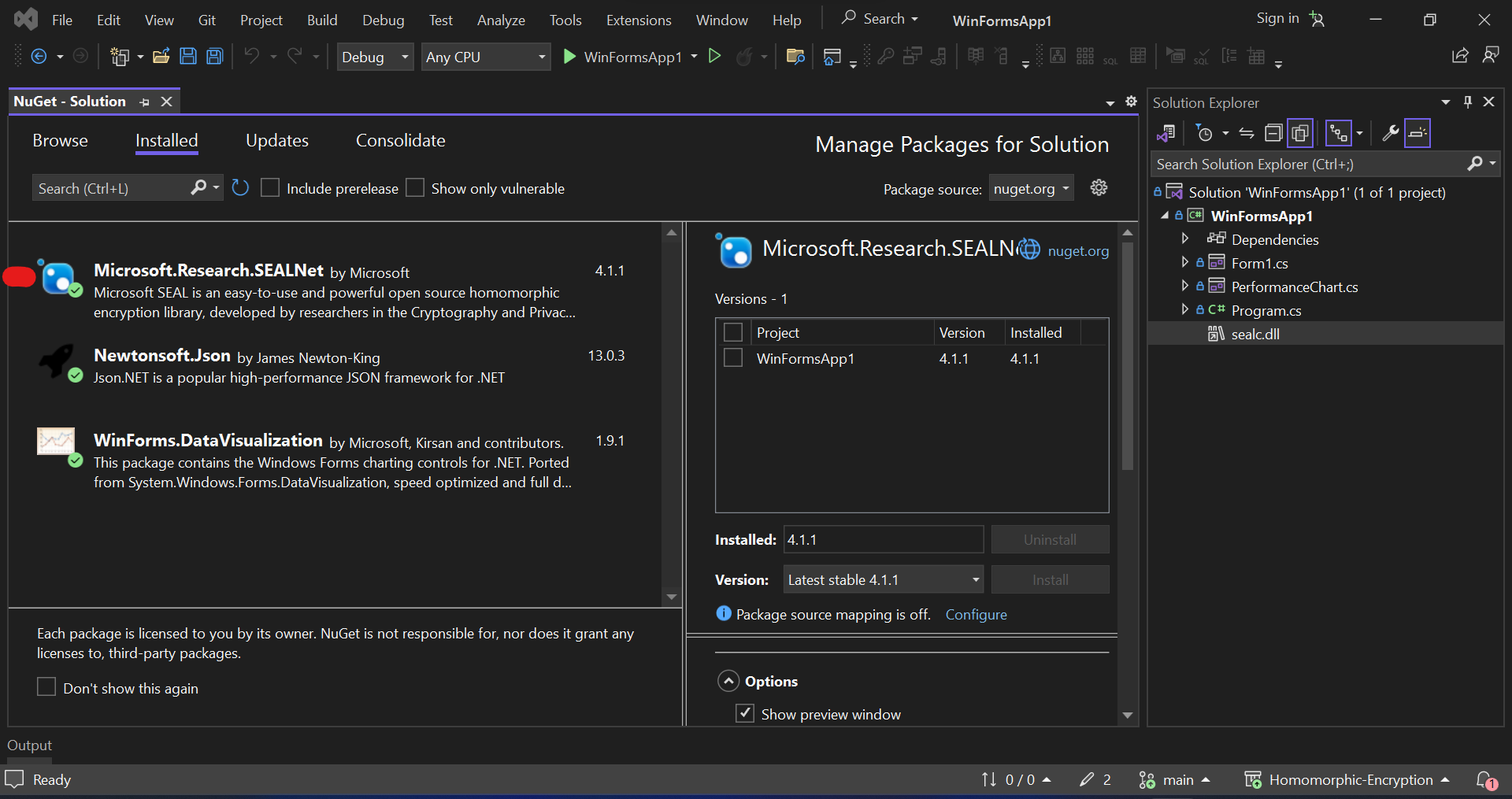
**Development:**

According to description, we need to develop a suitable use case of Homomorphic Encryption using Microsoft Seal library. We selected the ‘**Bank Finance Calculator**’ for this demo which is a good use case of Homomorphic Encryption which can help to improve privacy of data of bank users.

