# LOGIN AUTHENTICATION PAGE

### A MINI PROJECT REPORT

#### Submitted by

#### Group/Team No: G19/T1

**Krishna Bajaj, 2210931008,**

**Aadit Bagga, 2210990001,**

**Aadit Wadhwa, 2210990002,**

**Aaditi, 2210990003**

#### in partial fulfillment for the award of the degree of

## BACHELEOR OF ENGINEERING

***in***

COMPUTER SCIENCE & ENGINEERING

****

**CHITKARA UNIVERSITY**

**CHANDIGARH-PATIALA NATIONAL HIGHWAY**

**RAJPURA (PATIALA) PUNJAB-140401 (INDIA)**

##### MAY 2023

# ABSTRACT

#### Introduction

The purpose of this project is to create a secure and user-friendly login page that allows users to authenticate themselves before accessing the website's or application's restricted content. Login authentication is a crucial aspect of web development.

#### Implementation

We have used a combination of **HTML, CSS, and JavaScript** to develop this login authentication page.

Step 1: **Creating the HTML structure**. The HTML code includes a form with three input fields for the username, email and password and a submit button. The form is enclosed within a div container.

Step 2: **Styling the page with CSS.** The CSS code is used to style the login form, including the input fields, submit button, and div container.

Step 3: **Implementing the JavaScript code**. The JavaScript code is used to handle the login functionality. When the user submits the form, the JavaScript code retrieves the username, email and password values and sends them to the server for verification.

#### Layout

#### 

# TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **Sr.no** | **Section** | **Page no** |
|  | Introduction | 4 |
|  | Problem Statement | 4 |
|  | Technical Details | 5 |
|  | Key Features | 5 |
|  | Advantages | 6 |
|  | Result | 7 |
|  | Conclusion | 10 |
|  | Future Scope | 11 |

##### **INTRODUCTION**

Login authentication is a critical element in web development as it ensures that only authorized users can access sensitive information or perform specific actions on a website or application. In this project, we have utilized **HTML, CSS, and JavaScript** to build a login authentication page that is both secure and user-friendly.

HTML, the markup language used for structuring web pages, is employed to create the basic structure of the login page. It allows us to define the necessary elements such as the login form, input fields for username and password, buttons, and any additional components required.

CSS, the styling language, is utilized to enhance the visual appeal of the login page. Through CSS, we can customize the appearance of the elements, apply colors, fonts, and layouts, making the login page visually pleasing and aligned with the overall design of the website or application.

JavaScript, the programming language for web development, is utilized to implement the functionality of the login authentication system. With JavaScript, we can handle user interactions, validate input data, and perform necessary actions like verifying the entered credentials against a database, initiating a session, and redirecting users to the appropriate pages based on their authentication status.

The login authentication page includes several important components to ensure a comprehensive system. Firstly, the login form allows users to enter their credentials securely. Secondly, user registration functionality enables new users to create an account by providing essential information and storing it securely. Additionally, error handling mechanisms are implemented to provide appropriate feedback to users in case of invalid login attempts or other errors.

The purpose of this project is to create a secure and user-friendly login page that allows users to authenticate themselves before accessing the website's or application's restricted content.

# PROBLEM STATEMENT

The primary objective of creating a login webpage is to guarantee the **security and privacy** of users within a website or application. By implementing a login page, the identity of users can be verified, ensuring that only authorized individuals gain access to restricted content or features. This is of utmost importance as it prevents unauthorized users from potentially exploiting sensitive information or performing actions that are intended exclusively for authorized users. Neglecting to include a login page can leave a website or application vulnerable to security breaches, data leaks, and other detrimental consequences.

The login page serves as a vital component for any website or application that necessitates user authentication. It plays a significant role in safeguarding users' personal information and upholding the overall integrity of the website or application. By requiring users to authenticate themselves through a login process, the system can validate their credentials and grant access to appropriate resources or functionalities based on their authorized privileges.

Moreover, the login page also contributes to creating a personalized user experience. By distinguishing between authenticated and unauthenticated users, the website or application can tailor the content, features, and settings to suit the specific needs and preferences of each individual. This enhances user satisfaction and helps establish a sense of trust and confidence in the platform.

