In [ ]:
```
#Trevor Zeiger
#DSC - 680
#Week3 Milestone 3
```

In [ ]:
```
# Remote Work Salary Analysis - Data Cleaning and Combining Script (with Detailed Explanations)
```

In [2]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
```

In [3]:
```
warnings.filterwarnings('ignore')  # Suppress warnings for cleaner output
```

In [4]:
```
# Load all datasets

# These are the three datasets used in this analysis:
# - ds_salaries.csv: Global dataset containing remote work salary data across industries and countries
# - eda_data.csv: A U.S.-focused dataset derived from Glassdoor job postings
# - Salary_Dataset_with_Extra_Features.csv: An India-based dataset, largely focused on Android developer roles

# In a real implementation, replace the file paths with your own local or cloud-based file locations.

ds_salaries = pd.read_csv("ds_salaries.csv")
eda_data = pd.read_csv("eda_data.csv")
salary_dataset = pd.read_csv("Salary_Dataset_with_Extra_Features.csv")
```

In [8]:
```
# Display the first few rows of each dataset to understand their structure
ds_salaries_head = ds_salaries.head()
eda_data_head = eda_data.head()
salary_dataset_head = salary_dataset.head()

ds_salaries_head, eda_data_head, salary_dataset_head
```

Out[8]:
```
(   Unnamed: 0  work_year experience_level employment_type  \
 0           0       2020               MI              FT
 1           1       2020               SE              FT
 2           2       2020               SE              FT
 3           3       2020               MI              FT
 4           4       2020               SE              FT

                    job_title  salary salary_currency  salary_in_usd  \
 0             Data Scientist   70000             EUR          79833
 1  Machine Learning Scientist  260000             USD         260000
 2          Big Data Engineer   85000             GBP         109024
 3       Product Data Analyst   20000             USD          20000
 4   Machine Learning Engineer  150000             USD         150000

   employee_residence  remote_ratio company_location company_size
 0                 DE             0               DE            L
 1                 JP             0               JP            S
 2                 GB            50               GB            M
 3                 HN             0               HN            S
 4                 US            50               US            L  ,
    Unnamed: 0                Job Title             Salary Estimate  \
 0           0             Data Scientist   $53K-$91K (Glassdoor est.)
 1           1  Healthcare Data Scientist  $63K-$112K (Glassdoor est.)
 2           2             Data Scientist   $80K-$90K (Glassdoor est.)
 3           3             Data Scientist   $56K-$97K (Glassdoor est.)
 4           4             Data Scientist  $86K-$143K (Glassdoor est.)

                                    Job Description  Rating  \
 0  Data Scientist\nLocation: Albuquerque, NM\nEdu...     3.8
 1  What You Will Do:\n\nI. General Summary\n\nThe...     3.4
 2  KnowBe4, Inc. is a high growth information sec...     4.8
 3  *Organization and Job ID**\nJob ID: 310709\n\n...     3.8
 4  Data Scientist\nAffinity Solutions / Marketing...     2.9

                               Company Name         Location  \
 0                     Tecolote Research\n3.8  Albuquerque, NM
 1  University of Maryland Medical System\n3.4    Linthicum, MD
 2                               KnowBe4\n4.8   Clearwater, FL
 3                                  PNNL\n3.8     Richland, WA
 4                    Affinity Solutions\n2.9     New York, NY

      Headquarters                    Size  Founded  ... age python_yn R_yn  \
 0      Goleta, CA   501 to 1000 employees     1973  ...  47         1    0
 1   Baltimore, MD        10000+ employees     1984  ...  36         1    0
 2  Clearwater, FL   501 to 1000 employees     2010  ...  10         1    0
 3    Richland, WA  1001 to 5000 employees     1965  ...  55         1    0
 4    New York, NY     51 to 200 employees     1998  ...  22         1    0

   spark aws  excel        job_simp  seniority  desc_len  num_comp
 0     0   0      1  data scientist         na      2536         0
 1     0   0      0  data scientist         na      4783         0
 2     1   0      1  data scientist         na      3461         0
 3     0   0      0  data scientist         na      3883         3
 4     0   0      1  data scientist         na      2728         3

 [5 rows x 33 columns],
    Rating                     Company Name          Job Title   Salary  \
 0     3.8                           Sasken  Android Developer   400000
 1     4.5  Advanced Millennium Technologies  Android Developer   400000
 2     4.0                       Unacademy  Android Developer  1000000
 3     3.8                SnapBizz Cloudtech  Android Developer   300000
 4     4.4            Appoids Tech Solutions  Android Developer   600000

    Salaries Reported   Location Employment Status Job Roles
 0                  3  Bangalore         Full Time   Android
 1                  3  Bangalore         Full Time   Android
 2                  3  Bangalore         Full Time   Android
 3                  3  Bangalore         Full Time   Android
 4                  3  Bangalore         Full Time   Android  )
```

In [10]:
```python
# Define a consistent column structure

# To combine different datasets effectively, we define a set of standard column names
# that will be used across all datasets, even if some values are missing or estimated.
standard_columns = [
    'Job Title',              # Name or type of the position (e.g., Data Scientist)
    'Location',               # Geographic location or company base
    'Employment Type',        # Full-time, part-time, contract, etc.
    'Experience Level',       # Entry, mid, senior, executive (may be missing in some datasets)
    'Salary (USD)',           # Salary converted to USD where possible
```

```
    'Salary (INR)',           # Salaries specific to India-based roles (in Indian Rupees)
    'Salary Estimate',        # Text-based salary range estimates from sources like Glassdoor
    'Source'                  # Indicates the dataset origin: Global, Glassdoor, or India
]
```

In [12]:
```
# Clean and reformat each dataset

