PASSWORD MANAGER SYSTEM

Group 2:

Mufid Fata Rifki

Reyandanu Fajri Daniswara

Faculty:

Tri Agus Riyadi / Ivan Firdaus

Class:

3CS2

CEP CCIT FTUI

UNIVERSITY OF INDONESIA

2025

PROJECT INFORMATION

Project Title : Password Manager System

Class : 3CS2

Start Date : 20 September 2025

End Date : 25 September 2025

Faculty : Tri Agus Riyadi / Ivan Firdaus

Developer:

1. Mufid Fata Rifki
2. Reyandanu Fajri Daniswara

ACKNOWLEDGEMENT

As an author and developer, the author would like to acknowledge the finished product of this paper submission "**Password Manager System"**. This paper explains the author's project on the CRUD concept and applies aspects of cybersecurity using Python as the code language and MySQL for the database through MySQL workbench as the interface. This project focuses on the security measure to store website, url, and private credential password.

The authors also want to thank the faculty who mentored and helped with this whole project. And even though there are some imperfections, regardless of the imperfection, this imperfection will serve as a greater learning experience for future projects.

Overall, this paper serves the whole concept of this project, by documenting the whole process, from the flowchart, source code, database, and database design. By using Python as a main code language and MySQL for the query and database.

Depok, 25 September 2025

Authors

SYSTEM ANALYSIS

The primary objective of this project is to implement the CRUD concept which is a (Create, Read, Update, Delete) function to the project. So we came up with "**Password Manager System"** as the project.

So the concept are, to make a program that can store passwords for anything, such as website, application & software credentials, financial information, and personal & identity records safely and securely. So the first thing is to register to make the account, after the registration function it connects and inserts into the ***users*** table by inputting username and master password, and then login into the program by using the credentials. The next is to create the vault for storing the password by inputting the vault name, vault password, and description.

After making the vault, now we can put the entries (credentials), by pressing the add entry option the program will ask for input about the website name or anything, username, password, and description. That's for the main concept. For the addition users can update, delete, and read the vault. Same with the entries users can also update, delete, and read the entries. And inside the database, the passwords are all encrypted and hidden so the developer doesn't see the credentials, by using SHA-256 and salt to make sure every user feels secure.

However, just like any project, there might be some imperfection. There may be bugs or miss config if run on other devices, or unseen logical errors. Beside all of that, we make sure to make the best version from our effort for this project.

PREPARATION

For this project it needs a few preparations to get this program running, these are the essentials:

1. **Visual Studio Code (IDE)**

   In this case we're using Vscode as our main IDE

2. **Python**

   Make sure download the latest python version (python3.0+)