In summary, the creation of a login webpage is driven by the imperative need to ensure the security and privacy of users within a website or application. By incorporating a login page, the system can verify the identity of users, protect sensitive information, and regulate access to restricted content or features. This not only prevents unauthorized individuals from compromising the system but also contributes to a more personalized and secure user experience overall.

# TECHNICAL DETAILS

The login authentication page is implemented using the following steps:

**Step 1:** **Creating the HTML structure**. The HTML code includes a form with three input fields for the username, email and password and a submit button. The form is enclosed within a div container.

**Step 2:** **Styling the page with CSS.** The CSS code is used to style the login form, including the input fields, submit button, and div container.

**Step 3:** **Implementing the JavaScript code**. The JavaScript code is used to handle the login functionality. When the user submits the form, the JavaScript code retrieves the username, email and password values and sends them to the server for verification. If the login credentials are correct, the user is redirected to the website's or application's main page. Otherwise, an error message is displayed.

# KEY FEATURES

1. **User authentication**: The primary feature of a login authentication page is to authenticate the user before granting access to the website or application. This is done by requiring the user to enter a unique username and password or using other authentication methods such as biometrics or two-factor authentication.
2. **Security**: The login authentication page ensures the security and privacy of a website or application's users. It provides protection against unauthorized access to sensitive information, data leaks, and other harmful consequences.
3. **Personalization**: The login authentication page can also help to personalize the user experience by allowing users to save their preferences and track their activity on the website or application. This can improve user engagement and satisfaction, leading to a more successful and profitable product.
4. **Error handling**: The login authentication page should provide clear error messages if the user enters an incorrect username or password. This helps the user to identify and correct any errors and improves the overall user experience.
5. **Accessibility**: The login authentication page should be designed to be accessible to users with disabilities, ensuring that all users can access the website or application regardless of their abilities.

# ADVANTAGES

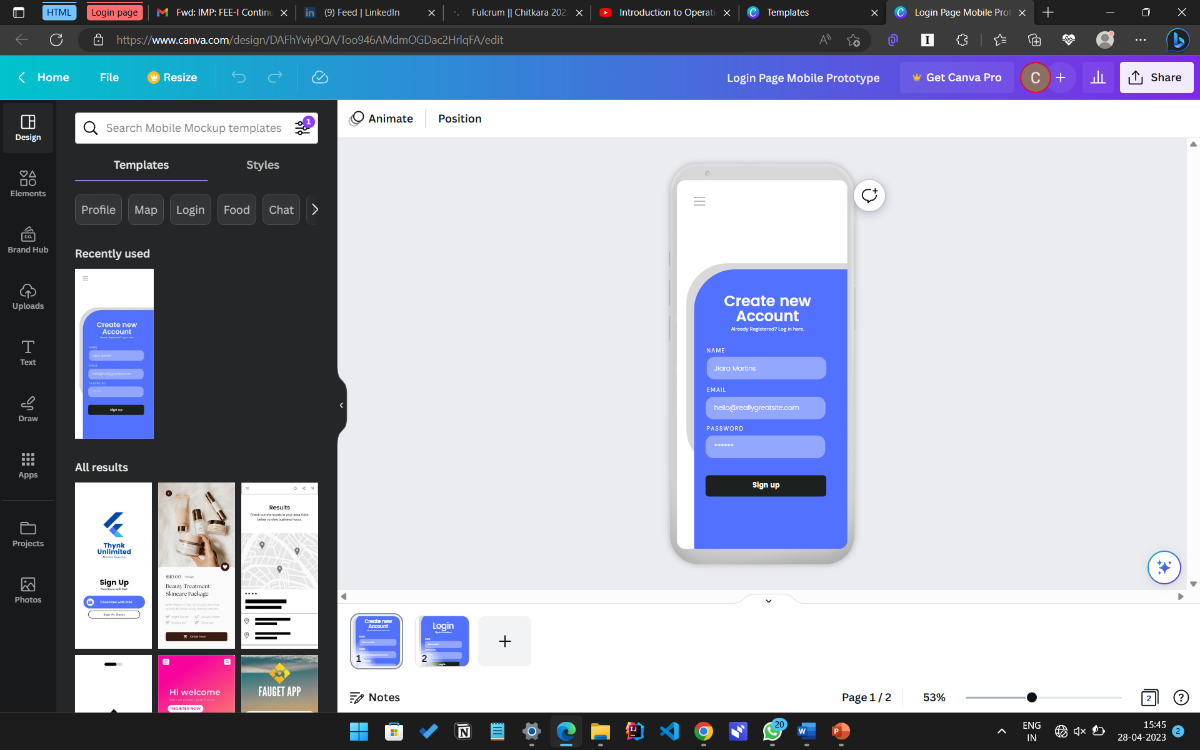
Here are some advantages of a login authentication page:

1. Enhanced security: The login authentication page ensures the security of a website or application by verifying the user's identity before granting access to sensitive information or features. This helps to prevent unauthorized access and protect against security breaches and data leaks.
2. User privacy: The login authentication page protects the user's privacy by ensuring that only authorized users can access the website or application's restricted content. This helps to maintain the integrity of the user's personal information and prevents it from being accessed by unauthorized third parties.
3. Personalization: The login authentication page can also help to personalize the user experience by allowing users to save their preferences and track their activity on the website or application. This can improve user engagement and satisfaction, leading to a more successful and profitable product.
4. Easy access: The login authentication page provides a simple and efficient way for users to access the website or application's content or features that require authentication. This saves time and effort for the user and makes the overall user experience more convenient.
5. Error handling: The login authentication page provides clear error messages if the user enters an incorrect username or password. This helps the user to identify and correct any errors and improves the overall user experience.
6. User feedback: The login authentication page provides feedback to the user, such as displaying loading indicators, success messages, or error messages, to improve the user experience.
7. Accessibility: The login authentication page is designed to be accessible to users with disabilities, ensuring that all users can access the website or application regardless of their abilities.

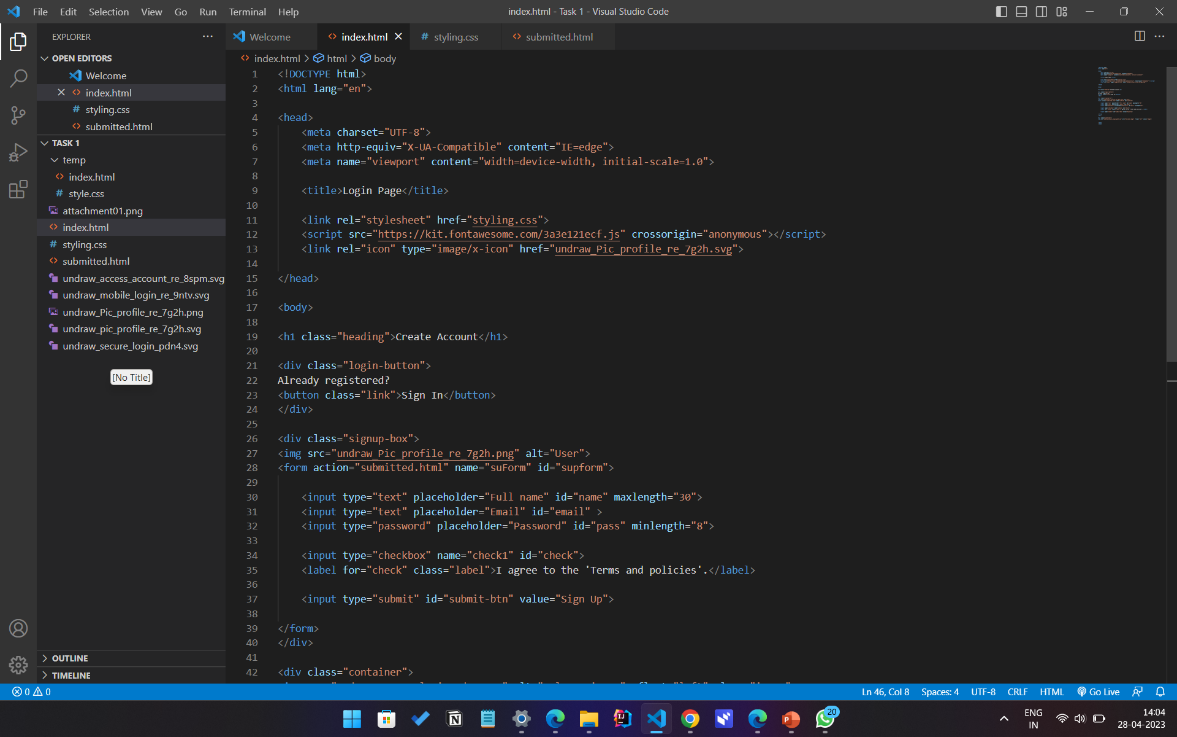
# RESULT

Sure, here are the step-by-step instructions followed for building this login authentication page using HTML, CSS, and JavaScript:

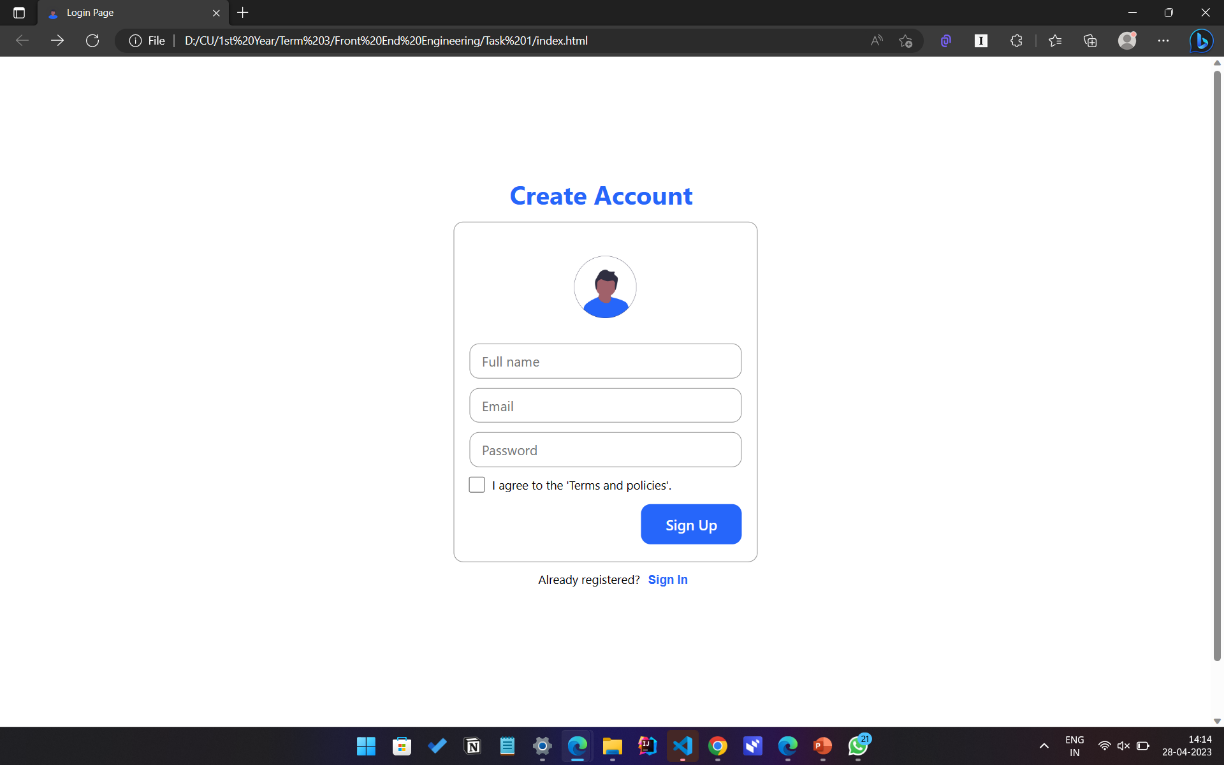
**Step 1:** **Plan and design the user interface**: The first step is to plan and design the user interface of the login authentication page. Tools like Canva, Sketch, or Figma to create wireframes and design mock-ups of the page.



