

1. INTRODUCTION TO PASSWORD MANAGER

In this world of technology where new technologies are introduced every single day, where the internet is an open source of information and where social media sites rule the world, accessing someone's personal data is not much of a difficult task. The internet has a wide scope and thus, accessing someone else's data and getting money out of it has become a new business.

Our negligence towards even the simplest things such as choosing and setting passwords could cost us a lump sum amount of loss. Setting passwords that contain personal information is one of the common mistakes that people across the globe do!! This makes passwords predictable and your data insecure. A person could easily identify your password and gain access to your data. A strong password is a ray of hope to keep you data secure.

The Password Manager application makes it easier to generate strong passwords that do not hold any of your personal information. The application allows you to specify your preferences at creating a password and automatically generates a password. Also, the user can regenerate the password if he/ she doesn't want to use a specific combination. The application also enables the storage of card details. The details are stored and can be viewed by the user at his/her convenience. We also need an efficient way to store the generated passwords, you can completely rely on us when it comes to safely storing passwords. The passwords are stored in an encrypted format which aren't visible to the database manager as well.

Our application is a one stop solution to all your password needs, including storage, generation and saving passwords and card details.

1.1 Project Objectives

- a. Generating a strong password that can be used by the user.
- b. Storing the user passwords safely.
- c. Retrieval of user passwords.
- d. Password Encryption.
- e. Checking for password breaches.
- f. Storage and retrieval of card details.

1.2 Project Goals

The main goal of this project is to make it easy and convenient for the user to generate and store passwords. The user can have all their passwords at their disposal (only if they've entered passwords into the application). One notable feature of this application is the ability to allow the user to copy the passwords and use them on websites. Instead of typing the passwords, the user can directly copy the passwords, but some specific conditions need to meet for this to happen. We'll be looking into this further in the project.

2. PROBLEM DEFINITION

2.1 Problem Statement

Development of a Password Manager application to make the generation, storage and retrieval of passwords and card information easier and more convenient for the user. To enable the storage of encrypted passwords at the backend and decrypt the passwords when the user wishes to view them.

2.2 Problem Finding

Creating strong passwords and remembering them is a task. Hence, users usually tend to set a password which contains their personal data, but this poses to be a threat to the security of their credentials. A small game of guess could turn tables and the users financial as well as organizational details could get in danger. Thus, we found the need for creating an application would make creation and storage of passwords easier for the user. And here we are with the solution!!

2.3 Purpose

The core purpose of our application is to generate passwords, collect card details and store them safely for future use. The user can access these passwords and details at their convenience. Also, they can copy passwords and use them, as well as delete passwords. The application is a one stop solution to all you password safety needs.

2.4 System Features

- Password Manager is a user-friendly application and any unsophisticated user is able to get one tap help.
- Provides complete security of user passwords.
- Makes generation of passwords easier and more convenient for the user.
- Encrypted password storage.
- Makes it possible to store card details.
- Makes card details available at one click.

3. SYSTEM REQUIREMENTS

Under this section, we will explore the system requirements: functional requirements, user requirements and performance requirements.

Requirement Analysis

Requirement Analysis is the first phase of software development process. This phase focuses to understand the problem. Requirement Analysis is on identifying what is need from these systems, not how the system will achieve its goals. In this phase often at least two parties are involved in Software Development-a client and a developer. The developer has to develop the system to satisfy the clients' needs. The developer and client arrange a meeting and discuss his/her own views. The developer asks the clients for his/her needs. After a meeting the developer understands what the requirements of the client are. Before starting of the development process, the developer analyze, test the requirements which are given by the clients. According to those requirements the developer starts development process. Hence the developer needs a user's problem.

In the software requirement we are dealing with the requirements of the proposed system, that's the capabilities of that system, which is yet to be developed, should have. The software requirement specification (SRS) is a document that completely describes what the proposed software should do without describing how the software will do it. So the basic goal of Requirement Phase is to produce the SRS, which describes the complete external behavior of the proposed software. The basic aim of problem analysis is to obtain the clear understanding of the needs of the clients and the user, what exactly described from the software, and what the constraints on the solution are? This involves a meeting of user and developers. The developer may ask the following questions to users:

- Who will use the developed software?
- What types of characteristics may have the software?

The above questions are to be answered by us. For the first question our answer will be that the system will be used by any person who wants to make it convenient to generate passwords, store passwords and store and retrieve card details. The user must just register to the application, and then he/she can generate and store passwords safely.

For the second question our answer is that the characteristics of our system are as follows:

- a. The login passwords of users are hashed and stored in the database, which makes the decryption of passwords impossible
- b. The user passwords for various logins are stored in an encrypted format in the database
- c. The card details of the user are stored in the database.
- d. Correct retrieval of passwords for the specified user.

3.1 Functional Requirements

A Functional Requirement (FR) is a description of the service that the software must offer. It describes a software system or its component. A function is nothing but inputs to the software system, its behavior, and outputs. Functional software requirements help you to capture the intended behavior of the system. There are several methods to write functional requirements, but the most common method is by constructing user stories and using user story formats: as a ____, I want to be able to ____ so that ____.

User story 1: as a user, I must be able to login into my created account so that I can successfully access all my passwords and card details.

FR No.	Functional Requirements
1	Register and Login page
2	Hashes of passwords stored in the database in accordance to the user id
3	Checking of the entered credentials along with the stored credentials
4	Error message “Invalid Credentials” in case the credentials are wrong
5	Successful login if the credentials are correct

User story 2: as a user, I must be able to enter passwords into the application so that I can use them for my various logins.

FR No.	Functional Requirements
1	Add Passwords Page
2	Ability to enter into the various input fields present
3	Storage of the entered details in the database
4	Storage of passwords in an encrypted form in the database

User story 3: as a user, I must be able to access passwords from my created account so that I can successfully use all my passwords.

FR No.	Functional Requirements
1	My passwords page
2	Retrieving correct data from the database as per the user id of the user
3	Viewing passwords on the page
4	Copying passwords and using them

User story 4: as a user, I must be able to generate passwords from using my created account so that I can generate and use passwords for various logins

FR No.	Functional Requirements
1	Generate passwords page
2	Ability to enter data into various input fields provided on the page
3	Generation of passwords as per user specifications
4	Clickable generate button
5	Copying passwords
6	Regeneration facility

User story 5: as a user, I must be able to delete passwords from my created account so that I can delete passwords as per my requirements.

FR No.	Functional Requirements
1	My passwords page
2	Delete button should be functional
3	The password should be deleted from the database
4	An alert message should be displayed saying that the passwords once deleted cannot be retrieved

User story 6: as a user, I must be able to store card details from my created account so that I can view and delete my card details as per my convenience.

FR No.	Functional Requirements
1	‘Enter Card Details’ page
2	‘Save’ button should be functional
3	View Cards page
4	‘Delete card’ button should be functional

3.2 User Requirements

User requirements, often referred to as user needs, describe what the user does with the system, such as what activities that users must be able to perform. User requirements are generally documented in a User Requirements Document (URD) using narrative text. User requirements are generally signed off by the user and used as the primary input for creating system requirements.

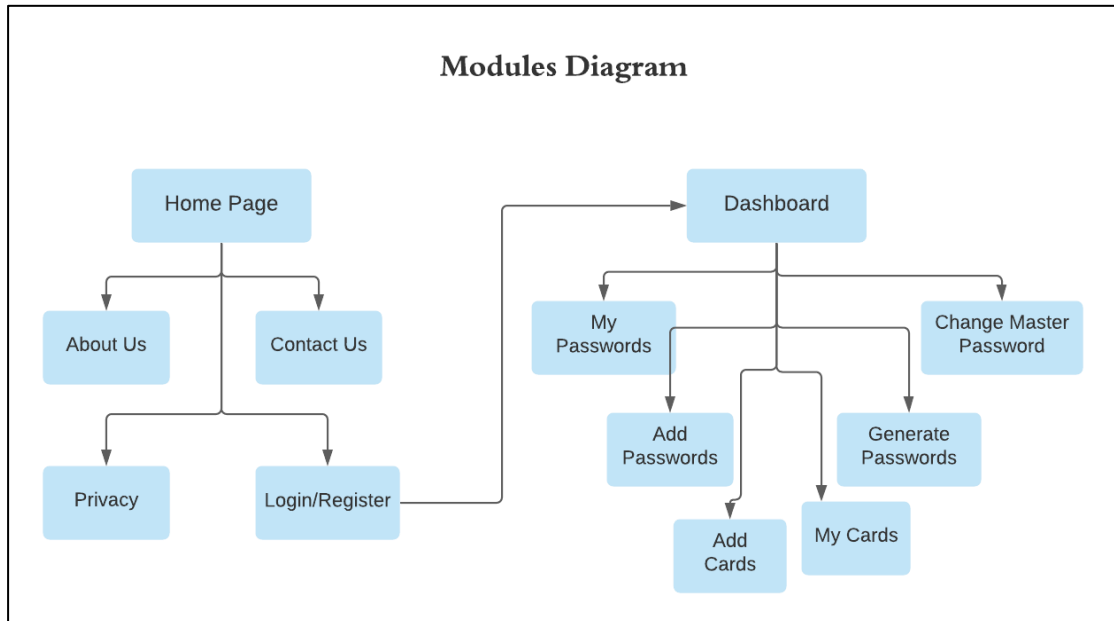
An important and difficult step of designing a software product is determining what the user actually wants it to do. This is because the user is often not able to communicate the entirety of their needs and wants, and the information they provide may also be incomplete, inaccurate and self-conflicting. The responsibility of completely understanding what the customer wants falls on the business analyst. This is why user requirements are generally considered separately from system requirements. The business analyst carefully analyzes user requirements and carefully constructs and documents a set of high quality system requirements ensuring that the requirements meet certain quality characteristics.

Currently, we have spotted a few user requirements for our application and they are as follows:

- User-friendly web page
- Generation of passwords as per the requirements specified
- Regeneration of passwords
- Saving passwords securely
- Accessing, viewing, copying and deleting passwords
- Adding new passwords
- Adding Card Details
- Accessing, viewing, copying and deleting card details
- Login only with correct credentials

4. SYSTEM DESIGN

4.1 Modules



Home Page Module:

The home page is the main page of the application that contains various sections that provide information regarding the application and the developers. It has sections like, About Us, Contact Us, Privacy, and Login/Register. It is the page that allows the user to login into his/her account or register with the application and create a new account.

- Login/Register

The login module allows the user to login into his already created account. The user must enter the registered email id and the correct master-password in order to get access to the dashboard. The register module allows the user to create a new account for storing and generating passwords. The user enters his/her email id, master-password and confirms the password.

Dashboard Module:

The dashboard module is available to the users only when they log in successfully. It provides the users with the interface to add passwords, view their own passwords or generate new passwords. Also the user is provided with the interface to change their master-password.

- My Passwords

All the user passwords stored with the application are available under this particular section. As the user adds passwords using the add password functionality, the passwords are added to the database and then displayed into

this section of the page. The user can also delete the passwords that are already saved.

- Add Passwords

The user can add the website title, the username for that site and the password for the site using the add passwords section. The data entered by the user is entered into the database and stored there in an encrypted form.

- Add Cards

The user can add his/her card details. The card name, card holder name, card number, expiry date, CVV, and brand is stored. The data entered by the user is stored under the cards table in the database.

- My Cards

All the card details previously entered by the user are displayed under this section. The user can also delete the previously stored card details.

- Generate Passwords

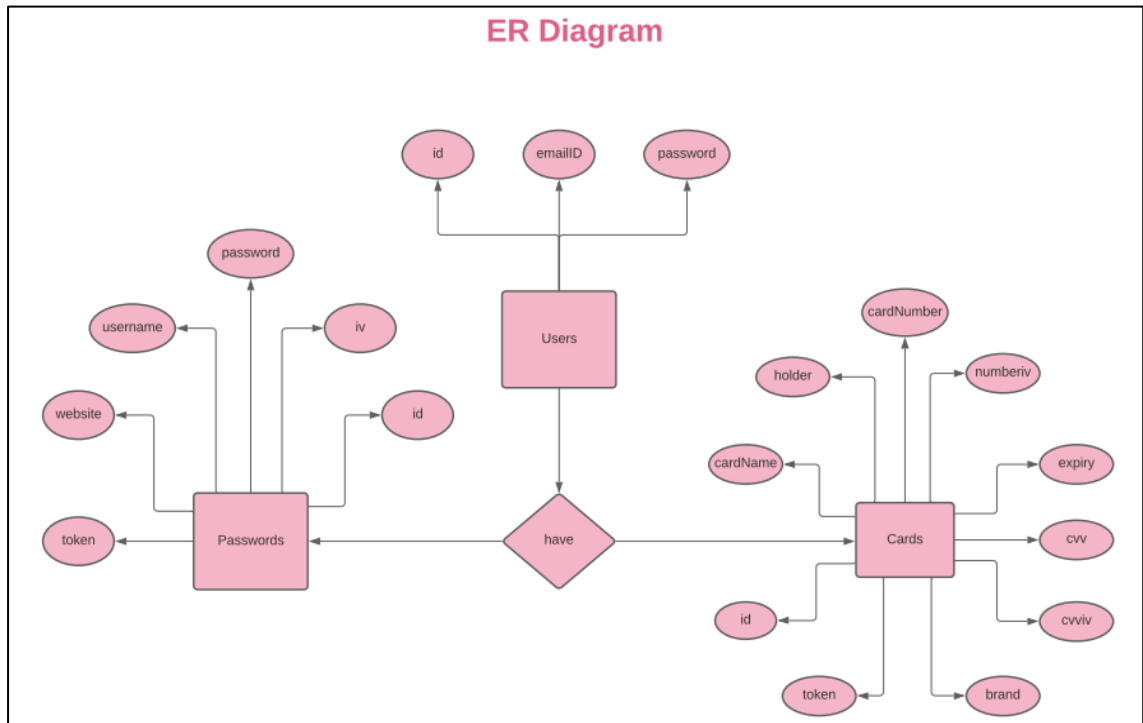
Under this section, the passwords are generated as per the length entered by the user. The passwords are generated and displayed to the user. If the user is satisfied with the password, he/ she can use the generated password for any of his logins, or else the user can generate a new password.

- Change Master Password

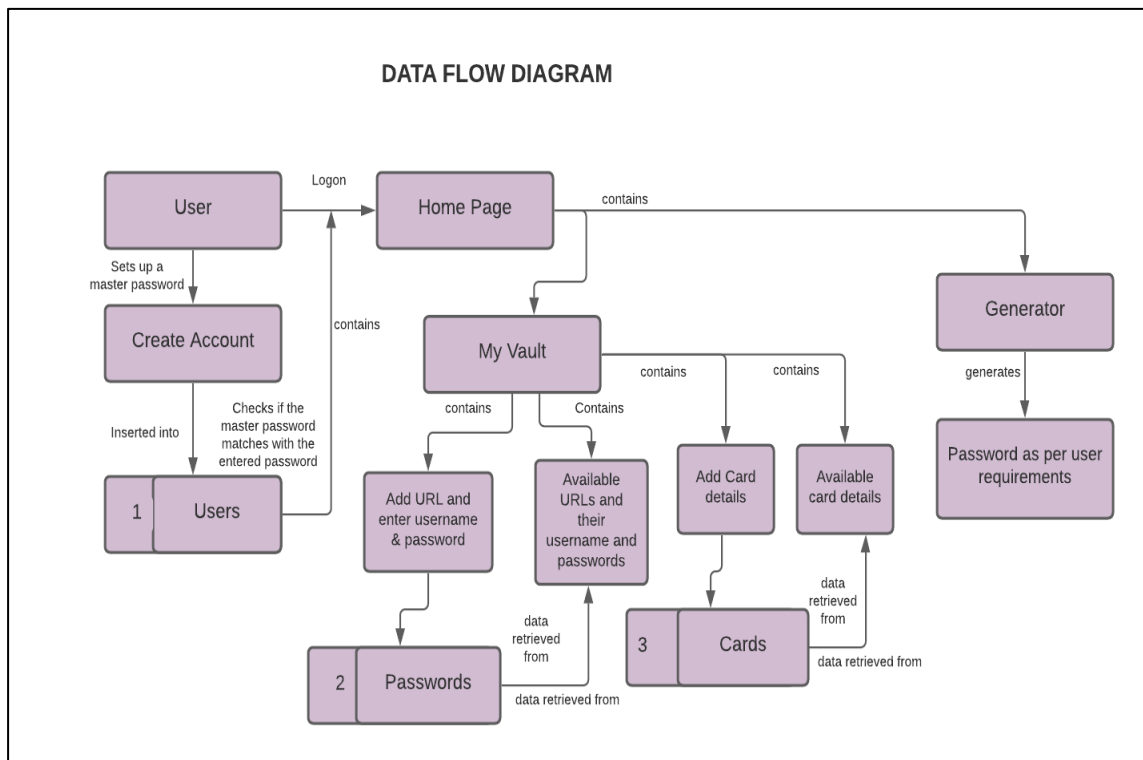
Once the user is logged in, if the user wants to change the master password, this section provides the UI for that functionality. For this the user must be first logged in successfully.

4.2 System Analysis Models

➤ Entity Relationship Diagram

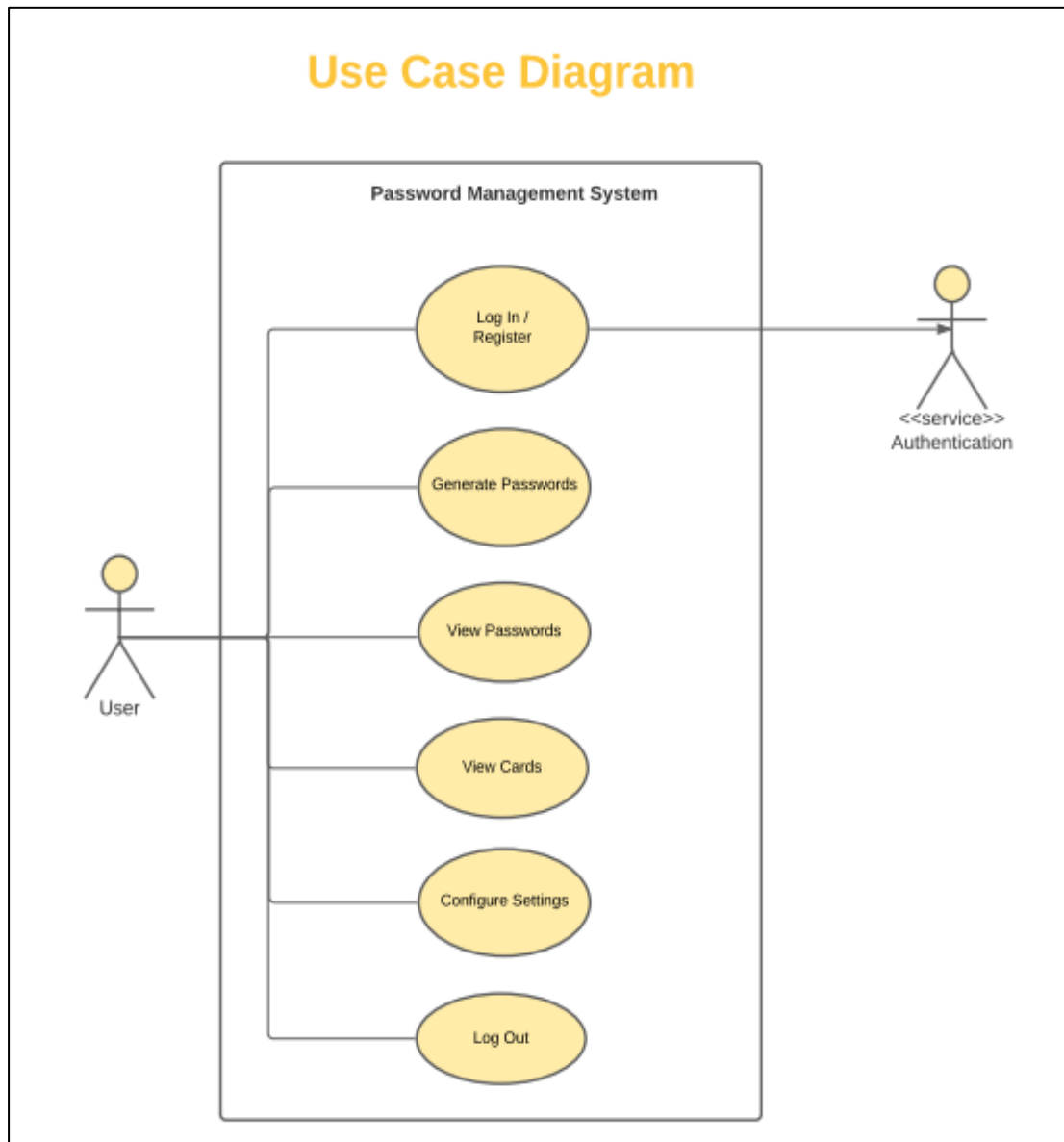


➤ Data Flow Diagram

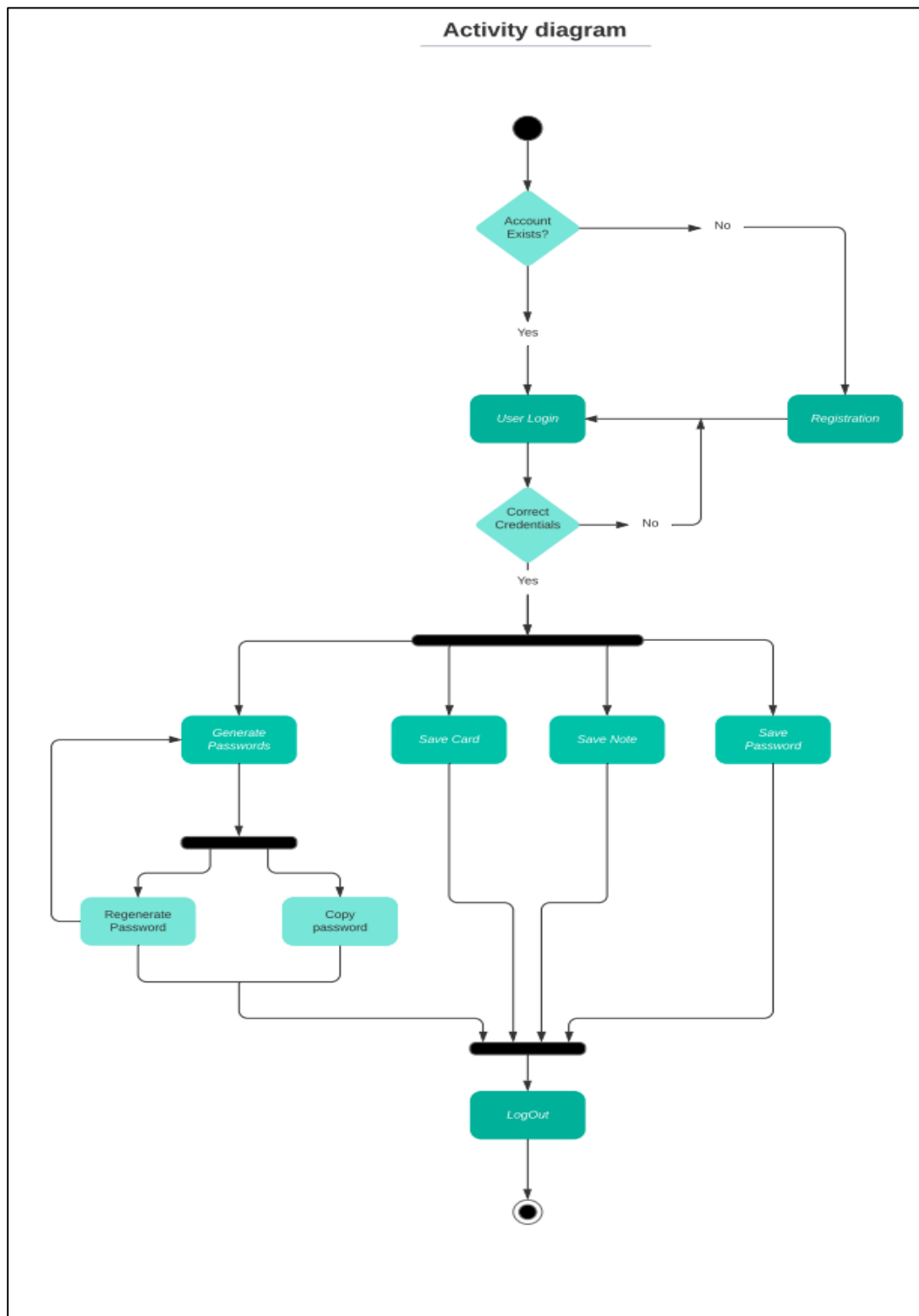


5. UML DIAGRAMS

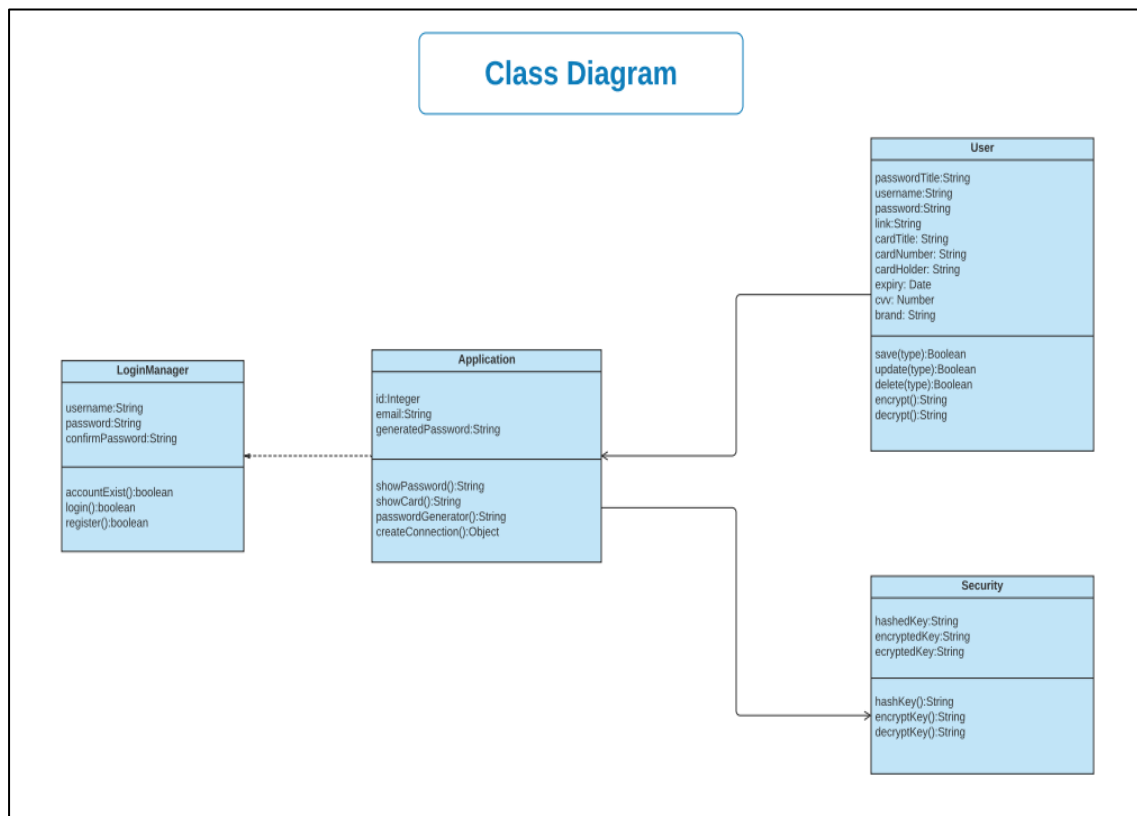
5.1 Use Case Diagram



5.2 Activity Diagram



5.3 Class Diagram

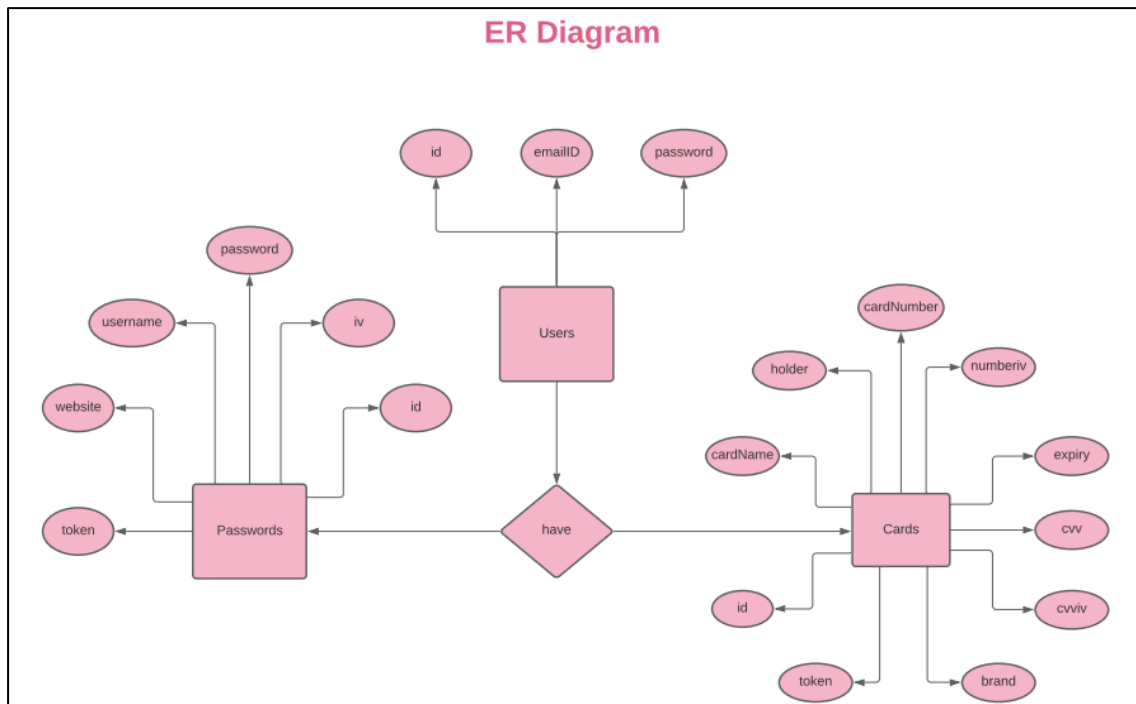


6. DATABASE

For our application, we have used MySQL to provide a database connectivity. **MySQL** is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language. A relational database organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

Our application uses 2 database tables: Users and Password.

ER Diagram:



Following is the description for the tables which includes the column names, their data types and if it accepts null values or not.

1. Users

Sr. No.	Field Name	Data Type	Null
1	id	Integer	No

2	emailID	Varchar	No
3	password	Varchar	No

The master password of the user is stored in an encrypted format. Here, we use the hashing technique for encryption. The password once hashed cannot be reconverted into its original form. The next time the user logs in, the password is again hashed and then the hash stored in the database is compared to the newly generated hash. If they match, the access is granted, else the application denies access to the user.

2. Passwords

Sr. No.	Field Name	Data Type	Null
1	id	Integer	No
2	username	Varchar	No
3	password	Varchar	No
4	link	Varchar	No
5	title	Varchar	No
6	token	Varchar	No

The user passwords are stored in the database in an encrypted format. For this, we have used The AES-128 block encryption technique. When the user wants to view the password, the password is decrypted and made available to the user.

3. Cards

Sr. No.	Field Name	Data Type	Null
1	cardName	Varchar	No
2	Holder	Varchar	No
3	cardNumber	Varchar	No

4	Numberiv	Varchar	No
5	expiry	Date	No
6	cvv	Int	No
7	token	Int	No
8	brand	Varchar	No
9	token	Int	No

The card number and CVV are stored in the database in an encrypted format. For this, we have used The AES-128 block encryption technique. When the user wants to view the card number and CVV, the details are decrypted and made available to the user.

7. TESTING

Testing is the process of executing a program with the aim of finding errors. To make our software perform well it should be error-free. If testing is done successfully it will remove all the errors from the software.

Principles of Testing:

- All the test should meet the customer requirements
- To make our software testing should be performed by a third party
- Exhaustive testing is not possible. As we need the optimal amount of testing based on the risk assessment of the application.
- All the test to be conducted should be planned before implementing it
- It follows the Pareto rule(80/20 rule) which states that 80% of errors come from 20% of program components.
- Start testing with small parts and extend it to large parts.

7.1 Test Deliverables

Test deliverables are the test artifacts that are given to the stakeholders of a software project during the SDLC. Some of the deliverables are provided before the testing phase commences and some are provided during the testing phase and some after the testing phase. The test deliverables are as follows:

- Test Strategy
- Test Plan
- Test Cases
- Test Data
- Requirement Traceability Matrix
- Test Execution Report
- Defect Report
- Test Status Report

7.2 Test Cases for Password Manager

Login Module:

Test Case No.	Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
1	Check results when the login button is clicked without entering email id and password	No data	Prompt : Enter credentials to login. Login Failed	Prompt : Enter credentials to login. Login Failed	Pass
2	Check results when the login button is clicked without entering email id	Username field is left blank, password: aditi@123	Prompt : Email id should be filled in	Prompt : Email id should be filled in	Pass
3	Check results when the login button is clicked without entering password	Username : aditi@gmail.com ,password field is left blank	Prompt : Password should be filled in	Prompt : Password should be filled in	Pass
4	Check results when the username and password doesn't meet required validations	Username : aditi561.gmail.com password : aditinikam	Prompt : Username and password doesn't meet the defined validations	Prompt : Username and password doesn't meet the defined validations	Pass
5	Check results when incorrect password is entered	Username : aditi@gmail.com , password : aditi123	Prompt : Incorrect password	Prompt : Incorrect password	Pass
6	Check results when incorrect username and password is entered	Username : aditi123@gmail.com , password : aditi123	Prompt : Incorrect username and password	Prompt : Incorrect username and password	Pass
7	Check results when correct username and password is entered	Username : aditi@gmail.com , password : aditi@123	Prompt : Login Successful	Prompt : Login Successful	Pass

Registration Module:

Test Case No.	Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
1	Check results when the register button is clicked without entering any data	No data	Prompt : Fill in the details to register.	Prompt : Fill in the details to register.	Pass
2	Check results when the register button is clicked without entering email id	Email field is left blank, password: aditi@123 confirm password: aditi@123	Prompt : Email id should be filled in	Prompt : Email id should be filled in	Pass
3	Check results when the register button is clicked without entering password	Email : aditi@gmail.com, password field and confirm password field is left blank	Prompt : Password should be filled in	Prompt : Password should be filled in	Pass
4	Check results when the email and password doesn't meet required validations	Email : aditi561.gmail.com password : aditinikam	Prompt : Email and password doesn't meet the defined validations	Prompt : Email and password doesn't meet the defined validations	Pass
5	Check results when password and confirm password field do not have same value.	Email: aditi@gmail.com , password : aditinikam Confirm password: aditiiiin	Prompt : Password and confirm password do not match each other	Prompt : Password and confirm password do not match each other	Pass
6	Check results when email id, password and confirm password fields are filled.	Email : aditi@gmail.com , password : aditi@123 Confirm password: aditi@123	Prompt : Registration Successful	Prompt : Registration Successful	Pass

Save Password Module:

Test Case No.	Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
1	No data is entered and the save password button is clicked	No data	Prompt : Fill in the details to save password.	Prompt : Fill in the details to save password	Pass
2	Check results after entering the username, password, website name.	Website: dribble.com Username: Aditiii12 Password: Aditii123	Prompt : Password successfully saved	Prompt : Password successfully saved	Pass
3	Check when any of the fields is left empty and the save button is clicked.	Website: dribble.com Username: Aditiii12	Prompt : All the fields should be filled in	Prompt : All the fields should be filled in	Pass

Show Passwords Module:

Test Case No.	Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
1	Checking if all the saved passwords are visible.	No data	All saved passwords are visible	All saved passwords are visible	Pass
2	Check the result when the show button is clicked.	Click the show button	The password should be visible for a fraction of a second	The password is visible for a fraction of a second	Pass
3	Check the result when the delete button is clicked	Click the delete button	The password must be deleted.	The password must be deleted.	Pass
4	Check what happens when the show button is clicked and the mouse is dragged without releasing it	Click on the show button and drag the mouse aside and release it	The password should be visible for a longer time	The password should be visible for a longer time	Pass

Generate Password Module:

Test Case No.	Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
1	Checking if the password generated is of the same length as entered by the user	Length = 10	The generated password must be of 10 characters	The generated password is of 10 characters	Pass
2	Check what happens when the generate button is clicked once again after the password is generated	Click the generate button	The password is regenerated	The password is regenerated	Pass

Save Cards Module:

Test Case No.	Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
1	No data is entered and the save card button is clicked	No data	Prompt : Fill in the details to save the card details	Prompt : Fill in the details to save the card details	Pass
2	Check results after entering the all the details	All details entered	Prompt : Card details successfully saved	Prompt : Card details successfully saved	Pass
3	Check when any of the fields is left empty and the save button is clicked.	All fields except the name are filled in	Prompt : All the fields should be filled in	Prompt : All the fields should be filled in	Pass

8. DEFECT REPORT

DEFECT INFORMATION	
Defect ID	106
Software/Project Name	Password Manager - CryptKey
Module Name	Password Generation Module
Defect Name	Generator Class - Incorrect password generation.
Date	10-06-2021
Tester	Ms. Aditi Nikam
Assigned to	Ms. Salonee Pathan
Defect Severity	Medium - It causes some undesirable behavior, but the system is still functional.
Defect Priority	High - It must be resolved as soon as possible.
Reproducible	No
Title	The generator generates a password regardless of the combination specified by the user.
Description	If the user does not want to include special symbols, still the password combination contains some special characters. Same happens with the other fields i.e uppercase letters, numbers, lowercase letters.
Steps to replicate	<ol style="list-style-type: none"> 1. Open the generator section of the password manager 2. Select lowercase & numbers for password generation and select length as 8. 3. Click on generate.
Actual Result	34Opy\$%A
Expected Result	ijr5ng87
Defect Probability	High - will always occur.
Status	Resolved
RESOLUTION	
Date resolved	13-06-2021
Resolved by	Ms. Salonee Pathan
Version	1.0
Resolution Comment	Resolved successfully - Ready for retesting.
RETEST	
Retested by	Ms. Aditi Nikam
Version tested	1.0
Date tested	14-06-2021

Retest comment	Defect resolved - Generator works as expected
SIGNATURES	
Originator	Ms. Salonee Pathan
Tester	Ms. Aditi Nikam
Programmer	Ms. Salonee Pathan
Project Manager	Mr. Himanshu Sangale
Marketing	-
Product Support	Ms. Salonee Pathan

DEFECT INFORMATION	
Defect ID	117
Software/Project Name	Password Manager - CryptKey
Module Name	Password Encryption Module
Defect Name	Encryption Class - Password is not encrypted properly.
Date	03-06-2021
Tester	Mr. Himanshu Sangale
Assigned to	Ms. Salonee Pathan
Defect Severity	Major - It is a highly severe defect and collapses the system. However, certain parts of the system remain functional.
Defect Priority	High - The defect must be resolved as soon as possible as it affects the system severely and cannot be used until it is fixed
Reproducible	No
Title	Passwords saved by the users do not get encrypted correctly.
Description	If the user saves a large combination of characters as password some characters do not undergo encryption and are visible directly upon opening.
Steps to replicate	1. Open the passwords section of the password manager. 2. Select any password amongst the saved ones.
Actual Result	pass**rd_*x*mpl*12*
Expected Result	*****
Defect Probability	High - will always occur.
Status	Resolved
RESOLUTION	
Date resolved	07-06-2021

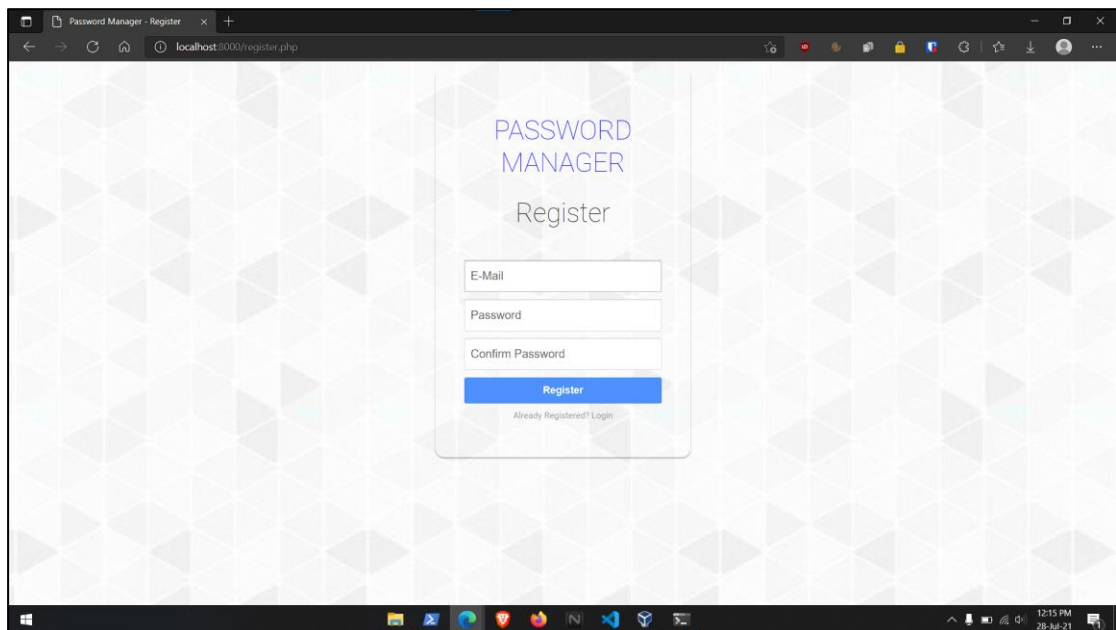
Resolved by	Ms. Salonee Pathan
Version	1.0
Resolution Comment	Resolved successfully - Ready for retesting.
RETEST	
Retested by	Mr. Himanshu Sangale
Version tested	1.0
Date tested	10-06-2021
Retest comment	Defect resolved - Passwords get encrypted as it should.
SIGNATURES	
Originator	Ms. Salonee Pathan
Tester	Mr. Himanshu Sangale
Programmer	Ms. Salonee Pathan
Project Manager	Ms. Aditi Nikam
Marketing	-
Product Support	Ms. Salonee Pathan

DEFECT INFORMATION	
Defect ID	121
Software/Project Name	Password Manager - CryptKey
Module Name	Cards Module
Defect Name	Cards Class - Allows entry of expired cards.
Date	12-06-2021
Tester	Ms. Salonee Pathan
Assigned to	Ms. Aditi Nikam
Defect Severity	Medium - It causes some undesirable behavior, but the system is still functional.
Defect Priority	High - It must be resolved as soon as possible.
Reproducible	No
Title	Expiry date field for cards gives option to select past years, allowing user to save expired cards.
Description	While entering expiry date for a particular credit or debit card, the dropdown list is displaying options for selection of past years as expiry date.
Steps to replicate	<ol style="list-style-type: none"> 1. Open the Cards section of the password manager. 2. Select add card option.

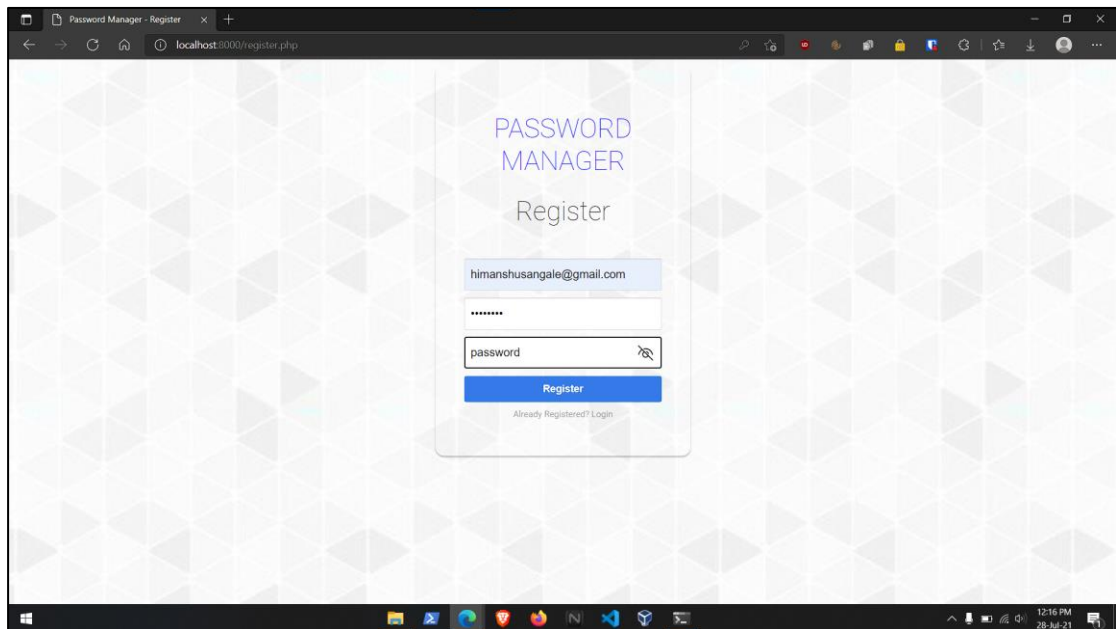
	3. Fill the details and choose expiry date option to open dropdown menu.
Actual Result	Allows selection of the years >2021
Expected Result	Should only allow selection of years <=2021
Defect Probability	High - will always occur.
Status	Resolved
RESOLUTION	
Date resolved	13-06-2021
Resolved by	Mr. Himanshu Sangale
Version	1.0
Resolution Comment	Resolved successfully - Ready for retesting.
RETEST	
Retested by	Ms. Salonee Pathan
Version tested	1.0
Date tested	14-06-2021
Retest comment	Defect resolved - Entry of expired cards prohibited.
SIGNATURES	
Originator	Ms. Aditi Nikam
Tester	Ms. Salonee Pathan
Programmer	Mr. Himanshu Sangale
Project Manager	Ms. Salonee Pathan
Marketing	-
Product Support	Ms. Salonee Pathan

