

**PROJECT (MBACS-SIV-1) REPORT ON**

**“Cloud-Based Vulnerability Detection System with AI Integration”**

**Submitted To**

**School of Management Studies, National Forensic Sciences University**

**MASTER OF BUSINESS ADMINISTRATION**

**In**

**CYBER SECURITY MANAGEMENT**

**Submitted**

**Ansh Modi**

**012300400011002007**

**Under the Supervision of Dr.Siddharth Dabhade**

**National Forensic Sciences University, Gandhinagar Campus, Gandhinagar – 382009, Gujarat, India.**

**STUDENTS DECLARATION**

This is to certify that I, **Ansh Modi,** have completed the Project titled "**Cloud-Based Vulnerability Detection System with AI Integration**" under the guidance of **Dr.Siddharth Dabhade** in partial fulfilment of the requirement for the award of Master of Business Administration in Cyber Security Management at National Forensic Sciences University – School Management Studies.

We hereby declare that this project is an original piece of work and has not been submitted earlier elsewhere.

**Date:** 08/05/2025

**Place:** NFSU, Gandhinagar

**CERTIFICATE FROM THE FACULTY GUIDE**

This is to certify that the Internship/Project "**Cloud-Based Vulnerability Detection System with AI Integration**" is an academic work done by **Ansh Modi** submitted in partial fulfilment of the requirement for the award of the degree of Master of Business Administration in Cyber Security Management at National Forensic Sciences University – School of Management Studies, under my guidance and direction.

To the best of my knowledge and belief, the data and information presented by him/her in the project has not been submitted earlier.

**Name of the Faculty Guide: Dr.Siddharth Dabhade**

**Designation: Assistant Professor**

**Faculty Guide Signature:**

**Preface**

This report is the culmination of my academic and practical experience in the discipline of Cybersecurity and Artificial Intelligence, created through the project entitled "AI-Powered Cloud-Based Vulnerability Scanner." The primary aim of this project was to create a lightweight, smart, and accessible platform for detecting, analyzing, and remediating web application vulnerabilities using both conventional tools and advanced AI integration.

The impetus behind this project came from the increased call for easier cybersecurity tools that may be accessed both by experts and non-professional users. During an era when web applications have become a popular target of cyber-attacks, there has never been a greater need for intelligent, cloud-based, and responsive scan tools. This project seeks to cover that void by combining established tools such as OWASP ZAP with React.js, Flask, AI-powered remediation software, and cloud deployment platforms such as Vercel and DigitalOcean.

The project was implemented over three phases: a CLI-based scanner, GUI-based web app, and full-responsive, AI-enabled frontend built with React. Each phase has been thoroughly documented and showcased to emphasize both learning achievements and the technical hurdles encountered.

This preface is done in the hope that this report can not only be a project submission but also an informative guide for future students, developers, and cybersecurity enthusiasts to implement artificial intelligence with security automation.

I do hope that this endeavor is found to be helpful and inspiring for readers and contributes positively to the academic and technical community.

**Acknowledgment**

I would like to thank all those who helped and advised me in the process of working on this project, "AI-Powered Cloud-Based Vulnerability Scanner." The project is an amalgamation of my theoretical learning and real-world industry exposure, and it would not have been feasible without the efforts of various individuals and organizations.

Above all, I would like to express my sincere gratitude to my project guide and faculty mentor, Dr.Siddharth Dabhade for their support, encouragement, and technical guidance throughout the course of this project. Their constructive criticism and technical expertise assisted in refining the objectives of the project and improving the overall quality of the final output.

I am also deeply grateful to the members of the teaching staff of the Department of School of Management, National Forensic Sciences University for giving me a rich intellectual environment and access to supporting resources that enriched my learning experience.

I also want to thank the authors and owners of open-source software and platforms like OWASP ZAP, Flask, React.js, Tailwind CSS, Vercel, and Google Gemini API, without whom it would not have been possible to develop this project.

Finally, I am highly indebted to my family and friends for their unconditional support, patience, and motivation throughout the entire period of this project.

This recognition is a token compared to the great support that I have had, and I am still genuinely indebted to all those who, directly or indirectly, helped the success of this venture.

**Executive Summary**

The project, "AI-Powered Cloud-Based Vulnerability Scanner," is an end-to-end cybersecurity solution for automating the process of detection and remediation of web application vulnerabilities through artificial intelligence and cloud computing. The report outlines the conceptualization, implementation, and deployment of a multi-stage system consistent with existing industry practices in web security and AI-based threat management.

As more and more web applications come under the crosshairs of cyber threats, security tools often lag behind when it comes to usability, convenience, and wisdom. This project tackles all that by pairing trusted scanning technologies (e.g., OWASP ZAP) with an easy-to-use interface implemented through React.js, a compact Flask backend, and embedded AI capabilities driven through Google Gemini API to provide context-aware remediation suggestions.

The system is engineered in three phases of development:

* **CLI-Based Scanner:** Command-line interface Python and OWASP ZAP APIs for rapid vulnerability scanning.
* **GUI Web Application:** A web-based interface with HTML, CSS, JavaScript, and Flask, augmented with WHOIS domain intelligence.
* **React + AI Integration:** Responsive, modular, full frontend engineered using React.js and Tailwind CSS, with more interactivity, visualization of scans, and AI-provided guidance.

Core features include:

* Automated web vulnerability scanning
* WHOIS domain search
* Remediation recommendations based on AI
* Structured scan reports and real-time terminal-style output
* Cloud deployment of frontend (Vercel) and backend (DigitalOcean/Oracle Cloud)

The report is organized into several chapters, starting with the project objectives, system analysis, and methodology, and ending with testing results, future scope, and references. Diagram placeholders and screenshots are included throughout for architecture visualization, workflows, and UI.

This project is an example of the confluence of university research, practitioner reality, and innovative application of AI in cybersecurity, providing a prototype that can be scaled up to become a commercial-ready SaaS product for proactive threat detection and management.

# **TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Title** | | **Page no.** |
| STUDENTS’ DECLARATION | | **ii** |
| CERTIFICATE FROM THE FACULTY GUIDE | | **iii** |
| PREFACE | | **iv** |
| ACKNOWLEDGMENT | | **v** |
| EXECUTIVE SUMMARY | | **vi** |
| TABLE OF CONTENTS | | **viii** |
| LIST OF TABLES | | **ix** |
| LIST OF FIGURES | | **x** |
| **1** | INTRODUCTION | **1** |
| **2** | OBJECTIVE | **5** |
| **3** | LITERATURE SURVEY | **9** |
| **4** | SYSTEM ANALYSIS | **13** |
| **5** | METHODOLOGY | **17** |
| **6** | SYSTEM DESIGN | **21** |
| **7** | IMPLEMENTATION | **25** |
| **8** | RESULTS & DISCUSSION | **33** |
| **9** | FUTURE SCOPE | **37** |
| **10** | CONCLUSION | **40** |
| **11** | REFERENCES | **43** |

# **LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Title** | **Page No.** |
|  | Table 3.1: Identify Gaps | 13 |
|  | Table 4.1: Feasibility Study | 16 |
|  | Table 5.1: Tools, Libraries & APIs | 20 |
|  | Table 5.2: SDLC Mapping | 21 |
|  | Table 6.1: Database Design | 23 |
|  | Table 7.1: Key Challenges & Solution | 32 |
|  | Table 7.2: Tools & Technologies use | 32 |
|  | Table 8.1: Functional Result | 34 |
|  | Table 8.2: Performance Metrics | 35 |
|  | Table 8.3: Comparative Analysis | 37 |

# **LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Title** | **Page No.** |
|  | Fig 1: Flow Chart | 25 |
|  | Fig 2: Output of ZAP Scanning | 27 |
|  | Fig 3: Scanning Code (ScannerForm.js) | 28 |
|  | Fig 4: AI Remediation Step (Gemini AI) | 29 |
|  | Fig 5: AI Prompt (Gemini AI) (app.py) | 29 |
|  | Fig 6: WHOIS Scan Result | 30 |
|  | Fig 8: Gantt Chart | 33 |
|  | Fig 9: Performance Metrics for Vulnerability Scan, AI Scan & WHOIS Scan | 35 |
|  | Fig 10: Backend Latency | 35 |
|  | Fig 11: Page Load Metrics | 36 |