Report

Decimal to Binary and Binary to Decimal Conversion

This report provides an overview of a Python program that serves as a versatile decimal-to-binary and binary-to-decimal converter. The program allows users to interactively enter decimal numbers to convert them to binary and vice versa. The user can convert back and forth until they choose to quit the program.

- **decimal_to_binary:** This function takes a decimal number as input and converts it to its binary representation. It utilizes the `bin()` function to perform the conversion and removes the '0b' prefix from the result.
- **binary_to_decimal:** This function converts a binary string to its decimal representation. It uses the `int()` function with base 2 to perform the conversion and handles any `ValueError` exceptions that may occur if the input is not a valid binary string.

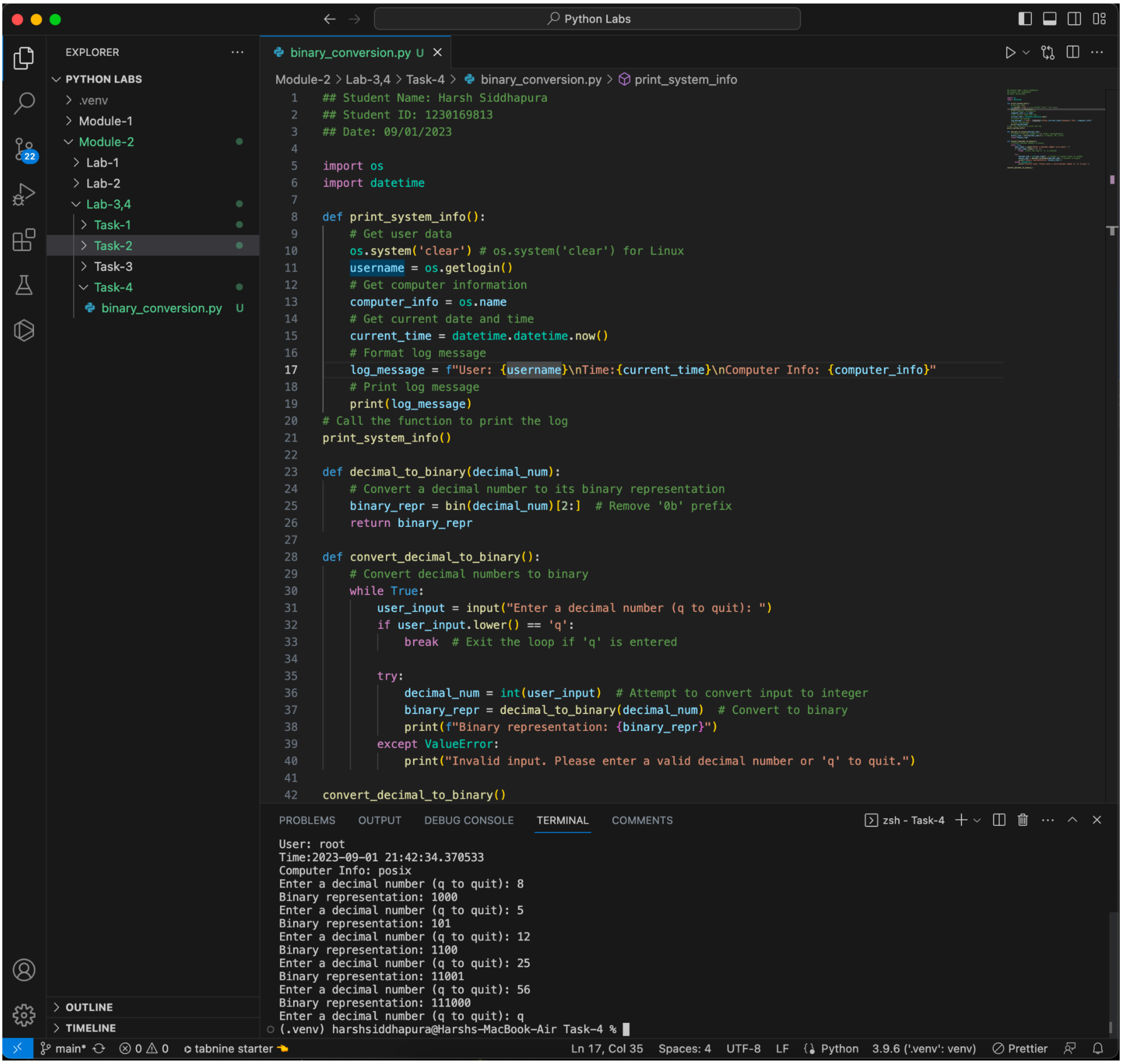
The core functionality of the program is encapsulated within the `convert_decimal_to_binary` function, which handles the interactive conversion process.

