**PROJECT REPORT ON**

**PASSWORD MANAGEMENT SYSTEM**

Project report submitted in partial fulfillment of the requirements for the award

of the degree of

**Bachelor of Computer Application (BCA)**

Submitted by:

|  |  |  |
| --- | --- | --- |
| ***Naorem Tamphasana***  ***Keithellakpam Oken***  ***Wigabou Abonmei***  ***Phanjoubam Manjit***  ***Jackson Yanglem*** | ***17310478***  ***18560205***  ***18560140***  ***18560141***  ***18560142*** | ***BC1097 of 2017***  ***BC1098 of 2018***  ***BC1099 of 2018***  ***BC1100 of 2018***  ***BC1101 of 2018*** |

Under the guidance of

**Lairenjam Ithoi Chanu**

Scientist-C,

NIELIT Imphal Centre

**May 2021**

**UNDERTAKING**

We declare that the work which is being presented in this project report entitled “ONLINE *PASSWORD MANAGEMENT SYSTEM*”, submitted to the National Institute of Electronics and Information Technology (NIELIT), Imphal Centre, for the award of the degree of ***Bachelor of Computer Application***, is an authentic record of our own work carried out under the supervision and guidance of Lairenjam Ithoi Chanu. We have not plagiarized or submitted the same work for the award of any other degree. In case this undertaking is found incorrect, we accept that our degree may be unconditionally withdrawn.

MAY,2021

IMPHAL

Submitted by:-

Naorem Tamphasana (BCA)

Keithellakpam Oken (BCA)

Wigabou Abonmei (BCA)

Phanjoubam Manjit (BCA)

Jackson Yanglem (BCA)

**Lairenjam Ithoi Chanu**

Scientist-C (project supervisor)

NIELIT, Imphal Centre, Manipur

**Th. Prameshwor Singh** Executive Director

NIELIT, Imphal Center, Manipur

**CERTIFICATE OF APPROVAL**

This is to certify that the project entitled “Online *Password Management System*” submitted *,*to the National Institute of Electronics and Information Technology, Imphal Centre, Manipur, India, for the award of the degree of Bachelor of Computer Application, is a record of an original project work carried out by them under our supervision and guidance. The project fulfils all the requirements as per the regulations of this Institute. Neither this project nor any part of it has been submitted for any degree or academic award elsewhere to the best of our knowledge.

**Submitted by:-**

Naorem Tamphasana Enroll no.: 17310478 Roll No.: BC1097 of 2017

Keithellakpam Oken Enroll no.: 18560205 Roll No.: BC1098 of 2018

Wigabou Abonmei Enroll no.: 18560140 Roll No.: BC1099 of 2018

Phanjoubm Manjit Enroll no.: 18560141 Roll No.: BC1100 of 2018

Jackson Yanglem Enroll no.: 18560142 Roll no.: BC1101 of 2018

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Lairenjam Ithoi Chanu**

Scientist-C

(project supervisor)

NIELIT, Imphal Centre, Manipur

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Th. Prameshwor Singh** Executive Director

NIELIT, Imphal Center Manipur

**CERTIFICATE**

This is to certify that the project entitled “Online *Password Management System*” submitted by the following students is approved for the award of Degree of Bachelor of Computer Application and the project evaluation and viva-voice examination has been held on

date.

|  |  |  |
| --- | --- | --- |
| Name of the Student | Roll no. | Registration No. |
| Naorem Tamphasana  Keithellakpam Oken  Wigabou Abonmei  Phanjoubam Manjit  Jackson Yanglem | BC1097  BC1098  BC1099  BC1100  BC1101 | 17310478 of 2017  17310478 of 2018  18560140 of 2018  18560141 of 2018  18560142 of 2018 |

INTERNAL EXAMINER EXTERNAL EXAMINER

# Acknowledgement

We would like to take this opportunity to express our deep sense of gratitude to all who helped us directly or indirectly during this thesis work.

Firstly, we would like to thank our supervisor, **Lairenjam Ithoi Chanu**, Scientist-C, NIELIT Imphal Centre, for being a great mentor and the best adviser we could ever have. Her advice, encouragement and critics are source of innovative ideas, inspiration and causes behind the successful completion of this dissertation. The confidence shown to us by her was the biggest source of inspiration for us. It has been a great privilege working with her from last six months.

We are highly obliged to all the faculty members of NIELIT Imphal Centre for their support and encouragement. We also would like to thank **Shri Prameshwor Singh,** Executive Director, NIELIT Imphal Centre for providing excellent computing and other facilities without which this work could not achieve its quality goal.

Finally, we are grateful to our parents and friends for their support. It was impossible for us to complete this thesis work without their love, blessing and encouragement.

**Naorem Tamphasana**

**Keithellakpam Oken**

**Wigabou Abonmei**

**Phanjoubam Manjit**

**Jackson Yanglem**

# Dedicated to

*To our Loving Family,*

*Friends for their kind love & support.*

*To our project supervisor/guidance Lairenjam Ithoi Chanu for sharing his valuable**knowledge,*

*Encouragement& showing confidence on us all the time*

***Where there’s a will there’s a way!***

*One should have unflincing , undaunting wish*

*to achive something.*

*If you have a will to achive your goal, you will positively overcome the difficulties*

*that come in your way…….*

*Path of success runs through many labyrinths of failures.*

*One, who has a will to achive success , never gets frustrated by the odds and failures of the path…’*

***inspirational quotes…***

# PREFACE

A project is an important entity in the line of professional courses for the very fact that it brings out the relationship between the theoretical knowledge and their practical implementation.

In today’s world of the advancement and essentials of the Information Technology in day-to-day life, software creation and development have become a culture in its own self, with Management Information Software being the heart of the talk. The biggest problem that is being encountered by Organizations or Govt. Departments today is that of maintaining database of information that they need a may have access to at any point of time in a convenient and a guiding manner.

Automation refers to computerization of a manual system. The aim is to reduce labor, time and to achieve reliable output. The automation in short, increases staff productivity and provides efficient management of process and time.

Keeping such aims and objectives in our minds, we work towards the Automation of a System. Last but not the least, we hope that all our efforts will be appreciated and the software will be put into practical use.

# Abstract

This project “Online Password Management System” has been developed on Python using the python framework called **Django**. With the use of this framework, building fast, scalable and secure websites has never been made easier and addition, open source with is free to use!. With built-in login and signup functionality and also, a option to reset password, Users can easily login, sign up and reset their password without having to worry about losing their accounts. Making sure that 2FA is implemented for better security, users need to verify their login by entering the OTP that was sent to their email to verify their login attempt. And of course, all of the data are encrypted and salted by using the python **cryptography** library. The secret key is only know to the developer and the admin doesn’t have the authority to see the decrypted hased password of the users and its account info data. It can be use anywhere online as long as one has an Internet connection available to them.

**Contents**

Declaration i

Acknowledgement ii

Abstract iii

Chapter 1 1

General Introduction 1

1.1 Introduction: 1

1.2 Purpose: 1

1.3 Scopes And Objective: 1

1.4 Software Selection: 2-9

Chapter *2* 10

System Analysis 10

2.1 Overview Of The System: 10

2.2 Preliminary Investigation: 10

2.3 Feasibility Studies: 10

Chapter *3* 11

System Design 11

3.1 Dbms Overview: 11-12

3.2 Database Design 13-14

3.3 Entity Relatioship Diagram 14-16

3.4 Data Flow Diagram 17-18

3.5 Context Diagram 19

Chapter *4* 18

System Testing And Implementation 18

4.1 System Testing: 18

4.2 Requirements: 18

4.3 Pasword Management Window: 19-24

4.4 Drawback And Limitation: 24

Chapter *5* 25

**Source Code** 25-45

Chapter *6* 46

Test Case 46-48

Chapter *7* 49

Conclusion 49

Chapter *8* 50

**Bibliography** 50

**List of Figures**

Database Table Figure 10

Entity Relationship Diagram 12

Data Flow Diagram 14

Context Diagram 15

**List of Tables**

Password Management Details Table 10

User Table 10

Password Table 10

# *Chapter 1*

# General Introduction

## **Introduction**

The project, entitile, “Online Password Management System” is a Web App built using the Python web framework, called Django which is popularly known. In this project, new users are able to sign up using their email address and existing users may sign in using the username they used to create along with their password. When logging in, the system sends a OTP to verify their login attempt sent to their email. Users are then, are able to add their own set off account deatails along with their password. Since it is made to be available for online uses, user only need an internet connection and a device that is capable to use a web browser.

## **1.2 PURPOSE:**

The Purpose of this system is to make sure that people can make use of password mangers to their potential. Tradional way of storing passwords on a piece of paper/sheet which is highly secure or even remembering complex passwords, because simple password can be hacked or social engineered by hackers.

This app allows the user to do just that, easily and securely storing those personal and private accounts with a ecryption and applying salt to the respective confidential details. And additional otp verification to ensure 2FA for better security option.

**1.3 Scopes and Objectives:**

The main aim of this module is to make people maintaining their online accounts easier and safe and available anywhere on the internet. With our cryptographic encryption and 2FA for login verification, its safe and secure to add and store account details along with their respective passwords and account details.

**1.4 Software Selection:**

The development of any system starts with an explicit and accomplished study of the available tools and the environment. The capacity and features of the system to be developed depends on the software used for the development. Each of the software is used particularly taking into account of their usefulness and the reliability. The environment/platform plays an important role. Since, it should be capable of supporting all the facilities to be incorporated to the system, hardware support is also important. The considerations for developing the system are the graphical user’s interface and mouse facility.

