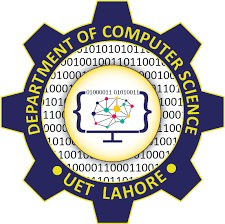
**INSIDER THREAT DETECTION SYSTEM**

****

**Session 2023 – 2027**

**Submitted By:**

Sheharyar Zahid 2023-CS-30

Syed M. Zaeem 2023-CS-38

**Supervised By:**

Dr. Faiza Iqbal

**Course:**

CSC-201L Information Security

**Department Of Computer Science**

**University Of Engineering And Technology**

**Lahore, Pakistan**

**Table of Contents**

[**1.** **Introduction** 3](#_Toc197987591)

[**2. Project Objectives** 3](#_Toc197987592)

[**3. Technologies Used** 4](#_Toc197987593)

[**4. Suspicious Activities** 4](#_Toc197987594)

[**5. How the System Works** 5](#_Toc197987595)

[**5.1 Employee Monitoring** 5](#_Toc197987596)

[**5.2 Admin Control Panel** 5](#_Toc197987597)

[**6. Features of the System** 5](#_Toc197987598)

[**6.1 Real-time Monitoring** 5](#_Toc197987599)

[**6.2 Threat Level Bar** 5](#_Toc197987600)

[**6.4 Simple and Clean Admin Panel** 6](#_Toc197987601)

[**7. Live Mouse Movement Tracking** 6](#_Toc197987602)

[**8. Email Alerts for Harmful Words** 6](#_Toc197987603)

[**9. Blocking Specific Keys for Users** 6](#_Toc197987604)

[**10. Blocking Based on Suspicious Activity** 7](#_Toc197987605)

[**11. Employee Notification on Blocking** 7](#_Toc197987606)

[**12. Account Locking After Multiple Failed Logins** 7](#_Toc197987607)

[**12.1 How It Works:** 7](#_Toc197987608)

[**12.2 Purpose and Benefits:** 8](#_Toc197987609)

[**12.3 Future Improvements:** 8](#_Toc197987610)

[**13. How We Built It (Methodology)** 8](#_Toc197987611)

[**13.1 Planning** 8](#_Toc197987612)

[**13.2 Designing** 8](#_Toc197987613)

[**13.3 Development** 9](#_Toc197987614)

[**13.4 Testing** 9](#_Toc197987615)

[**13.5 Final Product** 9](#_Toc197987616)

[**14. Challenges We Faced** 9](#_Toc197987617)

[**14.1 Tracking Data Accurately** 9](#_Toc197987618)

[**14.2 Handling IP Address Changes** 9](#_Toc197987619)

[**14.3 Designing a Clean Admin Panel** 9](#_Toc197987620)

[**15. Results and Achievements** 10](#_Toc197987621)

[**16. Conclusion** 10](#_Toc197987622)

[**Wire Frames** 11](#_Toc197987623)

[**Admin Pannel** 12](#_Toc197987624)

[**User Activities** 15](#_Toc197987625)

# **1.** **Introduction**

Every company depends on its employees to work honestly and safely. But sometimes, even trusted employees can cause problems either by mistake or on purpose. These problems can be very costly for a business.

Our project, **Insider Threat Detection**, is made to help companies keep an eye on employee activities and quickly find any suspicious behaviour. It helps protect the company before serious damage happens.

This system watches employee actions like what keys they press, how their mouse moves, which websites they visit, and their location and IP address. If anything, suspicious happens, the admin can take quick action.

# **2. Project Objectives**

The main goals of our project are:

* To detect risky or suspicious behaviour from employees.
* To track the activities of employees in real time.
* To give admins a simple dashboard to monitor all employees.
* To show a threat level for each employee account.
* To allow admins to block any employee based on their IP address if needed.

In short, the aim is to catch problems early and protect the company.

# **3. Technologies Used**

For this project, we used the following technologies:

* **Backend:** Node.js
* **Frontend:** HTML and CSS
* **Admin Panel:** Built with HTML, CSS, and connected to backend APIs
* **Other Features:**
  + IP tracking system
  + Threat level calculation based on user activities

This tech stack was chosen because it is lightweight, fast, and easy to update when needed.

