**Lab Exercise 3- Developing Features Faster with GitHub Copilot**

**Objective:**

Learn how to use GitHub Copilot to speed up feature development, including code generation, completion, and refactoring.

**Prerequisites:**

* GitHub Copilot enabled in **VS Code**.
* Basic knowledge of programming.

**Exercise Steps**

**1. Using Copilot for Code Completion**

**Task: Implement a Function from a Comment**

1. Open a Python or JavaScript file in VS Code.
2. Type the following comment and **press Enter** to let Copilot suggest code:

# Function to check if a number is prime

1. Accept Copilot’s suggestion by pressing **Tab**.

**Expected Outcome:**

Copilot will generate a function that checks if a number is prime.

**2. Generating a Feature with Copilot**

**Task: Create a User Authentication System**

1. Type the following comment in a new Python file:

# Function to register a user with a username and password

1. Press **Enter** and let Copilot suggest code.
2. Repeat for:

# Function to log in a user by checking credentials

**Expected Outcome:**

Copilot will generate basic user registration and login functions.

**3. Refactoring Code for Better Performance**

**Task: Optimize an Inefficient Function**

1. Copy the following Python code:

def find\_max(numbers):

max\_num = numbers[0]

for num in numbers:

if num > max\_num:

max\_num = num

return max\_num

1. Type the comment:

# Optimize this function for better performance

1. Let Copilot suggest improvements.

**Expected Outcome:**

Copilot will replace the loop with Python’s built-in max() function.

**4. Using Copilot for Unit Test Generation**

**Task: Generate Tests for a Function**

1. Write a simple function:

def add(a, b):

return a + b

1. Type the comment:

# Generate unit tests for the add function

1. Let Copilot generate test cases using unittest or pytest.

**Expected Outcome:**

Copilot will create test cases to verify the correctness of the function.

**5. Exploring Copilot’s Context Awareness**

**Task: Continue Writing Code Automatically**

1. Start writing a class for a **To-Do List App**:

class TodoList:

def \_\_init\_\_(self):

self.tasks = []

1. Press **Enter** and let Copilot predict the next methods.

**Expected Outcome:**

Copilot will suggest methods like add\_task, remove\_task, and view\_tasks.

**Conclusion**

* **Code faster** using Copilot’s autocomplete.
* **Generate features** with natural language comments.
* **Refactor code** for better performance.
* **Write unit tests** quickly.

By following this exercise, you can speed up feature development using GitHub Copilot!