Laboratory Assignment

IEMS 5710 Crypto., Info. Security and Privacy
(2$^{\text{nd}}$ Trimester, 2024-25)
Full Mark: 100

Deadline: 23:59 HKT
28 December 2024

Objectives

This assignment covers some key topics we will cover in the lectures. Upon completion, you will be able to:

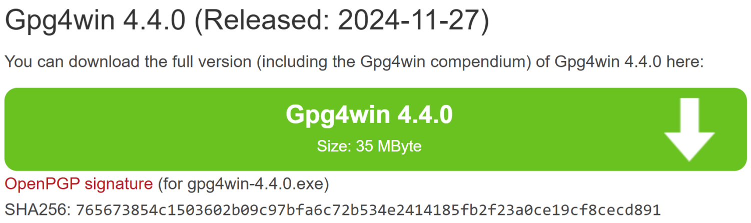
- Establish a secure communication channel using public-key encryption
- Change the access level of a file in a Linux machine
- Write some simple SQL queries for querying a database
- Write an HTML webpage file to interact with a "web application" written in PHP

(This "laboratory assignment" is designed to provide you with first-hand experiences in security-related computing environments. You are advised to complete at least the first three parts of the question early to have some hands-on experience before you commit to taking this course.)

1 Encryption and Decryption using GPG4WIN (25%)

1.1 Installation

1. Download GPG4WIN from the following URL: https://www.gpg4win.org/download.html.
2. Click the download button.



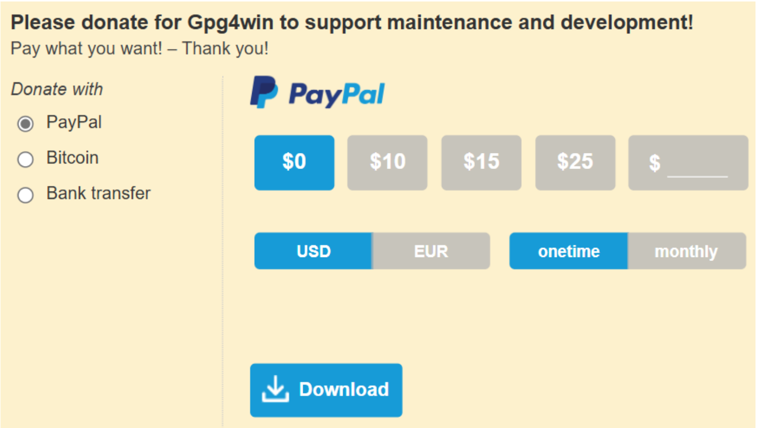
Gpg4win 4.4.0 (Released: 2024-11-27)

You can download the full version (including the Gpg4win compendium) of Gpg4win 4.4.0 here:

OpenPGP signature (for gpg4win-4.4.0.exe)

SHA256: 765673854c1503602b09c97bfa6c72b534e2414185fb2f23a0ce19cf8cecd891

3. Select "$0" and start to download. // Well, you can always choose to donate :)

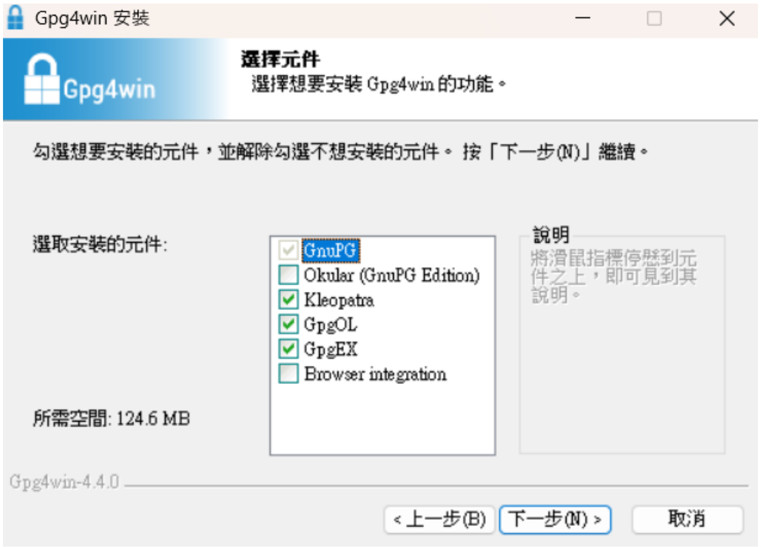


Please donate for Gpg4win to support maintenance and development!
Pay what you want! – Thank you!

Donate with
● PayPal
○ Bitcoin
○ Bank transfer

4. Double-click the file to start the installation process.

5. Select the default option.



選擇元件

選擇想要安裝 Gpg4win 的功能。

勾選想要安裝的元件，並解除勾選不想安裝的元件。 按「下一步(N)」繼續。

選取安裝的元件:

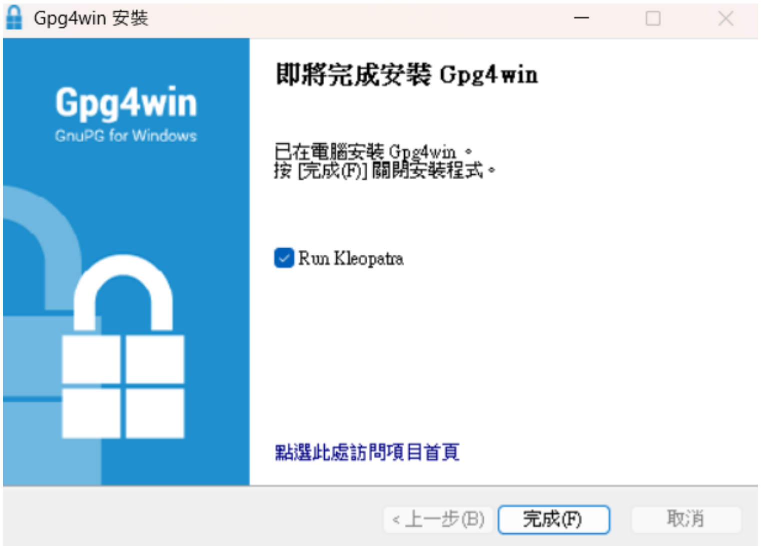
☑ GnuPG
□ Okular (GnuPG Edition)
☑ Kleopatra
☑ GpgOL
☑ GpgEX
□ Browser integration

說明
將滑鼠指標停懸到元件之上，即可見到其說明。

所需空間: 124.6 MB

Gpg4win-4.4.0

6. Finally, select “Run Kleopatra” and press **Finish**.



即將完成安裝 Gpg4win

已在電腦安裝 Gpg4win。
按 [完成(F)] 關閉安裝程式。

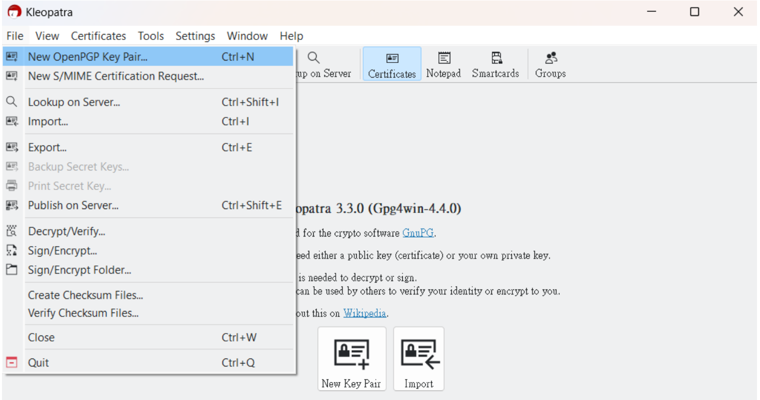
☑ Run Kleopatra

點選此處訪問項目首頁

1.2 Generate a public and private key pair

This section demonstrates the steps to generate a public and private key pair in the system.

1. After the software has started, select **New OpenPGP Key Pair**.
   (Question you should ask yourself: What does “PGP” stand for? Google is your friend.)



opatra 3.3.0 (Gpg4win-4.4.0)

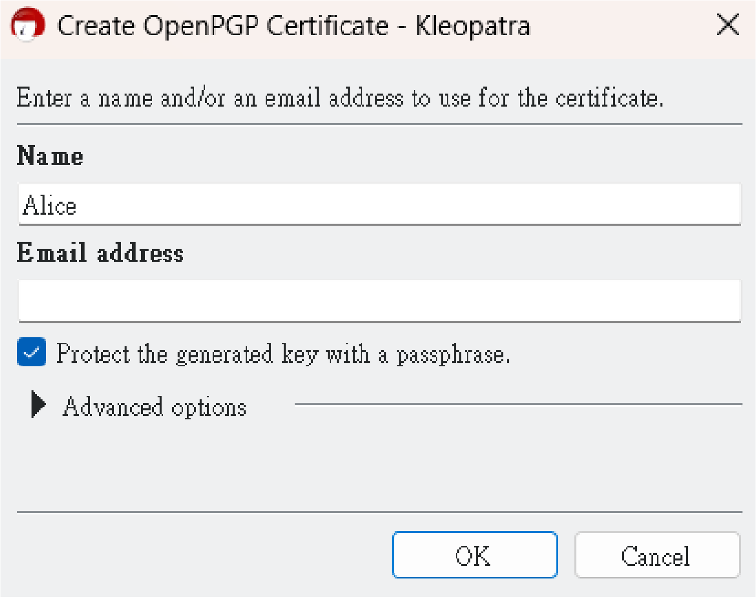
l for the crypto software GnuPG.

eed either a public key (certificate) or your own private key.

is needed to decrypt or sign.
can be used by others to verify your identity or encrypt to you.

out this on Wikipedia.

2. Select **Create a personal OpenPGP key pair**, fill in your name and email address if you want, and tick the box “Protect the generated key with a passphrase” (Why?).



