**CIS 476 Term Project**

**Document/Program Due Dates: Monday 11/27/23**

**Demo Time: 11/27/23 3pm**

This project is required for all students. You may finish the project all by yourself or team with others. 3 team members maximum.

In this project, you are required to design a password management software “MyPass”. MyPass can be a standalone or web-based software with the following functions/features. Bitwarden is a sample product.

**Functions/Features**

* Allow user to register an account (email address as username and a master password and three security questions) - Deniz
* Suggest strong password (password generator) - Deniz
* Weak master password warning - Deniz
* Once logs in, the user will have access to all sensitive data saved in the vault.
* MyPass has built-in data type in the vault, including Login, Credit Card, Identity, Secure Notes.
* MyPass allows user to create/modify/delete items in the vault. - Moe
* MyPass allows user to easily copy username/password and URL in Login data type
* MyPass allows user to easily copy credit card number and CVV in credit card data type - Moe
* Sensitive data such as username, password, credit card number, CVV, passport number, license number, social security number must be masked
* All masked data must be given an option to unmask. - Yehya
* Auto lock after x mins of inactivity - Yehya
* Auto delete copied sensitive data in clipboard after x mins - Moe

**Implementation**

* User Authentication and Encryption:
  + Implement the Singleton pattern to manage the user's session securely.

Class Diagram – Created by Moe

Source Code – Created by Moe

* Password Storage and Management:
  + Apply the Observer pattern to notify users in the events of weak password, credit card expiration, passport expiration, license expiration, etc.

Class Diagram Created by Yehya

Source Code – Created by Yehya

* User Interface and Interaction:
  + Implement the Mediator pattern to manage communication between various UI components.

Class Diagram Created by Moe

Source Code – Created by Moe

* Password Generation:
  + Apply the Builder pattern to create complex passwords with specific requirements (length, complexity).

Class Diagram – Created by Deniz

Source Code – Created by Deniz

* Data Mask and Unmask:
  + Implement the Proxy pattern to mask and unmask sensitive data.

Class Diagram – Created by Deniz

Source Code – Created by Yahya

* Master Password Recovery:
  + Apply the Chain of Responsibility pattern to create a secure process (using three security questions to build the chain) for recovering a forgotten master password.

Class Diagram – Created by Deniz

Source Code – Created by Yahya

* **Notes:** 
  + **Provide detailed design of patterns used in the solution via class diagram with mapping of pattern classes to the actual application classes.**
  + **The developed code must be thoroughly commented and synchronized with the model.**
* **Submission**
  + **Submit One .zip file that contains the followings**
    - **Source code**
    - **A report that includes class diagrams and their descriptions, database schema and descriptions, user-interface screen shots and descriptions, references.**