# Global dataset (ds_salaries.csv)
# This dataset already includes structured salary data and location info.
ds_clean = pd.DataFrame(columns=standard_columns)
ds_clean['Job Title'] = ds_salaries['job_title']
ds_clean['Location'] = ds_salaries['company_location']
ds_clean['Employment Type'] = ds_salaries['employment_type']
ds_clean['Experience Level'] = ds_salaries['experience_level']
ds_clean['Salary (USD)'] = ds_salaries['salary_in_usd']
ds_clean['Source'] = 'Global'
```

In [14]:
```
# Glassdoor dataset (eda_data.csv)

# This dataset is rich in job descriptions and salary estimates but lacks structured salary values.
eda_clean = pd.DataFrame(columns=standard_columns)
eda_clean['Job Title'] = eda_data['job_simp']                  # Simplified job titles
eda_clean['Location'] = eda_data['Location']                  # U.S. city/state information
eda_clean['Salary Estimate'] = eda_data['Salary Estimate']    # Salary range text (e.g., "$80K-$120K")
eda_clean['Source'] = 'Glassdoor'
```

In [16]:
```
# Indeed India dataset (Salary_Dataset_with_Extra_Features.csv)

# This is a more localized dataset with salary data mostly in INR for Indian roles.
salary_clean = pd.DataFrame(columns=standard_columns)
salary_clean['Job Title'] = salary_dataset['Job Title']
salary_clean['Location'] = salary_dataset['Location']
salary_clean['Salary (INR)'] = salary_dataset['Salary']
salary_clean['Employment Type'] = salary_dataset['Employment Status']
salary_clean['Source'] = 'India'
```

In [18]:
```
# Combine all cleaned datasets

# We concatenate the three datasets into a single DataFrame for unified analysis.
# Missing values are expected for some columns depending on the source.
combined_df = pd.concat([ds_clean, eda_clean, salary_clean], ignore_index=True)
```

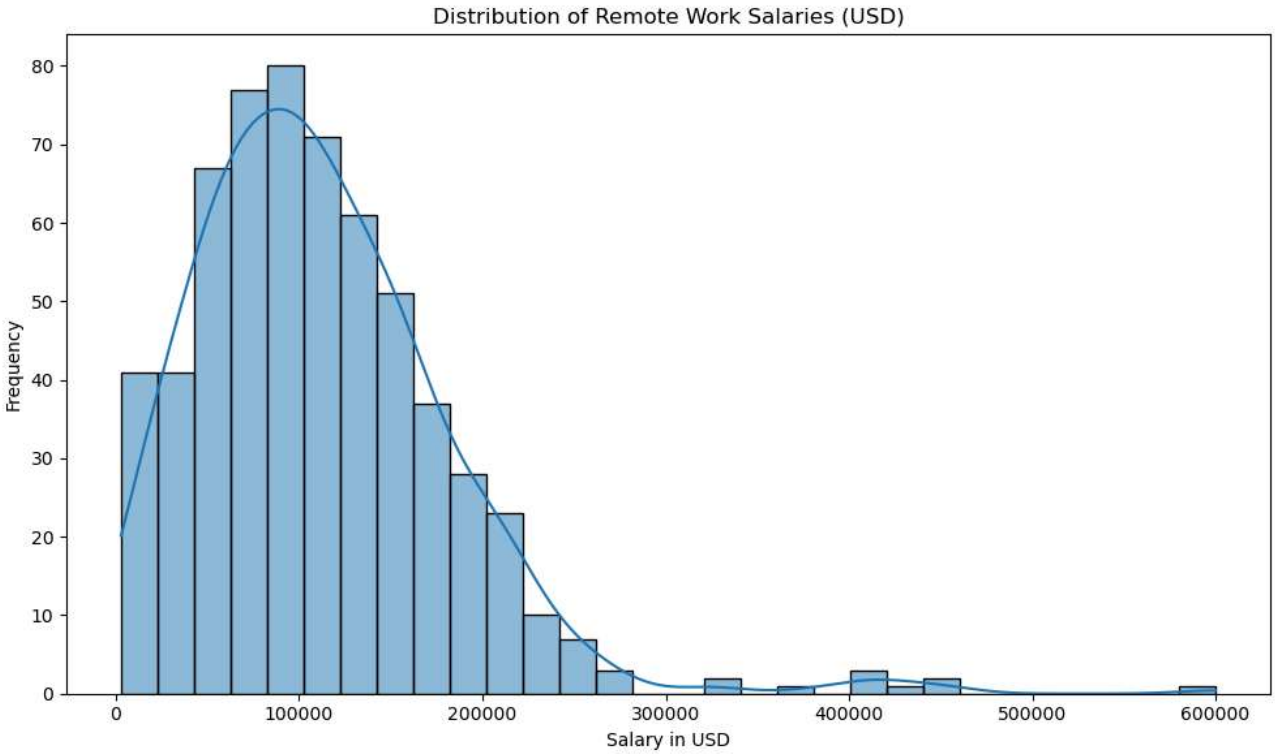
In [20]:
```
# Preview the combined dataset

# This output allows us to confirm the structure and integrity of the merged dataset.
print(combined_df.head())
```

```
                   Job Title Location Employment Type Experience Level  \
0             Data Scientist       DE              FT               MI
1  Machine Learning Scientist       JP              FT               SE
2          Big Data Engineer       GB              FT               SE
3       Product Data Analyst       HN              FT               MI
4   Machine Learning Engineer       US              FT               SE