**Step 1: Set Up the Development Environment**

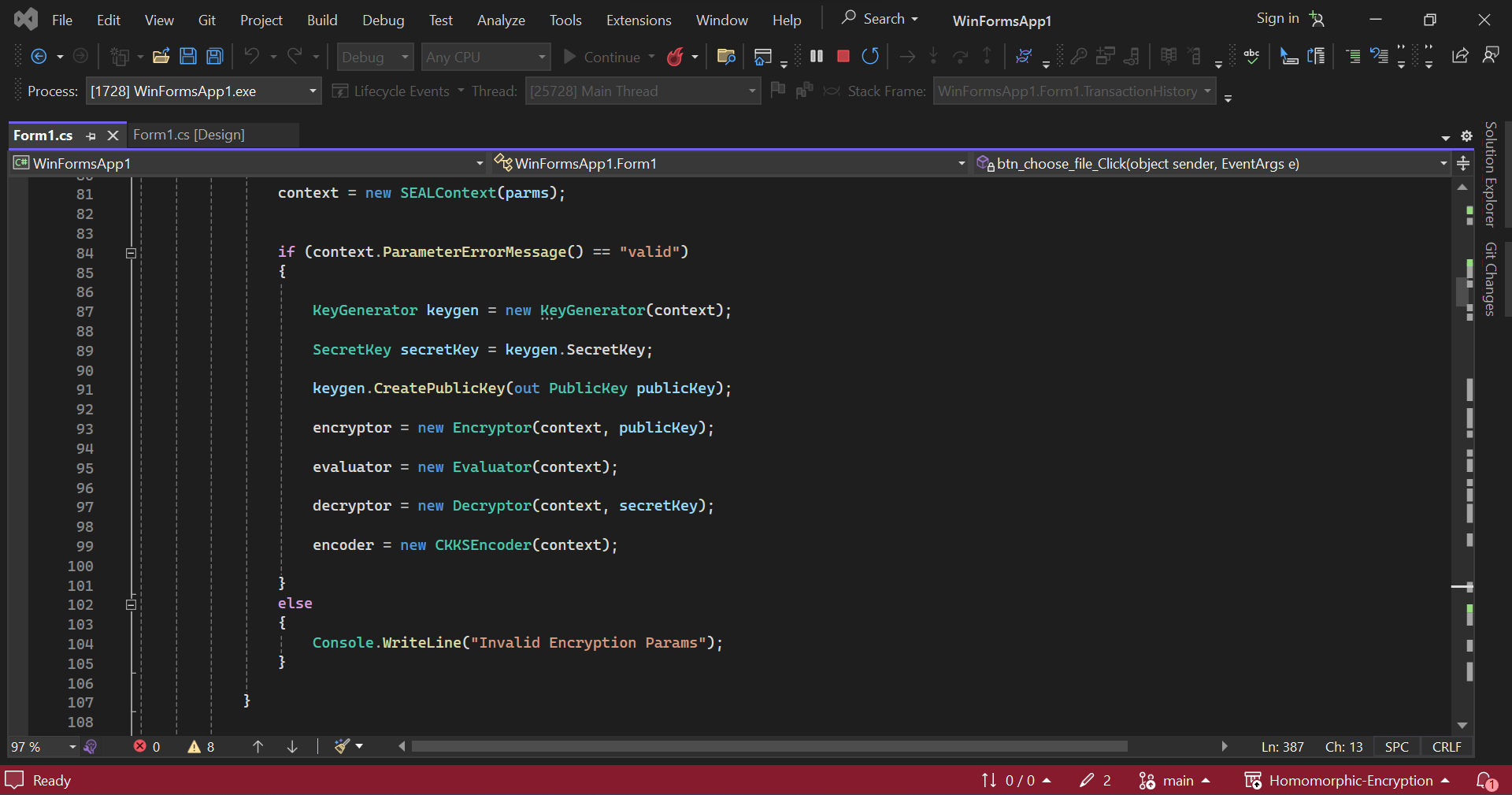
• Install the SEAL Homomorphic encryption library. and tools for homomorphic encryption in your programming environment.



