```
Addanki Varun
varun.21bce9948@vitapstudent.ac.in
CSE AI AND ML
VIT AP
```

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

```
In [5]: df = pd.read_csv("/content/penguins_size.csv")
        df
```

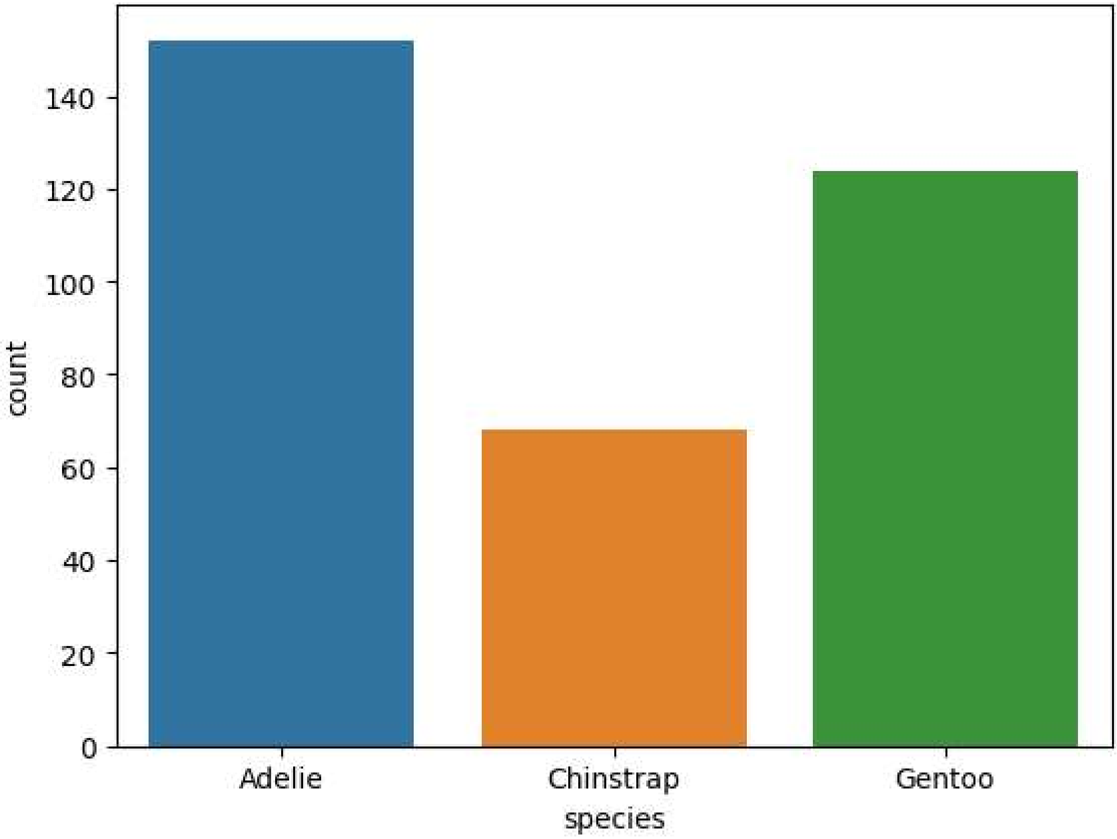
Out[5]:

	species	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_ma
0	Adelie	Torgersen	39.1	18.7	181.0	37
1	Adelie	Torgersen	39.5	17.4	186.0	38
2	Adelie	Torgersen	40.3	18.0	195.0	32
3	Adelie	Torgersen	NaN	NaN	NaN	
4	Adelie	Torgersen	36.7	19.3	193.0	34
...	...	...	...	...	...	
339	Gentoo	Biscoe	NaN	NaN	NaN	
340	Gentoo	Biscoe	46.8	14.3	215.0	48
341	Gentoo	Biscoe	50.4	15.7	222.0	57
342	Gentoo	Biscoe	45.2	14.8	212.0	52
343	Gentoo	Biscoe	49.9	16.1	213.0	54