- **User Input:** The program uses an infinite loop to repeatedly prompt the user for input. The user can enter either a decimal number to convert to binary or a binary number to convert to decimal. They can also choose to quit the program by entering 'q.'
- **Decimal to Binary:** If the user enters a decimal number, the program attempts to convert it to an integer. It checks if the input is non-negative and converts it to binary using the `decimal_to_binary` function. The binary representation is displayed.
- **Binary to Decimal:** If the user enters a binary number, the program checks if it consists of only '0' and '1' characters. If it's a valid binary string, the program converts it to a decimal number using the `binary_to_decimal` function. The decimal representation is displayed.
- **Input Validation:** The program provides robust input validation, handling various scenarios, such as invalid inputs or negative decimal numbers.

This program is a valuable tool for various applications, including:

- Educational purposes, teaching the binary numbering system and conversions.
- Quick conversions between decimal and binary for programmers and students.
- Debugging and verification of binary data in computer science and engineering.

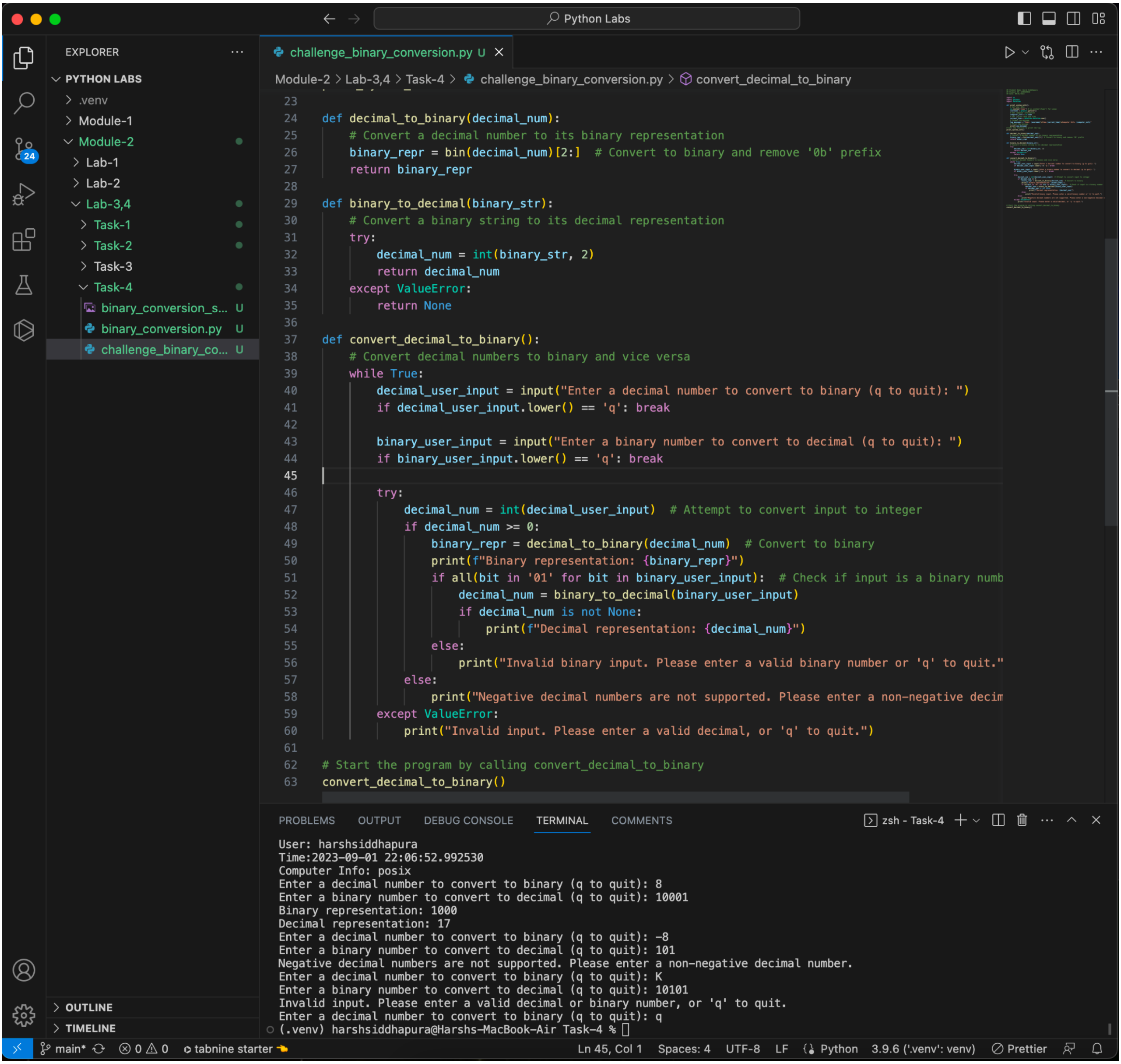
In conclusion, the Python program presented in this report offers a user-friendly and versatile decimal-to-binary and binary-to-decimal converter. Its interactive nature allows users to explore conversions conveniently and learn about the relationships between decimal and binary representations. With input validation and a user-friendly interface, it provides a practical tool for a wide range of users interested in binary conversions.



```python
## Student Name: Harsh Siddhapura
## Student ID: 1230169813
## Date: 09/01/2023

import os
import datetime

def print_system_info():
    # Get user data
    os.system('clear') # os.system('clear') for Linux
    username = os.getlogin()
    # Get computer information
    computer_info = os.name
    # Get current date and time
    current_time = datetime.datetime.now()
    # Format log message
    log_message = f"User: {username}\nTime:{current_time}\nComputer Info: {computer_info}"
    # Print log message
    print(log_message)
# Call the function to print the log
print_system_info()

def decimal_to_binary(decimal_num):
    # Convert a decimal number to its binary representation
    binary_repr = bin(decimal_num)[2:]  # Remove '0b' prefix
    return binary_repr

def convert_decimal_to_binary():
    # Convert decimal numbers to binary
    while True:
        user_input = input("Enter a decimal number (q to quit): ")
        if user_input.lower() == 'q':
            break  # Exit the loop if 'q' is entered

        try:
            decimal_num = int(user_input)  # Attempt to convert input to integer
            binary_repr = decimal_to_binary(decimal_num)  # Convert to binary
            print(f"Binary representation: {binary_repr}")
        except ValueError:
            print("Invalid input. Please enter a valid decimal number or 'q' to quit.")

convert_decimal_to_binary()
```

```
User: root
Time:2023-09-01 21:42:34.370533
Computer Info: posix
Enter a decimal number (q to quit): 8
Binary representation: 1000
Enter a decimal number (q to quit): 5
Binary representation: 101
Enter a decimal number (q to quit): 12
Binary representation: 1100
Enter a decimal number (q to quit): 25
Binary representation: 11001
Enter a decimal number (q to quit): 56
Binary representation: 111000
Enter a decimal number (q to quit): q
(.venv) harshsiddhapura@Harshs-MacBook-Air Task-4 %
```