  Salary (USD) Salary (INR) Salary Estimate  Source
0        79833          NaN             NaN  Global
1       260000          NaN             NaN  Global
2       109024          NaN             NaN  Global
3        20000          NaN             NaN  Global
4       150000          NaN             NaN  Global
```

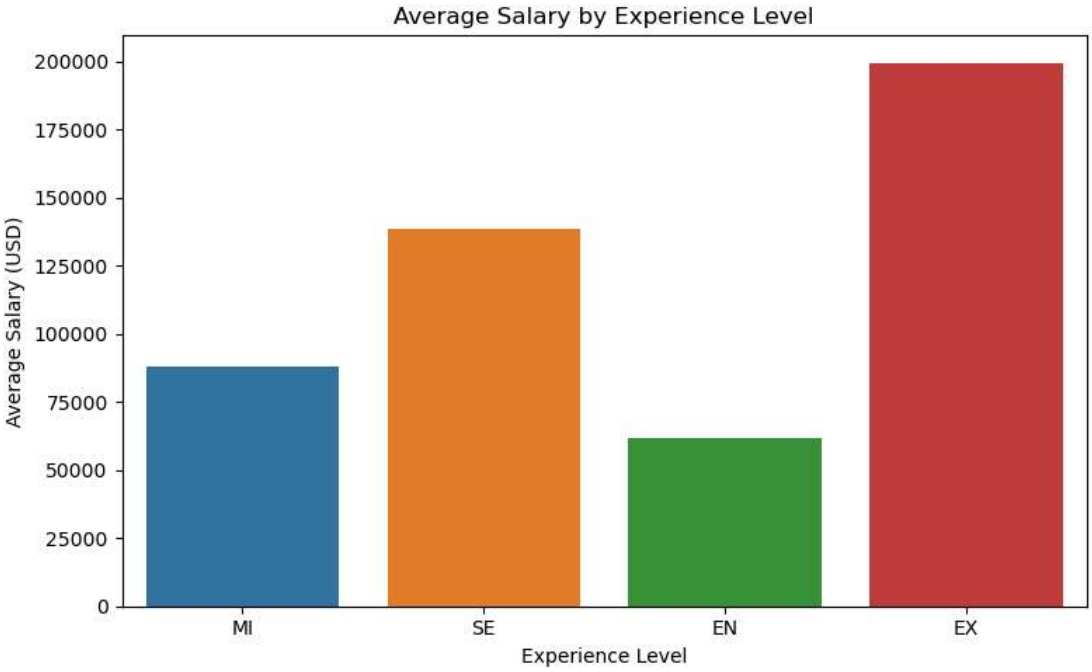
In [22]:
```
# Optional - Save the cleaned dataset

