**Project Report**

**Project Name**

***PassX***: A highly secure Web Based Password Manager

**Group Members :-**

2020BCS063 Raunaq Singh Jabbal

2020BCS078 Mark Verma

2020BCS079 Saahil Sabu Hameed

1. **Introduction**

**1.1 Overview**

PassX is a web-based password manager that allows users to store, generate, manage and update their passwords or store important notes easily. PassX assists in generating and retrieving complex passwords and storing such passwords in hashed and encrypted forms in our databases so that security is never compromised. With solutions for single sign-on, password management, and multifactor authentication, PassX helps consumers and businesses increase productivity and decrease the likelihood of password-related breaches.

Platforms used for this project are:

* Based on HTML, CSS and JavaScript
* MongoDB as database (with Mongoose)
* Node.js for backend framework
* HTML, CSS and JavaScript for frontend framework

**1.2 GitHub Repository**

Link for the Repository: [GitHub](https://github.com/SaahilSabu/password-manager)  
Links to GitHub profiles of group members:

1. [Raunaq Singh Jabbal](https://github.com/raunaqjabbal)
2. [Mark Verma](https://github.com/markvrma)
3. [Saahil Sabu Hameed](https://github.com/SaahilSabu/)
4. **Details of Implemented Modules**

**2.1 Frontend Modules**

1. Index.html:

Web page that welcomes the user

1. signInsignUp:

Web page where the user either signs in or signs up

1. createPasswords:

Web page that allows user to add a domain name, username, and a password

1. dashboard:

Web page where the user can see all the stores passwords

1. updatePasswords:

Web page where the user can add notes which a header and the note contents

1. assets:

Contains information that gives our website it’s unique look

**2.2 Backend Modules**

1. models:

Connects to the database to manage and save user’s login credentials and master password

1. routes:

Connects to the database to manage and save users login credentials for other websites and the websites password

1. services:

This is the code that uses hashes and encrypts the users password

**2.3 List of All Files:**

* \api\index.js
* \api\modules\Password.js
* \api\modules\User.js
* \api\routes\password.js
* \api\routes\userRoute.js
* \api\services\enryptPassword.js
* \api\services\getFirstHash.js
* \api\services\getSecondHash.js
* \frntend\index.html
* \frntend\index.js
* \frntend\style.css
* \frntend\assets\edit-notes.svg
* \frntend\assets\homepage-illustration.svg
* \frntend\pages\createPasswords\createPasswords.html
* \frntend\pages\createPasswords\index.js
* \frntend\pages\createPasswords\style.css
* \frntend\pages\dashboard\dashboard.html
* \frntend\pages\dashboard\index.js
* \frntend\pages\dashboard\style.css
* \frntend\pages\signInsignUp\authenticate.html
* \frntend\pages\signInsignUp\index.js
* \frntend\pages\signInsignUp\style.css
* \frntend\pages\updatePasswords\updatePasswords.html
* \frntend\pages\updatePasswords\index.js
* \frntend\pages\updatePasswords\style.css

