# Password Manager User Guide

Set Up

1. Download the application code from the Git repository.

2. To build and deploy the application, please use Visual Studio with ASP.NET Core 6.0 support. If you already have Visual Studio with ASP.NET Core 6.0 support, please skip to step 4.

Visual Studio 2022 Community Edition can be downloaded from the below link:

https://visualstudio.microsoft.com/vs/

Once you run the set up the file, please check the "ASP.NET and web development" option as indicated in the screenshot below:
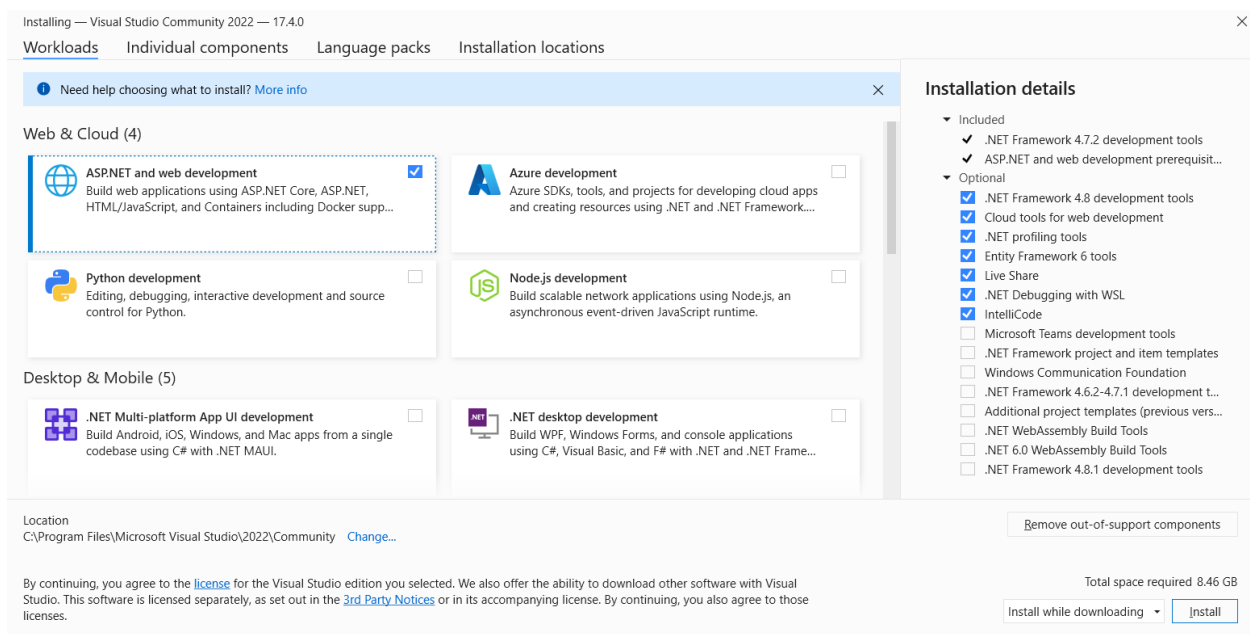
3. Once Visual Studio is installed, you may be prompted to sign in. You can choose to skip this step by selecting "Skip this for now."

4. Before opening the application solution in Visual Studio, please create a new folder called "**Passwords**" in the ENPM809WFall2022Project-sdasari\PM\PM folder.

5. Open the application solution file PM.sln present in the ENPM809WFall2022Project-sdasari\PM folder using Visual Studio.

6. Once the solution file has been loaded completely in Visual Studio, open the Nuget Packet Manager Console using **Tools -> Nuget Packet Manager -> Packet Manager Console**. After the console is up, run the command **Update-Database** as shown in the screenshot below:

Once the command is executed successfully, you should get the output shown in the screenshot below:

7. The application uses SMTP to send emails to users of the application. The SMTP credentials need to be provided in appsettings.json. You need to provide the credentials of an email account for which SMTP has been enabled. An Outlook email account is recommended for this purpose.

To obtain the SMTP details of an Outlook account, follow the steps below:

a) Create and log in to the Outlook account.
b) Navigate to Settings as shown in the screenshot below:

c) Search for IMAP in Settings. You should see an option called POP and IMAP.
d) After selecting the POP and IMAP option, select "Yes" for the "Let devices and apps use POP" option as shown below:

e) The SMTP server name and port number to be used can be obtained from the SMTP setting at the bottom

f) Use the obtained details to fill the SMTPConfiguration section in the ENPM809WFall2022Project-sdasari\PM\PM\appsettings.json file.

```
"SMTPConfiguration": {
    "From": "your_email@outlook.com",
    "SmtpServer": "your.smtp.server",
    "Port": 587,
    "Username": "your_email@outlook.com",
    "Password": "YourPassword"
},
```

8. You can now try to build and run the application by clicking the green play button. Alternatively, you could build the solution by navigating to **Build => Build Solution** and then running it by navigating to **Debug => Start Debugging**. When building and running the application for the first time, you may encounter SSL related Warnings as shown in the screenshots below. You can choose yes for all the prompts.

Trust ASP.NET Core SSL Certificate ✕

This project is configured to use SSL. To avoid SSL warnings in the browser you can choose to trust the self-signed certificate that ASP.NET Core has generated.

Would you like to trust the ASP.NET Core SSL certificate?

☐ Don't ask me again

Yes   No

Security Warning ✕

You are about to install a certificate from a certification authority (CA) claiming to represent:

localhost

Windows cannot validate that the certificate is actually from "localhost". You should confirm its origin by contacting "localhost". The following number will assist you in this process:

Thumbprint (sha1): 1B5607C4 6E189840 D93C47D9 AED56028 0033CF4C

Warning:
If you install this root certificate, Windows will automatically trust any certificate issued by this CA. Installing a certificate with an unconfirmed thumbprint is a security risk. If you click "Yes" you acknowledge this risk.

Do you want to install this certificate?

Yes   No

9. You can now begin using the application.

Note: For using the REST API features of the application, any client can be used. The user guide will make use of Postman to showcase the functionality of the application. Postman can be downloaded from the below link:
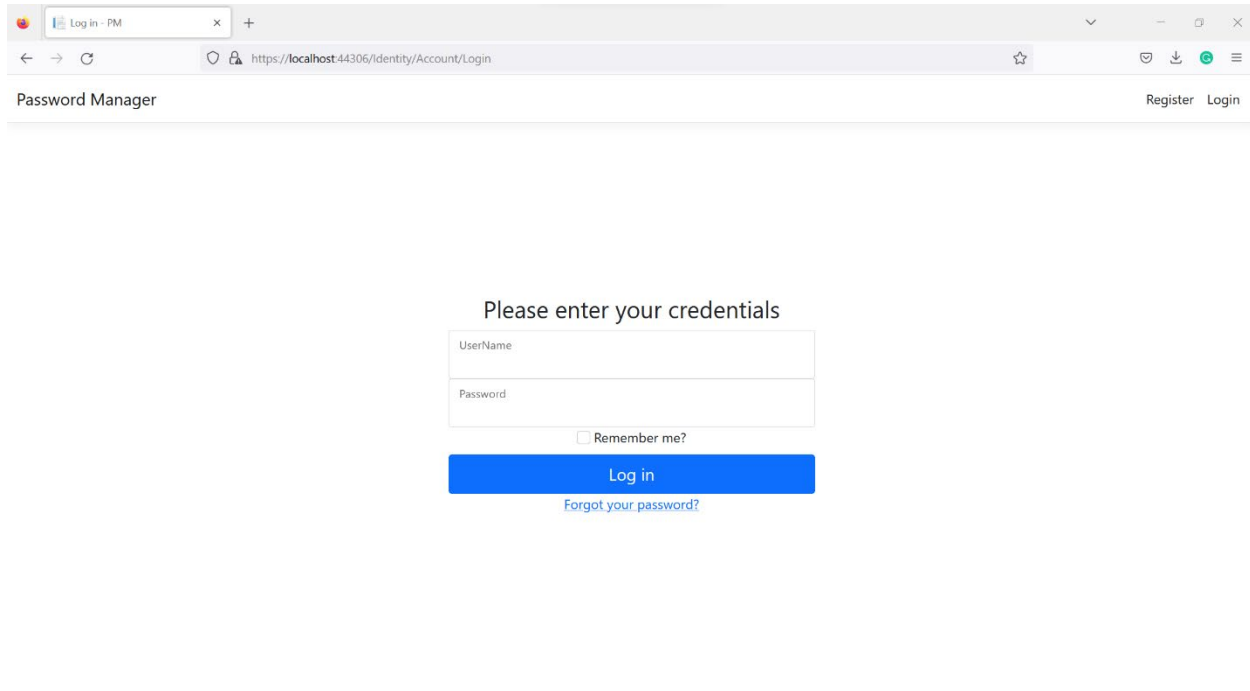
https://www.postman.com/downloads/

To run the unit tests for the application, right click on **PM_Test** and then select run tests. The result of the tests is displayed below.
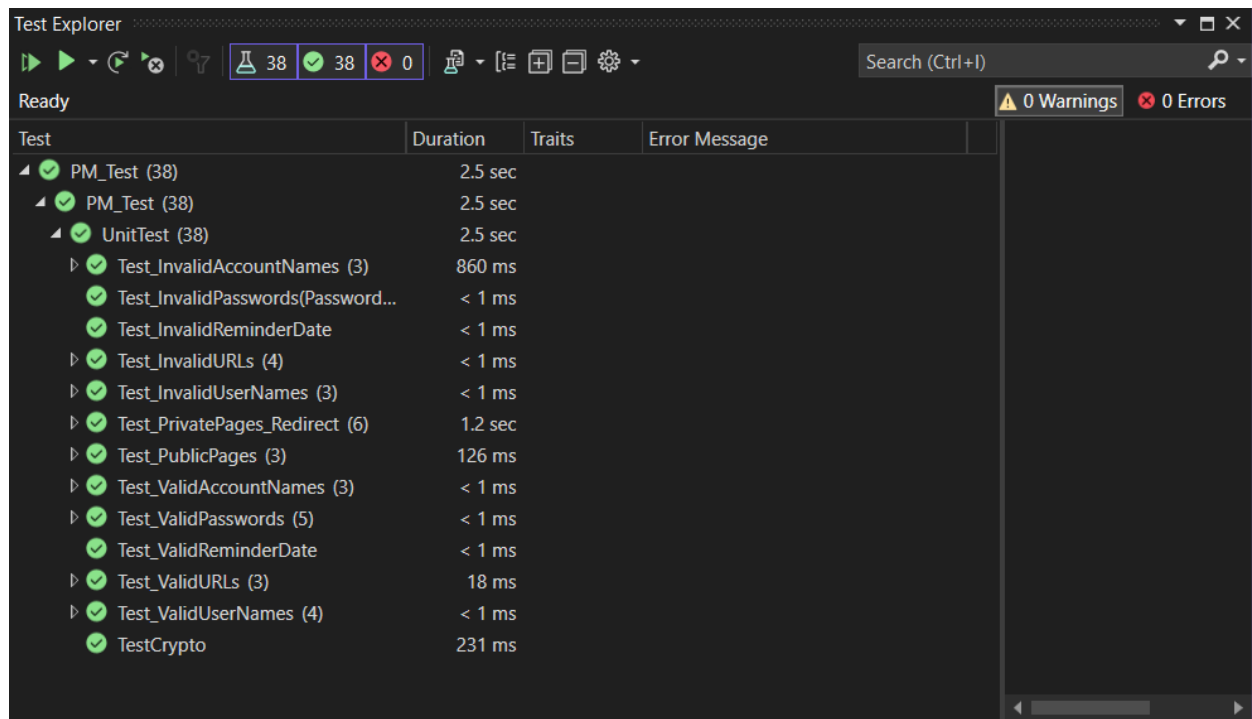
Usage

The application can be used by 2 types of users: a user who accesses their own account through a web interface, and a browser who uses the application on behalf of a user by making use of the REST APIs exposed by the application. To illustrate the usage of the REST APIs that a browser would utilize through a browser extension, this User Guide uses Postman.

For the web users, the landing page of the application is the login screen as shown below.
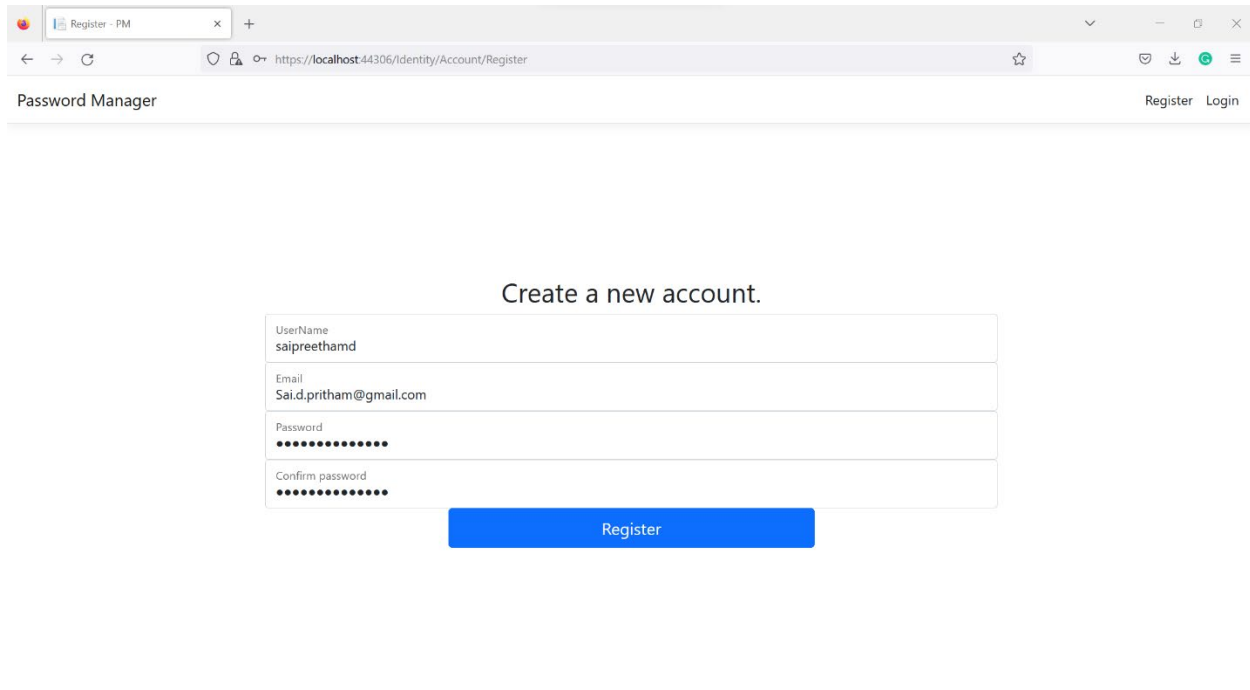
Before using the application, a user needs to create an account. This can be done by selecting the "Register" option in the navigation bar towards the top right of the web page. The registration page is shown below.

Users can fill in the registration details and click the Register button. If all the details are valid, they will be redirected to the page shown below and they will receive an email prompting them to confirm their email address.
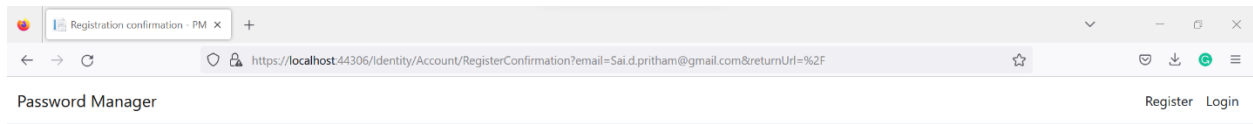
Without confirming the email address, users will not be able to log in to their accounts. Below is an example of the confirmation email.
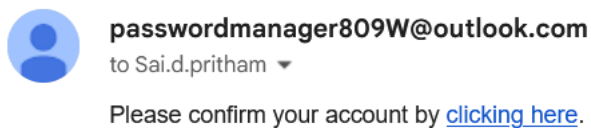
Upon clicking the link in the email, the user will receive confirmation that their email address is confirmed as shown in the screenshot below.

Users can now login to their account by using the credentials provided when registering. To avoid logging in repeatedly, they can select the Remember Me option provided.
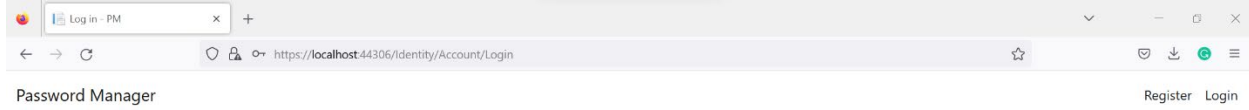

Once a user logs in successfully, they can view the details of the passwords they have saved. When they login for the first time, they can see the below screen.
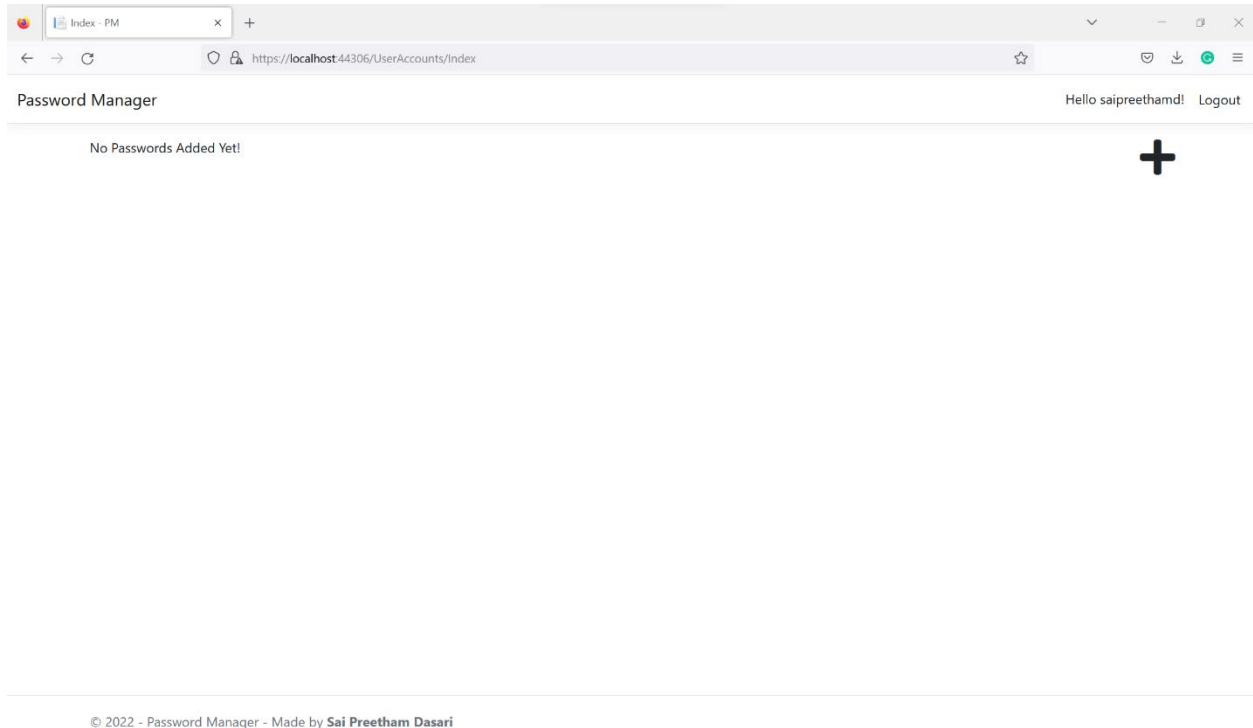
Users can add new password entries by using the add icon on the top right of the screen. The page to add password entries is shown below.
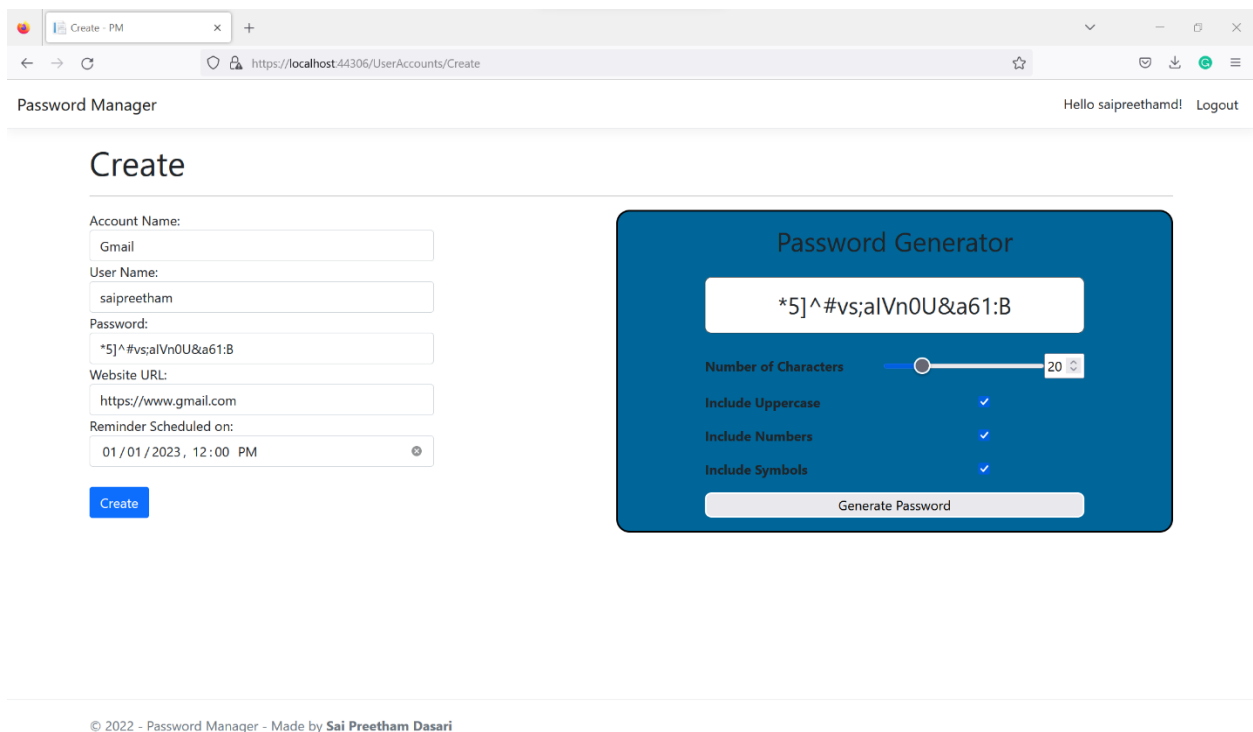
When creating new password entries, users can make use of the password generator to generate random passwords of up to 50 characters in length while also choosing the types of characters to be included in the generated password. Users can also set a reminder date to remind them to update the password for that account. **The URLs which are entered by the user must also begin with http or https**. After creating an entry, users can view their stored passwords in their home page after logging in, as seen below.
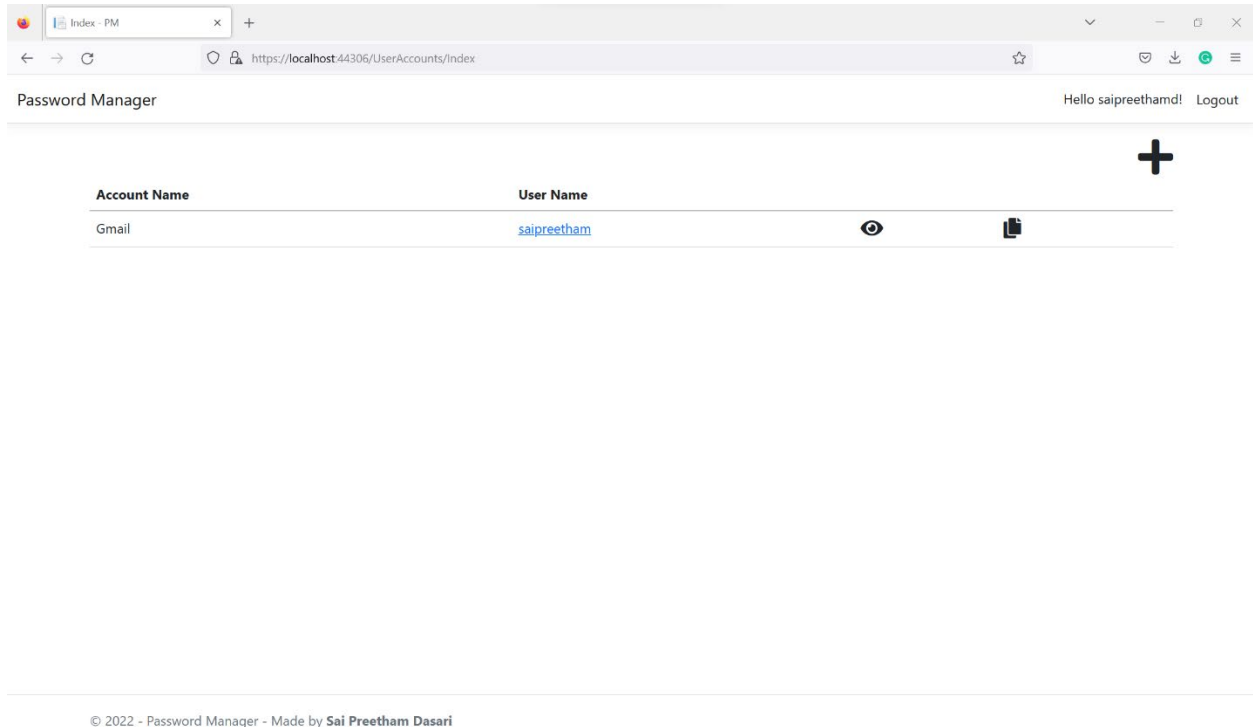

The users can directly view or copy their passwords using the respective icons from the above view without navigating to a different page.
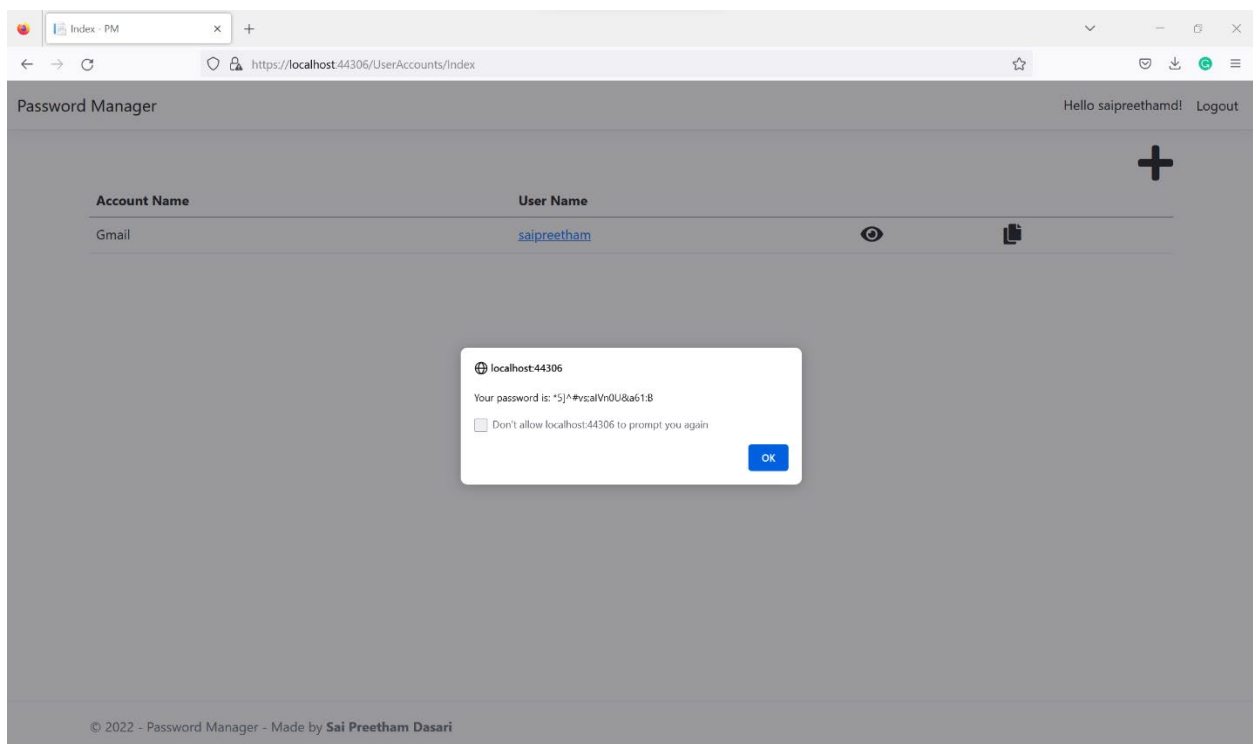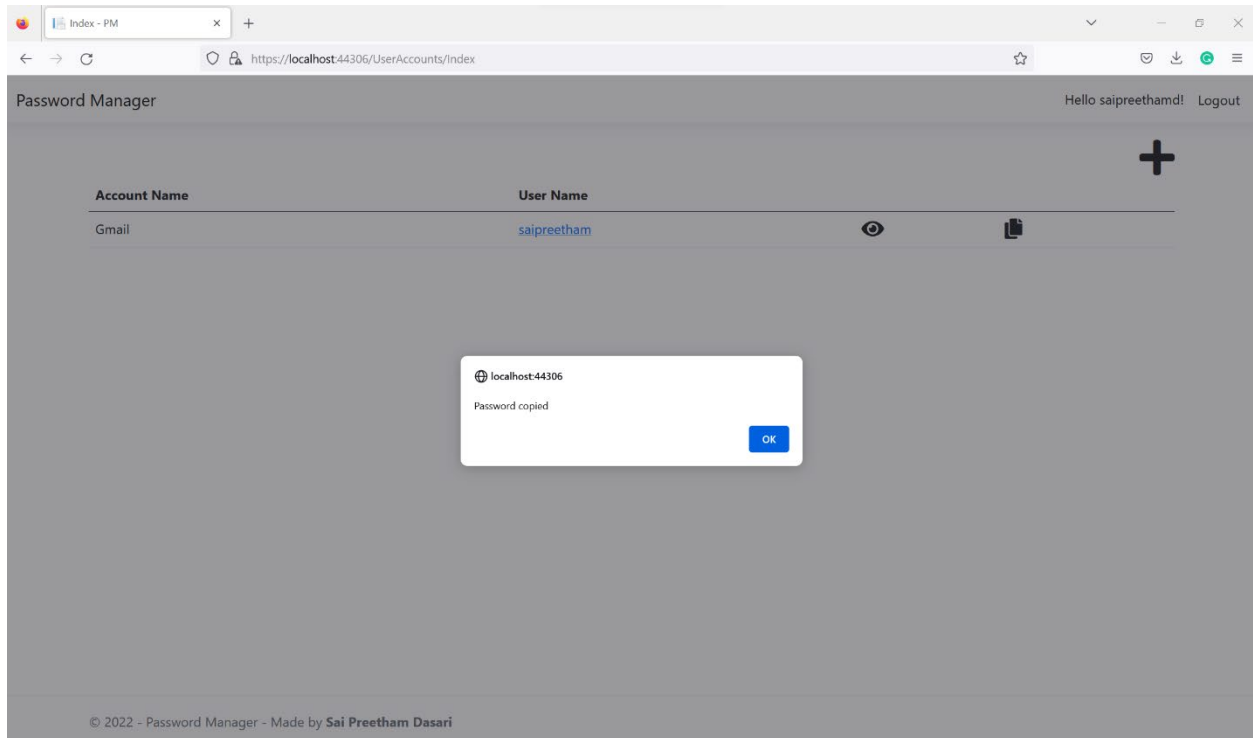
To view the complete details of an entry, users can click on the username for the desired entry using which they are redirected to a page displaying all the details of that account.
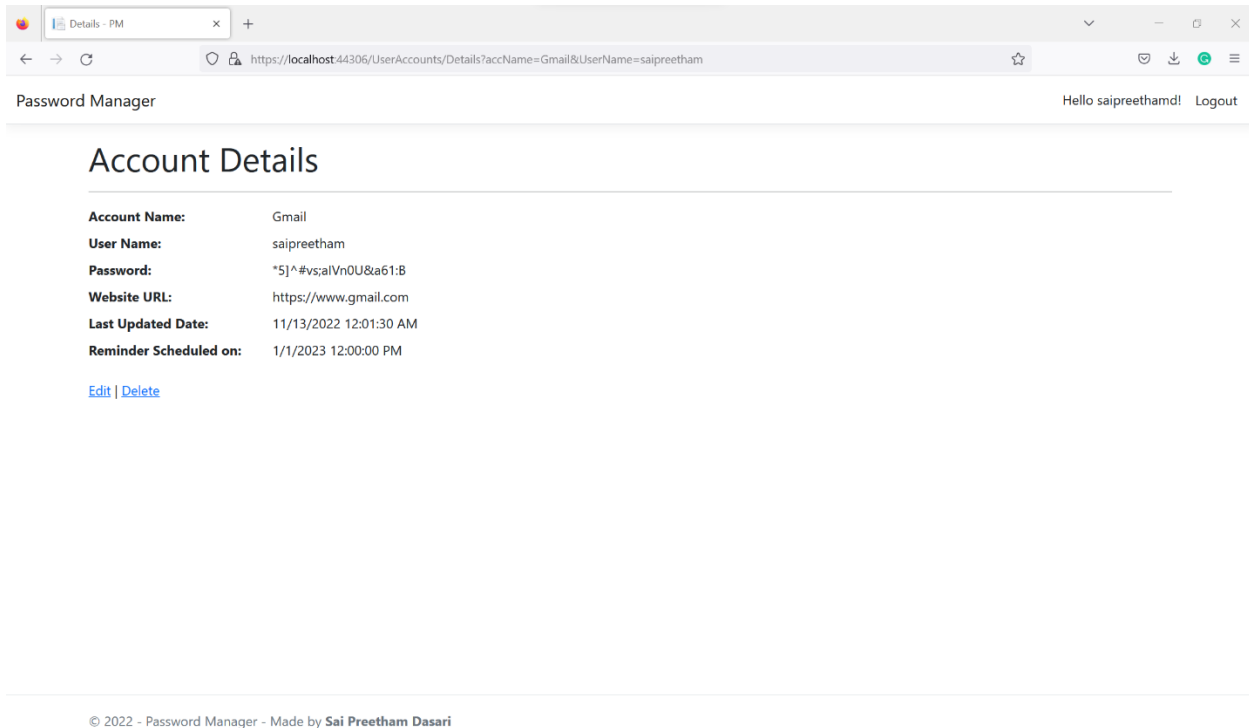
Through the edit option present on this page, users can edit the password, URL and reminder date of the account.
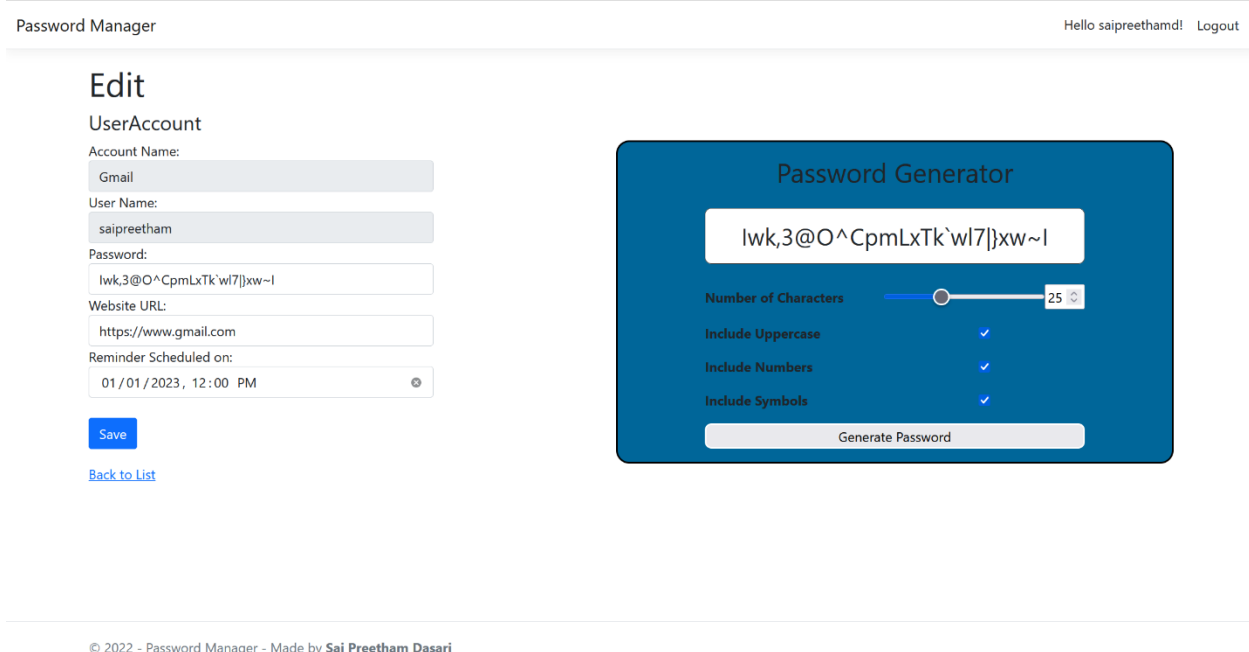
From the above figure, you can observe that users cannot edit the account name and the username when trying to edit an account. Once the changes have been made successfully, users are redirected to the details of the account.
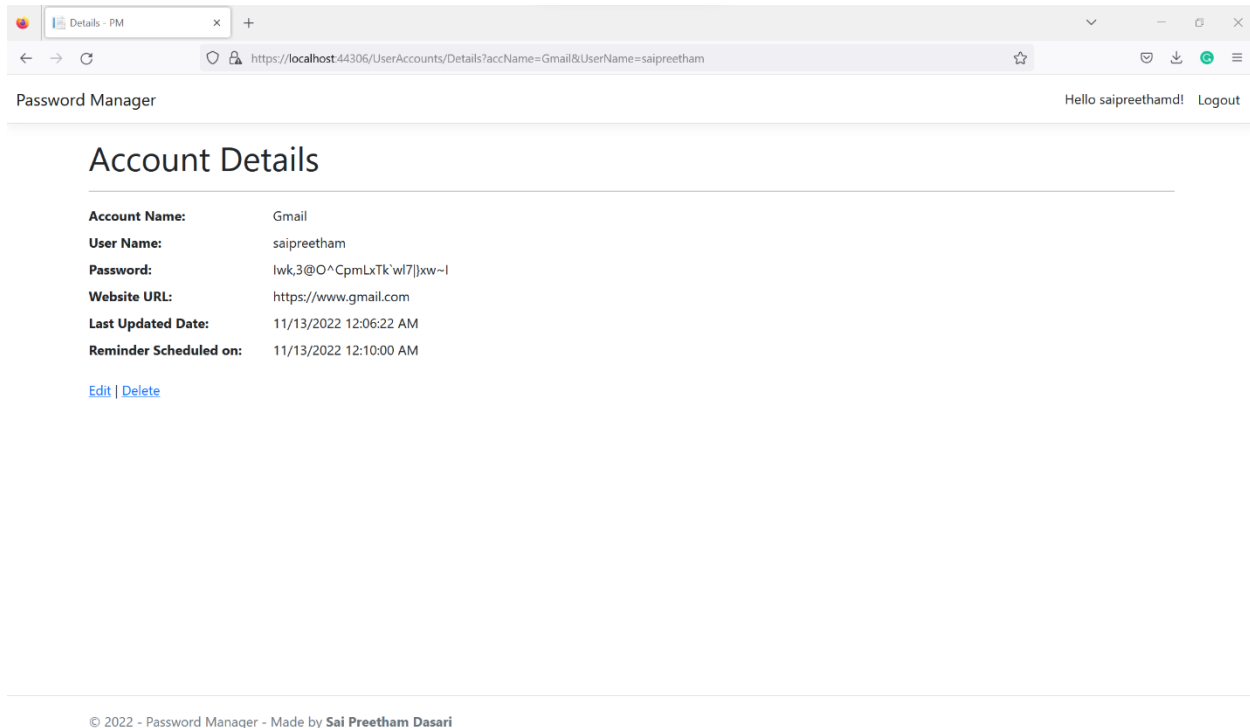
To manage their Password Manger account, users can click on their username in the navigation bar in the top right of the web page.
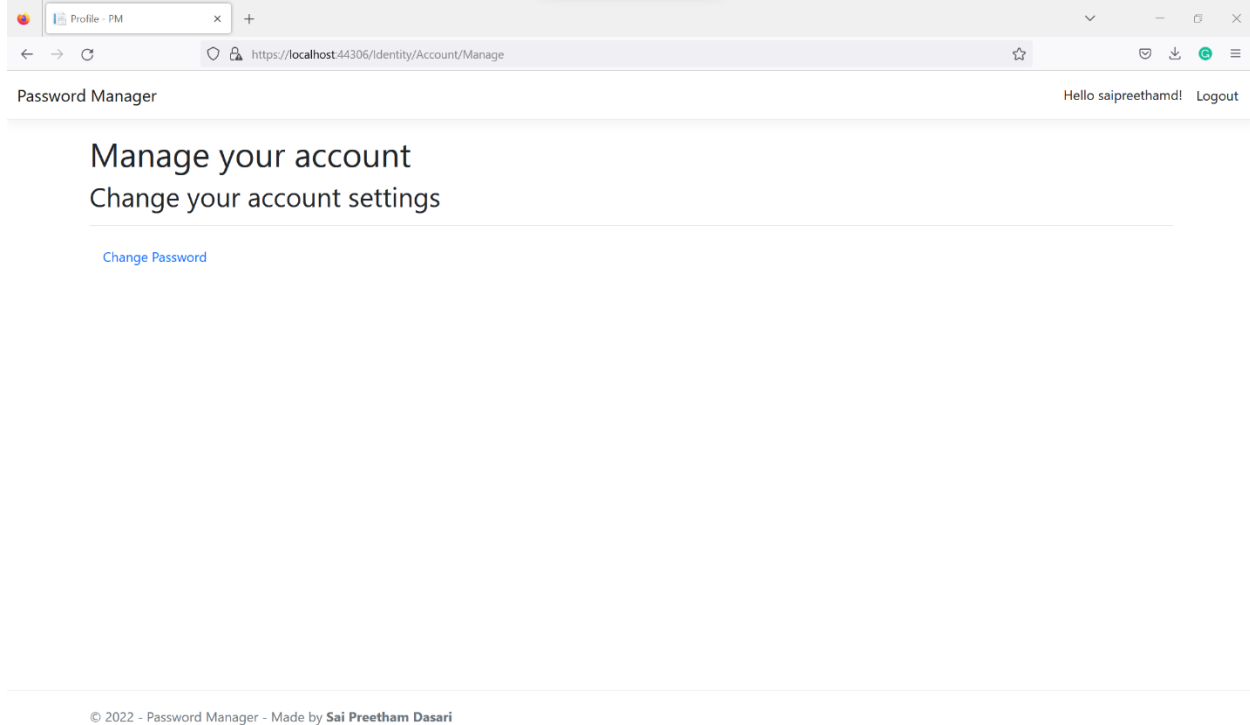
Currently, only the password change functionality is supported.

To change their password successfully, users must provide their current password as well as their new password. Upon changing their password successfully, they are notified of the same as shown below.

If the reminder date for any of the accounts has passed, an exclamation mark is present next to the password entry in the user's dashboard signifying that they should update their password for that account.

The delete entry feature can be accessed from the details page of a particular password entry. Upon clicking the delete option, the user is redirected to another page which asks for confirmation from the user before deleting the entry.
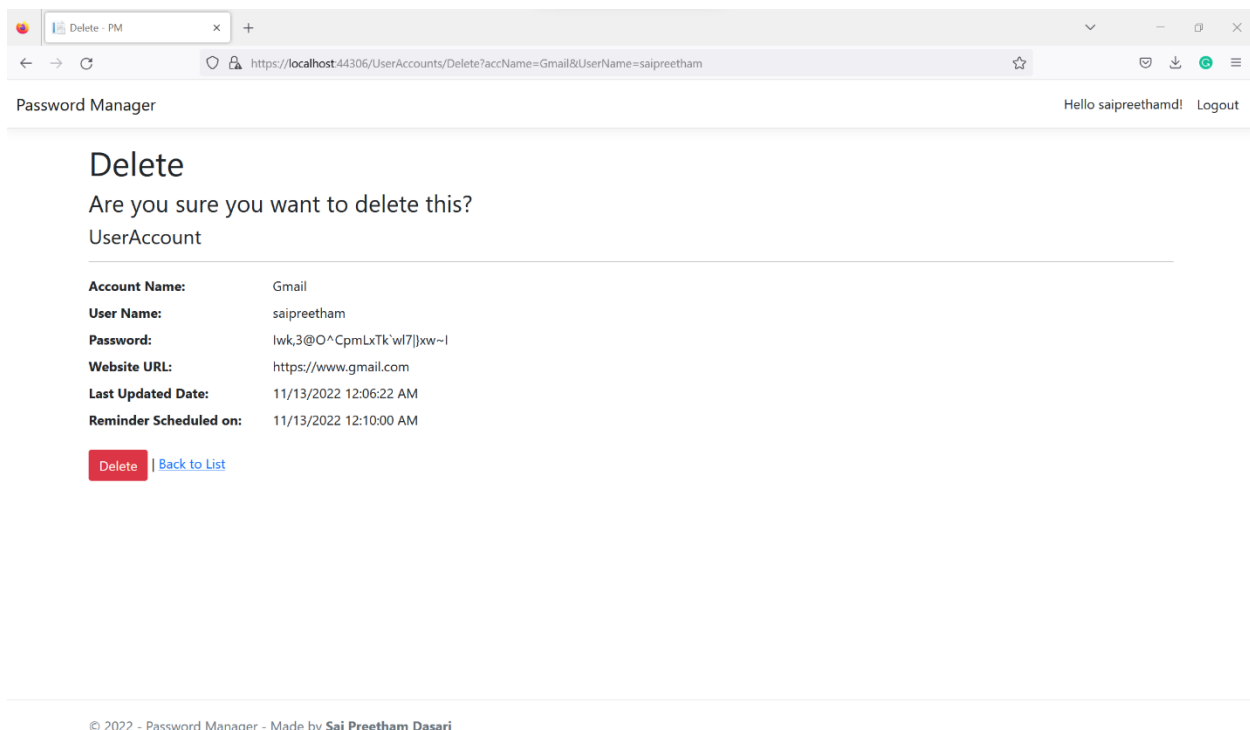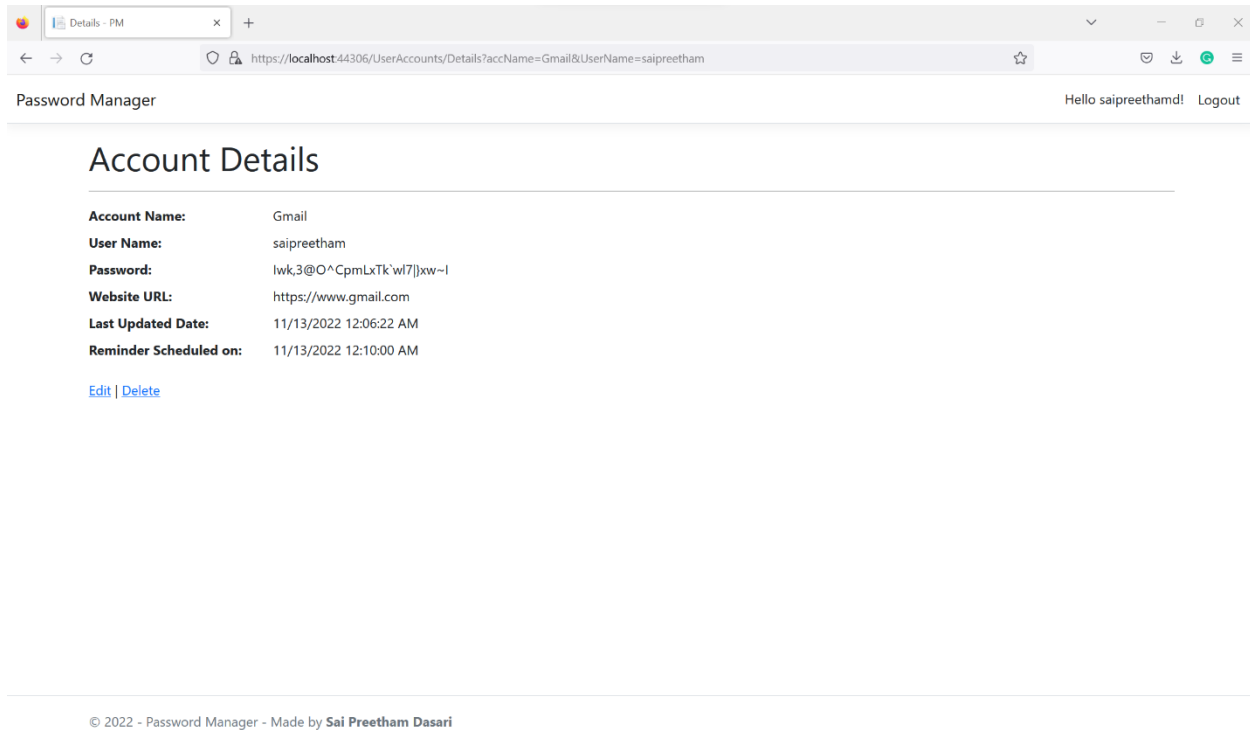
Once the delete option is selected on the page above, the password entry is deleted from the database.

In case a user has forgotten their master password for the Password Manager account, they can make use of the Forgot Password functionality using the option on the login page.

Upon clicking "Forgot your password?", a user is redirected to a page which prompts them to enter their email ID to which the application would then send an email with a password reset link.

Once the reset password button is clicked, the email is sent to the email ID of the registered user.
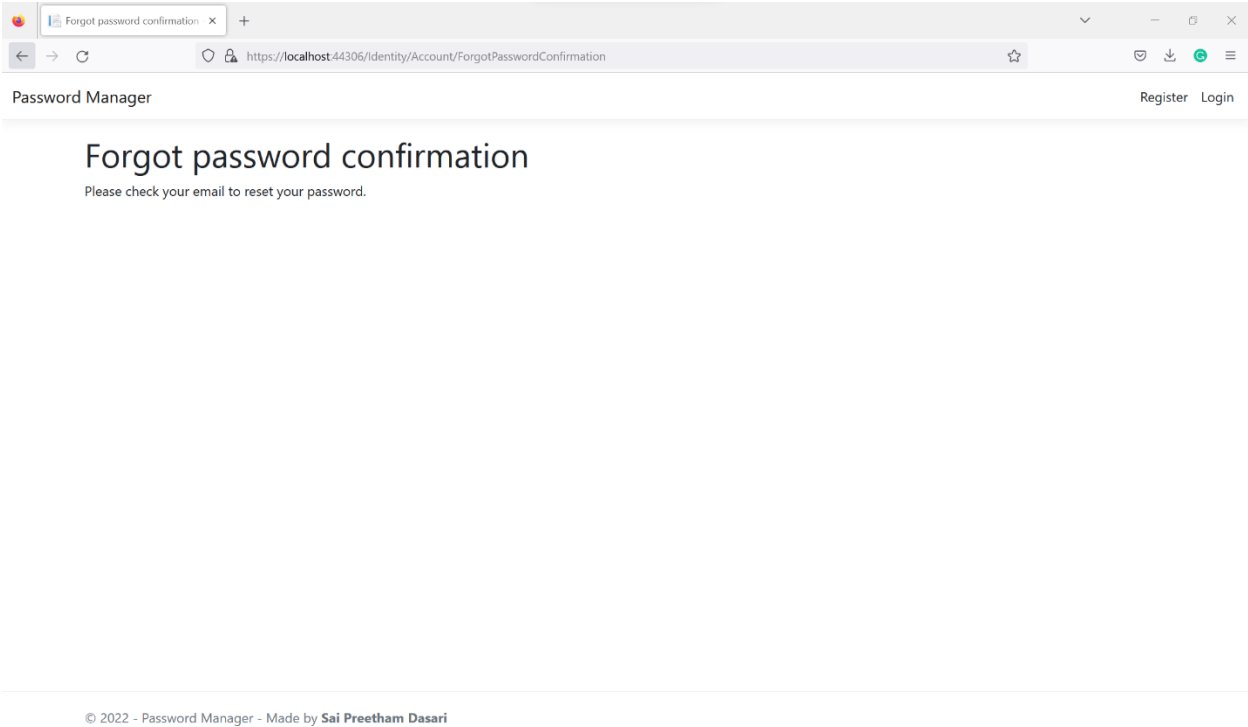
Below is an example of the password reset email.

Upon clicking the link in the email, the user is redirected to the application page asking for the email and the new password of the user.

If the email ID and the password are entered correctly, the password is reset successfully, and the user can log in to their accounts using the new password.

For the REST APIs, the functionality is illustrated using Postman, but any client can be used in a similar way.
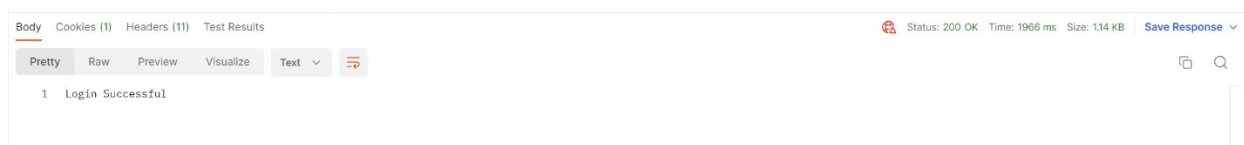
The URLs that are accessible via REST API are:

1) https://{IP Address}:{Port Number}/browser/Login
2) https://{IP Address}:{Port Number}/browser/getUserNames
3) https://{IP Address}:{Port Number}/browser/getPassword
4) https://{IP Address}:{Port Number}/browser/getUserNamesFromURL
5) https://{IP Address}:{Port Number}/browser/getPasswordUsingURL

IP Address refers to the IP address of the machine on which the application is deployed. IP Address can be replaced by localhost if the application is being used locally. Port number depends on which port the application is running on i.e. if the application is running on IIS Express, the port number would be 44306. Otherwise, the port number would be 7019.

Of the 5 URLs mentioned in the above list, only the first one i.e., Login, is to be accessed using a POST request. The username and password are to be sent to the server as part of Authorization Header, using Basic Authentication. No other parameters are required.
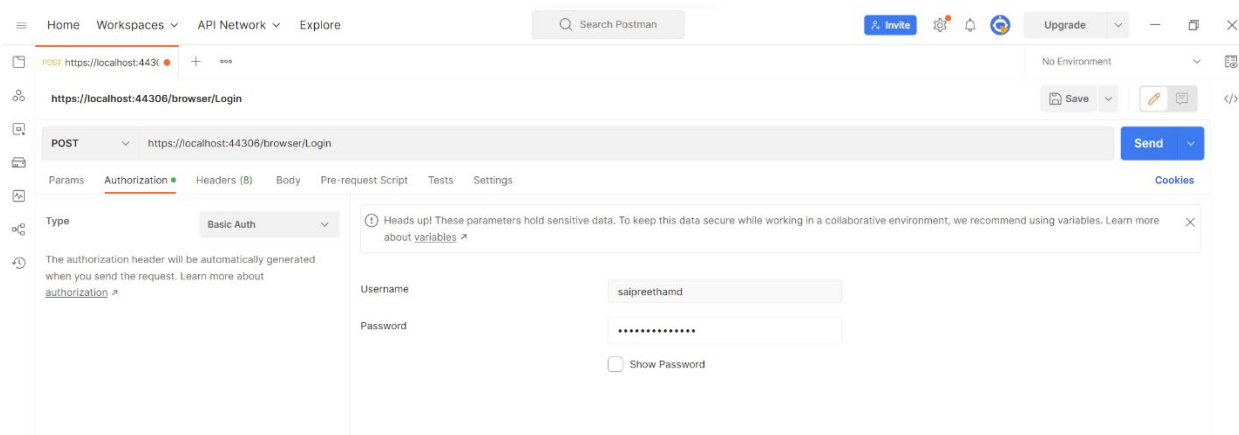

If the credentials were valid, the server returns a 200 OK response with the string "Login Successful".


In the response, the server also returns a cookie, which is to be used in future requests by the client to the server.

To get the usernames for a particular account name, a GET request to the 2nd URL (https://{IP Address}:{Port Number}/browser/getUserNames) is required. The account name for which the usernames are required is sent as a parameter (parameter key **accountName**) to the server. Along with the account name, the cookie returned by the server during the Login phase is also required to be sent to the server as shown below.
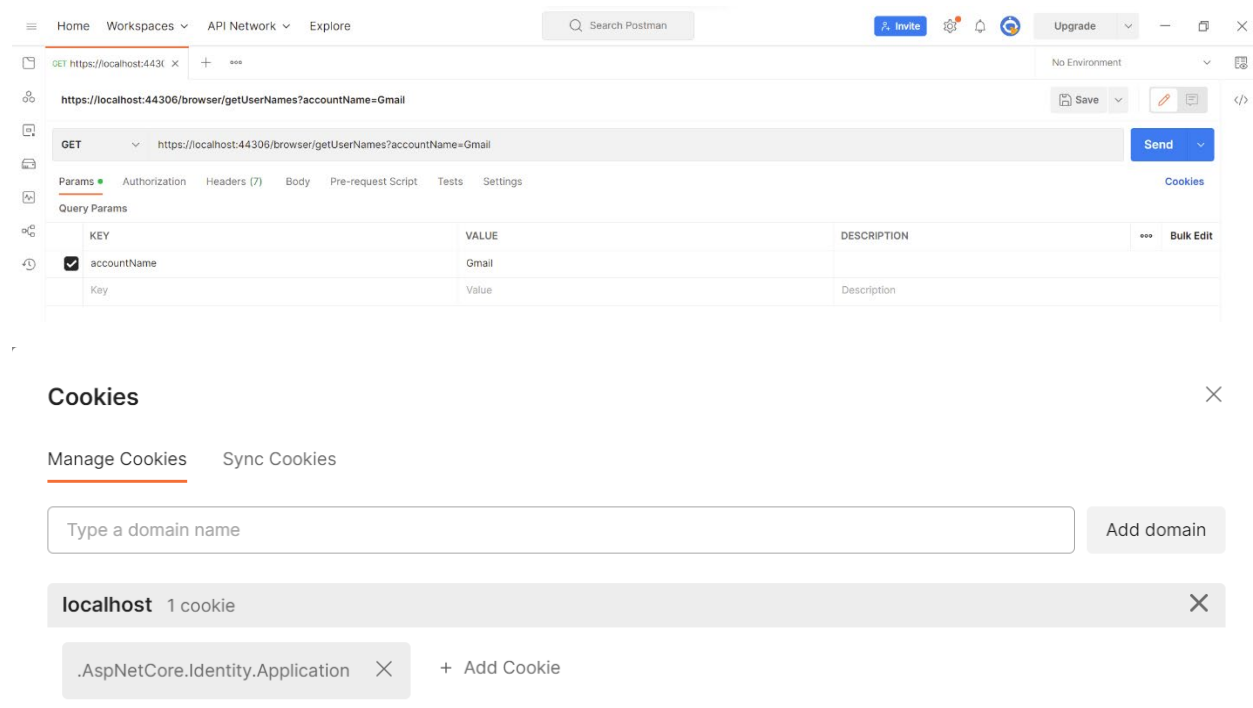
If the user has an entry corresponding to the account name sent in the parameters and the cookie is valid, the list of usernames for that account name is returned as a response.
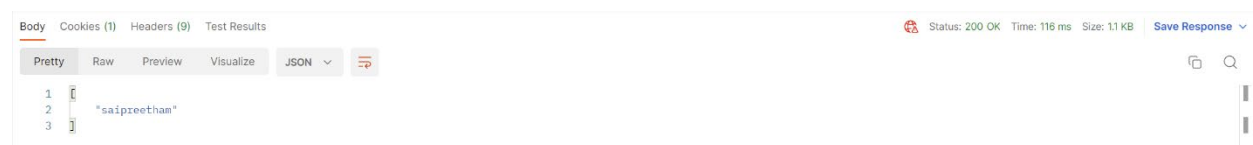

To obtain the password for a particular account name and username pair, a GET request to the 3rd URL (https://{IP Address}:{Port Number}/browser/getPassword) is made. Both the account name and the username are sent as parameters with parameter keys as **accountName** and **userName**, along with the cookie.
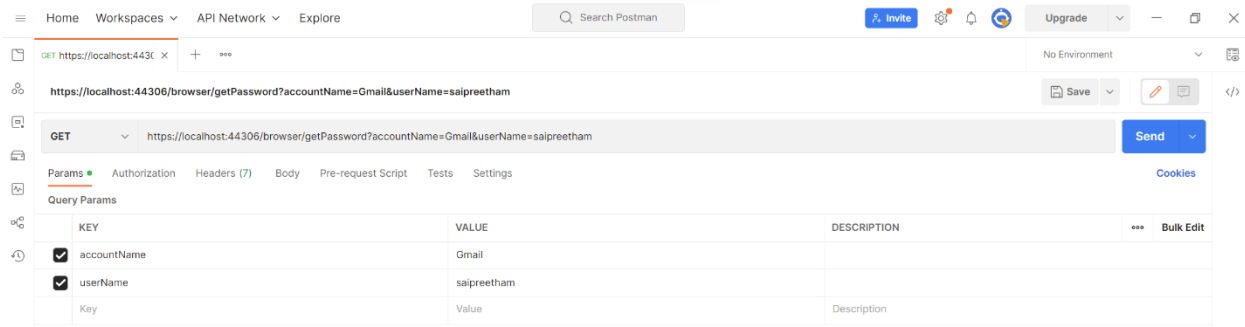
If the account name and username pair exists and the cookie is still valid, then the password corresponding to the pair is returned to the client.


Usernames and passwords can also be requested using the URL instead of the account name (since browsers would have access to the URL to which the password needs to be supplied). To obtain the usernames based on a specific URL, a GET request to https://{IP Address}:{Port Number}/browser/getUserNamesFromURL is made. The URL is added as a parameter in the request with the parameter key as **URL** along with the cookie.
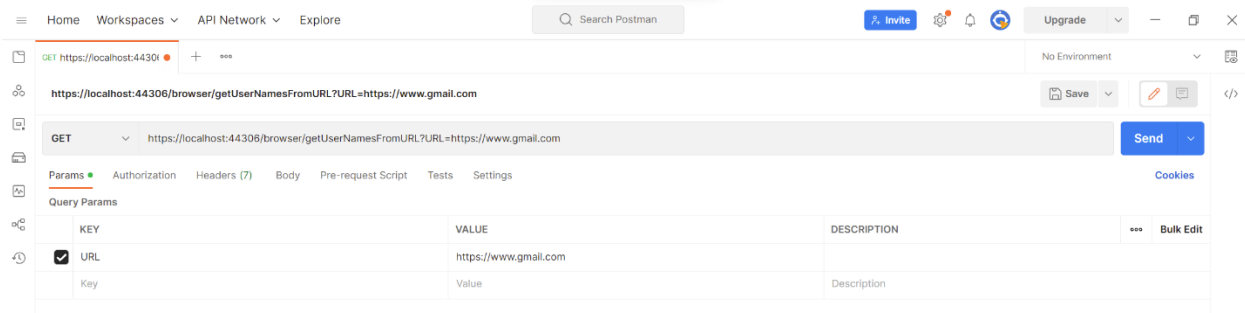
The server responds with all the usernames which are associated with the provided URL.
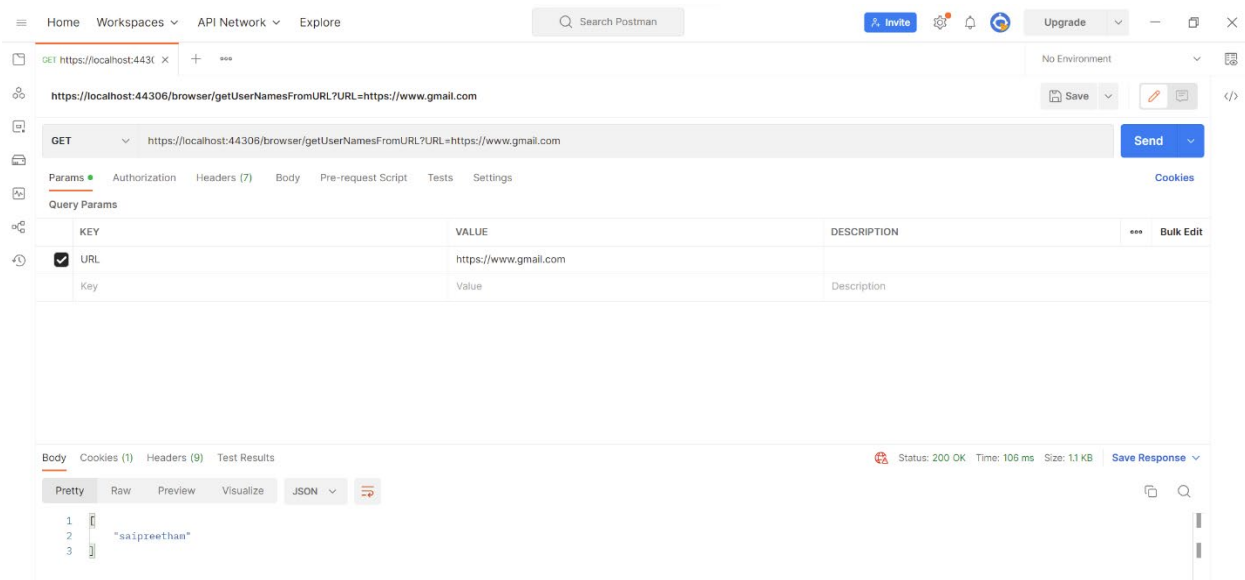
To obtain the password for a URL and username pair, a GET request needs to be made to https://{IP Address}:{Port Number}/browser/getPasswordUsingURL. The URL and the username need to be passed as parameters with keys as **URL** and **userName** along with cookie.
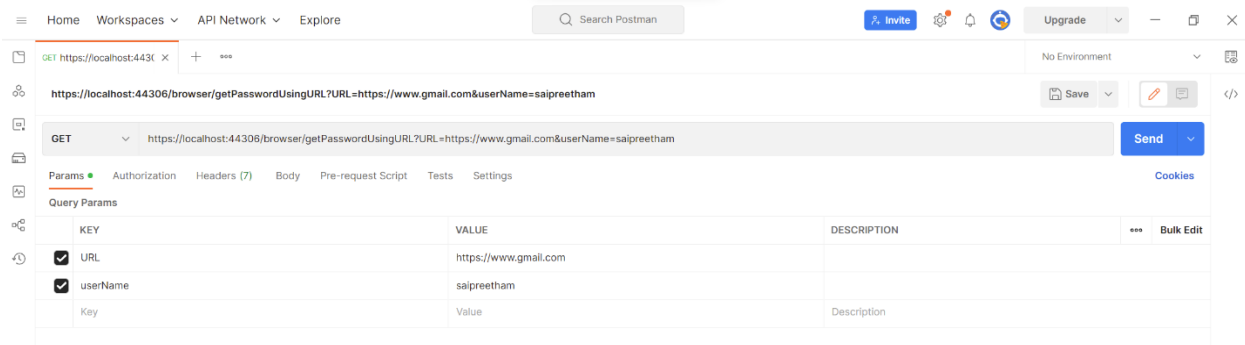
If the URL and username pair exists and the cookie is valid, the server responds with the password.

Search Postman

Invite Upgrade

GET https://localhost:4430(× +

No Environment ∨

https://localhost:44306/browser/getPasswordUsingURL?URL=https://www.gmail.com&userName=saipreetham

Save ∨

GET ∨ https://localhost:44306/browser/getPasswordUsingURL?URL=https://www.gmail.com&userName=saipreetham

Send ∨

Params ● Authorization Headers (7) Body Pre-request Script Tests Settings

Cookies

Query Params

KEY	VALUE	DESCRIPTION	
☑ URL	https://www.gmail.com		••• Bulk Edit
☑ userName	saipreetham		
Key	Value	Description	

Body Cookies (1) Headers (7) Test Results

Status: 200 OK Time: 140 ms Size: 235 B Save Response ∨

Pretty Raw Preview Visualize Text ∨

1 {yhx-y]}M9T#6