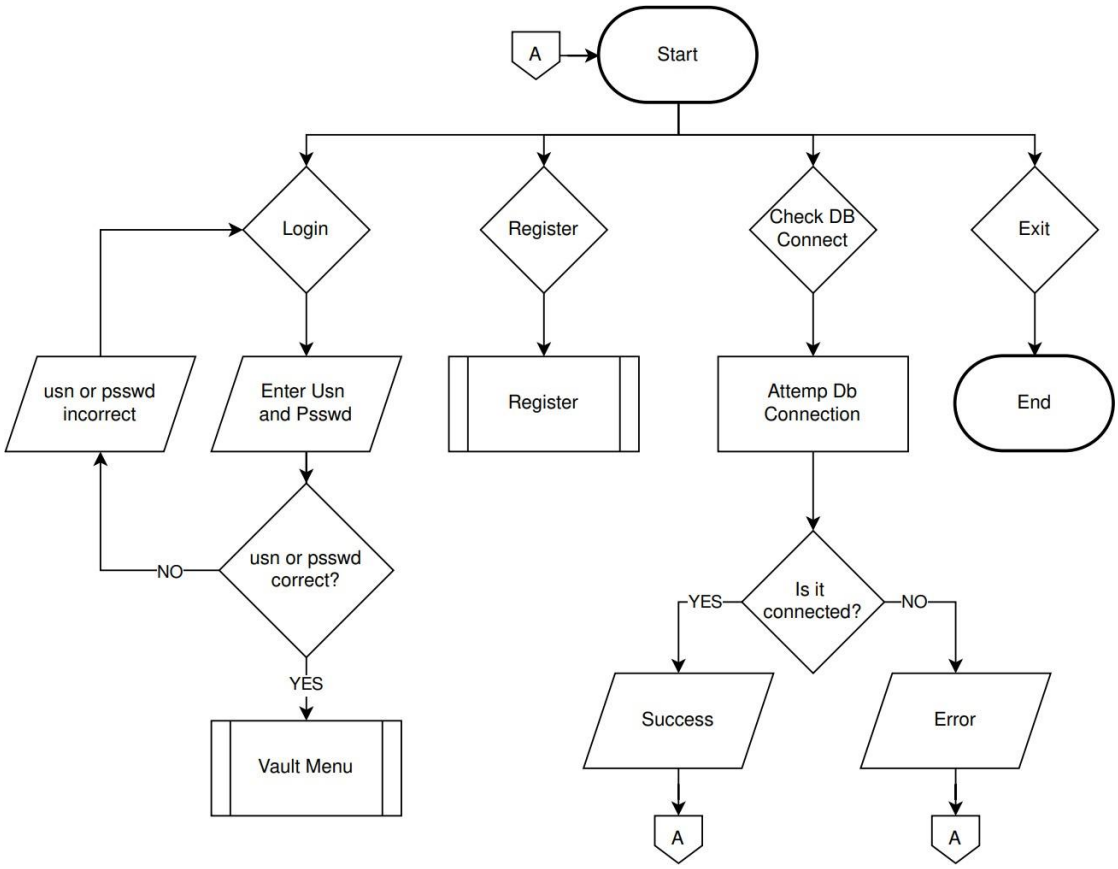
3. **Install MySQL Server / MySQL workbench**