# **4. Suspicious Activities**

The system keeps a close watch on employee behaviour and flags any actions that seem unusual or risky. This helps the admin spot possible threats early and take action before any harm is done.

Some of the suspicious activities the system detects include:

* **Frequent Login/Logout Attempts:** Logging in and out many times in a short period.
* **Use of Dangerous Tags:** Typing or sending suspicious tags like SQL Injection codes (' OR 1=1 --) or scripts.
* **Failed Login Attempts:** Trying to log in with the wrong password multiple times.
* **Access from Unknown IP:** Logging in or performing activities from a different or unknown IP address.

# **5. How the System Works**

Our system mainly has two parts:

## **5.1 Employee Monitoring**

* **Key Logger:** It records the keys the employee presses.
* **Mouse Tracker:** It keeps track of how and where the mouse moves.
* **Website Tracker:** It checks what websites the employee opens.
* **Location and IP Tracking:** It keeps updating where the employee is and what IP address they are using.

## **5.2 Admin Control Panel**

The admin has access to all employee activities through a dashboard.  
The admin can:

* See live updates of employee behaviour.
* Check the **threat level** bar for every user.
* Block any user by their IP address if suspicious activity is detected.

# **6. Features of the System**

Here are some important features:

## **6.1 Real-time Monitoring**

The system shows employee activities as they happen.

## **6.2 Threat Level Bar**

Shows how risky an employee's actions are.

**6.3 Easy User Blocking**

The admin can block a user in one click using their IP address.

## **6.4 Simple and Clean Admin Panel**

The admin dashboard is user-friendly and easy to understand.

These features help the company stay one step ahead of any inside threats.

# **7. Live Mouse Movement Tracking**

The admin can see the employee’s mouse movement in real time.  
This helps the admin understand if the employee is working normally or behaving suspiciously, like making random movements, trying to hide something, or showing signs of misuse.

# **8. Email Alerts for Harmful Words**

If an employee types harmful words like script, SQL Injection attempts, or any suspicious codes, the system automatically sends an **email alert** to the admin using **Mail.js**.  
This makes sure the admin is quickly informed about any possible attack or misuse without even checking the panel constantly.

# **9. Blocking Specific Keys for Users**

The admin has the ability to **block specific keys** on any employee’s keyboard.  
For example, if someone is misusing shortcuts or commands, the admin can block those particular keys to stop harmful actions without fully blocking the employee.

# **10. Blocking Based on Suspicious Activity**

The system can **analyse user behaviour** and allow the admin to **block an employee based on the activities** detected as suspicious.  
This means if the system sees too many login attempts, harmful code typing, or weird mouse movements, the admin can quickly block the user before things get worse.

# **11. Employee Notification on Blocking**

When an employee is blocked, they will **receive a notification** explaining that their access has been restricted.  
The message will also **mention the reason for the block**, so the employee knows what rule or activity caused it.  
This keeps communication clear and avoids confusion.

# **12. Account Locking After Multiple Failed Logins**

To strengthen system security and prevent brute-force attacks or unauthorized access attempts, a new functionality has been introduced into the Insider Threat Detection System: **automatic account lockout after multiple failed login attempts**.

## **12.1 How It Works**

* If a user enters the **wrong password five times consecutively**, the system will **automatically block the account**.
* Once blocked, the account will be **locked for a duration of 5 minutes**.
* During the lockout:
  + The user will be shown a **clear notification** that their account is temporarily locked due to repeated failed login attempts.
  + A **countdown timer** may be displayed to inform the user how much time remains before they can try again.
  + **All login requests for that user ID will be denied** until the timeout expires.
* After 5 minutes, the system will automatically **reactivate the account**, and the user can attempt to log in again.

## **12.2 Purpose and Benefits**

* Prevents **brute-force attacks** by limiting repeated password guessing.
* Adds an **extra layer of protection** against unauthorized access.
* Reduces the risk of internal misuse where employees might try to access another account.

## **12.3 Future Improvements**

* Adding **CAPTCHA verification** after multiple failed attempts.
* Tracking **geolocation of login attempts** to identify external threats.
* Implementing **adaptive lockout duration** based on threat level (e.g., increasing block time with more failed attempts).

This feature enhances overall system robustness and aligns with best practices in information security, ensuring that employee accounts remain protected against unauthorized access attempts.

# **13. How We Built It (Methodology)**

We followed these simple steps while making the project:

**13.1 Planning**  
We first understood the real problem companies face and listed all the features we needed.

**13.2 Designing**  
We made basic designs for the admin panel and employee monitoring systems.

## **13.3 Development**

* Backend was developed using **Node.js**.
* Frontend was designed using **HTML and CSS**.
* APIs were made for communication between frontend and backend.

**13.4 Testing**  
We tested the system by simulating normal and suspicious employee behaviour to make sure it detects correctly.

**13.5 Final Product**  
After testing, the system was made ready to use.

# **14. Challenges We Faced**

While building this project, we had some challenges:

**14.1 Tracking Data Accurately**  
Making sure we capture mouse movements and keystrokes without slowing down the system.

**14.2 Handling IP Address Changes**  
Sometimes employees' IPs change, so we had to ensure blocking still works.

**14.3 Designing a Clean Admin Panel**  
We wanted the admin dashboard to be easy to use even if there are many employees.

We overcame these challenges by doing extra research and making improvements during testing.

# **15. Results and Achievements**

After finishing the project, here’s what we achieved:

* A working system that tracks employee behaviour in real time.
* A simple and clean admin panel that shows threat levels.
* Easy IP-based blocking feature for admins.
* Accurate detection of suspicious activities based on user behaviour.

The project successfully meets its goal of helping companies detect insider threats early.

# **16. Conclusion**

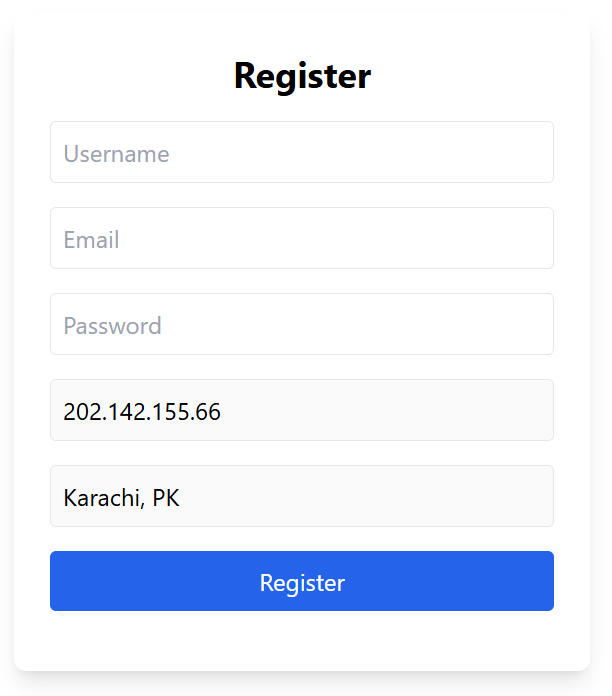
Insider threats can cause serious harm to any company if not detected early.  
Our project **Insider Threat Detection** provides a strong tool to monitor and manage employee activities and catch risks before they become big problems.

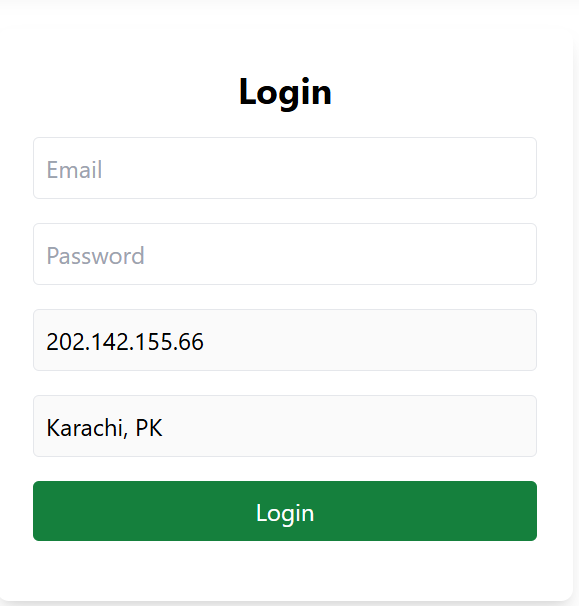
In the future, we can make this system even better by adding:

* Machine Learning to predict threats automatically.
* Better reporting tools for management.
* Make the mouse tracking even more batter that shows even the clicks

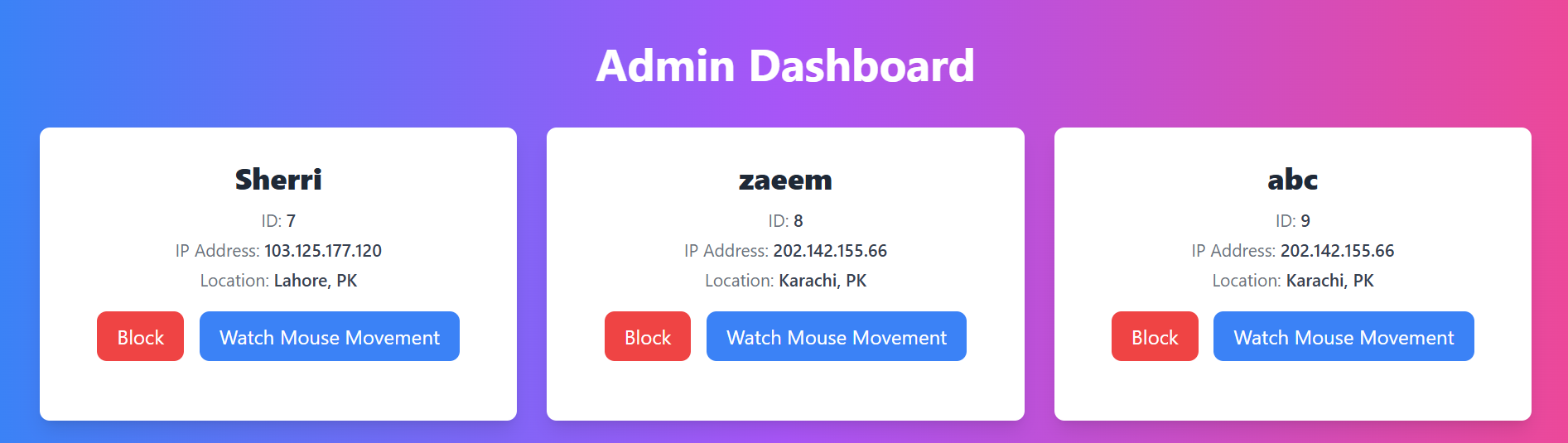
Overall, this project lays a strong foundation for a safer and smarter company environment.

# **Wire Frames**

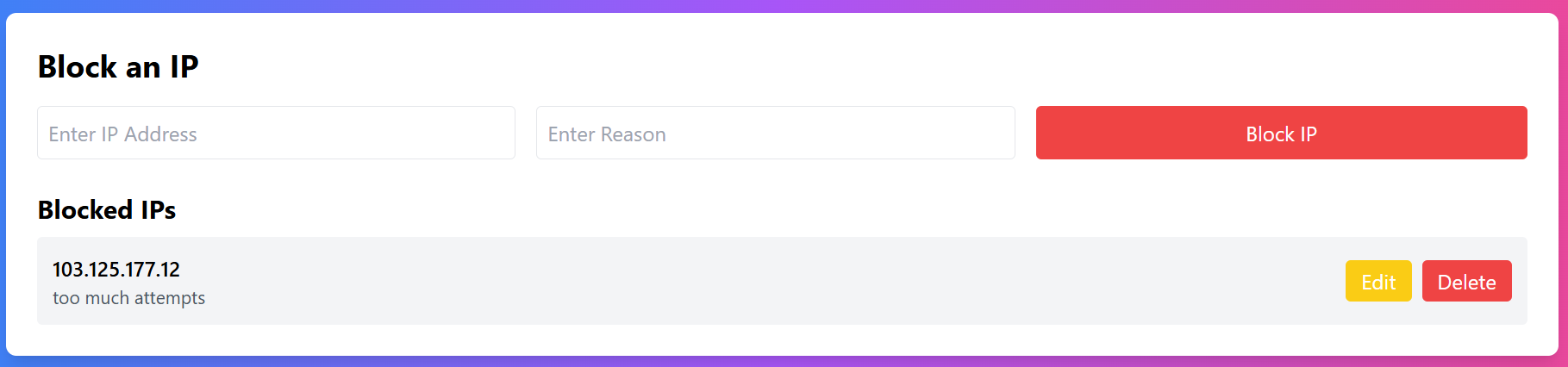
****



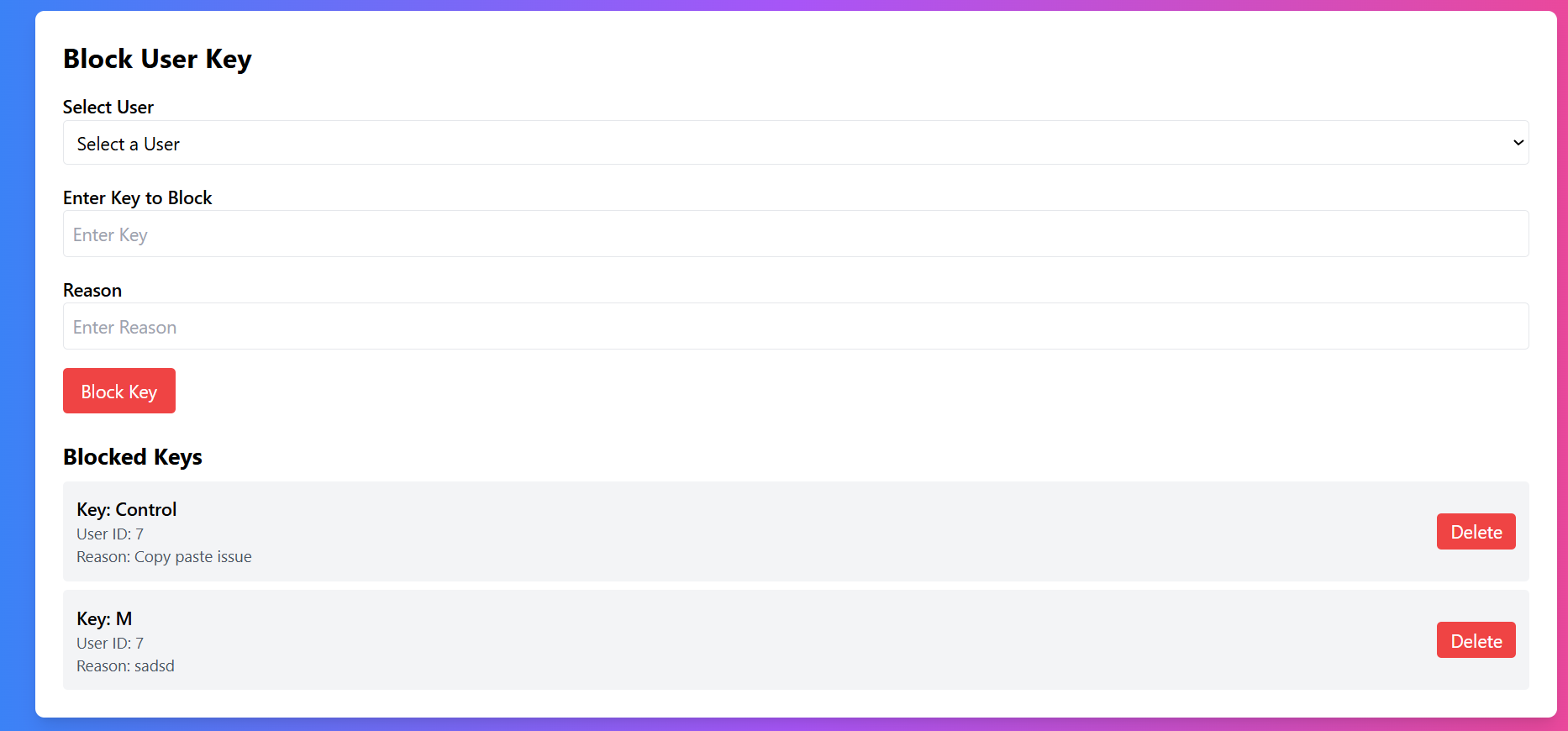
# **Admin Pannel**



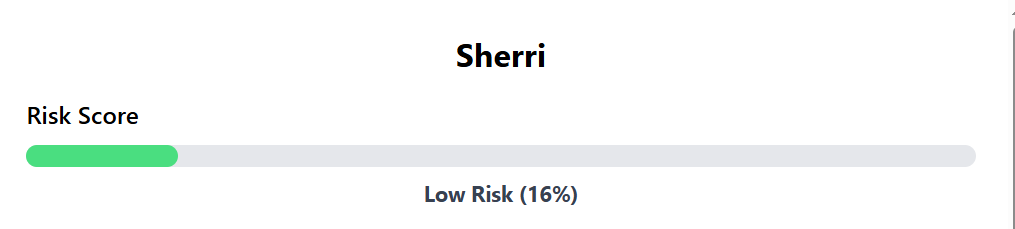
Shows All Users



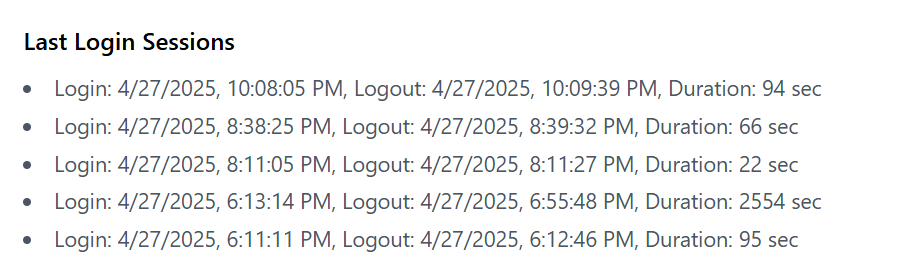
To Block Any IP



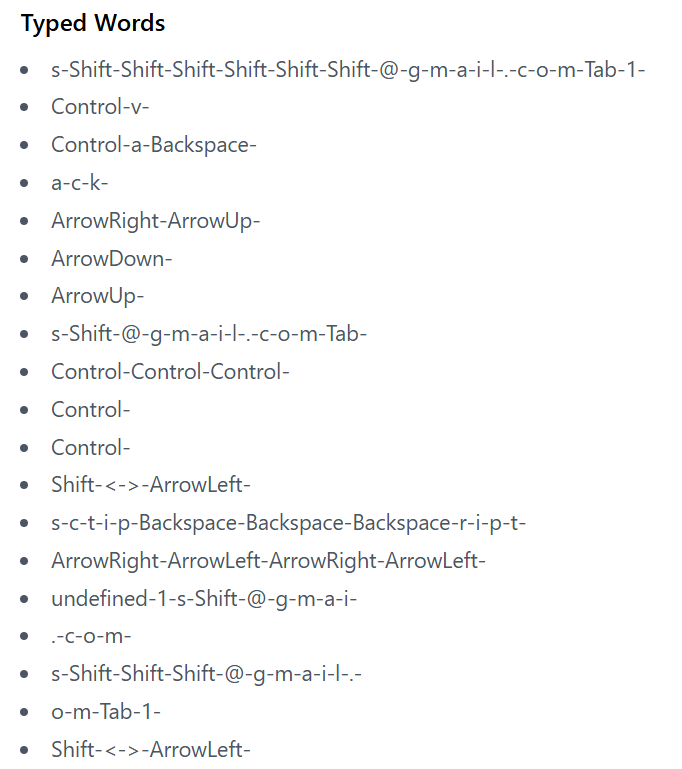
To Block Any Key



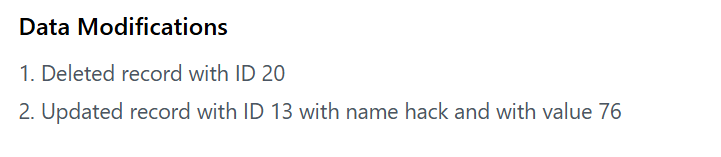
Shows Risk Percentage



Shows Loging And Logout Time

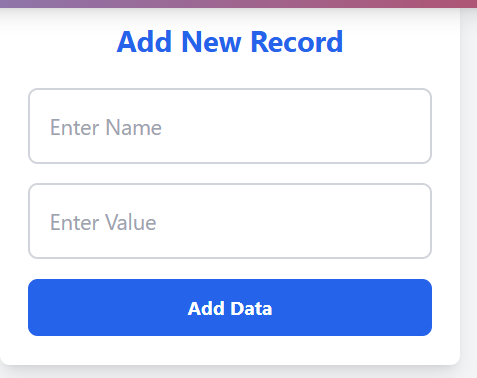


Shows Keys Pressed

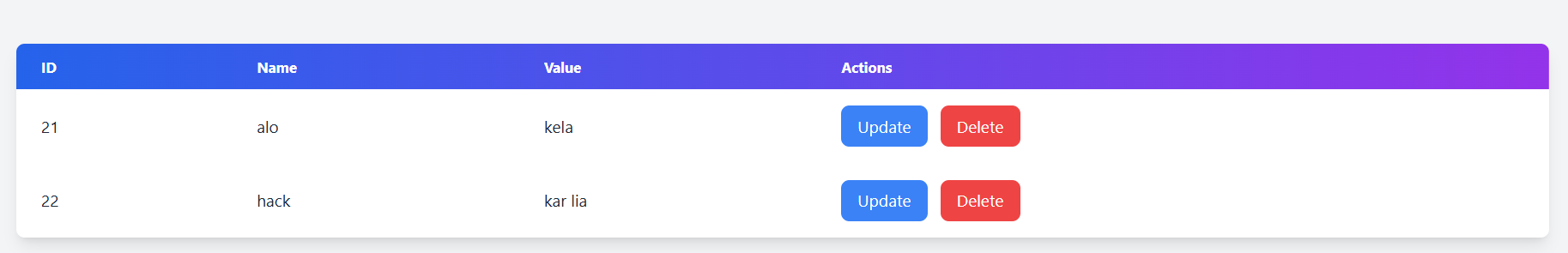


Track Activity

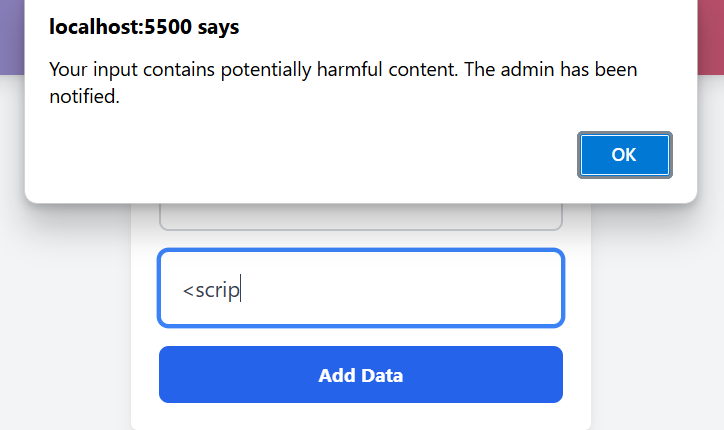
# **User Activities**

****

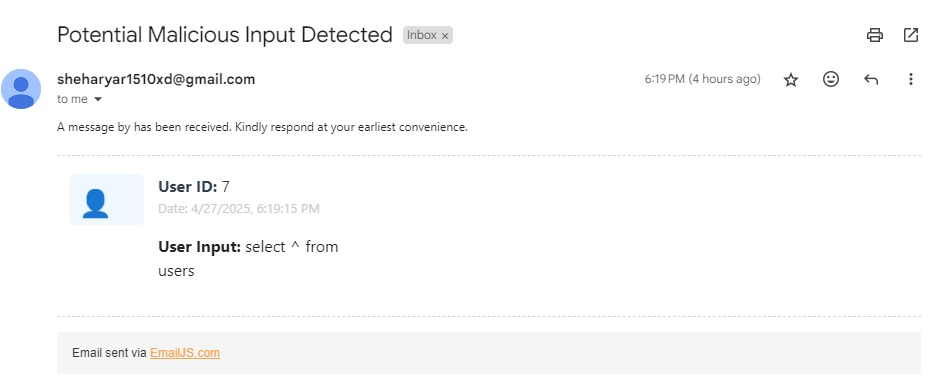
Add Any Record



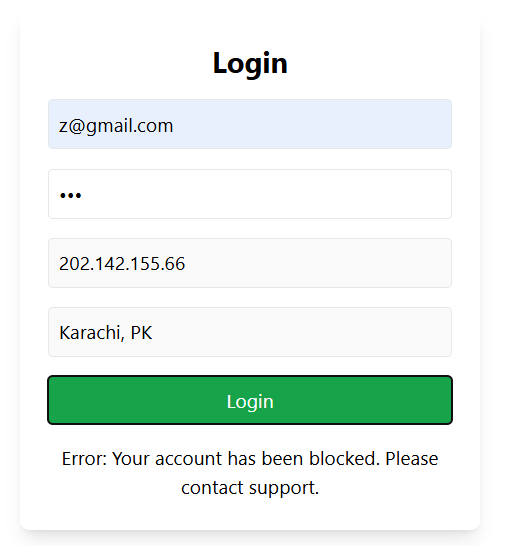
Data Manipulation



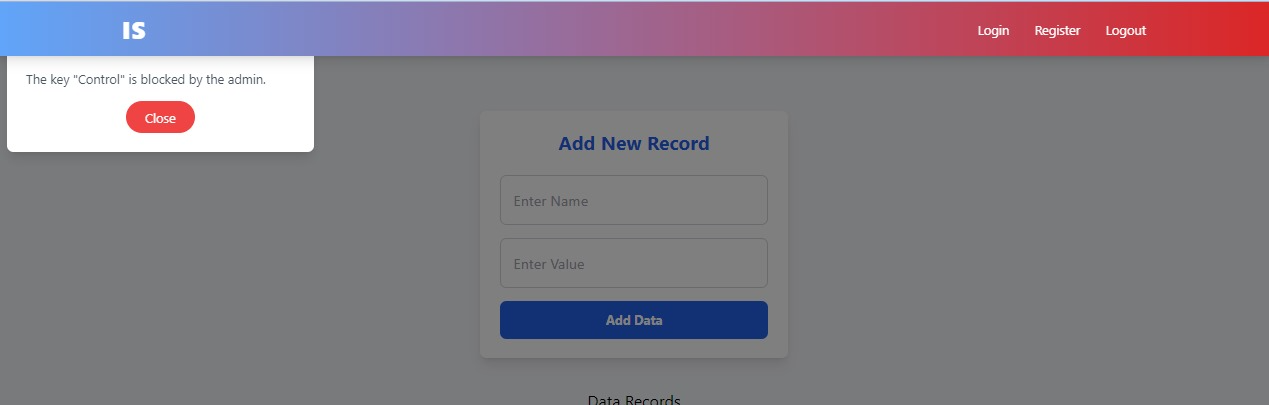
If Scripting Is Used



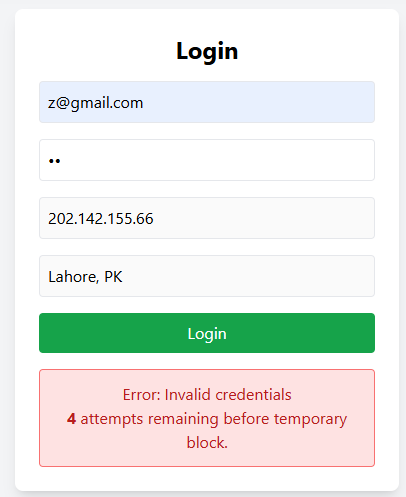
Sample Email



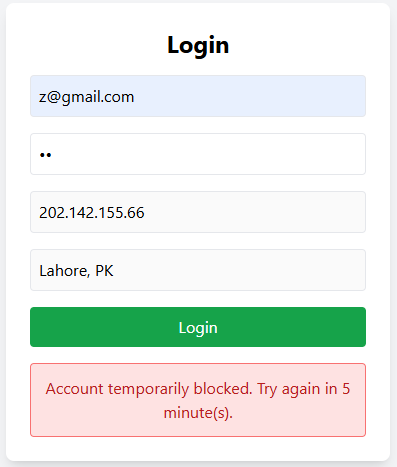
When Blocked User Try To Login



When User Try To Press Any Blocked Key



Message If Password Is Wrong



If Password Is Wrong For 5 Times