1. **Outputs Screenshots**

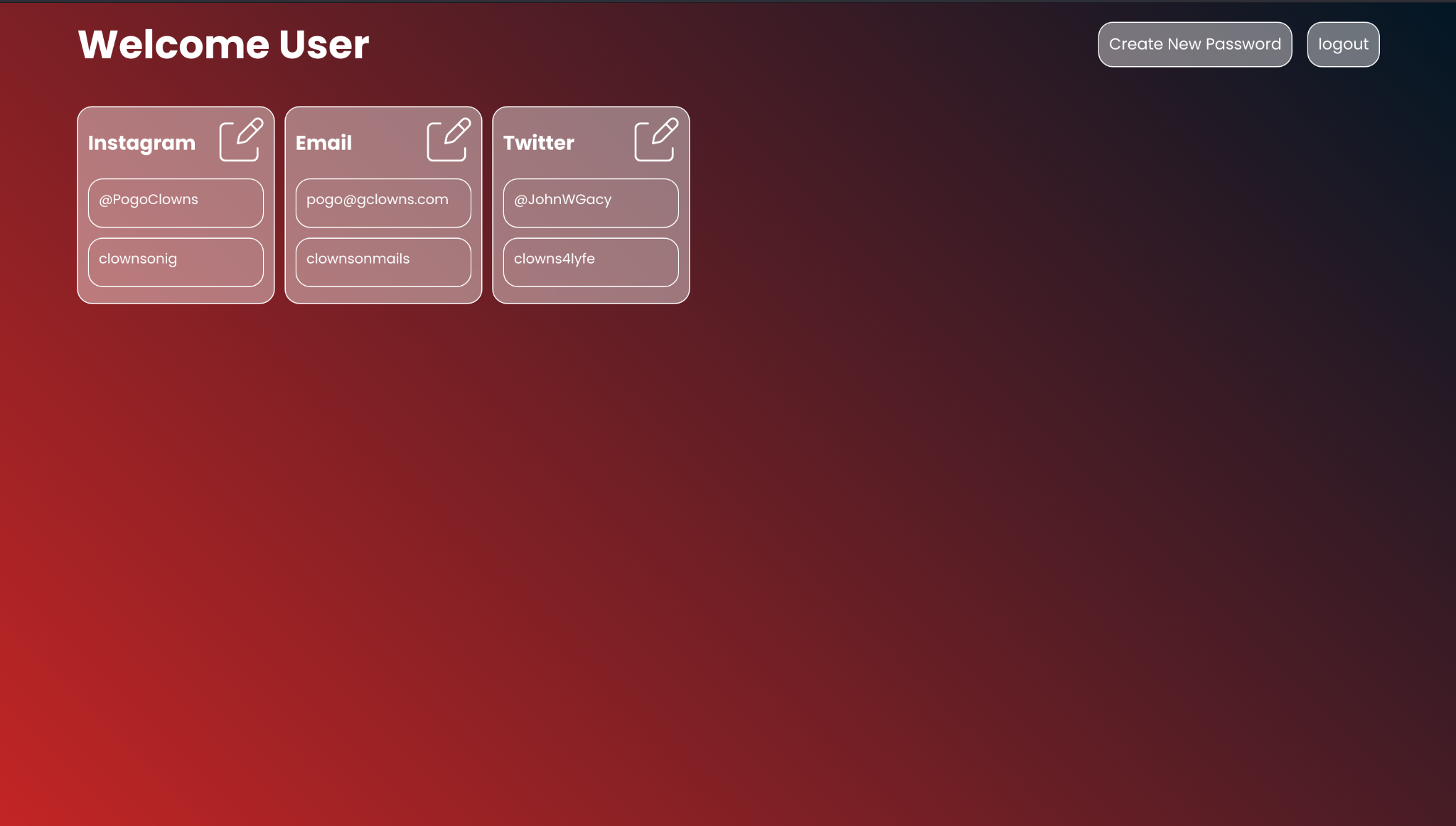
*Homepage*



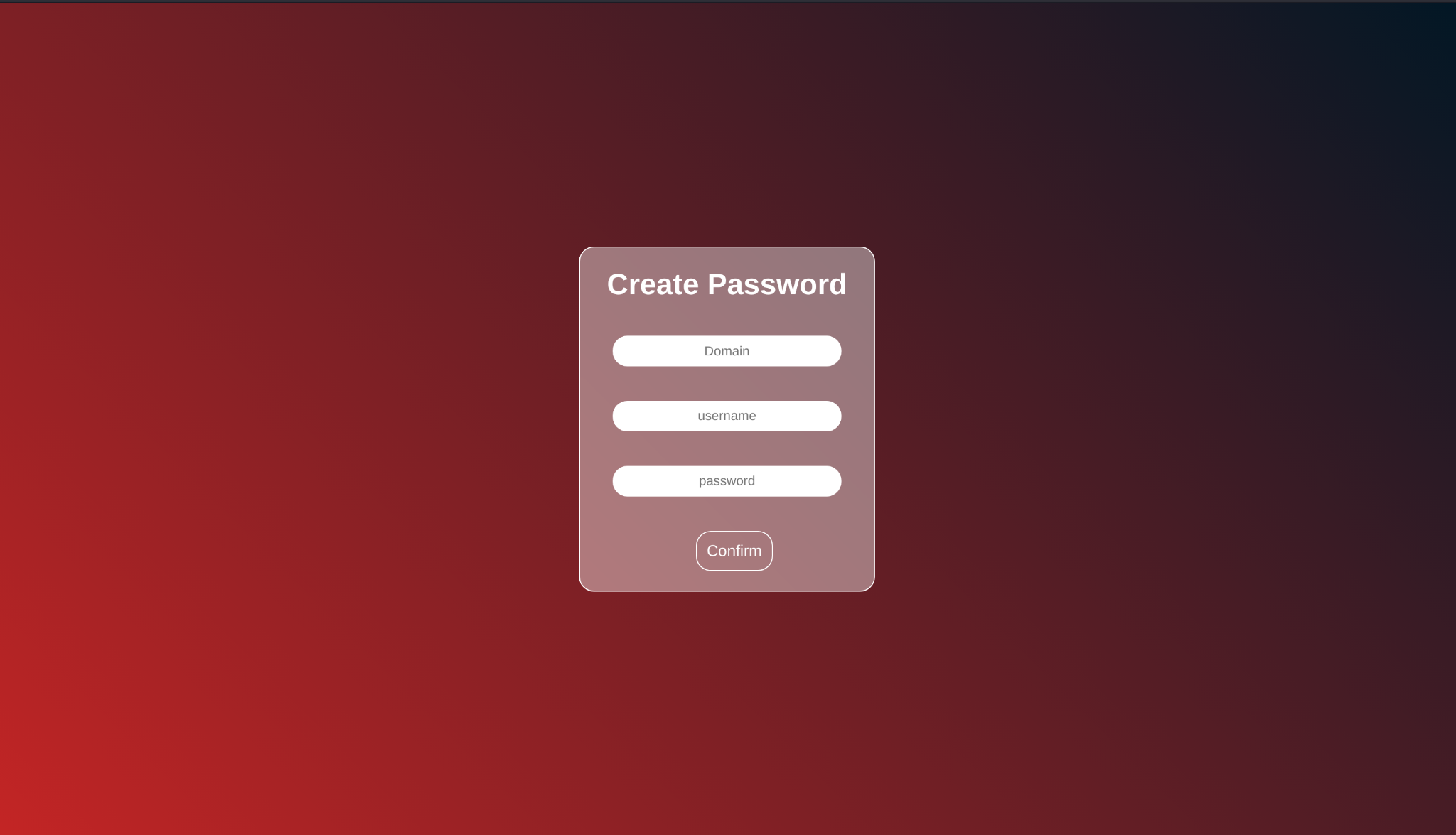
*Sign In / Sign Up*

**

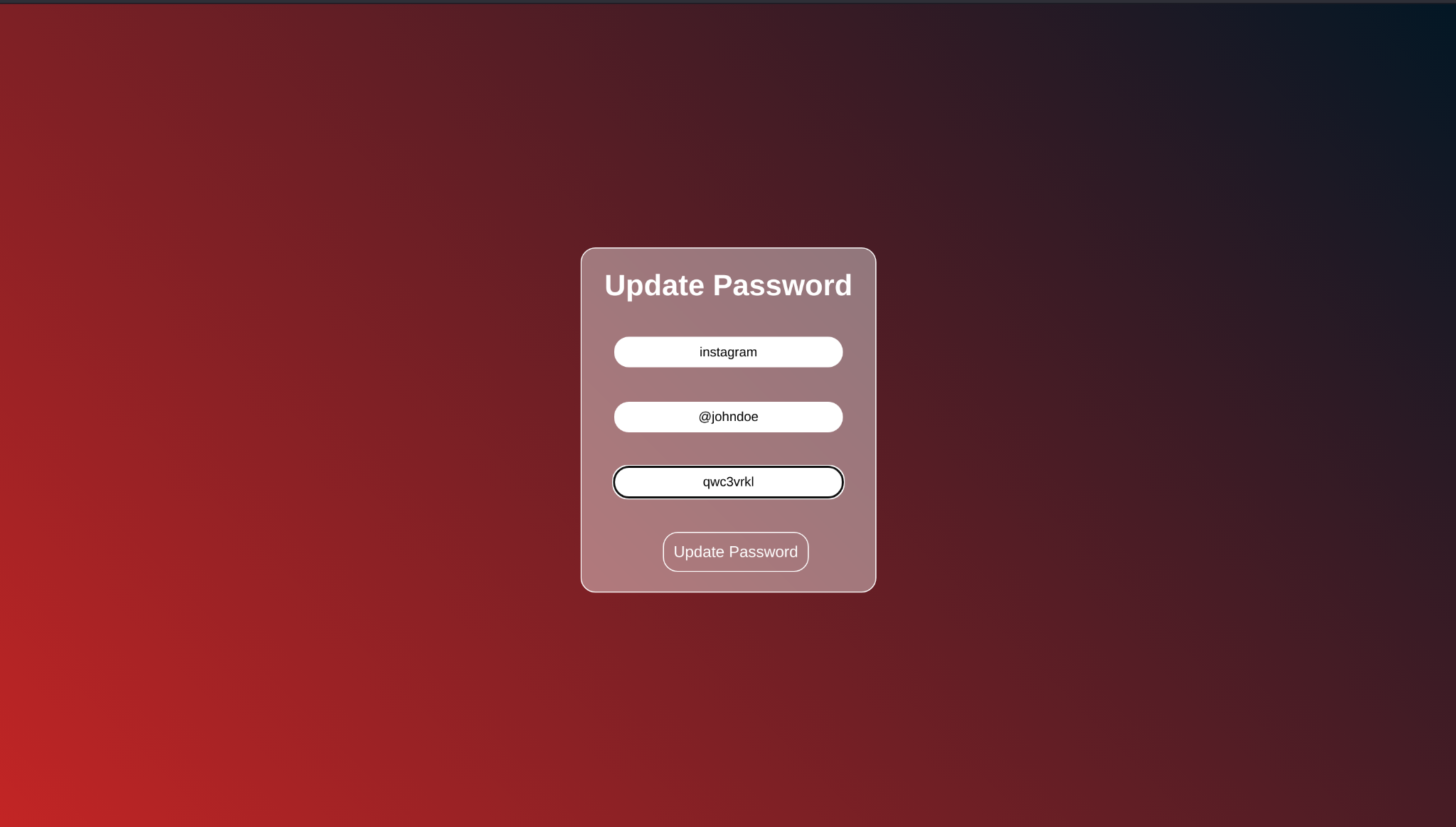
*Dashboard*

**

*Create Password*

**

*Update Password*

**