have 5 or more y

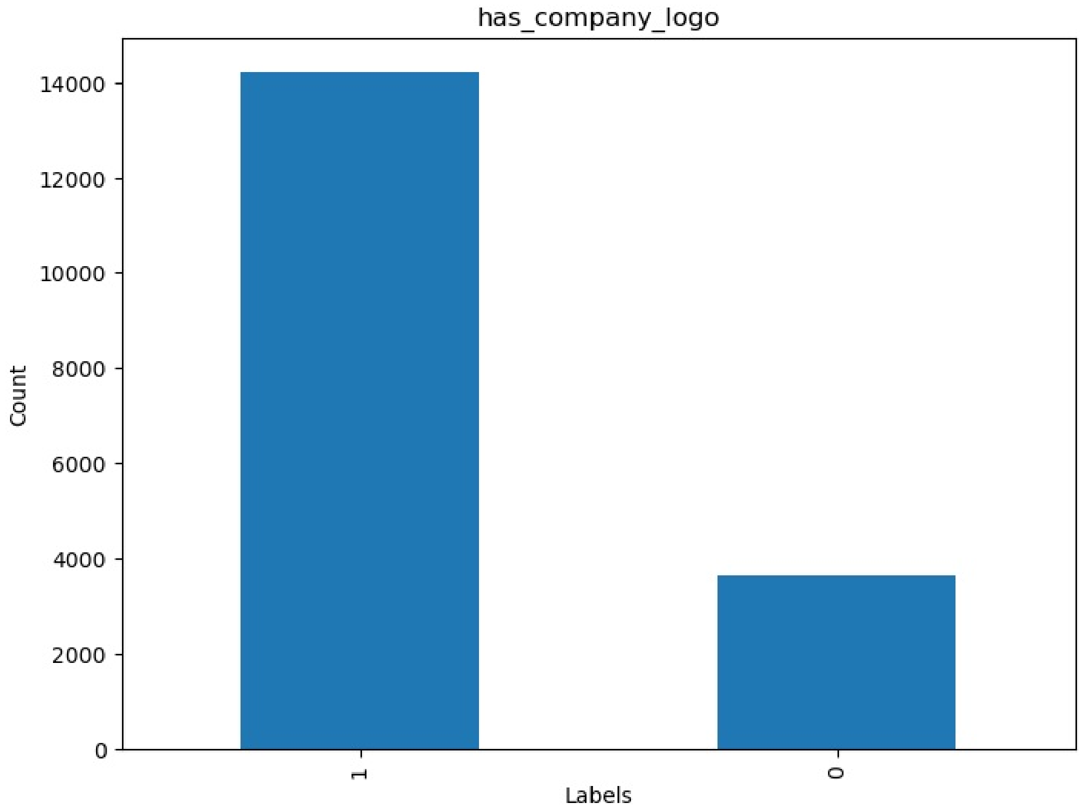
EDA

In [10]:
```
# Calculate the count of unique labels for each attribute
labelcount = df.nunique().reset_index()
labelcount.columns = ['Attribute', 'Count']

# Filter attributes with less than 100 unique labels
filtered_labels = labelcount[labelcount['Count'] < 100]['Attribute'].tolist()

# Store the filtered labels for comprehensible visualization
label = []

# Iterate over the filtered labels
for attr in filtered_labels:
    print('\n' + attr + '\n----------')
    unique_vals = df[attr].unique()
    print(str(list(unique_vals)) + "\n")
    print(df[attr].value_counts())
    label.append(attr)
# Plot a bar graph showing the count of each label
    plt.figure(figsize=(8, 6))
    df[attr].value_counts().plot(kind='bar')
    plt.title(attr)
    plt.xlabel('Labels')
    plt.ylabel('Count')
    plt.show()
```

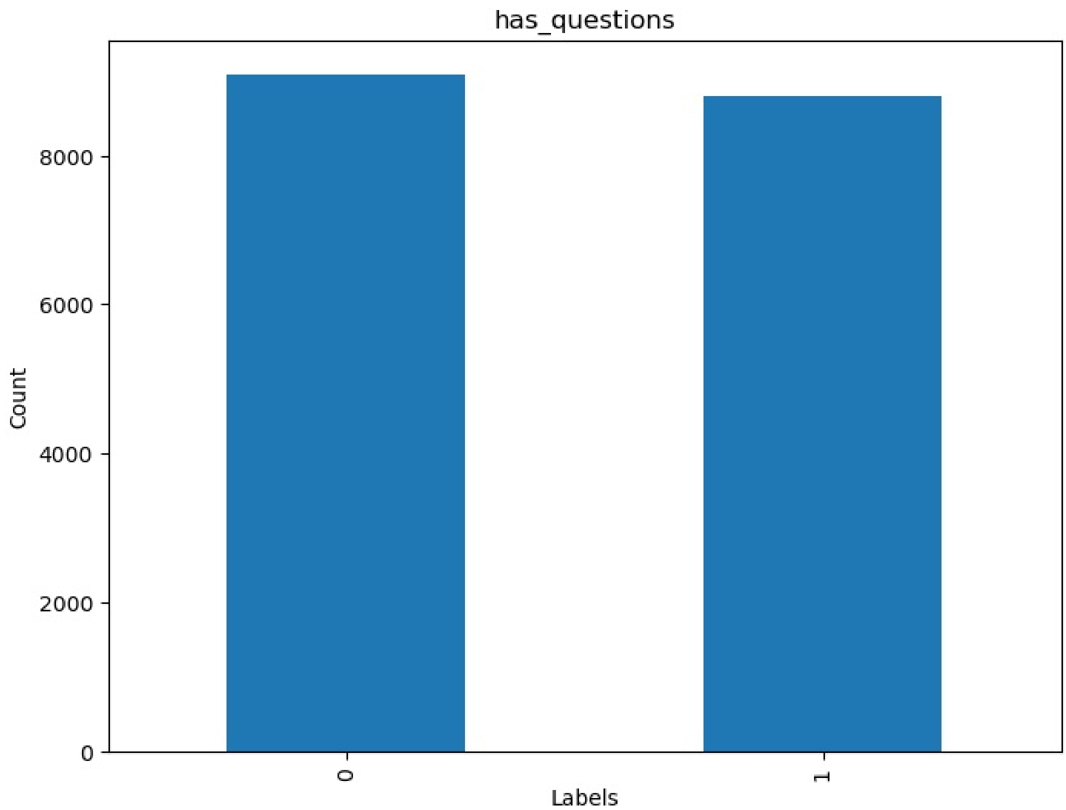
```
telecommuting
----------
[0, 1]

0    17113
1      767
Name: telecommuting, dtype: int64
```



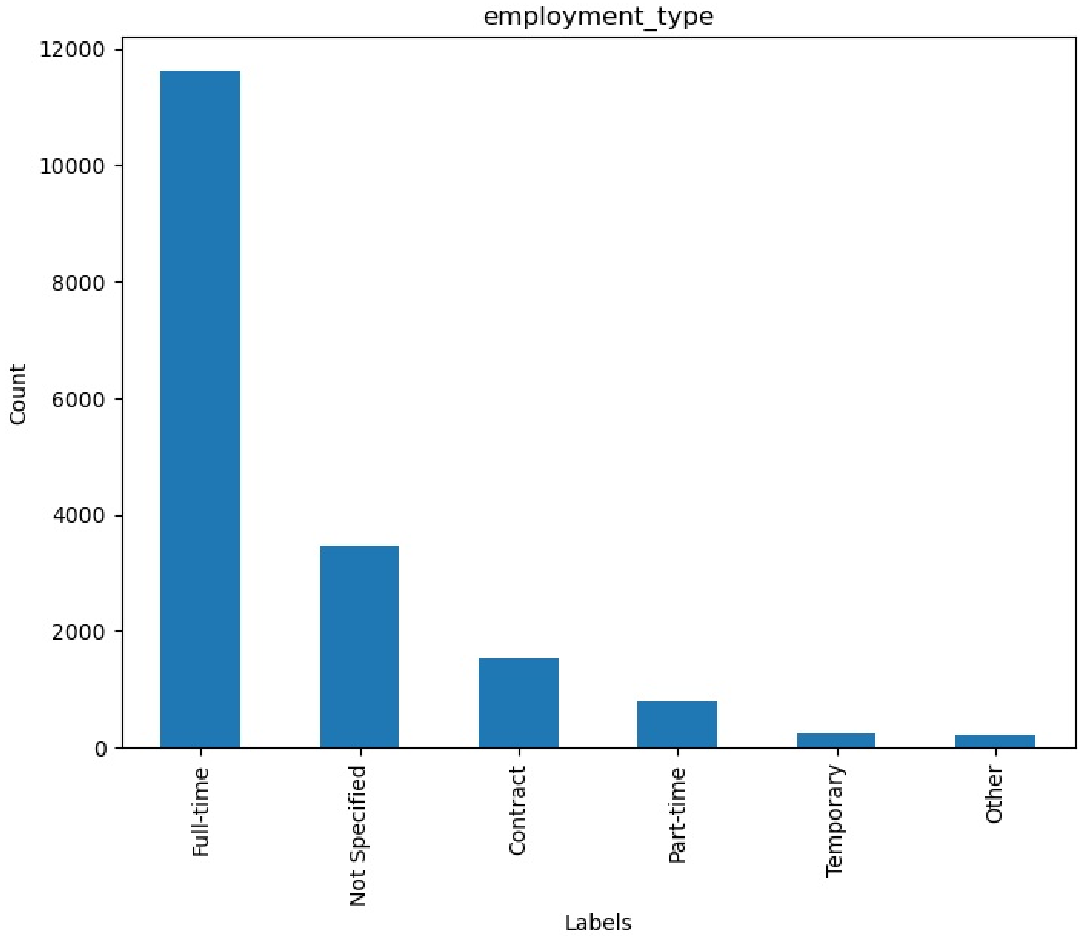

```
has_company_logo
----------
[1, 0]

1    14220
0     3660
Name: has_company_logo, dtype: int64
```



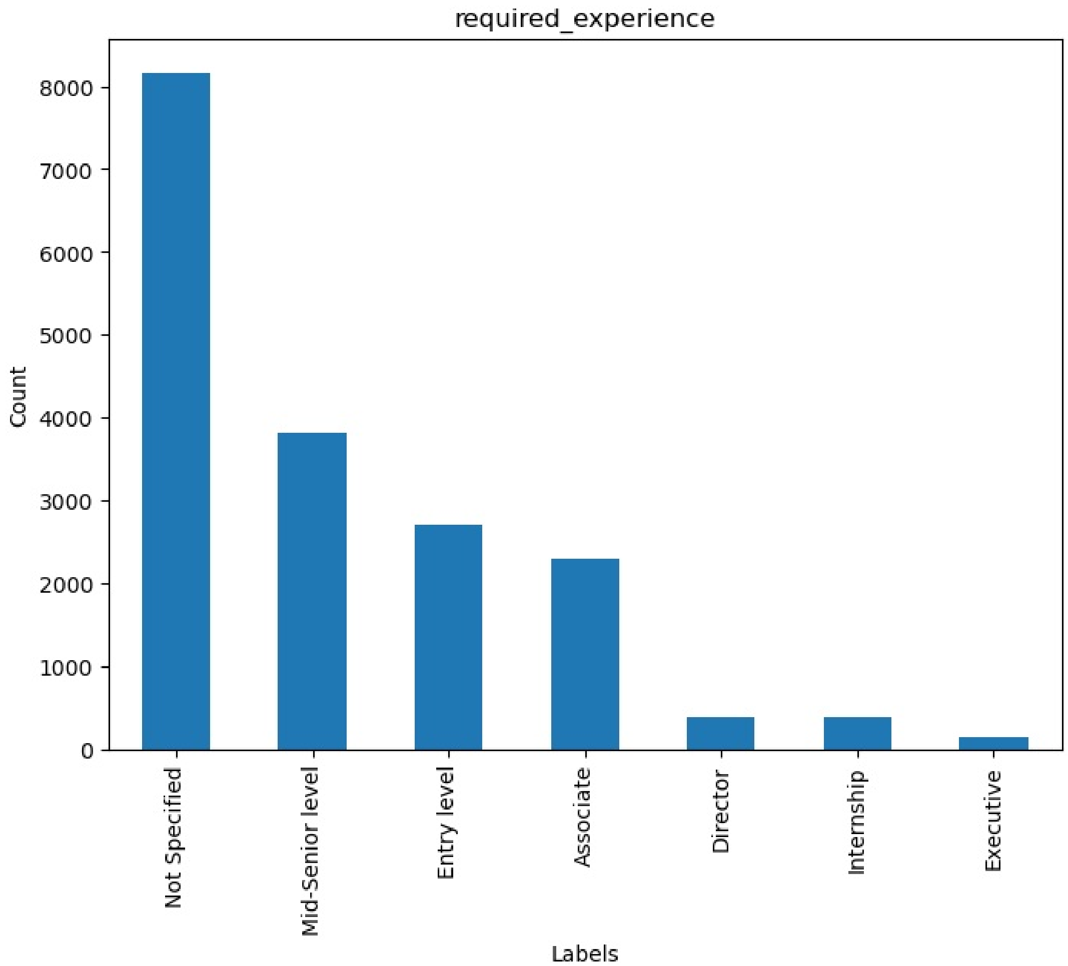

```
has_questions
----------
[0, 1]

0    9088
1    8792
Name: has_questions, dtype: int64
```