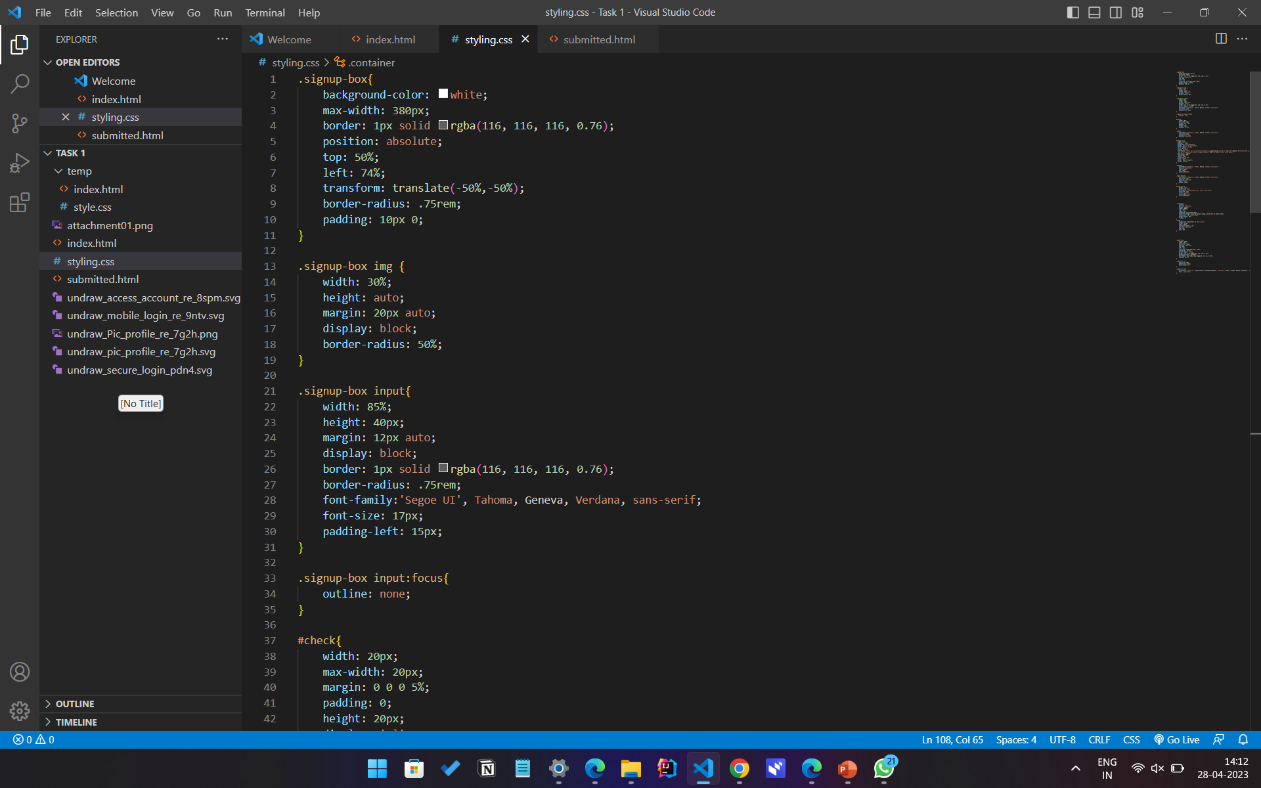
**Step 2:** **Create a new HTML**: file Once you have the design, create a new HTML file and add the necessary elements, such as the head and body tags. Add the required metadata and links to CSS and JavaScript files.



**Step 3:** **Add the login form**: The next step is to add the login form to the HTML file. This includes input fields for the username, email and password and a submit button. HTML input elements are used and set the type attribute to "text" and "password" for the username and password fields, respectively.



**Step 4:** **Style the login form with CSS**: After adding the login form, use CSS to style the form and make it look visually appealing. Use CSS properties like background colour, font size, and padding to style the form and the input fields.



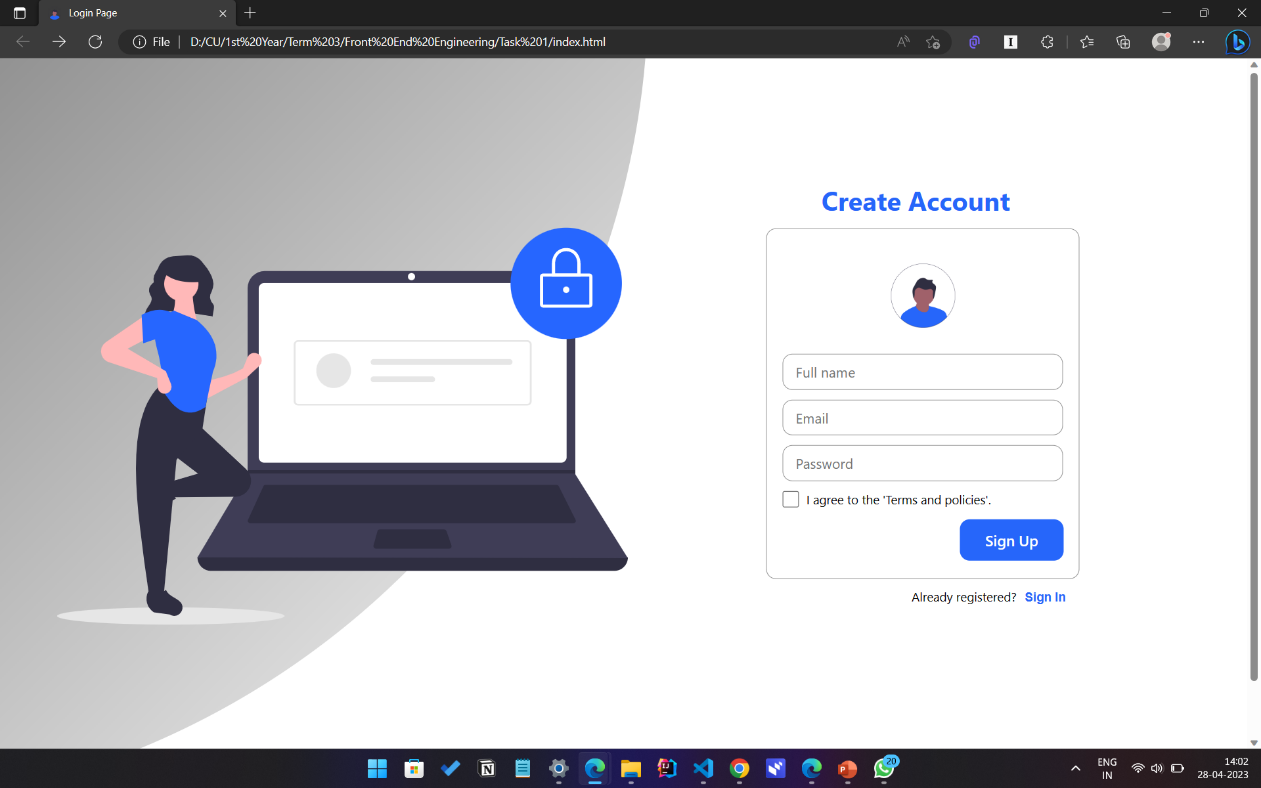
**Step 5:** **Write JavaScript code for form validation**: The next step is to write JavaScript code to validate the user's input in the login form. Use functions to check whether the username and password fields are empty or not and display error messages if they are.

**Step 6:** **Implement authentication:** Once the user's input is validated, you need to implement authentication. Use a server-side script or a third-party authentication service to verify the user's credentials. If the user's credentials are correct, redirect them to the authenticated page. If they are incorrect, display an error message.

# CONCLUSION

In conclusion, the login authentication page built using HTML, CSS, and JavaScript is an essential component of any website or application that requires user authentication.

This project has demonstrated how to create a secure and user-friendly login page using these technologies, providing a valuable resource for web developers and designers.



# FUTURE SCOPE

Future work on this project could involve adding additional security measures such as two-factor authentication, implementing password reset functionality, and improving the overall user experience. Additionally, the login authentication page could be integrated into a larger web application or website for a more comprehensive user experience.

# REFERENCES

* Canva.com
* Geeksforgeeks.com
* W3schools.com
* Youtube.com