   MySQL for the database and the data interface

4. And mind that this project was made in windows 11 OS. Reader can adjust the preparation based on their own OS

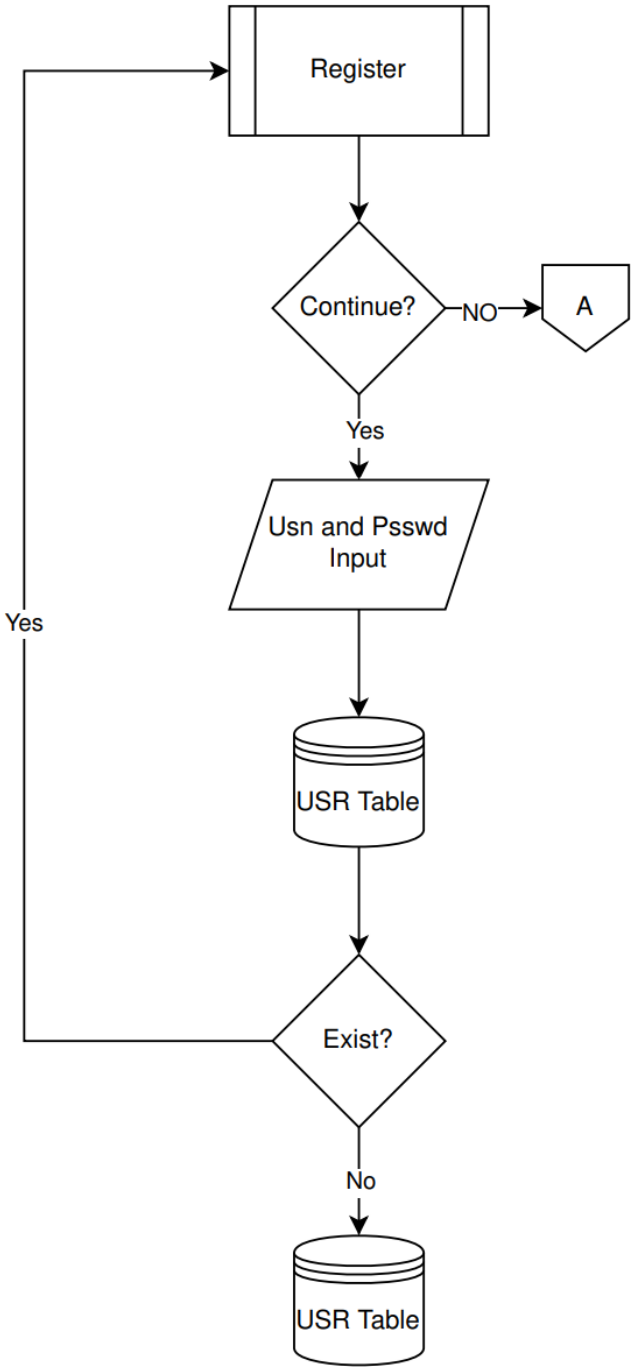
FLOW CHART

- **Main Menu**



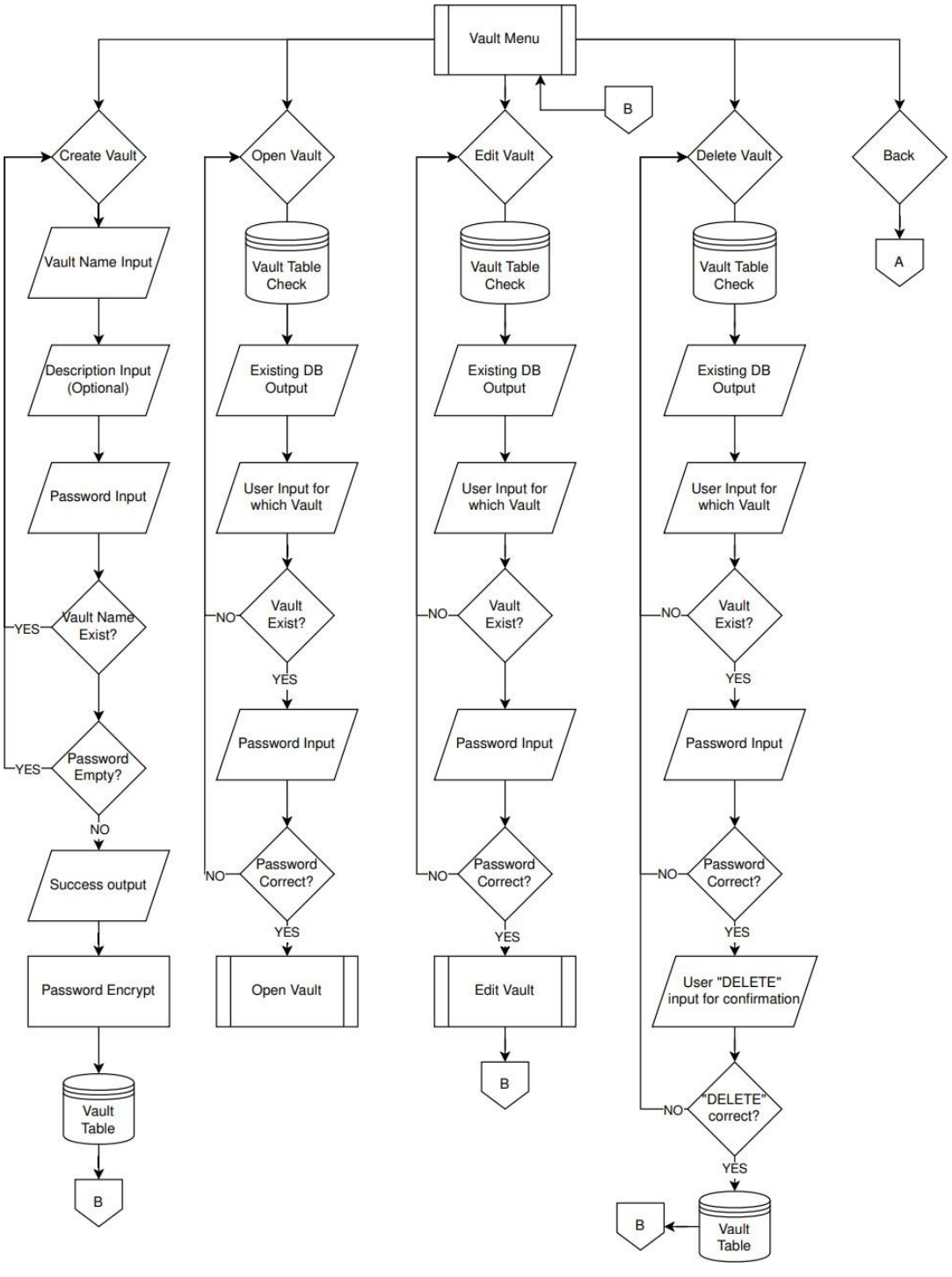
FLOW CHART

- **Register Subroutine**



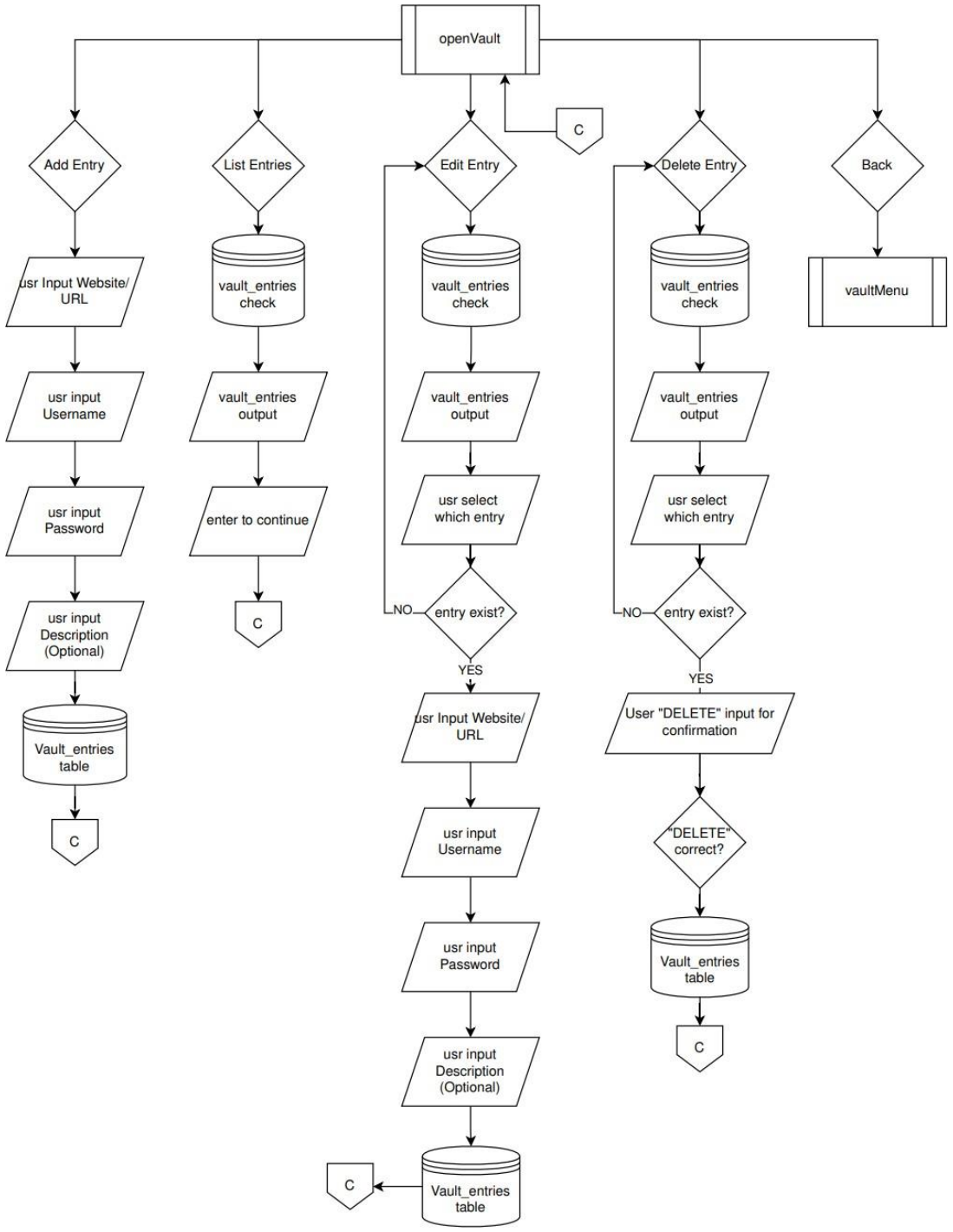
FLOW CHART

- **Vault Menu Subroutine**



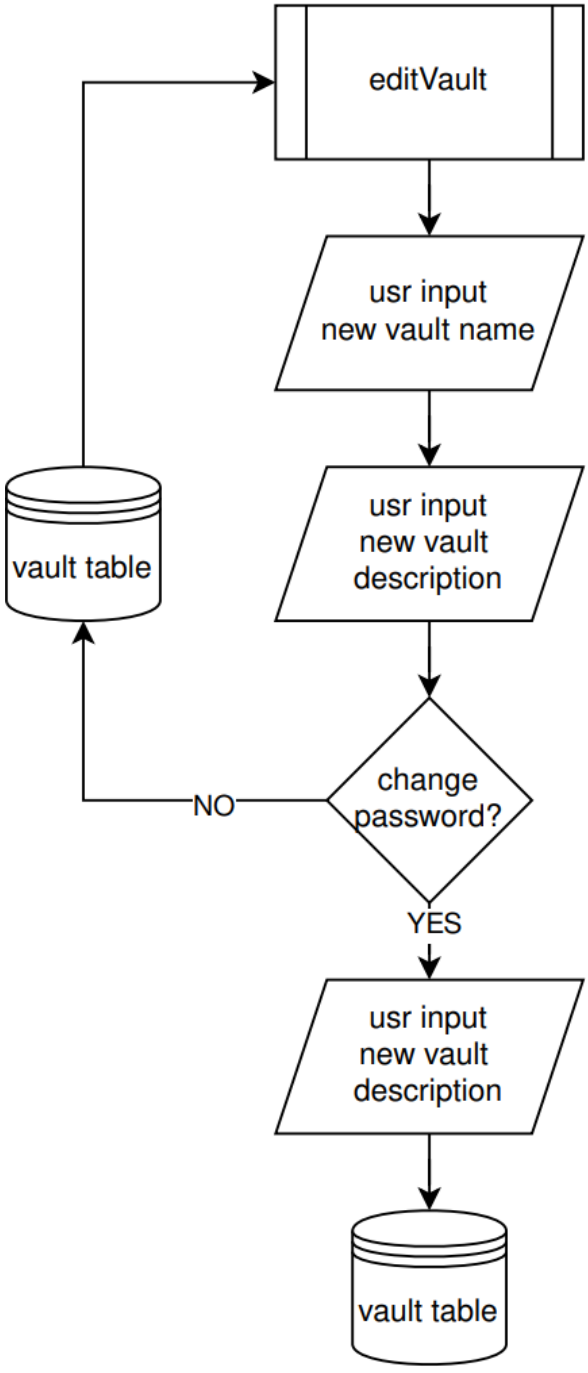
FLOW CHART

- **Open Vault Subroutine**



FLOW CHART

- **Edit Vault Subroutine**



ENTITY RELATIONSHIP DIAGRAM

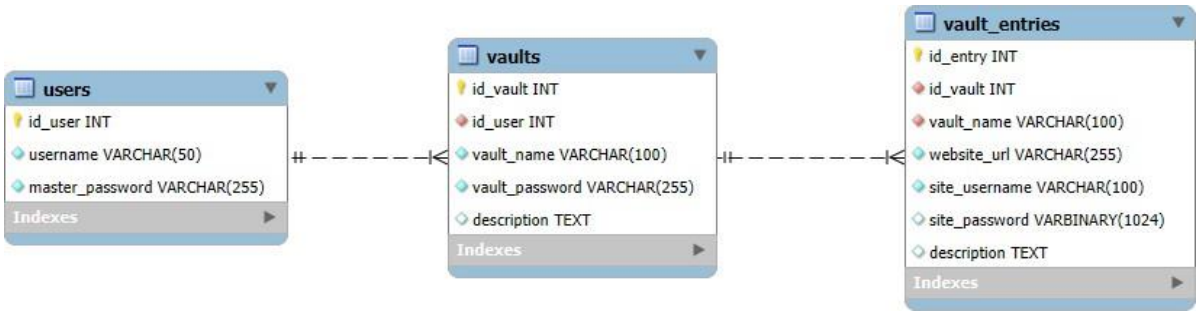


TABLE DESIGN

1. users

Column name	Data type	Size	Null	Key	Default	Notes	Delete rule
id_user	int	-	Not null	PK	Auto_increment	Primary key	-
username	varchar	50	Not null	unique	-	Unique across system	-
