Project Description: Combined Encryption Tool

This Java Swing application provides a simple interface for encrypting and decrypting text using two common algorithms:

- **Caesar Cipher:** A basic substitution cipher where each letter in the plaintext is shifted a certain number of positions down the alphabet.
- **AES (Advanced Encryption Standard):** A symmetric block cipher widely used for secure data encryption.

Features

- **User-friendly GUI:** A graphical interface built with Java Swing for easy interaction.
- **Encryption and Decryption:** Encrypt and decrypt text using either the Caesar Cipher or AES algorithm.
- **Caesar Cipher Key Input:** For the Caesar Cipher, the user can specify the shift key.
- **AES Key Generation:** The application generates an AES key for encryption and uses the same key for decryption. *(Note: The generated AES key is displayed in the application for demonstration purposes. In a production environment, secure key management practices are essential.)*
- **Output Display:** Encrypted and decrypted text is displayed in a text area within the application.

How to Use

1. **Clone the Repository:** Clone this GitHub repository to your local machine.
2. **Compile:** Use a Java compiler (JDK) to compile the CombinedEncryptionGUI.java file.
3. **Run:** Execute the compiled class to start the application.
4. **Select Encryption Method:** Choose either "Caesar Cipher" or "AES" from the dropdown menu.
5. **Enter Text:** Type the text you want to encrypt or decrypt in the "Input Text" field.
6. **Enter Caesar Key (if applicable):** If you selected "Caesar Cipher", enter the shift key in the "Caesar Key" field.
7. **Encrypt/Decrypt:** Click the "Encrypt" or "Decrypt" button.
8. **View Output:** The encrypted or decrypted text will be displayed in the text area.

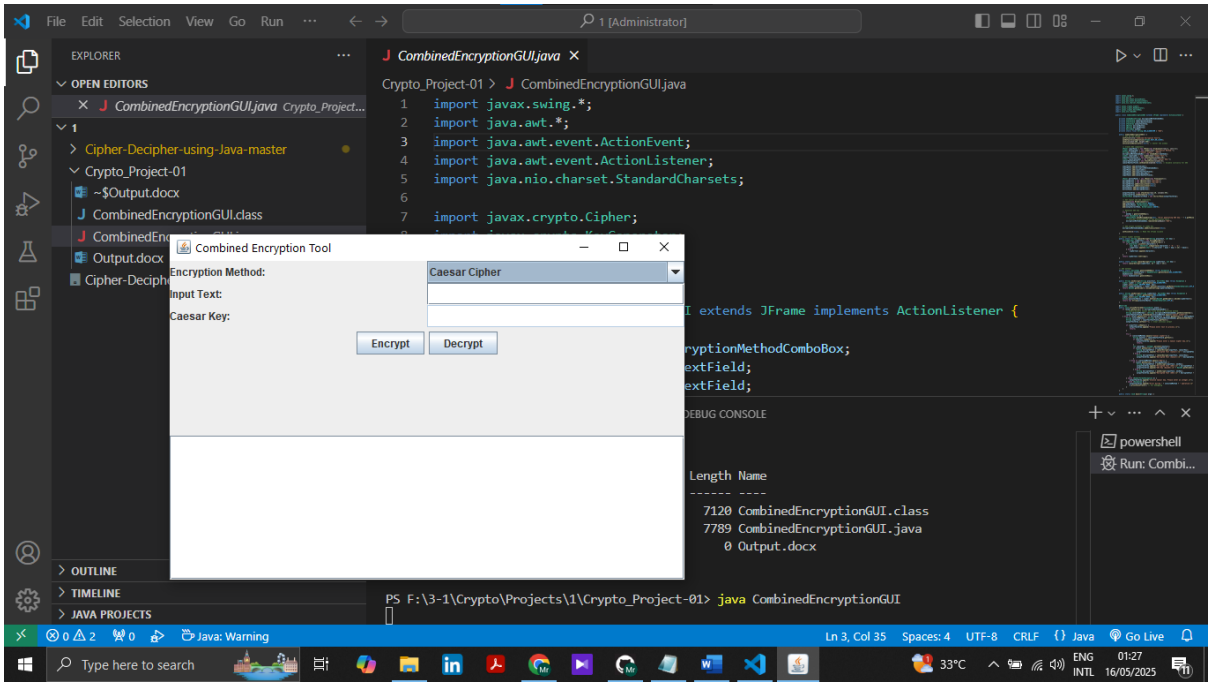
Technical Details

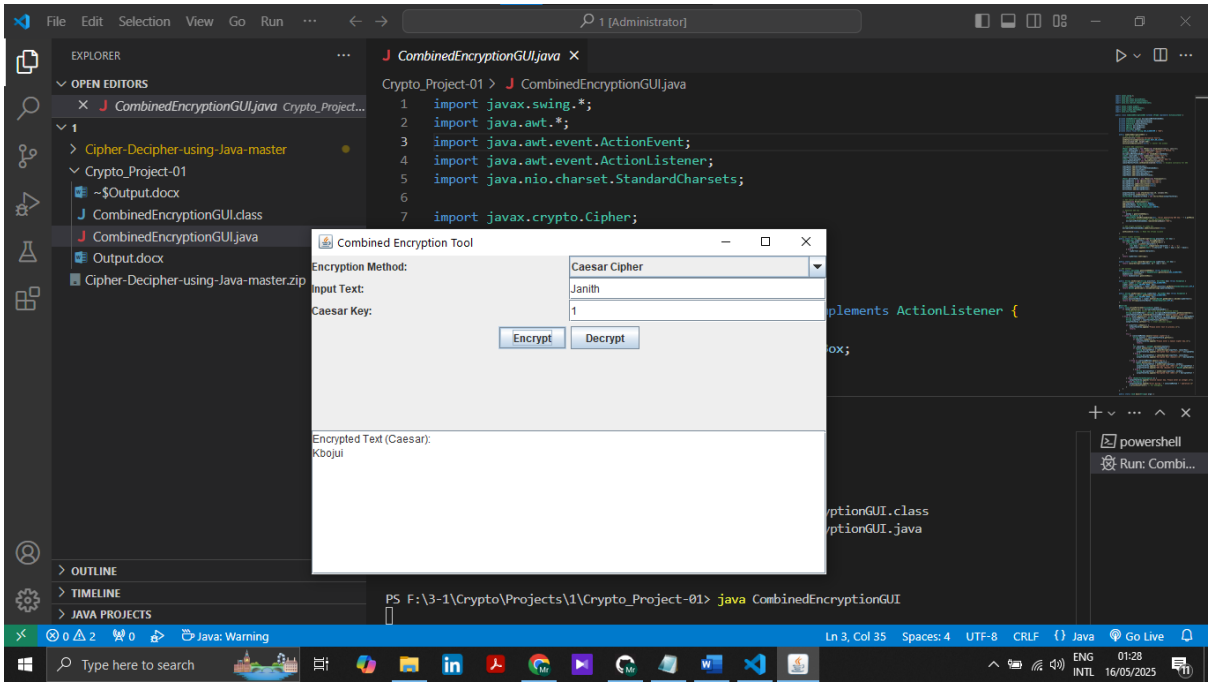
- **Language:** Java
- **GUI Library:** Java Swing
- **Encryption Algorithms:**
    - Caesar Cipher (simple substitution)
    - AES (symmetric block cipher)
- **Java Cryptography Architecture (JCA):** The application uses the JCA for AES encryption and decryption.

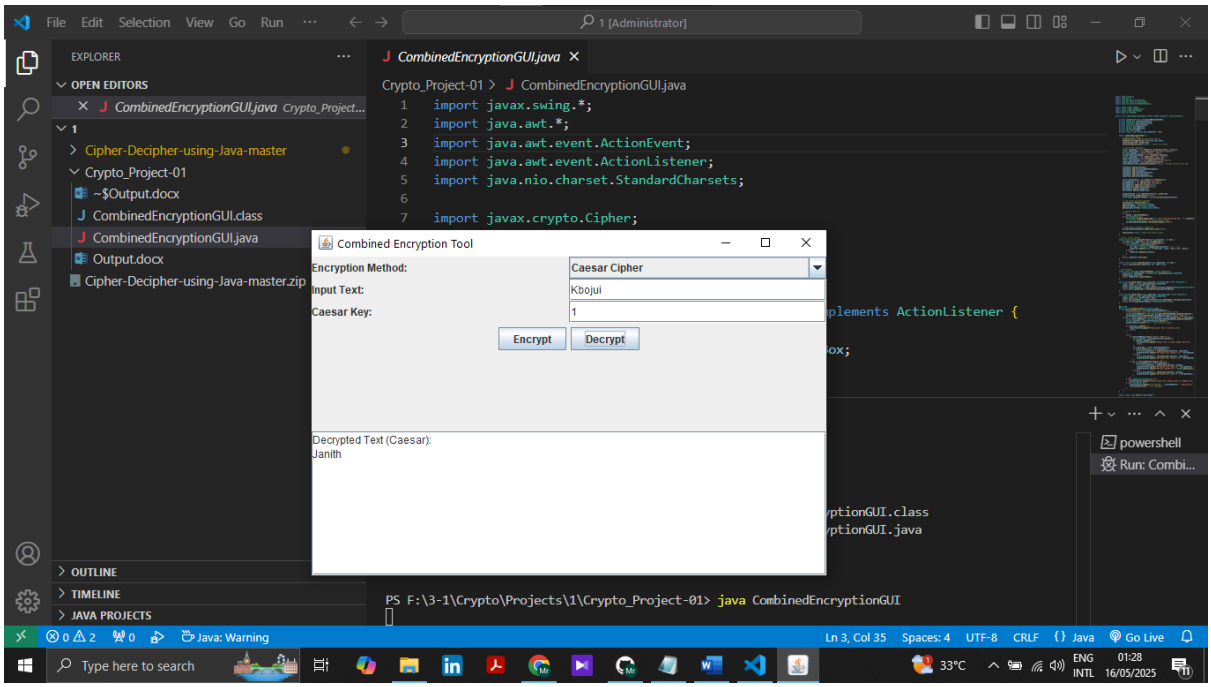
Important Notes

- **Security:** This application is intended for educational and demonstration purposes. It does not implement secure key management practices. For production systems, use robust key management solutions.
- **AES Key Handling:** The AES key is generated within the application and displayed for the user. This is **not secure** for real-world use.
- **Error Handling:** The application includes basic error handling, such as checking for invalid Caesar cipher keys.

OUTPUT




Combined Encryption Tool
Encryption Method:
Caesar Cipher
Input Text:
Janith
Caesar Key:
1
Encrypt
Decrypt
Encrypted Text (Caesar):
Kbojui
CombinedEncryptionGUI.java
PS F:\3-1\Crypto\Projects\1\Crypto_Project-01> java CombinedEncryptionGUI


Combined Encryption Tool
Encryption Method:
Caesar Cipher
Input Text:
Kbojui
Caesar Key:
1
Encrypt
Decrypt
Decrypted Text (Caesar):
Janith
CombinedEncryptionGUI.java
PS F:\3-1\Crypto\Projects\1\Crypto_Project-01> java CombinedEncryptionGUI