

# PROJECT BASED TASK

## ASSIGNMENT

NAME : Ch.Vyshnavi

BRANCH : CSE

REGD NO: 22PA1A0524

SECTION : A

SUBJECT : CNS

### GITHUB LINK:

<https://github.com/vyshu0111/CNS-Digital-signature-system>

### SCREENSHOTS OF OUTPUT:

#### SCREENSHOT -1

```
♦ Welcome to Digital Signature System ♦

✗ Key files not found! Generating new keys...

♦ Generating RSA Key Pair...
✓ Public and Private Keys Generated and Saved!

Options:
1 Sign a Message
2 Verify a Signature
3 Exit

Enter your choice (1/2/3): 2

Enter the original message: hello lakshman
✗ Signature file not found! Sign a message first.

Options:
1 Sign a Message
2 Verify a Signature
3 Exit

Enter your choice (1/2/3): 2

Enter the original message: hellow
✗ Signature file not found! Sign a message first.
```

#### SCREENSHOT-2

```
Options:
1 Sign a Message
2 Verify a Signature
3 Exit

Enter your choice (1/2/3): 1

Enter the message to sign: hai lucky

  * Signing the message...
  ✓ Digital Signature Generated and Saved!

  * Digital Signature: 3db38989a94240c8de4f2ab7091d589e18b553d7501ab019f50a6739e42b14847996c2ef05d851c2acc2e07ea482fd1f0149dd3d6ada493a18b7c9

Options:
1 Sign a Message
2 Verify a Signature
3 Exit

Enter your choice (1/2/3): 2

Enter the original message: hai lucky

  * Verifying the signature...
  ✓ Signature is VALID! Message is authentic.

Options:
1 Sign a Message
2 Verify a Signature
3 Exit

Enter your choice (1/2/3): 3

  ✨ Exiting... Thank you for using the Digital Signature System!
```

## File Structure

```
$bash
$Copy
$Edit
📁 Digital-Signature-System/
|---Digital_signature_system.ipynb #Jupyter Notebook containing the digital
signature system
|--- README.md
```

## Usage

### Generating RSA Keys

Upon the first execution of the program, it will automatically generate and save both the public and private RSA keys in PEM format.

### Signing a Message

To sign a message, follow these steps:

1. Run the `digital_signature.py` script.
2. Select the option "1 Sign a Message" from the menu.
3. Enter the message you wish to sign (e.g., "Hello, Secure World!").
4. The program will generate a digital signature for the entered message and save it to `signature.txt`.

### Verifying a Signature

To verify a signature, follow these steps:

1. Run the digital\_signature.py script.
2. Select the option "**2** Verify a Signature" from the menu.
3. Enter the original message that was signed.
4. The program will check if the signature stored in signature.txt is valid for the entered message.