Enter a name and/or an email address to use for the certificate.

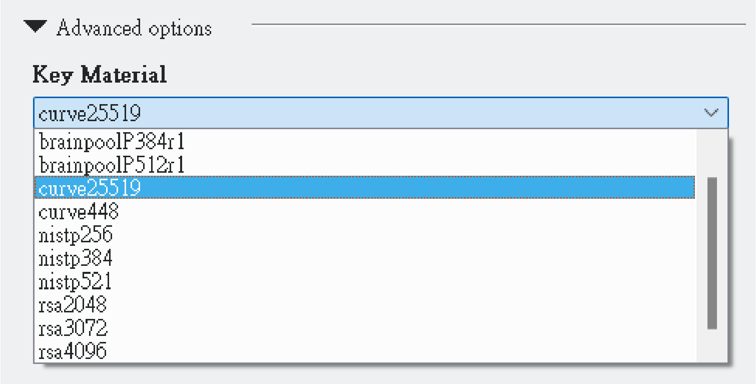
Name

Alice

Email address

☑ Protect the generated key with a passphrase.

You can click “Advanced options” for other available options. For example, you can increase the key length in the advanced setting dialog box. We will use the default options in this lab assignment. (Question to yourself: What are curve25519, nistp256, rsa2048?)

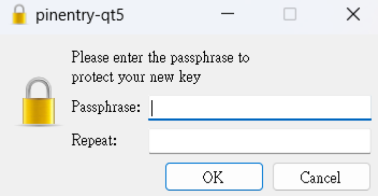


Key Material

curve25519

3. Click **OK** to create the key pair.

4. Enter a password to protect your new (private) key.

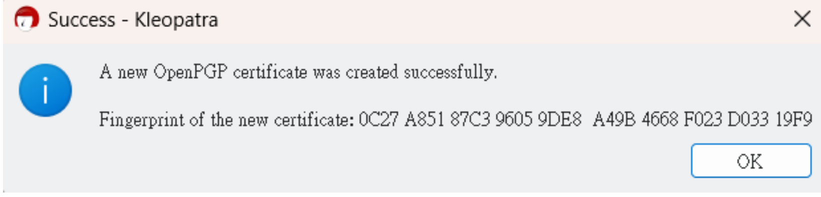


Please enter the passphrase to protect your new key

Passphrase: |

Repeat:

5. Your new key pair has been created.



A new OpenPGP certificate was created successfully.

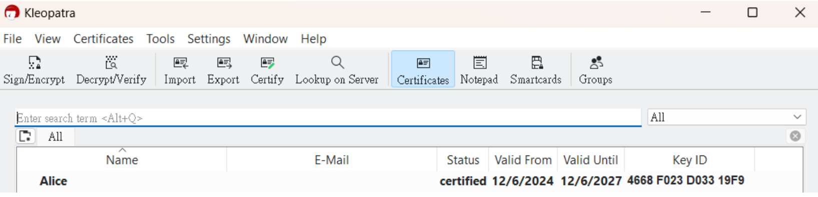
Fingerprint of the new certificate: 0C27 A851 87C3 9605 9DE8 A49B 4668 F023 D033 19F9

Figure 1: Generate a public and private key pair

1.3 Export your own public key

To establish a secure communication channel, you have to give your public key to others. This section details the steps to export your public key to a file so you can send it out.

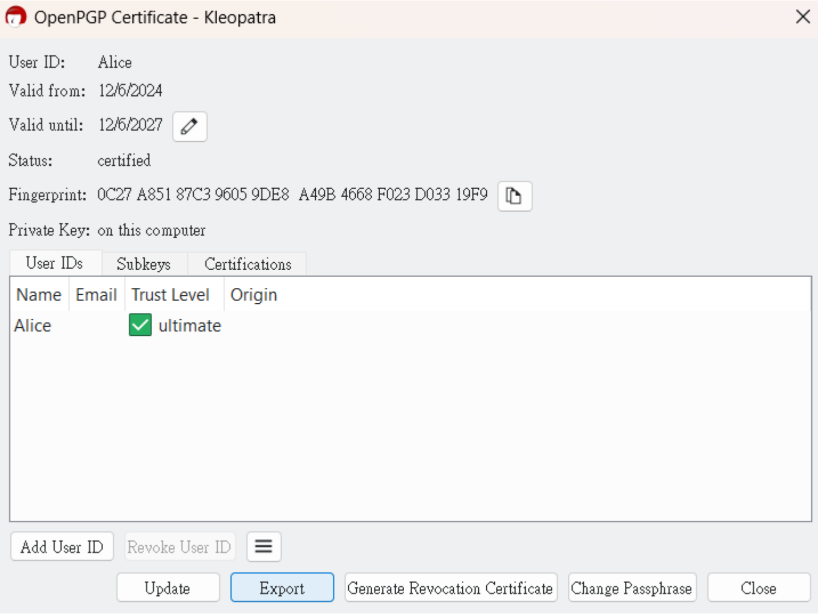
1. After you press **OK** in the previous step, you will see the following box.



Name	E-Mail	Status	Valid From	Valid Until	Key ID
Alice		certified	12/6/2024	12/6/2027	4668 F023 D033 19F9

2. Move your mouse cursor and double-click your name.

3. The certificate details will be shown.



User ID: Alice

Valid from: 12/6/2024

Valid until: 12/6/2027

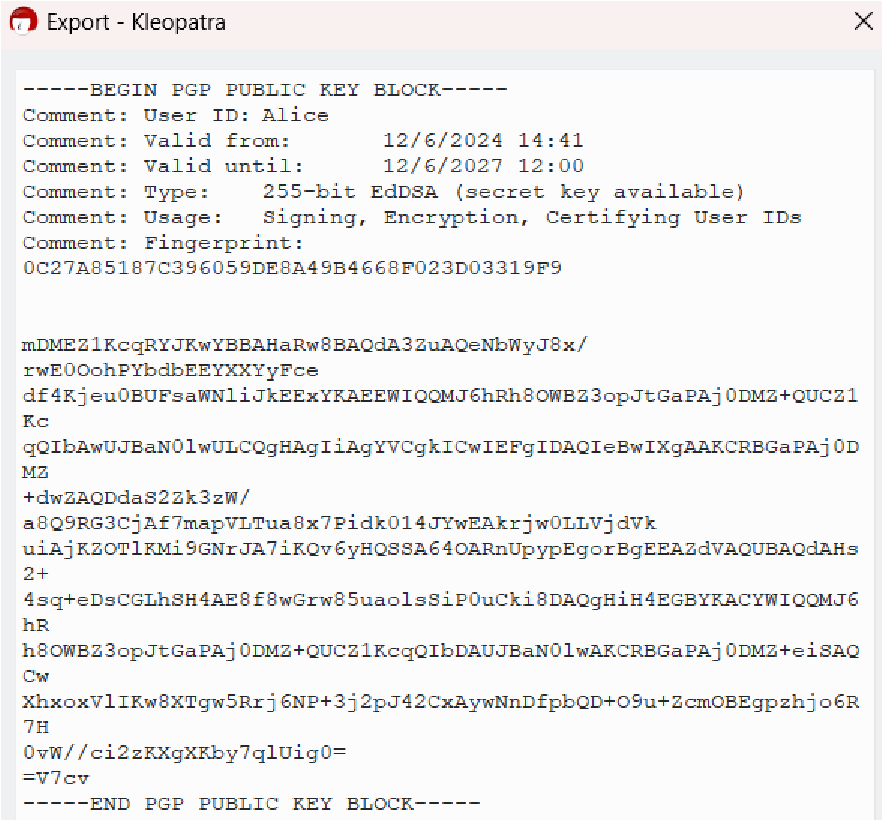
Status: certified

Fingerprint: 0C27 A851 87C3 9605 9DE8 A49B 4668 F023 D033 19F9

Private Key: on this computer

Name	Email	Trust Level	Origin
Alice		✓ ultimate	

4. Click the **Export** button to view your public key.

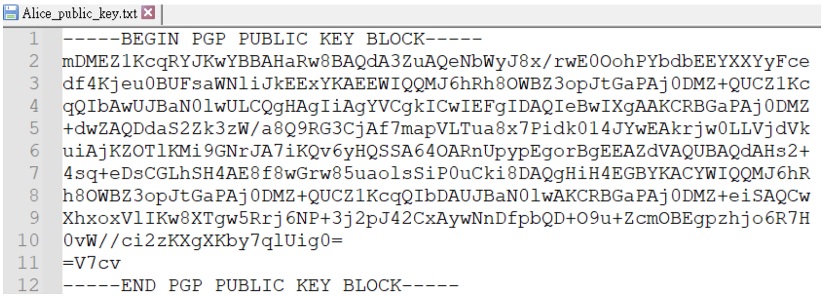


```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Comment: User ID: Alice
Comment: Valid from:      12/6/2024 14:41
Comment: Valid until:     12/6/2027 12:00
Comment: Type:    255-bit EdDSA (secret key available)
Comment: Usage:   Signing, Encryption, Certifying User IDs
Comment: Fingerprint:
0C27A85187C396059DE8A49B4668F023D03319F9


mDMEZ1KcqRYJKwYBBAHaRw8BAQdA3ZuAQeNbWyJ8x/
rwE0OohPYbdbEEYXXYyFce
df4Kjeu0BUFsaWNliJkEExYKAEEWIQQMJ6hRh8OWBZ3opJtGaPAj0DMZ+QUCZ1
Kc
qQIbAwUJBaN0lwULCQgHAgIiAgYVCgkICwIEFgIDAQIeBwIXgAAKCRBGaPAj0D
MZ
+dwZAQDdaS2Zk3zW/
a8Q9RG3CjAf7mapVLTua8x7Pidk014JYwEAkrjw0LLVjdVk
uiAjKZOTlKMi9GNrJA7iKQv6yHQSSA64OARnUpypEgorBgEEAZdVAQUBAQdAHs
2+
4sq+eDsCGLhSH4AE8f8wGrw85uaolsSiP0uCki8DAQgHiH4EGBYKACYWIQQMJ6
hR
h8OWBZ3opJtGaPAj0DMZ+QUCZ1KcqQIbDAUJBaN0lwAKCRBGaPAj0DMZ+eiSAQ
Cw
XhxoxVlIKw8XTgw5Rrj6NP+3j2pJ42CxAywNnDfpbQD+O9u+ZcmOBEgpzhjo6R
7H
0vW//ci2zKXgXKby7qlUig0=
=V7cv
-----END PGP PUBLIC KEY BLOCK-----
```