# This will create a CSV file for future analysis or visualization steps.
combined_df.to_csv("combined_remote_work_salary_data.csv", index=False)
```

In [24]:
```
# Filter out null salaries in USD for plotting
usd_data = combined_df[combined_df['Salary (USD)'].notnull()]
```

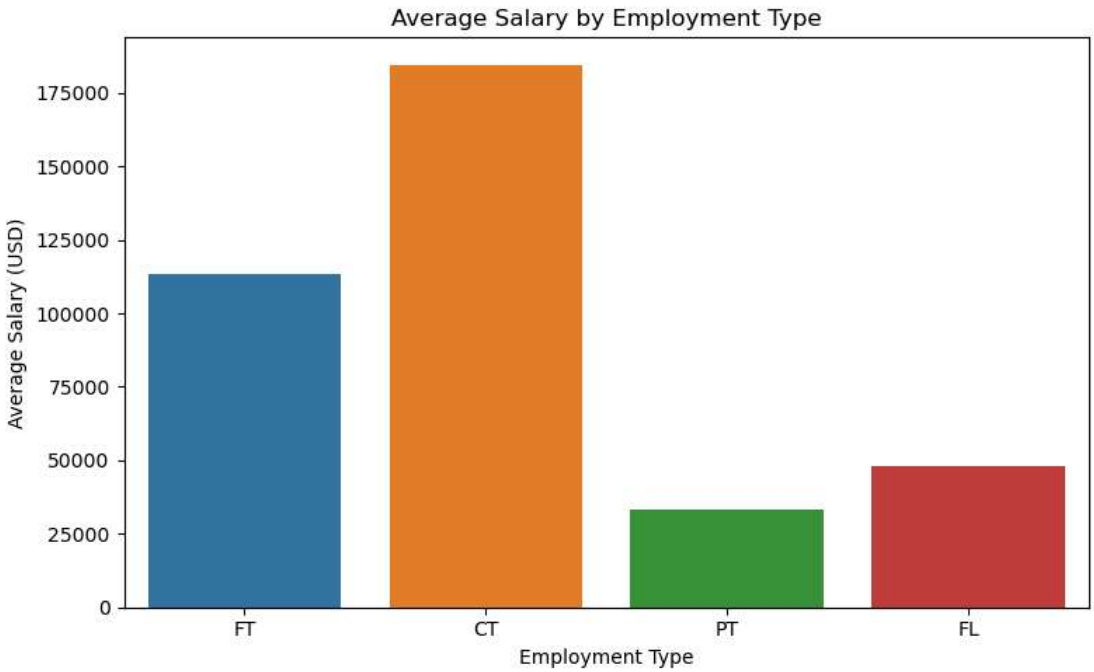
In [26]:
```
# Visualization 1: Distribution of salaries (USD)
plt.figure(figsize=(10, 6))
sns.histplot(usd_data['Salary (USD)'], bins=30, kde=True)
plt.title('Distribution of Remote Work Salaries (USD)')
plt.xlabel('Salary in USD')
plt.ylabel('Frequency')
plt.tight_layout()
plt.show()
```



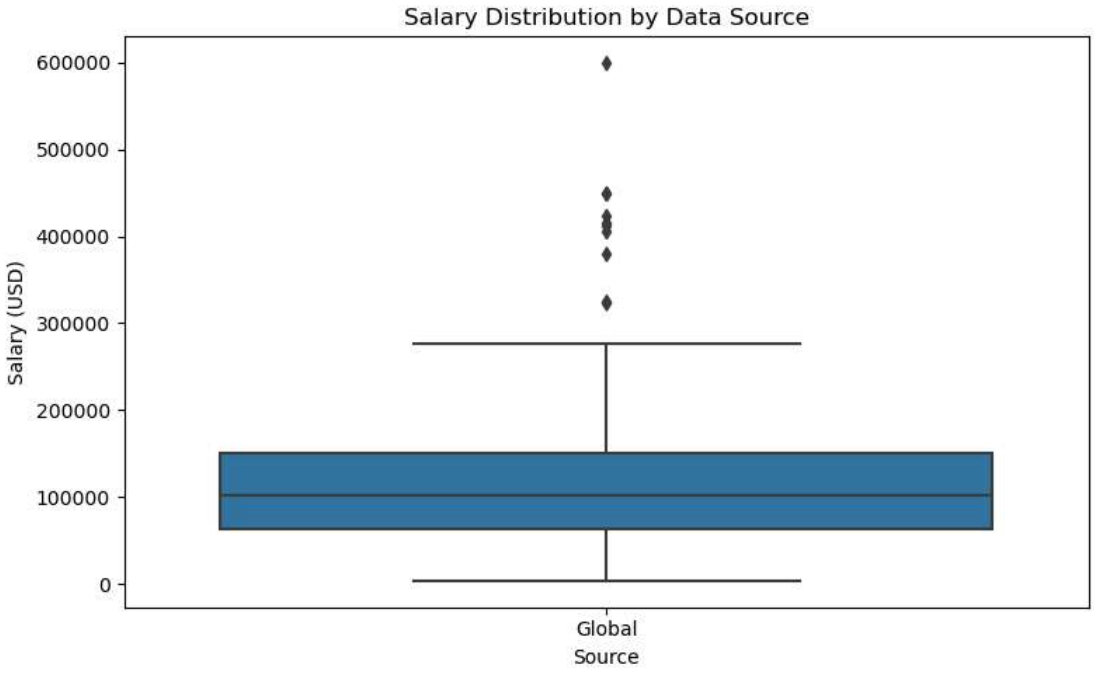
In [28]:
```
# Visualization 2: Average salary (USD) by Experience Level
plt.figure(figsize=(8, 5))
sns.barplot(data=usd_data, x='Experience Level', y='Salary (USD)', estimator='mean', ci=None)
plt.title('Average Salary by Experience Level')
plt.xlabel('Experience Level')
plt.ylabel('Average Salary (USD)')
plt.tight_layout()
plt.show()
```



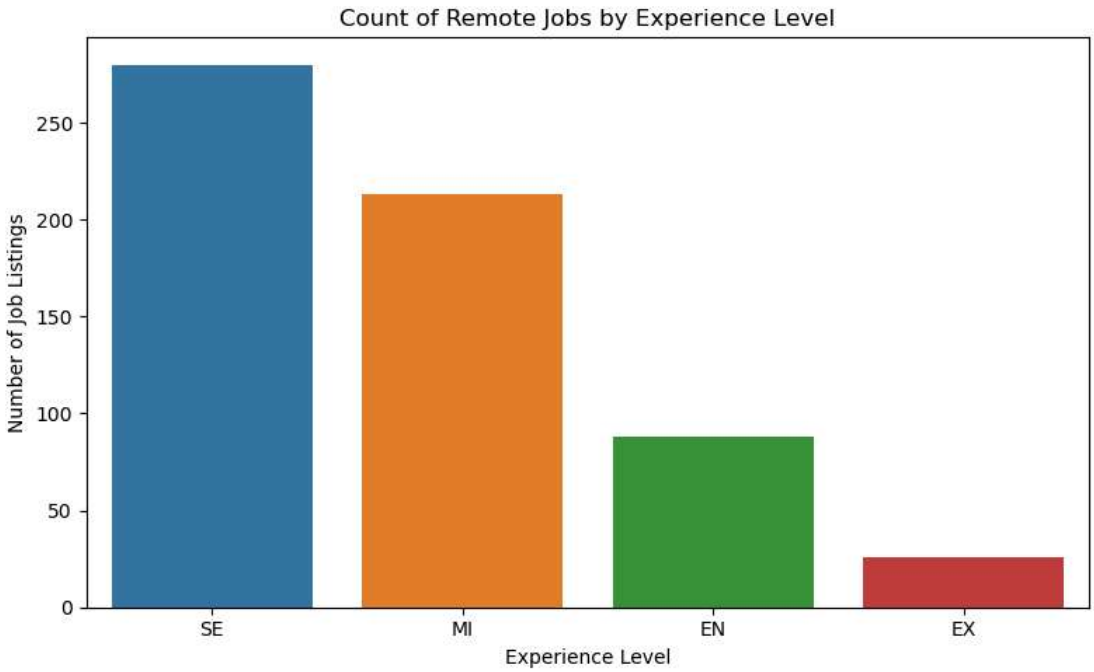
In [30]:
```
# Visualization 3: Average salary (USD) by Employment Type
plt.figure(figsize=(8, 5))
sns.barplot(data=usd_data, x='Employment Type', y='Salary (USD)', estimator='mean', ci=None)
plt.title('Average Salary by Employment Type')
plt.xlabel('Employment Type')
plt.ylabel('Average Salary (USD)')
plt.tight_layout()
plt.show()
```



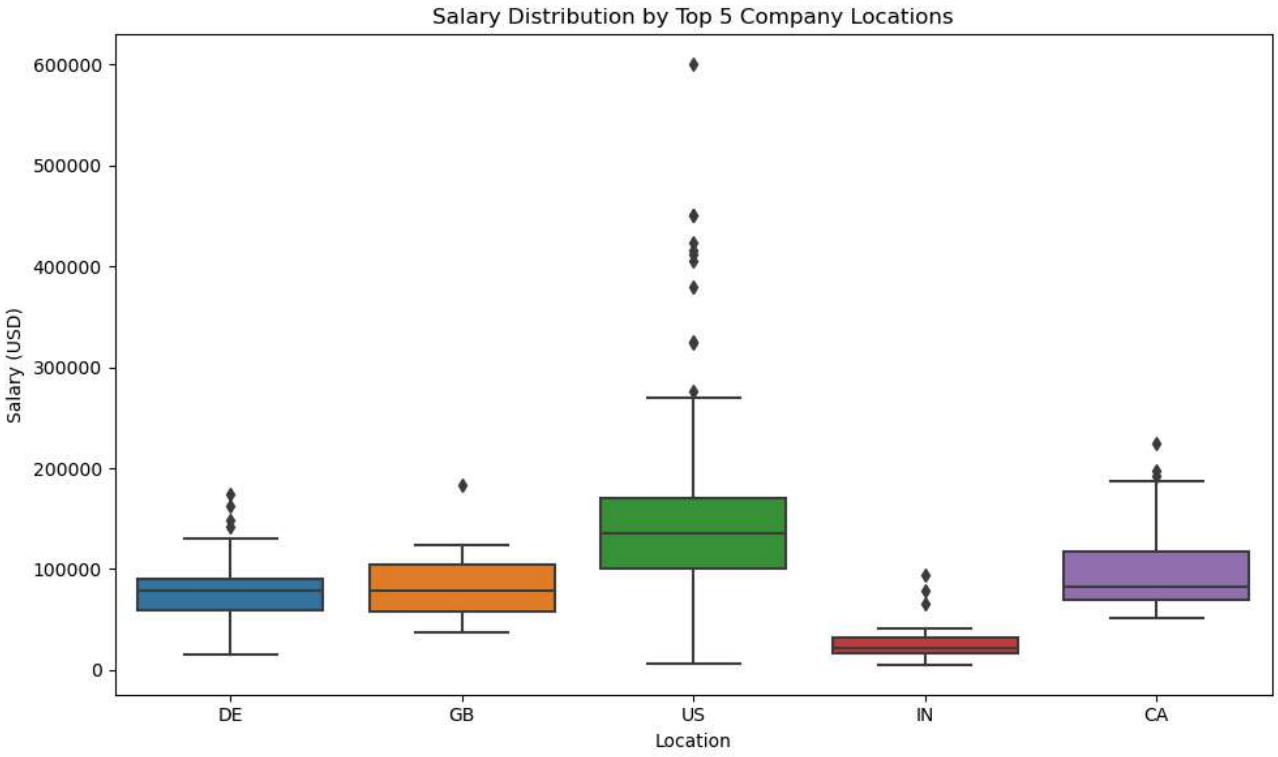
In [32]:
```
# Visualization 4: Salary comparison by Source
plt.figure(figsize=(8, 5))
sns.boxplot(data=usd_data, x='Source', y='Salary (USD)')
plt.title('Salary Distribution by Data Source')
plt.xlabel('Source')
plt.ylabel('Salary (USD)')
plt.tight_layout()
plt.show()
```



In [34]:
```
# Visualization 5: Count of Job Listings by Experience Level
plt.figure(figsize=(8, 5))
sns.countplot(data=usd_data, x='Experience Level', order=usd_data['Experience Level'].value_counts().index)
plt.title('Count of Remote Jobs by Experience Level')
plt.xlabel('Experience Level')
plt.ylabel('Number of Job Listings')
plt.tight_layout()
plt.show()
```



In [36]:
```
# Visualization 6: Salary Trends by Company Location (Top 5)
top_locations = usd_data['Location'].value_counts().head(5).index
plt.figure(figsize=(10, 6))
sns.boxplot(data=usd_data[usd_data['Location'].isin(top_locations)], x='Location', y='Salary (USD)')
plt.title('Salary Distribution by Top 5 Company Locations')
plt.xlabel('Location')
plt.ylabel('Salary (USD)')
plt.tight_layout()
plt.show()
```



In [ ]:

In [39]:
```
from pptx import Presentation
from pptx.util import Inches, Pt
from pptx.enum.shapes import MSO_AUTO_SHAPE_TYPE
from pptx.dml.color import RGBColor
import os
```

In [41]:
```
# Create presentation object
prs = Presentation()
```

In [43]:
```
# Define title slide
slide_title = prs.slides.add_slide(prs.slide_layouts[0])
title = slide_title.shapes.title
