Part 2 code snippets

Password Vault with Master Key Security to securely store and retrieve passwords.

By: Zeina.ElSharkawy

Part B

Screenshots of the programs

```
y), function(a){"use strict"; function b(b){return this.each(function()) var
**Refb()))) year c=function(b){this.element=a(b)};c.VERSION="3.3.7",c.TRANSITION_DURATION=150,c.prot
 opdown-menu)"),d-b.data("target");if(d||(d-b.attr("href"),d-d&&d.replace(/.*(?=#[^\5]*$)/,"")),!
 st a"), f=a.Event("hide.bs.tab", {relatedTarget:b[0]}), g=a.Event("show.bs.tab", {relatedTarget:e[0]
 faultPrevented()){var h-a(d);this.activate(b.closest("li"),c),this.activate(h,h.parent(),functio
 \textit{rigger}(\{\textit{type:"shown.bs.tab"}, \textit{related} \textit{Target:e[0]})\})\}\}\}, \textit{c.prototype.activate=function(b,d,e)} \{\textit{function(b,d,e)}\}\}
 v > .active*).removeClass("active*).end().find('[data-toggle="tab"]').attr("aria-expanded",!1),|
 ia-expanded". 10), h?(b[0].offsetWidth, b.addClass("in")):b.removeClass("fade"), b.parent(".dropdov
 ().find('[data-toggle="tab"]').attr("aria-expanded",!0),e&&e()}var g=d.find("> .active"),h=e&&
 /e")||!!d.find("> .fade").length);g.length&&h?g.one("bsTransitionEnd",f).emulateTransitionEnd
 jvar d-a.fn.tab;a.fn.tab=b,a.fn.tab.Constructor=c,a.fn.tab.noConflict=function(){return a.fn.t
 'show")};a(document).on("click.bs.tab.data-api",'[data-toggle="tab"]',e).on("click.bs.tab.data
 se strict"; function b(b){return this.each(function(){var d=a(this),e=d.data("bs.affix"),f="ob"
 -typeof b\&\&e[b]()\}) var c=function(b,d){this.options=a.extend({},c.DEFAULTS,d),this.$target=a
 ,a.proxy(this.checkPosition,this)).on("click.bs.affix.data-api",a.proxy(this.checkPositionWi
 null, this.pinnedOffset=null, this.checkPosition()};c.VERSION="3.3.7",c.RESET="affix affix-top
:state=function(a,b,c,d){var e=this.$target.scrollTop(),f=this.$element.offset(),g=this.$targ
State=runction(a,u,c,u){van e=tnis.yeargec.seroirrop(y);
"bottom"==this.affixed)return null!=c?!(e+this.unpin<=f.top)&&"bottom":!(e+g<=a-d)&&"bottom"
!!=c&&e<c;"top":null!=d&&i+j>=a-d&&"bottom"},c.prototype.getPinnedOffset=function(){if(this
.RESET).addClass("affix");var a=this.$target.scrollTop(),b=this.$element.offset();return
WithEventLoop=function(){setTimeout(a.proxy(this.checkPosition,this) 100
```



Part B.1: Master Key Creation and Hashing

1) User Interface for Master Key Creation

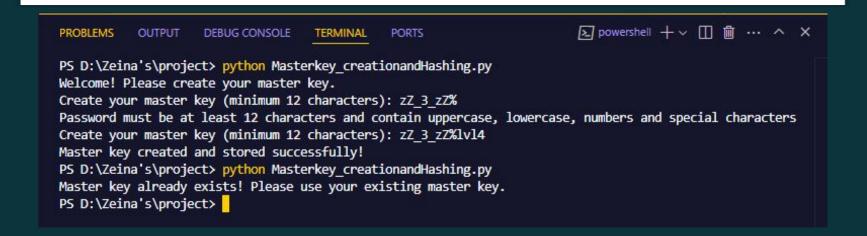
Code a prompt for the user to create a master key on first use.

2) Implement Hashing Function

- Hash the master key using a secure algorithm, optionally adding salt.
- → Use the SHA-256 algorithm for hashing

3 Secure Storage of Hashed Master Key

Store the hashed key securely, separate from the main password database.





Part B.2: Storing Passwords

1) Take User Input

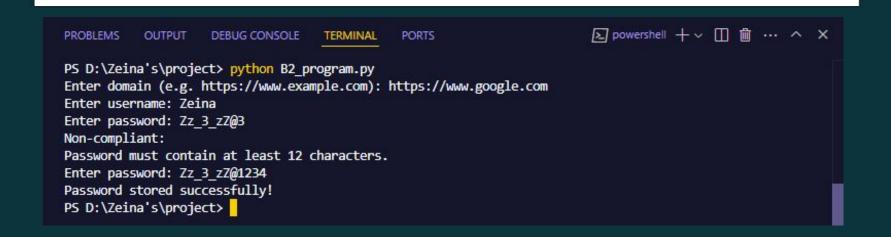
Implement python code to receive input - domain, username, and password.

2) Password Encryption

Encrypt passwords using a unique key for each entry.

3) Password Hashing and Storage

Hash the encrypted password and store all details securely in a password file.





Part B.3: Retrieving Passwords

1) Implement a function

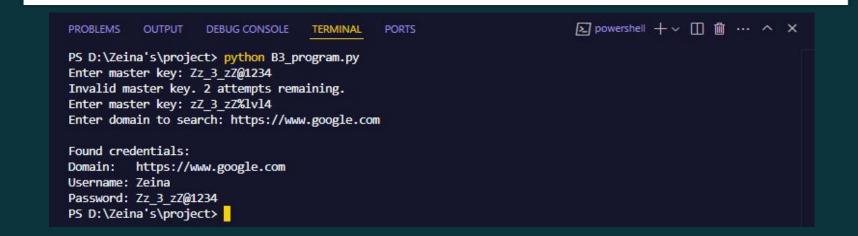
- Prompt the user for the master key upon request to retrieve a password.
- Hash the input and compare it with the stored master key hash for authentication.

2) Upon successful authentication

- Search the password file for the requested domain.
- Decrypt the password using the stored encryption key.

3) Password Hashing and Storage

Display the decrypted password to the user





Part B.4: Shell scripting

1)

a shell script that prompts the user for a domain name and then executes a Python script to retrieve a password for that domain.

2)

a shell script that authenticates the master key by calling the Python script written in the first part of the project, that performs the actual authentication process.

Retrieving an existing password command:

```
OUTPUT DEBUG CONSOLE TERMINAL
 Mohamed@DESKTOP-3C0KC3S MINGW64 /d/Zeina's/project
$ ./B4 program.sh
 Welcome to the Password Manager
 Please choose an option:
 1. Add a new password
 2. Retrieve an existing password
 3. Exit
 Enter your choice [1-3]: 2
 Enter the domain name for which to retrieve the password: https://www.google.com
 Retrieving password for https://www.google.com...
 Enter master key: zZ 3 zZ@1234
 Invalid master key. 2 attempts remaining.
 Enter master key: Zz 3 zZ@1234
 Invalid master key. 1 attempts remaining.
 Enter master key: zZ 3 zZ%lvl4
 Enter domain to search: https://www.google.com
 Found credentials:
 Domain: https://www.google.com
 Username: Zeina
 Password: Zz 3 zZ@1234
 Password retrieval completed.
```



Part B.4: Shell scripting

1)

a shell script that prompts the user for a domain name and then executes a Python script to retrieve a password for that domain.

2)

a shell script that authenticates the master key by calling the Python script written in the first part of the project, that performs the actual authentication process.

```
Adding a new password command:
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
 Mohamed@DESKTOP-3C0KC3S MINGW64 /d/Zeina's/project
$ ./B4 program.sh
 Welcome to the Password Manager
 Please choose an option:
 1. Add a new password
 2. Retrieve an existing password
 3. Exit
 Enter your choice [1-3]: 1
 Starting the process to add a new password...
 Enter domain (e.g. https://www.example.com): https://www.udacity.com
 Enter username: Zeina
 Enter password: haha
 Non-compliant:
 Password must contain at least 12 characters.
 Password must contain at least 1 uppercase character.
 Password must contain at least 1 digit.
 Password must contain at least 1 special character.
 Enter password: Zaserfdc3!
 Non-compliant:
 Password must contain at least 12 characters.
 Enter password: Zaswerdfxcv!!3
 Password stored successfully!
 New password entry added successfully.
```



Part B.4: Shell scripting

1)

a shell script that prompts the user for a domain name and then executes a Python script to retrieve a password for that domain.

2)

a shell script that authenticates the master key by calling the Python script written in the first part of the project, that performs the actual authentication process.

Exiting command:

Mohamed@DESKTOP-3C0KC3S MINGW64 /d/Zeina's/project

- \$./B4 program.sh
- Welcome to the Password Manager Please choose an option:
- 1. Add a new password
- 2. Retrieve an existing password
- 3. Exit

Enter your choice [1-3]: 3 Exiting.