344 rows × 7 columns

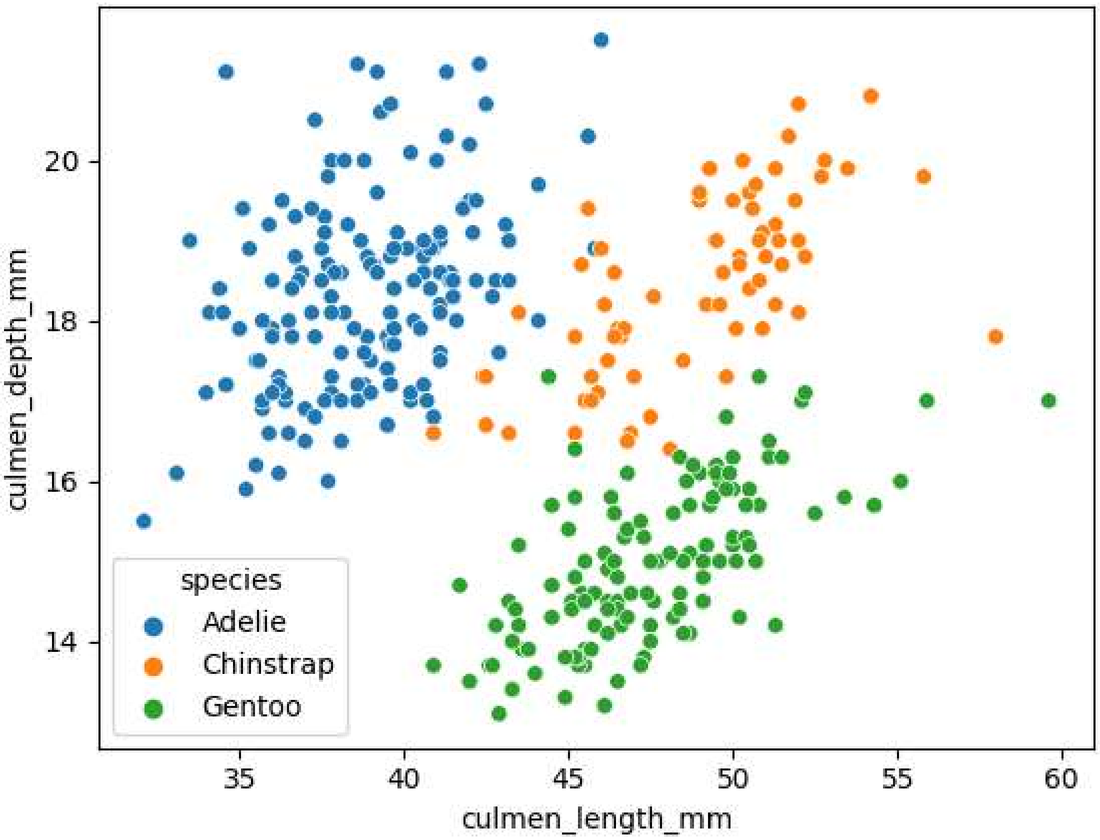
In [9]:

```
# Univariate Analysis
sns.countplot(x='species', data=df)
plt.show()
```

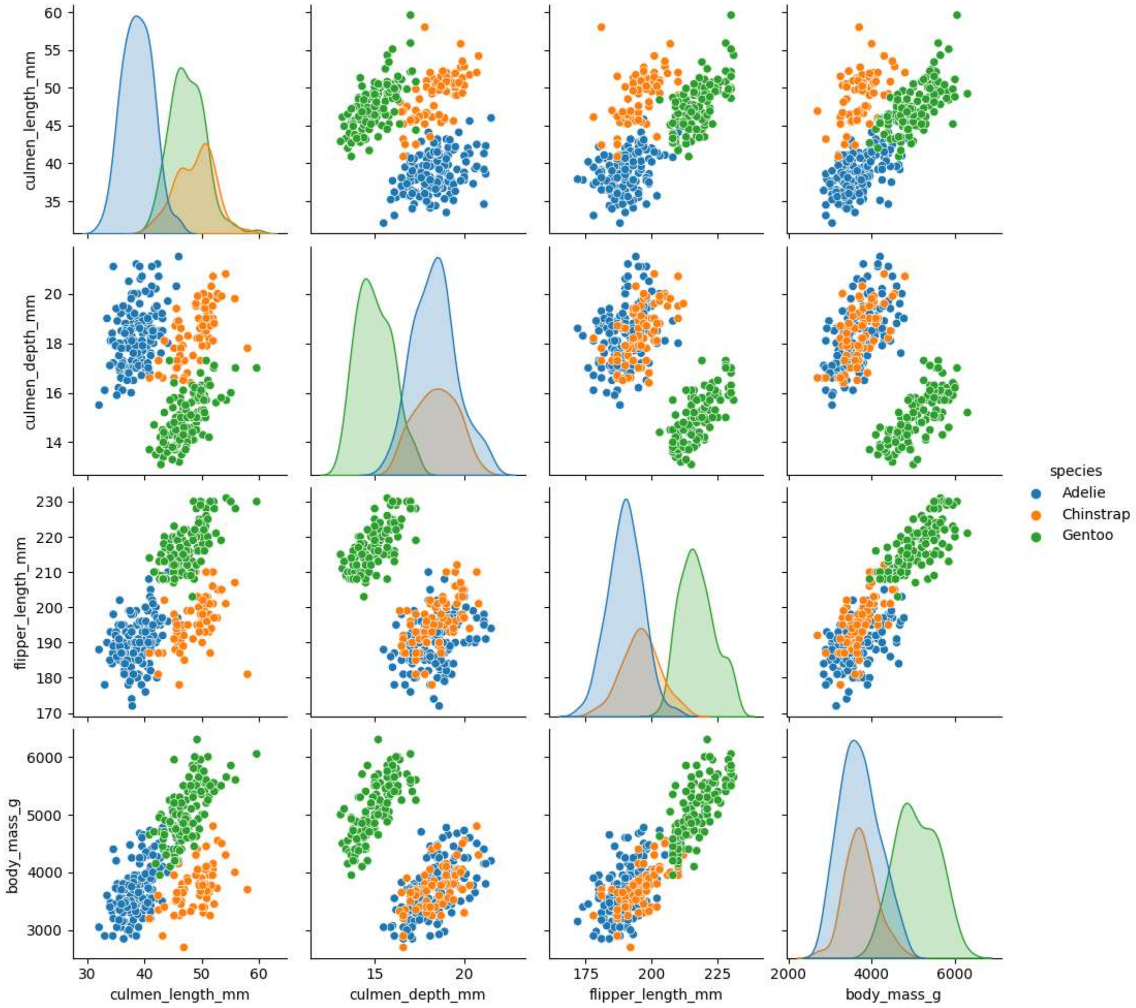


In [11]:

```
# Bi-Variate Analysis
sns.scatterplot(x=df['culmen_length_mm'], y=df['culmen_depth_mm'], hue=df['
plt.show()
```



In [13]:
```
# Multi-Variate Analysis
sns.pairplot(df, hue='species')
plt.show()
```



In [14]:
```
df.describe()
```

Out[14]:

	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g
count	342.000000	342.000000	342.000000	342.000000
mean	43.921930	17.151170	200.915205	4201.754386
std	5.459584	1.974793	14.061714	801.954536
min	32.100000	13.100000	172.000000	2700.000000
25%	39.225000	15.600000	190.000000	3550.000000
50%	44.450000	17.300000	197.000000	4050.000000
75%	48.500000	18.700000	213.000000	4750.000000
max	59.600000	21.500000	231.000000	6300.000000

In [15]:

```
# Check for missing values
print(df.isnull().sum())

# Impute or remove (example: using mean for numeric columns)
df['culmen_length_mm'].fillna(df['culmen_length_mm'].mean(), inplace=True)
```

```
species               0
island                0
culmen_length_mm      2
culmen_depth_mm       2
flipper_length_mm     2
body_mass_g           2
sex                  10
dtype: int64
```