Figure 2: Export your public key to a file

5. Right-click and select all content. Copy the content to the clipboard. Open a text editor (*e.g.*, notepad) and paste the content. Remove lines 2 to 9 from the content. You should have the following screen.



```
-----BEGIN PGP PUBLIC KEY BLOCK-----
mDMEZ1KcqRYJKwYBBAHaRw8BAQdA3ZuAQeNbWyJ8x/rwE0OohPYbdbEEYXXYyFce
df4Kjeu0BUFsaWNliJkEExYKAEEWIQQMJ6hRh8OWBZ3opJtGaPAj0DMZ+QUCZ1Kc
qQIbAwUJBaN0lwULCQgHAgIiAgYVCgkICwIEFgIDAQIeBwIXgAAKCRBGaPAj0DMZ
+dwZAQDdaS2Zk3zW/a8Q9RG3CjAf7mapVLTua8x7Pidk014JYwEAkrjw0LLVjdVk
uiAjKZOTlKMi9GNrJA7iKQv6yHQSSA64OARnUpypEgorBgEEAZdVAQUBAQdAHs2+
4sq+eDsCGLhSH4AE8f8wGrw85uaolsSiP0uCki8DAQgHiH4EGBYKACYWIQQMJ6hR
h8OWBZ3opJtGaPAj0DMZ+QUCZ1KcqQIbDAUJBaN0lwAKCRBGaPAj0DMZ+eiSAQCw
XhxoxVlIKw8XTgw5Rrj6NP+3j2pJ42CxAywNnDfpbQD+O9u+ZcmOBEgpzhjo6R7H
0vW//ci2zKXgXKby7qlUig0=
=V7cv
-----END PGP PUBLIC KEY BLOCK-----
```

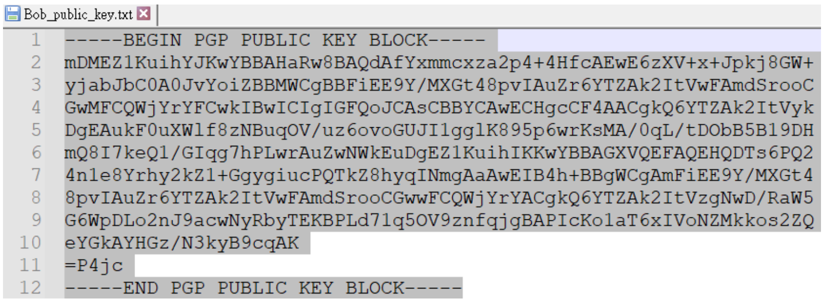
6. Save the file and name it **[your name]_public_key.txt**.

7. You can send this file to your partner for the following step.

1.4 Import others' public keys

You need other's public key to encrypt messages for secure communication. This section demonstrates the steps to import others' public keys into the system.

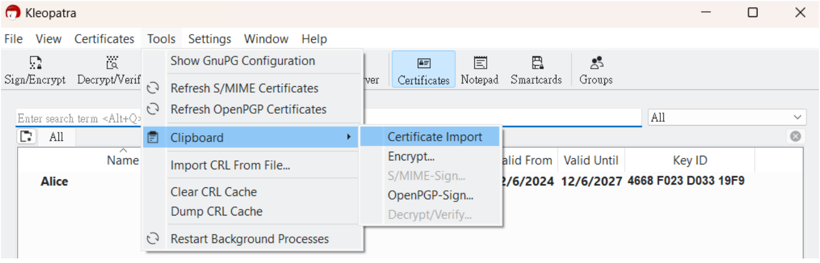
1. When you get the public key from others, open it with a text editor and copy the content to the clipboard.



