In [3]:
```
!pip install mysql-connector-python
```

```
Collecting mysql-connector-python
  Downloading mysql_connector_python-9.0.0-cp38-cp38-win_amd64.whl (14.3
MB)
Installing collected packages: mysql-connector-python
Successfully installed mysql-connector-python-9.0.0
```

In [17]:
```
# Establish a connection
import mysql.connector
db_connection = mysql.connector.connect(
    host='localhost',       # e.g., 'localhost' or '127.0.0.1'
    user='root',   # e.g., 'root'
    password='Heythere@12',
    database='animaladoption1'
)

# Check the connection
if connection.is_connected():
    print("Successfully connected to the database")
```

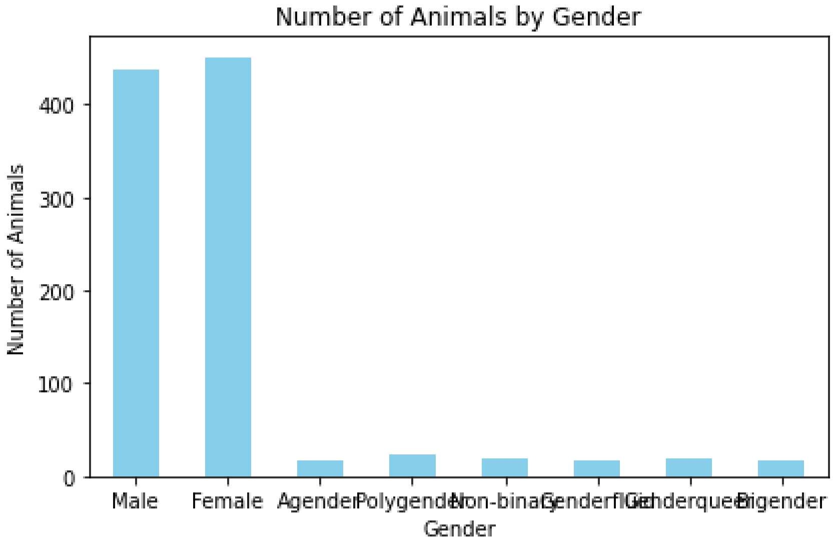
```
Successfully connected to the database
```

In [19]:
```
import mysql.connector

# Step 1: Connect to the database
db_connection = mysql.connector.connect(
    host='localhost',        # Replace with your host, e.g., '127.0.0.1'
    user='root',    # Your MySQL username
    password='Heythere@12', # Your MySQL password
    database='animaladoption1'  # Your database name
)

# Step 2: Create a cursor object
cursor = connection.cursor()
```

In [24]:
```
import warnings

# Suppress all warnings
warnings.filterwarnings("ignore")
```

```
In [26]: import mysql.connector
         import pandas as pd
         import matplotlib.pyplot as plt
         query = """
             SELECT AnimalGender, COUNT(*) AS NumAnimals
             FROM Animal
             GROUP BY AnimalGender
         """

         # Load the data into a pandas DataFrame
         df = pd.read_sql(query, db_connection)

         # Plot the number of animals by gender
         plt.figure(figsize=(8, 6))
         df.plot(kind='bar', x='AnimalGender', y='NumAnimals', color='skyblue', lege
         plt.title('Number of Animals by Gender')
         plt.xlabel('Gender')
         plt.ylabel('Number of Animals')
         plt.xticks(rotation=0)
         plt.tight_layout()
         plt.show()
```

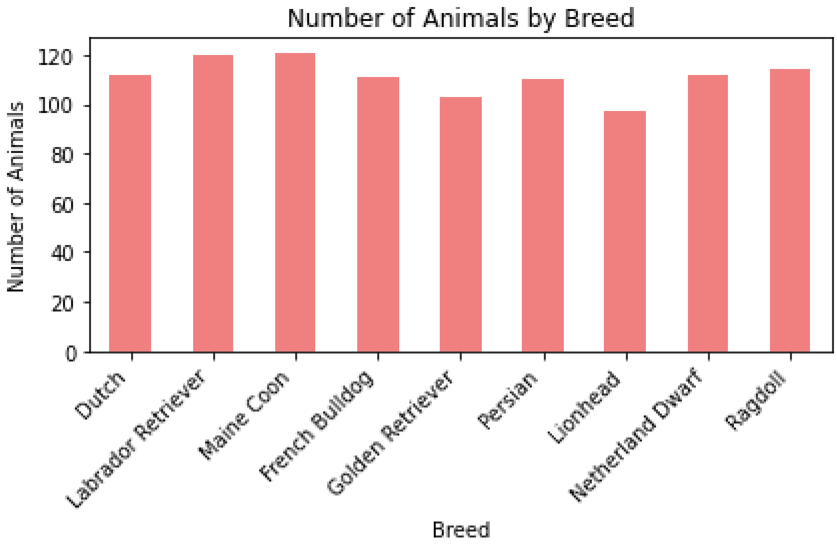
```
<Figure size 576x432 with 0 Axes>
```



In [27]:

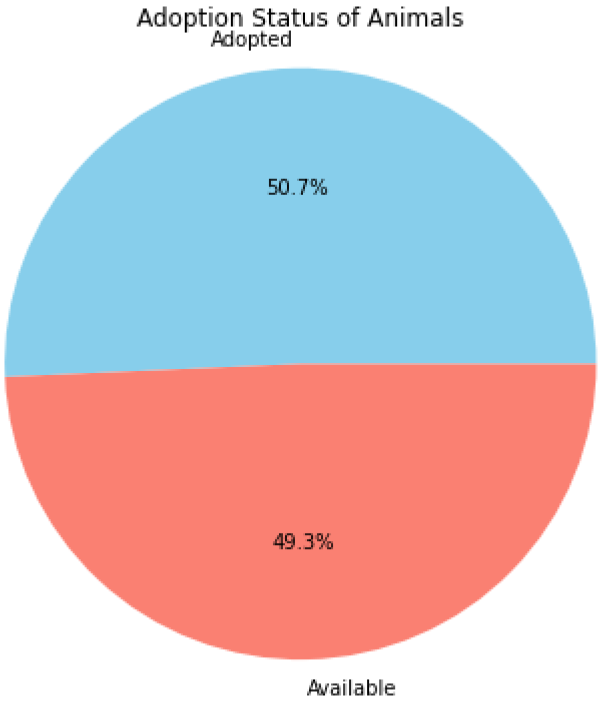
```
# Query to get the number of animals by breed
query = """
    SELECT Breed, COUNT(*) AS NumAnimals
    FROM Animal
    GROUP BY Breed
"""

# Load the data into a pandas DataFrame
df = pd.read_sql(query, db_connection)

# Plot the number of animals by breed
plt.figure(figsize=(12, 6))
df.plot(kind='bar', x='Breed', y='NumAnimals', color='lightcoral', legend=F
plt.title('Number of Animals by Breed')
plt.xlabel('Breed')
plt.ylabel('Number of Animals')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

```
<Figure size 864x432 with 0 Axes>
```



In [28]:

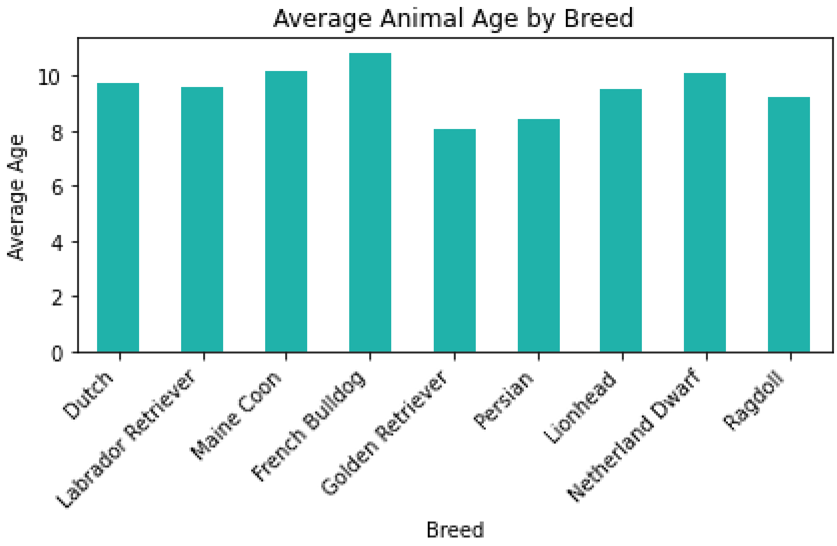
```
# Query to get the adoption status of animals
query = "SELECT AdoptionStatus, COUNT(*) AS NumAnimals FROM Animal GROUP BY

# Load the data into a pandas DataFrame
df = pd.read_sql(query, db_connection)

# Plot the adoption status of animals
plt.figure(figsize=(8, 6))
plt.pie(df['NumAnimals'], labels=df['AdoptionStatus'], autopct='%1.1f%%', c
plt.title('Adoption Status of Animals')
plt.axis('equal')  # Equal aspect ratio ensures that pie chart is circular.
plt.show()
```



In [29]:

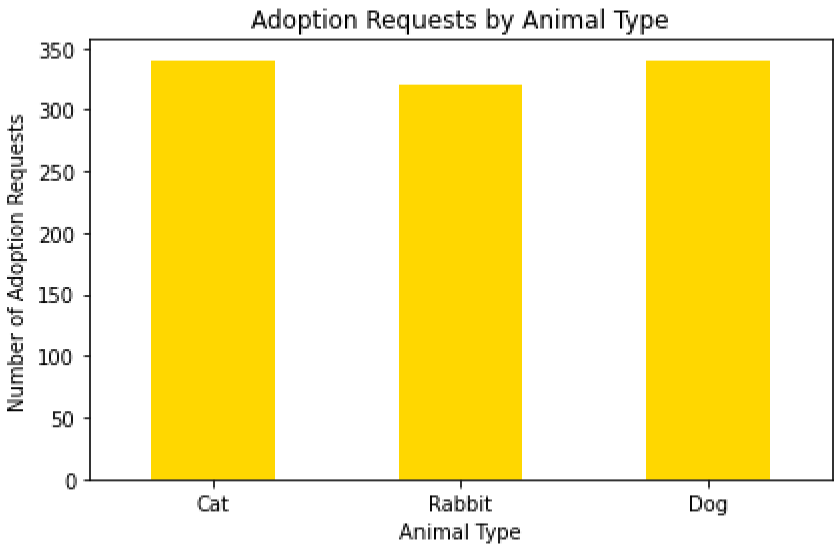
```
# Query to get the average animal age by breed
query = """
    SELECT Breed, AVG(AnimalAge) AS AvgAge
    FROM Animal
    GROUP BY Breed
"""

# Load the data into a pandas DataFrame
df = pd.read_sql(query, db_connection)

# Plot the average animal age by breed
plt.figure(figsize=(12, 6))
df.plot(kind='bar', x='Breed', y='AvgAge', color='lightseagreen', legend=Fa
plt.title('Average Animal Age by Breed')
plt.xlabel('Breed')
plt.ylabel('Average Age')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

```
<Figure size 864x432 with 0 Axes>
```



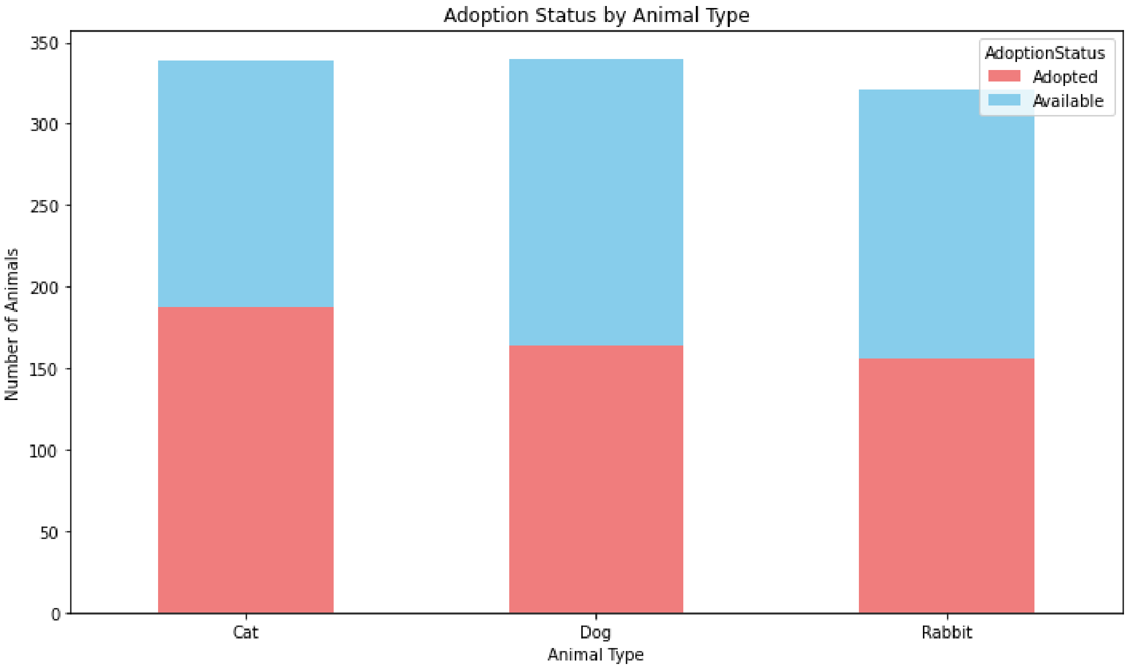
```
In [30]: # Query to get the number of adoption requests by animal type
         query = """
             SELECT a.AnimalType, COUNT(ar.RequestID) AS NumAdoptionRequests
             FROM Animal a
             JOIN AdoptionRequest ar ON a.AnimalID = ar.AnimalID
             GROUP BY a.AnimalType
         """

         # Load the data into a pandas DataFrame
         df = pd.read_sql(query, db_connection)

         # Plot the adoption requests by animal type
         plt.figure(figsize=(10, 6))
         df.plot(kind='bar', x='AnimalType', y='NumAdoptionRequests', color='gold',
         plt.title('Adoption Requests by Animal Type')
         plt.xlabel('Animal Type')
         plt.ylabel('Number of Adoption Requests')
         plt.xticks(rotation=0)
         plt.tight_layout()
         plt.show()
```

```
<Figure size 720x432 with 0 Axes>
```

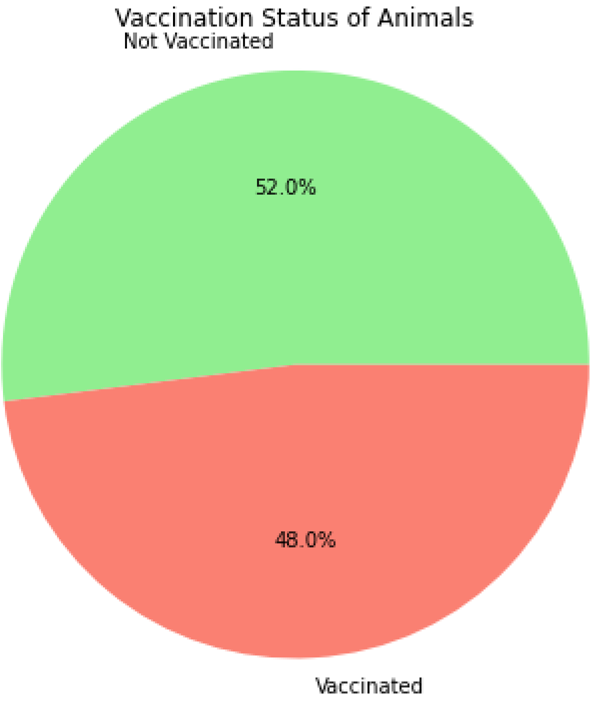


In [31]:

```
# Query to get adoption status by animal type
query = """
    SELECT AnimalType, AdoptionStatus, COUNT(*) AS NumAnimals
    FROM Animal
    GROUP BY AnimalType, AdoptionStatus
"""

# Load the data into a pandas DataFrame
df = pd.read_sql(query, db_connection)

# Pivot the data to get adoption status by animal type
df_pivot = df.pivot(index='AnimalType', columns='AdoptionStatus', values='N

# Plot the adoption status by animal type
df_pivot.plot(kind='bar', stacked=True, figsize=(10, 6), color=['lightcoral
plt.title('Adoption Status by Animal Type')
plt.xlabel('Animal Type')
plt.ylabel('Number of Animals')
plt.xticks(rotation=0)
plt.tight_layout()
plt.show()
```

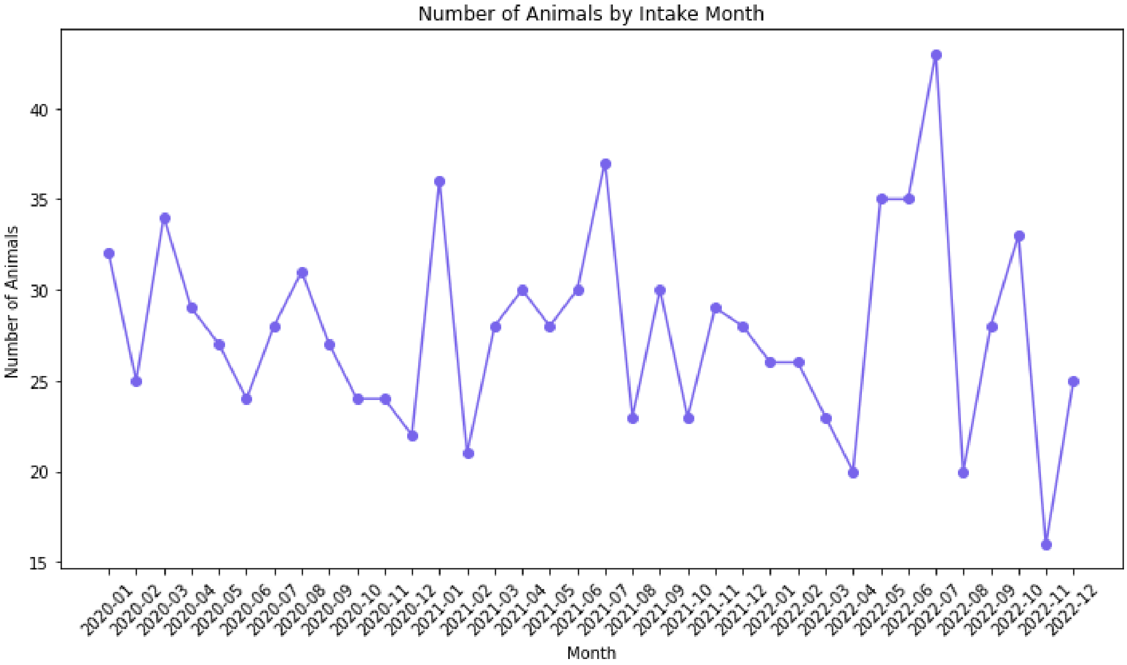


In [32]:
```
# Query to get the vaccination status of animals
query = "SELECT VaccinationStatus, COUNT(*) AS NumAnimals FROM Animal GROUP

# Load the data into a pandas DataFrame
df = pd.read_sql(query, db_connection)

# Plot the vaccination status distribution
plt.figure(figsize=(8, 6))
plt.pie(df['NumAnimals'], labels=df['VaccinationStatus'], autopct='%1.1f%%'
plt.title('Vaccination Status of Animals')
plt.axis('equal')  # Equal aspect ratio ensures that pie chart is circular
plt.show()
```



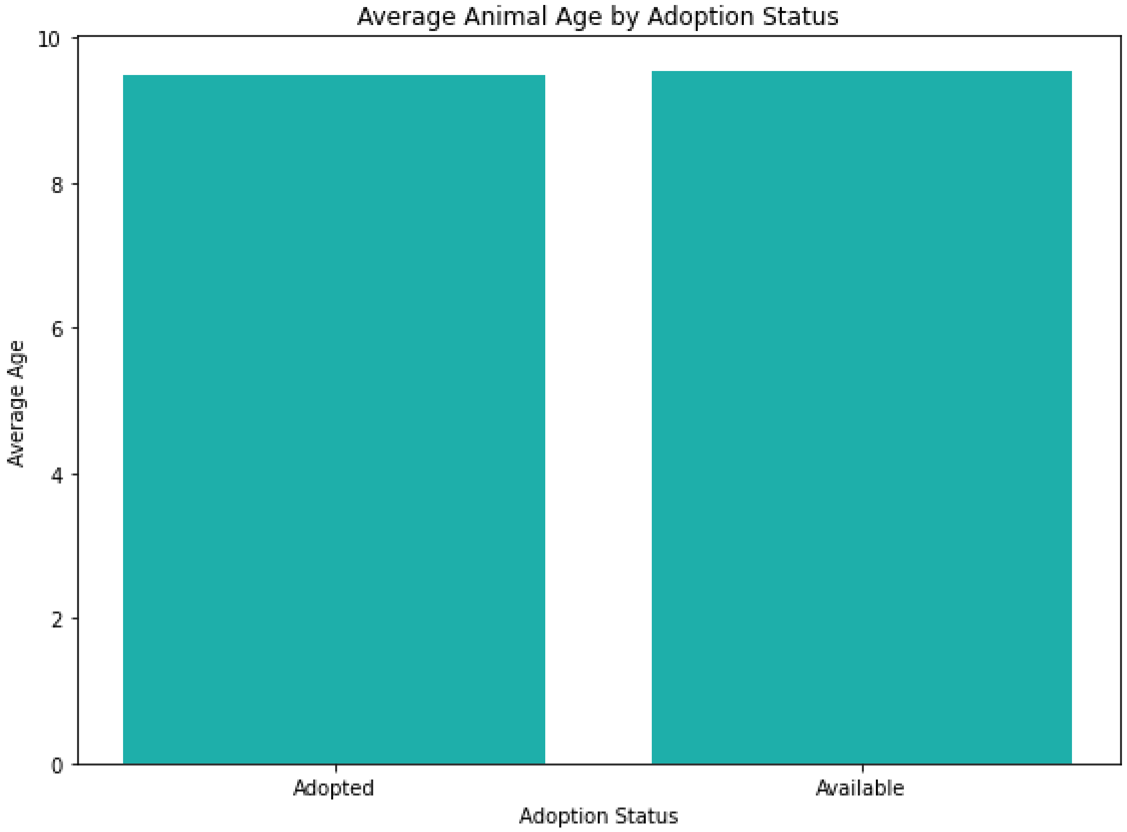
In [34]:

```
 #Query to get the number of animals by intake date (grouped by month)
query = """
    SELECT DATE_FORMAT(IntakeDate, '%Y-%m') AS Month, COUNT(*) AS NumAnimal
    FROM Animal
    GROUP BY Month
    ORDER BY Month;
"""

# Load the data into a pandas DataFrame
df = pd.read_sql(query, db_connection)

# Plot the number of animals by intake date (monthly)
plt.figure(figsize=(10, 6))
plt.plot(df['Month'], df['NumAnimals'], marker='o', color='mediumslateblue'
plt.title('Number of Animals by Intake Month')
plt.xlabel('Month')
plt.ylabel('Number of Animals')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



In [35]:

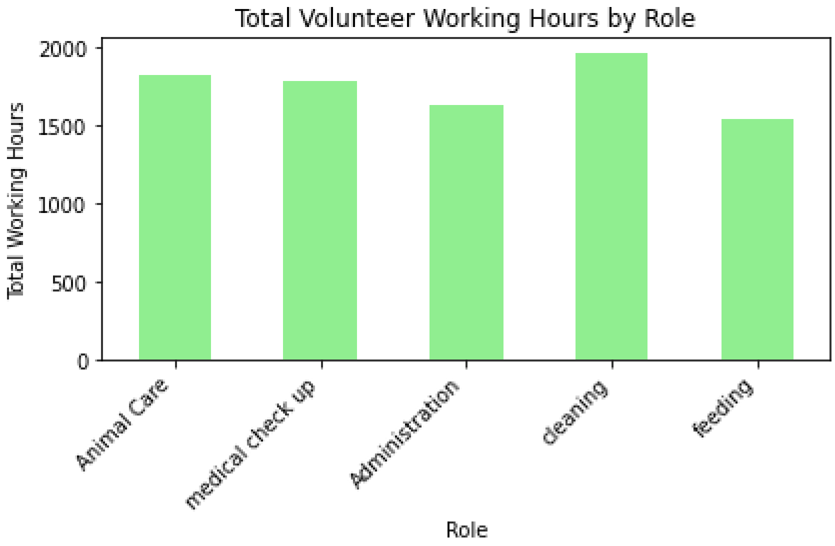
```
# Query to get the average age of animals by adoption status
query = """
    SELECT AdoptionStatus, AVG(AnimalAge) AS AvgAge
    FROM Animal
    GROUP BY AdoptionStatus
"""

# Load the data into a pandas DataFrame
df = pd.read_sql(query, db_connection)

# Plot the average animal age by adoption status
plt.figure(figsize=(8, 6))
plt.bar(df['AdoptionStatus'], df['AvgAge'], color='lightseagreen')
plt.title('Average Animal Age by Adoption Status')
plt.xlabel('Adoption Status')
plt.ylabel('Average Age')
plt.tight_layout()
plt.show()
```



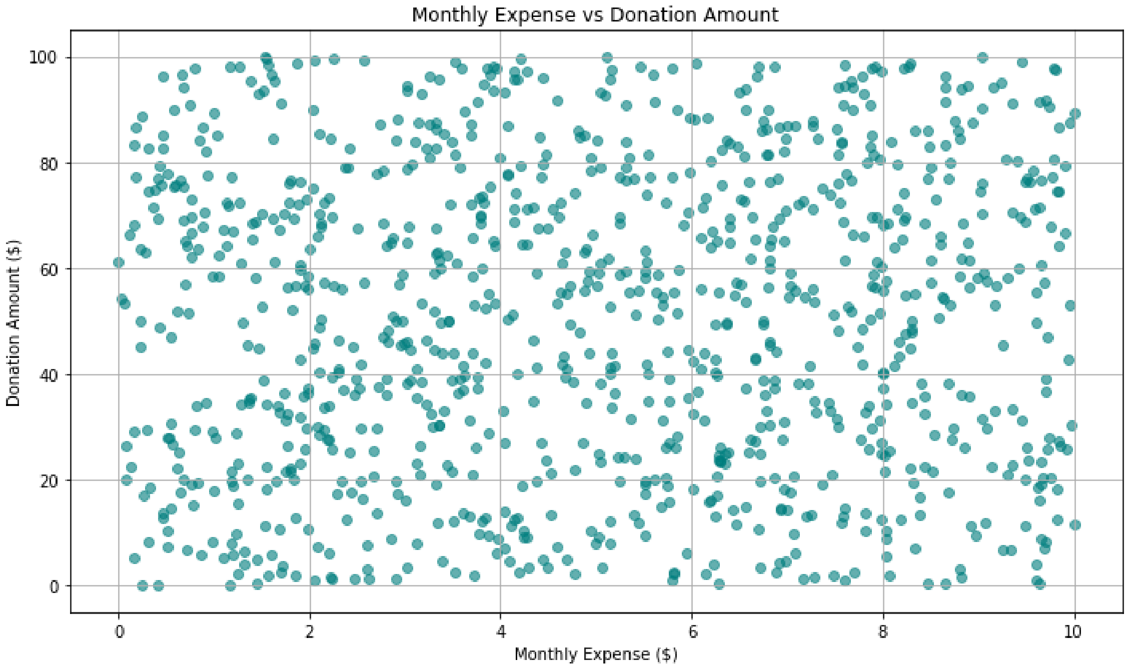
In [36]:

```
# Query to get total working hours by volunteer role
query = """
    SELECT Role, SUM(WorkingHours) AS TotalWorkingHours
    FROM Volunteer
    GROUP BY Role
"""

# Load the data into a pandas DataFrame
df = pd.read_sql(query, db_connection)

# Plot the total working hours by volunteer role
plt.figure(figsize=(10, 6))
df.plot(kind='bar', x='Role', y='TotalWorkingHours', color='lightgreen', le
plt.title('Total Volunteer Working Hours by Role')
plt.xlabel('Role')
plt.ylabel('Total Working Hours')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

```
<Figure size 720x432 with 0 Axes>
```



```
In [37]: #Query to get monthly expenses and donation amounts
         query = """
             SELECT f.MonthlyExpense, d.DonationAmount
             FROM FinancialRecord f
             JOIN Donation d ON f.FinancialRecordID = d.FinancialRecordID
         """

         # Load the data into a pandas DataFrame
         df = pd.read_sql(query, db_connection)

         # Scatter plot of monthly expenses vs donation amounts
         plt.figure(figsize=(10, 6))
         plt.scatter(df['MonthlyExpense'], df['DonationAmount'], color='teal', alpha

         # Add titles and labels
         plt.title('Monthly Expense vs Donation Amount')
         plt.xlabel('Monthly Expense ($)')
         plt.ylabel('Donation Amount ($)')
         plt.grid(True)

         # Show the plot
         plt.tight_layout()
         plt.show()
```



```
In [ ]:
```