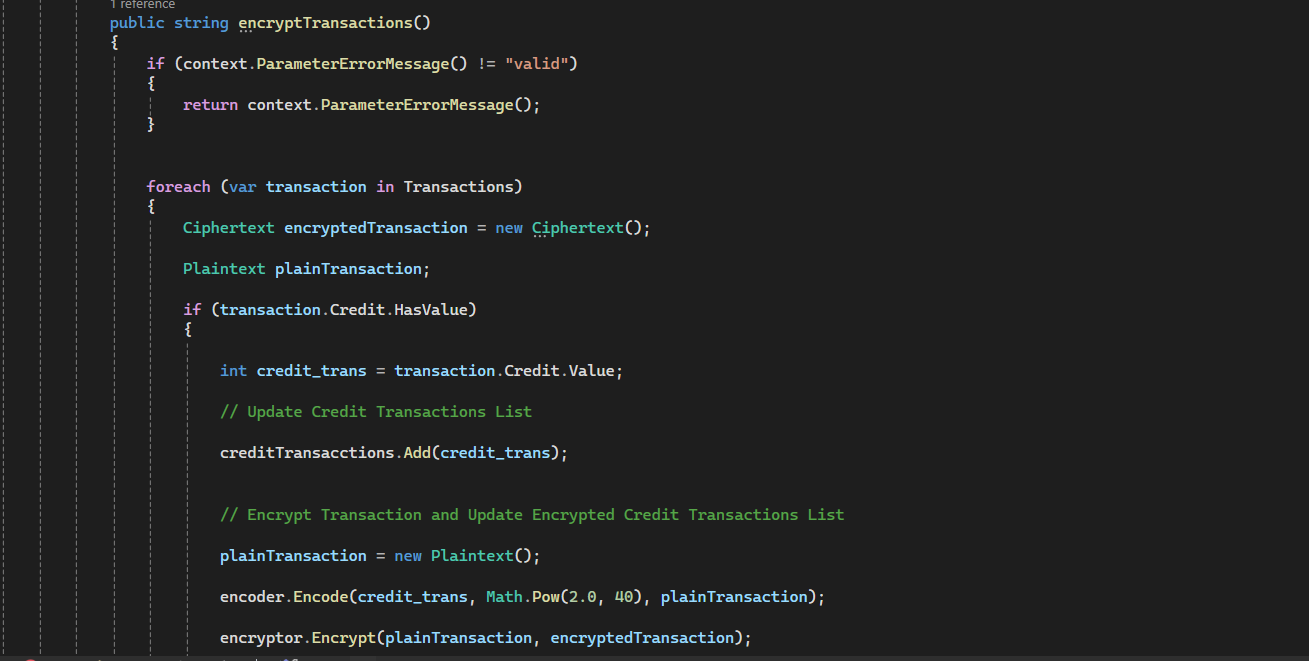
**Step 2: Implement Basic Homomorphic Operations**

• Start with the basics: key generation, encryption, and decryption. Create functions for homomorphic addition and multiplication.

