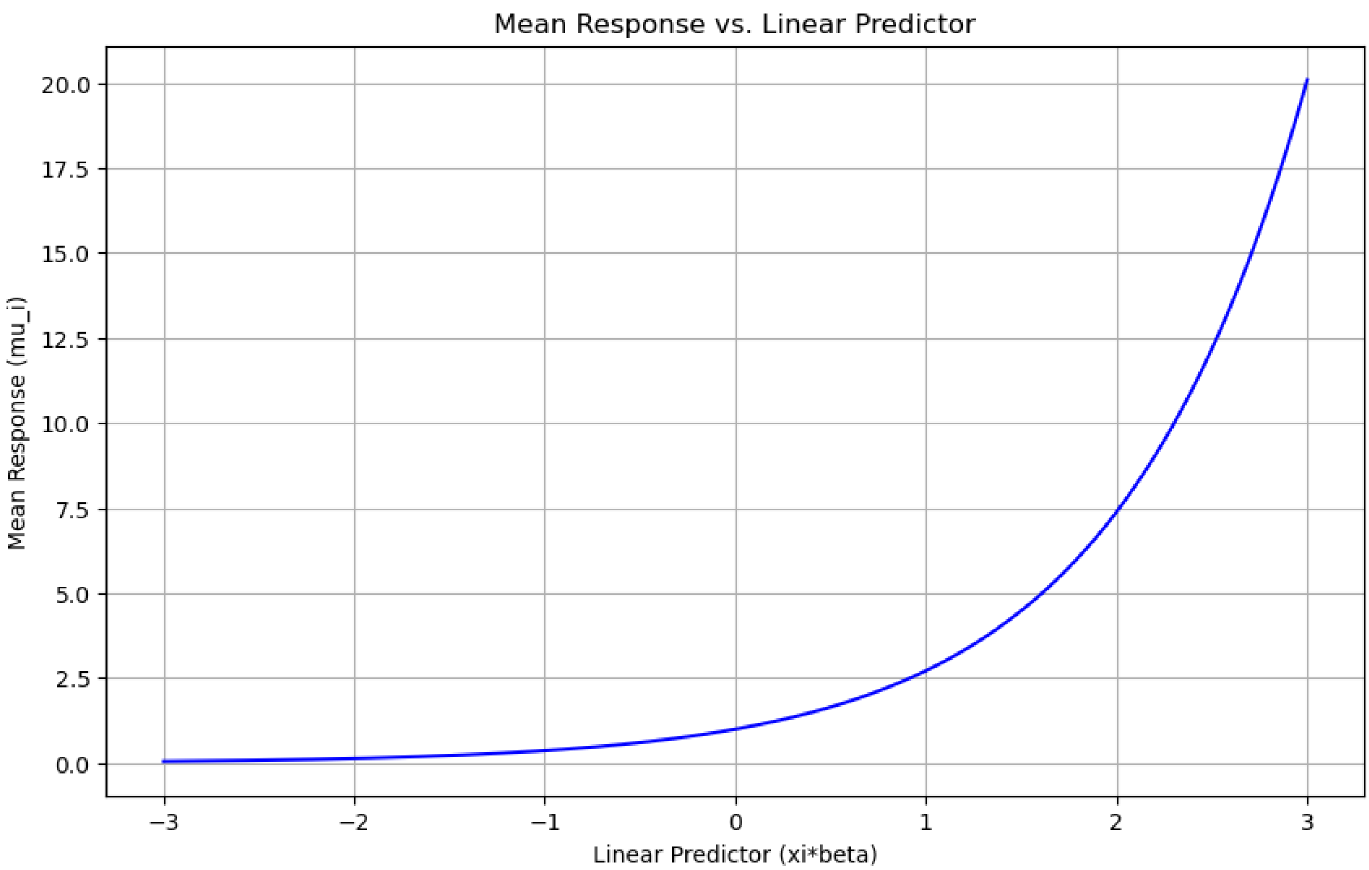
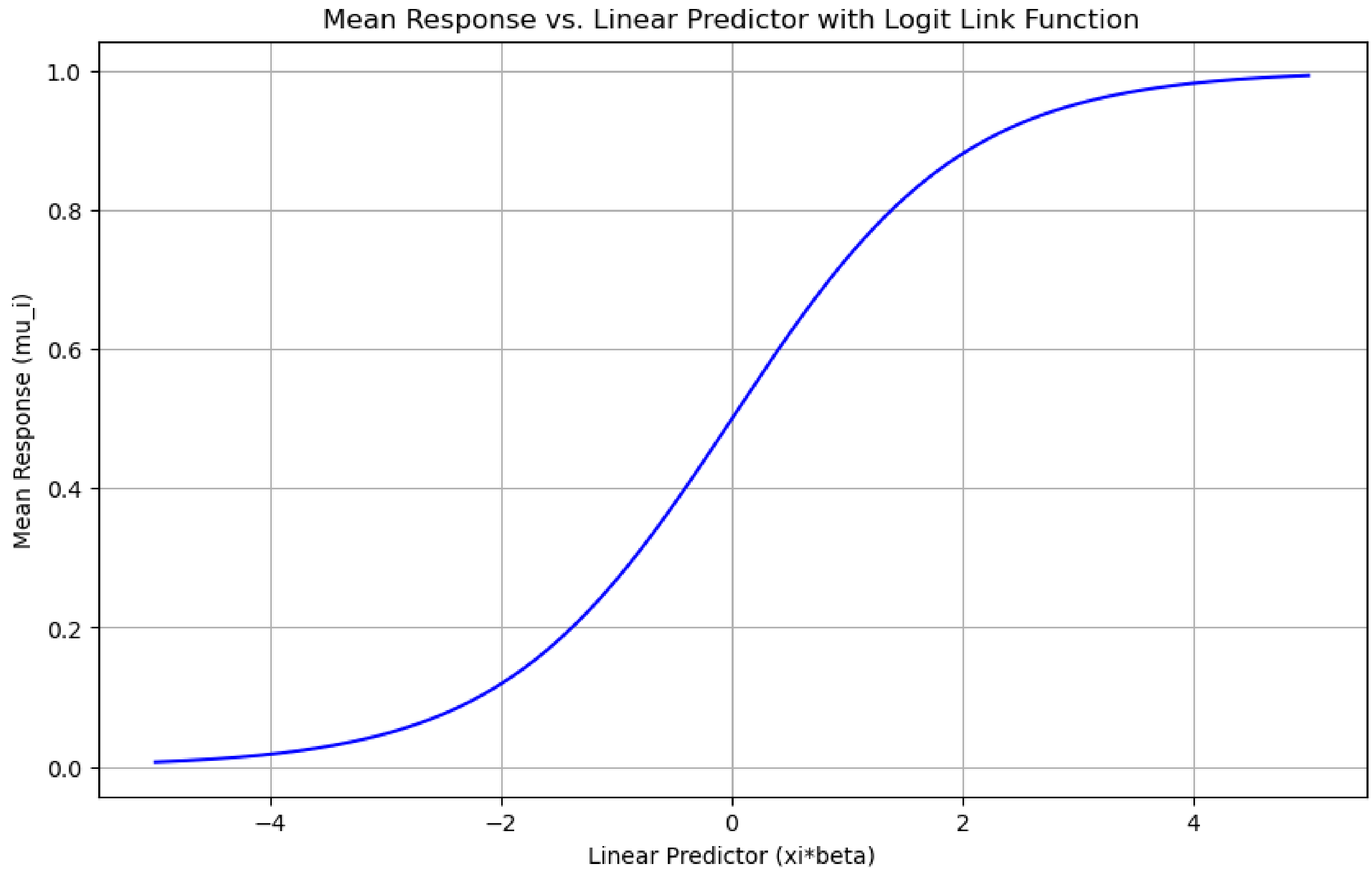
In [1]:
```python
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```python
# 1

# Given range for the linear predictor xi*beta
linear_predictor = np.arange(-3, 3.01, 0.01)

# Since log(mu_i) = xi*beta, we take the exponential to find mu_i
mean_response = np.exp(linear_predictor)

# Plotting the mean response against the linear predictor
plt.figure(figsize=(10, 6))
plt.plot(linear_predictor, mean_response, color='blue')
plt.title('Mean Response vs. Linear Predictor')
plt.xlabel('Linear Predictor (xi*beta)')
plt.ylabel('Mean Response (mu_i)')
plt.grid(True)
plt.show()
```