master_password	Varchar	255	Not null	-	-	Stores bcrypt hash	-

Indexes: UNIQUE KEY uq_user_username (username)

TABLE DESIGN

2. vaults

Column name	Data type	Size	Null	Key	Default	Notes	Delete rule
id_vault	int	-	Not null	PK	auto_increment	Primary Key	-
id_user	int	-	Not null	FK	-	References users(id_user)	ON DELETE CASCADE
vault_name	varchar	100	Not null	-	-	Display name for vault	-
vault_password	varchar	255	Not null	-	-	Bcrypt hash of vault master	-
description	TEXT	-	Null	-	-	Optional notes	-

Indexes:

- UNIQUE KEY uq_vaults_user_vaultname (id_user, vault_name) (unique vault name per user)
- UNIQUE KEY uq_vaults_idvault_username (id_vault, vault_name)

TABLE DESIGN

3. vault_entries

Column name	Data type	Size	Null	Key	Default	Notes	Delete rule
id_entry	int	-	Nut null	PK	auto_increment	Primary key	-
id_vault	int	-	Not null	FK	-	Part of composite FK to vaults	-
vault_name	varchar	100	Not null	FK	-	Part of composite FK to vaults	-
website_url	varchar	255	Not null	-	-	service/site or label url	-
site_username	varchar	100	Not null	-	-	Account username/email	-
site_password	VARBINARY	1024	Null	-	-	Encrypted/Encoded secret	-
description	TEXT	-	Null	-	-	Optional notes	-

Indexes&constraint:

- PRIMARY KEY (id_entry)
- KEY idx_entries_vault (id_vault)
- KEY fk_entries_vault_composite (id_vault, vault_name)
- FOREIGN KEY (id_vault, vault_name) REFERENCES vaults (id_vault, vault_name) ON DELETE CASCADE ON UPDATE CASCADE

DEMO

```
-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#
       PASSWORD MANAGER <||> (time: Friday, 26 September 2025)
-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#

=== MAIN MENU ===

1. Login
2. Register
3. Check Database Connection
4. Exit/logout
Please select an option:
```

First page, which is the main menu. Contains 4 option (login, register, check DB connection, and exit/logout)

```
-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#
       PASSWORD MANAGER <||> (time: Friday, 26 September 2025)
-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#

=== DATABASE CONNECTION CHECK ===

Checking database connection...'project_db'
Database connection successful.
Press Enter to return to the main menu...
```

Testing the database connection utils

DEMO

```
-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#
       PASSWORD MANAGER <||> (time: Friday, 26 September 2025)
-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#

=== USER LOGIN ===

Enter your username: demo
Enter your master password:

 Login successfully! Welcome, demo.
```

