**Project Report: Zero-Day Risk Analyzer**

**1. Title & Introduction**

**Project Title:** Zero-Day Risk Analyzer  
**Introduction:**  
Cybersecurity is a critical concern in today’s digital era, with zero-day vulnerabilities representing some of the most dangerous threats to organizations. These are flaws unknown to software vendors and unpatched at the time of exploitation. The Zero-Day Risk Analyzer is a lightweight, browser-based tool developed using HTML, CSS, and JavaScript to help users assess the risk level of vulnerability descriptions. It simulates how automated systems flag and prioritize potential threats based on natural language inputs.

**2. Problem Statement**

Zero-day vulnerabilities are exploited before the public or the software vendors are aware of them, leaving systems unprotected. Early detection and classification of such vulnerabilities are essential for reducing the attack surface. However, many organizations and learners lack access to intelligent, real-time, and easy-to-use triage tools.

**3. Solution Overview**

The Zero-Day Risk Analyzer provides a simple, effective way for users to input textual descriptions of vulnerabilities (e.g., copied from CVE reports) and receive an immediate risk-level prediction (High, Medium, or Low). It relies on predefined keyword detection logic in JavaScript to simulate basic vulnerability assessment.

**4. Architecture or Design**

**Input:** Text field where users paste or write the vulnerability description.  
**Process:** JavaScript analyzes the text for high-risk keywords (e.g., "remote code execution," "unauthenticated," "buffer overflow").  
**Output:** Displays a risk level with corresponding color (Red = High, Orange = Medium, Green = Low).

**Flow Diagram:**

[User Input] → [JS Keyword Analyzer] → [Risk Classification] → [Display Output]

**5. Technologies Used**

* **HTML5:** Structure of the web page
* **CSS3:** Styling, layout, and responsive design (dark mode UI)
* **JavaScript (Vanilla):** Core logic for risk analysis based on keyword detection
* **IDE:** Visual Studio Code
* **Browser Tools:** Chrome DevTools for testing and debugging

**6. Features**

* Real-time, client-side vulnerability risk prediction
* User-friendly and professional UI
* Mobile-responsive design
* Educational and demonstration tool with no installation or setup required

**7. Demo Screenshots/Output**

* Screenshot 1: Input text field and analyze button
* Screenshot 2: Example risk prediction displayed in red (High Risk)
* Screenshot 3: Medium and Low risk examples with explanations

**8. Industry Relevance**

* Zero-day threats affect all industries, including finance, healthcare, government, and tech
* The tool reflects industry need for accessible, lightweight risk analysis solutions
* Aligned with NIST and OWASP principles of risk-based security practices

**9. Market Trends & Acceptance**

* Cybersecurity market to exceed $300B by 2027
* Rise of AI-powered vulnerability detection and triage tools
* Lightweight, frontend-only tools gaining popularity due to ease of deployment
* Increasing adoption of vulnerability education tools in academia and corporate training

**10. Use Case (Industry/Company Fit)**

* **Education:** Teaching cybersecurity concepts to students
* **Startups:** Providing affordable vulnerability triage support
* **Security Teams:** Triage support for junior analysts or interns
* **Research:** Testing and prototyping GNN-based vulnerability classifiers

**11. Future Improvements**

* Integrate Graph Neural Network (GNN)-based backend for deeper, structural analysis
* Connect with public vulnerability databases (e.g., NVD, CVE APIs)
* Add a CLI version and backend services with FastAPI or Flask
* Expand keyword logic to include NLP-based context recognition

**12. Conclusion & Learning**

This project highlights how simple technologies can offer meaningful contributions to cybersecurity awareness and education. The Zero-Day Risk Analyzer lays a solid foundation for future development using AI models like GNNs. Throughout the process, knowledge of frontend development, risk-based thinking, and security analysis was strengthened.

**13. References**

* NIST Cybersecurity Framework
* CVE Details (<https://www.cvedetails.com/>)
* MITRE ATT&CK Framework
* OWASP Top Ten
* Google Project Zero Blog
* Microsoft Security Blog