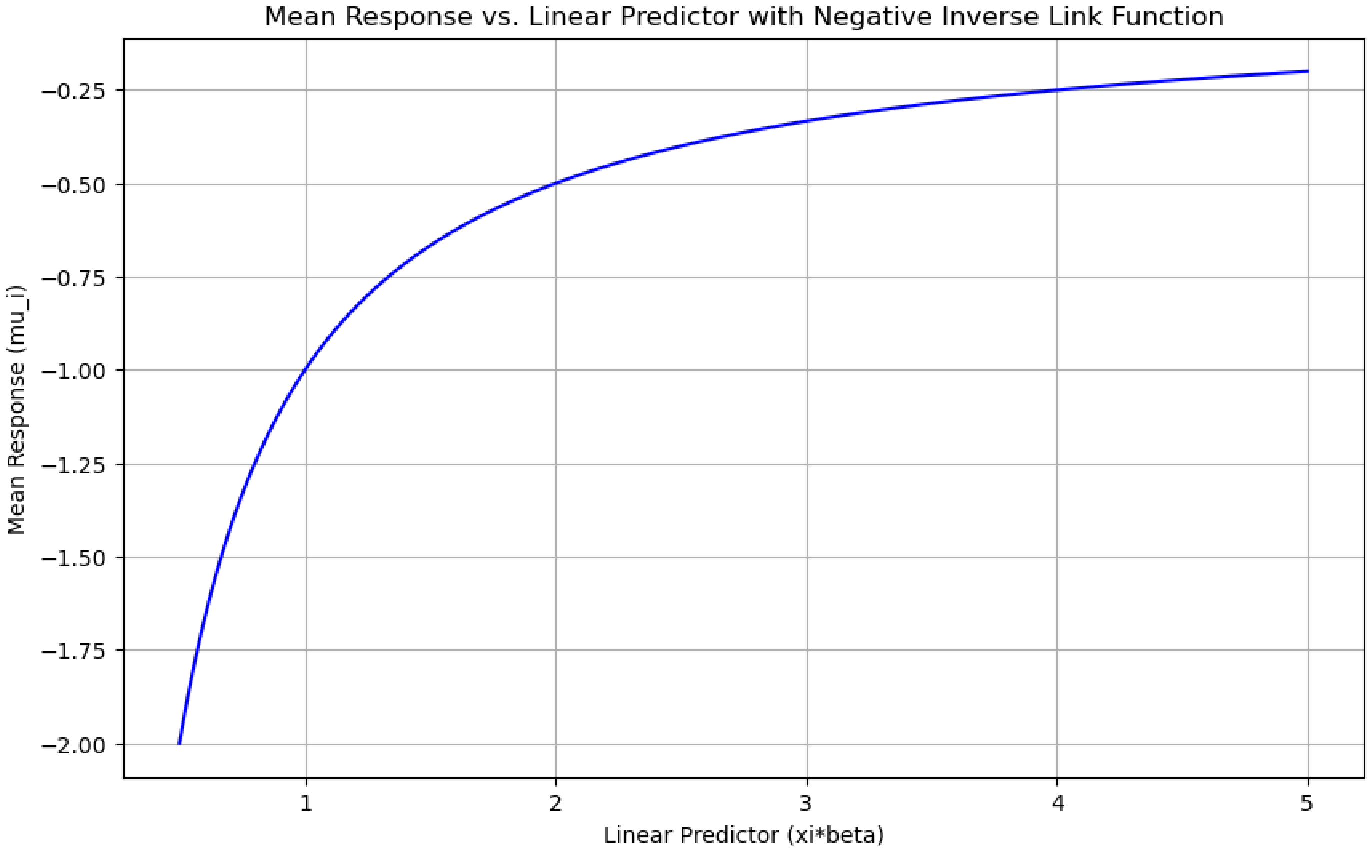


In [3]:
```python
# Given range for the linear predictor xi*beta for the new case
linear_predictor_logit = np.arange(-5, 5.01, 0.01)

# Since log(mu_i / (1 - mu_i)) = xi*beta, we solve for mu_i
# This involves the logistic function, which is the inverse of the logit function
mean_response_logit = np.exp(linear_predictor_logit) / (1 + np.exp(linear_predictor_logi

# Plotting the mean response against the linear predictor for the logit link function
plt.figure(figsize=(10, 6))
plt.plot(linear_predictor_logit, mean_response_logit, color='blue')
plt.title('Mean Response vs. Linear Predictor with Logit Link Function')
plt.xlabel('Linear Predictor (xi*beta)')
plt.ylabel('Mean Response (mu_i)')
plt.grid(True)
plt.show()
```



In [4]:

```
# Given range for the linear predictor xi*beta for the negative inverse link function
linear_predictor_negative_inverse = np.arange(0.5, 5.01, 0.01)

# Since -mu_i^(-1) = xi*beta, we solve for mu_i
# This involves inverting the negative of the linear predictor
mean_response_negative_inverse = -1 / linear_predictor_negative_inverse

# Plotting the mean response against the linear predictor for the negative inverse link
plt.figure(figsize=(10, 6))
plt.plot(linear_predictor_negative_inverse, mean_response_negative_inverse, color='blue'
plt.title('Mean Response vs. Linear Predictor with Negative Inverse Link Function')
plt.xlabel('Linear Predictor (xi*beta)')
plt.ylabel('Mean Response (mu_i)')
plt.grid(True)
plt.show()
```


Mean Response vs. Linear Predictor with Negative Inverse Link Function
Mean Response (mu_i)
−0.25
−0.50
−0.75
−1.00
−1.25
−1.50
−1.75
−2.00
1
2
3
4
5
Linear Predictor (xi*beta)

In [4]: