**Technical Design Document Exercise 6**

**Name:** Logan Flynn

**Date Created:** 10/12/2025

**Program Description:** This program checks whether user input for a phone number, social security number, and zip code is Valid. Using python’s re module to test patterns. The program has three validation functions and a main function that asks for user input, runs the checks, and shows if each input is valid. A test function runs lists of valid and invalid examples to make sure it works correctly.

**Functions used in the Program (list in order as they are called):**

### 

### **1. Function Name: ​​**valid\_phone

**Description:** Checks if the phone number entered by the user matches the U.S. phone number format.

**Parameters:** The phone number entered by the user.

**Variables:**

1. pattern

**Logical Steps:**

1. Create a regex pattern that accepts optional +1/1 parentheses, dashes, dots, or spaces.
2. Remove spaces from the start and the end of the input.
3. Match the input with the pattern and return true if valid.

**Returns:** True or false.

**2. Function name:** valid\_ssn

**Description:** Checks if the SSN follows ###-##-#### format and does not include invalid area/group/serial numbers.

**Parameters:** The SSN entered by the user.

**Variables:**

1. pattern

**Logical Steps:**

1. Create a regex pattern that matches ###-##-####.
2. Use lookahead rules to reject 000,666, or 9xx areas, 00 groups, or 0000 serials.
3. Strip whitespaces and check for a match.

**Returns:** True or False

**3. Function Name:** valid\_zip

**Description:** Checks if the zip code is 5 digits.

**Parameters:** The zip code entered by the user.

**Variables:**

1. pattern

**Logical Steps:**

1. Create a regex pattern that allows 5 digits with optional -#### ending.
2. Strip Whitespace and check for a match.

**Returns:** True or False.

**4. Function Name:** main

**Description:** Controls the flow of the program by asking the user for input, validating it, and displaying results.

**Parameters:** None

**Variables:**

1. Phone
2. Ssn
3. zipcode

**Logical Steps:**

1. Create lists of valid and invalid examples for phone, SSN, and zip.
2. Use Loops to check each example with the correct validator function.
3. Print of each test passed or failed and count totals.

