Risk Analysis for Anonymous Chat Web App

1: Identifying Key Risks

Every software project has risks that can affect its performance, security, and user experience. Below are the key risks for the **Anonymous Chat Web App**:

- **Security Risk** – Hackers may try to break into the system, steal user data, or cause disruptions.
- **Anonymity Misuse** – Some users might misuse anonymity to send harmful or inappropriate messages.
- **Performance Issues** – The system may slow down or crash if too many users are active at once.
- **Scalability Risk** – If the number of users increases, the server and database might not handle the load properly.
- **Data Storage Risk** – Storing chat history securely while maintaining anonymity is challenging.
- **Spam & Abuse Risk** – Without proper controls, the chat may be flooded with spam or harmful content.

2: Risk Table

A risk table helps in understanding the severity of each risk by calculating **Risk Exposure (RE)** using the formula:

RE=Probability(P)×Impact(C)RE = Probability (P) \times Impact (C)

Where:

- **Probability (P)**: Likelihood of the risk occurring (0.1 = low, 1.0 = high)
- **Impact (C)**: How serious the risk is (1 = low, 5 = critical)

Risk	Probability (P)	Impact (C)	Risk Exposure (RE = P × C)	How to Reduce the Risk?
Hacking & security breach	0.8	5	4.0	Use strong encryption, secure authentication, and firewalls.
Misuse of anonymity	0.7	4	2.8	Add moderation tools, silent user removal for admins.
Slow performance & crashes	0.6	4	2.4	Optimize database, use load balancing, and WebSockets for real-time chat.
Scalability issues	0.5	3	1.5	Use cloud hosting and scalable database solutions.
Data security concerns	0.5	3	1.5	Encrypt chat history, restrict access to admins only.
Spam & abusive messages	0.7	3	2.1	Implement AI-based spam filters and allow users to report abuse.

3: Managing and Monitoring Risks

To keep the system safe and efficient, we follow these three steps:

1. **Mitigation (Prevention)** – Taking steps to reduce risks before they happen (e.g., adding security measures and moderation tools).
2. **Monitoring (Tracking)** – Continuously checking system logs, user reports, and performance.
3. **Management (Handling Issues)** – Having backup plans in case of issues, like emergency fixes for security breaches or server overloads.