```python
import numpy as np
import matplotlib.pyplot as plt

def StandardTemp(h):
    """Returns Temperature in Rankine based on height in ft"""
    T_SL = 518.7  # R
    theta = 1 - 6.875e-6 * h
    T = T_SL * theta
    return T

def StandardDensity(h):
    """Returns density in slugs/ft^3"""
    rho_SL = 2.377e-3  # slugs/ft^3
    delta = (1 - 6.875e-6 * h) ** 5.2561  # pressure ratio
    theta = 1 - 6.875e-6 * h  # temperature ratio
    sigma = delta / theta  # density ratio
    rho = sigma * rho_SL
    return rho

def StallSpeed(W, rho, CL_max, S):
    """Returns the stall speed in ft/s"""
    return np.sqrt((2 * W) / (rho * S * CL_max))

# Weight values and labels
W_values = [8864, 4864, 8212.24, 4212.24]  # lb
weight_labels = ["Full Fuel + Payload", "Full Fuel + No Payload", "Reserves + Payload",
"Reserves + No Payload"]
CL_max_values = [1.6, 1.73, 2.0]  # Clean, Takeoff, Landing
config_labels = ["No Flaps", "Takeoff Flaps", "Landing Flaps"]
S = 200  # Assumed wing area in ft^2

# Altitude range
h_values = np.linspace(0, 10000, 100)

# Convert stall speeds from ft/s to knots (1 ft/s = 0.592484 knots)
ft_to_knots = 0.592484

# Create individual plots
for i, (W, label) in enumerate(zip(W_values, weight_labels)):
    plt.figure(figsize=(8, 6))

    for CL_max, config in zip(CL_max_values, config_labels):
        stall_speeds = [StallSpeed(W, StandardDensity(h), CL_max, S) * ft_to_knots for h in
h_values]
        plt.plot(h_values, stall_speeds, label=config)

    # Add 100-knot reference line
    plt.axhline(y=100, color='r', linestyle='--', label='100 knots RFP Requirement')
    plt.title(f'Stall Speed vs Altitude ({label})', fontsize = 14)
    plt.xlabel('Altitude (ft)', fontsize = 12)
    plt.ylabel('Stall Speed (knots)', fontsize = 12)
    plt.legend()
    plt.grid()
    plt.show()
```