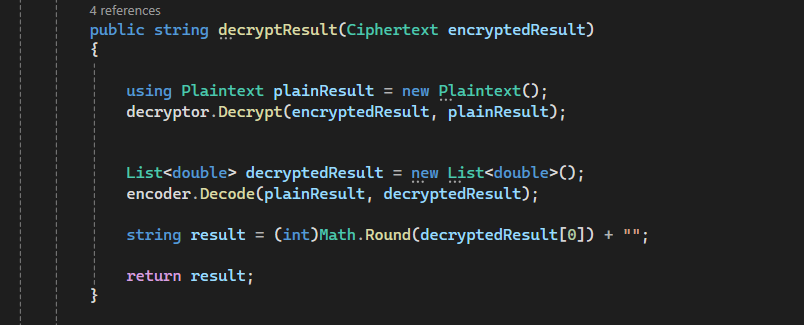
**Key Generation:**

****

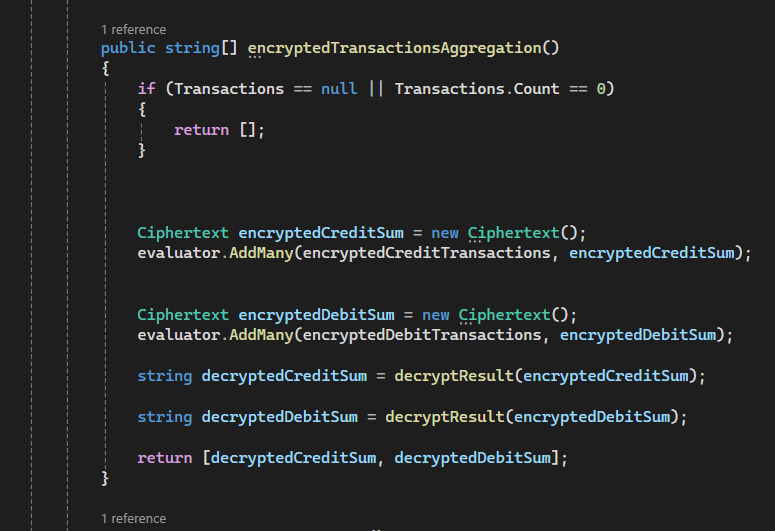
**Encryption:**

****

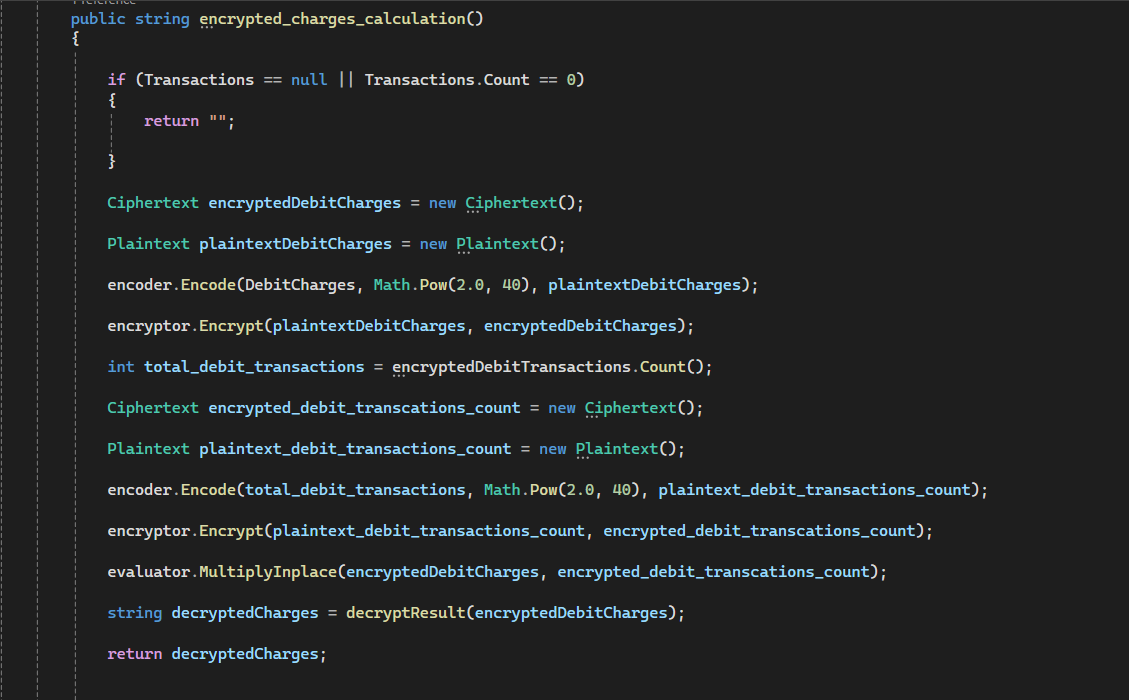
**Decryption:**

****

**Addition:**

****

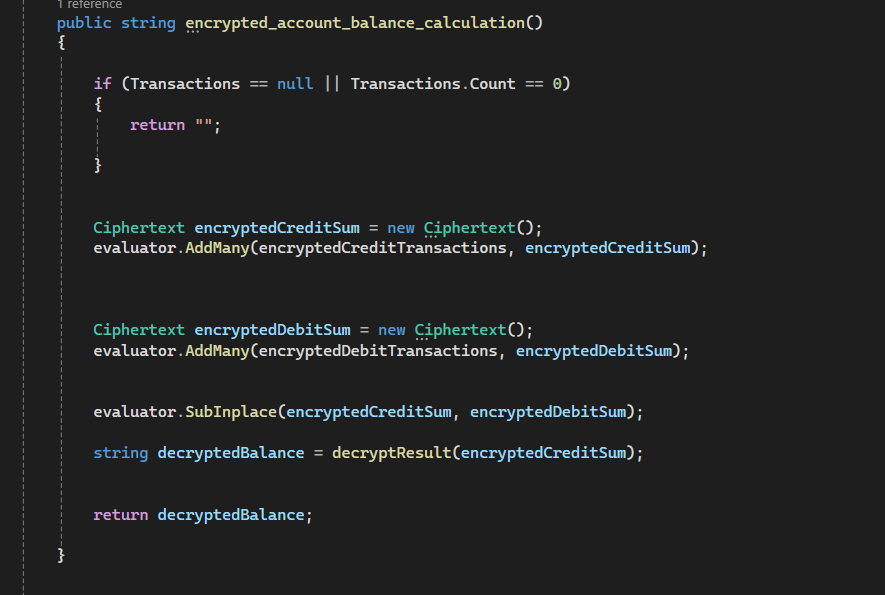