9. OUTPUTS

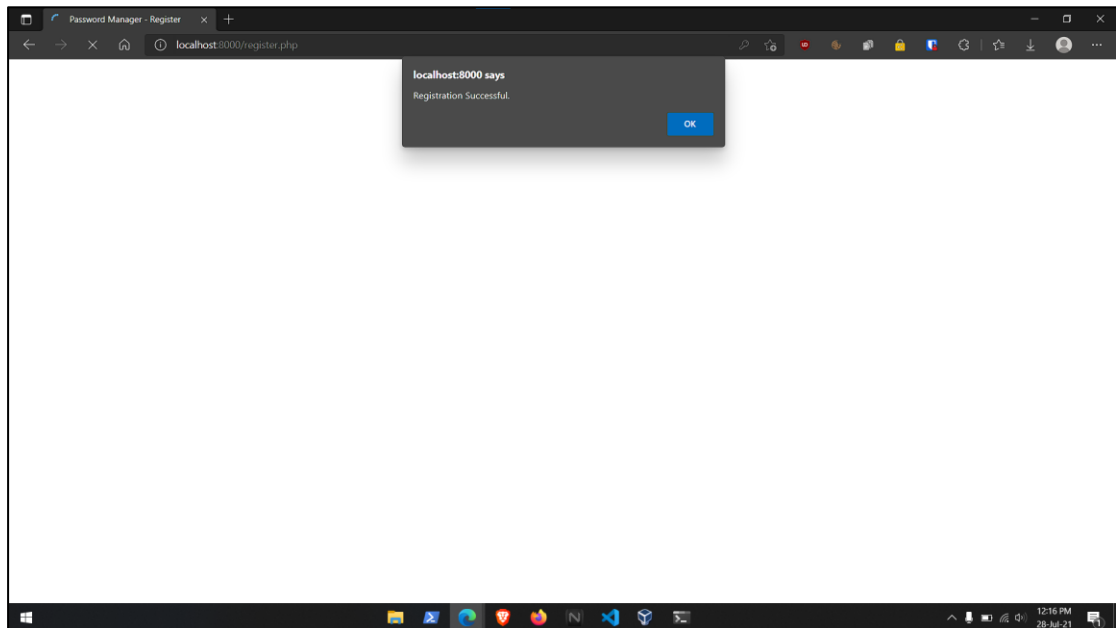
Registration Module



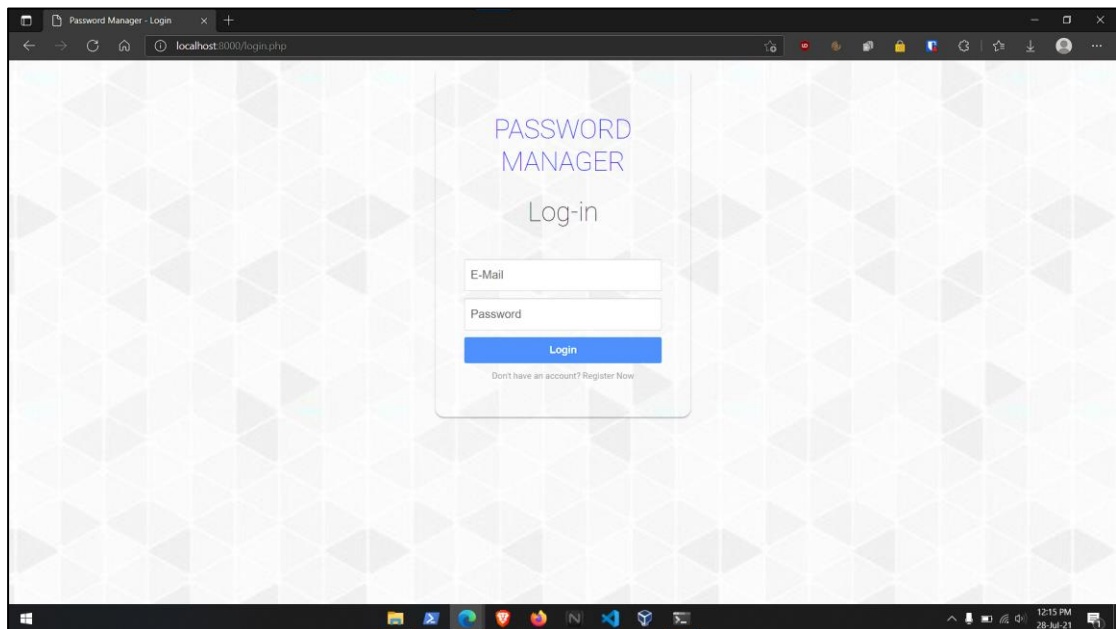
All details are entered as per the requirements



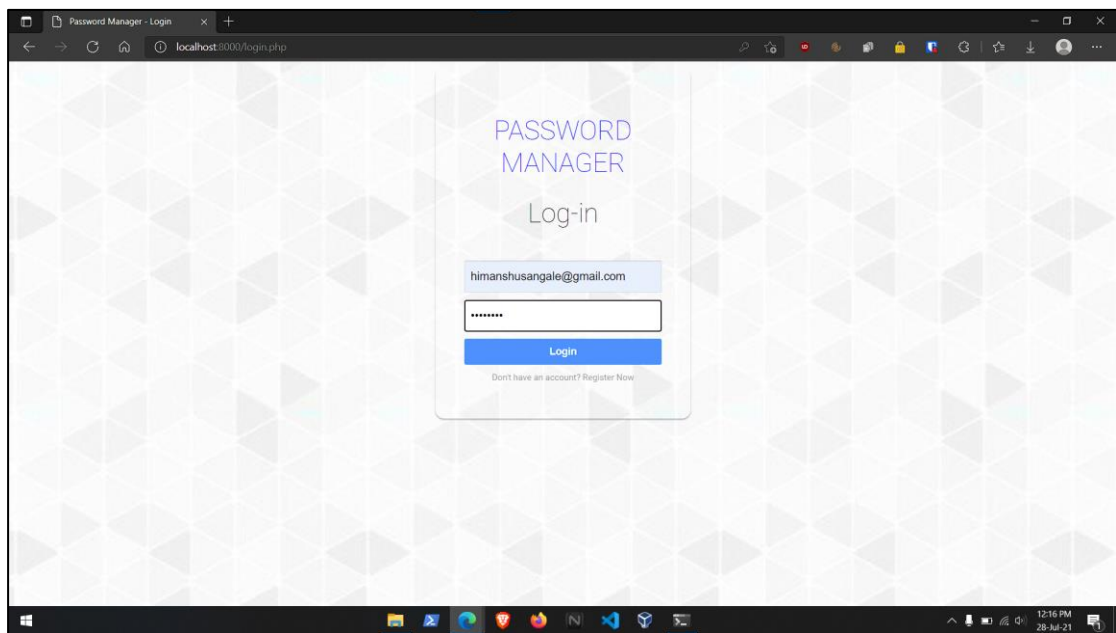
When the details are correctly entered, the page generates a prompt saying ‘Registration Successful’