A brief description of the environment (operating system) and the tools are listed as follows:



Fig: 1.1

Windows 10 - An overview

Windows 10 (codenamed Blue) is an upgrade for Windows 8, a version of Windows NT, a computer operating system released by Microsoft. First unveiled and released as a public beta in June 2013, it was released to manufacturing on August 27, 2013, and reached general availability on October 17, 2013, almost a year after the retail release of its predecessor. Windows 8.1 is available free of charge for retail copies of Windows 8 and Windows RT users via the Windows Store. Unlike service packs on previous versions of Windows, users who obtained 8 outside of retail copies or pre-loaded installations (i.e., volume licensing) must obtain 10 through new installation media from their respective subscription or enterprise channel.

Microsoft's support lifecycle policy treats Windows 8.1 similar to previous service packs of Windows: It is part of Windows 8's support lifecycle, and installing 8.1 is required to maintain access to support and Windows updates after January 12, 2016. However, unlike previous service packs, Windows 8.1 cannot be acquired via Windows Update and only accepts 10-specific product keys.

Released as part of a shift by Microsoft towards regular yearly major updates for its software platforms and services, Windows 10 aims to address complaints of Windows 8 users and reviewers on launch. Visible enhancements include an improved Start screen, additional snap views, additional bundled apps, tighter One Drive (formerly

SkyDrive) integration, Internet Explorer 11, a Bing-powered unified search system, restoration of a visible Start button on the taskbar, and the ability to restore the previous behavior of opening the user's desktop on login instead of the Start screen. Windows 10 also added support for such emerging technologies as high-resolution displays, 3D printing, Wi-Fi Direct, and Mira cast streaming.

Windows 8.1 received relatively positive reception, with critics praising the expanded functionality available to apps in comparison to 8, its One Drive integration, along with its user interface tweaks and the addition of expanded tutorials for operating the Windows 8 interface. Despite these improvements, 10 was still criticized for not addressing all digressions of Windows 8 (such as a poor level of integration between Metro-style apps and the desktop interface), and the potential privacy implications of the expanded use of online services.



Fig: 1.2

Python 3.9

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.



**Fig: 1.3**

django

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support.

Django helps you write software that is:

Complete

Django follows the "Batteries included" philosophy and provides almost everything developers might want to do "out of the box". Because everything you need is part of the one "product", it all works seamlessly together, follows consistent design principles, and has extensive and [up-to-date documentation](https://docs.djangoproject.com/en/stable/).

Versatile

Django can be (and has been) used to build almost any type of website — from content management systems and wikis, through to social networks and news sites. It can work with any client-side framework, and can deliver content in almost any format (including HTML, RSS feeds, JSON, XML, etc). The site you are currently reading is built with Django!  
  
Internally, while it provides choices for almost any functionality you might want (e.g. several popular databases, templating engines, etc.), it can also be extended to use other components if needed.

Secure

Django helps developers avoid many common security mistakes by providing a framework that has been engineered to "do the right things" to protect the website automatically. For example, Django provides a secure way to manage user accounts and passwords, avoiding common mistakes like putting session information in cookies where it is vulnerable (instead cookies just contain a key, and the actual data is stored in the database) or directly storing passwords rather than a password hash.  
  
A password hash is a fixed-length value created by sending the password through a [*cryptographic hash function*](https://en.wikipedia.org/wiki/Cryptographic_hash_function). Django can check if an entered password is correct by running it through the hash function and comparing the output to the stored hash value. However due to the "one-way" nature of the function, even if a stored hash value is compromised it is hard for an attacker to work out the original password.  
  
Django enables protection against many vulnerabilities by default, including SQL injection, cross-site scripting, cross-site request forgery and clickjacking (see [Website security](https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Website_security) for more details of such attacks).

Scalable

Django uses a component-based “[shared-nothing](https://en.wikipedia.org/wiki/Shared_nothing_architecture)” architecture (each part of the architecture is independent of the others, and can hence be replaced or changed if needed). Having a clear separation between the different parts means that it can scale for increased traffic by adding hardware at any level: caching servers, database servers, or application servers. Some of the busiest sites have successfully scaled Django to meet their demands (e.g. Instagram and Disqus, to name just two).

Maintainable

Django code is written using design principles and patterns that encourage the creation of maintainable and reusable code. In particular, it makes use of the Don't Repeat Yourself (DRY) principle so there is no unnecessary duplication, reducing the amount of code. Django also promotes the grouping of related functionality into reusable "applications" and, at a lower level, groups related code into modules (along the lines of the [Model View Controller (MVC)](https://developer.mozilla.org/en-US/docs/Glossary/MVC) pattern).

Portable

Django is written in Python, which runs on many platforms. That means that you are not tied to any particular server platform, and can run your applications on many flavours of Linux, Windows, and Mac OS X. Furthermore, Django is well-supported by many web hosting providers, who often provide specific infrastructure and documentation for hosting Django sites.



Fig: 1.4

HTML 5

The **HyperText Markup Language**, or **HTML** is the standard [markup language](https://en.wikipedia.org/wiki/Markup_language) for documents designed to be displayed in a [web browser](https://en.wikipedia.org/wiki/Web_browser). It can be assisted by technologies such as [Cascading Style Sheets](https://en.wikipedia.org/wiki/Cascading_Style_Sheets) (CSS) and [scripting languages](https://en.wikipedia.org/wiki/Scripting_language) such as [JavaScript](https://en.wikipedia.org/wiki/JavaScript).

[Web browsers](https://en.wikipedia.org/wiki/Web_browser) receive HTML documents from a [web server](https://en.wikipedia.org/wiki/Web_server) or from local storage and [render](https://en.wikipedia.org/wiki/Browser_engine) the documents into multimedia web pages. HTML describes the structure of a [web page](https://en.wikipedia.org/wiki/Web_page) [semantically](https://en.wikipedia.org/wiki/Semantic_Web) and originally included cues for the appearance of the document.

[HTML elements](https://en.wikipedia.org/wiki/HTML_element) are the building blocks of HTML pages. With HTML constructs, [images](https://en.wikipedia.org/wiki/HTML_element#Images_and_objects) and other objects such as [interactive forms](https://en.wikipedia.org/wiki/Fieldset) may be embedded into the rendered page. HTML provides a means to create [structured documents](https://en.wikipedia.org/wiki/Structured_document) by denoting structural [semantics](https://en.wikipedia.org/wiki/Semantics) for text such as headings, paragraphs, lists, [links](https://en.wikipedia.org/wiki/Hyperlink), quotes and other items. HTML elements are delineated by *tags*, written using [angle brackets](https://en.wikipedia.org/wiki/Bracket#Angle_brackets). Tags such as <img /> and <input /> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a [scripting language](https://en.wikipedia.org/wiki/Scripting_language) such as [JavaScript](https://en.wikipedia.org/wiki/JavaScript), which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The [World Wide Web Consortium](https://en.wikipedia.org/wiki/World_Wide_Web_Consortium) (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.[[2]](https://en.wikipedia.org/wiki/HTML#cite_note-deprecated-2)



Fig 1.5

CSS3

**Cascading Style Sheets** (**CSS**) is a [style sheet language](https://en.wikipedia.org/wiki/Style_sheet_language) used for describing the [presentation](https://en.wikipedia.org/wiki/Presentation_semantics) of a document written in a [markup language](https://en.wikipedia.org/wiki/Markup_language) such as [HTML](https://en.wikipedia.org/wiki/HTML).[[1]](https://en.wikipedia.org/wiki/CSS#cite_note-1) CSS is a cornerstone technology of the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web), alongside HTML and [JavaScript](https://en.wikipedia.org/wiki/JavaScript).[[2]](https://en.wikipedia.org/wiki/CSS#cite_note-2)

CSS is designed to enable the separation of presentation and content, including [layout](https://en.wikipedia.org/wiki/Page_layout), [colors](https://en.wikipedia.org/wiki/Color), and [fonts](https://en.wikipedia.org/wiki/Typeface).[[3]](https://en.wikipedia.org/wiki/CSS#cite_note-3) This separation can improve content [accessibility](https://en.wikipedia.org/wiki/Accessibility), provide more flexibility and control in the specification of presentation characteristics, enable multiple [web pages](https://en.wikipedia.org/wiki/Web_page) to share formatting by specifying the relevant CSS in a separate .css file which reduces complexity and repetition in the structural content as well as enabling the .css file to be [cached](https://en.wikipedia.org/wiki/Cache_(computing)) to improve the page load speed between the pages that share the file and its formatting.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or [screen reader](https://en.wikipedia.org/wiki/Screen_reader)), and on [Braille-based](https://en.wikipedia.org/wiki/Braille_display) tactile devices. CSS also has rules for alternate formatting if the content is accessed on a [mobile device](https://en.wikipedia.org/wiki/Mobile_device).[[4]](https://en.wikipedia.org/wiki/CSS#cite_note-4)

The name *cascading* comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

The CSS specifications are maintained by the [World Wide Web Consortium](https://en.wikipedia.org/wiki/World_Wide_Web_Consortium) (W3C). Internet media type ([MIME type](https://en.wikipedia.org/wiki/MIME_media_type)) text/css is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free [CSS validation service](https://en.wikipedia.org/wiki/W3C_Markup_Validation_Service#CSS_validation) for CSS documents.[[5]](https://en.wikipedia.org/wiki/CSS#cite_note-5)



Fig: 1.6

**Sublime Text**

**Sublime Text** is a text editor and source code editor for use with Microsoft Windows . It supports tabbedd editing, which allows working with multiple open files in a single window. The project's name comes from the Cincrement operator



Fig: 1.7

**sqlite**

SQLite is an in-process library that implements a [self-contained](https://www.sqlite.org/selfcontained.html), [serverless](https://www.sqlite.org/serverless.html), [zero-configuration](https://www.sqlite.org/zeroconf.html), [transactional](https://www.sqlite.org/transactional.html) SQL database engine. The code for SQLite is in the [public domain](https://www.sqlite.org/copyright.html) and is thus free for use for any purpose, commercial or private. SQLite is the [most widely deployed](https://www.sqlite.org/mostdeployed.html) database in the world with more applications than we can count, including several [high-profile projects.](https://www.sqlite.org/famous.html)

# Chapter 2

# SYSTEM ANALYSIS

**2.1 Overview of the system**

This system is intended to help users to store and maintain their accounts details and their password easy and secure. This system overcomes the tradition way of writing down passwords or rememberiung them. Password especially needs to be complex and unknowable to outsiders which makes it hard to remember or exreemly at risk when we write it down on a paper or book.

