

User Login Activity Analyzer

This Python project defines a function, `analyze_logins()`, that evaluates user login activity by comparing the number of logins on a given day to the user's average daily logins. The function calculates a **login ratio** and displays messages about the user's activity. Additionally, a **conditional alert** triggers if the login activity exceeds a specified threshold, helping identify unusually high login behavior, which could indicate suspicious activity.

```
# Define a function named `analyze_logins()` that takes in three parameters,
# → `username`, `current_day_logins`, and `average_day_logins`

def analyze_logins(username, current_day_logins, average_day_logins):

    # Display a message about how many login attempts the user has made that day
    print("Current day login total for", username, "is", current_day_logins)

    # Display a message about average number of login attempts the user has
    # → made that day
    print("Average logins per day for", username, "is", average_day_logins)

    # Calculate the ratio of the logins made on the current day to the logins
    # → made on an average day, storing in a variable named `login_ratio`
    login_ratio = current_day_logins / average_day_logins

    # Return the ratio
    return login_ratio

# Call `analyze_logins()` and store the output in a variable named
# → `login_analysis`

login_analysis = analyze_logins("ejones", 9, 3)

# Conditional statement that displays an alert about the login activity if it's
# → more than normal

if login_analysis >= 3:
    print("Alert! This account has more login activity than normal.")
```

```
Current day login total for ejones is 9
Average logins per day for ejones is 3
Alert! This account has more login activity than normal.
```

The project demonstrates the use of functions, parameters, arithmetic operations, and conditional statements in Python, while providing practical insight into monitoring user behavior.