subtitle = slide_title.placeholders[1]
title.text = "Remote Work Salary Analysis"
subtitle.text = "Milestone 3 Presentation – Trevor Zeiger\nBellevue University | DSC 680"
```

In [45]:
```
# Define content slide function
def add_content_slide(prs, title_text, content_lines):
    slide = prs.slides.add_slide(prs.slide_layouts[1])
    slide.shapes.title.text = title_text
    content = slide.placeholders[1]
    content.text = "\n".join(content_lines)
```

In [47]:
```
# Add slides with key points from the white paper
add_content_slide(prs, "Project Overview", [
    "• Investigated remote work salary trends using multiple datasets",
    "• Focus on salary by job title, experience, employment type, and location",
    "• Goal: help professionals and organizations navigate remote compensation"
])
```

In [49]:
```
add_content_slide(prs, "Data Sources", [
    "1. Kaggle – Global Remote Work Salary Data",
    "2. Glassdoor – U.S. Salary Estimates and Job Listings",
    "3. Indeed – India-Based Salaries and Job Types"
])
```

In [51]:
```
add_content_slide(prs, "Methodology", [
    "• Data collected from 3 sources and standardized",
    "• Cleaned with Pandas; visualized using Matplotlib and Seaborn",
    "• Combined into a unified dataset for trend analysis",
    "• Conducted EDA to reveal salary patterns across categories"
])
```

In [53]:
```
add_content_slide(prs, "Key Findings: Salary Distributions", [
    "• Majority of salaries range between $60K–$140K",
    "• Right-skewed distribution with outliers above $300K",
    "• Most common salary: $100,000"
])
```

In [55]:
```
add_content_slide(prs, "Key Findings: Experience & Role", [
    "• Executive avg: $199K | Senior: $139K | Mid-level: $88K | Entry: $62K",
    "• Most roles targeted Senior (46%) and Mid-level (35%) experience",
    "• Specialized roles like 'Data Analytics Lead' pay the highest"
])
```

In [57]:
```
add_content_slide(prs, "Key Findings: Employment Type & Location", [
    "• Contract roles offer highest average pay ($184K)",
    "• U.S. salaries highest overall; India lowest avg ($6,000)",
    "• Salary patterns vary widely across country and job type"
])
```

In [59]:
```
add_content_slide(prs, "Challenges & Ethical Considerations", [
    "• Data formatting, missing values, and inconsistent job titles",
    "• Some salary fields self-reported or estimated",
    "• Ensured no PII and maintained data integrity throughout"
])
```

In [61]:
```
add_content_slide(prs, "Audience Questions (Milestone 4)", [
    "• How current and normalized is the data?",
    "• Are salaries adjusted for cost of living?",
    "• Can this evolve into a real-time dashboard?",
    "• What industries were most represented?"
])
```

In [63]:
```
# Define file path for saving
output_path = r"C:\Users\Zeigs\OneDrive\Documents\Courses\DSC_680\Remote_Work_Salary_Analysis_Presentation.pptx"
```