In this way the our app can easily accomplish this by overcoming the weak system of traditionally managing password by writing them down or remembering which inevitably will get forgotten or lost

**2.2 Preliminary Investigation**

Before the actual development of the project "ONLINE PASSWORD MANAGEMENT SYSTEM" a thorough preliminary investigation is done at NIELIT IMPHAL CENTRE with the help of numerous faculties working in the institute who not only gives the require proceedings of developing the system but also support us throughout the development process until final completion of the project.

**2.3 Feasibility Studies**

After thorough preliminary investigation, all the require technical support and economic funding is done by the institute in time thus allowing us to complete the system in time.

# Chapter *3*

# SYSTEMDESIGN

**3.1 DBMS OVERVIEW**

A Database Management System (DBMS) is a software package with computer programs that control the creation, maintenance, and the use of a database. It allows organizations to conveniently develop databases for various applications by database administrators (DBAs) and other specialists. A database is an integrated collection of data records, files, and other database objects needed by an application. A DBMS allows different user application programs to concurrently access the same database. DBMSs may use a variety of database models, such as the relational model or object model, to conveniently describe and support applications. It typically supports query languages, which are in fact high-level programming languages, dedicated database languages that considerably simplify writing database application programs. Database languages also simplify the database organization as well as retrieving and presenting information from it. A DBMS provides facilities for controlling data access, enforcing data integrity, managing concurrency control, recovering the database after failures and restoring it from backup files, as well as maintaining database security.

**A Database System:**

A database is an organized collection of data for one or more purposes, usually in digital form. The data are typically organized to model relevant aspects of reality (for example, the availability of rooms in hotels), in a way that supports processes requiring this information (for example, finding a hotel with vacancies).

The term "database" refers both to the way its users view it, and to the logical and physical materialization of its data, content, in files, computer memory, and computer data storage. This definition is very general, and is independent of the technology used. However, not every collection of data is a database; the term database implies that the data is managed to some level of quality (measured in terms of accuracy, availability, usability, and resilience) and this in turn often implies the use of a general-purpose Database management system (DBMS). A general - purpose DBMS is typically a complex software system that meets many usage requirements, and the databases that it maintains are often large and complex.

The term database is correctly applied to the data itself, and is different from the DBMS which is a software system that allows storing and changing the data, as well as retrieving information from it. The structure of a database is generally too complex to be handled without its DBMS, and any attempt to do otherwise is very likely to result in data corruption. DBMS are packaged as computer software products: well-known products include the Oracle DBMS, Access and SQL Server from Microsoft, DB2 from IBM and the Open source DBMS MySQL. Each such DBMS product currently supports many thousands of databases all over the world. The stored data in a database is not generally portable across different DBMS, but can inter-operate to some degree (while each DBMS type controls a database of its own database type) using standards like SQL and ODBC. A successful general-purpose DBMS is designed in a way that is can satisfy as many as possible different applications and application designers. A DBMS also needs to provide effective run-time execution to properly support (e.g., in terms of performance, availability, and security) as many end-users (the database's application users) as needed. Sometimes the combination of a database and its respective DBMS is referred to as a Database system (DBS).

**The Relational Approach:**

The due considerations have been given on the relational model of Database System. The implementations of the project are based on the relational approach. The reasons are obvious – the uncomplicated and easy to implement approach.

The main feature of the relational data structure is that associations between tuples (rows) are represented solely by data values in columns drawn from a common domain. It is a characteristic of the relational approach that all the information in the database both “entities” and “relationship” is represented in a single uniform manner namely in the form of the tables. It is clear that relational structure is very easy to understand. The uniformity of data representation leads to a corresponding uniformity in the operator set.

**3.2 Database Design**

The following are the tables with the fields and field properties used in the system**: -**

**1.Online Password Management User table:** This table stores the username, email address and password for the password management page to log in**.**

**2. Online Password Management passwords table:** This table stores the user (which is a foreign key link to primary key to the username of User table), email address, url, name, password and logo for the registered user to add their online accounts.

**1. User Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| SL. NO | FIELD NAME | TYPE | DESCRIPTION |
| 1 | username | PRIMARY KEY | A unique identity and the Name of the account |
| 2. | password | string | The account password |
| 3. | email | string | Email used for the account |

**2. Password Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl. No. | FIELD NAME | TYPE | DESCRIPTION |
| 1. | user | FOREGIN KEY | This is used to link to the address of the username column from the table “user” |
| 2. | name | string | the url of the website |
| 3. | email | string | the email used |
| 4. | password | string | the password |
| 5. | logo | image | the image that is generated from the url |

**3.4 Data Flow Diagram (DFD)**

A data flow diagram (DFD) is a graphical representation of the "flow" of data through a system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing(structured design).DFDs consist of four major components: entities, processes, data stores, and data flows. The symbols used to depict how these components interact in a system are simple and easy to understand.

The data flow diagram represents all the possible flow through the software system. It can give us the knowledge about the software pagination and flow of control.

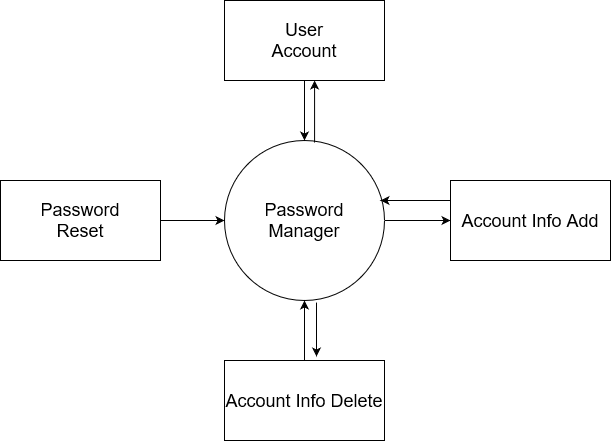
Symbols and their representations used in drawing a data flow diagram:

**Table: 3.9**

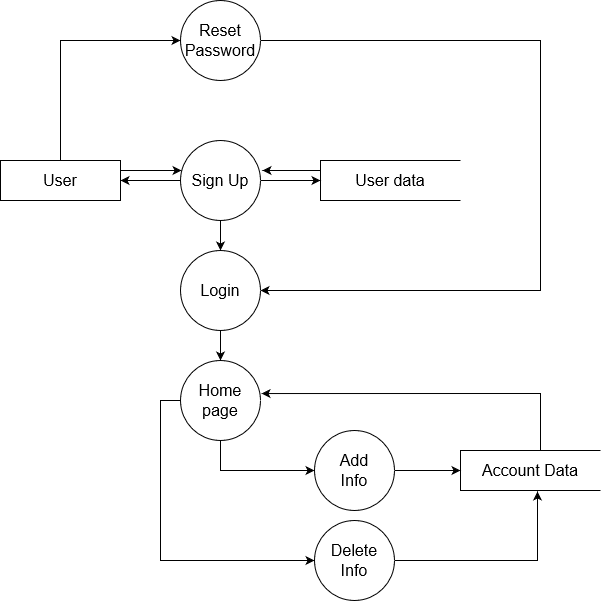
|  |  |  |  |
| --- | --- | --- | --- |
| Sl No. | NAME | SYMBOL | DESCRIPTION |
| 1 | Rectangle | |  | | --- | |  | | External Entities |
| 2 | Circle |  | Process |
| 3 | Arrow |  | Data Flow |
| 4 | Parallel Line |  | Data Store |

The data flow diagram can be shown as follows:

**Fig: 3.3 Data Flow Diagram Level 0 (DFD Level 0)**



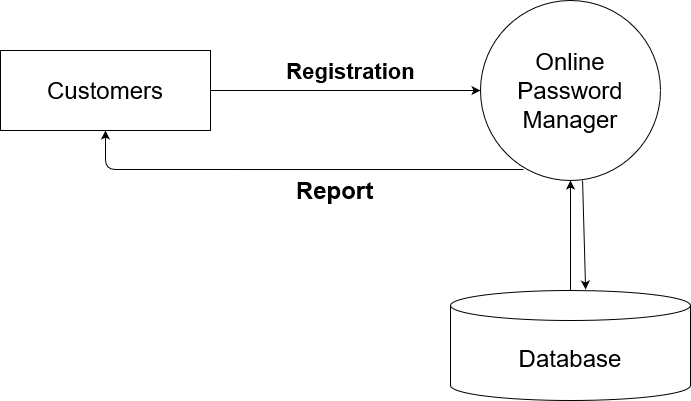
**Fig: 3.4 Data Flow Diagram Level 1 (DFD Level 1)**

****

**3.5 Context Diagram**

The context diagram shows the nature of the software system. This top level diagram is often called a context diagram. Context Diagram contains a single process and it shows an overall view of the system under development.