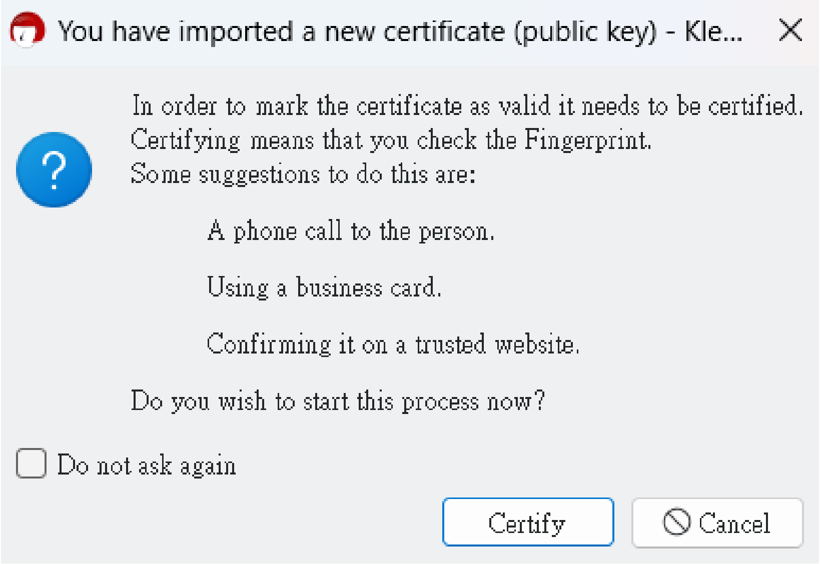
Bob_public_key.txt

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
mDMEZ1KuihYJKwYBBAHaRw8BAQdAfYxmmcxza2p4+4HfcAEwE6zXV+x+Jpkj8GW+
yjabJbC0A0JvYoiZBBMWCgBBFiEE9Y/MXGt48pvIAuZr6YTZAk2ItVwFAmdSrooC
GwMFCQWjYrYFCwkIBwICIgIGFQoJCAsCBBYCAwECHgcCF4AACgkQ6YTZAk2ItVyk
DgEAukF0uXWlf8zNBuqOV/uz6ovoGUJI1gglK895p6wrKsMA/0qL/tDObB5B19DH
mQ8I7keQ1/GIqg7hPLwrAuZwNWkEuDgEZ1KuihIKKwYBBAGXVQEFAQEHQDTs6PQ2
4n1e8Yrhy2kZ1+GgygiucPQTkZ8hyqINmgAaAwEIB4h+BBgWCgAmFiEE9Y/MXGt4
8pvIAuZr6YTZAk2ItVwFAmdSrooCGwwFCQWjYrYACgkQ6YTZAk2ItVzgNwD/RaW5
G6WpDLo2nJ9acwNyRbyTEKBPLd71q5OV9znfqjgBAPIcKo1aT6xIVoNZMkkos2ZQ
eYGkAYHGz/N3kyB9cqAK
=P4jc
-----END PGP PUBLIC KEY BLOCK-----
```

2. Go to Kleopatra and select **Certificate Import** (the button is grey if nothing is copied).



3. In this dialog box, click **Certify**.



You have imported a new certificate (public key) - Kle...

In order to mark the certificate as valid it needs to be certified.
Certifying means that you check the Fingerprint.
Some suggestions to do this are:

A phone call to the person.

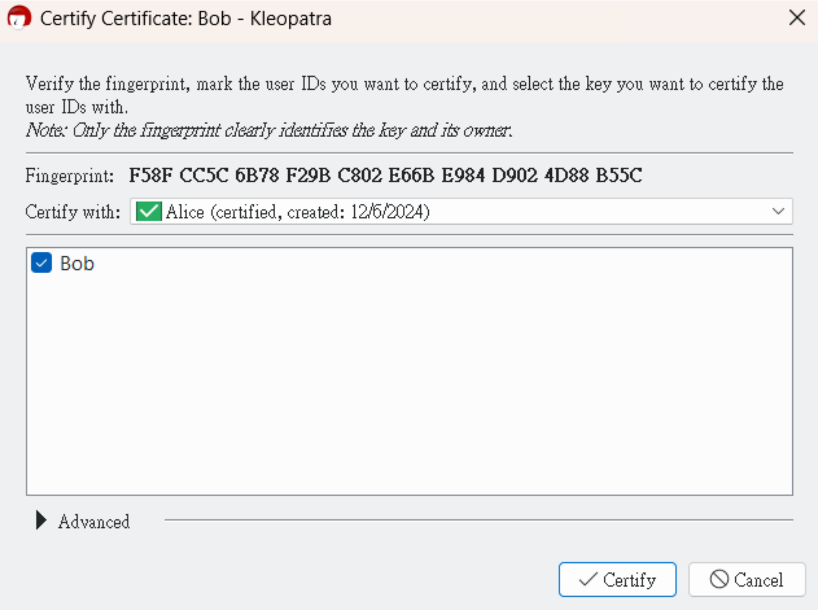
Using a business card.

Confirming it on a trusted website.

Do you wish to start this process now?

□ Do not ask again

4. Click **Certify**.



Certify Certificate: Bob - Kleopatra

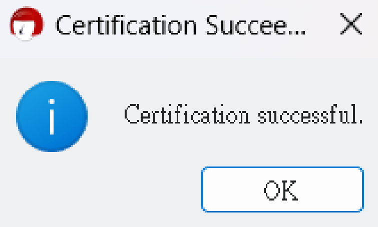
Verify the fingerprint, mark the user IDs you want to certify, and select the key you want to certify the user IDs with.
Note: Only the fingerprint clearly identifies the key and its owner.

Fingerprint: F58F CC5C 6B78 F29B C802 E66B E984 D902 4D88 B55C

Certify with: ✓ Alice (certified, created: 12/6/2024)

☑ Bob

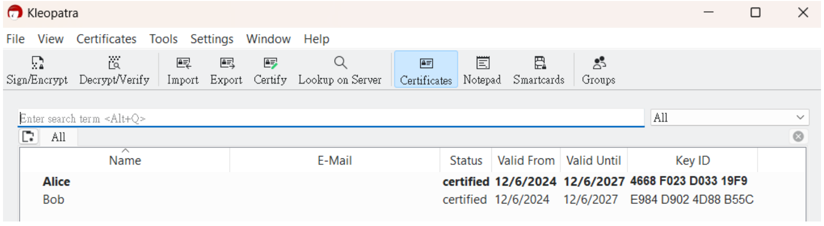
5. Click **OK**. You may need to input the password that you have set in Section 1.2.



Certification Succee...

Certification successful.

6. The public key of your partner has been imported successfully.



Kleopatra
File View Certificates Tools Settings Window Help
Sign/Encrypt Decrypt/Verify Import Export Certify Lookup on Server Certificates Notepad Smartcards Groups
Enter search term <Alt+Q>
All
All

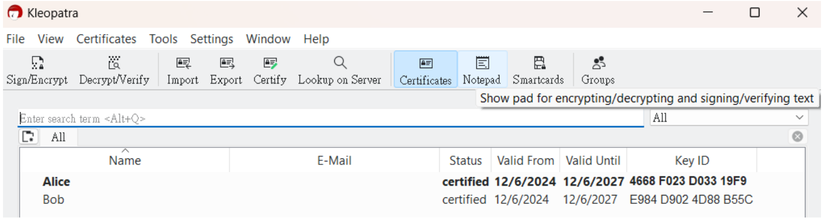
Name	E-Mail	Status	Valid From	Valid Until	Key ID
Alice		**certified**	**12/6/2024**	**12/6/2027**	**4668 F023 D033 19F9**
Bob		certified	12/6/2024	12/6/2027	E984 D902 4D88 B55C

Figure 3: Import your partner’s public key to the system

1.5 Encrypt messages by using the public key and sign by using your private key

After you have imported other's public key to the system, you can use his/her public key to encrypt a message and use your own private key to sign the message. This section demonstrates the steps to do so.

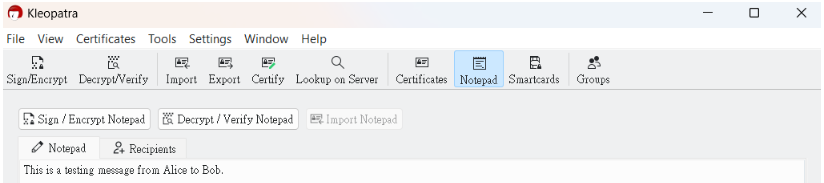
1. Open Kleopatra and select **Notepad** in the toolbar.



Show pad for encrypting/decrypting and signing/verifying text

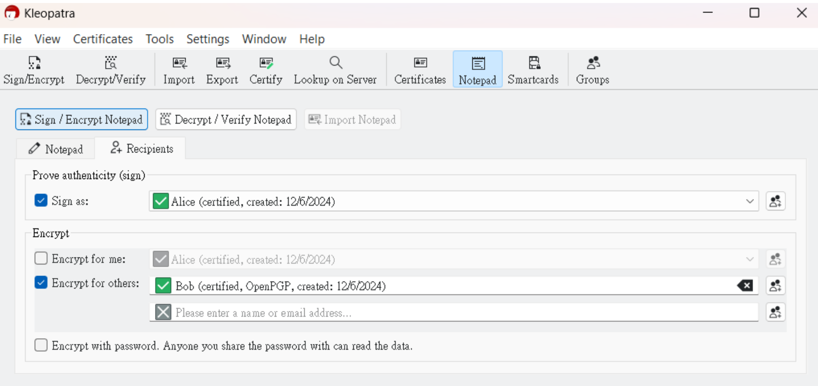
Name	E-Mail	Status	Valid From	Valid Until	Key ID
Alice		**certified**	**12/6/2024**	**12/6/2027**	**4668 F023 D033 19F9**
Bob		certified	12/6/2024	12/6/2027	E984 D902 4D88 B55C

2. Type in the message that you are going to encrypt.



This is a testing message from Alice to Bob.

3. In **Recipients**, tick "Sign as" and input the recipient in "Encrypt for others"; click **Sign/Encrypt Notepad**.



Prove authenticity (sign)

☑ Sign as: Alice (certified, created: 12/6/2024)

Encrypt

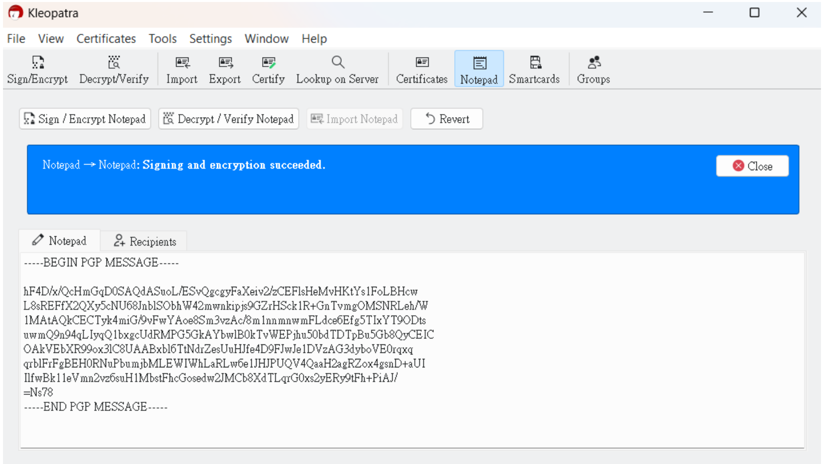
□ Encrypt for me: Alice (certified, created: 12/6/2024)

☑ Encrypt for others: Bob (certified, OpenPGP, created: 12/6/2024)

Please enter a name or email address...

□ Encrypt with password. Anyone you share the password with can read the data.

4. The encryption is done successfully. Select the **Notepad** tab to view the encrypted message.



Notepad → Notepad: **Signing and encryption succeeded.**

```
-----BEGIN PGP MESSAGE-----

hF4D/x/QcHmGqD0SAQdASuoL/ESvQgcgyFaXeiv2/zCEFlsHeMvHKtYs1FoLBHcw
L8sREFfX2QXy5cNU68JnblSObhW42mwnkipjs9GZrHSck1R+GnTvmgOMSNRLeh/W
1MAtAQkCECTyk4miG/9vFwYAoe8Sm3vzAc/8m1nnmnwmFLdce6Efg5TIxYT9ODts
uwmQ9n94qLIyqQ1bxgcUdRMPG5GkAYbwlB0kTvWEPjhu50bdTDTpBu5Gb8QyCEIC
OAkVEbXR99ox3lC8UAABxbl6TtNdrZesUuHJfe4D9FJwJe1DVzAG3dyboVE0rqxq
qrblFrFgBEH0RNuPbumjbMLEWIWhLaRLw6e1JHJPUQV4QaaH2agRZox4gsnD+aUI
IlfwBk11eVmn2vz6suH1MbstFhcGosedw2JMCb8XdTLqrG0xs2yERy9tFh+PiAJ/
=Ns78
-----END PGP MESSAGE-----
```

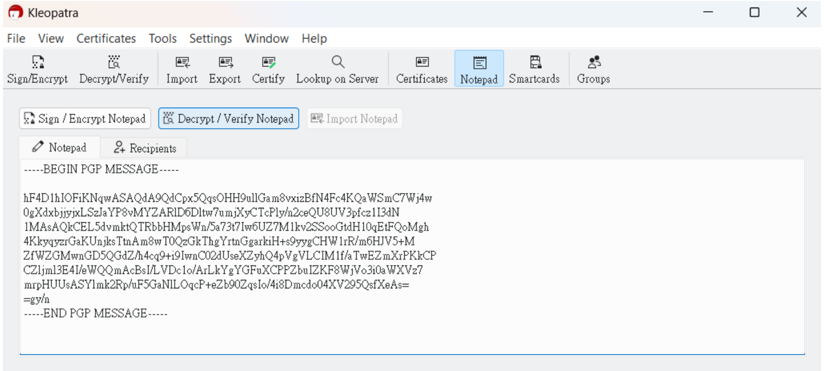
Figure 4: Encrypt a message to your partner

5. You can copy and send the content as a file or by email.

1.6 Decrypt message by using the private key and verify using other's public key

After receiving the encrypted message from the sender, you can use your private key to decrypt the message and use the sender's public key to verify the sender's identity.

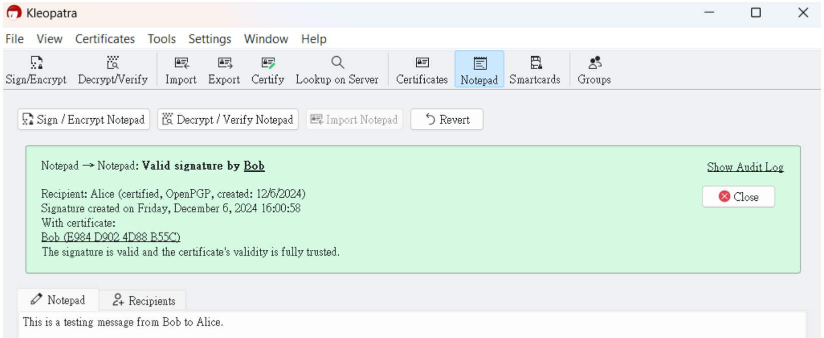
1. On the recipient's side, open Kleopatra and click the **Notepad** tab.
2. Paste the encrypted message to the box and click **Decrypt/Verify Notepad**.



