**Specification Document**

# Document Details

Document Information

|  |  |
| --- | --- |
| Category | Information |
| Document | Specification Document |
| Version | 1.0 |
| Author | Abdullah Arshad |
| Reviewer | Yasir Latif |
| Approver | Yasir Latif |
| Issue Date | 25/09/2024 |
| Document Category | Internal ☐ External ☒ |
| Privacy Level | Public ☐ Company Confidential ☒ Customer Confidential ☒ |
| Disclaimer | This document contains specifications of the Blog Application. |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Author | Date | Version | Description |
| Abdullah Arshad | 25/09/2024 | 1.0 | * Document Created |
|  |  |  |  |

# Contents

[Document Details 2](#_Toc178159654)

[Contents 3](#_Toc178159655)

[List of Figures 3](#_Toc178159656)

[1 Introduction 5](#_Toc178159657)

[1.1 Purpose 5](#_Toc178159658)

[1.2 Definitions, Acronyms and Abbreviations 5](#_Toc178159659)

[1.3 Personas 6](#_Toc178159660)

[1.4 Platforms 7](#_Toc178159661)

[2 Specifications 8](#_Toc178159662)

[2.1 Description 8](#_Toc178159663)

[2.2 Functional Requirements 8](#_Toc178159664)

[Login/Sign-Up Page: 8](#_Toc178159665)

[Dashboard/Home Page: 8](#_Toc178159666)

[Explore Page: 8](#_Toc178159667)

[Blog Management: 8](#_Toc178159668)

[Profile Management: 9](#_Toc178159669)

[Search and Filtering: 9](#_Toc178159670)

[Menu Bar and Navigation: 9](#_Toc178159671)

[Toasts and Modals: 9](#_Toc178159672)

[2.3 Non-Functional Requirements 9](#_Toc178159673)

[2.4 Variables 11](#_Toc178159674)

[3 GUI 12](#_Toc178159675)

[3.1 GUI of Different Pages 12](#_Toc178159676)

[4 CONCLUSIONS 19](#_Toc178159677)

# List of Figures

[Figure 1: Sign In Page 12](#_Toc178159678)

[Figure 2: Sign Up page 12](#_Toc178159679)

[Figure 3: Request Password Reset Page 13](#_Toc178159680)

[Figure 4: Home/Following Page 13](#_Toc178159681)

[Figure 5: Explore Page 14](#_Toc178159682)

[Figure 6: View Blog Page 14](#_Toc178159683)

[Figure 7: Add Blog Page 15](#_Toc178159684)

[Figure 8: Edit Blog Page 15](#_Toc178159685)

[Figure 9: Profile View Page 16](#_Toc178159686)

[Figure 10: Update Profile Page 16](#_Toc178159687)

[Figure 11: Search Page 17](#_Toc178159688)

[Figure 12: List of Followers 17](#_Toc178159689)

[Figure 13: List of Following 18](#_Toc178159690)

[Figure 14: Other User's Profile Page 18](#_Toc178159691)

[Figure 15: Toasts used to display alerts 19](#_Toc178159692)

[Figure 16: Modals for confirmation 19](#_Toc178159693)

# Introduction

The Blog Application is a versatile platform designed for creating and managing blog posts, built using Flask for the backend, SQLite for database management, and Tailwind CSS for a modern and responsive user interface. Users can sign up, log in, and securely reset their passwords. The Home Page displays posts from the user and their followed accounts, while the Explore Page showcases blogs from all users. Each post has detailed views where users can edit or delete their own content or comment on others' posts, with notifications sent to the blog author.

The application features a comprehensive Profile Page for managing user information and viewing follower statistics. A search bar enables finding blogs by keywords, and pagination ensures smooth navigation across the platform. With its robust technology stack and intuitive design, this Blog Application offers an engaging and efficient blogging experience for users.

## Purpose

The purpose of this Blog Application is to offer users a comprehensive platform for sharing their thoughts, ideas, and stories through blog posts. It facilitates easy content creation and management with features like rich text editing, image uploads, and user-friendly navigation. By enabling interactions such as commenting and following, it fosters a sense of community among users. The application also ensures secure user authentication and data handling, providing a reliable environment for both bloggers and readers to engage with high-quality content.

## Definitions, Acronyms and Abbreviations

* **Flask**: A lightweight web framework for Python that allows developers to build web applications quickly and easily. It is designed to be simple and flexible, making it suitable for both small and large applications.
* **SQLite**: A self-contained, serverless, zero-configuration database engine. It is a relational database management system (RDBMS) known for its portability, performance, and simplicity.
* **Tailwind CSS**: A utility-first CSS framework that provides low-level, reusable styling options. It allows developers to create custom designs quickly without leaving the HTML file.
* **JWT**: JSON Web Token, a compact, URL-safe method for representing claims securely between two parties. It is commonly used for authentication and data exchange.
* **WTForms**: A flexible forms validation and rendering library for Python web development. It is used to create and validate form inputs in Flask applications.
* **CRUD**: Create, Read, Update, Delete – the four basic operations performed on database records.
* **GUI**: Graphical User Interface, a visual interface that allows users to interact with a software application through graphical elements like buttons and icons.
* **API**: Application Programming Interface, a set of rules and protocols for building and interacting with software applications. It allows different software systems to communicate with each other.
* **HTML**: Hypertext Markup Language, the standard markup language used for creating web pages and web applications.
* **JavaScript**: A high-level, dynamic programming language used to create interactive effects within web browsers.
* **CSS**: Cascading Style Sheets, a stylesheet language used for describing the presentation of a document written in HTML or XML.
* **JWT**: JSON Web Token, a compact and URL-safe means of representing claims to be transferred between two parties. Often used for authorization and information exchange.
* **REST**: Representational State Transfer, an architectural style for designing networked applications, typically using HTTP requests for communication.
* **UI**: User Interface, the space where interactions between humans and machines occur, often involving elements like buttons, icons, and other visual components.

## Personas

***General User:***

* **Role**: A general user has full access to their own content and can interact with the community through blogging and commenting.
* **Capabilities**:
  1. **Create & Manage Blogs**: Can write, edit, and delete their own blog posts.
  2. **View Content**: Can browse the home page for posts by followed users and the Explore page for posts from all users.
  3. **Comment**: Can leave comments on other users' blog posts.
  4. **Profile Management**: Can update their profile information, view their own posts, and follow or unfollow other users.
  5. **Interaction**: Can follow users to see their posts on the home page and engage with the community through comments and likes.
  6. **Search & Navigation**: Can use the search bar to find specific blogs by keywords and navigate easily with pagination.

## Platforms

1. ***Web Browsers:***

* Accessible on popular web browsers like Chrome, Firefox, Safari, and Edge.
* Compatible with both desktop and mobile versions, ensuring a responsive and user-friendly experience across devices.

1. ***Desktops/Laptops:***

* Works seamlessly on Windows, macOS, and Linux systems, as long as a modern web browser is available.

# 2 Specifications

## 2.1 Description

**Key features include:**

* User Authentication (Login/Sign-Up/Password Reset)
* Home Page (User and Followed Blogs)
* Explore Page (All Blogs)
* Post Creation and Management
* Commenting System
* Profile Management
* Follow/Unfollow Users
* Search Functionality
* Pagination on Blog Pages
* Navbar with User Profile and Options
* Toasts for Notifications
* Modals for Confirmations

## 2.2 Functional Requirements

### ****Login/Sign-Up Page****:

* **Users must enter a valid username and password** to access their account.
* Users can reset their password via a **JWT-based email link**.

### ****Dashboard/Home Page****:

* The home page displays **blogs posted by the user and the users they follow.**
* Users can **click on a blog to view details**.
* The home page shows the **date on which each blog was posted**.

### ****Explore Page****:

* Users can **view all blogs** from all users.
* Blogs are listed with their title, author, and upload date.
* Users can **click on any blog to view details**.

### ****Blog Management****:

* ***Post Creation****:*
  + Users can **create new blogs** by providing a title, description (using a rich text editor), and an image (either uploaded or linked).
  + The **date and time of the blog** are shown.
* ***Post Editing and Deleting****:*
  + Users can **edit and delete their own blogs**.
  + Deleting a blog requires **confirmation** to prevent accidental deletion.
* ***Commenting****:*
  + Users can **comment on their own and other user’s blog**.
  + The blog author will receive an **email notification** when a comment is posted.

### ****Profile Management****:

* ***Profile Viewing****:*
  + Users can view their own profile and others' profiles, which display their avatar, username, and last seen status.
  + Users can view other users' posts and follow/unfollow them.
* ***Profile Editing****:*
  + Users can **edit or delete their own profile** details.
* ***Follow/Unfollow****:*
  + Users can **follow or unfollow other users**.

### ****Search and Filtering****:

* Users can **search for blogs** using keywords in titles or descriptions.
* **Pagination** is applied to blog listing pages (Home, Explore, Search) to show a set number of posts per page.

### ****Menu Bar and Navigation****:

* **Navbar** displays the current user’s avatar and username.
* **Dropdown Menu** includes options to:
  + View the user’s profile
  + Update profile
  + Log out

### ****Toasts and Modals****:

* **Toasts display success, error, or info messages** for actions like posting a comment, deleting a post, or logging out.
* **Modals appear for confirmation** of critical actions such as deleting posts, accounts, or logging out.

## 2.3 Non-Functional Requirements

 ***Performance:***

* The application loads blog posts and profiles within 2 seconds on average.
* Pagination ensures each page loads smoothly without significant delays.

*** Security:***

* All passwords are securely stored and encrypted.
* JWT-based password reset ensures secure access recovery.
* Protection against common vulnerabilities (e.g., SQL injection, cross-site scripting).

 ***Usability:***

* The user interface is simple and intuitive for users of all technical skill levels.
* Mobile responsiveness is ensured using Tailwind CSS for a consistent experience across devices.

 ***Availability****:*

* The application is available at all times, ensuring minimal downtime.
* It supports concurrent users without significant slowdowns or failures.

 ***Maintainability****:*

* The codebase is well-documented and structured to allow easy updates and fixes.
* Components are modular to simplify future changes and scaling.

 ***Reliability:***

* The application functions correctly under all conditions, ensuring data integrity during operations like post creation, edits, and deletions.

 ***Compatibility:***

* The application works seamlessly on all modern web browsers (Chrome, Firefox, Safari, etc.).
* It is compatible with various screen sizes and devices (desktop, tablet, mobile).

 ***Responsiveness****:*

* The user interface provides real-time feedback (using toasts or loading indicators) to keep users informed about the system's status.

## 2.4 Variables

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Control Name** | **Control Type** | **Format** | **Required Fields** | **Boundary Limits** | **Action** | **Response** |
| Username | Text Field | Alphanumeric with underscore and hyphens | <Mandatory> | NA | NA | NA |
| Password | Text Field | Alphanumeric with specific special characters | <Mandatory> | >=8 char | NA | (1) (If less than 8 characters), error is displayed |
| Email | text | alphanumeric | <Mandatory> | NA | Email validation | (1) (if field is empty) error message: Please fill all the fields. |
| About me | Text field | Alphanumeric | <Optional> | <=180 char | NA |  |
| Confirm password | Text field | Alphanumeric with specific special characters | <Mandatory> | >=8 char | Should be equal to password field | (1) (if field is empty) error message: Please fill all the fields.  (2) (if field isn’t equal to confirm password) error message: The passwords don’t match |
| Title | Text field | Alphanumeric | <Mandatory> | <=100 char | NA | (1) (if field is empty) error message: Please fill all the fields. |
| Description | Rich text editor | Alphanumeric | <Mandatory> | NA | NA | (1) (if field is empty) error message: Please fill  all the fields. |
| Image URL | string | Alphanumeric | <Optional> |  | NA | NA |
| Image File | File-field | ‘jpg’, ‘png’ | <Optional> |  | NA | NA |

# 3 GUI

The GUI of the application emphasizes simplicity and usability. It features a clean layout with treeview to display items in inventory and the GUI is built using Tkinter library.

## 3.1 GUI of Different Pages

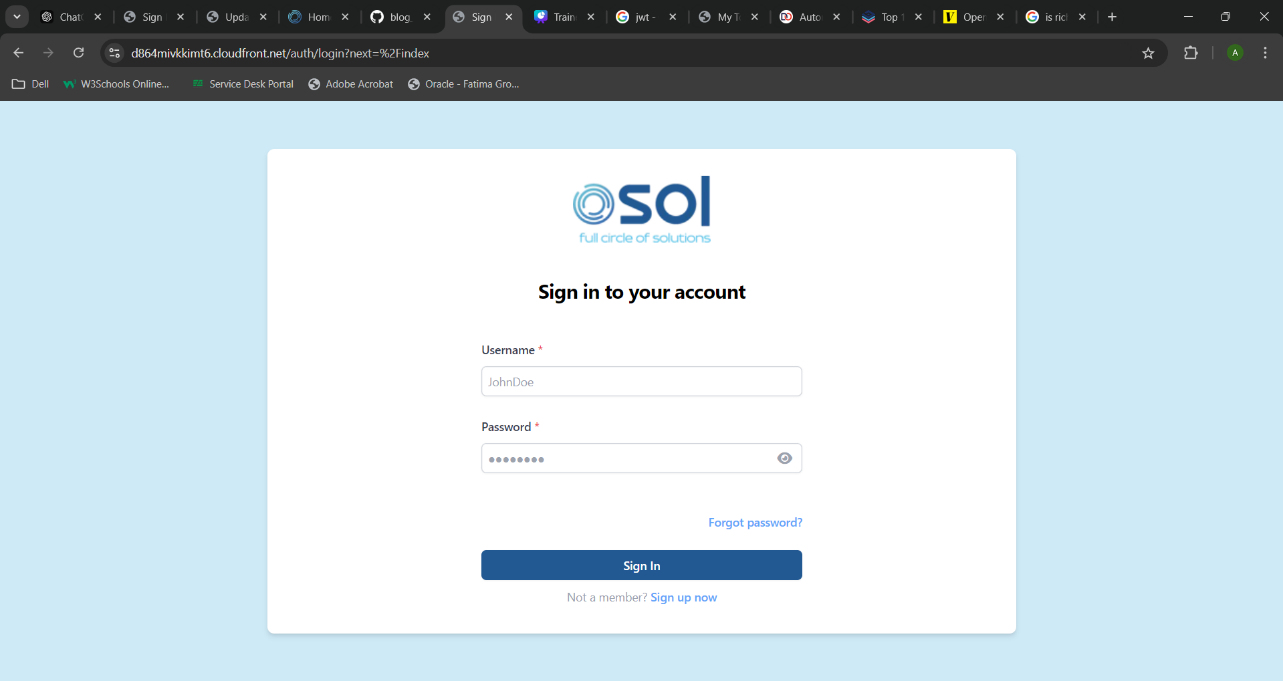


Figure 1: Sign In Page

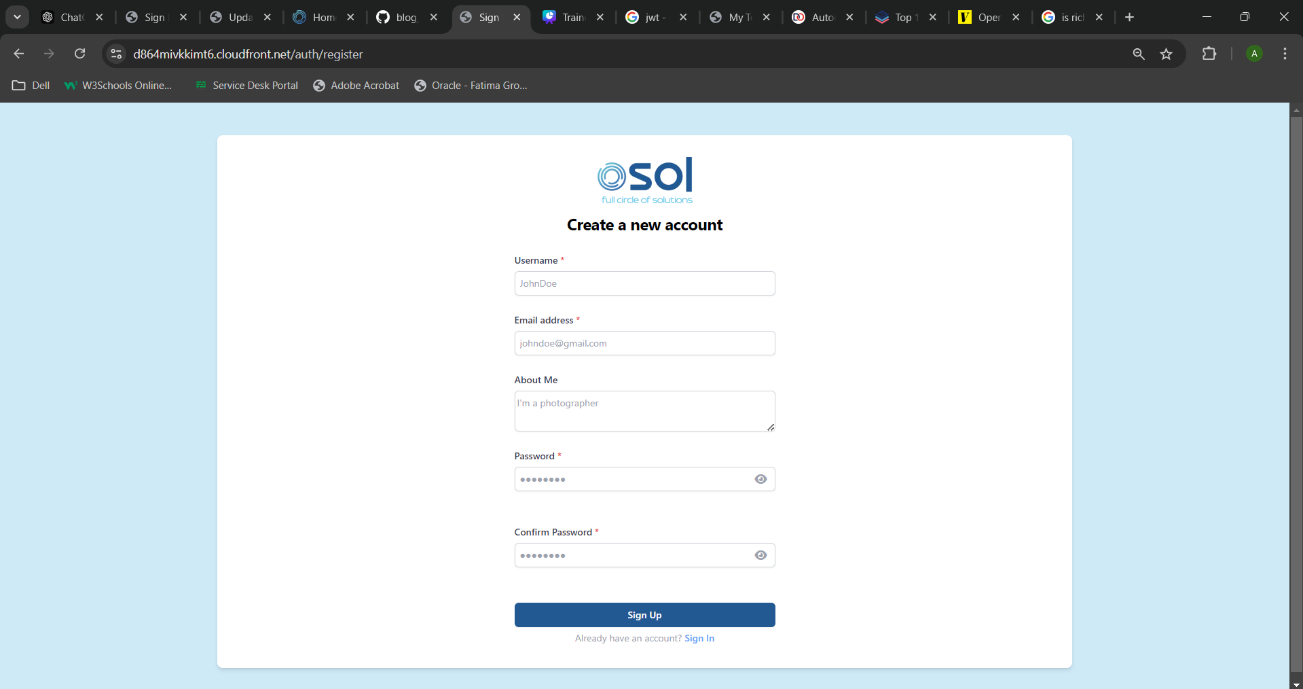


Figure 2: Sign Up page

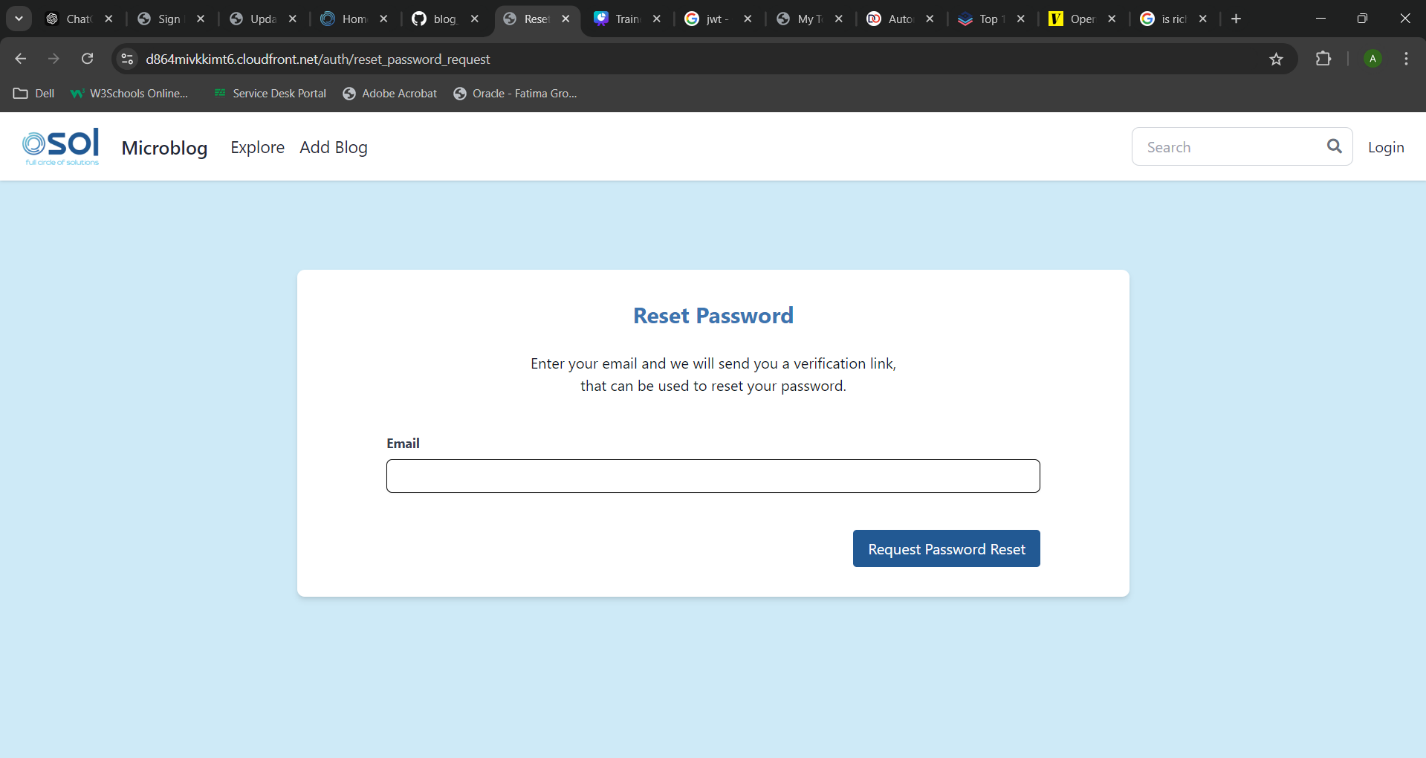


Figure 3: Request Password Reset Page

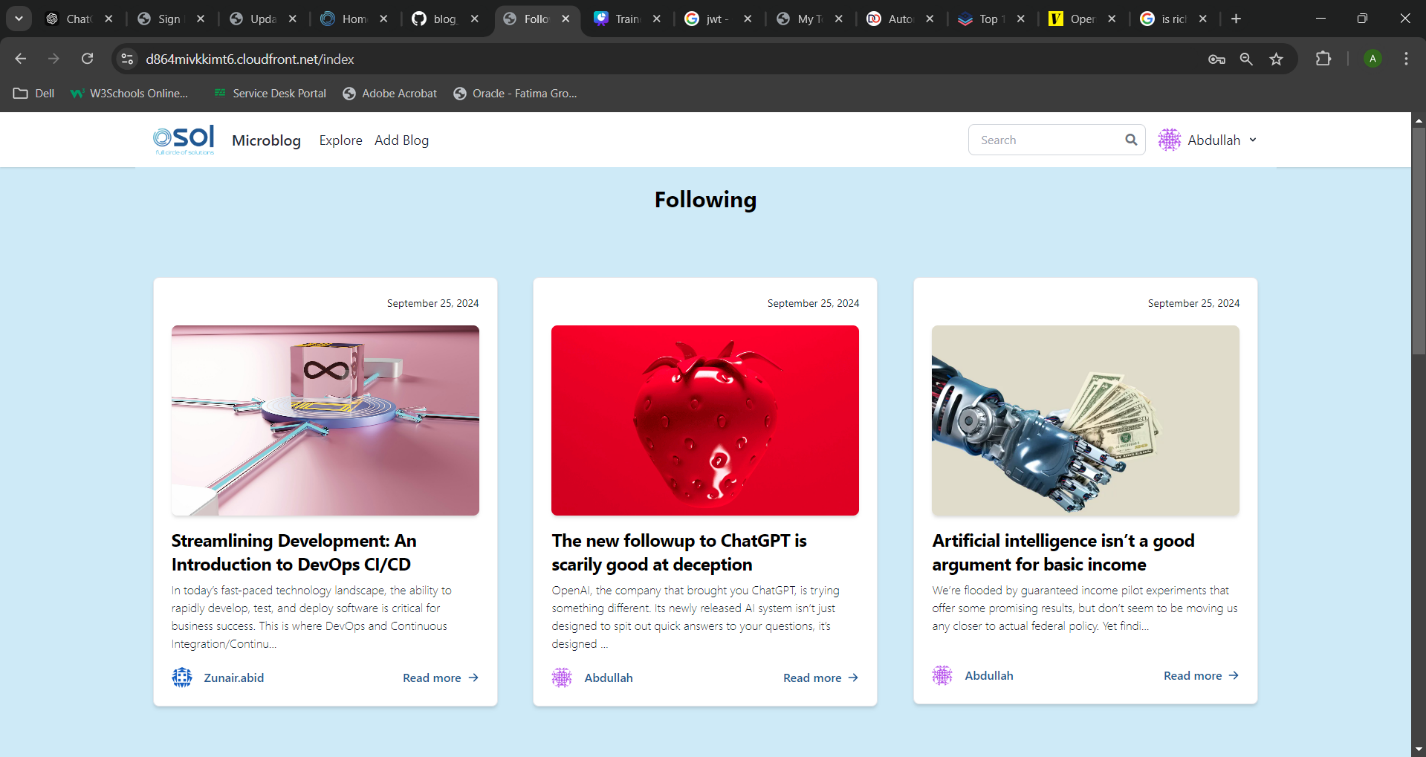


Figure 4: Home/Following Page

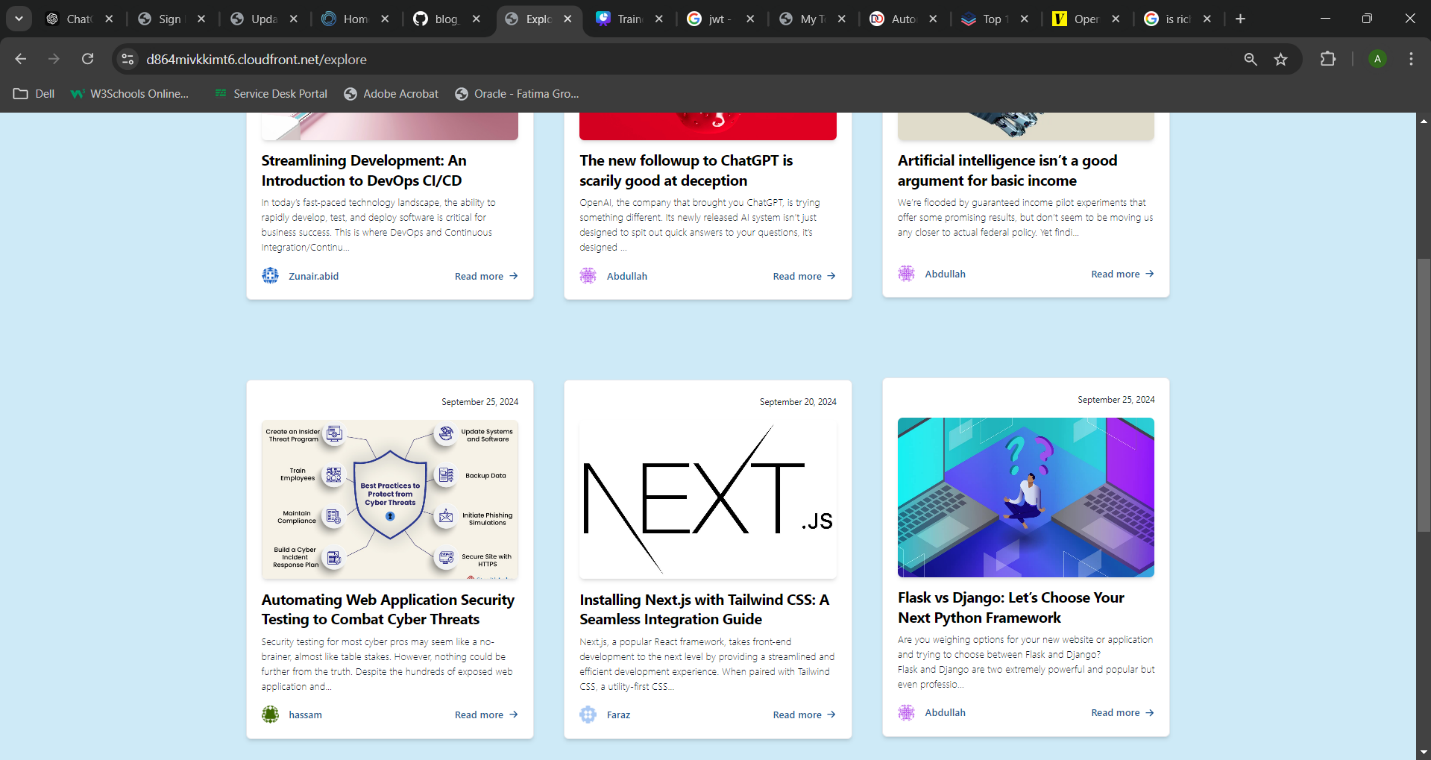


Figure 5: Explore Page

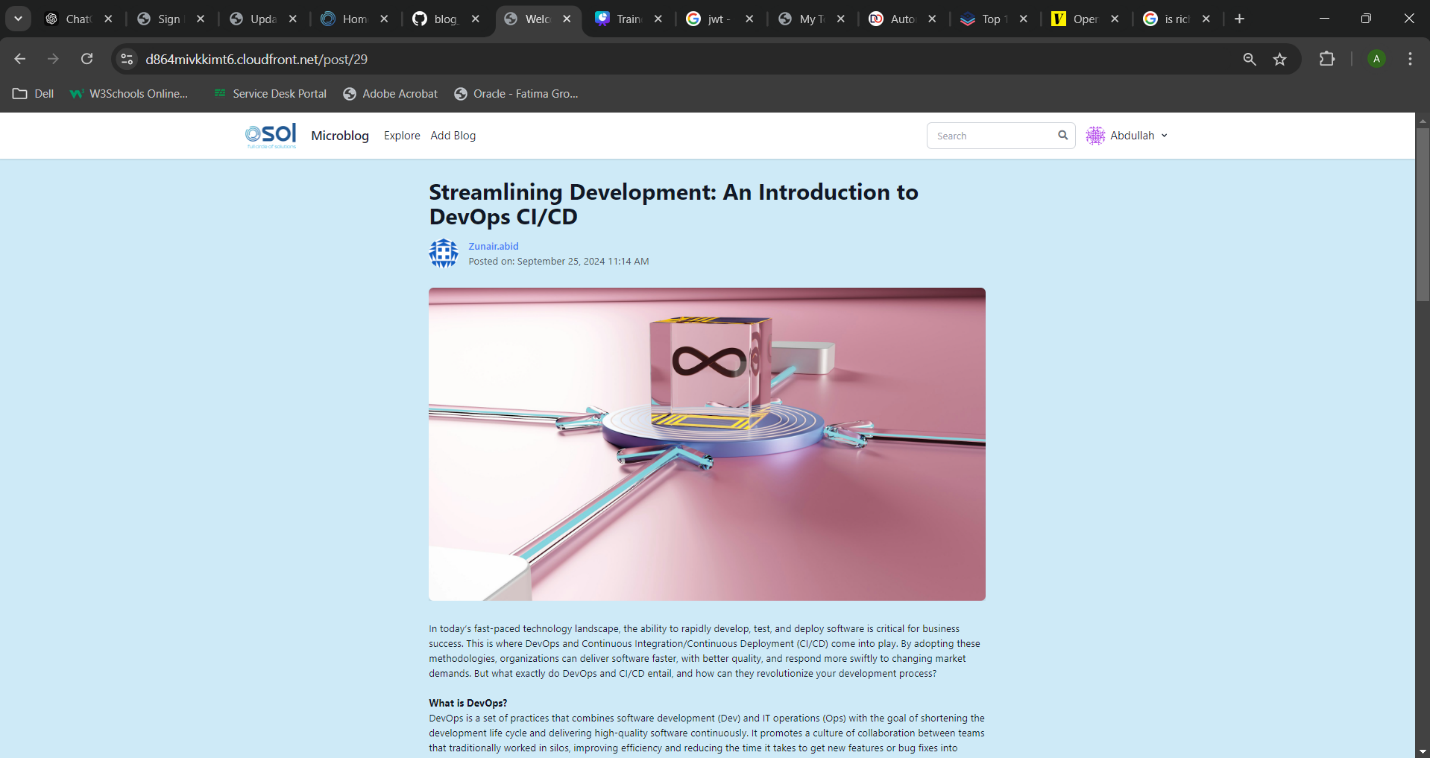


Figure 6: View Blog Page

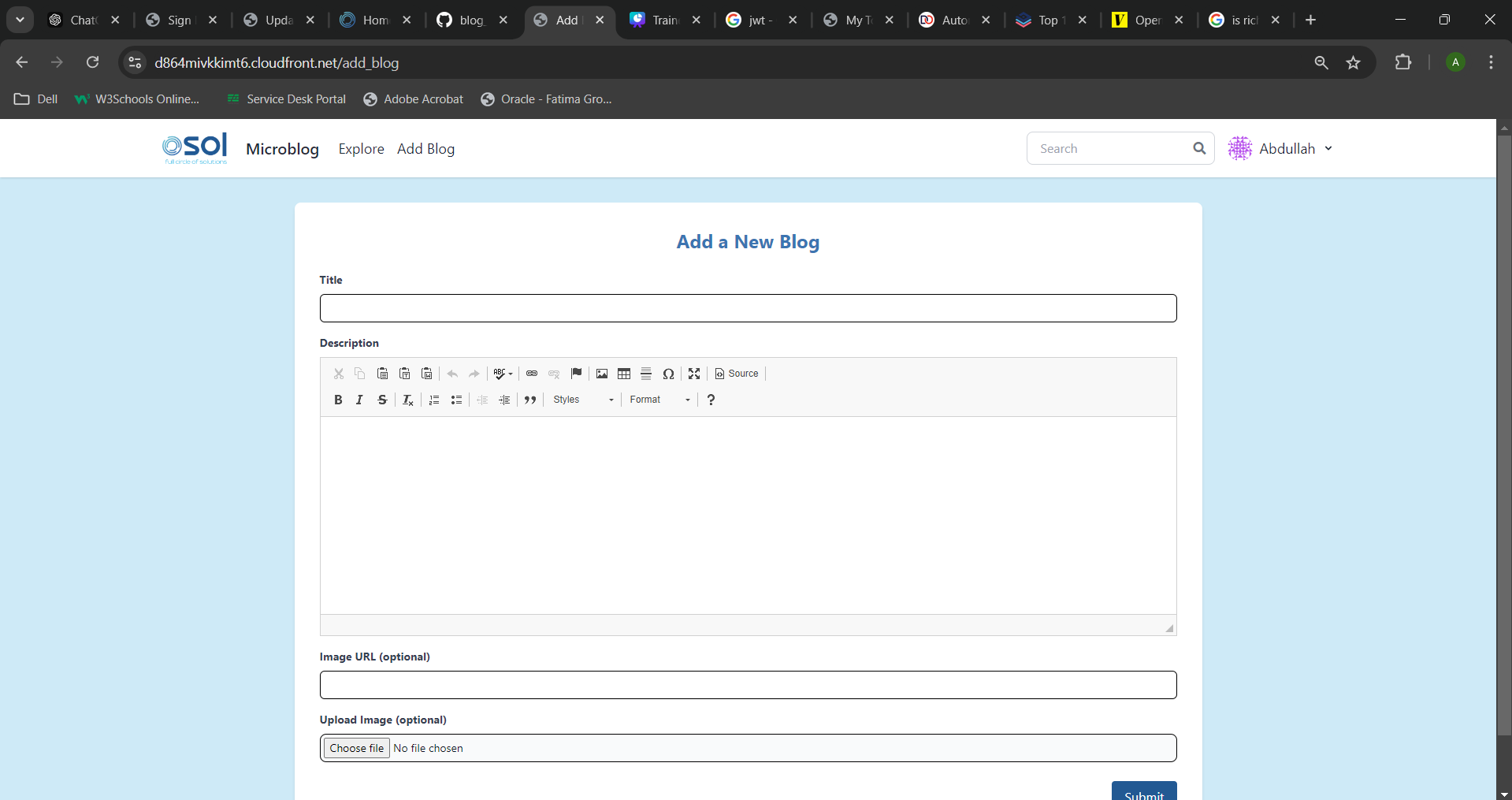


Figure 7: Add Blog Page

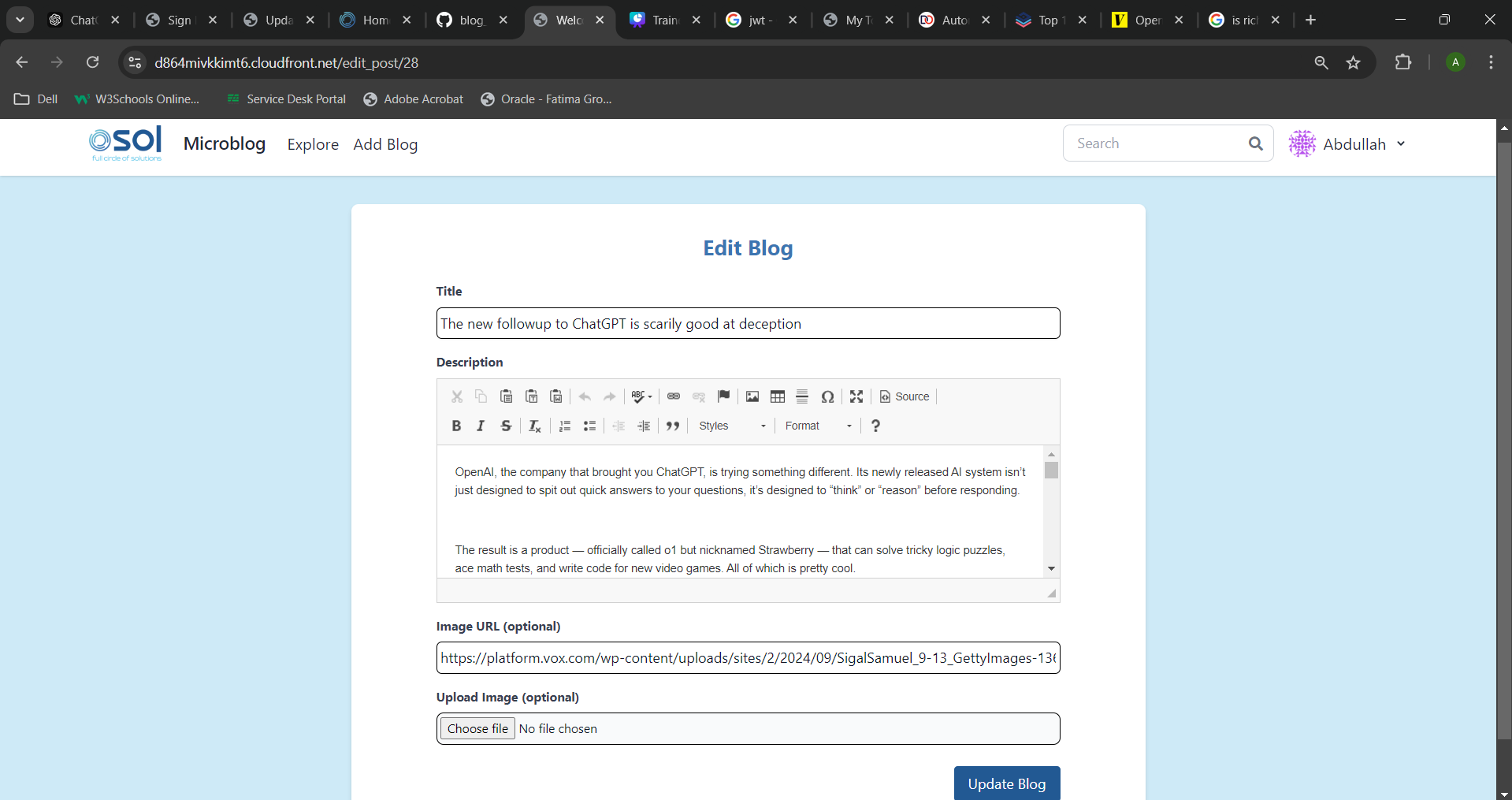


Figure 8: Edit Blog Page

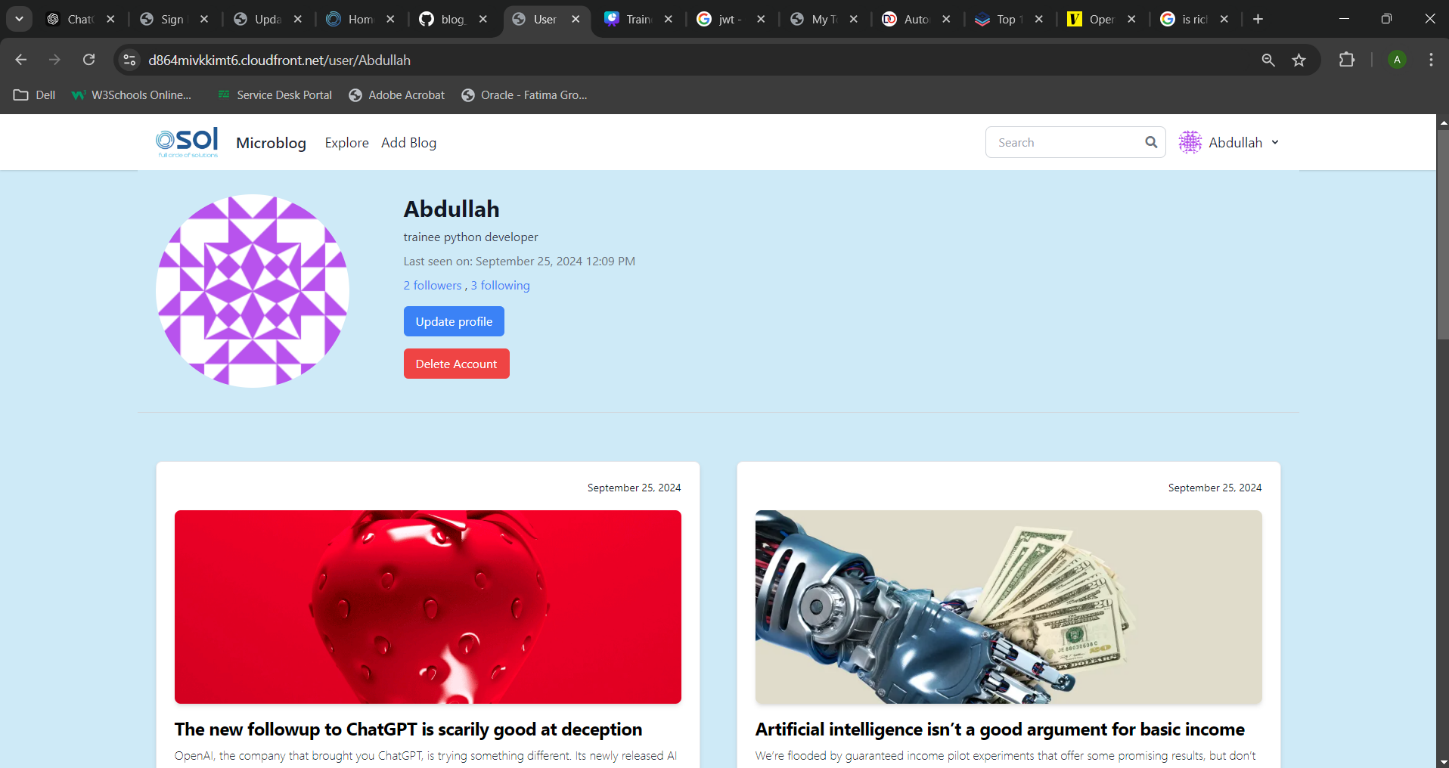


Figure 9: Profile View Page

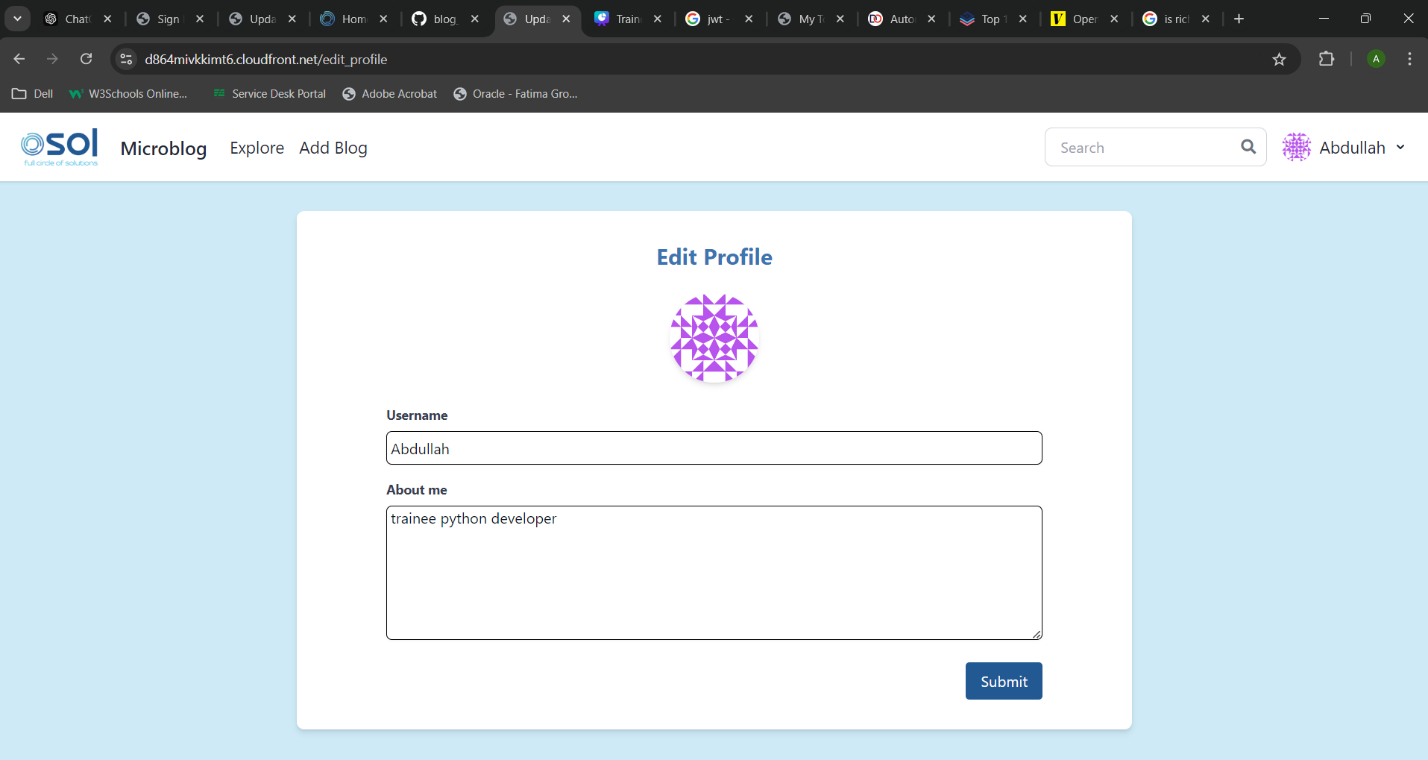


Figure 10: Update Profile Page

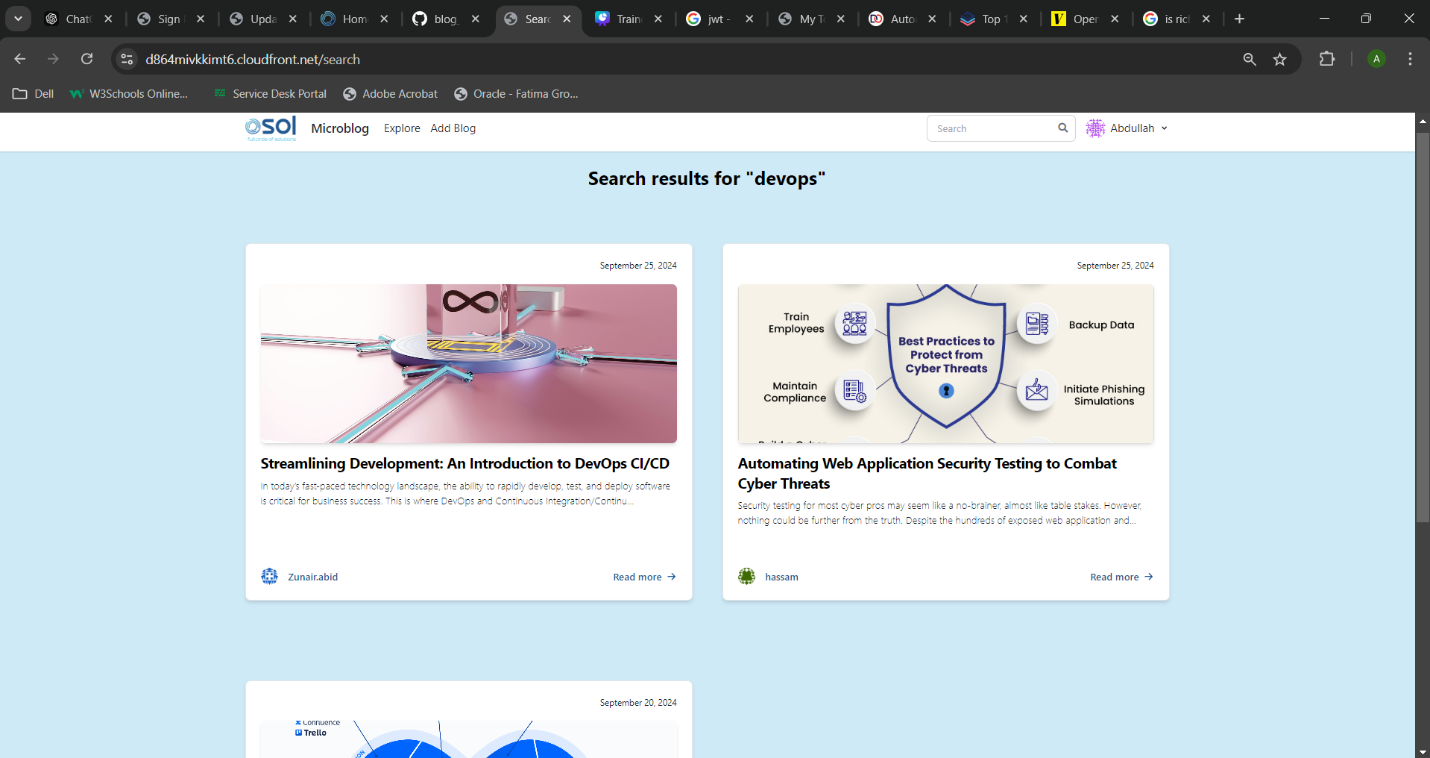
**

Figure 11: Search Page

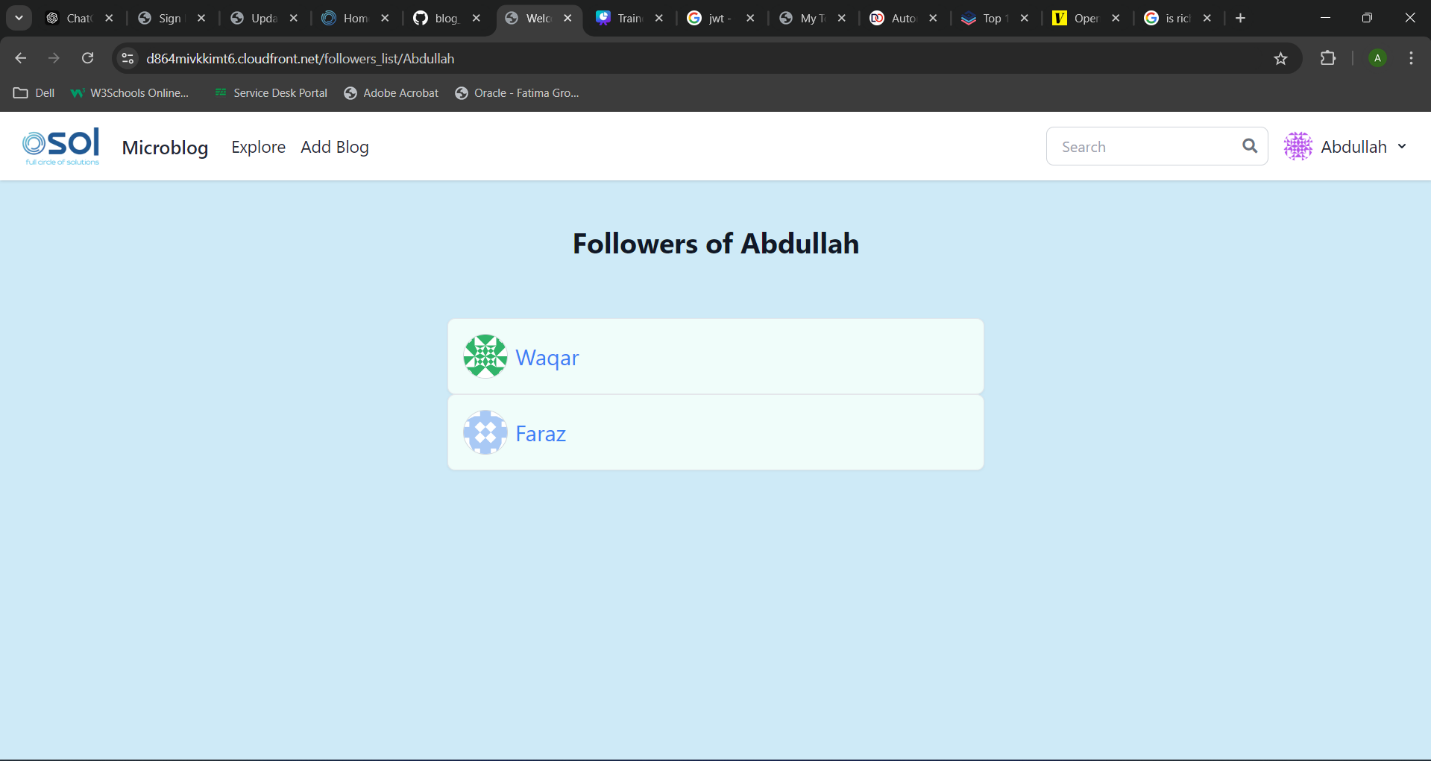


Figure 12: List of Followers

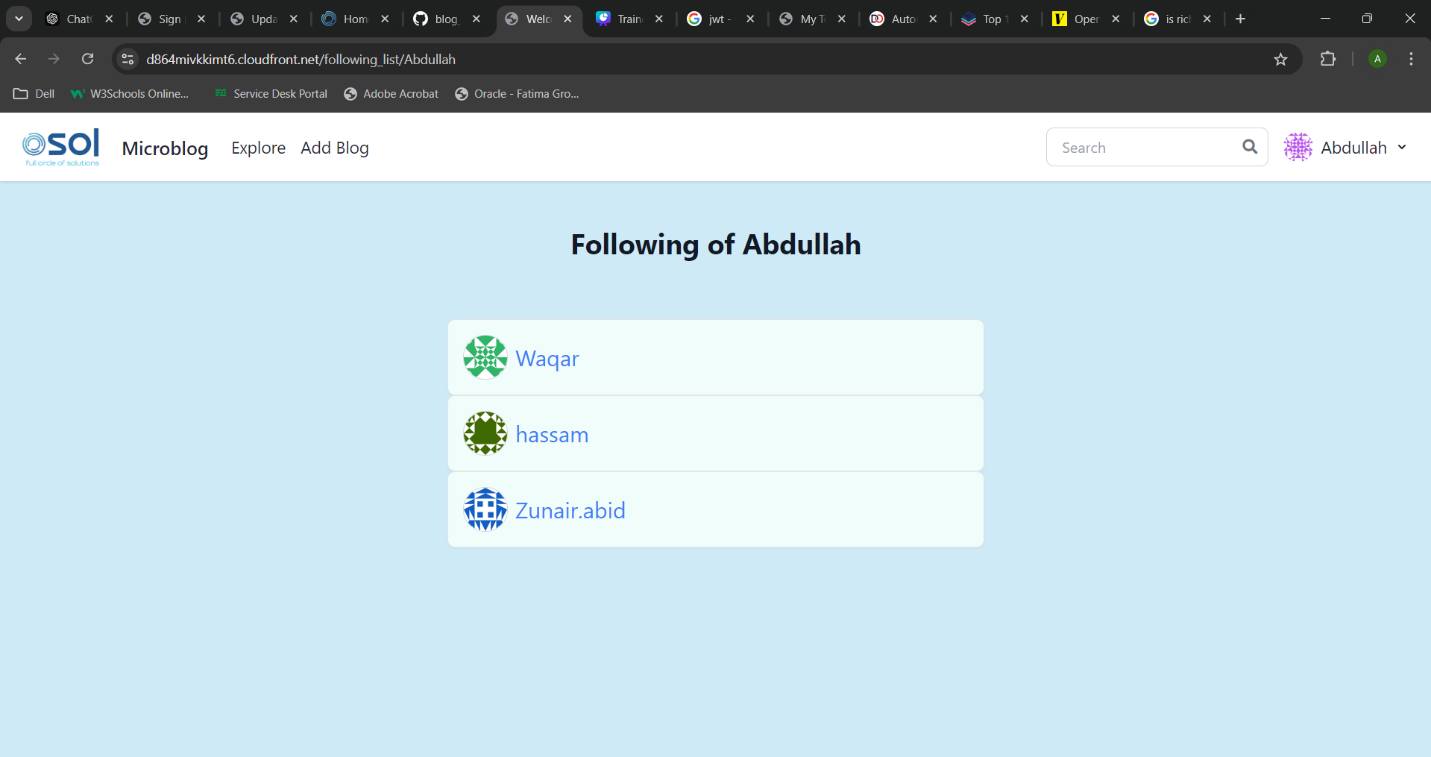


Figure 13: List of Following

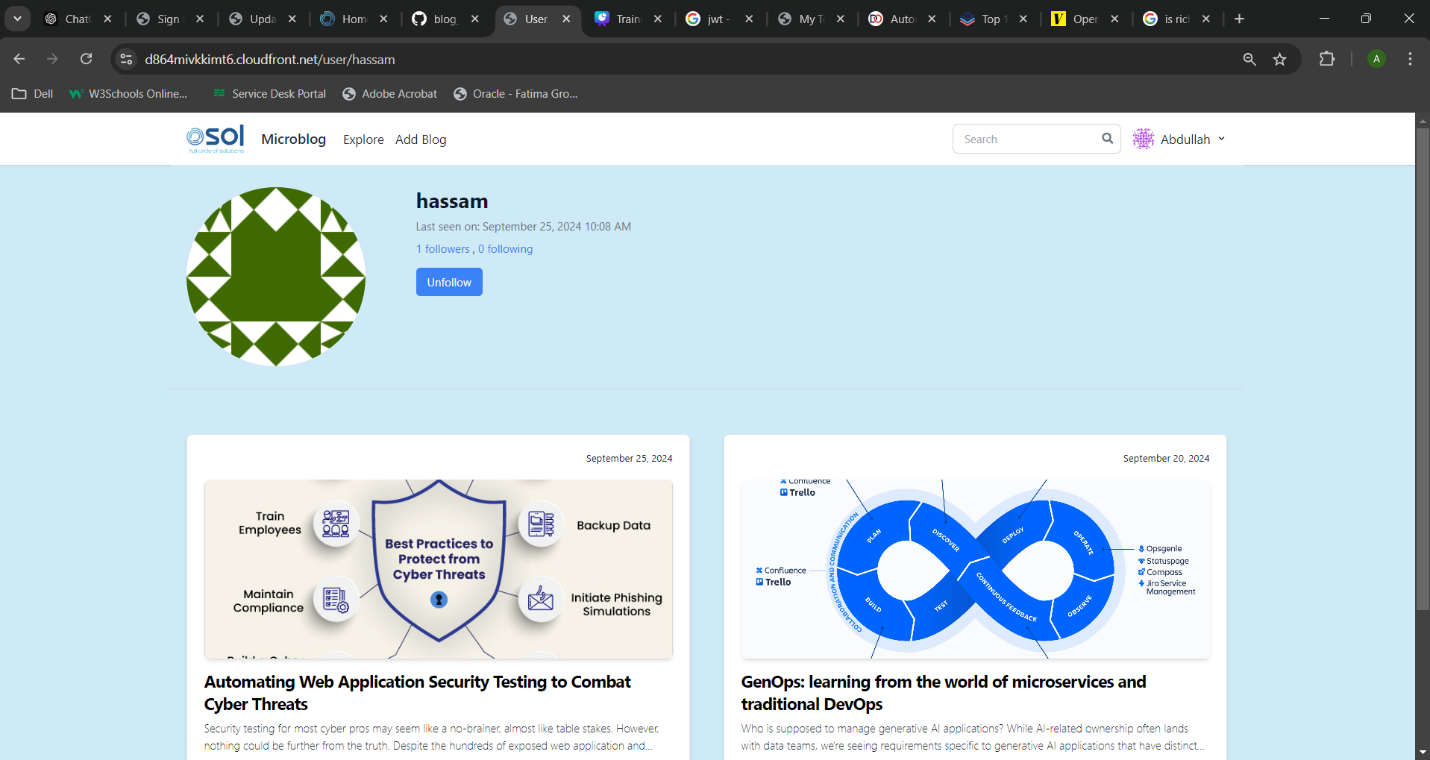


Figure 14: Other User's Profile Page

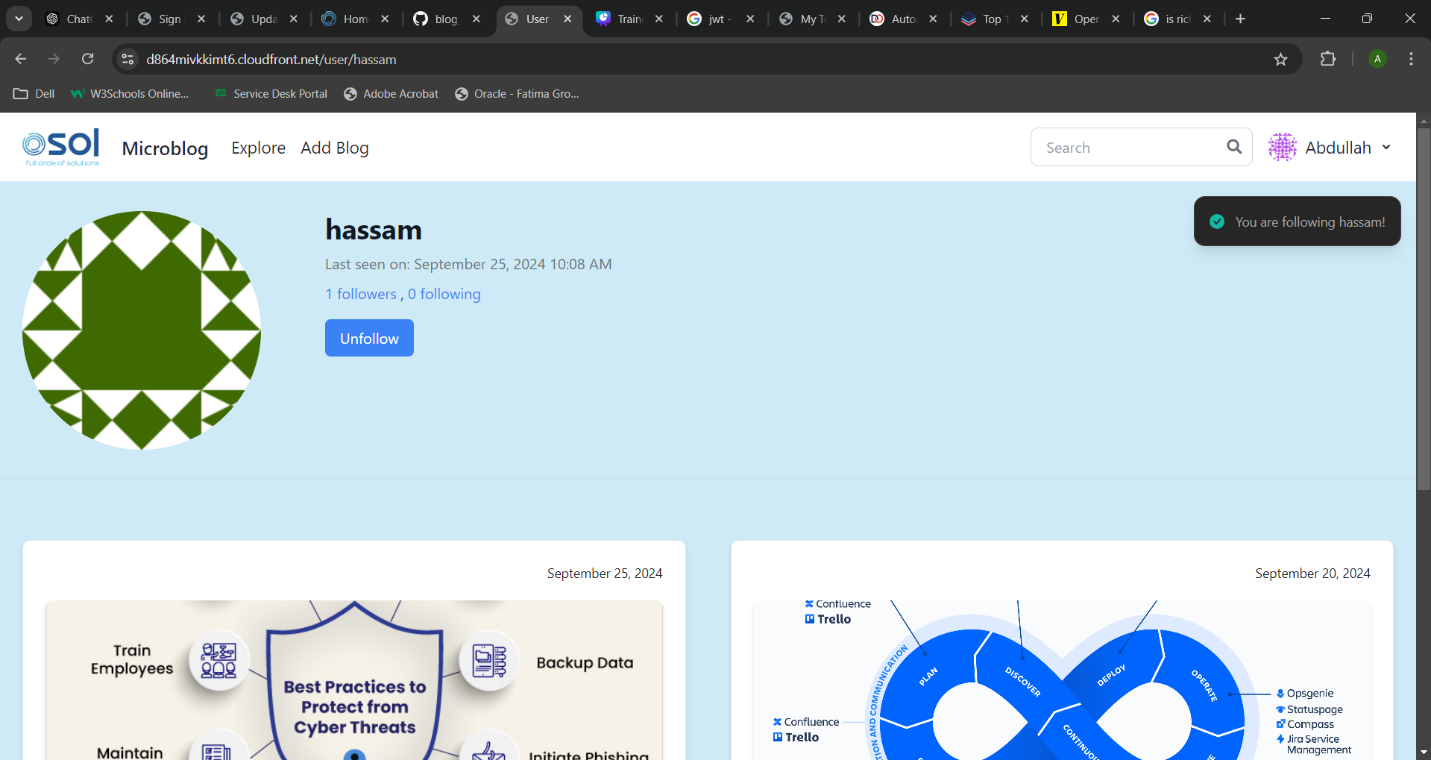


Figure 15: Toasts used to display alerts

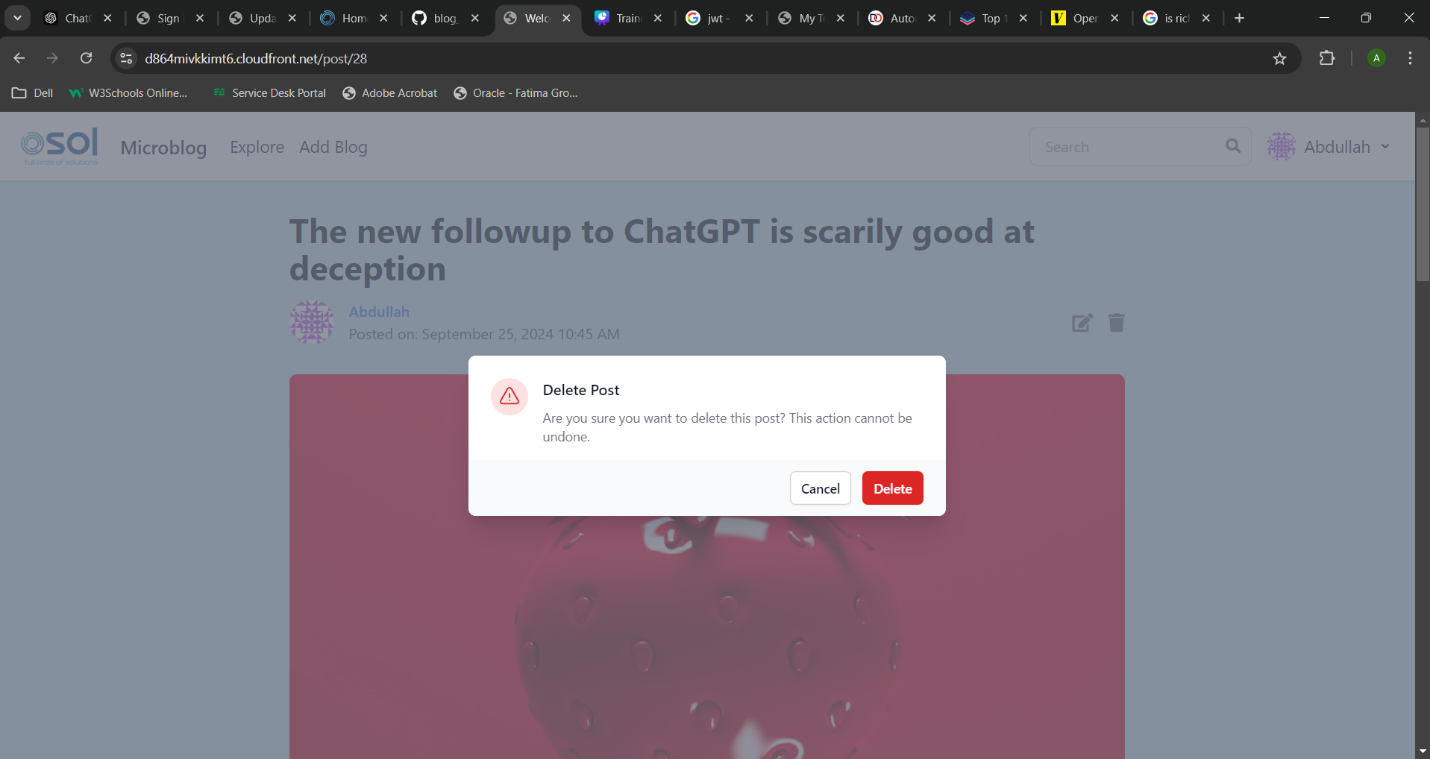


Figure 16: Modals for confirmation

# 4 CONCLUSIONS

The blog application successfully implements a comprehensive and user-friendly platform for creating, managing, and exploring blog content. By utilizing Flask for the backend, HTML, JavaScript, and Tailwind CSS for the frontend, and SQLite for the database, the project offers a smooth and responsive user experience across devices. Key features like secure authentication, rich text editing for posts, commenting, profile management, and the ability to follow/unfollow users enhance user interaction and engagement. With pagination, search functionality, and email notifications, the application ensures both performance and ease of use.

The non-functional requirements, including security, reliability, performance, and scalability, ensure that the system operates efficiently while maintaining data integrity and providing a secure environment. Additionally, the modular design and well-documented codebase facilitate easy maintenance and future updates, ensuring that the platform is adaptable to user growth and evolving needs.

Overall, the project showcases a robust, scalable, and user-centered blogging platform, adhering to both functional and non-functional requirements while delivering a seamless blogging experience.