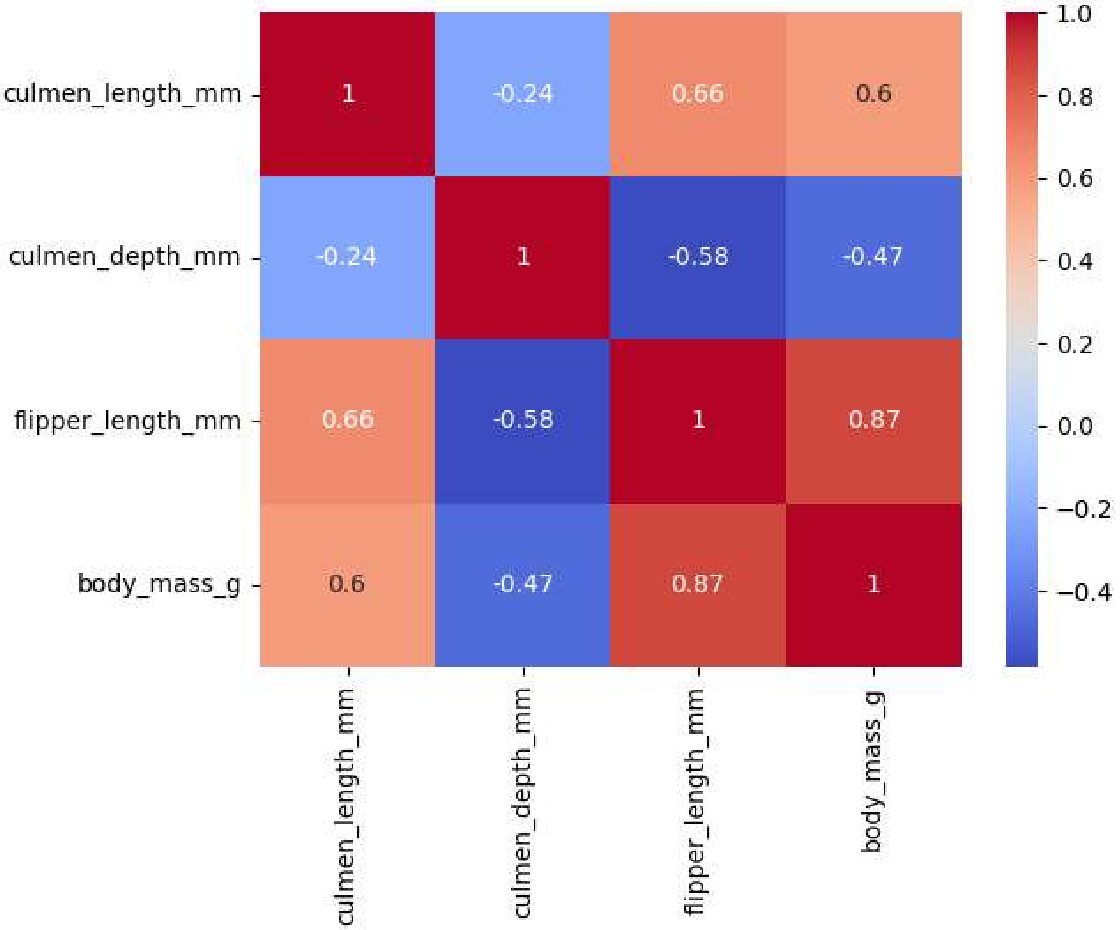
In [18]:

```
Q1 = df.quantile(0.25, numeric_only=True)
Q3 = df.quantile(0.75, numeric_only=True)
IQR = Q3 - Q1

numerical_cols = df.select_dtypes(include=['float64', 'int64']).columns
outliers = ((df[numerical_cols] < (Q1 - 1.5 * IQR)) | (df[numerical_cols] >

df = df[~outliers.any(axis=1)]
```

In [19]:

```
correlation = df.corr()
sns.heatmap(correlation, annot=True, cmap='coolwarm')
plt.show()
```

```
<ipython-input-19-b80295645867>:1: FutureWarning: The default value of num
eric_only in DataFrame.corr is deprecated. In a future version, it will de
fault to False. Select only valid columns or specify the value of numeric_
only to silence this warning.
  correlation = df.corr()
```



In [36]:

```
print(df.columns)
```

```
Index(['species', 'culmen_length_mm', 'culmen_depth_mm', 'flipper_length_m
m',
       'body_mass_g', 'island_Dream', 'island_Torgersen', 'sex_FEMALE',
       'sex_MALE'],
      dtype='object')
```

In [30]:

```
X = df.drop("species", axis=1)
y = df["species"]
print("Data split into dependent and independent variables!")
```

```
Data split into dependent and independent variables!
```

In [31]:

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
print("Data scaled!")
```

```
Data scaled!
```

In [32]:

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=
print("Data split into training and testing sets!")
```

```
Data split into training and testing sets!
```

In [26]:

```
print("X_train shape:", X_train.shape)
print("y_train shape:", y_train.shape)
print("X_test shape:", X_test.shape)
print("y_test shape:", y_test.shape)
```

```
X_train shape: (275, 8)
y_train shape: (275,)
X_test shape: (69, 8)
y_test shape: (69,)
```