**Multiplication:**

****

**Step 3: Extend Functionality to Include Homomorphic Subtraction**

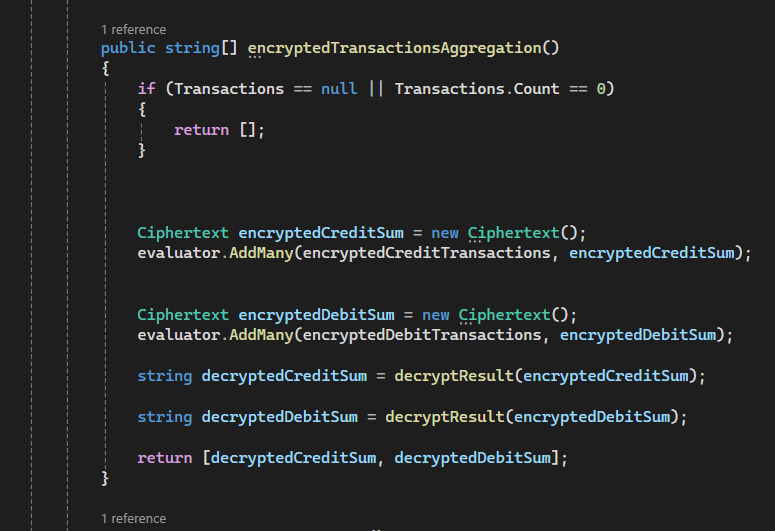
• Implement a function for homomorphic subtraction to showcase a broader range of arithmetic operations.