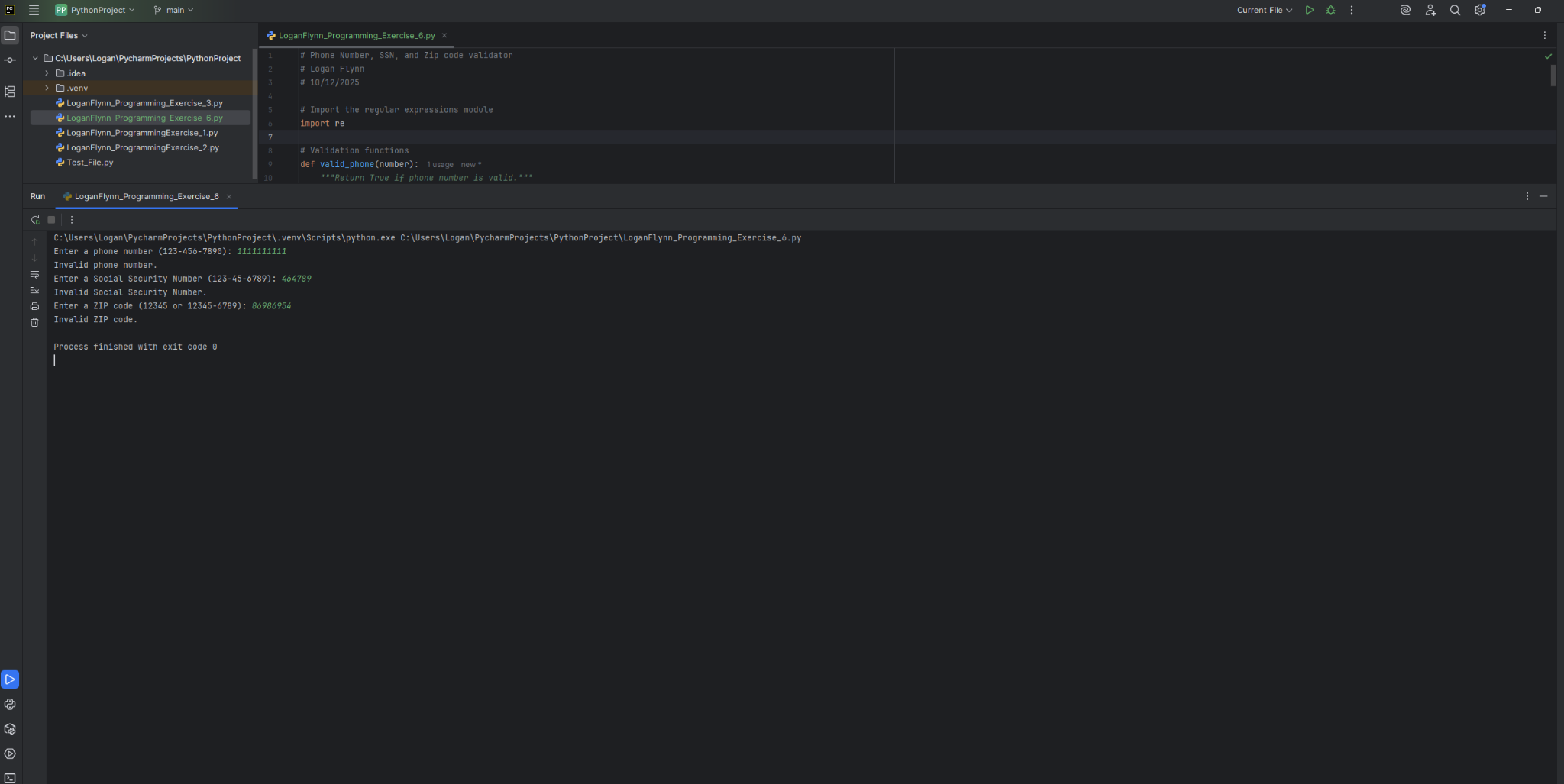
**Returns:** None

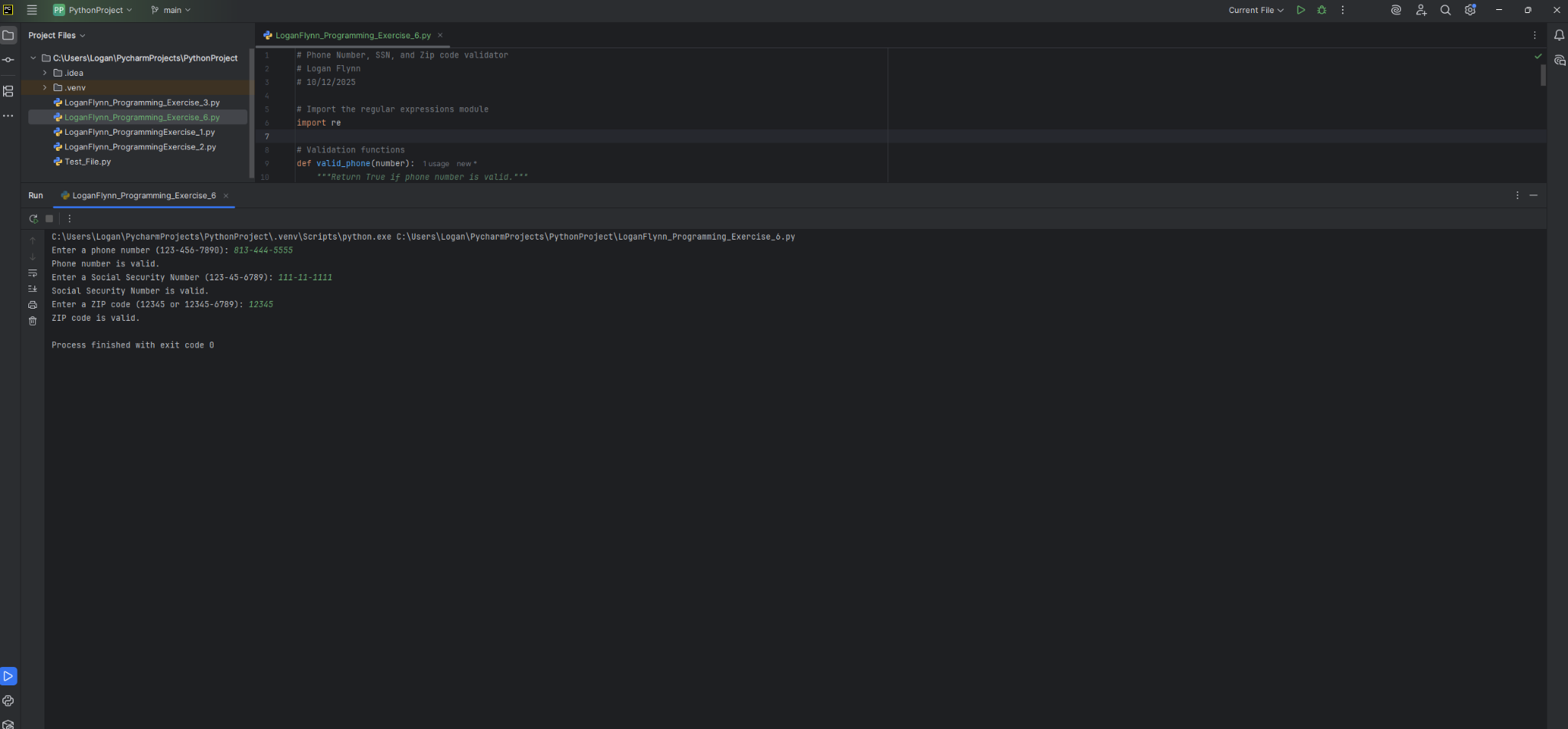
**Constants:** None

**Link to your repository:**

[loganflynnn (Logan Flynn)](https://github.com/loganflynnn)

**Output Screenshot:**



****