Challenge Code (Optional)



```python

def decimal_to_binary(decimal_num):
    # Convert a decimal number to its binary representation
    binary_repr = bin(decimal_num)[2:]  # Convert to binary and remove '0b' prefix
    return binary_repr

def binary_to_decimal(binary_str):
    # Convert a binary string to its decimal representation
    try:
        decimal_num = int(binary_str, 2)
        return decimal_num
    except ValueError:
        return None

def convert_decimal_to_binary():
    # Convert decimal numbers to binary and vice versa
    while True:
        decimal_user_input = input("Enter a decimal number to convert to binary (q to quit): ")
        if decimal_user_input.lower() == 'q': break

        binary_user_input = input("Enter a binary number to convert to decimal (q to quit): ")
        if binary_user_input.lower() == 'q': break

        try:
            decimal_num = int(decimal_user_input)  # Attempt to convert input to integer
            if decimal_num >= 0:
                binary_repr = decimal_to_binary(decimal_num)  # Convert to binary
                print(f"Binary representation: {binary_repr}")
                if all(bit in '01' for bit in binary_user_input):  # Check if input is a binary numb
                    decimal_num = binary_to_decimal(binary_user_input)
                    if decimal_num is not None:
                        print(f"Decimal representation: {decimal_num}")
                else:
                    print("Invalid binary input. Please enter a valid binary number or 'q' to quit."
            else:
                print("Negative decimal numbers are not supported. Please enter a non-negative decim
        except ValueError:
            print("Invalid input. Please enter a valid decimal, or 'q' to quit.")

# Start the program by calling convert_decimal_to_binary
convert_decimal_to_binary()
```

```
User: harshsiddhapura
Time:2023-09-01 22:06:52.992530
Computer Info: posix
Enter a decimal number to convert to binary (q to quit): 8
Enter a binary number to convert to decimal (q to quit): 10001
Binary representation: 1000
Decimal representation: 17
Enter a decimal number to convert to binary (q to quit): -8
Enter a binary number to convert to decimal (q to quit): 101
Negative decimal numbers are not supported. Please enter a non-negative decimal number.
Enter a decimal number to convert to binary (q to quit): K
Enter a binary number to convert to decimal (q to quit): 10101
Invalid input. Please enter a valid decimal or binary number, or 'q' to quit.
Enter a decimal number to convert to binary (q to quit): q
(.venv) harshsiddhapura@Harshs-MacBook-Air Task-4 % 
```