**Subtraction:**



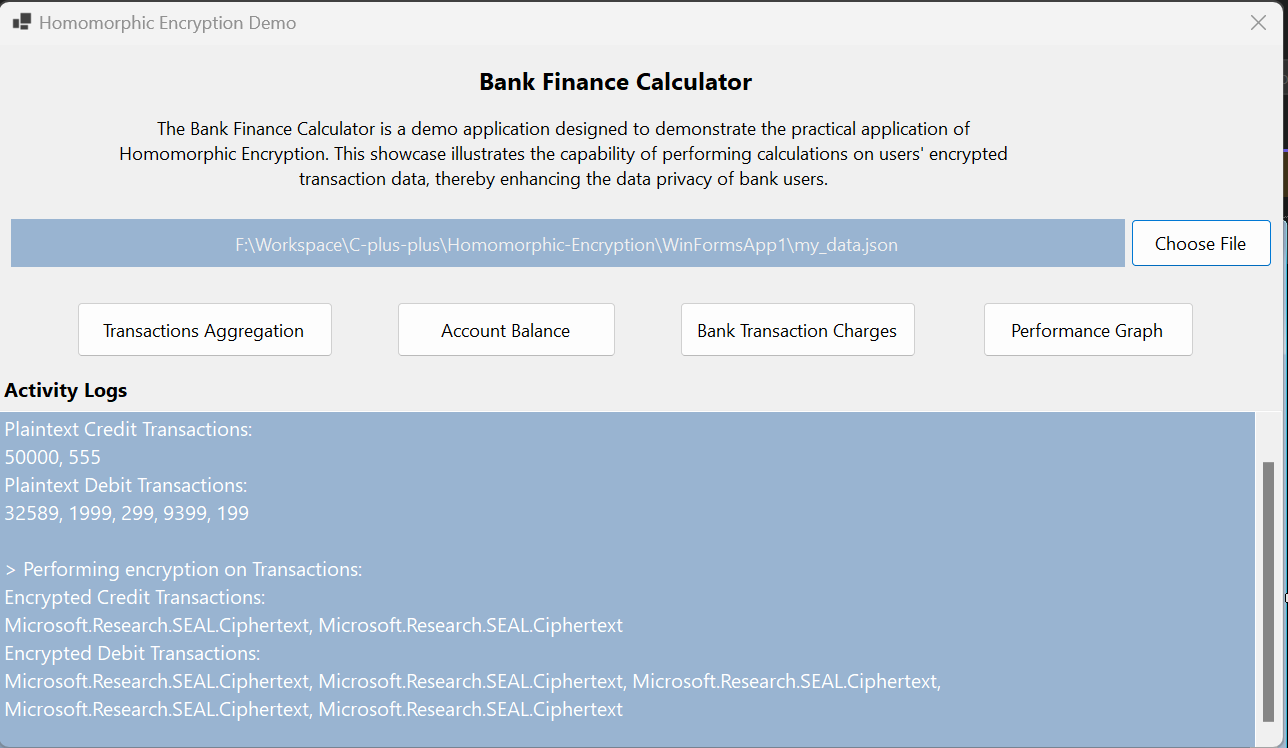
**Step 4: Implement Aggregation Operations**

• Extend your demo to support operations like calculating the sum or average of encrypted data, showcasing homomorphic encryption's usefulness for privacy-preserving data aggregation.

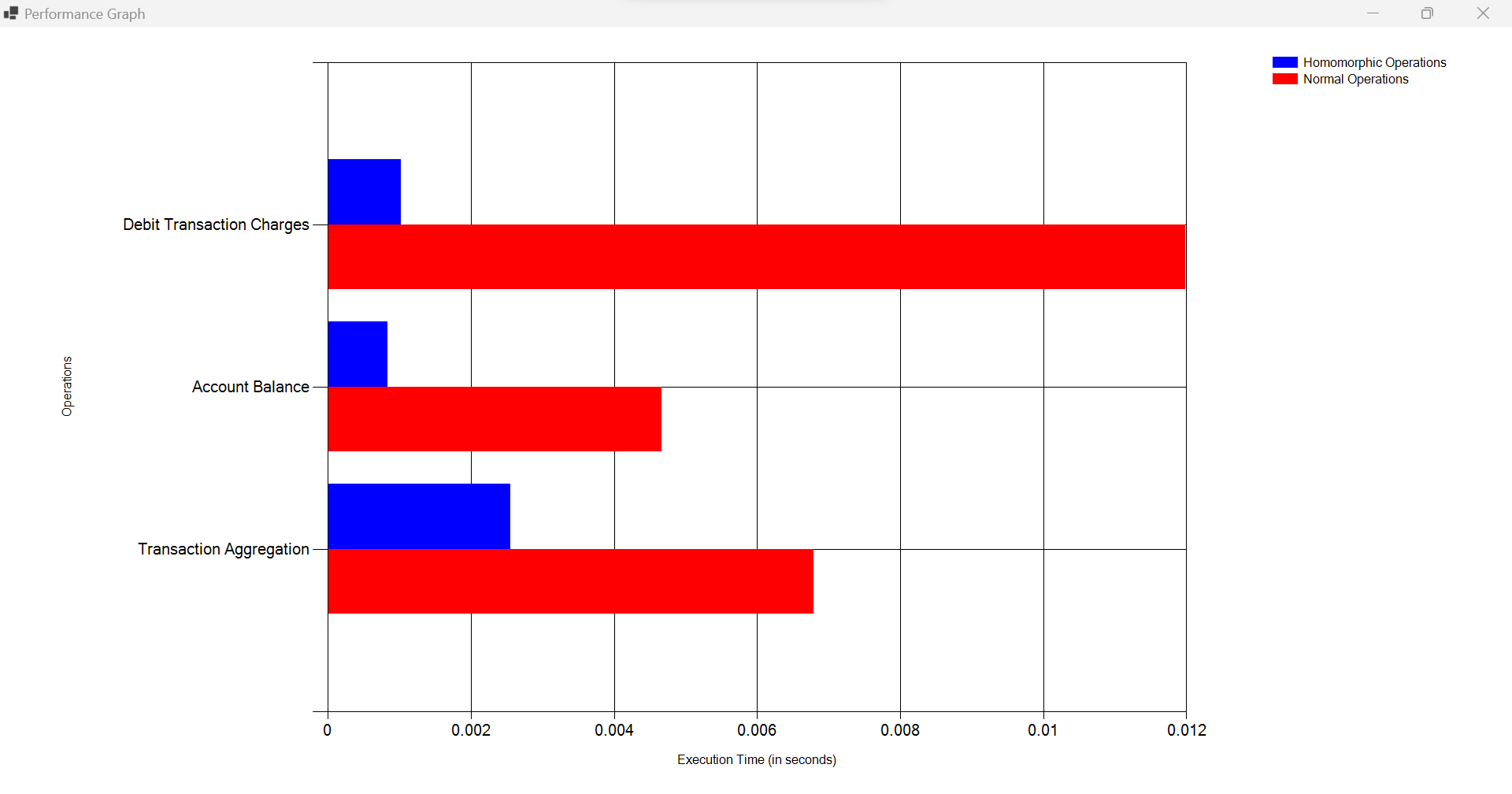
****

**Step 5: Frontend Development**

• Design a user-friendly interface that allows users to input values and view encrypted and decrypted results.



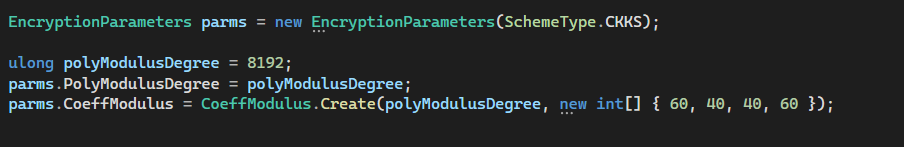
• Incorporate interactive charts or visualizations to enhance user understanding.



**Step 6: Implement Approximation Techniques**

• Explore and implement approximation techniques to balance accuracy and efficiency in your homomorphic encryption operations.

We used CKKS scheme because it gives accurate results as compared to BFV scheme. We used specific encryption parameters to ensure balance accuracy and efficiency.



We used scale of 2^40 for handling large value. Hence, improving accuracy.



**Step 7: Security Analysis**

• Perform a basic security analysis of your homomorphic encryption implementation. Document the security guarantees it provides and highlight any limitations or considerations.

Security Analysis of application is provided in separate document.