The context level DFD diagram can be shown as:



**Fig: 3.5 Context Diagram**

# Chapter *4*

# SYSTEM TESTING AND IMPLEMENTATION

## **4.1 System Testing:**

In the development stages, the software was tested with artificial values made up solely for the purpose of testing. All the data were tested individually. In these tests, steps were taken to remove the errors that occurred when testing individually. After a lot of debugging and re-coding of procedures and functions, these tests yield positive results. All the data, encryption and decryption were checked and rechecked to see whether outputs were correct, and also in the generation of report, positive results were achieved, after debugging a few errors.

**4.2 Requirements:**

**Hardware Requirements**

The software can be run on any Windows Operating System with a processor of around 2.20GH, hard disk space of 1 Tb and 4Gb of RAM. The software was originally developed on a computer running Windows 10, with a intel core i5 with intel® core ™ i5-5200U CPC processor speed, networking (Ethernet) capabilities. The screen resolution is 1024 X 768. There should not be much problem on other features.

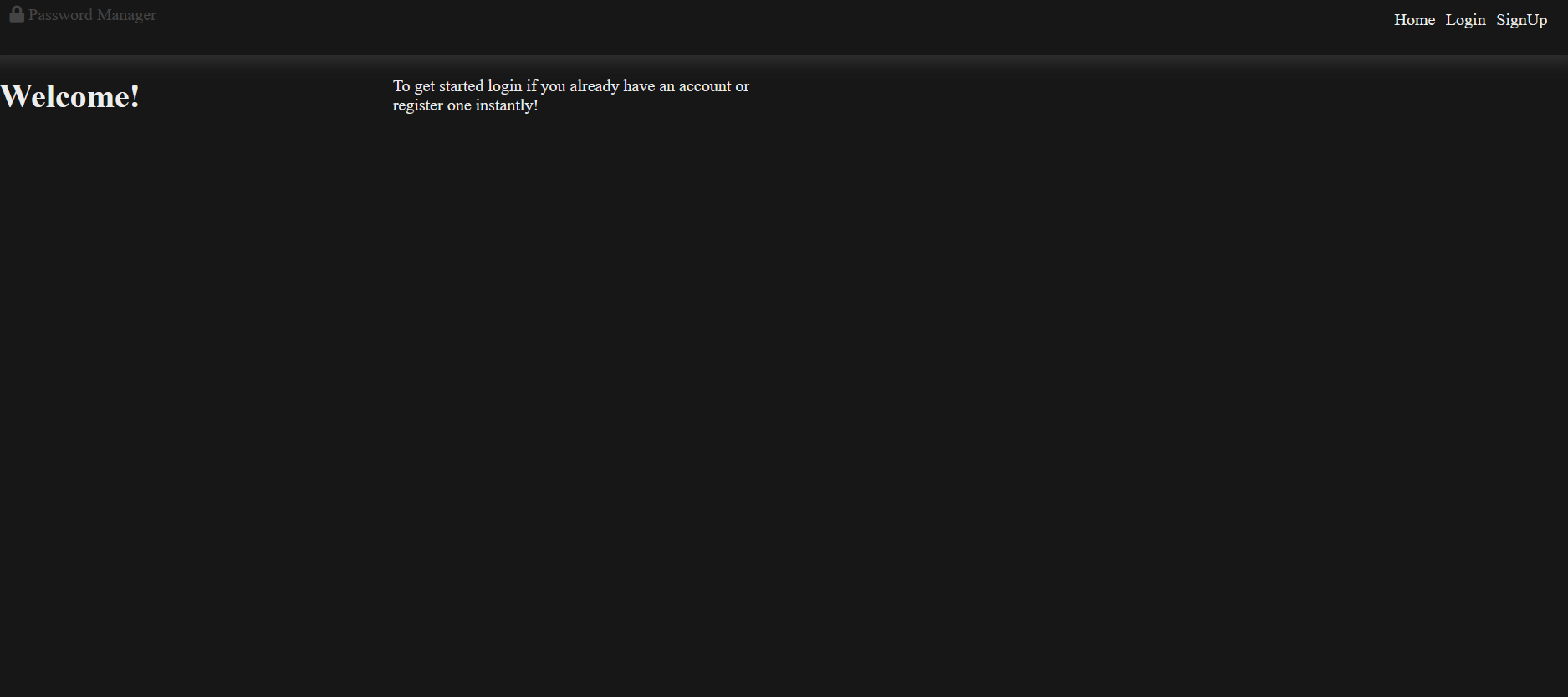
**Software Requirements**

Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Since the platform of Python is multi platform, it is easy to run the code on any Operating Software machine

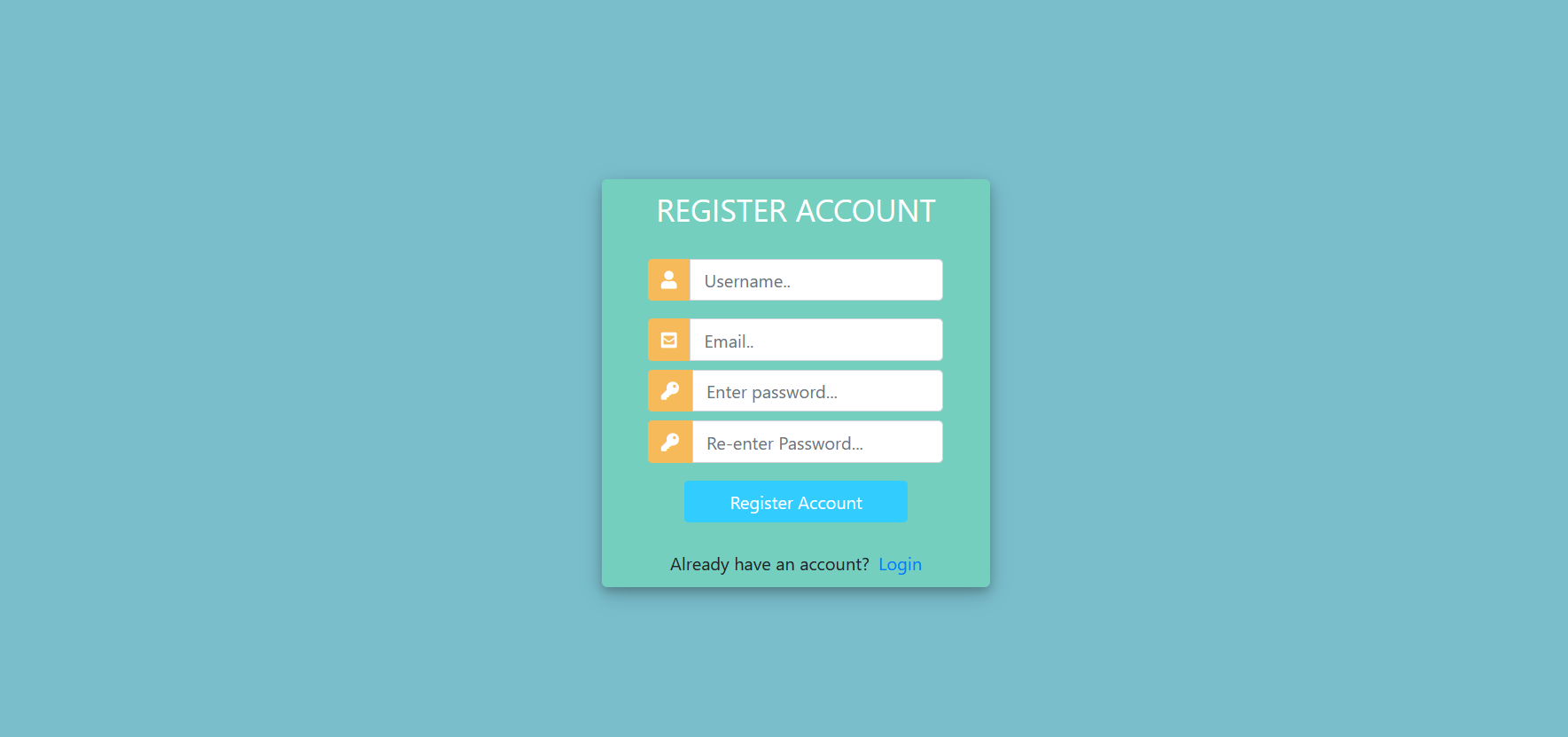
**4.3 Screenshots:**

The homepage starts with a blank notice with a welcoming message saying “Welcome! To get started login if you already have an account or register one instantly!”, gives the slight indiction to the user of what to do and how to get started using the Online Password Manager.

**Homepage:** This is the Homepage when a user loads the website



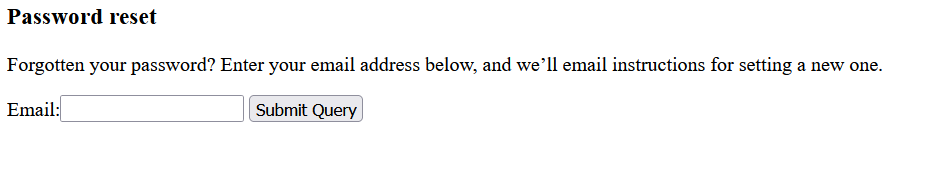
**Register Window**: This is the register window that is displayed when user clicks on the register button



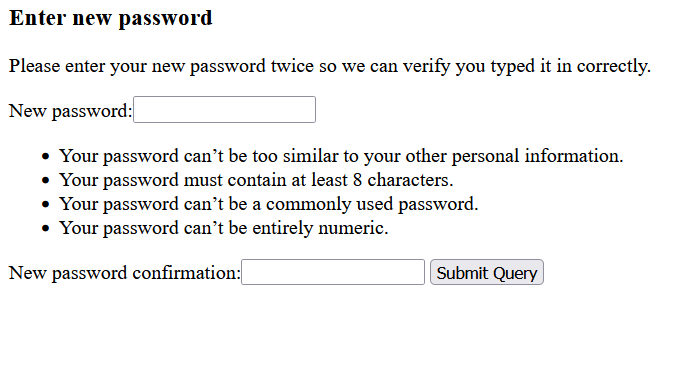
**Login Window:** This is the window that is displayed when user clicks on the login button



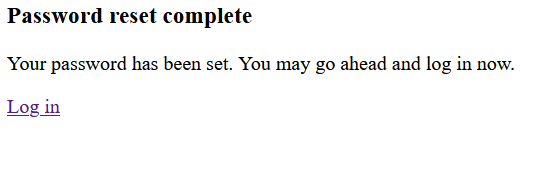
**Reset Password:** This is the window if user wants to reset their forgotten password



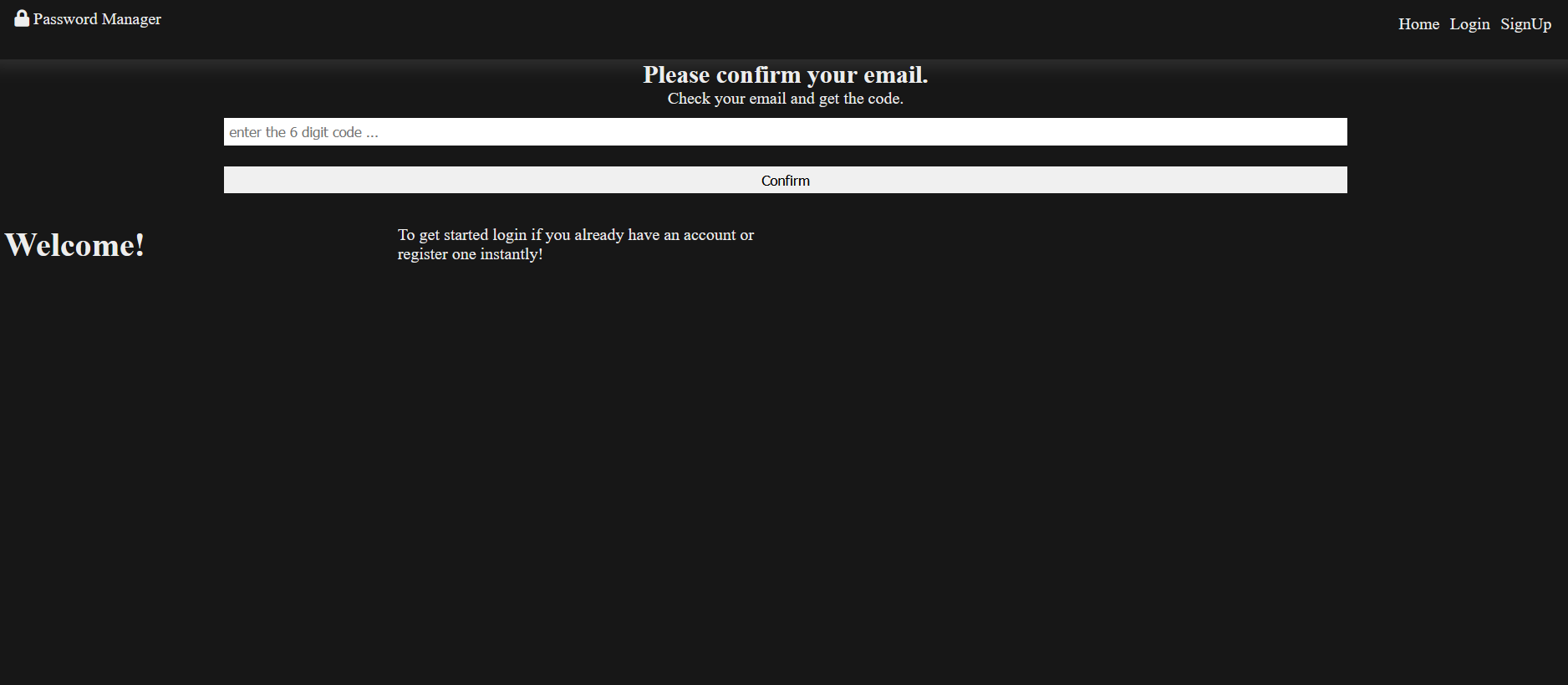
**Resetting Password:** This is the window when user click on the link that was sent to their email to reset their password



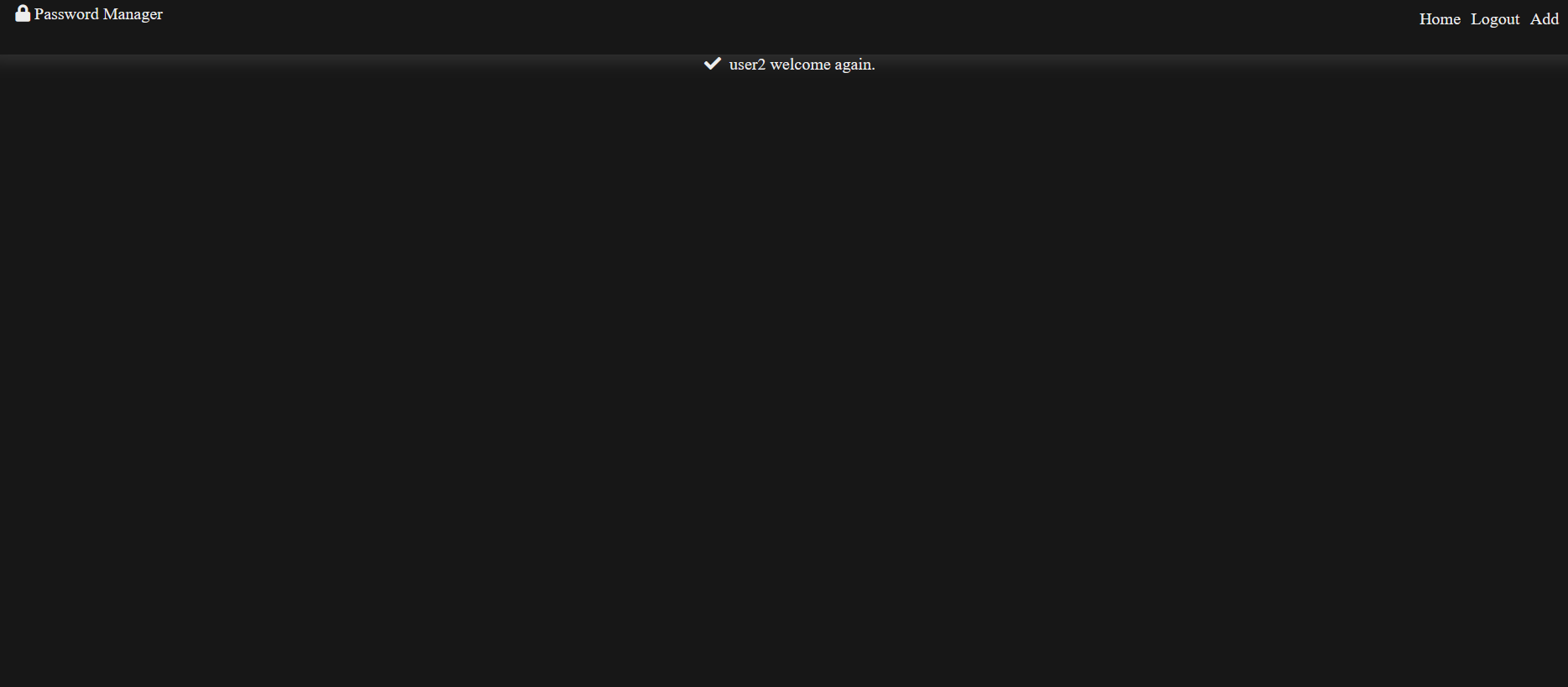
**Password reset Confirmation:** This is the final msg that is displayed for successful password reset



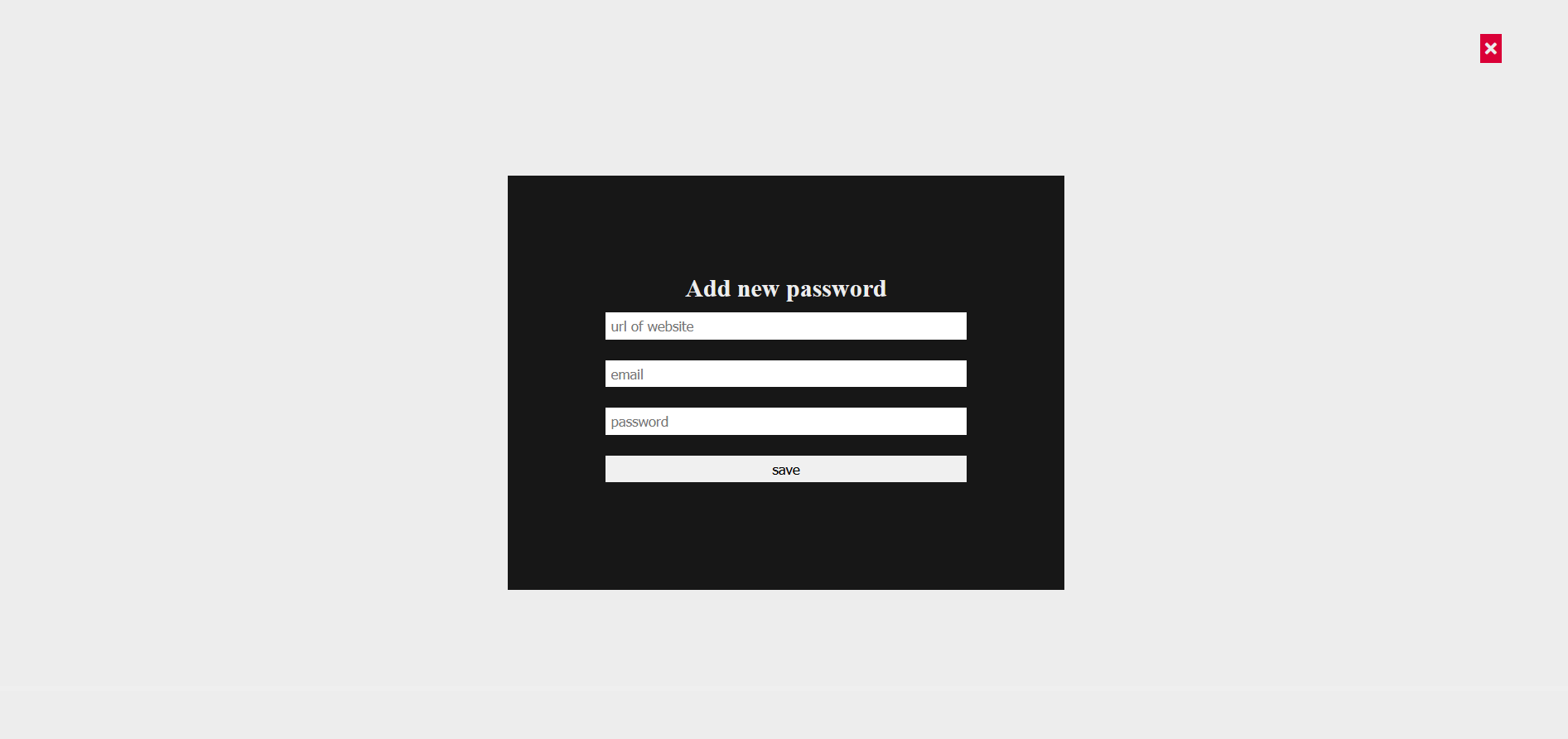
**OTP Verification:** This window is displayed to verify user, log in attempt



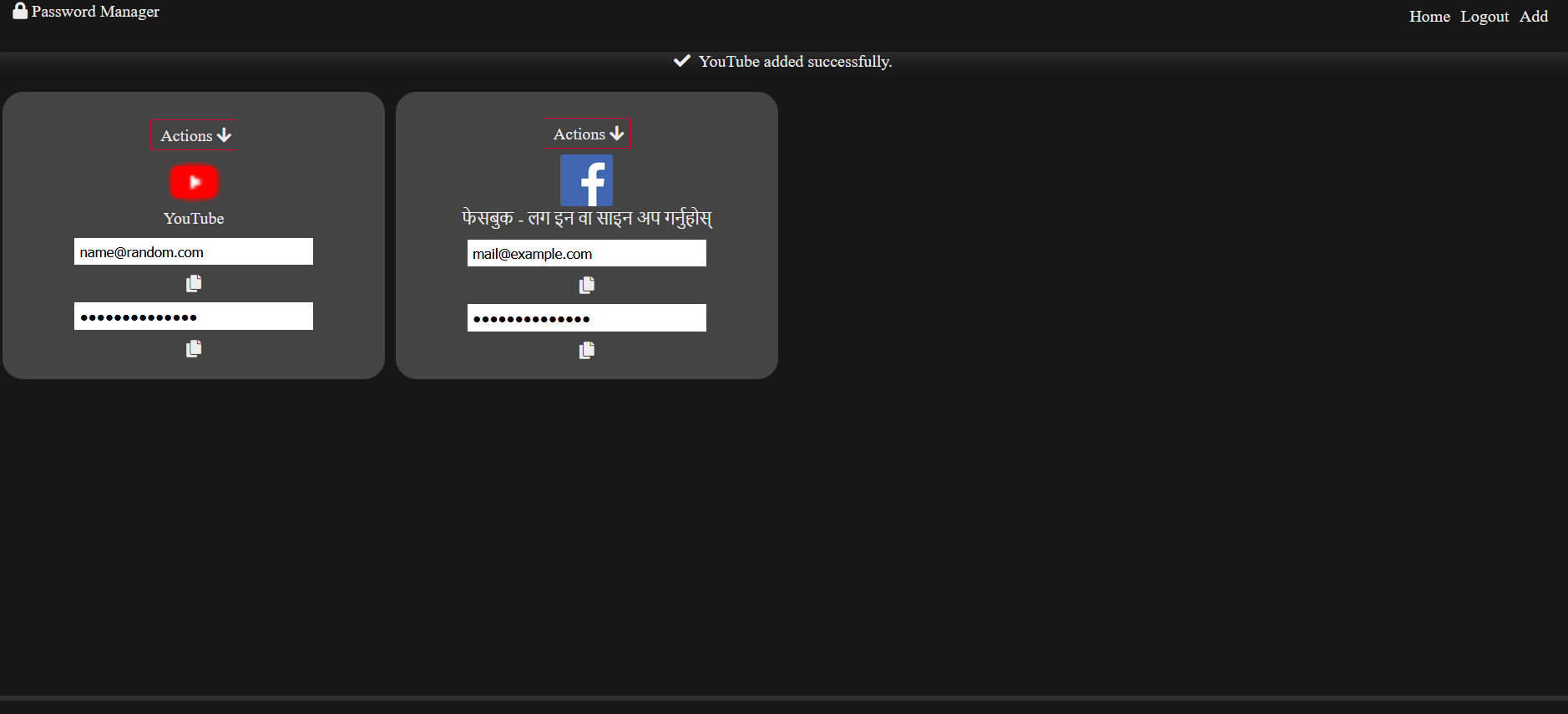
**Logged in Window:** This is the started point when user logs in for the first time