```
-----BEGIN PGP MESSAGE-----

hF4D1hIOFiKNqwASAQdA9QdCpx5QqsOHH9ullGam8vxizBfN4Fc4KQaWSmC7Wj4w
0gXdxbjjyjxLSzJaYP8vMYZARlD6Dltw7umjXyCTcPly/n2ceQU8UV3pfcz1I3dN
1MAsAQkCEL5dvmktQTRbbHMpsWn/5a73t7Iw6UZ7M1kv2SSooGtdH10qEtFQoMgh
4KkyqyzrGaKUnjksTtnAm8wT0QzGkThgYrtnGgarkiH+s9yygCHW1rR/m6HJV5+M
ZfWZGMwnGD5QGdZ/h4cq9+i9IwnC02dUseXZyhQ4pVgVLCIM1f/aTwEZmXrPKkCP
CZljml3E4I/eWQQmAcBsI/LVDc1o/ArLkYgYGFuXCPPZbuIZKF8WjVo3i0aWXVz7
mrpHUUsASYlmk2Rp/uF5GaNlLOqcP+eZb90ZqsIo/4i8Dmcdo04XV295QsfXeAs=
=gy/n
-----END PGP MESSAGE-----
```

3. Kleopatra will verify the sender's identity and start decrypting the content.



Notepad → Notepad: **Valid signature by Bob**

Recipient: Alice (certified, OpenPGP, created: 12/6/2024)
Signature created on Friday, December 6, 2024 16:00:58
With certificate:
Bob (E984 D902 4D88 B55C)
The signature is valid and the certificate's validity is fully trusted.

This is a testing message from Bob to Alice.

Figure 5: Decrypt an encrypted message from your partner

Your Tasks

In this question, you need to submit the following:

- Generate a public and private key pair (Figure 1).
- Export your public key to a file (Figure 2).
- Import your partner's public key to the system (Figure 3).
- Encrypt a message to your partner (Figure 4).
- Decrypt an encrypted message from your partner (Figure 5).

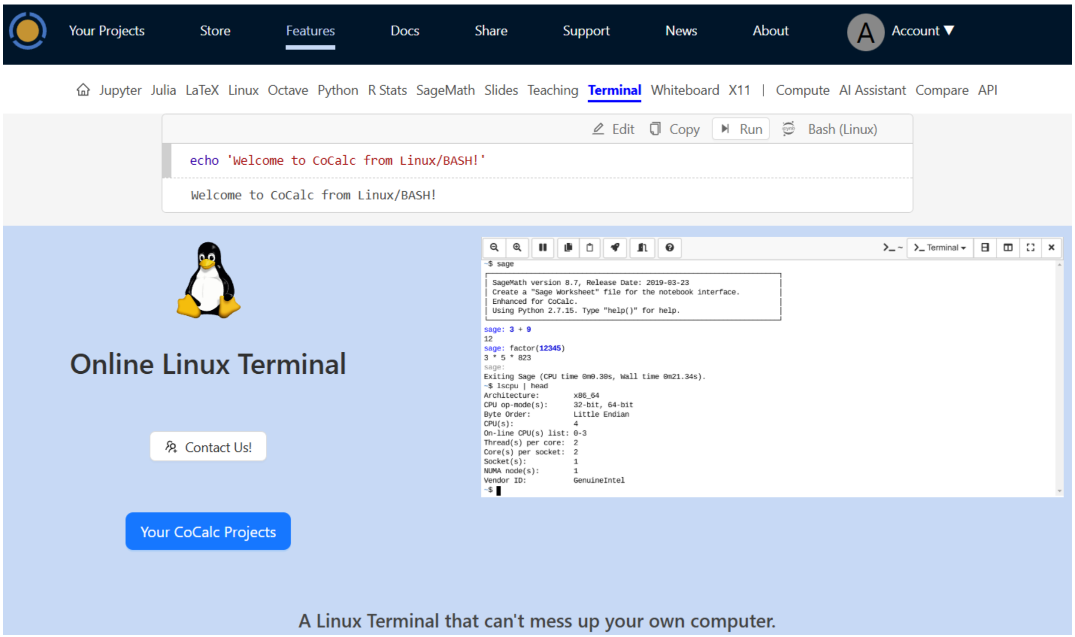
To show your work, you need to capture your computer screen in these steps (*i.e.*, Figures 1-5) and include them in your report. Optionally, you can include the long form of the acronyms you encountered.

2 Access Control in Linux Machine (25%)

2.1 Connect to a Linux VM

1. Go to the following URL: https://cocalc.com/doc/terminal.html.

2. Sign up for an account and click **Your CoCalc Projects** to start an Online Linux Terminal.



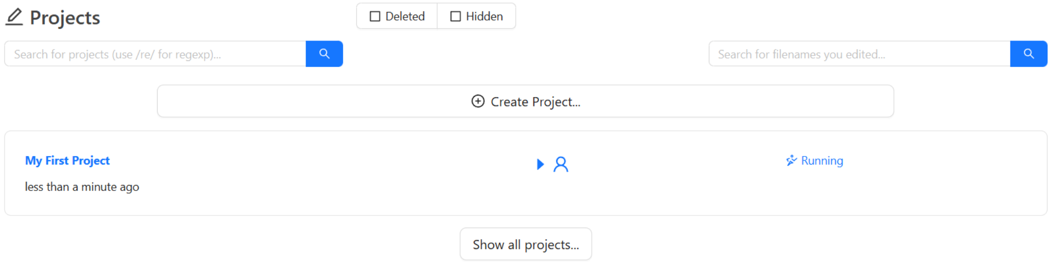
```
echo 'Welcome to CoCalc from Linux/BASH!'
```

```
Welcome to CoCalc from Linux/BASH!
```

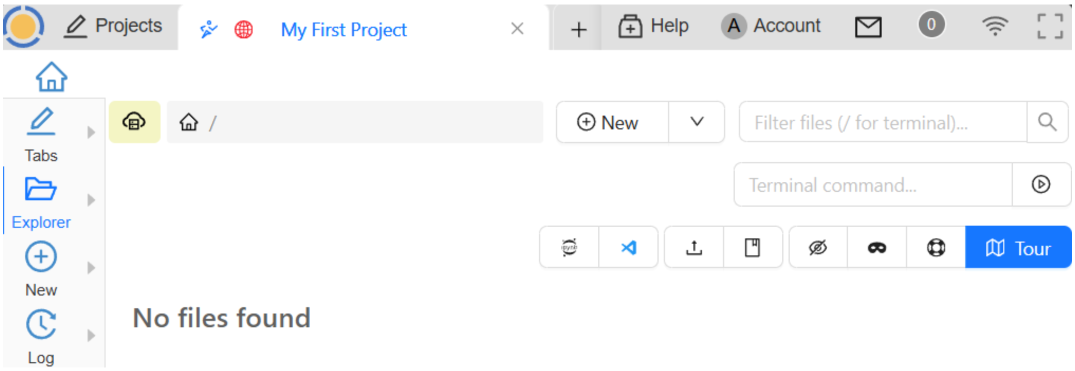
```
~$ sage
┌────────────────────────────────────────────────────────────────────┐
│ SageMath version 8.7, Release Date: 2019-03-23                     │
│ Create a "Sage Worksheet" file for the notebook interface.         │
│ Enhanced for CoCalc.                                               │
│ Using Python 2.7.15. Type "help()" for help.                       │
└────────────────────────────────────────────────────────────────────┘
sage: 3 + 9
12
sage: factor(12345)
3 * 5 * 823
sage:
Exiting Sage (CPU time 0m0.30s, Wall time 0m21.34s).
~$ lscpu | head
Architecture:        x86_64
CPU op-mode(s):      32-bit, 64-bit
Byte Order:          Little Endian
CPU(s):              4
On-line CPU(s) list: 0-3
Thread(s) per core:  2
Core(s) per socket:  2
Socket(s):           1
NUMA node(s):        1
Vendor ID:           GenuineIntel
~$
```

A Linux Terminal that can't mess up your own computer.

3. Click **My First Project**.



4. After entering the project, you can type commands in "Terminal command...".



2.2 Create a sample file

1. Type the following command in the shell and press enter to execute it:

```
echo "This is a test file" > test_1.txt
```

   - `echo` is a command for displaying something on the console display.
   - > is a symbol for redirecting from the console display.
   - `test_1.txt` will be the destination for the above redirection.

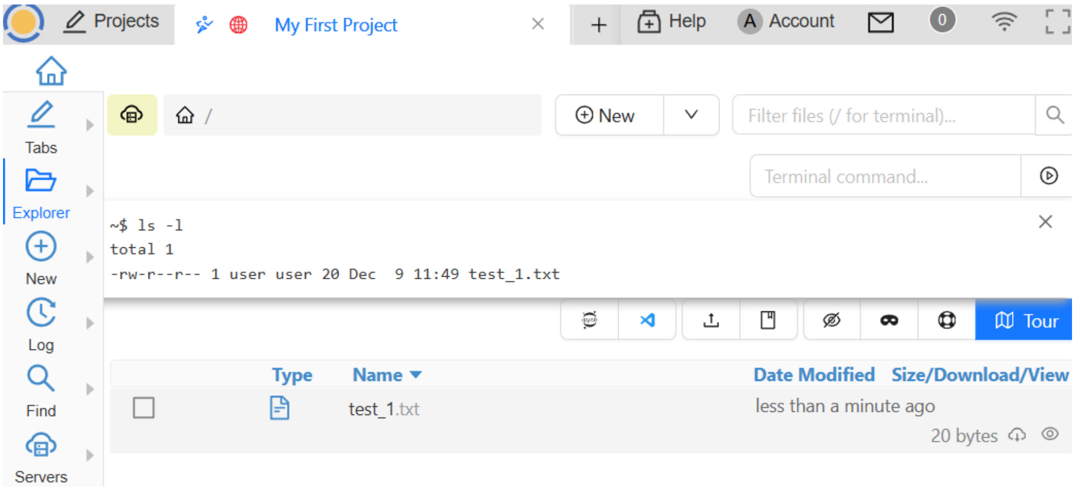
   Simply put, the effect is it will create a file called `test_1.txt` with one line.
   Warning: it *overwrites* the file if the file exists.

2. Execute the following command to list the file in the current directory:

```
ls -l
```

   - `ls` is a command for listing directory (folder in Windows terminology) contents.
   - `ls -l` means “using a long listing format.”
   - You can try `ls --help` or `man ls` to know more about `ls`.

   After running the command, you should be able to see the following:



```
~$ ls -l
total 1
-rw-r--r-- 1 user user 20 Dec  9 11:49 test_1.txt
```

	Type	Name ▾	Date Modified	Size/Download/View
□		test_1.txt	less than a minute ago	20 bytes

2.3 Set the access level of a file in a Linux machine

1. In Linux, we use the `chmod` command with the following syntax to set/change the access level of a file:

   ```
   chmod [option] permissions file_name
   ```

   - There are three kind of users: `u` = owner, `g` = the group where the file belongs to, `o` = others.
   - Also, there are three kinds of access restrictions: `r` $= 4 =$ read, `w` $= 2 =$ write, `x` $= 1 =$ execute.
   - Here, `option` is not necessary to set.

2. To **add** the **executing** permission of `test_1.txt` to the **owner**, we execute the following:

   ```
   chmod u+x test_1.txt
   ```

3. To **remove** the **reading** permission of `test_1.txt` to the **group of users** and **others**:

   ```
   chmod g-r,o-r test_1.txt
   ```

4. To add the **reading**, **writing**, and **executing** permission to the **owner**, **group of users**, and **others**:

   ```
   chmod u+wxr,g+rwx,o+rwx test_1.txt
   ```

   The above command is equivalent to the following **numerical** representations:

   ```
   chmod 777 test_1.txt
   ```

   Here 7 equals `+wxr` $(4 + 2 + 1)$.

Your Tasks

In this question, you need to submit the **command** to do the following:

1. Create a file named `[the last 4 digits of your SID].txt` with “Submission to Q2” as content.
2. Set the following access restriction of the above file using **numerical** representations:
   (a) Give the **reading**, **writing**, and **executing** permission to the **owner**.
   (b) Give the **reading** permission to the **group of users**.
   (c) Give the **executing** permission to **others**.
3. Set the following access restriction of the above file using **alphabetical** representations:
   (a) Remove the **executing** permission to the **owner** and **others**.
   (b) Give the **writing** permission to the **group of users**.

3 Basic SQL (25%)

3.1 Connect to the SQL Online IDE

- Go to the following URL: https://sqliteonline.com.
- To run a query, just click **Run** or press **Shift + Enter**.

3.2 Learn SQL

Structured Query Language (SQL) is a database query language that allows the management of data in a relational database. In this part, we provide some basic SQL query examples to give you a taste of how they work, which helps you better understand SQL injection, to be covered in the lecture later.

1. **CREATE** a table named `users_info` with three columns: `ID (unique int)`, `username (varchar(225))`, `password (varchar(225))`: (Question to yourself: What are `NOT NULL`, `UNIQUE`, `VARCHAR`?)

   ```
   CREATE TABLE users_info (
       ID INT NOT NULL UNIQUE,
       username VARCHAR(225) NOT NULL,
       password VARCHAR(225) NOT NULL
   );
   ```

   Once a table is created, it cannot be created again; hence, the above query can be run only once.

   To DELETE the table, we use DROP:

   ```
   DROP TABLE users_info;
   ```

   A query ends with ";".

2. **INSERT** the following rows into `user_info`:

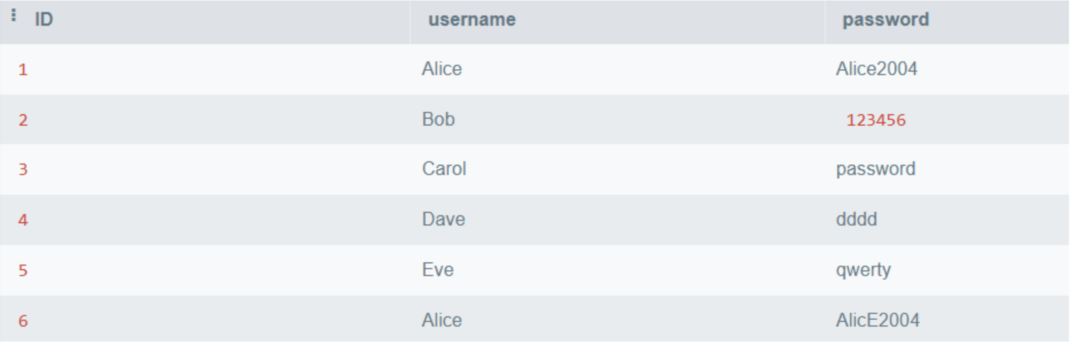
   ```
   INSERT INTO users_info (ID, username, password)
   VALUES (1, 'Alice', 'Alice2004'),
   (2, 'Bob', '123456'),
   (3, 'Carol', 'password'),
   (4, 'Dave', 'dddd'),
   (5, 'Eve', 'qwerty'),
   (6, 'Alice', 'AlicE2004');
   ```

   (Question to yourself: Are the above passwords strong enough? If no, what makes a stronger password?)

3. **SELECT** all columns of `users_info`:

   ```
   SELECT * FROM users_info;
   ```

   You should see the following output:

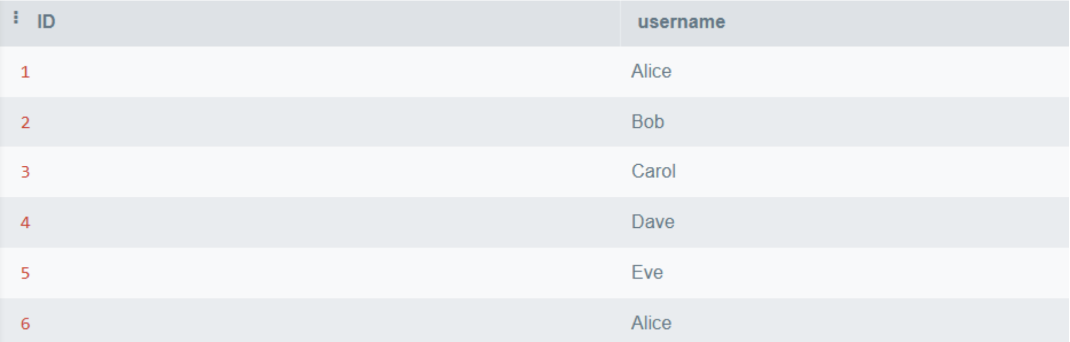


ID	username	password
1	Alice	Alice2004
2	Bob	123456
3	Carol	password
4	Dave	dddd
5	Eve	qwerty
6	Alice	AlicE2004

Figure 6: Select the table `users_info` after creation and insertion

4. **SELECT** specific columns from `users_info`:

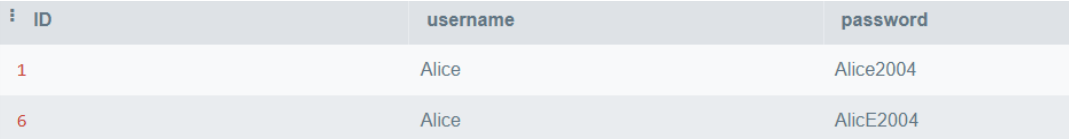
```
SELECT ID, username FROM users_info;
```



ID	username
1	Alice
2	Bob
3	Carol
4	Dave
5	Eve
6	Alice

5. Use the **WHERE** clause to filter some of the records:

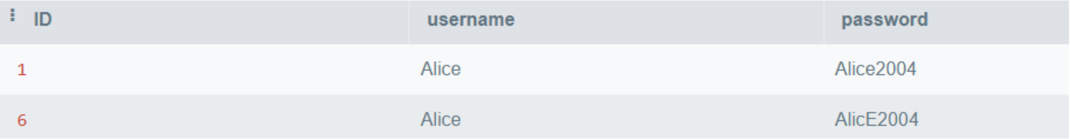
```
SELECT * FROM users_info WHERE username='Alice';
```



ID	username	password
1	Alice	Alice2004
6	Alice	AlicE2004

6. Use the **LIKE** operator to search for records containing a specific pattern:
   Username starts with “a”:

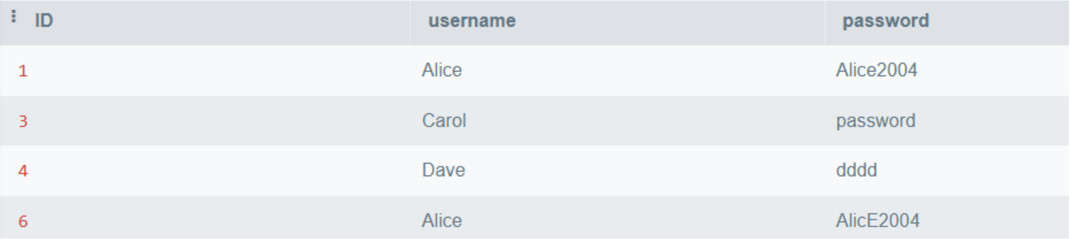
```
SELECT * FROM users_info WHERE username LIKE 'a%';
```



ID	username	password
1	Alice	Alice2004
6	Alice	AlicE2004

Username contains with “a”:

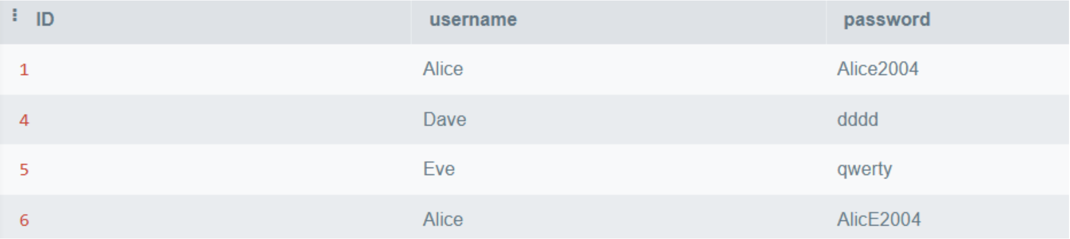
```
SELECT * FROM users_info WHERE username LIKE '%a%';
```



ID	username	password
1	Alice	Alice2004
3	Carol	password
4	Dave	dddd
6	Alice	AlicE2004

Username ends with “e”:

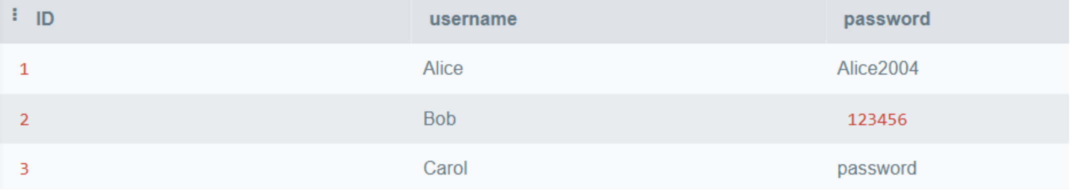
```
SELECT * FROM users_info WHERE username LIKE '%e';
```



ID	username	password
1	Alice	Alice2004
4	Dave	dddd
5	Eve	qwerty
6	Alice	AlicE2004

7. Use the **BETWEEN** operator to select records given a range:

```
SELECT * FROM users_info WHERE ID BETWEEN 1 AND 3;
```



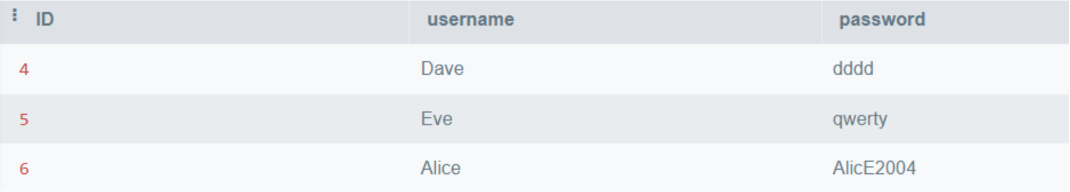
ID	username	password
1	Alice	Alice2004
2	Bob	123456
3	Carol	password

8. Use **comparison** operators to select records given a condition. They can be:

Operator	Description
=	Equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
<>	Not equal to

Table 1: A list of comparison operators

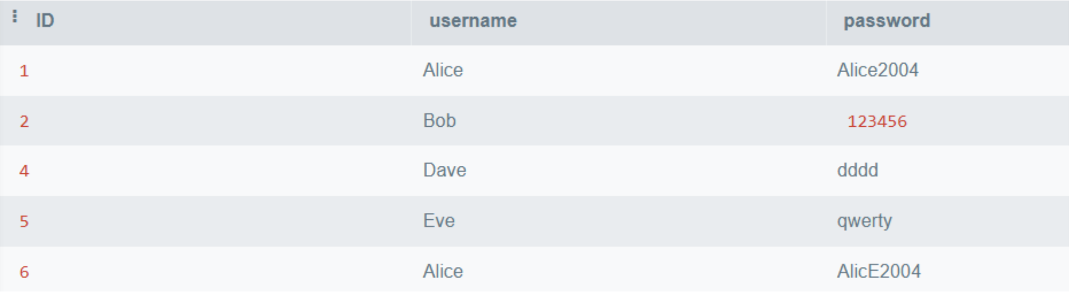
```
SELECT * FROM users_info WHERE ID > 3;
```



ID	username	password
4	Dave	dddd
5	Eve	qwerty
6	Alice	AlicE2004

9. Use **AND**, **OR** operator to select records based on more than one condition:

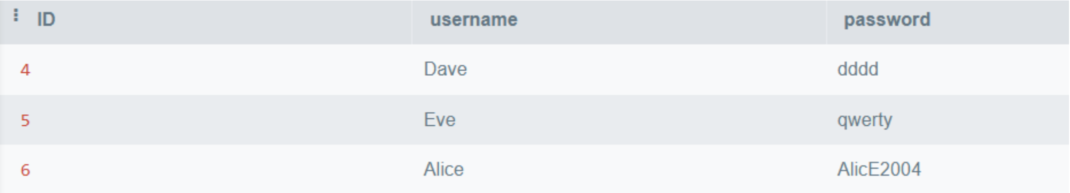
```
SELECT * FROM users_info WHERE ID BETWEEN 1 AND 2 OR username LIKE '%e';
```



ID	username	password
1	Alice	Alice2004
2	Bob	123456
4	Dave	dddd
5	Eve	qwerty
6	Alice	AlicE2004

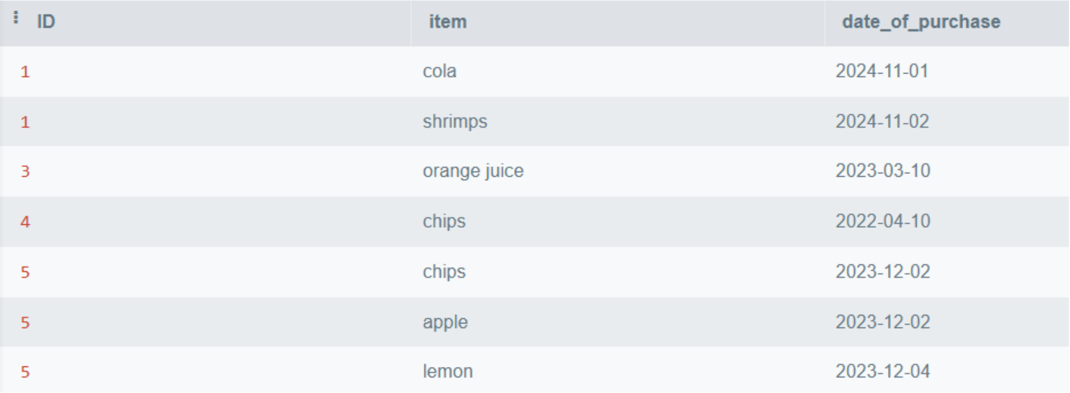
10. Use **NOT** operator to select records that are not true for the given condition:

```
SELECT * FROM users_info WHERE ID NOT BETWEEN 1 AND 3;
```



ID	username	password
4	Dave	dddd
5	Eve	qwerty
6	Alice	AlicE2004

11. **CREATE** another table named purchase_record with three columns: ID (int), item (varchar(225)), date_of_purchase (date). **INSERT** the following records into purchase_record:



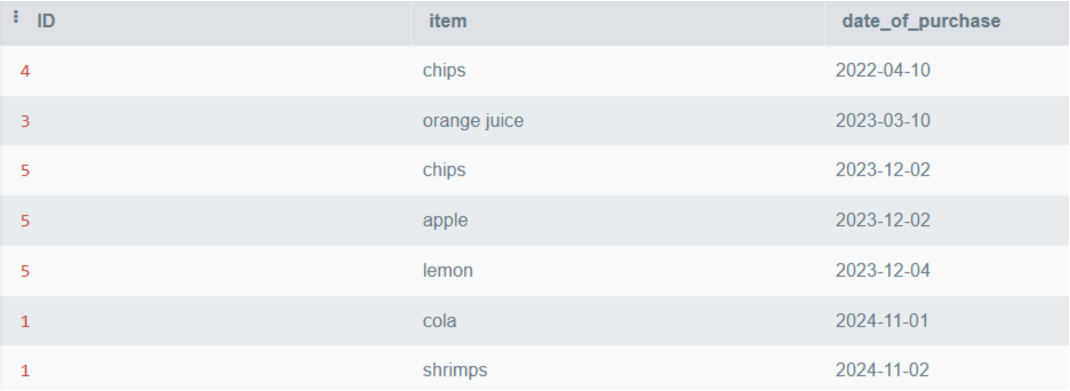
ID	item	date_of_purchase
1	cola	2024-11-01
1	shrimps	2024-11-02
3	orange juice	2023-03-10
4	chips	2022-04-10
5	chips	2023-12-02
5	apple	2023-12-02
5	lemon	2023-12-04

Figure 7: Select the table purchase_record after creation and insertion

Try to write the query by yourself.

12. Use **ORDER BY** to sort the records based on some column(s); the default setting is in ascending order (**ASC** or **DSEC** to specify outputting in descending order):

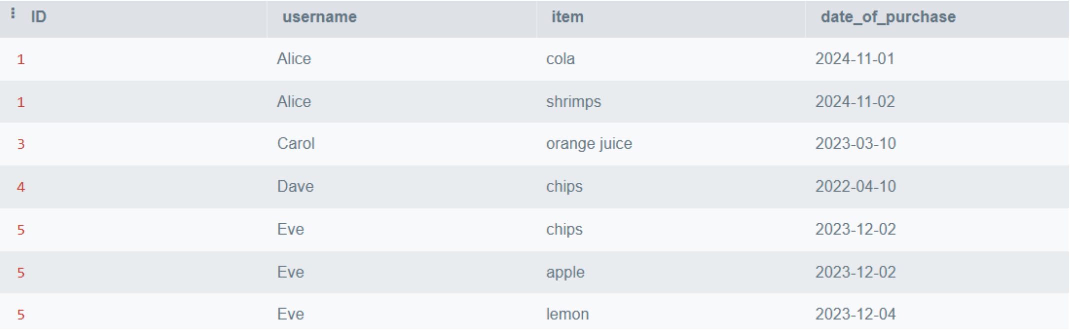
```
SELECT * FROM purchase_record ORDER BY date_of_purchase;
```



ID	item	date_of_purchase
4	chips	2022-04-10
3	orange juice	2023-03-10
5	chips	2023-12-02
5	apple	2023-12-02
5	lemon	2023-12-04
1	cola	2024-11-01
1	shrimps	2024-11-02

13. Use a **JOIN** clause to combine the ID rows from users_info and purchase_record:

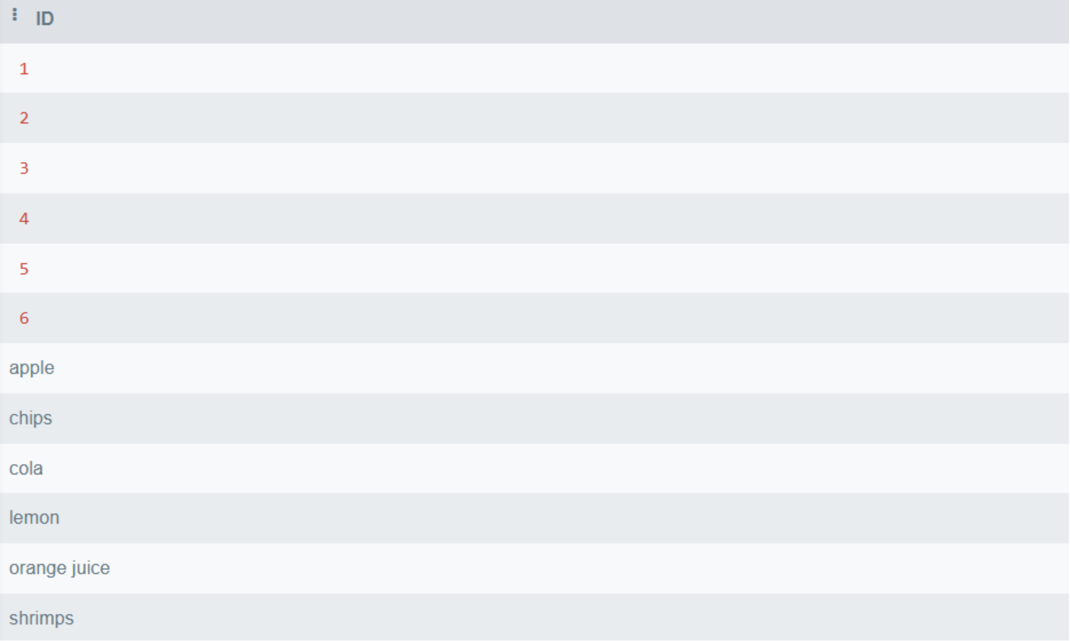
```
SELECT users_info.ID, users_info.username, purchase_record.item,
purchase_record.date_of_purchase
FROM users_info
JOIN purchase_record ON users_info.ID = purchase_record.ID;
```



ID	username	item	date_of_purchase
1	Alice	cola	2024-11-01
1	Alice	shrimps	2024-11-02
3	Carol	orange juice	2023-03-10
4	Dave	chips	2022-04-10
5	Eve	chips	2023-12-02
5	Eve	apple	2023-12-02
5	Eve	lemon	2023-12-04

14. Use the **UNION** operator to combine the result of two or more **SELECT** statements:

```
SELECT ID FROM users_info
UNION
SELECT item FROM purchase_record;
```



ID
1
2
3
4
5
6
apple
chips
cola
lemon
orange juice
shrimps

Your Tasks

In this question, you need to submit the **queries** and the **screenshot of the output** to do the following:

1. **Create** the table `users_info` (Figure 6) and **insert** the records, which includes an **additional row**:
   - `ID`: [the last 4 digits of your SID]
   - `username`: [your nickname]
   - `password`: [any password you like]
2. **Create** the table `purchase_record` (Figure 7) and **insert** the records, which include an **additional row**:
   - `SID`: [the last 4 digits of your SID]
   - `item`: [anything you like]
   - `date_of_purchase`: [the date you conduct this task]
3. **Join** `users_info` and `purchase_record` on their `ID`; **select** the `users_info.ID`, `users_info.username`, `purchase_record.item`, `purchase_record.date_of_purchase` with date of purchase **after "2023-12-03"** and **order** the records by date of purchase in **descending** order.

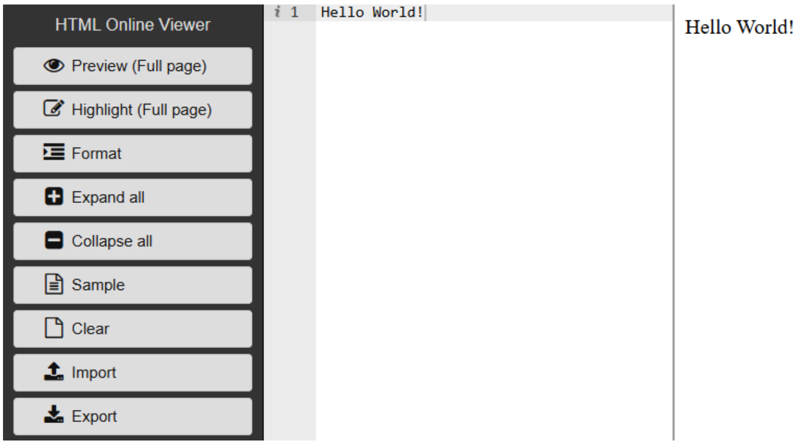
4 Basic HTML (25%)

HTML stands for Hypertext Markup Language. It is the standard markup language for web documents. In this part, we will write simple HTML to interact with a PHP file using an online editor.

4.1 Start Coding with an Online Editor

1. Go to the following URL: https://html.onlineviewer.net

   The left part of this website is a coding platform that allows you to write your code, and the right part is a preview pane showing your code's result.



```
Hello World!
```

Hello World!

4.2 HTML

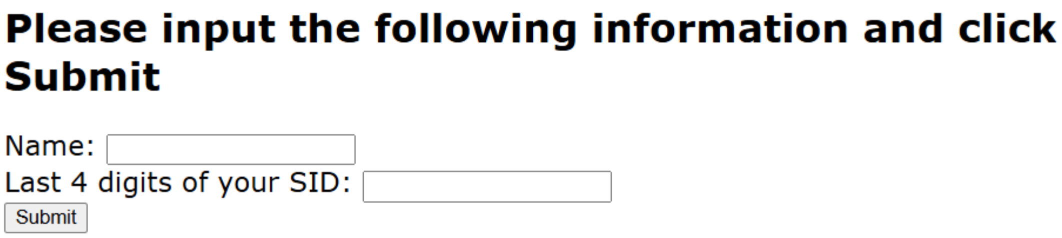
1. We are going to write two HTML webpages, which submit your name and the last 4 digits of your SID to a "web file" written in PHP hosted at http://iems5710.42web.io.

   In HTML, a tag (e.g., <b></b>) is used to tell the browser what the type of content is, and an element (e.g., <b>12345</b>) is used to tell the browser what to display. In this part, we use <form>element to collect the inputs and submit them to the hosted PHP file via the GET or POST method.

2. Copy and paste the following code on the coding platform:

```
<html>
<head>
    <style>
        header {
            font-family: Verdana;
            font-size: 20pt;
            font-weight: bold;
            margin-bottom: 20px;
        }
        body {
            font-family: Verdana;
            font-size: 14pt;
            margin: 40px;
        }
    </style>
</head>
<body>
<header>
    Please input the following information and click Submit
</header>
<form action="http://iems5710.42web.io" method="get">
Name: <input type="text" name="name"><br>
Last 4 digits of your SID: <input type="text" name="sid"><br>
<input type="submit">
</form>
</body>
</html>
```

Then, the webpage is shown on the right part:

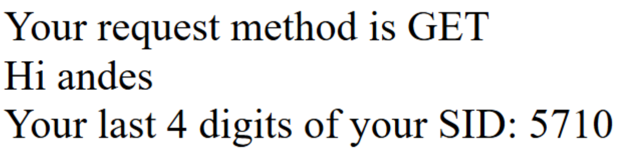


Please input the following information and click Submit

Name:

Last 4 digits of your SID:

3. Click **Preview (Full Page)** on the top-left to view the website in the full-page mode.
4. Input your **name** and **last 4 digits of your SID** on the boxes, then click the **Submit** button:



Your request method is GET
Hi andes
Your last 4 digits of your SID: 5710

You may see the warning “The information you’re about to submit is not secure” before submitting the form. Click **Send anyway** for this task. (Question to yourself: Why does this warning appear, and what is the potential risk for submitting a form in this way?)

5. The above example uses the **GET** method to submit the request to the host. Now, let’s modify the code to submit the request to the host via the **POST** method:

   Change the following code in Step 1:

```
<form action="http://iems5710.42web.io" method="get">
```

   to

```
<form action="http://iems5710.42web.io" method=post">
```

Your Tasks

In this question, you need to do the following:

1. Do Steps 1-3 for submitting **GET** and **POST** requests.
2. Take the screenshots of Step 3 and include them in your report.
3. From the response pages of both HTML files, what are the differences between the **GET** and **POST** methods? (Hints: look at the URL of the response pages.)

Assignment Submission

Put all screenshots, queries, and answers required in **Your Tasks** of each question into a PDF file.
Please name the file in the following format: **1155001234 Chan Tai Man.pdf**
Deadline: Dec 28th, 2024, 11:59 pm Hong Kong Time.

- End -