Login Module

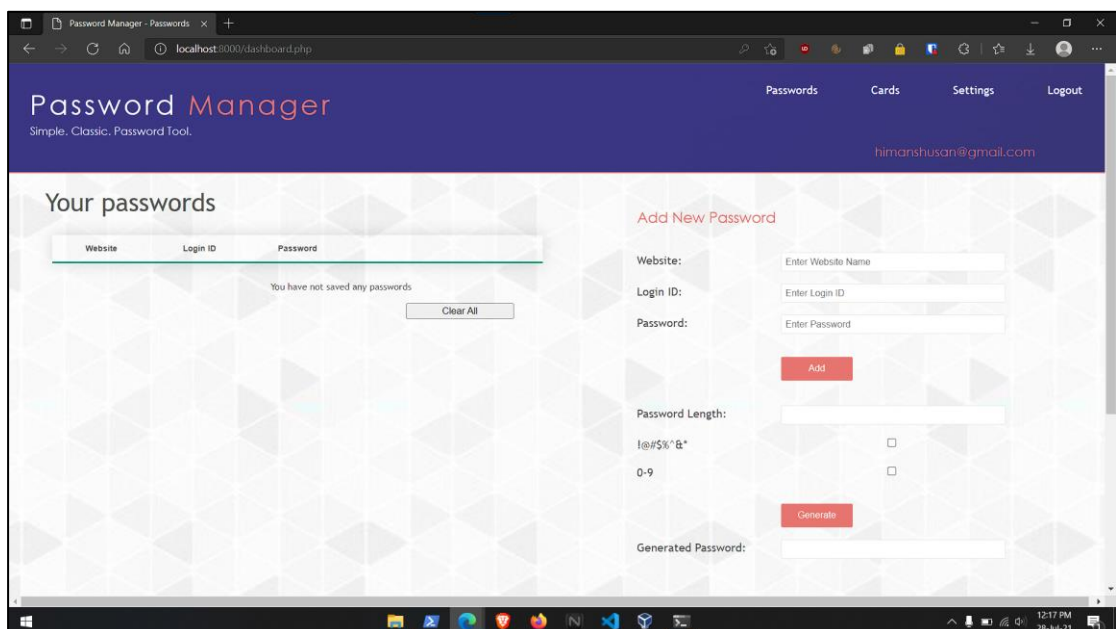


Once the registration is successfully done, you can login with your credentials

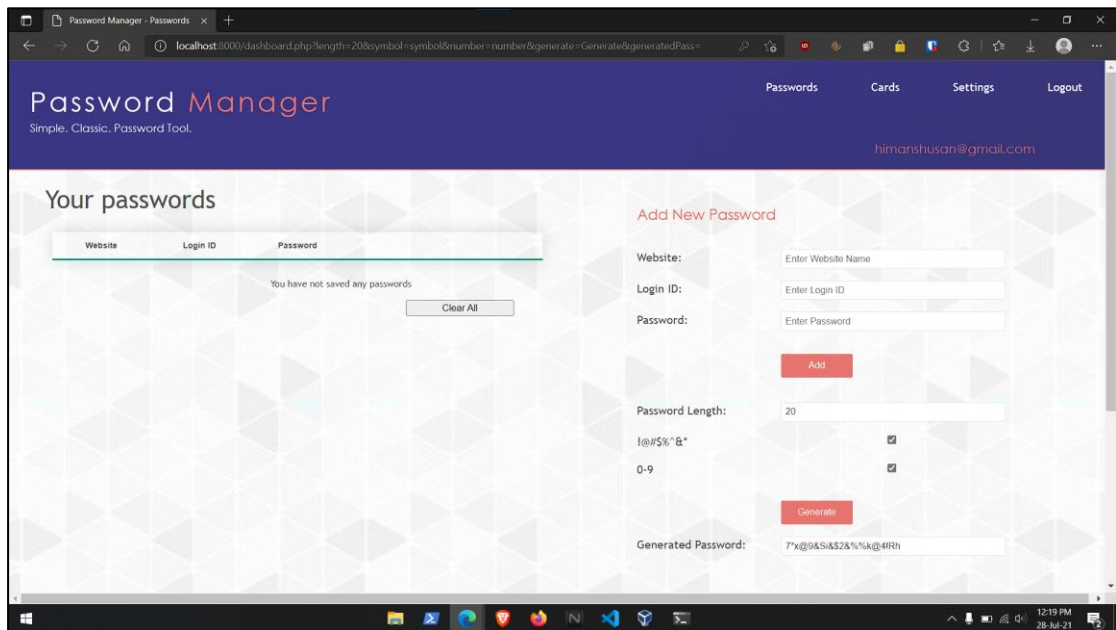


My Vault Module

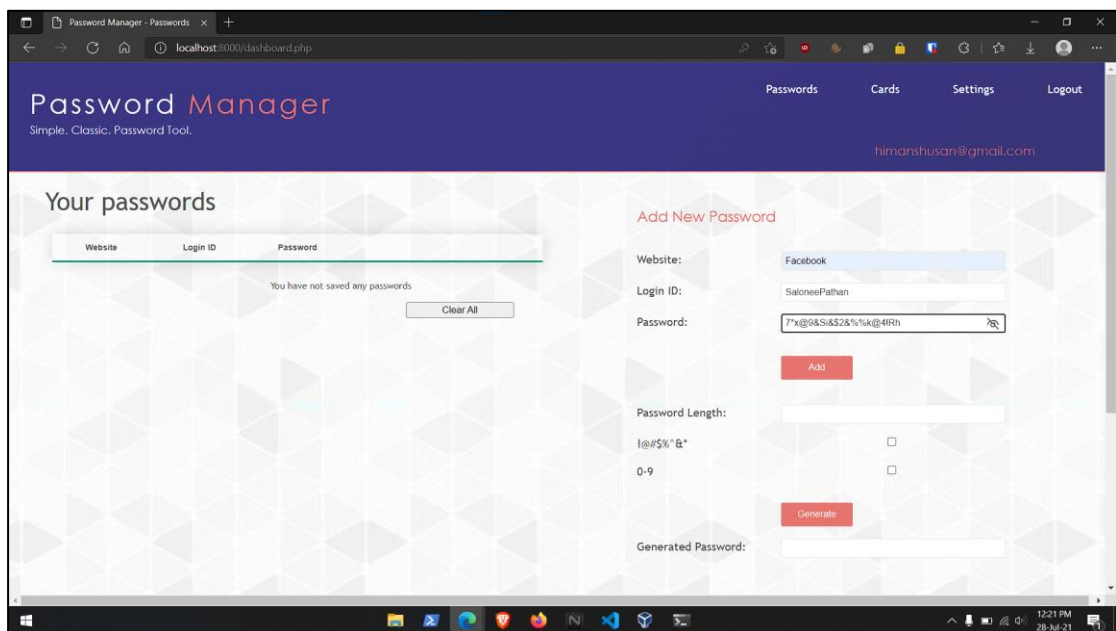
This is the page that appears when the user logs in to the system. The page shows all the previously stored passwords, a section to enter new passwords and allows the generation of passwords.



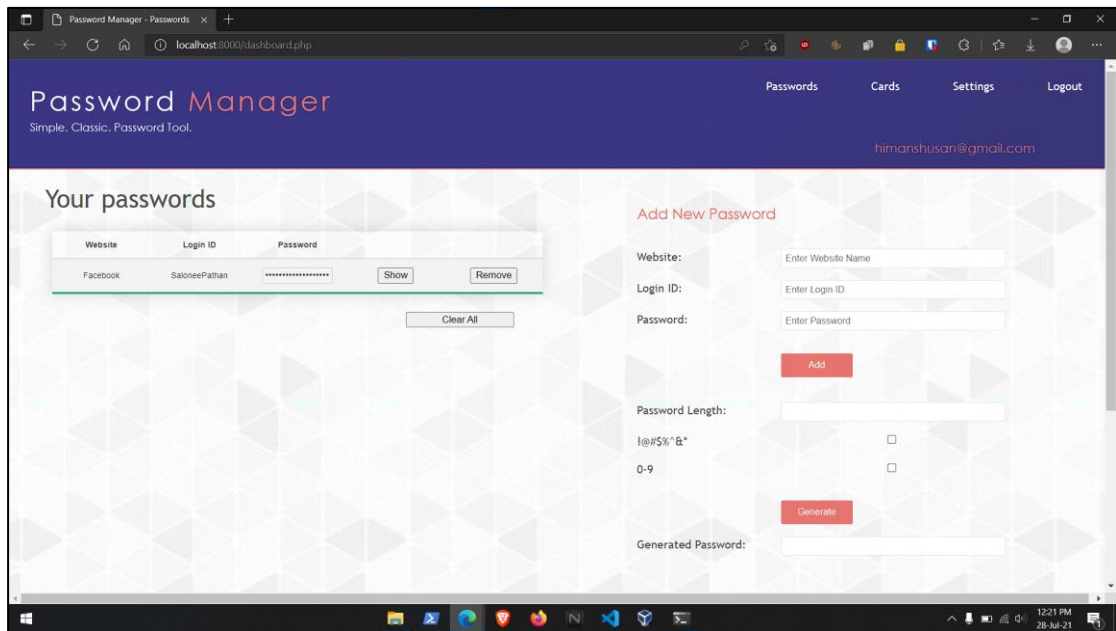
The user enters the length of the password that is desired and the system generates the password as per the user requirements. To regenerate the password the user can refresh the page.



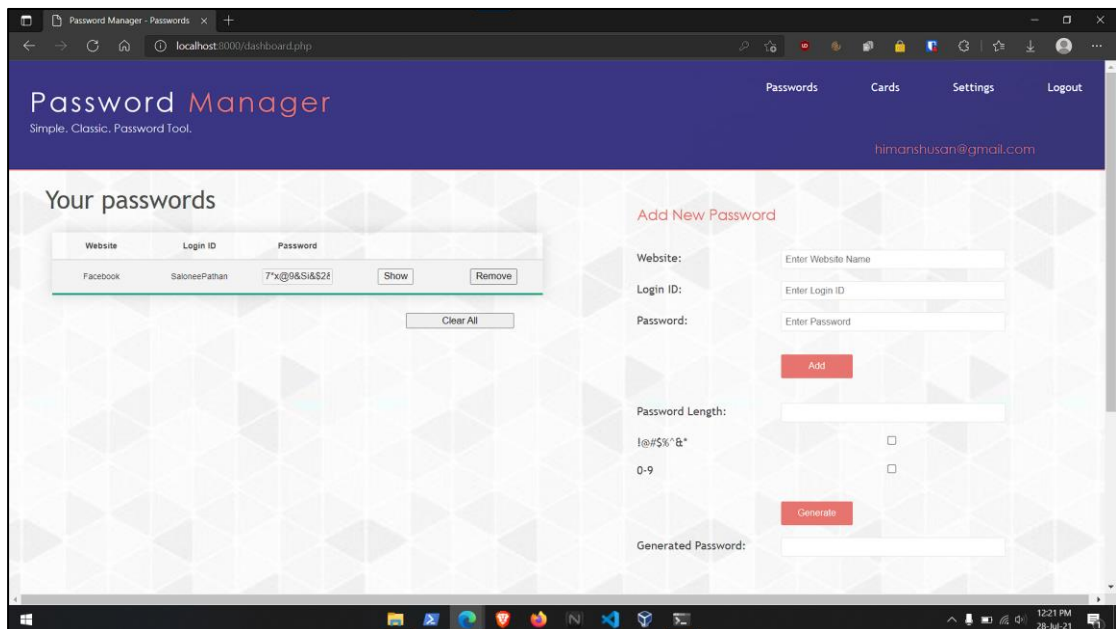
The user can enter the details in the 'Add New Password' section and then click on the add button to save the password.



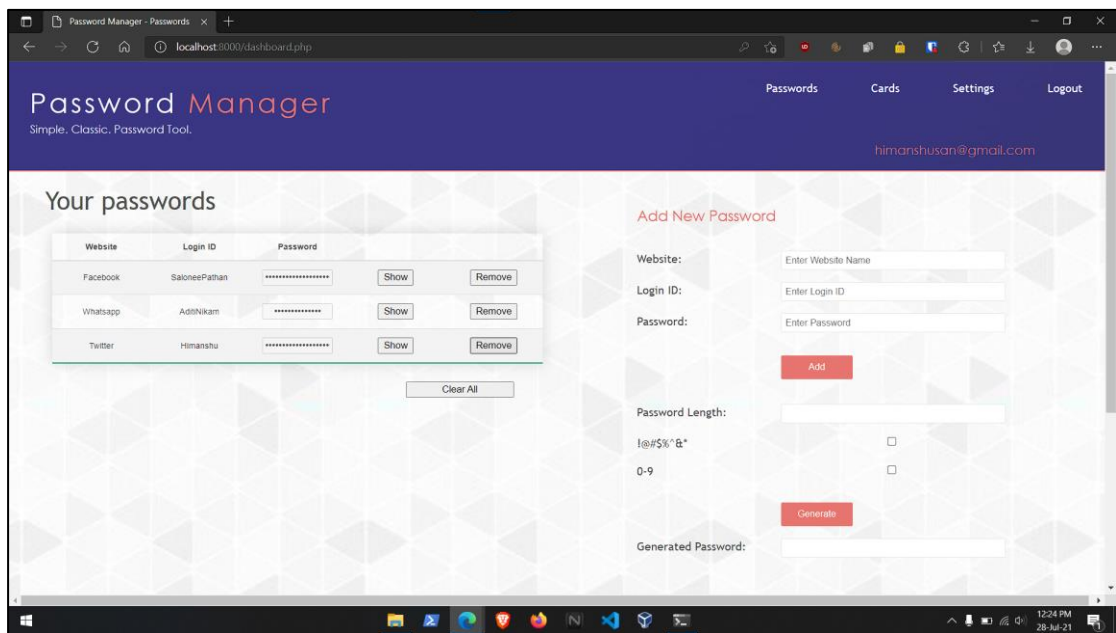
The user can see the stored passwords in the ‘Your Passwords’ section.



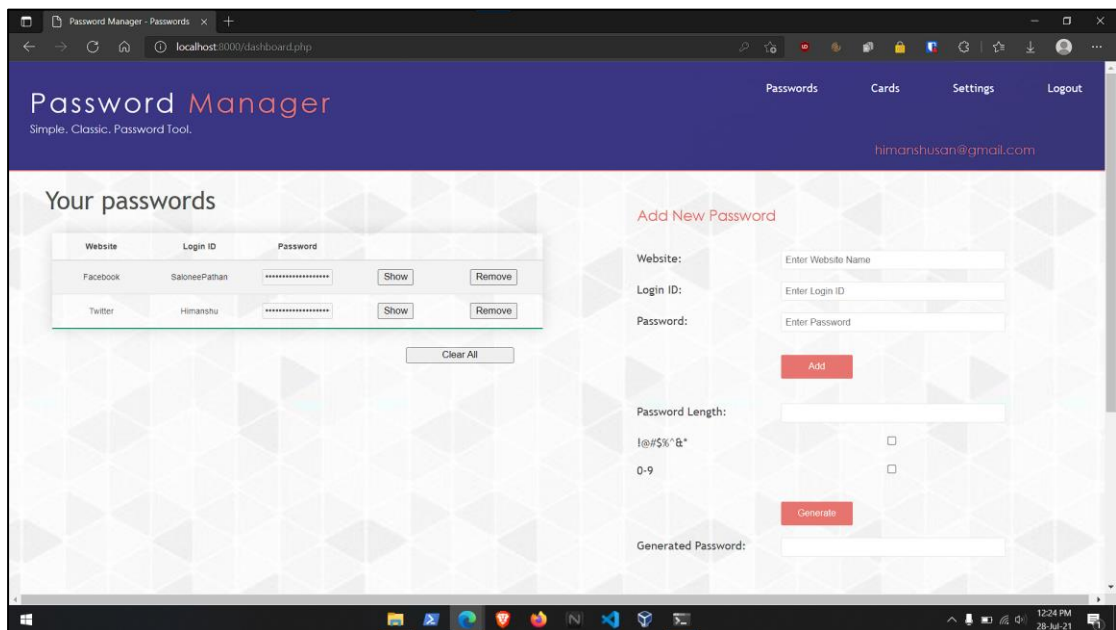
The user can view his/her passwords by clicking on the show button. Doing this, the password is visible only for a fraction of a second. To keep the password visible for a longer while, the user can press the show button and move the cursor away from the button and then release it.



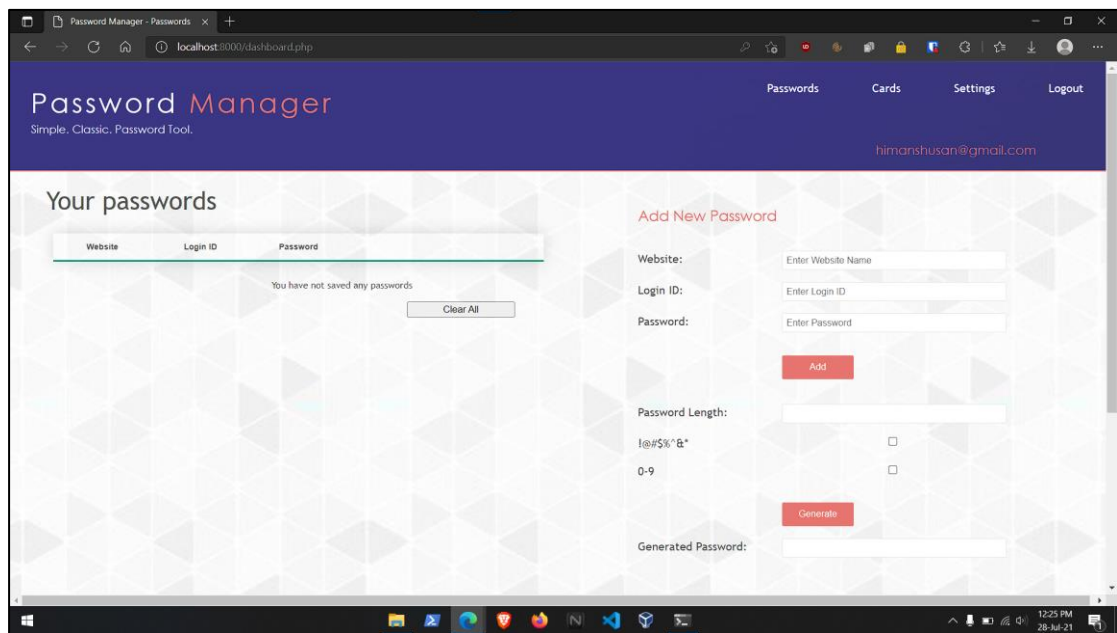
Multiple passwords can be saved and viewed in the same way.



By clicking on the 'remove' button, the user can delete the stored passwords.

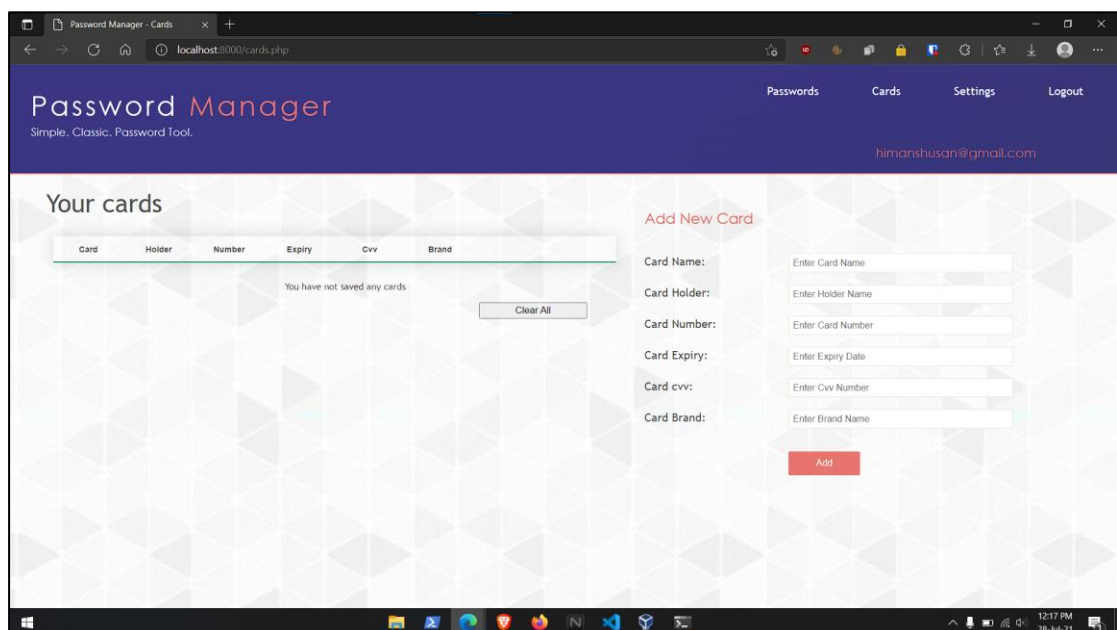


By clicking on the ‘Clear All’ button, the user can delete all the saved passwords.



Cards Module

The cards module is responsible for storing and displaying card details of the user.



Enter the card details to be saved in the ‘Add New Card’ section, and click on the add button.

The screenshot shows the 'Password Manager - Cards' interface. The header includes the title 'Password Manager' with the tagline 'Simple. Classic. Password Tool.' and navigation links for 'Passwords', 'Cards', 'Settings', and 'Logout'. The user's email 'himanshusan@gmail.com' is displayed in the top right. The main content area is titled 'Your cards' and features a table with columns: Card, Holder, Number, Expiry, Cvv, and Brand. Below the table, a message states 'You have not saved any cards' with a 'Clear All' button. To the right, the 'Add New Card' form is visible, containing input fields for Card Name (HDFC Bank), Card Holder (Salonee Pathan), Card Number (masked with asterisks), Card Expiry (09Aug2025), Card cvv (masked with asterisks), and Card Brand (MasterCard). An 'Add' button is located at the bottom of the form.

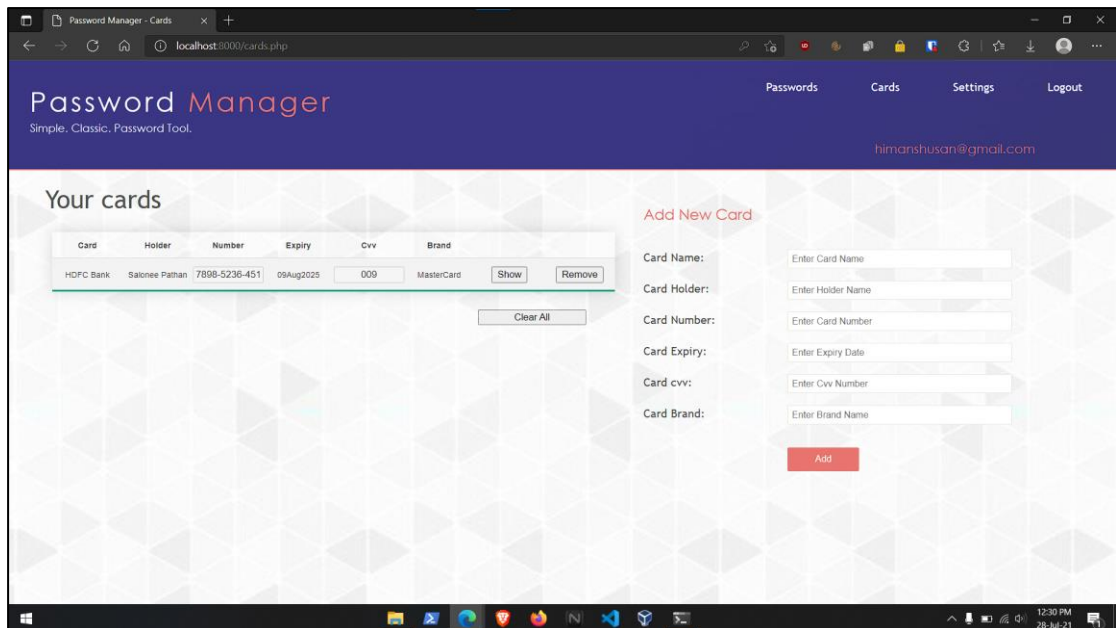
Once the cards details are saved in the database, the details are displayed in the ‘Your Cards’ section. The card number and CVV are encrypted and are hence hidden.

The screenshot shows the 'Password Manager - Cards' interface after a card has been saved. The 'Your cards' section now displays a table with one entry:

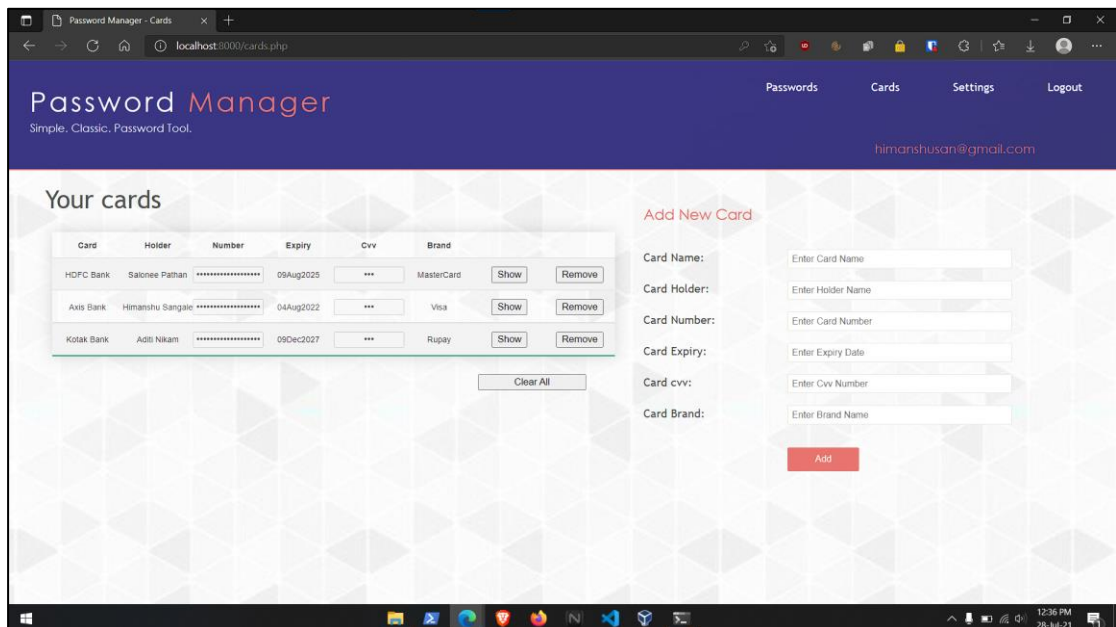
Card	Holder	Number	Expiry	Cvv	Brand
HDFC Bank	Salonee Pathan	*****	09Aug2025	***	MasterCard

Below the table, there are 'Show' and 'Remove' buttons for the card, and a 'Clear All' button. The 'Add New Card' form on the right is reset, with input fields for Card Name, Card Holder, Card Number, Card Expiry, Card cvv, and Card Brand, each with a placeholder text like 'Enter Card Name'. An 'Add' button is at the bottom of the form.

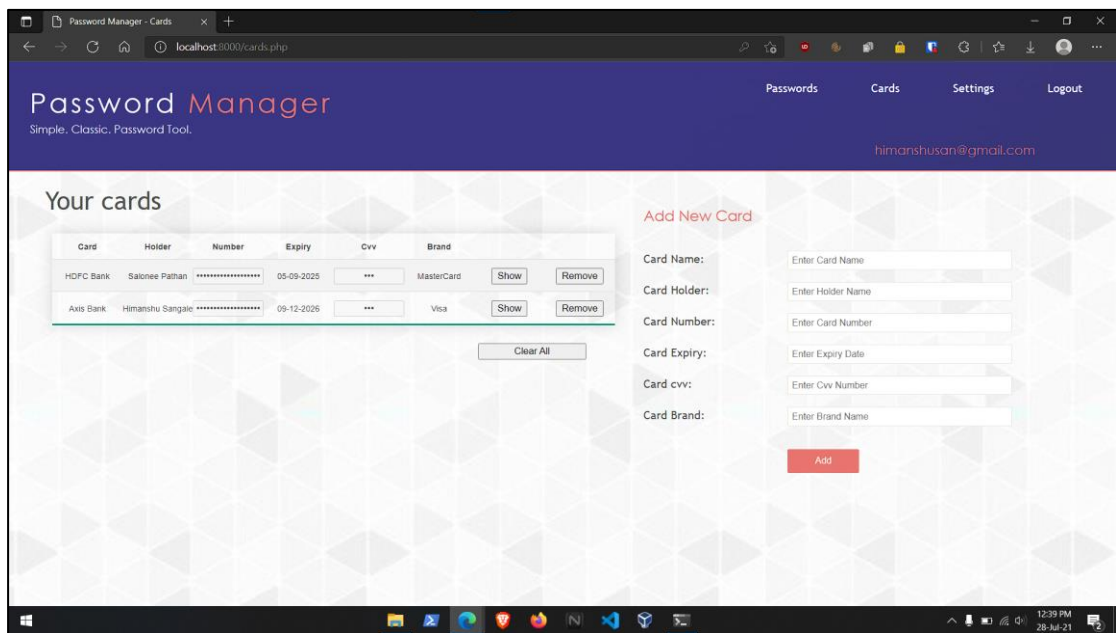
The user can view his/her card details by clicking on the show button. Doing this, the CVV and Card Number is visible only for a fraction of a second. To keep them visible for a longer while, the user can press the show button and move the cursor away from the button and then release it.



Multiple passwords can be saved in a similar fashion.

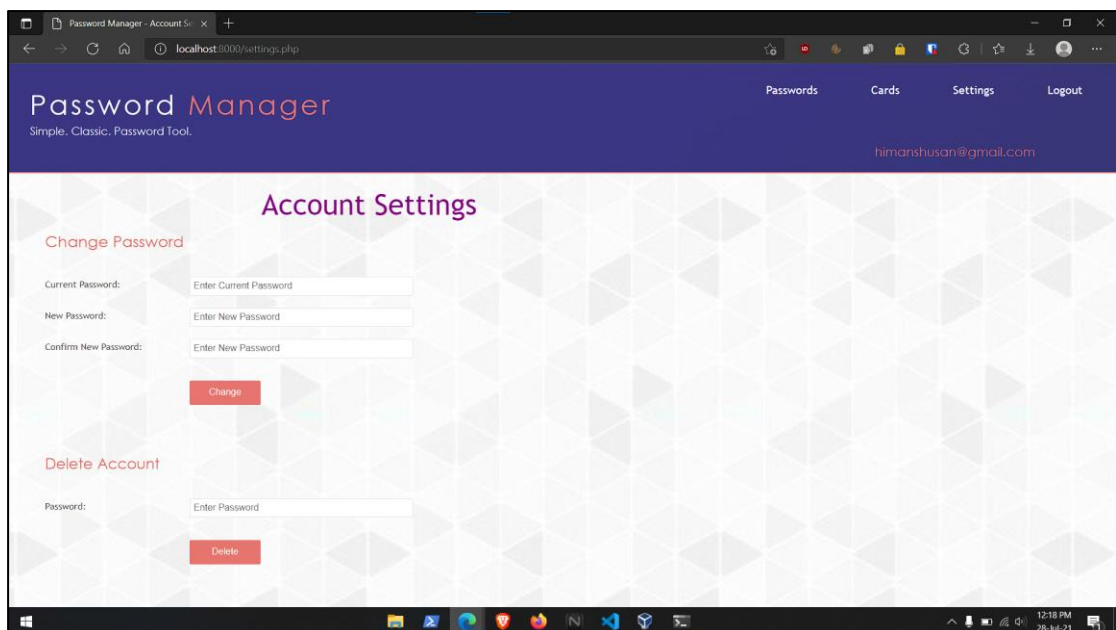


By clicking on the remove button, the user can delete the stored card details.

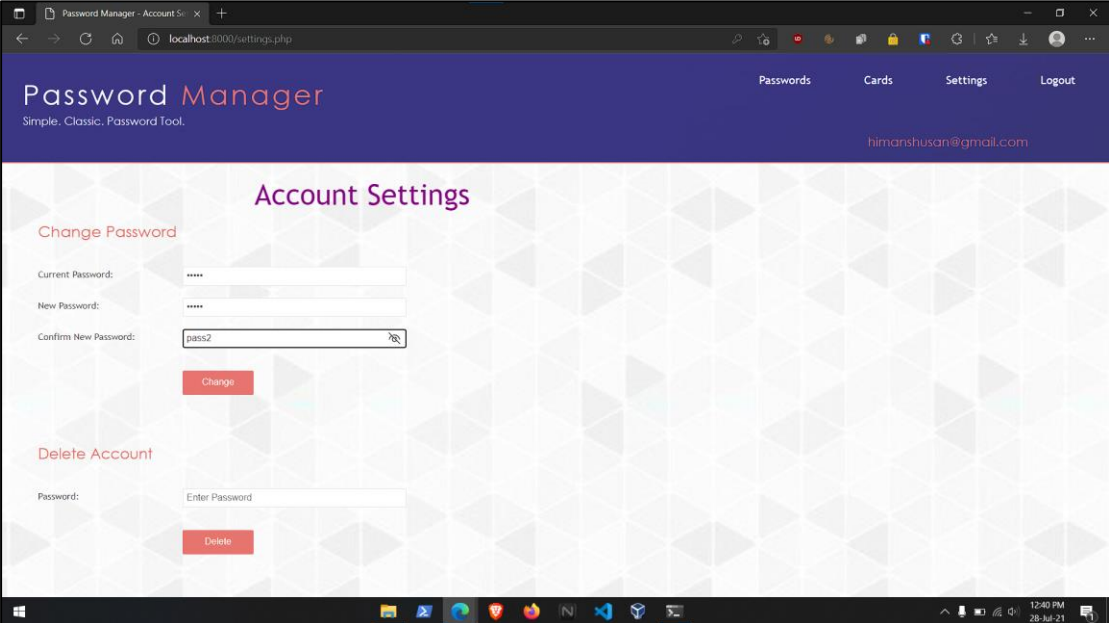


Account Module

The account settings can be changed using this module. The user can change his/her master password with which the account can be accessed.

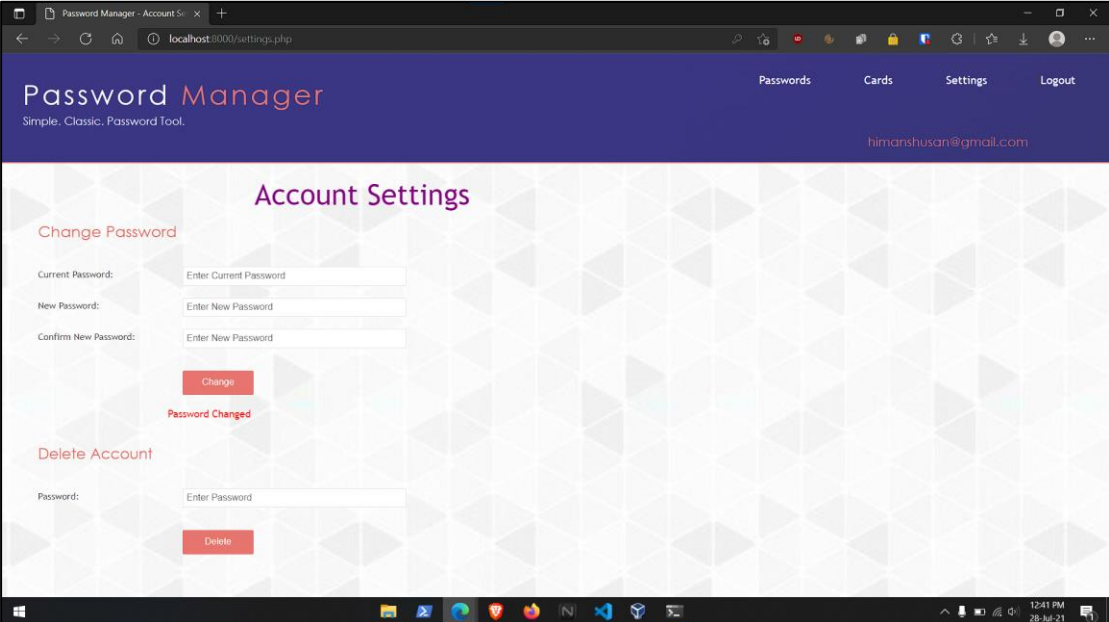


To change the password, the user must enter the current password and then enter the new password and confirm it. When the change button is clicked, the master password will be changed and the next time the user should use this password for logging in.



The screenshot shows the 'Account Settings' page of the Password Manager application. The page has a dark blue header with the title 'Password Manager' and the tagline 'Simple, Classic, Password Tool.' Below the header, there are navigation links for 'Passwords', 'Cards', 'Settings', and 'Logout'. The user's email 'himanshusan@gmail.com' is displayed in the top right corner. The main content area is titled 'Account Settings' and contains two sections: 'Change Password' and 'Delete Account'. The 'Change Password' section has three input fields: 'Current Password:', 'New Password:', and 'Confirm New Password:'. The 'Confirm New Password' field contains the text 'pass2'. Below these fields is a red 'Change' button. The 'Delete Account' section has a single input field labeled 'Password:' with the placeholder text 'Enter Password' and a red 'Delete' button below it.

To delete the account, the user can click on the delete button and then it will redirect to the login page.



This screenshot shows the 'Account Settings' page after the password change operation. The 'Change Password' section now displays three input fields with placeholder text: 'Enter Current Password', 'Enter New Password', and 'Enter New Password'. A red 'Change' button is present, and a red message 'Password Changed' is displayed below it. The 'Delete Account' section remains unchanged, with the 'Password:' input field and the 'Delete' button.

10. ADVANTAGES AND DISADVANTAGES

Advantages of Password Manager:

1. Encrypted password storage
2. Quick password retrieval
3. Password generation according to the user specifications
4. Secure system which allows access only using master-password
5. Password copying and reuse

FUTURE SCOPE

The application is built with a view to make password generation, storage and copying easier. Every single day, we logon and register to a variety of different websites and applications. As humans, we usually tend to forget the passwords that are used onto various websites. The application, as per our analysis, is one of the necessary applications and will have a huge demand in the future.

CONCLUSION

Thus, we have successfully completed the creation of the Password Manager application.

REFERENCES:

- [1]<http://www.1202performance.com/atricles/how-to-write-performance-requirements-with-example/>
- [2]<https://www.guru99.com/functional-requirement-specification-example.html>
- [3]<https://www.geeksforgeeks.org/types-of-feasibility-study-in-software-project-development/>