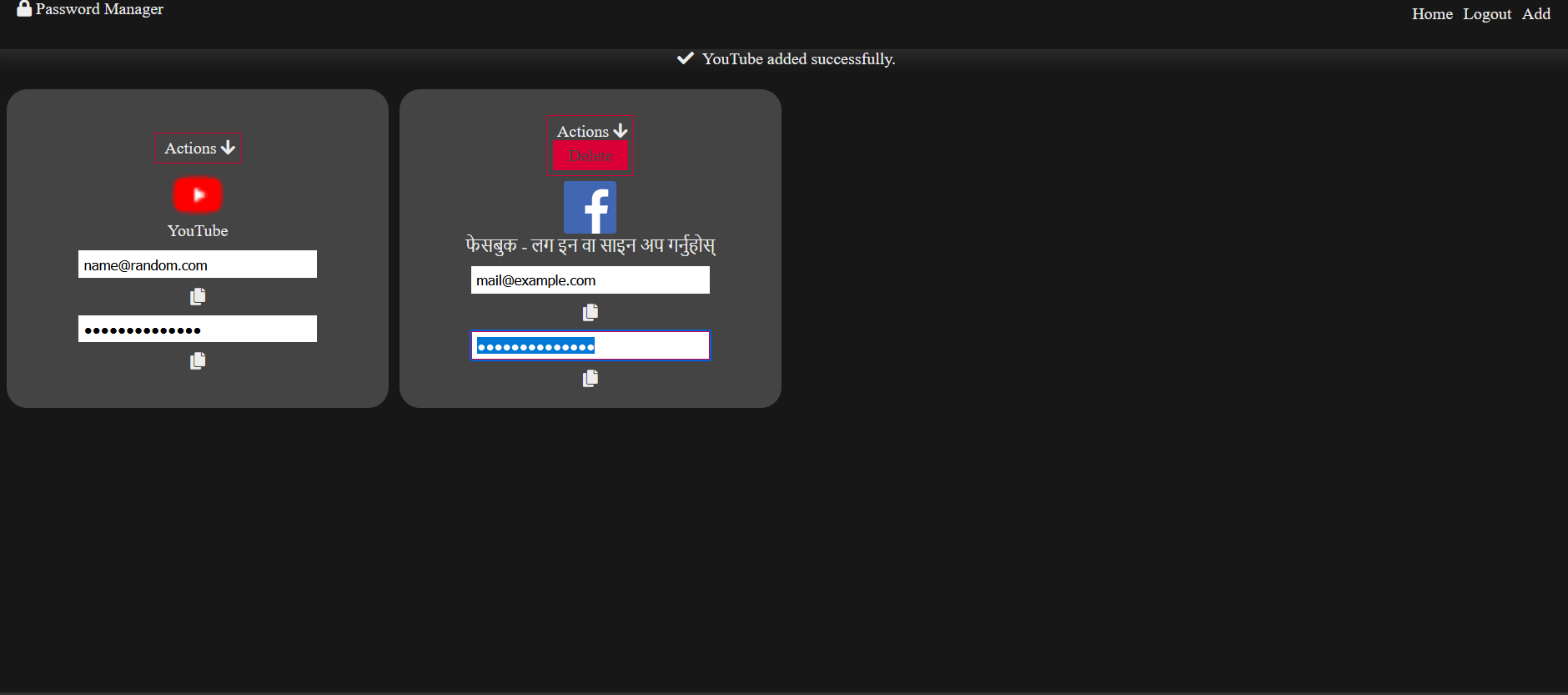


**Add Window:** This is the pop up window that is displayed when user click on the add button

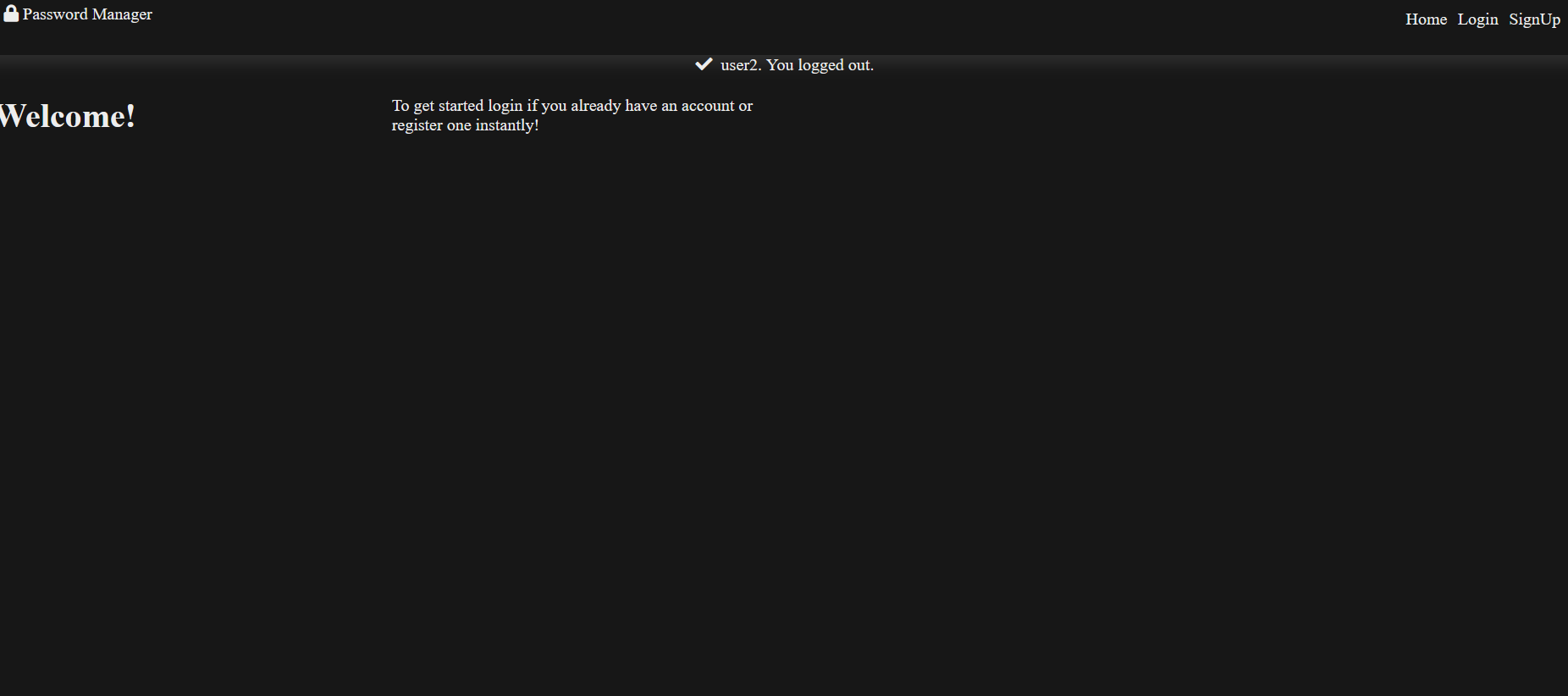


**Password List window:** This is how the hompage page looks like when a registered user adds multiple account details





**Logged out window:** This message is displayed when user successfully logs out



**4.4 Drawback And Limitations**

* User needs to have an internet connection to use the system.
* UI and UX is not the best of the design.

# Chapter *5*

## **SOURCE CODE**

**home.html**

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Django Password Manager</title>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css">

<link rel="stylesheet" href="{% static 'style.css' %}">

<script src="{% static 'main.js' %}" defer></script>

<!-- <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"> -->

<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" ></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" ></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>

</head>

<body>

<header>

<nav>

<div class="brand"><a href=""><i class="fa fa-lock" aria-hidden="true"></i>&nbsp;Password Manager</a></div>

<ul>

<li><a href="">Home</a></li>

{% if request.user.is\_anonymous %}

<li><a href="javascript:;" onclick="displayModal('login-modal');">Login</a></li>

<li><a href="{% url 'register' %}" class="ml-2">SignUp</a></li>

{% else %}

<li><a href="javascript:;" onclick="document.getElementById('logout-btn').click();">Logout</a></li>

<form hidden="true" action="." method="POST">

{% csrf\_token %}

<input type="submit" id="logout-btn" name="logout">

</form>

<!-- <li><a href="javascript:;" onclick="displayModal('password-modal');">View</a></li> -->

<li><a href="javascript:;" onclick="displayModal('add-password-modal');">Add</a></li>

{% endif %}

</ul>

</nav>

</header>

<!--Display messages-->

{% if messages %}

<div class="messages" style="text-align: center;">

{% for message in messages %}

<p>

{% if message.tags == "error" %}

<i class="fa fa-exclamation" aria-hidden="true"></i>

{% else %}

<i class="fa fa-check" aria-hidden="true"></i>

{% endif %}

&nbsp;{{message}}

</p>

{% endfor %}

</div>

{% endif %}

<!--Confirm email-->

{% if code %}

<div style="text-align: center;">

<form action="." method="POST" role="form">

{% csrf\_token %}

<h2>Please confirm your email.</h2>

<p>Check your email and get the code.</p>

<input type="text" name="code" placeholder="enter the 6 digit code ...">

<input type="text" hidden="true" name="user" value="{{user.username}}">

<input type="submit" value="Confirm" name="confirm">

</form>

</div>

{% endif %}

<!--Modals-->

<div class="modals-wrapper">

<span id="close-modal" title="close"><i class="fa fa-times" aria-hidden="true"></i></span>

<!--login modal-->

<div class="modal" id="login-modal">

<form action="." role="form" method="POST">

{% csrf\_token %}

<h2>Login</h2>

<input type="text" name="username" placeholder="username">

<input type="password" name="password" placeholder="password">

<input type="submit" value="login" name="login-form"><br/>

<p>Forgot password? <a href="{% url 'reset\_password' %}"><button type="button">Reset password</button></a></p>

</form>

</div>

<!--signup modal-->

<div class="modal" id="signup-modal">

<form action="." role="form" method="POST">

{% csrf\_token %}

<h2>SignUp</h2>

<input type="text" name="username" placeholder="username">

<input type="email" name="email" placeholder="email">

<input type="password" name="password" placeholder="password">

<input type="password" name="password2" placeholder="confirm password">

<input type="submit" value="register" name="signup-form">

</form>

</div>

<!--add password modal-->

<div class="modal" id="add-password-modal">

<form action="." role="form" method="POST">

{% csrf\_token %}

<h2>Add new password</h2>

<input type="text" name="url" placeholder="url of website">

<input type="email" name="email" placeholder="email">

<input type="password" name="password" placeholder="password">

<input type="submit" value="save" name="add-password">

</form>

</div>

</div>

<div class="container">

{% if request.user.is\_anonymous %}

<h1>Welcome!</h1>

<p>To get started login if you already have an account or register one instantly!</p>

{% endif %}

<!--One card-->

{% for password in passwords %}

<div class="card">

<div class="actions">&nbsp;Actions <i class="fa fa-arrow-down" aria-hidden="true"></i>

<a href="javascript:;" onclick="document.getElementById('delete-btn{{password.id}}').click();" class="action-item">Delete</a>

<form hidden="true" action="." method="POST">

{% csrf\_token %}

<input type="text" hidden="true" name="password-id" value="{{password.id}}">

<input type="submit" id="delete-btn{{password.id}}" name="delete">

</form>

</div>

<img width="50" height="50" src="{{password.logo}}" alt="">

<div class="name"> {{password.name}}</div>

<!-- password modal-->

<input type="email" name="email" value="{{password.email}}"><i class="fas fa-copy copy" title="copy to clipboard"></i>

<input type="password" name="username" id="pass" value="{{password.password}}"><i class="fas fa-copy copy" title="copy to clipboard"></i>

</div>

{% endfor %}

</div>

<footer>

<a href="">Link To The Project</a>

</footer>

</body>

</html>

**password\_reset.html**

<h3>Password reset</h3>

<p>Forgotten your password? Enter your email address below, and we’ll email instructions for setting a new one.</p>

<form method="post">

{% csrf\_token %}

{{form}}

<input type="Submit" name="Send email">

</form>

**password\_reset\_done.html**

<h3>Password reset complete</h3>

<p>Your password has been set. You may go ahead and log in now.</p>

<a href="/">Log in</a>

**password\_reset\_form.html**

<h3>Enter new password</h3>

<p>Please enter your new password twice so we can verify you typed it in correctly.</p>

<form method="post">

{% csrf\_token %}

{{form}}

<input type="Submit" name="Update Password">

</form>

**password\_reset\_done.html**

<h3>Password reset sent</h3>

<p>We’ve emailed you instructions for setting your password, if an account exists with the email you entered. You should receive them shortly.</p>

<p>If you don’t receive an email, please make sure you’ve entered the address you registered with, and check your spam folder.</p>

**password\_reset\_done.html**

<!DOCTYPE html>

<html>

<head>

<title>Register</title>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.6.1/css/all.css">

<style>

body,

html {

margin: 0;

padding: 0;

height: 100%;

background: #7abecc !important;

}

.user\_card {

width: 350px;

margin-top: auto;

margin-bottom: auto;

background: #74cfbf;

position: relative;

display: flex;

justify-content: center;

flex-direction: column;

padding: 10px;

box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);

-webkit-box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);

-moz-box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);

border-radius: 5px;

}

.form\_container {

margin-top: 20px;

}

#form-title{

color: #fff;

}

.login\_btn {

width: 100%;

background: #33ccff !important;

color: white !important;

}

.login\_btn:focus {

box-shadow: none !important;

outline: 0px !important;

}

.login\_container {

padding: 0 2rem;

