MIRA BHAYANDAR MUNICIPAL CORPORATION

Open Land Tax

Web Application

Report Release Date	February 12, 2026
Type of Audit	Application Security Assessment
Type of Audit Report	First Audit Report
Period	February 03, 2026 to February 05, 2026

Document Control

Document Preparation	
Document Title	VAPT Report of Open Land Tax Web Application
Document ID	A3S/MBMC/ Open Land Tax /2526/00012
Document Version	1.0
Prepared by	Jasmeet Singh
Reviewed by	Sagar Gupta
Approved by	Sagar Gupta
Released by	Jasmeet Singh
Release date	February 12, 2026

Document Change History		
Version	**Date**	**Remarks / Reason of change**
1.0	February 12, 2026	New Report

Document Distribution List			
Name	**Organization**	**Designation**	**Email Id**
Rajkumar Gharat	Mira Bhayandar Municipal Corporation	System Manager	it@mbmc.gov.in

SAGAR GUPTA

Digitally signed by SAGAR GUPTA
Date: 2026.02.12 14:06:56 +05'30'

Contents

Table of Contents

1. Introduction

A3S Tech & Co. (A3S) was engaged by Mira Bhayandar Municipal Corporation to perform VAPT, for Open Land Tax Web Application. The report highlights gaps identified during the review and recommendations to remediate the gaps.

The objective of Web Application VAPT was to provide independent evaluation of the vulnerabilities in scope to fulfil the objectives of confidentiality, integrity, and availability and to perform controlled attack to assess the immunity level, to assess the overall level of security, discover weak links and provide recommendations and compliance status to vulnerable entities discovered. The report highlights gaps identified during the VAPT review, recommendations, risk ratings and impact of the vulnerabilities.

2. Engagement Scope

Below are the details of assets covered in the scope:

S. No.	Asset Description	Criticality of Asset	Internal IP Address	URL	Public IP Address	Location	Hash Value (in case of applications)	Version (in case of applications)	Other details such as make and model in case of network devices or security devices.
1.	Web Application – Open Land Tax	Not available	Not Available	devolt.mbmconline.in	Not Available	MUMBAI	Not available	Not available	Not Applicable

3. Details of the Auditing team

S. no.	Name	Designation	Email Id	Professional Qualifications/ Certifications	Whether the resource has been listed in the Snapshot information published on CERT-In's website (Yes/No)
1.	Jasmeet Singh	Senior IS Consultant	jasmeet@a3stech.co.in	CEH	Yes

4. Audit Activities and Timelines

The audit was conducted in the following phases:

S. no.	Audit Activity	Timeline
1.	Information Gathering	February 3, 2026
2.	Scanning	February 3, 2026
3.	Information Analysis	February 4, 2026
4.	Vulnerability Assessment	February 4, 2026
5.	Penetration Testing	February 4, 2026
6.	Revalidation Testing	NA

5. Audit Methodology and Criteria / Standard referred for audit

The Audit Approach and Methodology was a Risk based Audit Approach. In a risk-based audit approach, IS auditors are not just relying on risk; they also are also relying on internal and operational controls as well as knowledge of the organization and its business. The audit was conducted based on combination of tools and manual testing. The audit methodology and approach are based on global best practice framework such as OWASP Top 10 Vulnerabilities, OSSTMM, SANS 25, CIS benchmarks. These are globally accepted standard and a benchmark for IT security across a large number of organizations.

List of OWASP vulnerabilities (Web Application) is:

S. no	Attack Type	Description
1.	A1- Broken Access Control	Improperly configured or missing restrictions on authenticated users allow them to access unauthorized functionality or data, such as accessing other users' accounts, viewing sensitive documents, and modifying data and access rights
2.	A2- Cryptographic Failures	Applications and APIs that don't properly protect sensitive data such as financial data, usernames and passwords, or health information, could enable attackers to access such information to commit fraud or steal identities.
3.	A3- Injection	Injection flaws, such as SQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing unauthorized data
4.	A4- Insecure Design	Insecure design is a broad category representing different weaknesses, expressed as "missing or

S. no	Attack Type	Description
		ineffective control design". An insecure design cannot be fixed by a perfect implementation as by definition, needed security controls were never created to defend against specific attacks. One of the factors that contribute to insecure design is the lack of business risk profiling inherent in the software or system being developed, and thus the failure to determine what level of security design is required.
5.	A5- Security Misconfiguration	Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, and platform. All these settings should be defined, implemented, and maintained as many are not shipped with secure defaults. This includes keeping all software up to date, including all code libraries used by the application
6.	A6- Vulnerable and Outdated Components	Developers frequently don't know which open source and third-party components are in their applications, making it difficult to update components when new vulnerabilities are discovered. Attackers can exploit an insecure component to take over the server or steal sensitive data.
7.	A7- Identification and Authentication Failures	Application functions related to authentication and session management are often not implemented correctly, allowing attackers to compromise passwords, keys, session tokens, or exploit other implementation flaws to assume other users' identities
8.	A8- Software and Data	Software and data integrity failures relate to code and infrastructure that does not protect against integrity violations. An insecure CI/CD pipeline can introduce the potential for unauthorized access, malicious code,

S. no	Attack Type	Description
	Integrity Failures	or system compromise. Lastly, many applications now include auto-update functionality, where updates are downloaded without sufficient integrity verification and applied to the previously trusted application. Attackers could potentially upload their own updates to be distributed and run on all installations
9.	A9- Security Logging and Monitoring Failures	The time to detect a breach is frequently measured in weeks or months. Insufficient logging and ineffective integration with security incident response systems allow attackers to pivot to other systems and maintain persistent threats
10.	A10- Server-Side Request Forgery	SSRF flaws occur whenever a web application is fetching a remote resource without validating the user-supplied URL. It allows an attacker to coerce the application to send a crafted request to an unexpected destination, even when protected by a firewall, VPN, or another type of network access control list (ACL).

This document is an exception report highlighting the vulnerabilities and their compliance status.

Our review has been based on the assumption that the information provided to us was accurate and complete, as existing at the time of review, and that all relevant information, system access for review, and supporting documents, as asked for by A3S, were shared with us for the area that was subject of the review.

6. Tools/ Software used

S. no.	Name of Tool/Software used	Version of the tool /Software used	Open Source/Licensed
1.	Burp Suite	2025.12.5	Licensed

7. Executive Summary

The details of the vulnerabilities identified during the testing as mentioned as below:

S. No.	Affected Asset i.e. IP/URL/Application etc.	Observation/Vulnerability title	CVE/CWE	Severity	Recommendation	Reference	New or Repeat observation
1	devolt.mbmconline.in	Brute Force Attack – Improper Restriction of Authentication Attempts	CWE-307	**High**	To mitigate this vulnerability, it is recommended to Implement account lockout after 5–10 failed login attempts. Apply rate limiting on authentication endpoints. Introduce CAPTCHA after multiple failed attempts. Add progressive delay between login attempts. Monitor and log suspicious login behavior. Enforce strong password policies	https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/	New

2	devolt.mbmconline.in	Missing Rate Limiting on Authentication (Password) Functionality	CWE-307	**Medium**	It is recommended to implement the following controls. Apply rate limiting on login endpoints (e.g., 5–10 attempts per minute). Enforce temporary IP blocking after threshold breach. Introduce CAPTCHA after multiple failed attempts. Implement progressive delay (exponential backoff). Monitor and alert on abnormal login behavior.	https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/	New
3	devolt.mbmconline.in	Clear Text Password Transmission in Login Request	CWE-319	**Medium**	It is strongly recommended to: Enforce HTTPS (TLS 1.2 or higher) across the application. Avoid transmitting passwords in clear text. Implement secure encryption mechanisms for data in transit. Ensure passwords are hashed and salted on the server side.	https://owasp.org/Top10/A02_2021-Cryptographic_Failures/	New

					Disable login access over HTTP. Use secure authentication frameworks (OAuth, SSO, etc.)		
4	devolt.mbmconline.in	Missing Security Headers	CWE-693	**Medium**	It is recommended to configure and enable the required HTTP security headers at the web server or application level. At a minimum, implement the following: Content-Security-Policy: default-src 'self'; X-Content-Type-Options: nosniff Strict-Transport-Security: max-age=31536000; includeSubDomains Referrer-Policy: no-referrer Permissions-Policy: geolocation=(), camera=(), microphone=() X-Frame-Options: DENY	https://www.invicti.com/blog/web-security/missing-http-security-headers	New
5	devolt.mbmconline.in	Out of date (ASP.NET Version)	CWE-200	**Medium**	Upgrade the application to the latest supported and fully patched version of ASP.NET / .NET Framework (4.8.1). Apply all relevant Microsoft security updates.	https://cwe.mitre.org/data/definitions/200.html	New

					Remove or suppress the X-AspNet-Version and X-Powered-By headers from HTTP responses to prevent version disclosure.		
6	devolt.mbmconline.in	Out of date (jQuery UI Version)	CWE-200	**Medium**	Upgrade jQuery UI to the latest stable and supported version(1.14.2). Ensure that all unused or deprecated JavaScript libraries are removed, and regularly monitor third-party dependencies for security updates to reduce the risk of client-side attacks.	https://cwe.mitre.org/data/definitions/200.html	New
7	devolt.mbmconline.in	Out of Date (Bootstrap Version)	CWE-200	**Medium**	Upgrade Bootstrap to the latest stable and supported version (Bootstrap 5.3.8). Remove deprecated Bootstrap 3 components and ensure all dependent libraries are compatible with the	https://owasp.org/Top10/2021/A06_2021-Vulnerable_and_Outdated_Components/	New

					updated version. Regularly update third-party libraries to mitigate known vulnerabilities.		
8	devolt.mbmconline.in	Weak Ciphers	CWE-326	**Medium**	Disable all weak and deprecated cipher suites, including CBC-based, RSA key exchange, and 3DES ciphers. Configure the server to support only strong, modern cipher suites, such as AES-GCM with ECDHE key exchange, and enforce TLS 1.2 and TLS 1.3. Regularly review SSL/TLS configurations using trusted tools to ensure continued compliance with security best practices.	https://owasp.org/Top10/A02_2021-Cryptographic_Failures/	New
9	devolt.mbmconline.in	Captcha Not Implemented	CWE-326	**Medium**	Implement a CAPTCHA mechanism (such as reCAPTCHA or equivalent) on the login page to distinguish between human users	https://cwe.mitre.org/data/definitions/307.html	New

					and automated requests. Additionally, enforce account lockout, rate limiting, and monitoring of failed login attempts to further mitigate brute-force and automated attacks.		
10	devolt.mbmconline.in	Improper Input Validation	CWE-20	**Medium**	Enforce strict server-side validation for Email ID fields (allow only valid email formats). Sanitize and encode all user inputs before processing or rendering. Implement output encoding based on context (HTML, JavaScript). Use secure frameworks or libraries that provide built-in XSS protection. Enable a strong Content-Security-Policy (CSP) header to mitigate script execution.	https://cwe.mitre.org/data/definitions/20.html	New

11	devolt.mbmconline.in	Improper Input Validation (Missing Input Length Validation)	CWE-20	**Medium**	Enforce server-side length validation for all numeric fields: Mobile Number: exactly 10 digits, PAN Number: 10-character alphanumeric format, Pincode: exactly 6 digits. Implement both client-side and server-side validation. Reject inputs exceeding defined length with proper error messages. Apply database-level constraints where applicable. Perform centralized input validation across the application.	https://owasp.org/www-project-top-ten/2021/A04_2021-Insecure_Design/	New
12	devolt.mbmconline.in	Version Disclosure (jQuery UI)	CWE-200	**Low**	It is recommended to remove or obfuscate client-side library version information and upgrade jQuery UI to the latest supported version(1.14.2). Additionally, ensure that	https://owasp.org/www-community/attacks/Information_Disclosure	New

					only required libraries are exposed and regularly review third-party components for security updates.		
13	devolt.mbmconline.in	Version Disclosure (Bootstrap)	CWE-200	**Low**	Remove or obfuscate version comments from client-side files where feasible. Upgrade Bootstrap to the latest supported version (v5.3.8) and ensure outdated components are removed. Regularly review and update third-party libraries to minimize information disclosure and reduce the attack surface.	https://owasp.org/www-community/attacks/Information_Disclosure	New
14	devolt.mbmconline.in	Version Disclosure (ASP.NET)	CWE-200	**Low**	Disable or suppress version disclosure headers by removing X-AspNet-Version and X-Powered-By from HTTP responses. Ensure the application runs on a fully patched and	https://owasp.org/www-community/attacks/Information_Disclosure	New

					supported version of ASP.NET and regularly apply Microsoft security updates.		
15	devolt.mbmconline.in	Server Disclosure	CWE-200	**Low**	It is recommended to remove or obfuscate server identification headers by configuring the web server to suppress detailed version information. Implement secure server hardening practices, ensure unnecessary headers are disabled, and regularly review HTTP response headers to minimize information disclosure.	https://learn.microsoft.com/en-us/archive/blogs/varunm/remove-unwanted-http-response-headers https://www.acunetix.com/vulnerabilities/web/version-disclosure-iis/	New
16	devolt.mbmconline.in	Stack Trace Disclosure (ASP.NET)	CWE-209	**Low**	Disable detailed error messages and stack trace disclosure in production by configuring customErrors in web.config. Ensure that detailed exceptions are logged	https://www.acunetix.com/vulnerabilities/web/stack-trace-disclosure-asp-net/	New

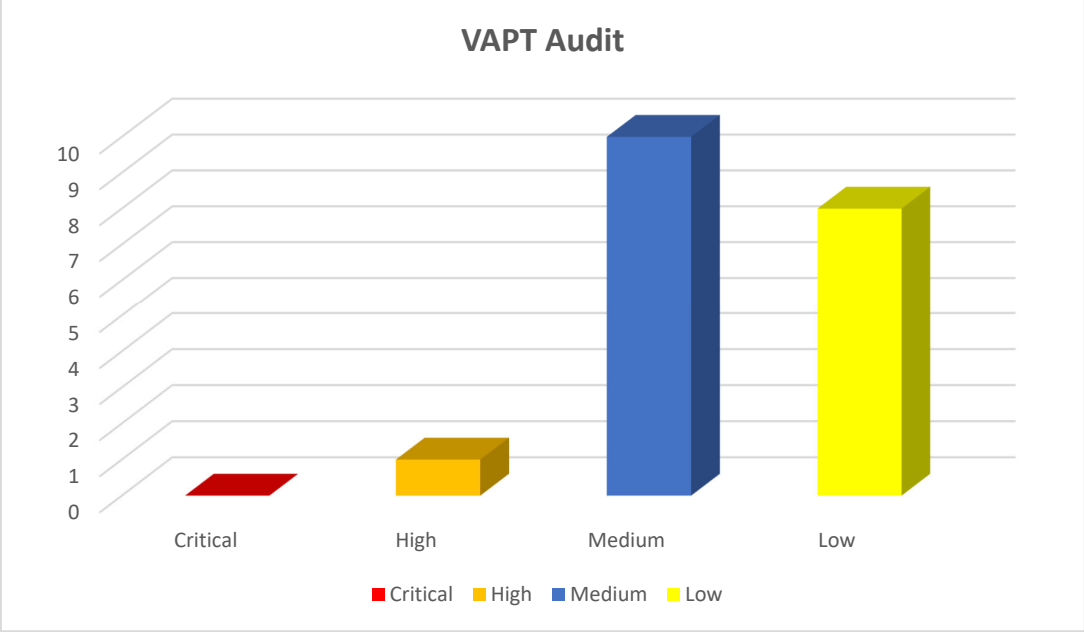
					internally while presenting generic error messages to users. Regularly review error-handling configurations to prevent leakage of sensitive debugging information.		
17	devolt.mbmconline.in	Cookie not marked as Secure site attribute	CWE-614	**Low**	Configure the application to set the Secure flag on all sensitive cookies to ensure they are transmitted only over HTTPS connections. Additionally, enforce HTTPS across the application and review cookie attributes such as HttpOnly and SameSite for improved session security.	https://cwe.mitre.org/data/definitions/614.html	New
18	devolt.mbmconline.in	Clickjacking	CWE-1021 CWE-693	**Low**	Implement the X-Frame-Options header with the value DENY or SAMEORIGIN. Additionally, configure a Content-Security-Policy with the frame-	https://cwe.mitre.org/data/definitions/693.html https://cwe.mitre.org/data/definitions/1021.html	New

					ancestors directive to explicitly restrict which domains are allowed to embed the application. These controls will prevent unauthorized framing and mitigate clickjacking attacks.		
19	devolt.mbmconline.in	Internal Server Error	CWE-209	**Low**	Configure the application to display generic, user-friendly error messages instead of detailed system exceptions. Disable detailed error messages in production by setting customErrors to On in the ASP.NET configuration. Additionally, implement proper server-side input validation and centralized exception handling to prevent unhandled errors from being exposed to end users.	https://cwe.mitre.org/data/definitions/209.html	New

Tabular Representation of the vulnerabilities:

Risk Rating	Count of Observations
Critical	-
High	1
Medium	10
Low	8

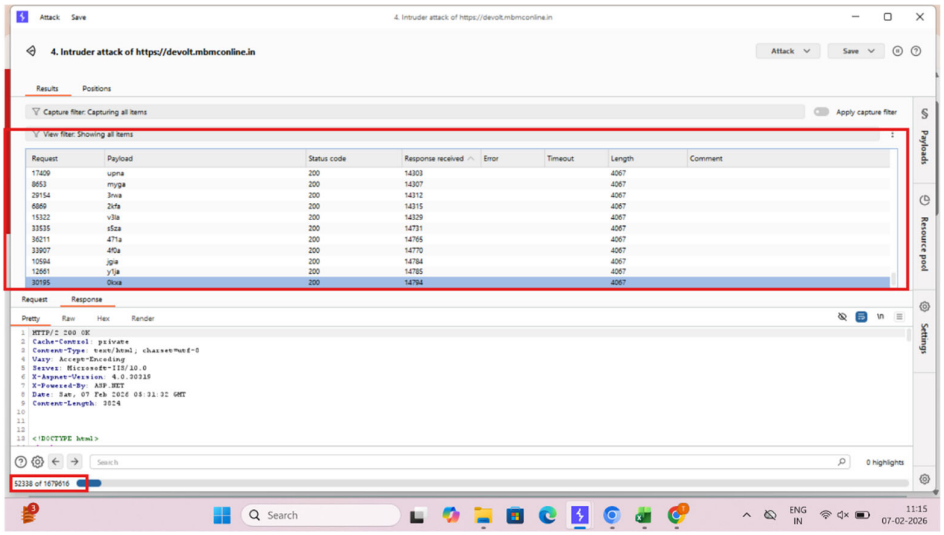
Graphical Representation of vulnerabilities



8. Detailed Observations

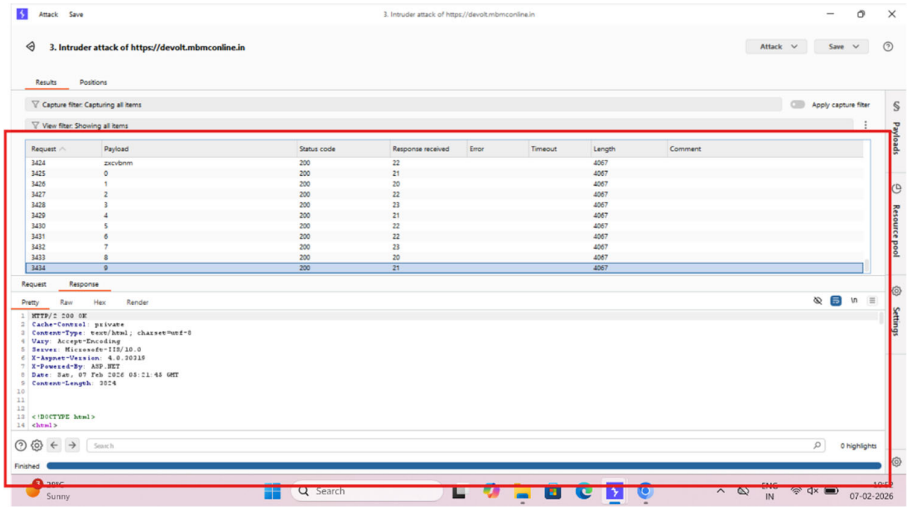
1. Brute Force Attack – Improper Restriction of Authentication Attempts

Vulnerability Title	*Affected URLs/IP*
Brute Force Attack – Improper Restriction of Authentication Attempts	devolt.mbmconline.in

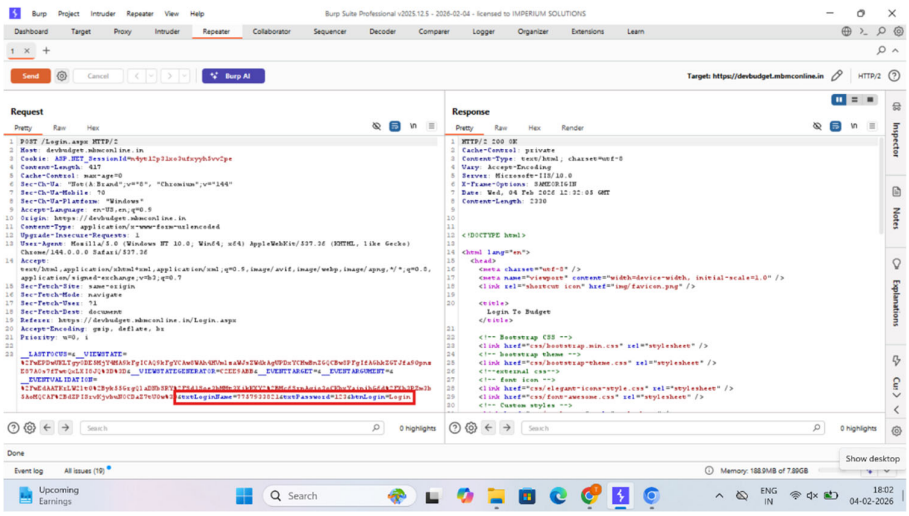
Detailed Observation	Multiple login requests were sent with different payloads. The application returned HTTP 200 OK responses for all attempts. No account lockout mechanism was triggered. No CAPTCHA challenge was enforced. No request rate-limiting was observed. Response length and behavior remained consistent across attempts. This confirms that the application allows unlimited authentication attempts.
Vulnerability Reference (CWE/CVE)	CWE-307
Severity	**High**
Recommendation	To mitigate this vulnerability, it is recommended to Implement account lockout after 5–10 failed login attempts. Apply rate limiting on authentication endpoints. Introduce CAPTCHA after multiple failed attempts. Add progressive delay between login attempts. Monitor and log suspicious login behavior. Enforce strong password policies
Reference	https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/
New/ Repeat Observation	New Observation
References to evidences / Proof of Concept (POCs)	

2. Missing Rate Limiting on Authentication (Password) Functionality

Vulnerability Title	***Affected URLs/IP***

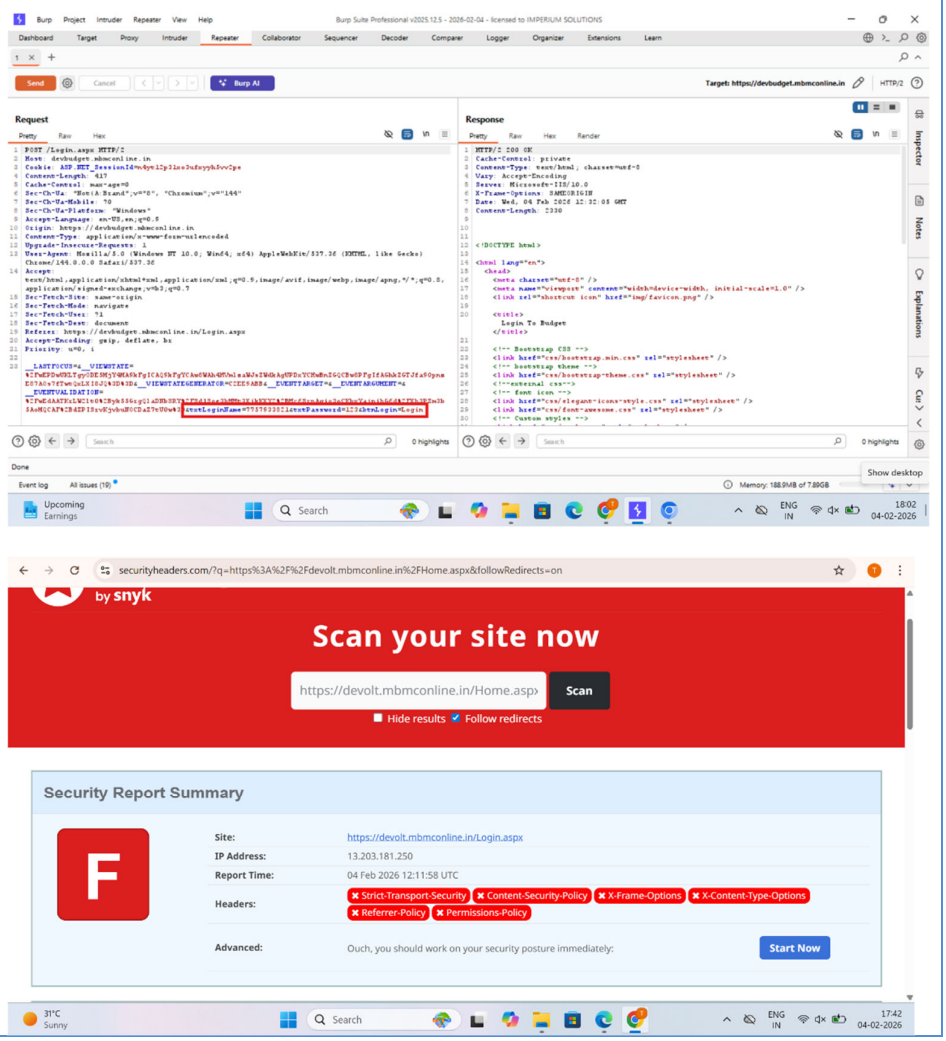
Missing Rate Limiting on Authentication (Password) Functionality	devolt.mbmconline.in
Detailed Observation	Rate limiting is a security control used to restrict the number of requests a user or IP address can make within a specific time period. During the security assessment, it was observed that the application does not enforce rate limiting on the login/password endpoint, allowing an attacker to send multiple authentication requests continuously.
Vulnerability Reference (CWE/CVE)	CWE-307
Severity	**Medium**
Recommendation	It is recommended to implement the following controls. Apply rate limiting on login endpoints (e.g., 5–10 attempts per minute). Enforce temporary IP blocking after threshold breach. Introduce CAPTCHA after multiple failed attempts. Implement progressive delay (exponential backoff). Monitor and alert on abnormal login behavior.
Reference	https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/
New/ Repeat Observation	New Observation
References to evidences / Proof of Concept (POCs)	

3. Clear Text Password Transmission in Login Request

Vulnerability Title	***Affected URLs/IP***
Clear Text Password Transmission in Login Request	devolt.mbmconline.in
Detailed Observation	Clear text password transmission occurs when user credentials are sent from the client to the server without proper encryption or protection. This allows attackers to view sensitive information such as usernames and passwords in readable format. During the security assessment, it was observed that the application transmits the password parameter in plain readable text within the HTTP request.
Vulnerability Reference (CWE/CVE)	CWE-319
Severity	**Medium**
Recommendation	It is strongly recommended to: Enforce HTTPS (TLS 1.2 or higher) across the application. Avoid transmitting passwords in clear text. Implement secure encryption mechanisms for data in transit. Ensure passwords are hashed and salted on the server side. Disable login access over HTTP. Use secure authentication frameworks (OAuth, SSO, etc.)
Reference	https://owasp.org/Top10/A02_2021-Cryptographic_Failures/
New/ Repeat Observation	New Observation
References to evidences / Proof of Concept (POCs)	

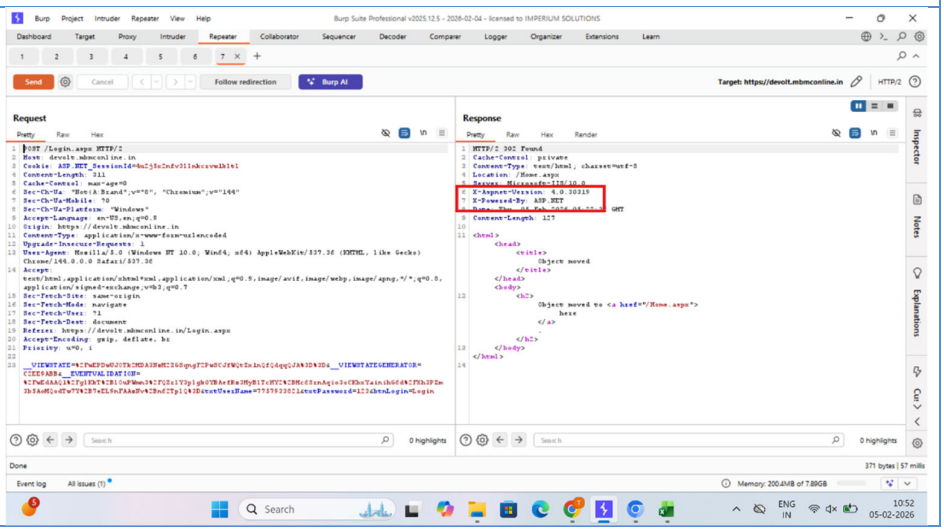
4. Missing Security Headers

Vulnerability Title	*Affected URLs/IP*
Missing Security Headers	devolt.mbmconline.in
Detailed Observation	The application does not implement one or more recommended HTTP security headers. Security headers help browsers enforce security controls that reduce the risk of common web attacks such as Cross-Site Scripting (XSS), Clickjacking, MIME-type sniffing, and information disclosure. Absence of these headers weakens the overall security posture of the application.
Vulnerability Reference (CWE/CVE)	CWE-693
Severity	**Medium**
Recommendation	It is recommended to configure and enable the required HTTP security headers at the web server or application level. At a minimum, implement the following: Content-Security-Policy: default-src 'self'; X-Content-Type-Options: nosniff Strict-Transport-Security: max-age=31536000; includeSubDomains Referrer-Policy: no-referrer Permissions-Policy: geolocation=(), camera=(), microphone=() X-Frame-Options: DENY
Reference	https://www.invicti.com/blog/web-security/missing-http-security-headers
New/ Repeat Observation	New Observation

References to evidences / Proof of Concept (POCs)	

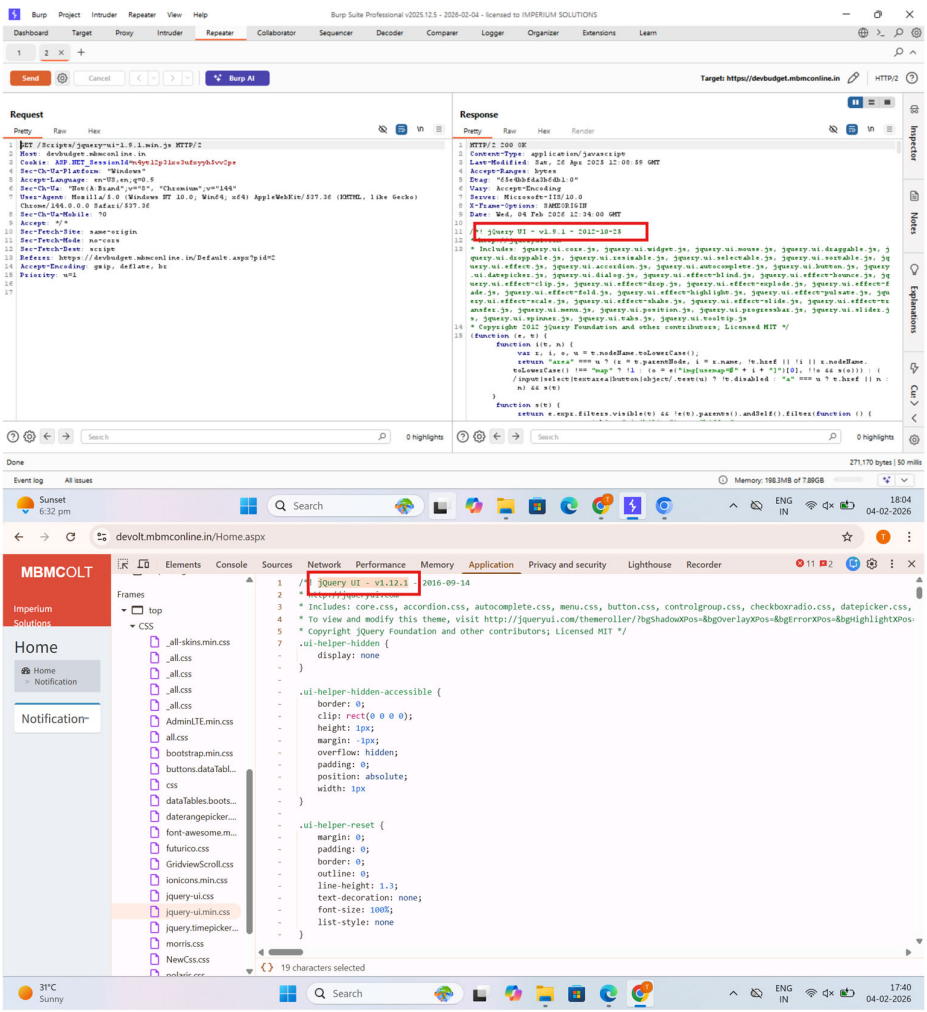
5. Out of Date (ASP.NET Version)

Vulnerability Title	***Affected URLs/IP***
Out of date (ASP.NET Version)	devolt.mbmconline.in
Detailed Observation	The application discloses the ASP.NET framework version through HTTP response headers. The disclosed version is outdated and may contain known security vulnerabilities. This information disclosure can assist attackers in identifying and exploiting framework-specific weaknesses.
Vulnerability Reference (CWE/CVE)	CWE-200
Severity	**Medium**

Recommendation	Upgrade the application to the latest supported and fully patched version of ASP.NET / .NET Framework (4.8.1). Apply all relevant Microsoft security updates. Remove or suppress the X-AspNet-Version and X-Powered-By headers from HTTP responses to prevent version disclosure.
Reference	https://cwe.mitre.org/data/definitions/200.html
New/ Repeat Observation	New Observation
References to evidences / Proof of Concept (POCs)	

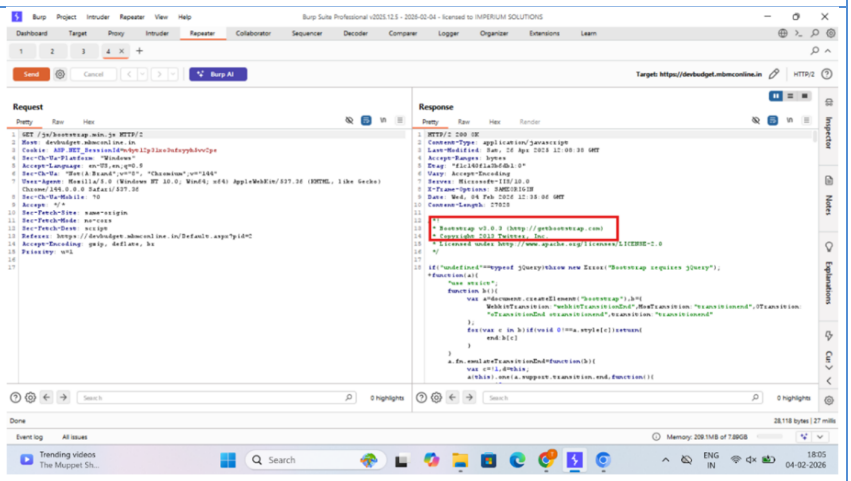
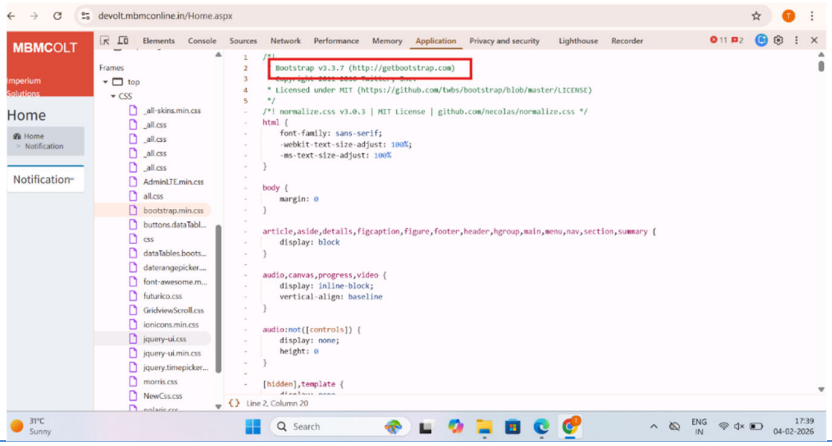
6. Out of Date (jQuery UI Version)

Vulnerability Title	***Affected URLs/IP***
Out of date (jQuery Version)	devolt.mbmconline.in
Detailed Observation	It was observed that the application is using an outdated version of jQuery UI (v1.12.1), as identified from the JavaScript file response. This version is deprecated and contains publicly known security vulnerabilities that may be exploited by an attacker. Using outdated client-side libraries increases the application's exposure to security risks.
Vulnerability Reference (CWE/CVE)	CWE-200
Severity	**Medium**
Recommendation	Upgrade jQuery UI to the latest stable and supported version(1.14.2). Ensure that all unused or deprecated JavaScript libraries are removed, and regularly monitor third-party dependencies for security updates to reduce the risk of client-side attacks.

Reference	https://cwe.mitre.org/data/definitions/200.html
New/ Repeat Observation	New Observation
References to evidences / Proof of Concept (POCs)	

7. Out of Date (Bootstrap Version)

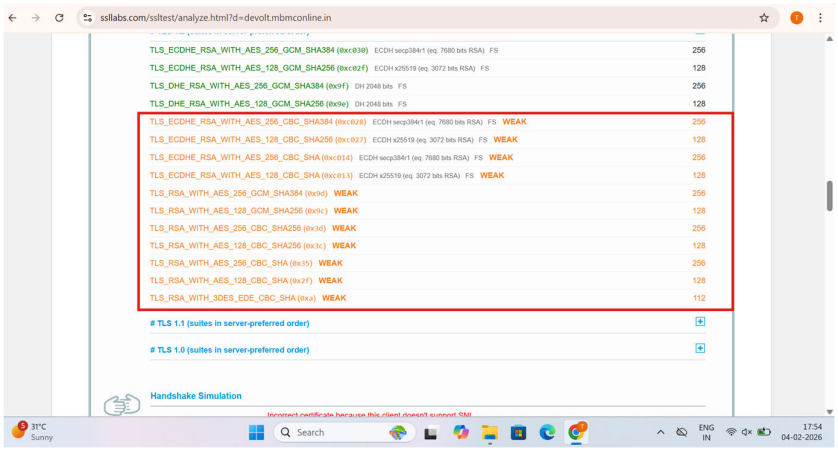
Vulnerability Title	***Affected URLs/IP***
Out of Date (Bootstrap Version)	devolt.mbmconline.in
Detailed Observation	The application was found to be using Bootstrap version 3.3.7. Bootstrap 3.3.7 is an outdated version and is no longer actively maintained. Older versions of Bootstrap may contain

	known security vulnerabilities and compatibility issues, increasing the risk of client-side attacks.
Vulnerability Reference (CWE/CVE)	CWE-200
Severity	**Medium**
Recommendation	Upgrade Bootstrap to the latest stable and supported version (Bootstrap 5.3.8). Remove deprecated Bootstrap 3 components and ensure all dependent libraries are compatible with the updated version. Regularly update third-party libraries to mitigate known vulnerabilities.
Reference	https://owasp.org/Top10/2021/A06_2021-Vulnerable_and_Outdated_Components/
New/ Repeat Observation	New Observation
References to evidences / Proof of Concept (POCs)	 

8. Weak Ciphers

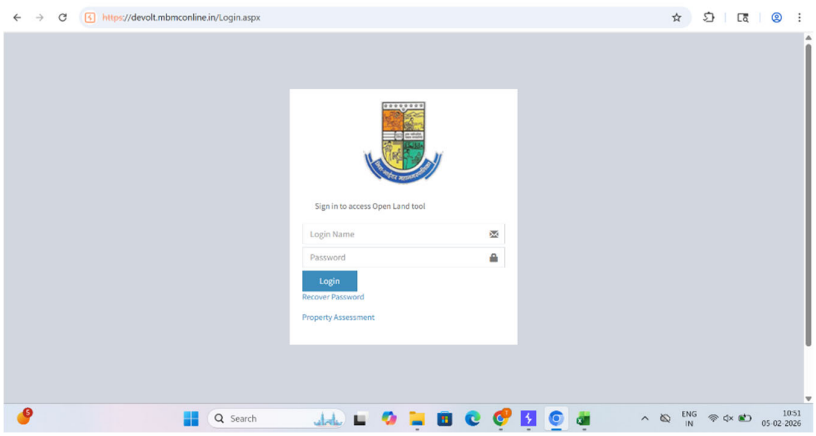
Vulnerability Title	***Affected URLs/IP***

Weak Ciphers	devolt.mbmconline.in
Detailed Observation	It was observed that the server supports weak cipher suites, including CBC-mode ciphers, RSA key exchange–based ciphers, and 3DES. These cipher suites are flagged as WEAK by SSL Labs and are considered cryptographically insecure due to known weaknesses and lack of forward secrecy in some cases. An attacker may exploit these weaknesses to compromise the confidentiality of encrypted communications. The presence of these weak cipher suites may allow an attacker to downgrade the encryption strength and potentially compromise the confidentiality and integrity of data transmitted between the client and the server.
Vulnerability Reference (CWE/CVE)	CWE-326
Severity	**Medium**
Recommendation	Disable all weak and deprecated cipher suites, including CBC-based, RSA key exchange, and 3DES ciphers. Configure the server to support only strong, modern cipher suites, such as AES-GCM with ECDHE key exchange, and enforce TLS 1.2 and TLS 1.3. Regularly review SSL/TLS configurations using trusted tools to ensure continued compliance with security best practices. Recommended Cipher Examples: TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256
Reference	https://owasp.org/Top10/A02_2021-Cryptographic_Failures/
New/ Repeat Observation	New Observation

References to evidences / Proof of Concept (POCs)	

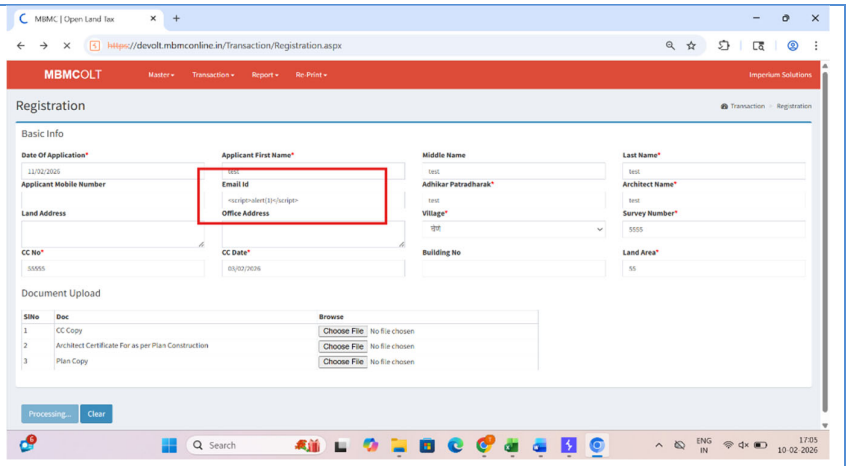
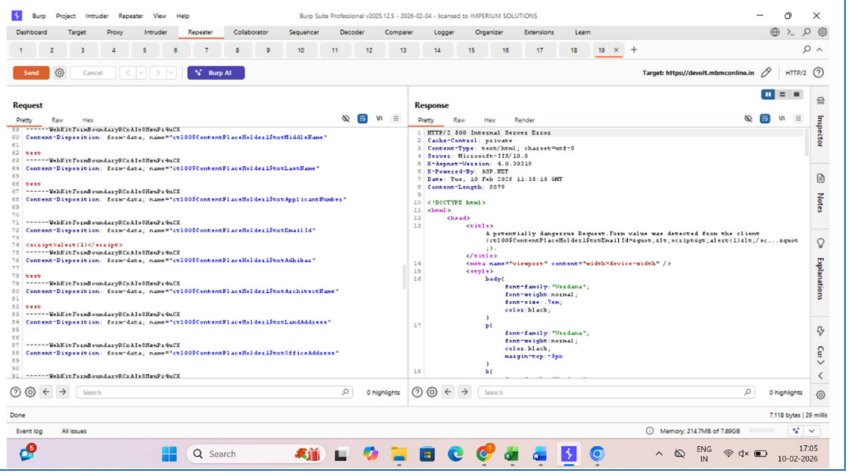
9. Captcha Not Implemented

Vulnerability Title	***Affected URLs/IP***
Captcha Not Implemented	devolt.mbmconline.in
Detailed Observation	The application login page does not implement any CAPTCHA or human verification mechanism. The absence of CAPTCHA allows automated scripts or bots to perform repeated authentication attempts, increasing the risk of brute-force attacks, credential stuffing, and automated abuse of the login functionality.
Vulnerability Reference (CWE/CVE)	CWE-326
Severity	**Medium**
Recommendation	Implement a CAPTCHA mechanism (such as reCAPTCHA or equivalent) on the login page to distinguish between human users and automated requests. Additionally, enforce account lockout, rate limiting, and monitoring of failed login attempts to further mitigate brute-force and automated attacks.
Reference	https://cwe.mitre.org/data/definitions/307.html
New/ Repeat Observation	New Observation

References to evidences / Proof of Concept (POCs)	

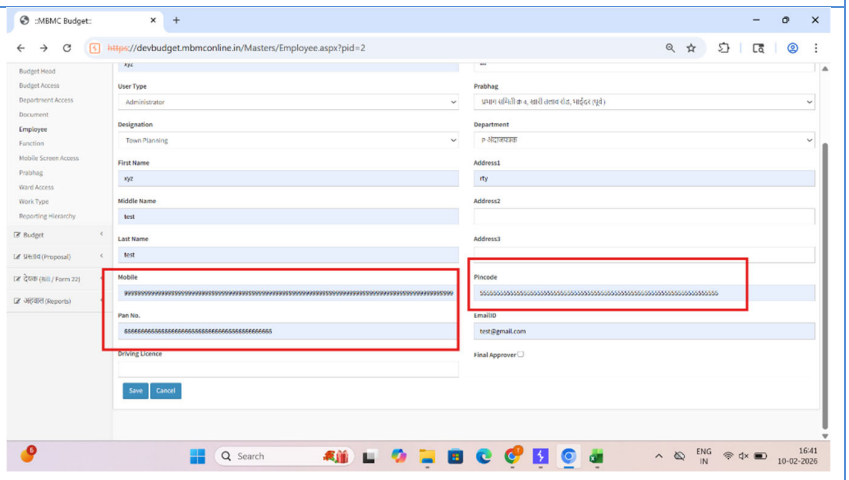
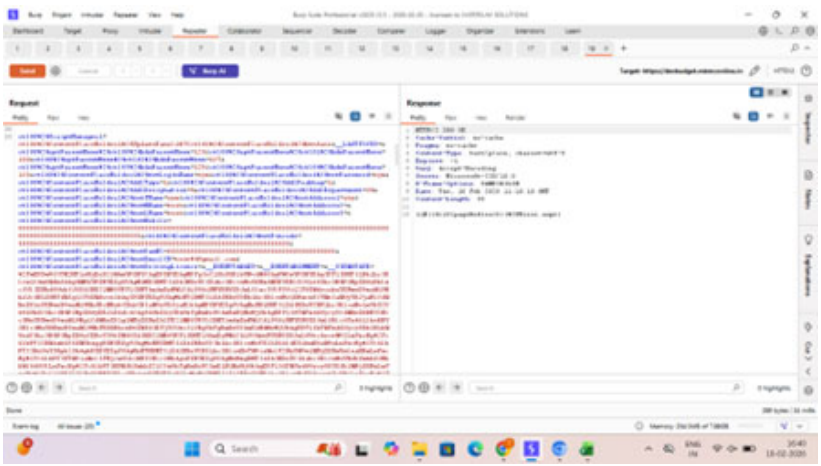
10. Improper Input Validation

Vulnerability Title	***Affected URLs/IP***
Improper Input Validation	devolt.mbmconline.in
Detailed Observation	The application does not properly validate or sanitize user-supplied input in the Email ID field. As a result, malicious scripts can be injected and potentially executed in the user's browser. During testing, a JavaScript payload was accepted as valid input, indicating insufficient server-side and client-side validation.
Vulnerability Reference (CWE/CVE)	CWE-20
Severity	**Medium**
Recommendation	Enforce strict server-side validation for Email ID fields (allow only valid email formats). Sanitize and encode all user inputs before processing or rendering. Implement output encoding based on context (HTML, JavaScript). Use secure frameworks or libraries that provide built-in XSS protection. Enable a strong Content-Security-Policy (CSP) header to mitigate script execution.
Reference	https://cwe.mitre.org/data/definitions/20.html
New/ Repeat Observation	New Observation

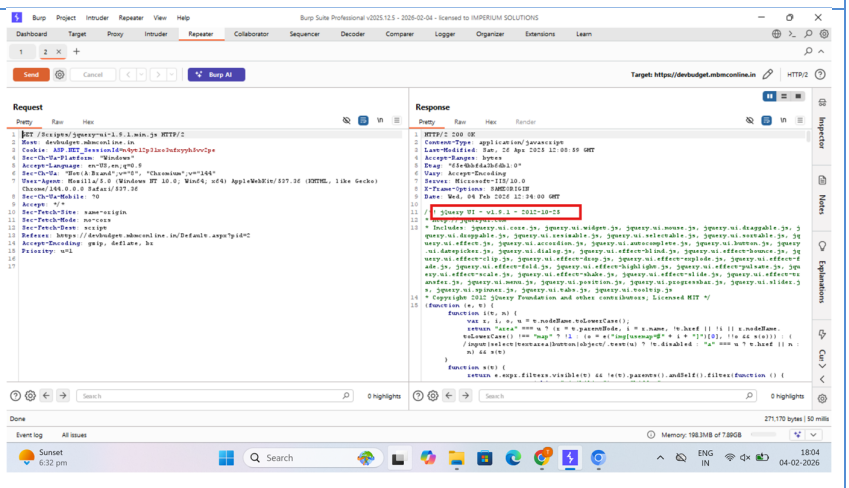
References to evidences / Proof of Concept (POCs)	 

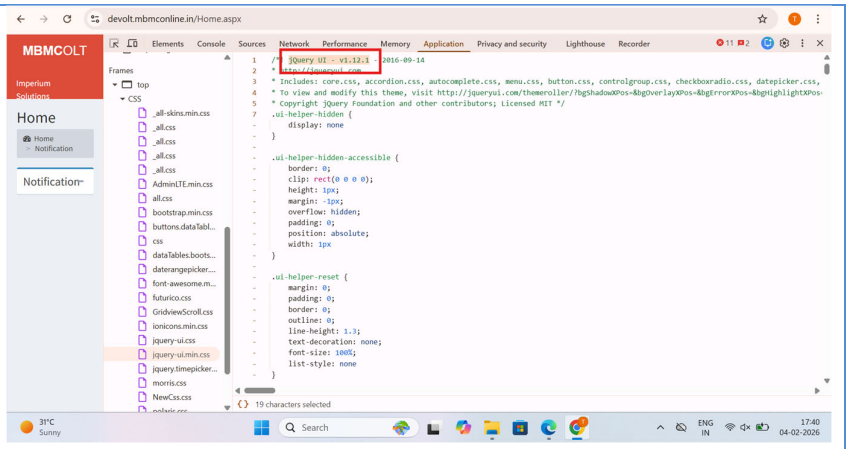
11. Improper Input Validation (Missing Input Length Validation)

Vulnerability Title	***Affected URLs/IP***
Improper Input Validation (Missing Input Length Validation)	devolt.mbmconline.in
Detailed Observation	The application does not enforce proper length and format validation on numeric input fields such as Mobile Number, PAN Number, and Pincode. These fields accept excessively long numeric values beyond their expected length, indicating missing server-side validation. Such improper validation may lead to data integrity issues, application logic bypass, and potential security risks.
Vulnerability Reference (CWE/CVE)	CWE-20

Severity	**Medium**
Recommendation	Enforce server-side length validation for all numeric fields: Mobile Number: exactly 10 digits, PAN Number: 10-character alphanumeric format, Pincode: exactly 6 digits. Implement both client-side and server-side validation. Reject inputs exceeding defined length with proper error messages. Apply database-level constraints where applicable. Perform centralized input validation across the application.
Reference	https://owasp.org/www-project-top-ten/2021/A04_2021-Insecure_Design/
New/ Repeat Observation	New Observation
References to evidences / Proof of Concept (POCs)	 

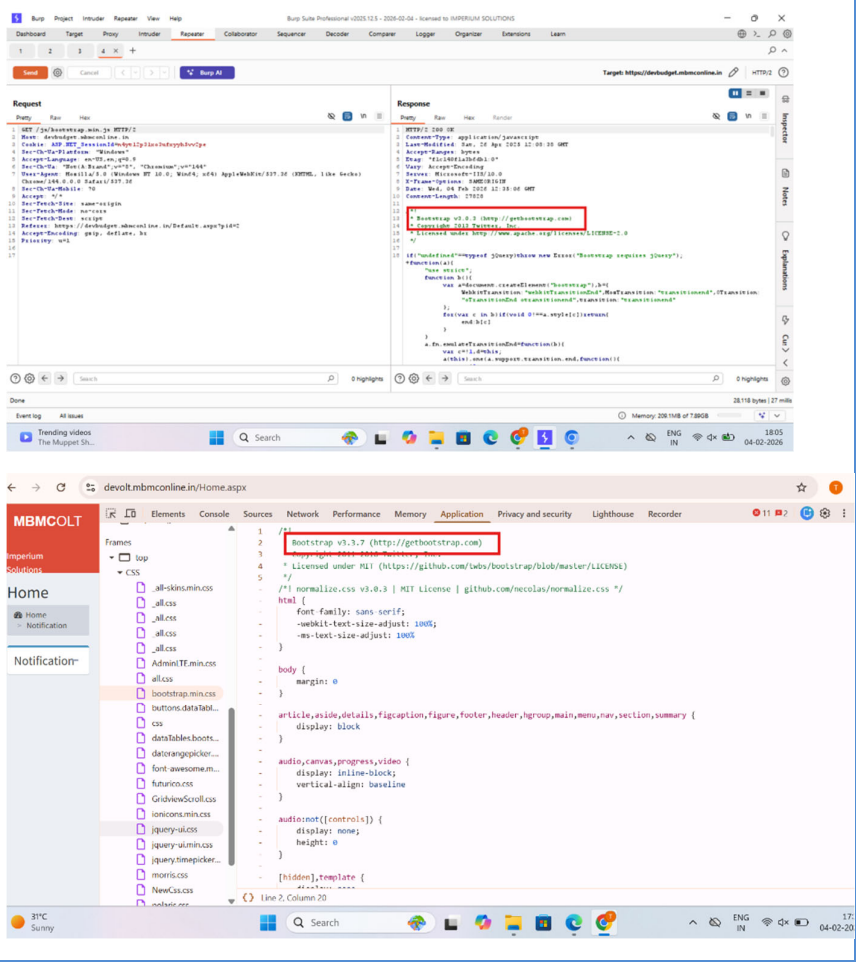
12. Version Disclosure (jQuery UI)

Vulnerability Title	***Affected URLs/IP***
Version Disclosure (jQuery UI)	devolt.mbmconline.in
Detailed Observation	It was observed that the application discloses the jQuery UI library version (v1.12.1) within client-side JavaScript files. Exposing library version details allows attackers to identify the exact component in use and correlate it with publicly known vulnerabilities, increasing the likelihood of targeted attacks.
Vulnerability Reference (CWE/CVE)	CWE-200
Severity	**Low**
Recommendation	It is recommended to remove or obfuscate client-side library version information and upgrade jQuery UI to the latest supported version(1.14.2). Additionally, ensure that only required libraries are exposed and regularly review third-party components for security updates.
Reference	https://owasp.org/www-community/attacks/Information_Disclosure
New/ Repeat Observation	New Observation
References to evidences / Proof of Concept (POCs)	



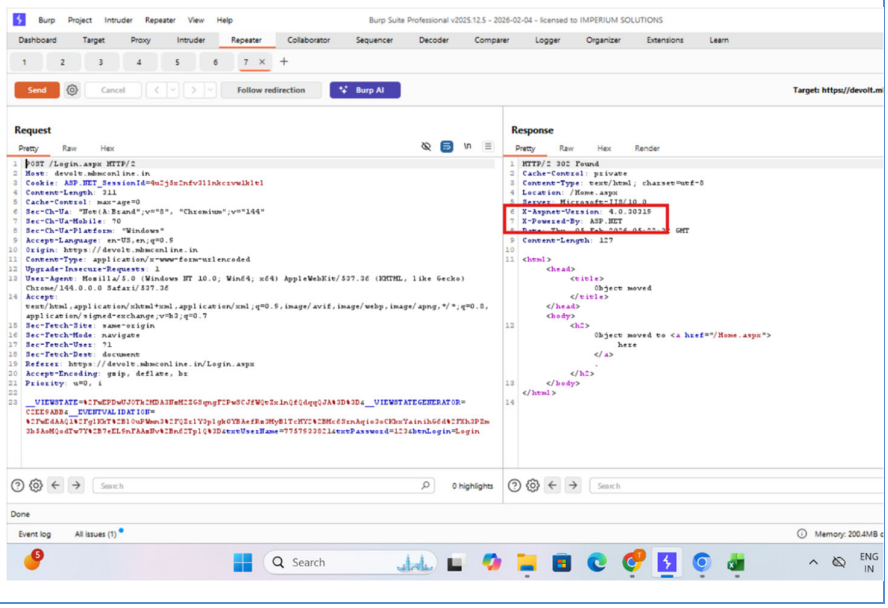
13. Version Disclosure (Bootstrap)

Vulnerability Title	***Affected URLs/IP***
Version Disclosure (Bootstrap)	devolt.mbmconline.in
Detailed Observation	The application discloses the Bootstrap framework version through client-side resources where the version information is explicitly mentioned (e.g., Bootstrap v3.3.7). Disclosure of third-party library versions allows an attacker to identify the exact framework version in use and potentially exploit known vulnerabilities associated with that version.
Vulnerability Reference (CWE/CVE)	CWE-200
Severity	**Low**
Recommendation	Remove or obfuscate version comments from client-side files where feasible. Upgrade Bootstrap to the latest supported version (v5.3.8) and ensure outdated components are removed. Regularly review and update third-party libraries to minimize information disclosure and reduce the attack surface.
Reference	https://owasp.org/www-community/attacks/Information_Disclosure
New/ Repeat Observation	New Observation

References to evidences / Proof of Concept (POCs)	

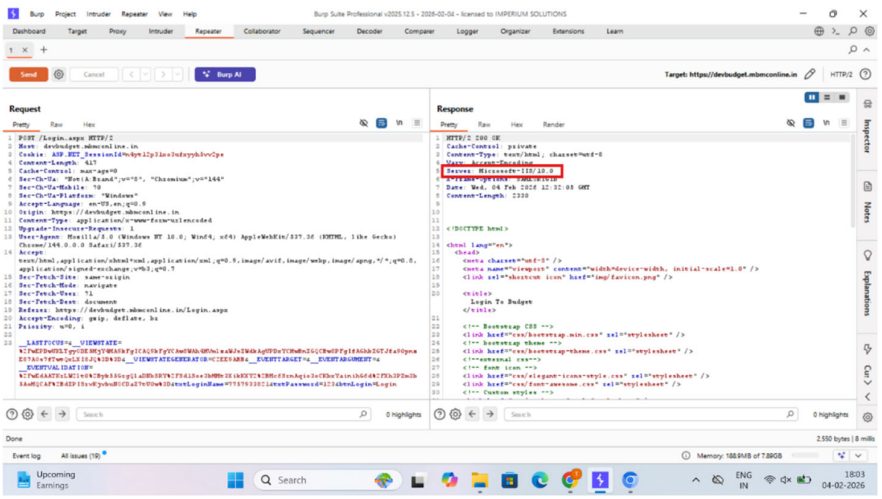
14. Version Disclosure (ASP.NET)

Vulnerability Title	***Affected URLs/IP***
Version Disclosure (jQuery UI)	devolt.mbmconline.in
Detailed Observation	The application discloses ASP.NET framework version information through HTTP response headers such as X-AspNet-Version and X-Powered-By. This information reveals the underlying technology and framework version in use, which can assist an attacker in identifying and exploiting known vulnerabilities specific to that ASP.NET version.
Vulnerability Reference (CWE/CVE)	CWE-200

Severity	**Low**
Recommendation	Disable or suppress version disclosure headers by removing X-AspNet-Version and X-Powered-By from HTTP responses. Ensure the application runs on a fully patched and supported version of ASP.NET and regularly apply Microsoft security updates.
Reference	https://owasp.org/www-community/attacks/Information_Disclosure
New/ Repeat Observation	New Observation
References to evidences / Proof of Concept (POCs)	

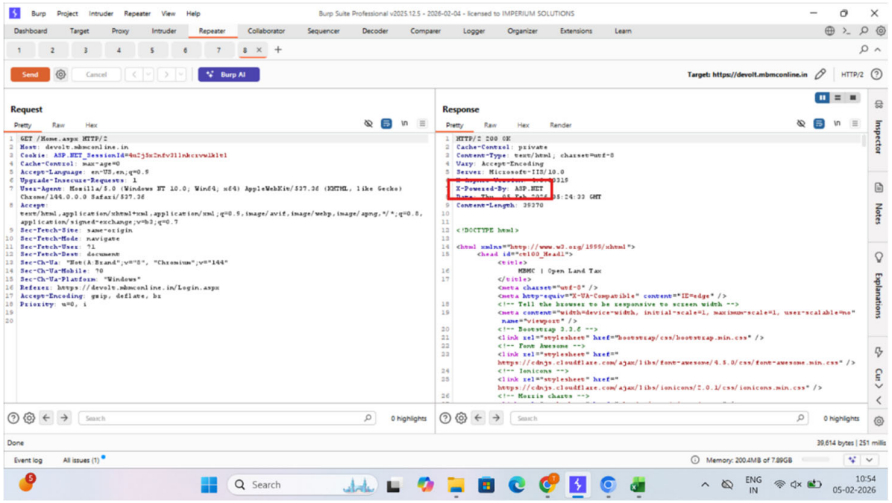
15. Server Disclosure

Vulnerability Title	***Affected URLs/IP***
Server Disclosure	devolt.mbmconline.in
Detailed Observation	It was observed that the application discloses server technology details in the HTTP response headers. The Server header reveals the backend web server and version information (e.g., Microsoft-IIS/10.0). Such disclosures provide attackers with valuable information about the underlying infrastructure, which can be leveraged to identify known vulnerabilities,

	misconfigurations, or targeted exploits specific to the disclosed server version.
Vulnerability Reference (CWE/CVE)	CWE-200
Severity	**Low**
Recommendation	It is recommended to remove or obfuscate server identification headers by configuring the web server to suppress detailed version information. Implement secure server hardening practices, ensure unnecessary headers are disabled, and regularly review HTTP response headers to minimize information disclosure.
Reference	https://learn.microsoft.com/en-us/archive/blogs/varunm/remove-unwanted-http-response-headers https://www.acunetix.com/vulnerabilities/web/version-disclosure-iis/
New/ Repeat Observation	New Observation
References to evidences / Proof of Concept (POCs)	

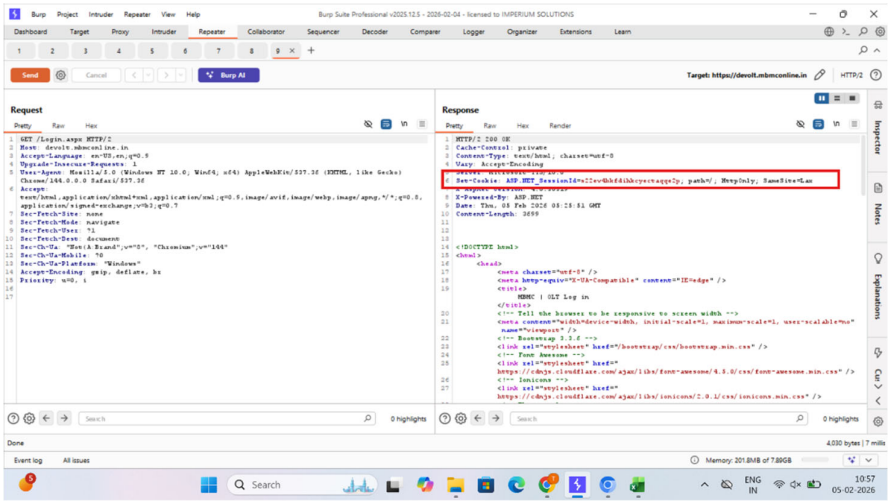
16. Stack Trace Disclosure (ASP.NET)

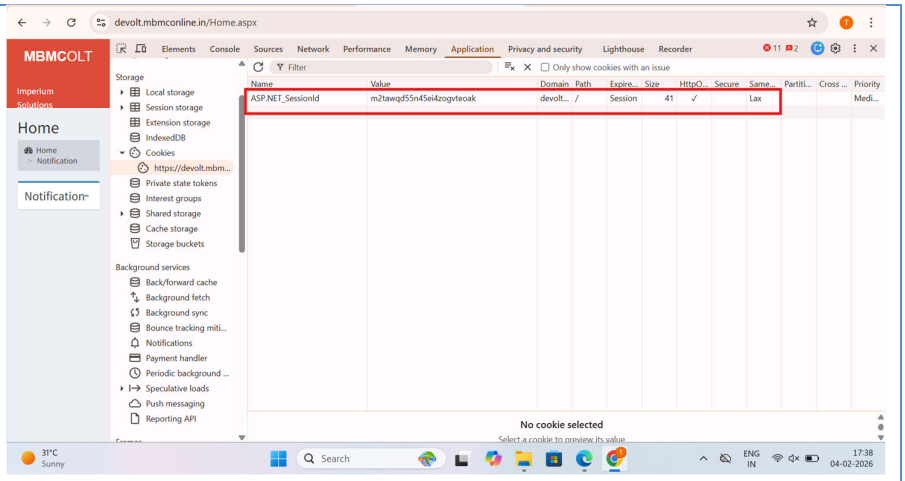
Vulnerability Title	***Affected URLs/IP***
Stack Trace Disclosure (ASP.NET)	devolt.mbmconline.in

Detailed Observation	During testing, the application response was found to disclose ASP.NET internal details through HTTP response headers and application behavior. Such disclosures may expose internal application logic, framework details, or error-handling configurations. If stack traces are exposed during error conditions, an attacker could gain insights into application structure, file paths, and underlying technologies, increasing the risk of targeted attacks.
Vulnerability Reference (CWE/CVE)	CWE-209
Severity	**Low**
Recommendation	Disable detailed error messages and stack trace disclosure in production by configuring customErrors in web.config. Ensure that detailed exceptions are logged internally while presenting generic error messages to users. Regularly review error-handling configurations to prevent leakage of sensitive debugging information.
Reference	https://www.acunetix.com/vulnerabilities/web/stack-trace-disclosure-asp-net/
New/ Repeat Observation	New Observation
References to evidences / Proof of Concept (POCs)	

17. Cookie not marked as securesite attribute

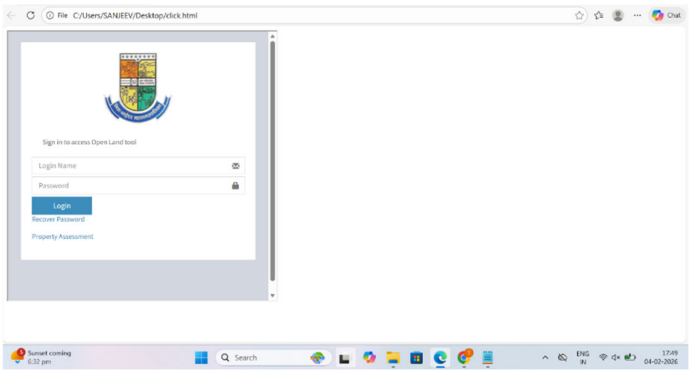
Vulnerability Title	***Affected URLs/IP***

Cookie not marked as secure site attribute	devolt.mbmconline.in
Detailed Observation	During testing, the application was observed setting a session cookie without the Secure attribute. As a result, the cookie may be transmitted over unencrypted HTTP connections if accessed improperly, which could allow an attacker to intercept or hijack the session through man-in-the-middle (MITM) attacks.
Vulnerability Reference (CWE/CVE)	CWE-614
Severity	**Low**
Recommendation	Configure the application to set the Secure flag on all sensitive cookies to ensure they are transmitted only over HTTPS connections. Additionally, enforce HTTPS across the application and review cookie attributes such as HttpOnly and SameSite for improved session security.
Reference	https://cwe.mitre.org/data/definitions/614.html
New/ Repeat Observation	New Observation
References to evidences / Proof of Concept (POCs)	

18. Clickjacking

Vulnerability Title	***Affected URLs/IP***
Clickjacking	devolt.mbmconline.in
Detailed Observation	The application was successfully loaded within an HTML <iframe> from an external source, as demonstrated during testing. This indicates that the application does not implement proper anti-clickjacking protections such as the X-Frame-Options header or an appropriate Content-Security-Policy (frame-ancestors) directive. An attacker could exploit this by embedding the application in a malicious page and tricking users into performing unintended actions.
Vulnerability Reference (CWE/CVE)	CWE-1021
Severity	**Low**
Recommendation	Implement the X-Frame-Options header with the value DENY or SAMEORIGIN. Additionally, configure a Content-Security-Policy with the frame-ancestors directive to explicitly restrict which domains are allowed to embed the application. These controls will prevent unauthorized framing and mitigate clickjacking attacks.

Reference	https://cwe.mitre.org/data/definitions/693.html https://cwe.mitre.org/data/definitions/1021.html
New/ Repeat Observation	New Observation
References to evidences / Proof of Concept (POCs)	

19. Internal Server Error

Vulnerability Title	***Affected URLs/IP***
Internal Server Error	devolt.mbmconline.in
Detailed Observation	During testing, the application returned a detailed ASP.NET error page when special characters and script payloads were submitted in the input field. The error response disclosed sensitive internal information such as framework type (ASP.NET), exception class (HttpRequestValidationException), file system paths, temporary ASP.NET file locations, and stack trace details. This information can assist attackers in understanding the application's internal structure and technology stack.
Vulnerability Reference (CWE/CVE)	CWE-209
Severity	**Low**
Recommendation	Configure the application to display generic, user-friendly error messages instead of detailed system exceptions. Disable detailed error messages in production by setting customErrors to On in the ASP.NET configuration. Additionally, implement proper server-side input validation and centralized exception handling to prevent unhandled errors from being exposed to end users.

Reference	https://cwe.mitre.org/data/definitions/209.html
New/ Repeat Observation	New Observation
References to evidences / Proof of Concept (POCs)