Login as demo:demo for testing

```
-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#
       PASSWORD MANAGER <||> (time: Friday, 26 September 2025)
                         Logged in as: demo
-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#

=== VAULT MENU ===

1. Create Vault
 2. Open vault
 3. Edit vault
 4. Delete vault
 5. Back to main menu
Please select an option:
```

Inside the vault menu with 5 options

DEMO

```
-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#
       PASSWORD MANAGER <||> (time: Friday, 26 September 2025)
                         Logged in as: demo
-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#

=== CREATE VAULT ===

Vault name: demo
Description (optional): vault for demonstration
Set vault password:
(^▽^)👍 Vault created.
```

Create vault function for creating the vault

```
-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#
       PASSWORD MANAGER <||> (time: Friday, 26 September 2025)
                         Logged in as: demo
-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#

=== OPEN VAULT ===

1. demo  —  vault for demonstration
Select number:
```

Open vault function showing the vault inside, if there is no vault it will error and return to the previous menu

DEMO

```
-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#
       PASSWORD MANAGER <||> (time: Friday, 26 September 2025)
                         Logged in as: demo
-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#

=== VAULT: demo ===

1. Add entry
2. List entries
3. Edit entry
4. Delete entry
5. Exit to vault menu
Select:
```

Inside the vault, after inputting the vault password to access the vault, user can add entry, list entry, edit entry, delete entry, and exit

```
-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#
       PASSWORD MANAGER <||> (time: Friday, 26 September 2025)
                         Logged in as: demo
-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#

=== VAULT: demo ===

1. Add entry
2. List entries
3. Edit entry
4. Delete entry
5. Exit to vault menu
Select: 1

Website/URL: demo
Site username: demo
Site password (hidden input):
Description (optional): demo entry
(^▽^)👍 Entry added.
```

Inputting entry

DEMO

```
-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#
       PASSWORD MANAGER <||> (time: Friday, 26 September 2025)
                          Logged in as: demo
-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#

=== VAULT: demo ===

1. Add entry
2. List entries
3. Edit entry
4. Delete entry
5. Exit to vault menu
Select: 2
1. demo | demo | b'demo' | demo entry

Website | Username | Password | Description

Enter to continue...
```

List entries, the password will not shown inside the database because of the varbinary type

```
-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#
       PASSWORD MANAGER <||> (time: Friday, 26 September 2025)
                          Logged in as: demo
-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#

=== VAULT: demo ===

1. Add entry
2. List entries
3. Edit entry
4. Delete entry
5. Exit to vault menu
Select: 3
1. demo | demo | b'demo' | demo entry

Website | Username | Password | Description
Entry number to edit:
```

Edit entry option to edit the entry that is already inputted, it will ask the user to input a new website, username, password, and description. If input is blank nothing changes

DEMO

```
-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#
      PASSWORD MANAGER <||> (time: Friday, 26 September 2025)
                       Logged in as: demo
-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#

=== VAULT: demo ===

1. Add entry
2. List entries
3. Edit entry
4. Delete entry
5. Exit to vault menu
Select: 4
1. demo | demo | b'demo' | demo entry

Website | Username | Password | Description
Entry number to delete: 1
Type 'DELETE' to confirm:
```

Delete entry function for delete existing entry, and it also ask the user to confirm by typing 'DELETE'.

And the edit and delete function for the vault is also the same logic as the entry, but the vault requires vault password for every update or delete

REQUIREMENTS

Hardware :

1. Asus Zenbook Duo

Operating System :

1. Windows 11

Software :

1. Vscode
2. Python
3. MySQL server

Dependencies :

1. Mysql-connector
2. Bcrypt

PROJECT FILE DETAILS

Num	Filename	Remarks
1	PASSWORD MANAGER SYSTEM 3CS2	Paper documentation of the project
2	main.py Entries_menu.py utils.py	Files contain the source code
3	Password Manager presentation.pptx	Presentation files