}

.input-group-text {

background: #f7ba5b !important;

color: white !important;

border: 0 !important;

border-radius: 0.25rem 0 0 0.25rem !important;

}

.input\_user,

.input\_pass:focus {

box-shadow: none !important;

outline: 0px !important;

}

</style>

</head>

<body>

<div class="container h-100">

<div class="d-flex justify-content-center h-100">

<div class="user\_card">

<div class="d-flex justify-content-center">

<h3 id="form-title">REGISTER ACCOUNT</h3>

</div>

<div class="d-flex justify-content-center form\_container">

<form method="POST" action="">

{% csrf\_token %}

<div class="input-group mb-3">

<div class="input-group-append">

<span class="input-group-text"><i class="fas fa-user"></i></span>

</div>

{{form.username}}

</div>

<div class="input-group mb-2">

<div class="input-group-append">

<span class="input-group-text"><i class="fas fa-envelope-square"></i></span>

</div>

{{form.email}}

</div>

<div class="input-group mb-2">

<div class="input-group-append">

<span class="input-group-text"><i class="fas fa-key"></i></span>

</div>

{{form.password1}}

</div>

<div class="input-group mb-2">

<div class="input-group-append">

<span class="input-group-text"><i class="fas fa-key"></i></span>

</div>

{{form.password2}}

</div>

<div class="d-flex justify-content-center mt-3 login\_container">

<input class="btn login\_btn" type="submit" value="Register Account">

</div>

</form>

</div>

{{form.errors}}

<div class="mt-4">

<div class="d-flex justify-content-center links">

Already have an account? <a href="/" class="ml-2">Login</a>

</div>

</div>

</div>

</div>

</div>

<script>

/\* Because i didnt set placeholder values in forms.py they will be set here using vanilla Javascript

//We start indexing at one because CSRF\_token is considered and input field

\*/

//Query All input fields

var form\_fields = document.getElementsByTagName('input')

form\_fields[1].placeholder='Username..';

form\_fields[2].placeholder='Email..';

form\_fields[3].placeholder='Enter password...';

form\_fields[4].placeholder='Re-enter Password...';

for (var field in form\_fields){

form\_fields[field].className += ' form-control'

}

</script>

</body>

</html>

**style.css**

/\*############# global #######""\*/

\*{

margin: 0;

padding: 0;

box-sizing: border-box;

}

:root{

--global-background-color: #171717;

--link-background-collor: #DA0037;

--global-text-color: #EDEDED;

--text-hover-color: #444444;

--gray-color: rgba(255,255,255,0.1);

}

html{

background-color: var(--global-background-color);

color: var(--global-text-color);

}

a{

text-decoration: none;

cursor: pointer;

color: var(--global-text-color);

}

a:hover{

color: var(--text-hover-color);

}

input{

width: 70%;

margin: 10px auto;

padding: 5px;

border: none;

}

input:focus{

border: solid 1px var(--link-background-collor);

}

/\*########### header #######"\*/

header{

box-shadow: 0 10px 15px -3px var(--gray-color);

}

header nav{

display: flex;

flex-direction: row;

justify-content: space-between;

padding: 20px 30px;

}

header nav ul{

list-style-type: none;

display: inline-flex;

}

header nav ul li{

margin: 5px;

}

@media screen and (max-width: 640px) {

header nav{

flex-direction: column;

justify-content: center;

align-items: center;

}

.brand{

margin-bottom: 30px;

}

}

/\*######### footer #######"\*/

footer{

position: fixed;

bottom: 0;

left: 0;

padding: 20px;

width: 100%;

box-shadow: 0 10px 0 15px var(--gray-color);

}

/\*######### modals #######"\*/

.modals-wrapper{

width: 100%;

height: 100%;

background-color: var(--global-text-color);

display: none;

justify-content: center;

align-items: center;

position: fixed;

}

.modal{

background-color: var(--global-background-color);

min-width: 400px;

min-height: 400px;

padding: 20px;

}

#close-modal{

background-color: var(--link-background-collor);

position: absolute;

top: 5%;

right: 5%;

cursor: pointer;

padding: 5px;

}

#login-modal, #signup-modal{

display: none;

justify-content: center;

align-items: center;

text-align: center;

}

#password-modal, #add-password-modal{

display: none;

flex-direction: column;

justify-content: center;

align-items: center;

text-align: center;

}

.copy{

cursor: pointer;

}

/\*######### card #######"\*/

.container{

margin-top: 20px;

padding: 0 20px;

display: grid;

grid-template-columns: 1fr 1fr 1fr 1fr;

grid-gap: 10px;

margin-bottom: 80px;

}

.card{

background-color: var(--text-hover-color);

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

padding: 20px;

border-radius: 20px;

}

.actions{

margin: 5px;

}

.actions a{

display: none;

background-color: var(--link-background-collor);

flex-direction: column;

padding: 5px;

align-items: center;

}

.actions:hover .action-item{

display: flex;

}

@media screen and (max-width: 680px) {

.container{

grid-template-columns: 1fr 1fr;

}

}

@media screen and (max-width: 420px) {

.container{

grid-template-columns: 1fr;

}

}

/\* I added this later, didn't cover it in the tutorial\*/

.actions{

cursor: pointer;

border: solid 0.3px var(--link-background-collor);

padding: 5px;

}

/\*Change the size of the modal in the small devices\*/

@media screen and (max-width: 680px) {

.modal{

min-height: 300px;

}

.modals-wrapper{

max-height: 740px;

}

}

**main.js**

//display modal on click

const modalWrapper = document.querySelector(".modals-wrapper");

if (modalWrapper){

function displayModal(id){

const modal = document.getElementById(id);

modalWrapper.style.display = "flex";

modal.style.display = "flex";

//close modal

const close = document.getElementById("close-modal");

close.addEventListener("click", () =>{

modalWrapper.style.display = "none";

modal.style.display = "none";

//I added this later, didn't cover it on the tutorial

document.querySelector("header").style.display = "unset";

})

//I added this later, didn't cover it on the tutorial

document.querySelector("header").style.display = "none";

}

}

//copy to clipboard

const copies = document.querySelectorAll(".copy");

copies.forEach(copy =>{

copy.onclick = () =>{

let elemntToCopy = copy.previousElementSibling;

elemntToCopy.select();

document.execCommand("copy");

}

})

//I added this later, didn't cover it on the tutorial

//Display the actions of the password card for mobile devices

const actions = document.querySelectorAll(".actions");

if (actions){

actions.forEach(action =>{

action.onclick = () =>{

const links = action.querySelectorAll("a");

links.forEach(link =>{

link.style.display = "flex";

})

setTimeout(function(){

links.forEach(link =>{

link.style.display = "none";

})}

, 3000)

}

})

}

# Chapter *6*

# TEST CASE

A test case is a document, which has asset of data preconditions, expected result and post conditions, developed for a particular scenario in order to verify compliance against a specific requirement.

Test case acts as a starting point for the execution, and after applying a set of input values, the application has a definitive outcome and leave the system at some end point or also known as execution post condition.

**Typical test case parameter:**

* Test case id
* Test scenario
* Test case Description
* Test steps
* Prerequisite
* Test data
* Expected result
* Test parameters
* Actual result
* Environment Information
* Comment

**Table:6.1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case id** | **Scenario** | | **Test step** | **Expected**  **Result** | **Actual outcome** |
| **1. User Registration** | User should be able to crate a new account using their email | | Once the account is created, user should see a message saying “thanks for subscribing” | User shuld be able to sign up without any issues and be redirected to the homepage and be finally logged in | Successfully created |
| **2. Login** | User should be able to redirected to an otp verification page after confirmation of login credential and final enter the otp for successful login verification | | User should be redirected to a otp verification page after confirmation of login credentials to verify their login | User should be able to logged in using the otp that was sent to their email after verification of login credentials | Successfully logged in |
| **3. Add Accounts** | | User should be able to add their accounts along with their required details | User should add their account provided with the proper details along with the appropriate dataset | The Account info should be displayed on the dashboard on successful addition | Successfully added |
| **4. Delete Account** | | User should be able to delete their accounts | The selected account should be deleted | The name of the account should be deleted from the database and shold not be displayed on the homepage | Successfully deleted |
| **5. Password Reset** | | User should be able to reset their password if they ever forgot their username or password through a link that is to be sent after verifying their email | User should input the email of the account that was used to create their account to receive a link to reset their password | User should see a link to their email along with the username to finally reset their password | Successfully reset |

# Chapter *7*

# CONCLUSION

We strongly believe that, due to its high security standard impact, research has to look and come up with the ideas of “password management system ”,for it has a function that not only provides complete authentication, but it also provides secure ,hassle-free and avoid transparent in any fraudulent security also lot of convince than the old system

Implementation of this system is one of the most effective ways for people with multiple or with many accounts that need security and private in a way that helps save on way power, makes managing passwords proficient and decreases the amount of insecurities and risk.

After all our hard work and painstaking efforts, we bring **“**ONLINE PASSWORD MANAGEMENT SYSTEM**”** as a reality in front of all. This is the effort to develop on a commercial modeling. We have tried our best to bring out the best of our efforts.

In conclusion, we would like to say that it was an enriching and enjoyable experience that had given us an insight of System Analysis, Design and analysis of commercial software packages. We keep confident that in future, in the line of professional expertise, we shall bring credit to NIELIT IMPHAL Centre, under which we carried out this project on **“ONLINE PASSWORD MANAGEMENT SYSTEM”.**

# Chapter *8*

## **BIBLIOGRAPHY**

### 

* <https://www.youtube.com>
* <https://www.google.com>
* <https://www.github.com>
* <https://www.stackoverflow.com>