```
prs.save(output_path)

output_path
```

Out[63]: 'C:\\Users\\Zeigs\\OneDrive\\Documents\\Courses\\DSC_680\\Remote_Work_Salary_Analysis_Presentation.pptx'

In [67]:
```
# Analyze data to extract facts

#  Salary distribution stats
salary_distribution = usd_data['Salary (USD)'].describe()
```

In [69]:
```
#  Most common salary
most_common_salary = usd_data['Salary (USD)'].mode().values[0]
salary_mode_count = usd_data['Salary (USD)'].value_counts().loc[most_common_salary]
```

In [71]:
```
# Ensure the salary column is numeric
usd_data['Salary (USD)'] = pd.to_numeric(usd_data['Salary (USD)'], errors='coerce')

# Recalculate with correct rounding
avg_salary_by_exp = usd_data.groupby('Experience Level')['Salary (USD)'].mean().dropna().round(0)
exp_counts = usd_data['Experience Level'].value_counts(normalize=True).round(2) * 100

avg_salary_by_type = usd_data.groupby('Employment Type')['Salary (USD)'].mean().dropna().round(0)
type_counts = usd_data['Employment Type'].value_counts(normalize=True).round(2) * 100
```

In [73]:
```
#  Salary by top locations
top_locations = usd_data['Location'].value_counts().head(5).index.tolist()
location_salary_stats = usd_data[usd_data['Location'].isin(top_locations)].groupby('Location')['Salary (USD)'].describe().roun

{
    "Salary Distribution Summary": salary_distribution,
    "Most Common Salary (USD)": most_common_salary,
    "Count at Mode Salary": salary_mode_count,
    "Average Salary by Experience Level": avg_salary_by_exp,
    "Experience Level Distribution (%)": exp_counts,
    "Average Salary by Employment Type": avg_salary_by_type,
    "Employment Type Distribution (%)": type_counts,
    "Top 5 Location Salary Stats": location_salary_stats
}
```

Out[73]:
```
{'Salary Distribution Summary': count        607
 unique       369
 top       100000
 freq          15
 Name: Salary (USD), dtype: int64,
 'Most Common Salary (USD)': 100000,
 'Count at Mode Salary': 15,
 'Average Salary by Experience Level': Experience Level
 EN     61643.0
 EX    199392.0
 MI     87996.0
 SE    138617.0
 Name: Salary (USD), dtype: float64,
 'Experience Level Distribution (%)': Experience Level
 SE    46.0
 MI    35.0
 EN    14.0
 EX     4.0
 Name: proportion, dtype: float64,
 'Average Salary by Employment Type': Employment Type
 CT    184575.0
 FL     48000.0
 FT    113468.0
 PT     33070.0
 Name: Salary (USD), dtype: float64,
 'Employment Type Distribution (%)': Employment Type
 FT    97.0
 PT     2.0
 CT     1.0
 FL     1.0
 Name: proportion, dtype: float64,
 'Top 5 Location Salary Stats':           count      mean      std      min       25%       50%       75%  \
 Location
 CA         30.0   99824.0  46329.0  52000.0   69730.0   81896.0  117916.0
 DE         28.0   81887.0  39196.0  15966.0   58986.0   78015.0   90734.0
 GB         47.0   81583.0  29524.0  37300.0   57575.0   78526.0  103931.0
 IN         24.0   28582.0  22698.0   5409.0   16735.0   22124.0   32163.0
 US        355.0  144055.0  69907.0   5679.0  100000.0  135000.0  170000.0