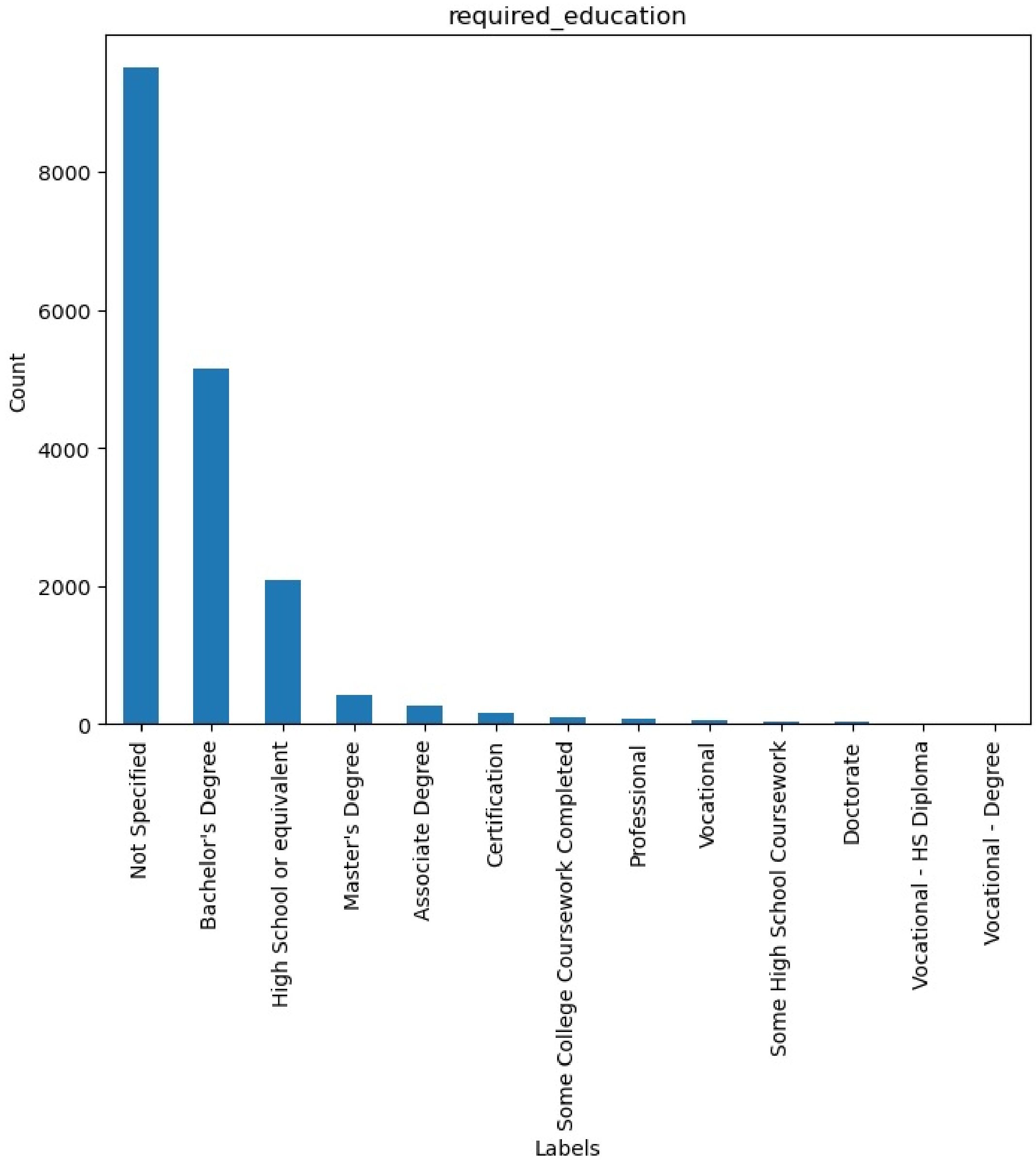
```
employment_type
----------
['Other', 'Full-time', 'Not Specified', 'Part-time', 'Contract', 'Temporary']

Full-time        11620
Not Specified     3471
Contract          1524
Part-time          797
Temporary          241
Other              227
Name: employment_type, dtype: int64
```



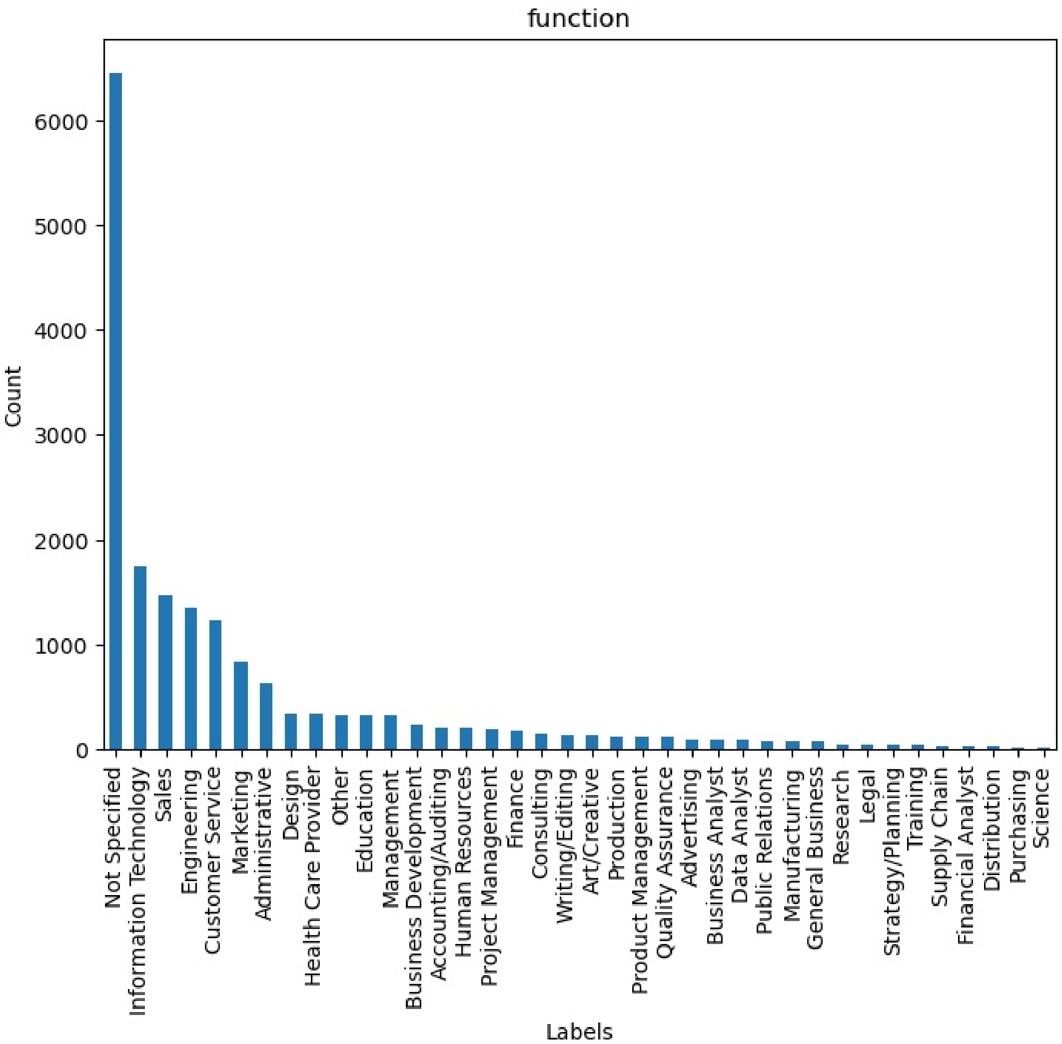

```
required_experience
----------
['Internship', 'Not Specified', 'Mid-Senior level', 'Associate', 'Entry level', 'Executive', 'Director']

Not Specified       8166
Mid-Senior level    3809
Entry level         2697
Associate           2297
Director             389
Internship           381
Executive            141
Name: required_experience, dtype: int64
```



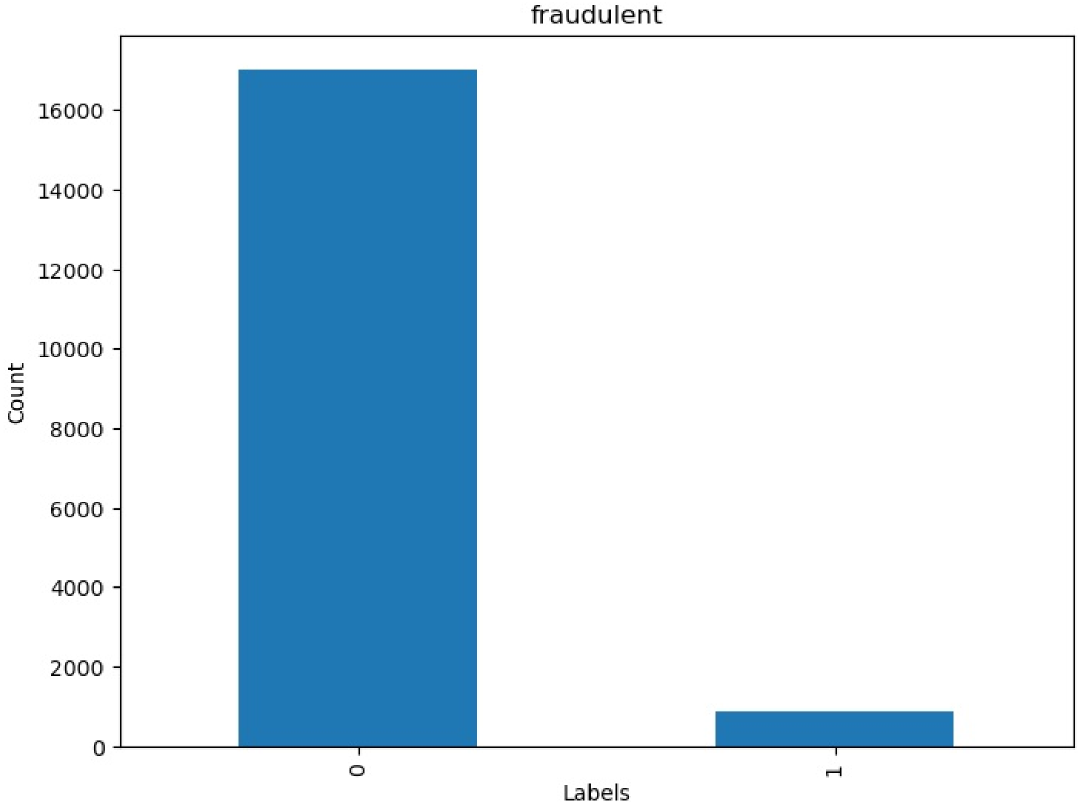

```
required_education
----------
['Not Specified', "Bachelor's Degree", "Master's Degree", 'High School or equivalent', 'Some College Coursework Completed', 'Vocational', 'Certification', 'Associate Degree', 'Professional', 'Doctorate', 'Some High School Coursework', 'Vocational - Degree', 'Vocational - HS Diploma']

Not Specified                        9502
Bachelor's Degree                    5145
High School or equivalent            2080
Master's Degree                       416
Associate Degree                      274
Certification                         170
Some College Coursework Completed     102
Professional                           74
Vocational                             49
Some High School Coursework            27
Doctorate                              26
Vocational - HS Diploma                 9
Vocational - Degree                     6
Name: required_education, dtype: int64
```


required_education
Count
8000
6000
4000
2000
0
Not Specified
Bachelor's Degree
High School or equivalent
Master's Degree
Associate Degree
Certification
Some College Coursework Completed
Professional
Vocational
Some High School Coursework
Doctorate
Vocational - HS Diploma
Vocational - Degree
Labels

```
function
----------
['Marketing', 'Customer Service', 'Not Specified', 'Sales', 'Health Care Provider', 'Management', 'Information
Technology', 'Other', 'Engineering', 'Administrative', 'Design', 'Production', 'Education', 'Supply Chain', 'Bu
siness Development', 'Product Management', 'Financial Analyst', 'Consulting', 'Human Resources', 'Project Manag
ement', 'Manufacturing', 'Public Relations', 'Strategy/Planning', 'Advertising', 'Finance', 'General Business',
'Research', 'Accounting/Auditing', 'Art/Creative', 'Quality Assurance', 'Data Analyst', 'Business Analyst', 'Wr
iting/Editing', 'Distribution', 'Science', 'Training', 'Purchasing', 'Legal']

Not Specified             6455
Information Technology    1749
Sales                     1468
Engineering               1348
Customer Service          1229
Marketing                  830
Administrative             630
Design                     340
Health Care Provider       338
Other                      325
Education                  325
Management                 317
Business Development       228
Accounting/Auditing        212
Human Resources            205
Project Management         183
Finance                    172
Consulting                 144
Writing/Editing            132
Art/Creative               132
Production                 116
Product Management         114
Quality Assurance          111
Advertising                 90
Business Analyst            84
Data Analyst                82
Public Relations            76
Manufacturing               74
General Business            68
Research                    50
Legal                       47
Strategy/Planning           46
Training                    38
Supply Chain                36
Financial Analyst           33
Distribution                24
Purchasing                  15
Science                     14
Name: function, dtype: int64
```


function
Count
Labels
0
1000
2000
3000
4000
5000
6000
Not Specified
Information Technology
Sales
Engineering
Customer Service
Marketing
Administrative
Design
Health Care Provider
Other
Education
Management
Business Development
Accounting/Auditing
Human Resources
Project Management
Finance
Consulting
Writing/Editing
Art/Creative
Production
Product Management
Quality Assurance
Advertising
Business Analyst
Data Analyst
Public Relations
Manufacturing
General Business
Research
Legal
Strategy/Planning
Training
Supply Chain
Financial Analyst
Distribution
Purchasing
Science

```
fraudulent
----------
[0, 1]


0    17014
1      866
Name: fraudulent, dtype: int64
```



Conclusion

From above we can say that Telecommuting: The dataset contains information about telecommuting, with most job postings (approximately 95%) indicating that telecommuting is not available.

Company Logo: The majority of job postings (around 80%) have a company logo, indicating that employers often include a logo in their job postings.

Questions: The presence of questions in job postings is relatively balanced, with a similar number of postings having questions (approximately 50%) and not having questions.

Employment Type: The dataset includes various types of employment, with full-time positions being the most common (over 70% of job postings). Other types include not specified, contract, part-time, temporary, and other.

Required Experience: The required experience for job postings varies, with a significant portion (around 40%) not specifying any particular experience level. The remaining postings indicate different levels, such as mid-senior level, entry level, associate director, internship, and executive.

Required Education: The required education for job postings is diverse, with a considerable number (around 50%) not specifying any specific education requirement. Other education levels include bachelor's degree, high school or equivalent, master's degree, associate degree, certification, and various other categories.

Function: The dataset covers a wide range of job functions, with many postings (around 40%) not specifying a particular function. The most common functions include information technology, sales, engineering, customer service, marketing, and administrative.

Fraudulent: A small portion of the job postings (approximately 5%) are marked as fraudulent, suggesting that caution should be exercised when dealing with such postings.

NLP

In [11]:
```
import spacy
import nltk
import warnings
warnings.filterwarnings("ignore")
```

```python
from nltk.corpus import stopwords
from sklearn.metrics import *
from sklearn import preprocessing
from sklearn import metrics
```

In [12]:
```
pip install spacy
```

```
Requirement already satisfied: spacy in c:\users\acer\anaconda3\lib\site-packages (3.5.3)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in c:\users\acer\anaconda3\lib\site-packages (from spa
cy) (1.0.9)
Requirement already satisfied: smart-open<7.0.0,>=5.2.1 in c:\users\acer\anaconda3\lib\site-packages (from spac
y) (5.2.1)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in c:\users\acer\anaconda3\lib\site-packages (from spacy)
(3.0.8)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in c:\users\acer\anaconda3\lib\site-packages (from s
pacy) (3.0.12)
Requirement already satisfied: numpy>=1.15.0 in c:\users\acer\anaconda3\lib\site-packages (from spacy) (1.23.5)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in c:\users\acer\anaconda3\lib\site-packages (from spacy
) (2.28.1)
Requirement already satisfied: jinja2 in c:\users\acer\anaconda3\lib\site-packages (from spacy) (2.11.3)
Requirement already satisfied: packaging>=20.0 in c:\users\acer\anaconda3\lib\site-packages (from spacy) (21.3)
Requirement already satisfied: pathy>=0.10.0 in c:\users\acer\anaconda3\lib\site-packages (from spacy) (0.10.1)
Requirement already satisfied: thinc<8.2.0,>=8.1.8 in c:\users\acer\anaconda3\lib\site-packages (from spacy) (8
.1.10)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in c:\users\acer\anaconda3\lib\site-packages (from spacy) (2
.0.7)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<1.11.0,>=1.7.4 in c:\users\acer\anaconda3\lib\site-packag
es (from spacy) (1.10.8)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in c:\users\acer\anaconda3\lib\site-packages (from s
pacy) (1.0.4)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in c:\users\acer\anaconda3\lib\site-packages (from spacy) (2
.4.6)
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in c:\users\acer\anaconda3\lib\site-packages (from spacy
) (3.3.0)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in c:\users\acer\anaconda3\lib\site-packages (from spacy
) (2.0.8)
Requirement already satisfied: typer<0.8.0,>=0.3.0 in c:\users\acer\anaconda3\lib\site-packages (from spacy) (0
.7.0)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in c:\users\acer\anaconda3\lib\site-packages (from spacy) (4
.64.1)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in c:\users\acer\anaconda3\lib\site-packages (from spacy) (
1.1.1)
Requirement already satisfied: setuptools in c:\users\acer\anaconda3\lib\site-packages (from spacy) (63.4.1)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in c:\users\acer\anaconda3\lib\site-packages (from pack
aging>=20.0->spacy) (3.0.9)
Requirement already satisfied: typing-extensions>=4.2.0 in c:\users\acer\anaconda3\lib\site-packages (from pyda
ntic!=1.8,!=1.8.1,<1.11.0,>=1.7.4->spacy) (4.3.0)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\acer\anaconda3\lib\site-packages (from requ
ests<3.0.0,>=2.13.0->spacy) (2.0.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\acer\anaconda3\lib\site-packages (from request
s<3.0.0,>=2.13.0->spacy) (1.26.11)
Requirement already satisfied: idna<4,>=2.5 in c:\users\acer\anaconda3\lib\site-packages (from requests<3.0.0,>
=2.13.0->spacy) (3.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\acer\anaconda3\lib\site-packages (from requests<3
.0.0,>=2.13.0->spacy) (2022.9.14)
Requirement already satisfied: confection<1.0.0,>=0.0.1 in c:\users\acer\anaconda3\lib\site-packages (from thin
c<8.2.0,>=8.1.8->spacy) (0.0.4)
Requirement already satisfied: blis<0.8.0,>=0.7.8 in c:\users\acer\anaconda3\lib\site-packages (from thinc<8.2.
0,>=8.1.8->spacy) (0.7.9)
Requirement already satisfied: colorama in c:\users\acer\anaconda3\lib\site-packages (from tqdm<5.0.0,>=4.38.0-
>spacy) (0.4.6)
Requirement already satisfied: click<9.0.0,>=7.1.1 in c:\users\acer\anaconda3\lib\site-packages (from typer<0.8
.0,>=0.3.0->spacy) (8.0.4)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\acer\anaconda3\lib\site-packages (from jinja2->spac
y) (2.0.1)
Note: you may need to restart the kernel to use updated packages.
```

In [17]:
```python
#Remove stopword
nltk.download('stopwords')
stop = stopwords.words()
sym = "!@#$%^&*+-={}[]|\"':;<>,.?/`~()_" #SYMBOLS TO BE REMOVED
listsym = ([*sym])
listsym.append("'")
listsym.append('"')
```

```
[nltk_data] Error loading stopwords: <urlopen error [WinError 10060] A
[nltk_data]     connection attempt failed because the connected party
[nltk_data]     did not properly respond after a period of time, or
[nltk_data]     established connection failed because connected host
[nltk_data]     has failed to respond>
```

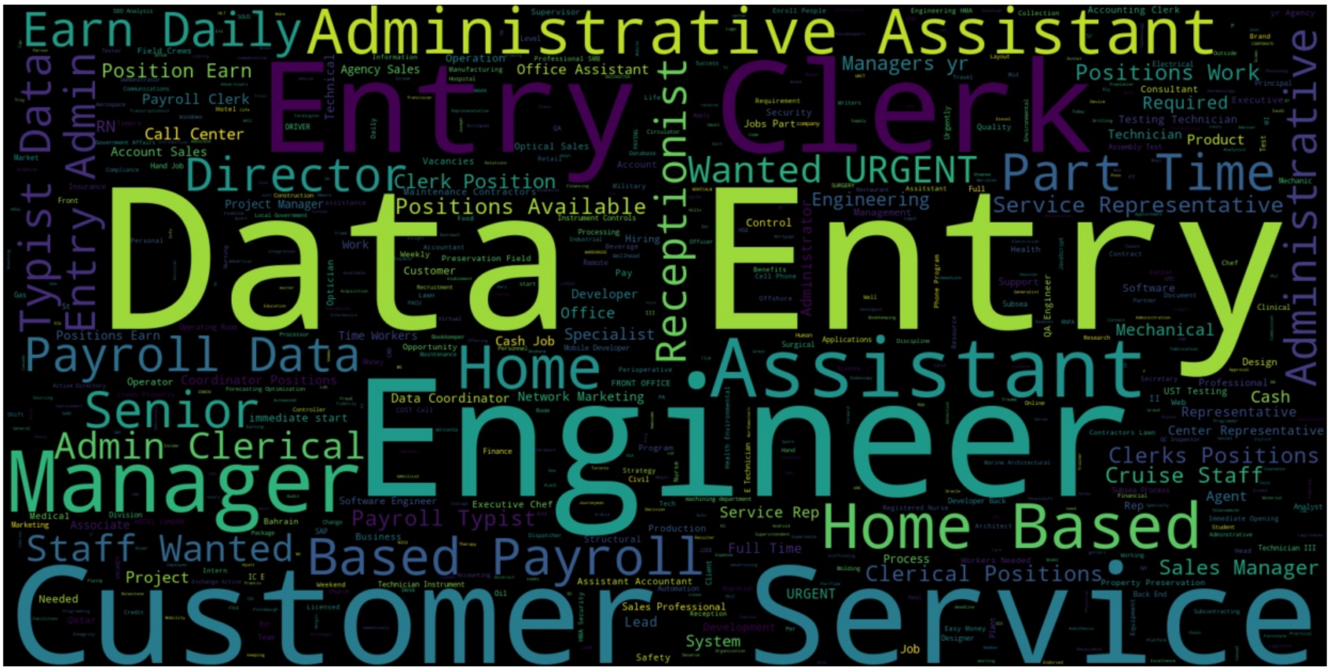
In [18]:
```python
string_labels = ['company_profile','description','requirements','benefits']
for label in string_labels:
    df[label] =  df[label].apply(lambda x: ' '.join([word for word in x.split() if word not in (stop)]))
    for j in range(df.shape[0]):
        for i in listsym:
            df.at[j,label] = df.at[j,label].replace(i,"")
```

In [19]:
```python
# checking Real and Fake Job words
```

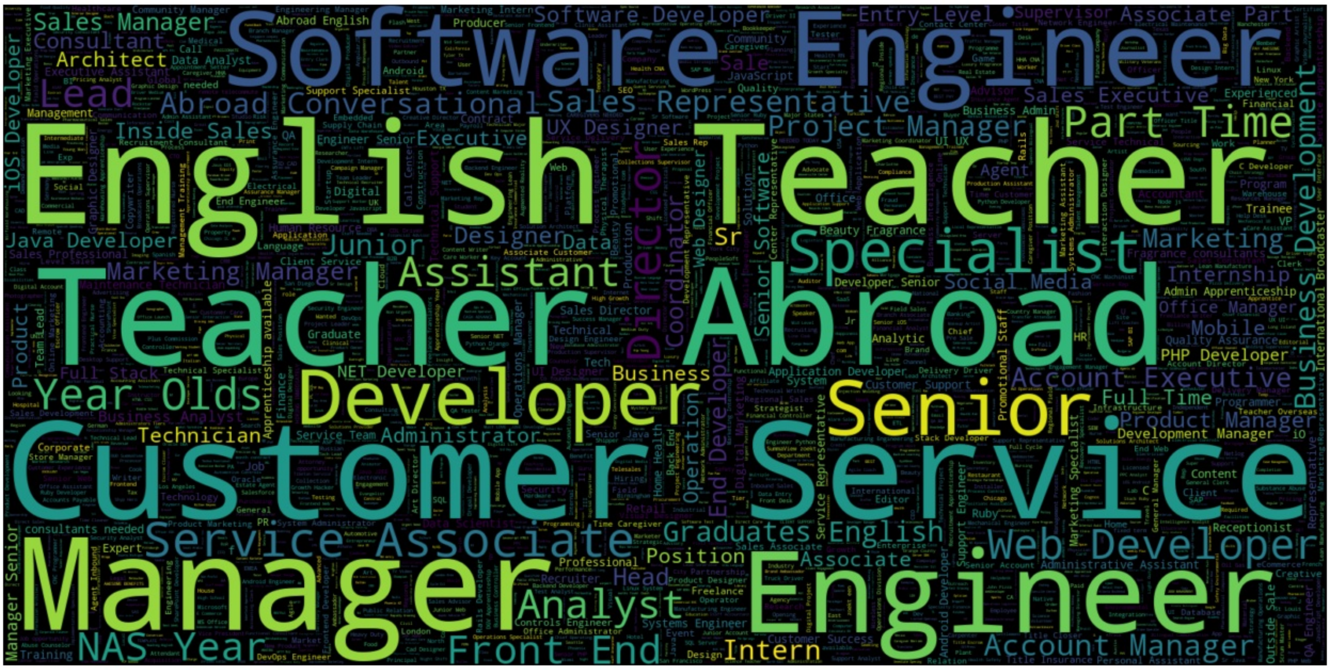
In [19]:
```
# Checking Real and Fake Job words
fraudjobs_text = df[df['fraudulent'] == 1]['title']
actualjobs_text = df[df['fraudulent'] == 0]['title']
```

In [20]:
```
import matplotlib.pyplot as plt
from wordcloud import WordCloud
from nltk.corpus import stopwords

# Download stopwords if not already downloaded
import nltk
nltk.download('stopwords')

# Define the stopwords
STOPWORDS = set(stopwords.words('english'))

# Create the word cloud
plt.figure(figsize=(16, 14))
wc = WordCloud(min_font_size=3, max_words=3000, width=1600, height=800, stopwords=STOPWORDS).generate(" ".join(
plt.imshow(wc, interpolation='bilinear')
plt.axis("off")
plt.show()
```

```
[nltk_data] Error loading stopwords: <urlopen error [WinError 10060] A
[nltk_data]     connection attempt failed because the connected party
[nltk_data]     did not properly respond after a period of time, or
[nltk_data]     established connection failed because connected host
[nltk_data]     has failed to respond>
```



In [21]:
```
# Create the word cloud
plt.figure(figsize=(16, 14))
wc = WordCloud(min_font_size=3, max_words=3000, width=1600, height=800, stopwords=STOPWORDS).generate(" ".join(
plt.imshow(wc, interpolation='bilinear')
plt.axis("off")
plt.show()
```




```
In [22]:
realcount = (df['fraudulent'] == 0).sum()  # Number of real applications
fakecount = (df['fraudulent'] == 1).sum()  # Number of fake applications

# FUNCTION TO CALCULATE THE NUMBER OF NOT SPECIFIED ENTRIES IN VARIOUS ATTRIBUTES ALONG WITH THE RATIO OF NOT S

def not_specified(labelname, name):
    df_real = df[df['fraudulent'] == 0][labelname]
    not_specreal = (df_real == 'Not Specified').sum()
    print(name + '\n-----------------\n\nREAL\n-----------')
    print(f"Number of Real applications that have not specified {name} = {not_specreal:.0f}")
    print(f"Number of Real applications = {realcount:.0f}")
    print(f"Ratio (Not Specified Real applications / Real applications) = {not_specreal / realcount:.6f}")

    df_fake = df[df['fraudulent'] == 1][labelname]
    not_specfake = (df_fake == 'Not Specified').sum()
    print('\n\nFAKE\n-----------')
    print(f"Number of Fake applications that have not specified {name} = {not_specfake:.0f}")
    print(f"Number of Fake applications = {fakecount:.0f}")
    print(f"Ratio (Not Specified Fake applications / Fake applications) = {not_specfake / fakecount:.6f}")

for column in df.columns:
    not_specified(column, column.upper())
    print('\n')
```

```
TITLE
-----------------

REAL
-----------
Number of Real applications that have not specified TITLE = 0
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.000000


FAKE
-----------
Number of Fake applications that have not specified TITLE = 0
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.000000


LOCATION
-----------------

REAL
-----------
Number of Real applications that have not specified LOCATION = 327
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.019219


FAKE
-----------
Number of Fake applications that have not specified LOCATION = 19
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.021940


DEPARTMENT
-----------------
```

```
REAL
-----------
Number of Real applications that have not specified DEPARTMENT = 11016
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.647467


FAKE
-----------
Number of Fake applications that have not specified DEPARTMENT = 531
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.613164


SALARY_RANGE
-----------------

REAL
-----------
Number of Real applications that have not specified SALARY_RANGE = 14369
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.844540


FAKE
-----------
Number of Fake applications that have not specified SALARY_RANGE = 643
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.742494


COMPANY_PROFILE
-----------------

REAL
-----------
Number of Real applications that have not specified COMPANY_PROFILE = 2721
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.159927


FAKE
-----------
Number of Fake applications that have not specified COMPANY_PROFILE = 587
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.677829


DESCRIPTION
-----------------

REAL
-----------
Number of Real applications that have not specified DESCRIPTION = 0
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.000000


FAKE
-----------
Number of Fake applications that have not specified DESCRIPTION = 1
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.001155


REQUIREMENTS
-----------------

REAL
-----------
Number of Real applications that have not specified REQUIREMENTS = 2541
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.149348


FAKE
-----------
Number of Fake applications that have not specified REQUIREMENTS = 154
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.177829


BENEFITS
-----------------

REAL
-----------
Number of Real applications that have not specified BENEFITS = 6846
```

```
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.402375


FAKE
-----------
Number of Fake applications that have not specified BENEFITS = 364
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.420323


TELECOMMUTING
-----------------

REAL
-----------
Number of Real applications that have not specified TELECOMMUTING = 0
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.000000


FAKE
-----------
Number of Fake applications that have not specified TELECOMMUTING = 0
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.000000


HAS_COMPANY_LOGO
-----------------

REAL
-----------
Number of Real applications that have not specified HAS_COMPANY_LOGO = 0
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.000000


FAKE
-----------
Number of Fake applications that have not specified HAS_COMPANY_LOGO = 0
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.000000


HAS_QUESTIONS
-----------------

REAL
-----------
Number of Real applications that have not specified HAS_QUESTIONS = 0
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.000000


FAKE
-----------
Number of Fake applications that have not specified HAS_QUESTIONS = 0
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.000000


EMPLOYMENT_TYPE
-----------------

REAL
-----------
Number of Real applications that have not specified EMPLOYMENT_TYPE = 3230
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.189844


FAKE
-----------
Number of Fake applications that have not specified EMPLOYMENT_TYPE = 241
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.278291


REQUIRED_EXPERIENCE
-----------------

REAL
-----------
Number of Real applications that have not specified REQUIRED_EXPERIENCE = 7671
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.450864
```

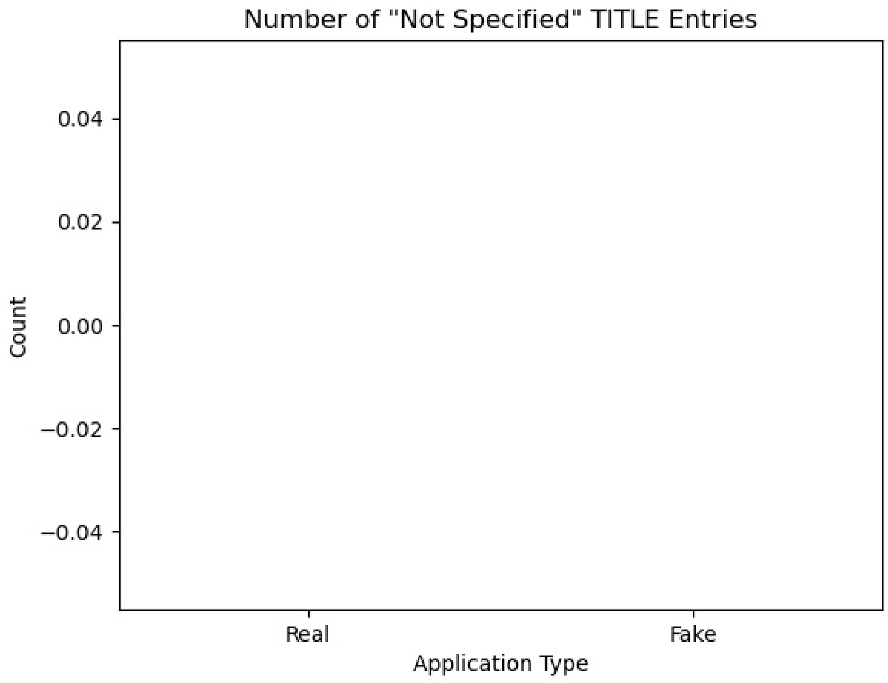
```
FAKE
-----------
Number of Fake applications that have not specified REQUIRED_EXPERIENCE = 495
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.571594


REQUIRED_EDUCATION
-----------------

REAL
-----------
Number of Real applications that have not specified REQUIRED_EDUCATION = 8990
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.528388


FAKE
-----------
Number of Fake applications that have not specified REQUIRED_EDUCATION = 512
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.591224


INDUSTRY
-----------------

REAL
-----------
Number of Real applications that have not specified INDUSTRY = 4628
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.272011


FAKE
-----------
Number of Fake applications that have not specified INDUSTRY = 275
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.317552


FUNCTION
-----------------

REAL
-----------
Number of Real applications that have not specified FUNCTION = 6118
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.359586


FAKE
-----------
Number of Fake applications that have not specified FUNCTION = 337
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.389145


FRAUDULENT
-----------------

REAL
-----------
Number of Real applications that have not specified FRAUDULENT = 0
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.000000


FAKE
-----------
Number of Fake applications that have not specified FRAUDULENT = 0
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.000000
```

In [23]:

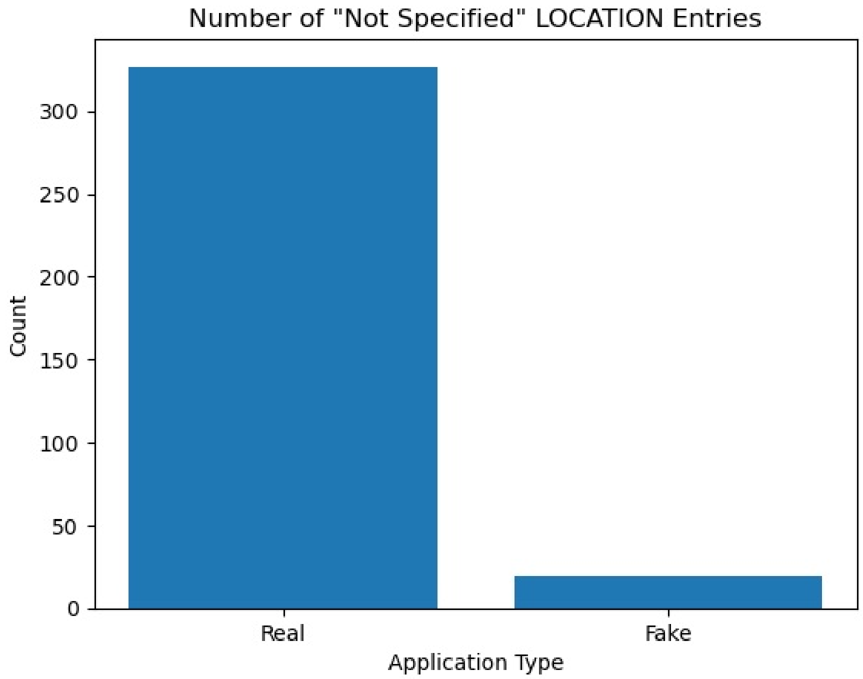
```python
import matplotlib.pyplot as plt

realcount = (df['fraudulent'] == 0).sum()  # Number of real applications
fakecount = (df['fraudulent'] == 1).sum()  # Number of fake applications

# FUNCTION TO CALCULATE THE NUMBER OF NOT SPECIFIED ENTRIES IN VARIOUS ATTRIBUTES ALONG WITH THE RATIO OF NOT S

def not_specified(labelname, name):
    df_real = df[df['fraudulent'] == 0][labelname]
    not_specreal = (df_real == 'Not Specified').sum()
    print(name + '\n-----------------\n\nREAL\n-----------')
    print(f"Number of Real applications that have not specified {name} = {not_specreal:.0f}")
    print(f"Number of Real applications = {realcount:.0f}")
```

```
    print(f"Ratio (Not Specified Real applications / Real applications) = {not_specreal / realcount:.6f}")

    df_fake = df[df['fraudulent'] == 1][labelname]
    not_specfake = (df_fake == 'Not Specified').sum()
    print('\n\nFAKE\n-----------')
    print(f"Number of Fake applications that have not specified {name} = {not_specfake:.0f}")
    print(f"Number of Fake applications = {fakecount:.0f}")
    print(f"Ratio (Not Specified Fake applications / Fake applications) = {not_specfake / fakecount:.6f}")

    # Create a bar chart to visualize the number of "Not Specified" entries for real and fake applications
    labels = ['Real', 'Fake']
    values = [not_specreal, not_specfake]

    plt.figure()
    plt.bar(labels, values)
    plt.title(f'Number of "Not Specified" {name} Entries')
    plt.xlabel('Application Type')
    plt.ylabel('Count')
    plt.show()

for column in df.columns:
    not_specified(column, column.upper())
    print('\n')
```

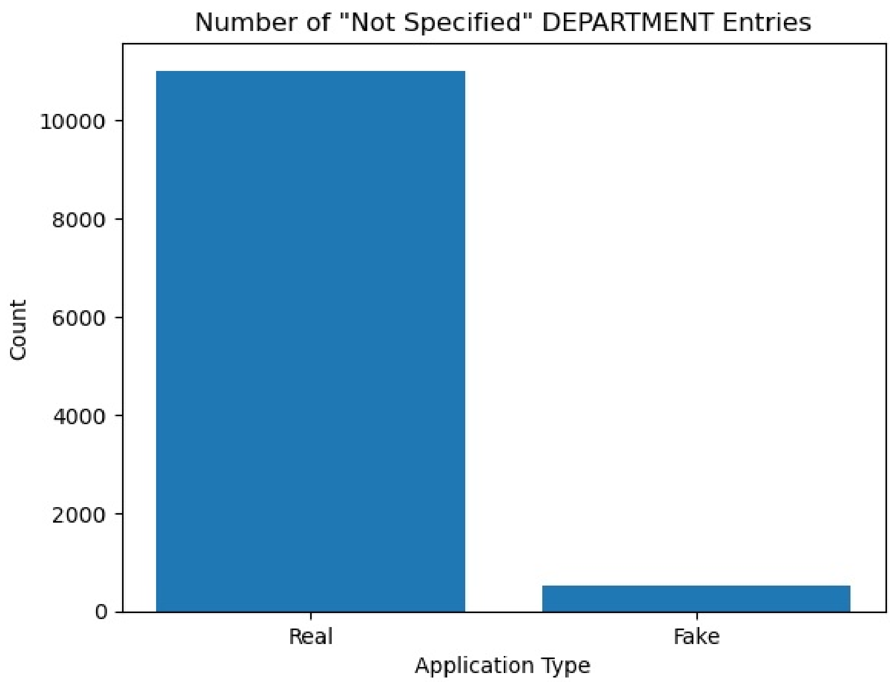
```
TITLE
-----------------

REAL
-----------
Number of Real applications that have not specified TITLE = 0
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.000000


FAKE
-----------
Number of Fake applications that have not specified TITLE = 0
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.000000
```



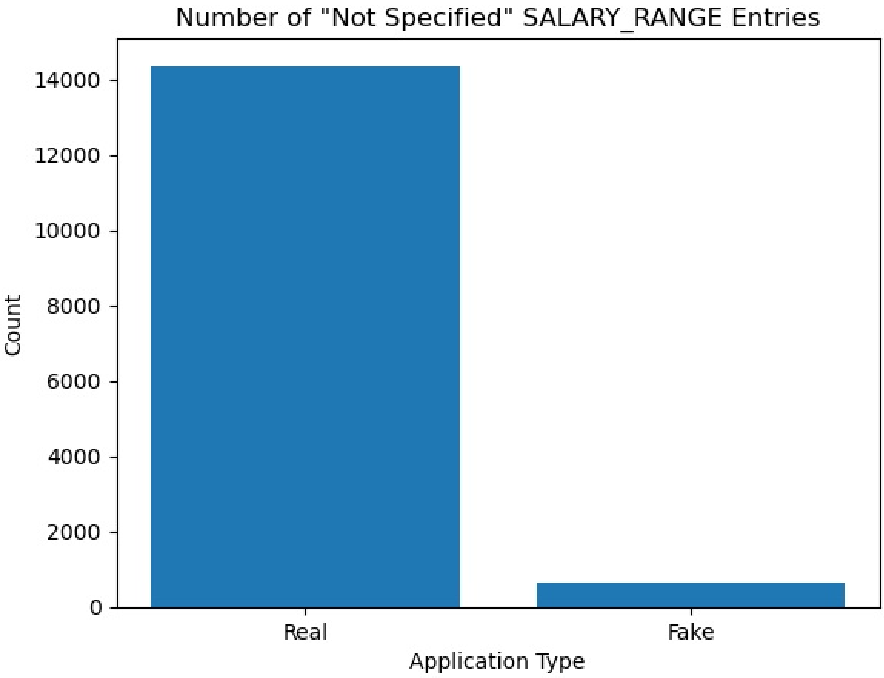

```
LOCATION
-----------------

REAL
-----------
Number of Real applications that have not specified LOCATION = 327
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.019219


FAKE
-----------
Number of Fake applications that have not specified LOCATION = 19
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.021940
```



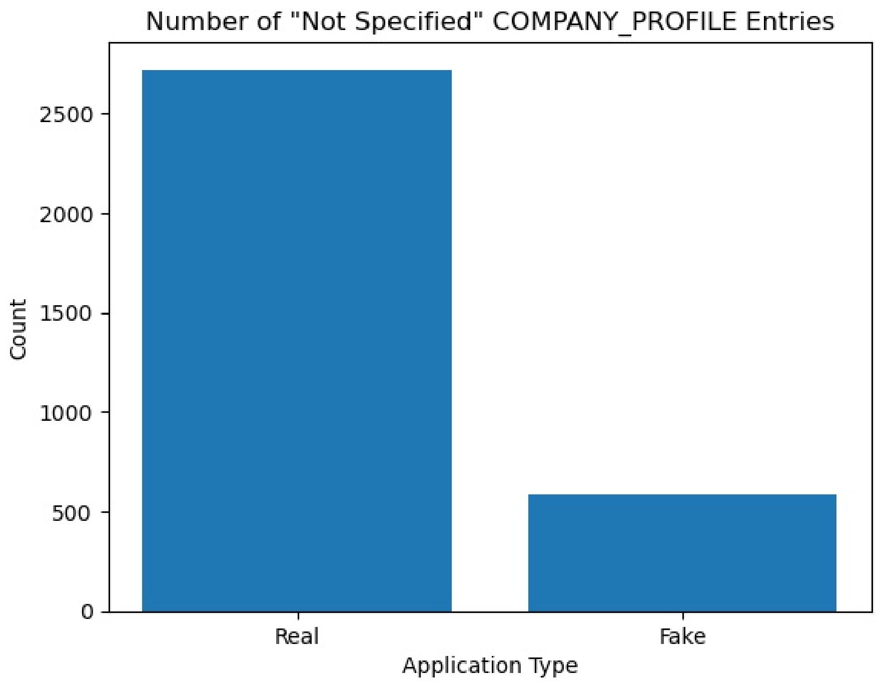

```
DEPARTMENT
-----------------

REAL
-----------
Number of Real applications that have not specified DEPARTMENT = 11016
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.647467


FAKE
-----------
Number of Fake applications that have not specified DEPARTMENT = 531
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.613164
```



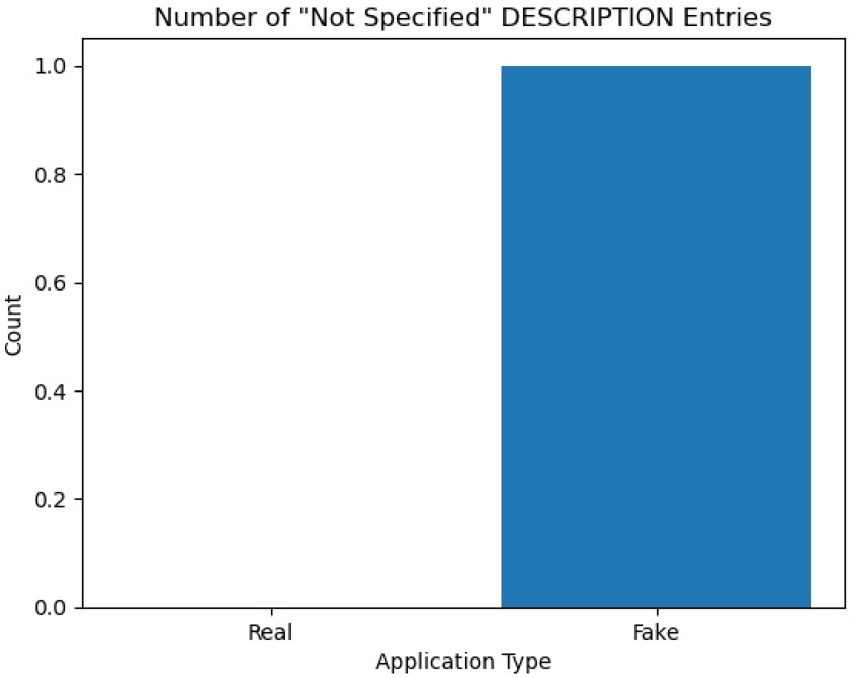
```
SALARY_RANGE
-----------------

REAL
-----------
Number of Real applications that have not specified SALARY_RANGE = 14369
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.844540


FAKE
-----------
Number of Fake applications that have not specified SALARY_RANGE = 643
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.742494
```



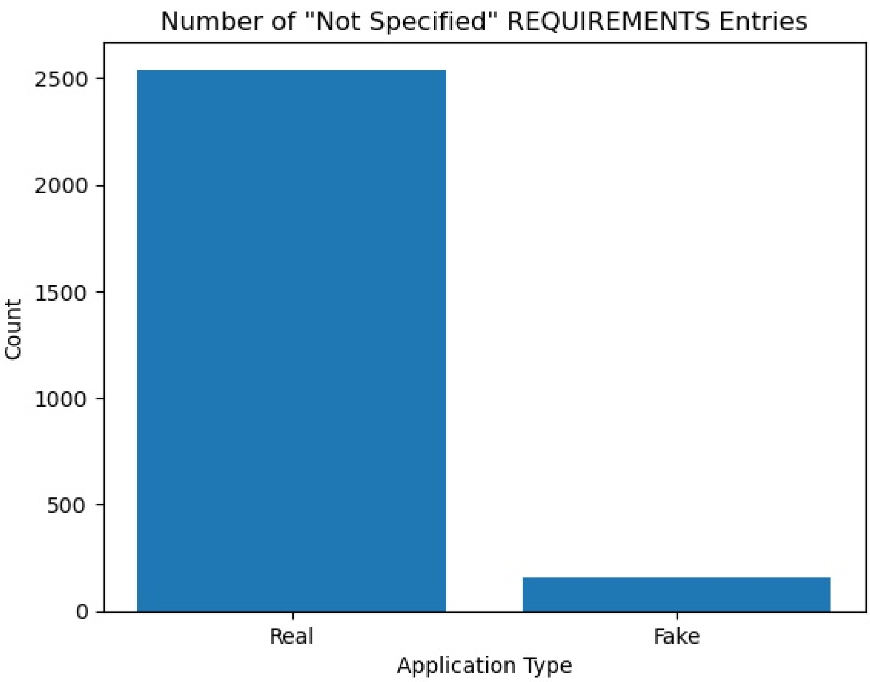

```
COMPANY_PROFILE
-----------------

REAL
-----------
Number of Real applications that have not specified COMPANY_PROFILE = 2721
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.159927


FAKE
-----------
Number of Fake applications that have not specified COMPANY_PROFILE = 587
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.677829
```



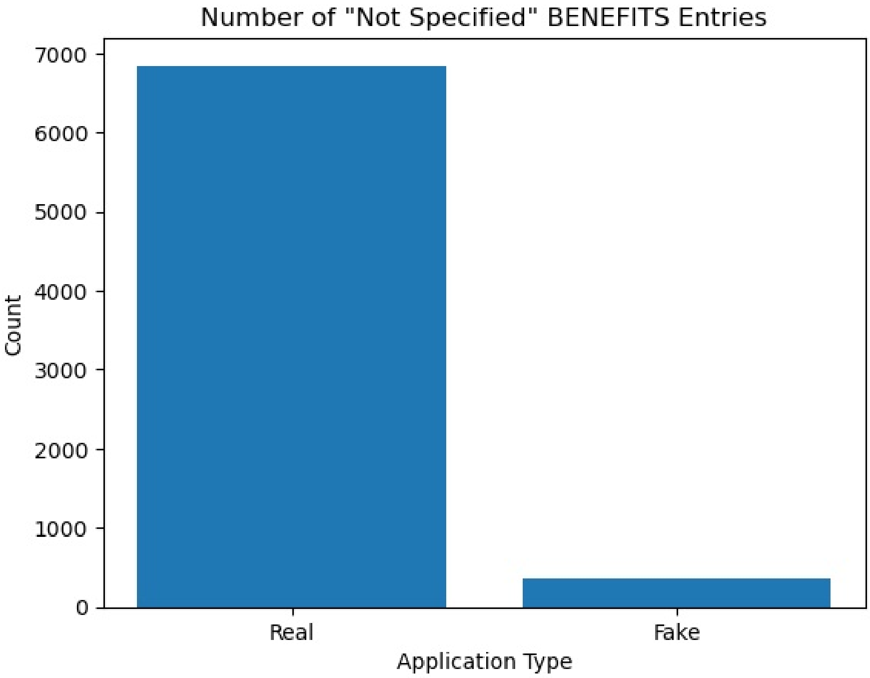

```
DESCRIPTION
-----------------

REAL
-----------
Number of Real applications that have not specified DESCRIPTION = 0
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.000000


FAKE
-----------
Number of Fake applications that have not specified DESCRIPTION = 1
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.001155
```



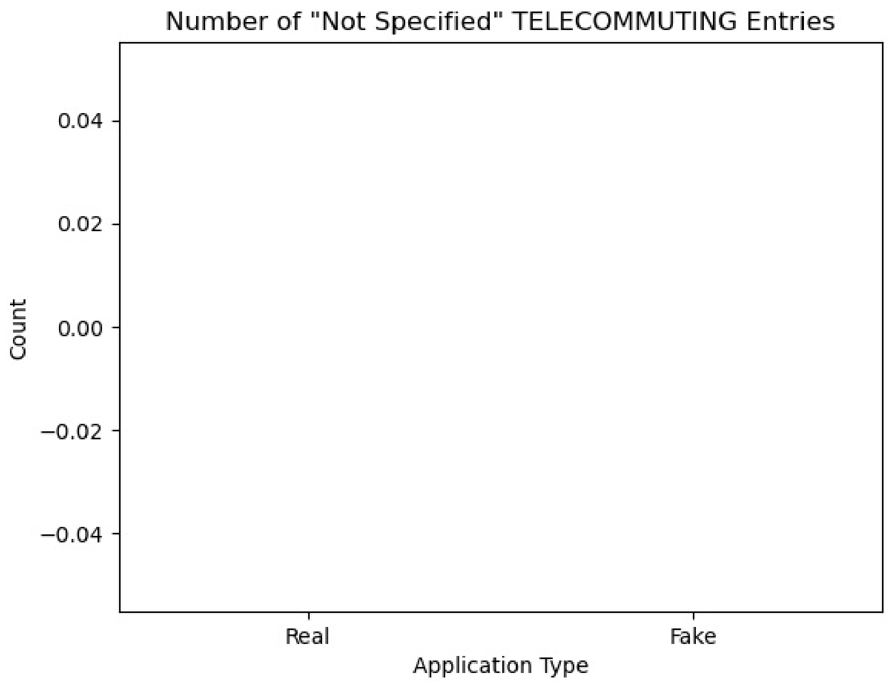

```
REQUIREMENTS
-----------------

REAL
-----------
Number of Real applications that have not specified REQUIREMENTS = 2541
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.149348


FAKE
-----------
Number of Fake applications that have not specified REQUIREMENTS = 154
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.177829
```



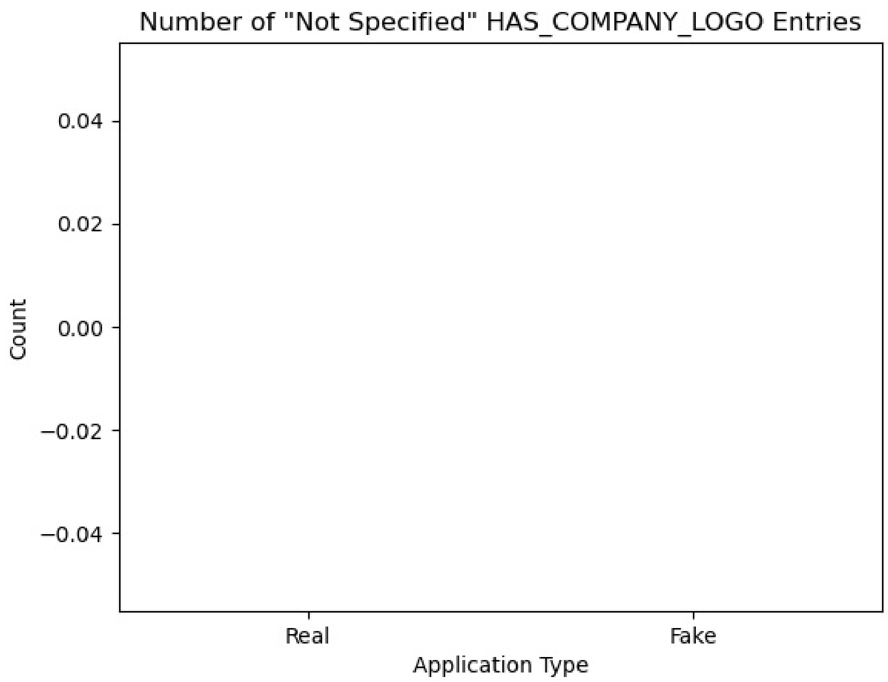

```
BENEFITS
-----------------

REAL
-----------
Number of Real applications that have not specified BENEFITS = 6846
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.402375


FAKE
-----------
Number of Fake applications that have not specified BENEFITS = 364
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.420323
```



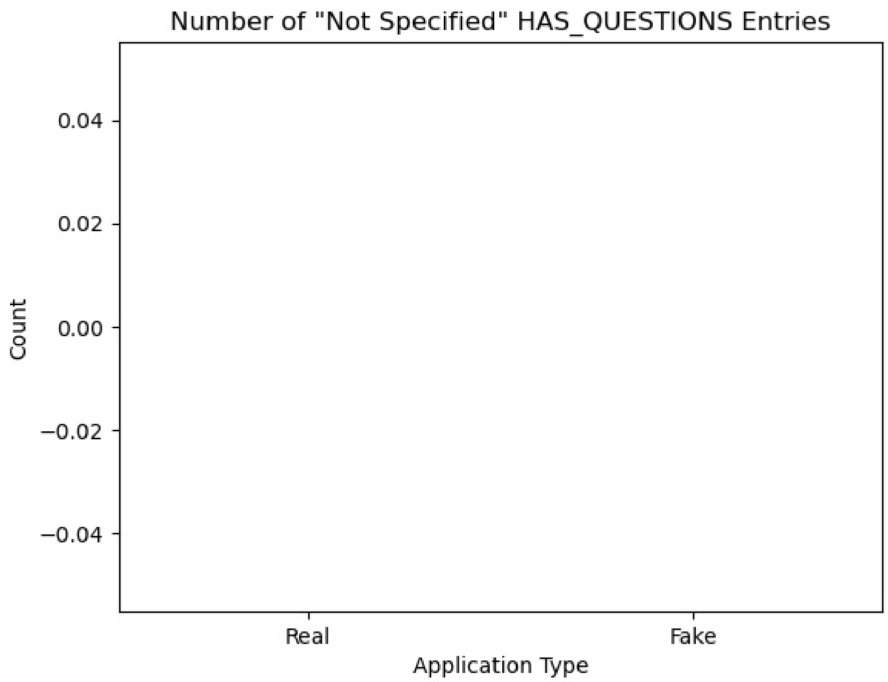

```
TELECOMMUTING
-----------------

REAL
-----------
Number of Real applications that have not specified TELECOMMUTING = 0
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.000000


FAKE
-----------
Number of Fake applications that have not specified TELECOMMUTING = 0
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.000000
```



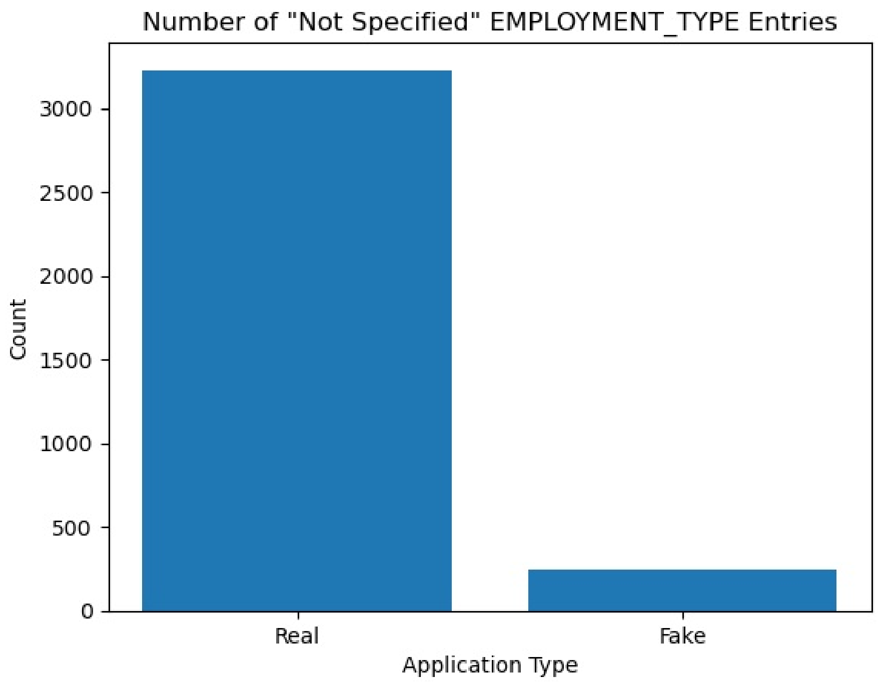

```
HAS_COMPANY_LOGO
-----------------

REAL
-----------
Number of Real applications that have not specified HAS_COMPANY_LOGO = 0
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.000000


FAKE
-----------
Number of Fake applications that have not specified HAS_COMPANY_LOGO = 0
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.000000
```



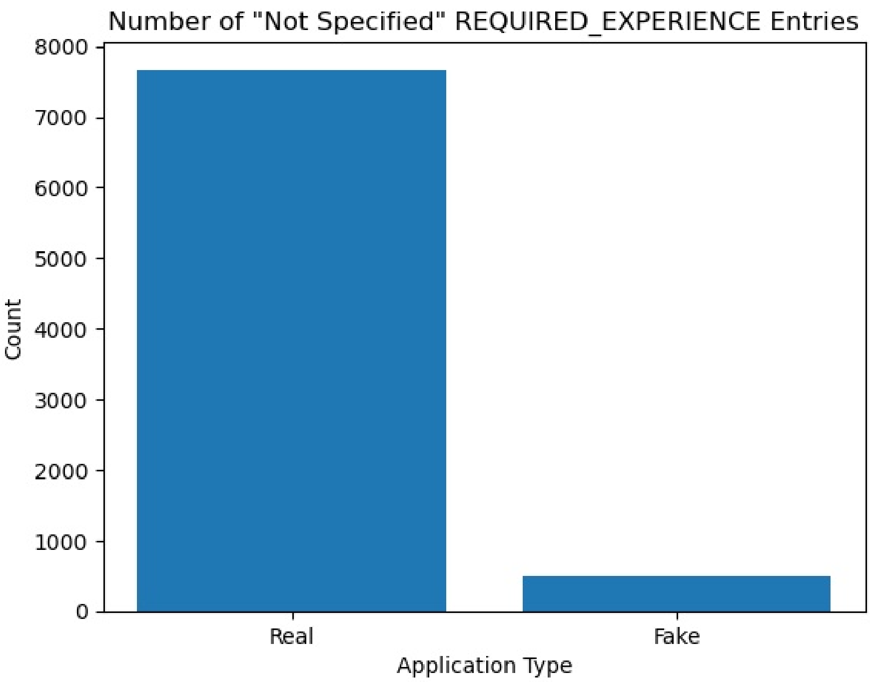

```
HAS_QUESTIONS
-----------------

REAL
-----------
Number of Real applications that have not specified HAS_QUESTIONS = 0
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.000000


FAKE
-----------
Number of Fake applications that have not specified HAS_QUESTIONS = 0
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.000000
```



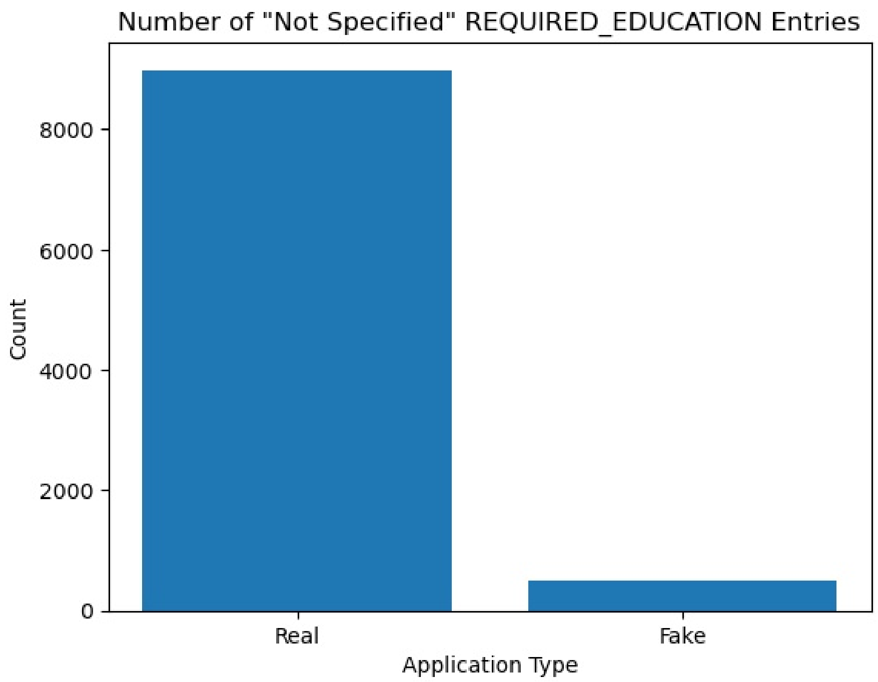

```
EMPLOYMENT_TYPE
-----------------

REAL
-----------
Number of Real applications that have not specified EMPLOYMENT_TYPE = 3230
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.189844


FAKE
-----------
Number of Fake applications that have not specified EMPLOYMENT_TYPE = 241
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.278291
```



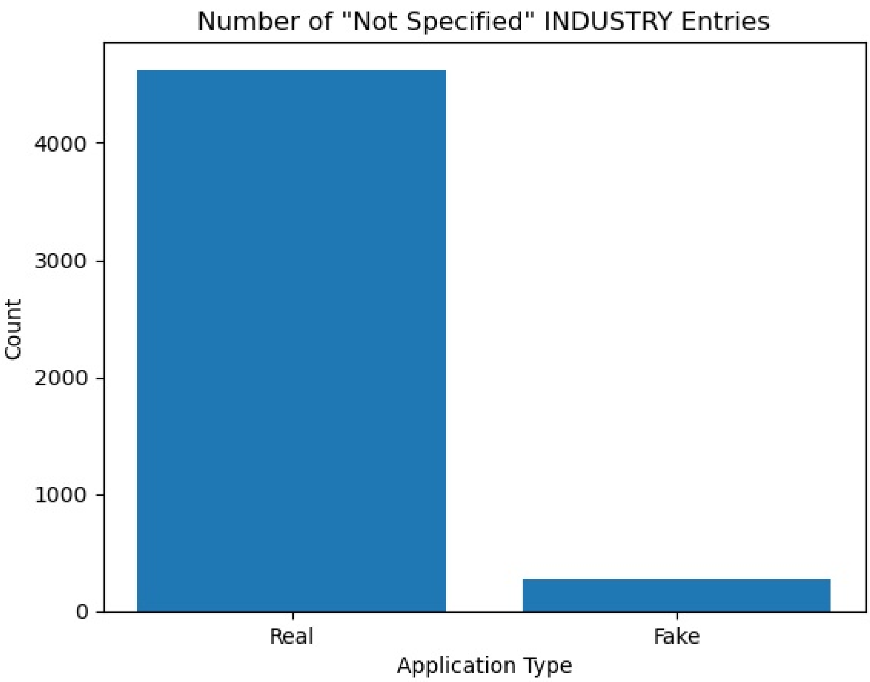

```
REQUIRED_EXPERIENCE
-----------------

REAL
-----------
Number of Real applications that have not specified REQUIRED_EXPERIENCE = 7671
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.450864


FAKE
-----------
Number of Fake applications that have not specified REQUIRED_EXPERIENCE = 495
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.571594
```



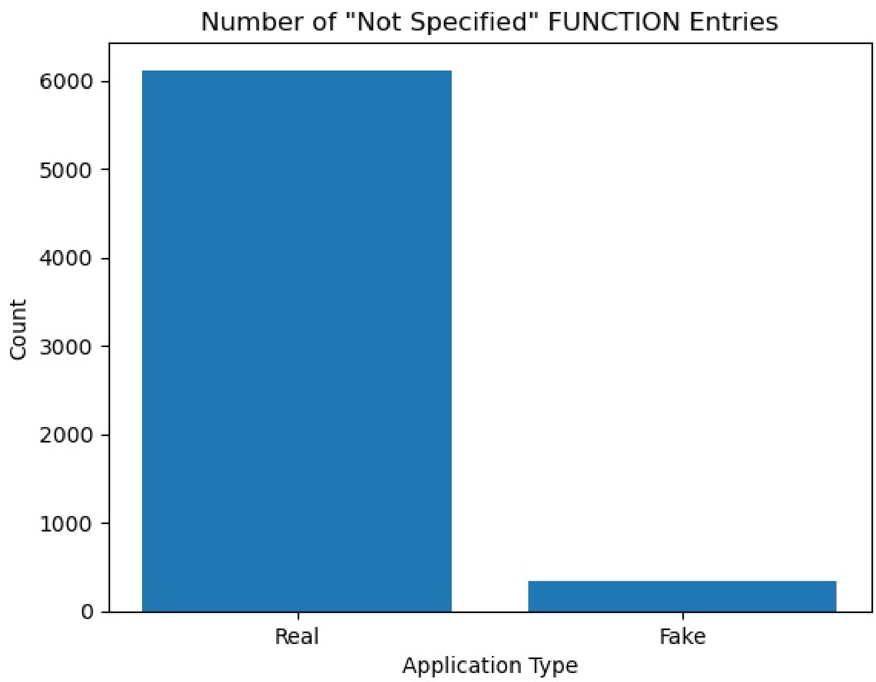

```
REQUIRED_EDUCATION
-----------------

REAL
-----------
Number of Real applications that have not specified REQUIRED_EDUCATION = 8990
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.528388


FAKE
-----------
Number of Fake applications that have not specified REQUIRED_EDUCATION = 512
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.591224
```



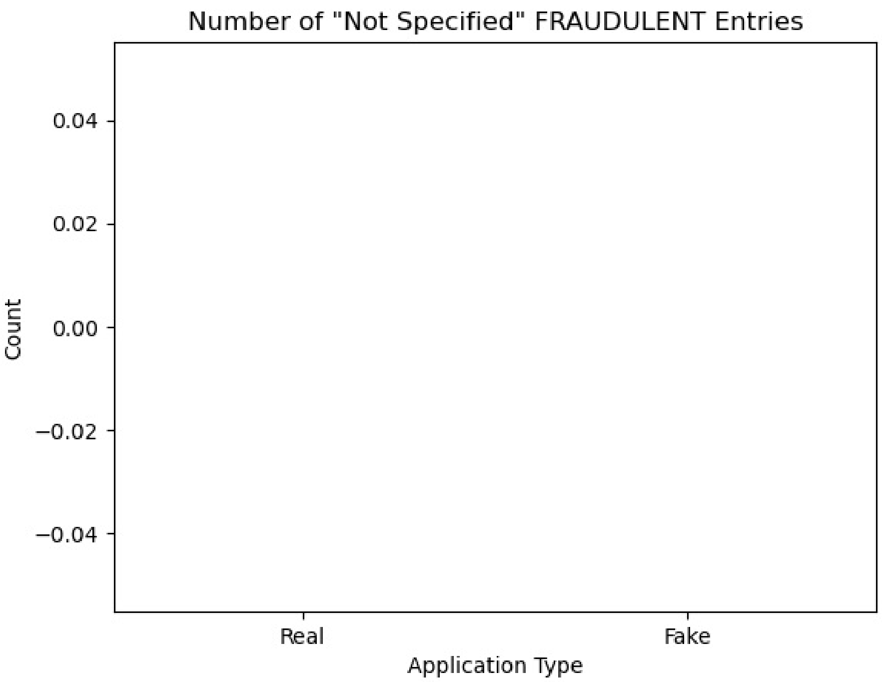

```
INDUSTRY
-----------------

REAL
-----------
Number of Real applications that have not specified INDUSTRY = 4628
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.272011


FAKE
-----------
Number of Fake applications that have not specified INDUSTRY = 275
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.317552
```




```
FUNCTION
-----------------

REAL
-----------
Number of Real applications that have not specified FUNCTION = 6118
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.359586


FAKE
-----------
Number of Fake applications that have not specified FUNCTION = 337
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.389145
```




```
FRAUDULENT
-----------------

REAL
-----------
Number of Real applications that have not specified FRAUDULENT = 0
Number of Real applications = 17014
Ratio (Not Specified Real applications / Real applications) = 0.000000


FAKE
-----------
Number of Fake applications that have not specified FRAUDULENT = 0
Number of Fake applications = 866
Ratio (Not Specified Fake applications / Fake applications) = 0.000000
```



In [ ]:

```
#FUNCTION TO RETURN THE 20 MOST FREQUENTLY OCCURING WORDS IN REAL/FAKE APPLCATIONS GIVEN THE ATTRIBUTE

#GIVEN WHETHER AN APPLICATION IS REAL OR FAKE, THE PROBABILITY OF THE WORD APPEARING IN THAT CATEGORY IS DIPLAY

def frequent(lab,key):

    list_of_words = []
    if key == "real":
        f=0
        count = realcount
    else:
        f=1
        count = fakecount

    for i in (df1[lab].loc[df1['fraudulent']==f]):
        list_of_words.append((' '.join(dict.fromkeys(i.split()))))

    rand = ' '.join(list_of_words)
    listx = list(rand.split(" "))
    ratiolist = list(pd.Series(listx).value_counts()/count)
    _count = pd.DataFrame(pd.Series(listx).value_counts())
    _count. rename(columns = {_count.columns[0]:'Count'}, inplace = True)
    _count['Probability'] = ratiolist
    print("Frequently appearing words in " + lab + " of " + key + " applications")
    print(_count.head(20))
    list_of_words.clear()
```

In [24]:

```
def frequent(lab, key):
    list_of_words = []
```

```
    if key == "real":
        f = 0
        count = realcount
    else:
        f = 1
        count = fakecount

    for i in df[lab].loc[df['fraudulent'] == f]:
        list_of_words.append((' '.join(dict.fromkeys(i.split()))))

    rand = ' '.join(list_of_words)
    listx = list(rand.split(" "))
    ratiolist = list(pd.Series(listx).value_counts() / count)
    _count = pd.DataFrame(pd.Series(listx).value_counts())
    _count.rename(columns={_count.columns[0]: 'Count'}, inplace=True)
    _count['Probability'] = ratiolist

    print("Frequently appearing words in " + lab + " of " + key + " applications")
    print(_count.head(20))

    list_of_words.clear()
```

In [25]:
```
frequent('location','real')
frequent('location','fake')
```

```
Frequently appearing words in location of real applications
           Count  Probability
US,         9868     0.579993
GB,         2353     0.138298
CA,         2351     0.138180
,           2085     0.122546
NY,         1191     0.070001
London      1105     0.064947
LND,         986     0.057952
GR,          937     0.055072
San          829     0.048725
TX,          823     0.048372
New          807     0.047432
York         766     0.045022
I,           688     0.040437
Athens       568     0.033384
Francisco    498     0.029270
IL,          477     0.028036
DE,          398     0.023393
FL,          385     0.022628
IN,          380     0.022335
OH,          354     0.020806
Frequently appearing words in location of fake applications
             Count  Probability
US,            725     0.837182
CA,            155     0.178984
TX,            152     0.175520
Houston         92     0.106236
NY,             68     0.078522
,               57     0.065820
San             57     0.065820
AU,             40     0.046189
MD,             35     0.040416
NSW,            32     0.036952
Sydney          31     0.035797
FL,             30     0.034642
Bakersfield     24     0.027714
Mateo           24     0.027714
Los             23     0.026559
Angeles         23     0.026559
New             23     0.026559
York            22     0.025404
GB,             21     0.024249
GA,             20     0.023095
```

encodng

In [27]:
```
en = preprocessing.LabelEncoder()
#ASSIGNS NUMBER TO EVERY LABEL
for i in df.columns:
    en.fit(df[i])
    df[i]=en.transform(df[i])
```

In [28]:
```
df.head(25)
```

Out[28]:

	title	location	department	salary_range	company_profile	description	requirements	benefits	telecommuting	has_company_logo	has_
0	6043	2536	758	872	1548	4038	3684	3038	0	1	
1	2183	1073	1162	872	15	6855	10491	5350	0	1	
2	1763	1868	831	872	1393	7017	4514	3038	0	1	
3	299	1704	1055	872	946	9211	3077	3174	0	1	
4	975	1742	831	872	1182	5258	6540	2114	0	1	
5	375	2085	831	872	896	5417	5852	3038	0	0	
6	4296	216	50	296	522	14294	11219	5688	0	1	
7	5550	1565	831	872	90	13907	3583	1402	0	1	
8	4201	1773	831	872	1169	4973	5076	3038	0	1	
9	2210	1384	831	872	899	9642	5472	3038	0	1	
10	244	2401	831	68	896	7577	6137	843	0	0	
11	10118	625	567	872	1426	11179	10410	5636	0	1	
12	654	1689	831	872	899	9401	7663	3038	0	1	
13	4881	1764	831	872	582	3640	9994	3038	0	1	
14	298	38	1055	872	79	1138	11464	2649	0	1	
15	10668	1218	1055	126	708	644	4874	781	0	1	
16	4263	898	5	872	149	12034	6297	3038	0	1	
17	9663	724	831	872	449	4372	365	1136	0	1	
18	10762	2536	831	872	719	5746	5852	3038	0	1	
19	7336	2752	831	872	1440	3776	5546	3038	0	0	
20	5999	2835	831	872	678	5104	4704	3038	0	1	
21	3716	1083	831	872	546	11677	10999	5647	0	1	
22	3164	0	430	872	1385	10446	7895	3776	0	1	
23	10725	1429	186	68	1413	11366	4701	1078	0	1	
24	2179	625	831	872	896	12281	5852	3038	0	0	

Model

In [29]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
import time
```

In [30]:
```
x=df.drop(['fraudulent'],axis=1)
y=df["fraudulent"]
x_train,x_test,y_train,y_test = (train_test_split(x, y, test_size=0.25, shuffle=True))
```

In [31]:
```
def traintest(model,modelname):

    start = time.time()
    print("\n------------------\nMODEL - "+ modelname + "\n----------------\n")

    #Training the model
    model.fit(x_train, y_train)

    #Predicting
    y_pred = model.predict(x_test)

    #Calculating the accuracy
    accuracy = metrics.accuracy_score(y_test, y_pred)
    print("Accuracy = " + '{:.2f}%'.format(accuracy*100))
    #Calculating the precision
    precision = metrics.precision_score(y_test, y_pred)
    print("Precision = " + '{:.2f}%'.format(precision*100))

    #Total Time
    end = time.time() - start
    print("Time = " + '{:.2f}s'.format(end))
```

In [39]:
```
#ACCURACY ALONG WITH THE TIME IS NOTED

import warnings
```

```python
warnings.filterwarnings('ignore')

traintest(GaussianNB(),"NAIVE BAYES")
traintest(DecisionTreeClassifier(),"DECISION TREE")
traintest(RandomForestClassifier(),"RANDOM FOREST")
traintest(KNeighborsClassifier(),"KNN")
traintest(SVC(),"SVM")
traintest(LogisticRegression(solver='liblinear'),"LOGISITC REGRESSION")
```

```
------------------
MODEL - NAIVE BAYES
-----------------

Accuracy = 93.47%
Precision = 29.83%
Time = 0.03s

------------------
MODEL - DECISION TREE
-----------------

Accuracy = 96.89%
Precision = 68.18%
Time = 0.17s

------------------
MODEL - RANDOM FOREST
-----------------

Accuracy = 98.05%
Precision = 95.83%
Time = 2.85s

------------------
MODEL - KNN
-----------------

Accuracy = 95.37%
Precision = 55.36%
Time = 5.38s

------------------
MODEL - SVM
-----------------

Accuracy = 95.10%
Precision = 0.00%
Time = 4.88s

------------------
MODEL - LOGISITC REGRESSION
-----------------

Accuracy = 95.23%
Precision = 80.00%
Time = 0.38s
```

Final Model

In [42]:

```python
traintest(RandomForestClassifier(),"RANDOM FOREST")
```

```
------------------
MODEL - RANDOM FOREST
-----------------

Accuracy = 98.08%
Precision = 95.86%
Time = 1.91s
```