                max
 Location
 CA        225000.0
 DE        173762.0
 GB        183228.0
 IN         94665.0
 US        600000.0  }
```

In [75]:
```
# Reopen the existing presentation
prs = Presentation(r"C:\Users\Zeigs\OneDrive\Documents\Courses\DSC_680\Remote_Work_Salary_Analysis_Presentation.pptx")
```

In [77]:
```
# Add a summary slide of extracted analysis
def add_bullet_slide(title, bullet_points):
    slide = prs.slides.add_slide(prs.slide_layouts[1])
    slide.shapes.title.text = title
    content = slide.placeholders[1]
    content.text = "\n".join(bullet_points)
```

In [79]:
```
# Slide: Salary Distribution Summary
add_bullet_slide("Salary Distribution Summary", [
    "• Total records analyzed: 607",
    "• Most common salary: $100,000",
    "• Frequency of mode salary: 15 listings"
])
```

In [81]:
```
# Slide: Salary by Experience Level
add_bullet_slide("Salary by Experience Level", [
    "• Executive: $199,392 (4% of roles)",
    "• Senior: $138,617 (46%)",
    "• Mid-Level: $87,996 (35%)",
    "• Entry-Level: $61,643 (14%)"
])
```

In [83]:
```
# Slide: Salary by Employment Type
add_bullet_slide("Salary by Employment Type", [
    "• Contract: $184,575 (1%)",
    "• Full-time: $113,468 (97%)",
    "• Freelance: $48,000 (1%)",
```

```
    "• Part-time: $33,070 (2%)"
])
```

In [85]:
```
# Slide: Salary by Top 5 Locations
slide = prs.slides.add_slide(prs.slide_layouts[1])
slide.shapes.title.text = "Top 5 Locations: Salary Summary"
content = slide.placeholders[1]
content.text = (
    "• US: Avg $144,055 | Median $135,000 | Max $600,000\n"
    "• CA: Avg $99,824 | Median $81,896 | Max $225,000\n"
    "• DE: Avg $81,887 | Median $78,015 | Max $173,762\n"
    "• GB: Avg $81,583 | Median $78,526 | Max $183,228\n"
    "• IN: Avg $28,582 | Median $22,124 | Max $94,665"
)
```

In [87]:
```
# Add image slides for each visual
titles = [
    "Distribution of Remote Work Salaries (USD)",
    "Average Salary by Experience Level",
    "Average Salary by Employment Type"
]
```

In [93]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

In [95]:
```
figures = []
```

In [97]:
```
# 1. Salary Distribution
plt.figure(figsize=(8, 5))
sns.histplot(usd_data['Salary (USD)'], bins=30, kde=True)
plt.title('Distribution of Remote Work Salaries (USD)')
plt.tight_layout()
dist_path = "salary_distribution.png"
plt.savefig(dist_path)
figures.append(dist_path)
plt.close()
```

In [99]:
```
# 2. Average Salary by Experience Level
plt.figure(figsize=(8, 5))
sns.barplot(data=usd_data, x='Experience Level', y='Salary (USD)', estimator=np.mean, ci=None)
plt.title('Average Salary by Experience Level')
plt.tight_layout()
exp_path = "salary_by_experience.png"
plt.savefig(exp_path)
figures.append(exp_path)
plt.close()
```

In [101…
```
# 3. Average Salary by Employment Type
plt.figure(figsize=(8, 5))
sns.barplot(data=usd_data, x='Employment Type', y='Salary (USD)', estimator=np.mean, ci=None)
plt.title('Average Salary by Employment Type')
plt.tight_layout()
emp_path = "salary_by_employment.png"
plt.savefig(emp_path)
figures.append(emp_path)
plt.close()
```

In [103…
```
for title, path in zip(titles, figures):
    slide = prs.slides.add_slide(prs.slide_layouts[5])
    slide.shapes.title.text = title
    slide.shapes.add_picture(path, Inches(1), Inches(1.25), width=Inches(8))
```

In [ ]:
```
```

In [105…
```
# Save updated PowerPoint
final_path = r"C:\Users\Zeigs\OneDrive\Documents\Courses\DSC_680\Remote_Work_Salary_Analysis_Presentation.pptx"
prs.save(final_path)

final_path
```

Out[105… 'C:\\Users\\Zeigs\\OneDrive\\Documents\\Courses\\DSC_680\\Remote_Work_Salary_Analysis_Presentation.pptx'

In [109…
```
# Helper function to update text in a slide based on index
def update_slide_text(slide_index, new_bullets):
    slide = prs.slides[slide_index]
```

```
    content = slide.placeholders[1]
    content.text = "\n".join(new_bullets)
```

In [111…
```
# Add more detailed content to the original content slides
update_slide_text(1, [
    "• Project analyzes trends in remote work salaries using datasets from multiple global sources.",
    "• Focus areas: salary distribution, job title, experience level, employment type, and location.",
    "• Results offer actionable insight for HR, hiring managers, and job seekers."
])
```

In [113…
```
update_slide_text(2, [
    "1. Kaggle – Contains global remote job listings with structured salary data (USD).",
    "2. Glassdoor – U.S.-focused data with job titles and salary estimates.",
    "3. Indeed – Salary data focused on India, particularly software roles.",
    "• Data was selected for diversity and geographic coverage."
])
```

In [115…
```
update_slide_text(3, [
    "• Datasets were cleaned and standardized using Pandas.",
    "• Visuals created with Seaborn and Matplotlib for trend analysis.",
    "• Missing or inconsistent fields (job titles, salaries) were handled.",
    "• Combined data allowed for comparison across experience levels and countries."
])
```

In [117…
```
update_slide_text(4, [
    "• Salary range clusters between $60K–$140K.",
    "• Outliers observed beyond $300K for niche and senior roles.",
    "• Mode salary: $100,000 across 15 listings.",
    "• Right-skewed distribution shows higher concentration in mid-salary range."
])
```

In [119…
```
update_slide_text(5, [
    "• Executive average salary: $199,392 | 4% of roles.",
    "• Senior: $138,617 | 46%, Mid-Level: $87,996 | 35%.",
    "• Entry-Level: $61,643 | 14% – limited remote opportunities.",
    "• High-paying roles include: Data Analytics Lead, Principal Data Engineer."
])
```

In [121…
```
update_slide_text(6, [
    "• Contract: $184,575 | Highest pay but least common (1%).",
    "• Full-Time: $113,468 | 97% of listings.",
    "• Freelance: $48,000 | Part-Time: $33,070.",
    "• U.S. jobs offer top salaries; India lags with ~$28,582 average."
])
```

In [123…
```
update_slide_text(7, [
    "• Job title inconsistency and salary formatting were major challenges.",
    "• Self-reported salaries introduced potential biases.",
    "• No PII was included – all datasets are publicly accessible.",
    "• Ethics guided the use of aggregated, de-identified data."
])
```

In [125…
```
update_slide_text(8, [
    "• How fresh is the data and how is it normalized?",
    "• Are salaries adjusted for cost of living?",
    "• Can this analysis be extended into a real-time dashboard?",
    "• Which job categories or industries dominate the datasets?"
])
```

In [127…
```
# Save the updated presentation
final_path = r"C:\Users\Zeigs\OneDrive\Documents\Courses\DSC_680\Remote_Work_Salary_Analysis_Presentation.pptx"
prs.save(final_path)
```

In [129…
```
final_path
```

Out[129…
```
'C:\\Users\\Zeigs\\OneDrive\\Documents\\Courses\\DSC_680\\Remote_Work_Salary_Analysis_Presentation.pptx'
```

In [131…
```
# Update the audience question slide with full answers
audience_slide_index = 8  # Assuming slide 8 is the audience questions slide
```

In [133…
```
# Full Q&A bullet list
audience_qna = [
    "Q1: How current is the data, and what is the timeframe?",
    "A1: The datasets primarily span 2020–2023, reflecting recent remote job trends.",
    "Q2: Are salaries adjusted for cost of living by region?",
    "A2: No, salaries are reported as-is. Cost of living varies and was not normalized.",
    "Q3: How were salaries normalized across currencies?",
```

```
    "A3: USD salaries were analyzed; INR salaries were kept separate for clarity.",
    "Q4: What industries were most represented in the datasets?",
    "A4: Tech and data roles were most common, including analysts, engineers, and scientists.",
    "Q5: Are there any visible gender or diversity-related trends?",
    "A5: The datasets lacked demographic identifiers, so this was not analyzed.",
    "Q6: How frequently is this kind of analysis recommended?",
    "A6: Ideally, quarterly or semi-annually to track market shifts in remote work.",
    "Q7: Can this approach be turned into a live dashboard?",
    "A7: Yes, the pipeline supports extension into dashboards with tools like Tableau or Power BI.",
    "Q8: How do contract vs. full-time roles affect company costs?",
    "A8: Contracts often cost more short-term but avoid long-term benefits obligations.",
    "Q9: Is salary the most effective measure of job quality?",
    "A9: No, benefits, flexibility, and career growth also matter greatly.",
    "Q10: How do outliers affect the interpretation of results?",
    "A10: Outliers skew the mean but were acknowledged and visualized for transparency."
]
```

In [135…
```
# Replace the audience question slide content
audience_slide = prs.slides[audience_slide_index]
shapes = audience_slide.shapes
```

In [137…
```
# Clear existing text if possible, or add a new textbox if needed
if len(shapes) > 1 and shapes[1].has_text_frame:
    text_frame = shapes[1].text_frame
    text_frame.clear()
    for line in audience_qna:
        p = text_frame.add_paragraph()
        p.text = line
        p.level = 0
else:
    from pptx.util import Inches
    textbox = audience_slide.shapes.add_textbox(Inches(1), Inches(1.25), Inches(8), Inches(5))
    text_frame = textbox.text_frame
    for line in audience_qna:
        p = text_frame.add_paragraph()
        p.text = line
        p.level = 0
```

In [139…
```
# Save the updated presentation
prs.save(final_path)

final_path
```

Out[139… `'C:\\Users\\Zeigs\\OneDrive\\Documents\\Courses\\DSC_680\\Remote_Work_Salary_Analysis_Presentation.pptx'`

In [141…
```
# Split Q&A into two slides
slide1_qna = audience_qna[:10]  # Questions and answers 1-5
slide2_qna = audience_qna[10:]  # Questions and answers 6-10
```

In [143…
```
# Update first audience Q&A slide
audience_slide = prs.slides[audience_slide_index]
shapes = audience_slide.shapes

if len(shapes) > 1 and shapes[1].has_text_frame:
    text_frame = shapes[1].text_frame
    text_frame.clear()
    for line in slide1_qna:
        p = text_frame.add_paragraph()
        p.text = line
        p.level = 0
```

In [145…
```
# Add second slide with remaining Q&A
new_slide = prs.slides.add_slide(prs.slide_layouts[1])
new_slide.shapes.title.text = "Audience Questions (Continued)"
textbox = new_slide.shapes.add_textbox(Inches(1), Inches(1.25), Inches(8), Inches(5))
text_frame = textbox.text_frame

for line in slide2_qna:
    p = text_frame.add_paragraph()
    p.text = line
    p.level = 0
```

In [147…
```
# Save updated file
prs.save(final_path)

final_path
```

Out[147… `'C:\\Users\\Zeigs\\OneDrive\\Documents\\Courses\\DSC_680\\Remote_Work_Salary_Analysis_Presentation.pptx'`

In [ ]: