Web Application Security – Tools & Middleware

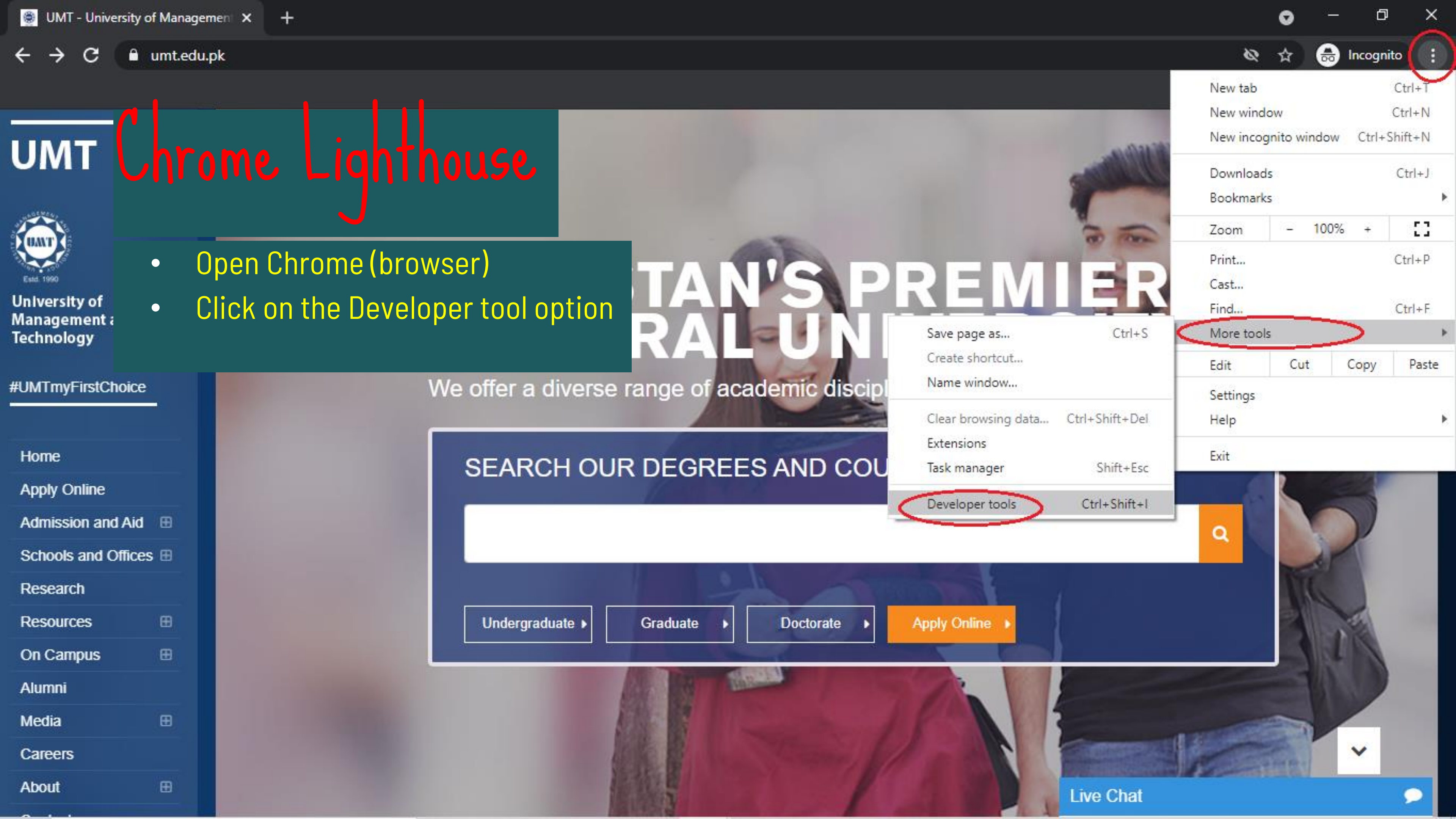
Information Security – Lecture 17

Aadil Zia Khan

First Stop - Evaluating Security

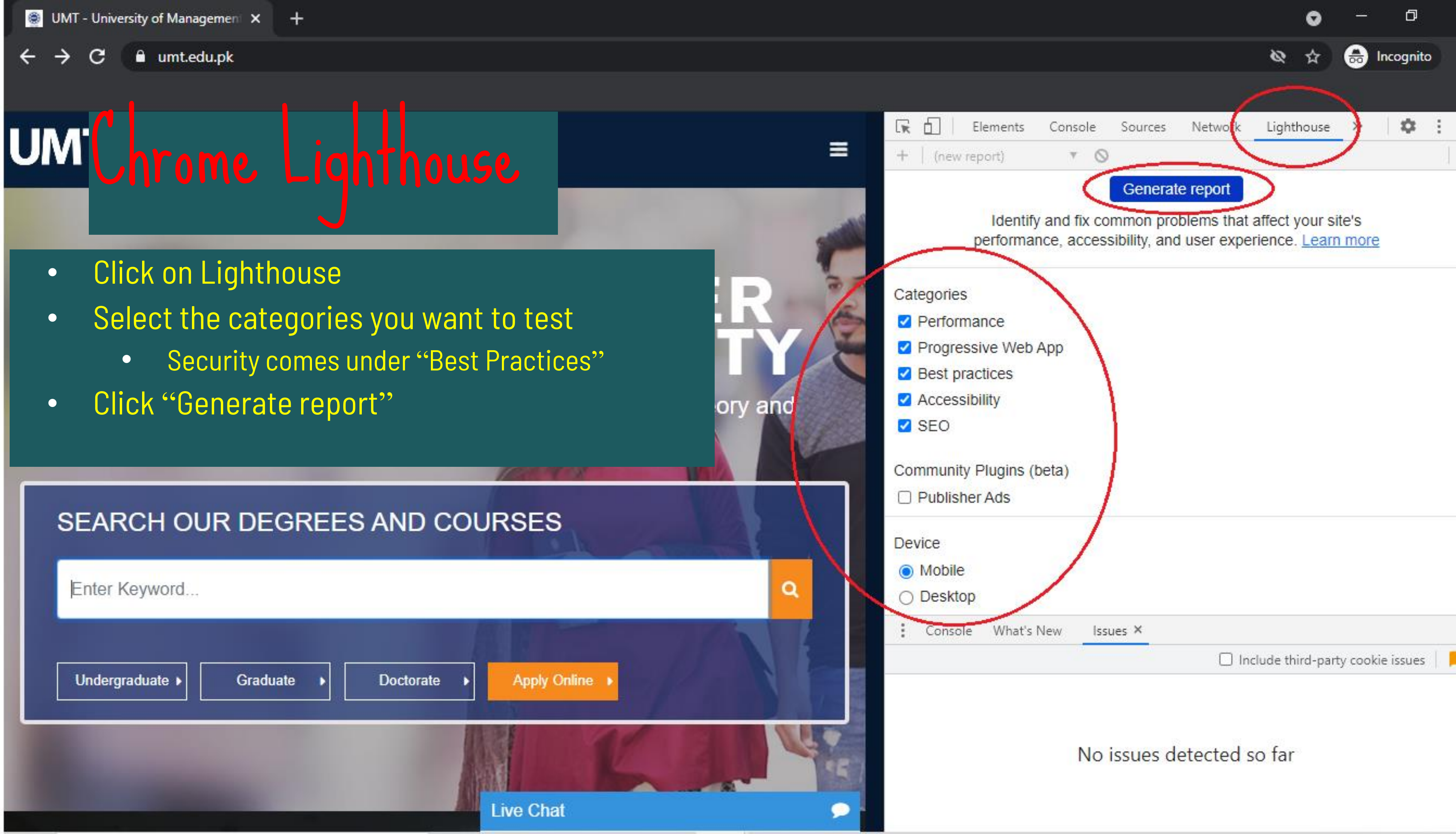
Chrome Lighthouse

- Open Chrome (browser)
- Click on the Developer tool option



Chrome Lighthouse

- Click on Lighthouse
- Select the categories you want to test
  - Security comes under "Best Practices"
- Click "Generate report"



UMT - University of Managemen

umt.edu.pk

Incognito

Elements Console Sources Network Lighthouse

(new report)

Generate report

Identify and fix common problems that affect your site's performance, accessibility, and user experience. Learn more

Categories

- [✓] Performance
- [✓] Progressive Web App
- [✓] Best practices
- [✓] Accessibility
- [✓] SEO

Community Plugins (beta)

- [ ] Publisher Ads

Device

- [✓ Mobile]
- Desktop

Console What's New Issues

Include third-party cookie issues

No issues detected so far

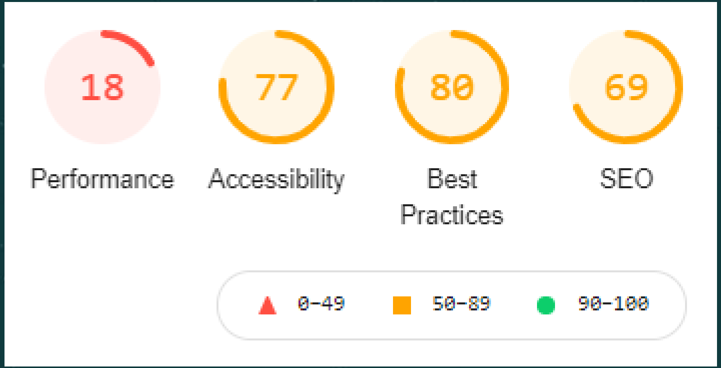
SEARCH OUR DEGREES AND COURSES

Enter Keyword...

Undergraduate Graduate Doctorate Apply Online

Live Chat

Chrome-Lighthouse : UMT Homepage Security

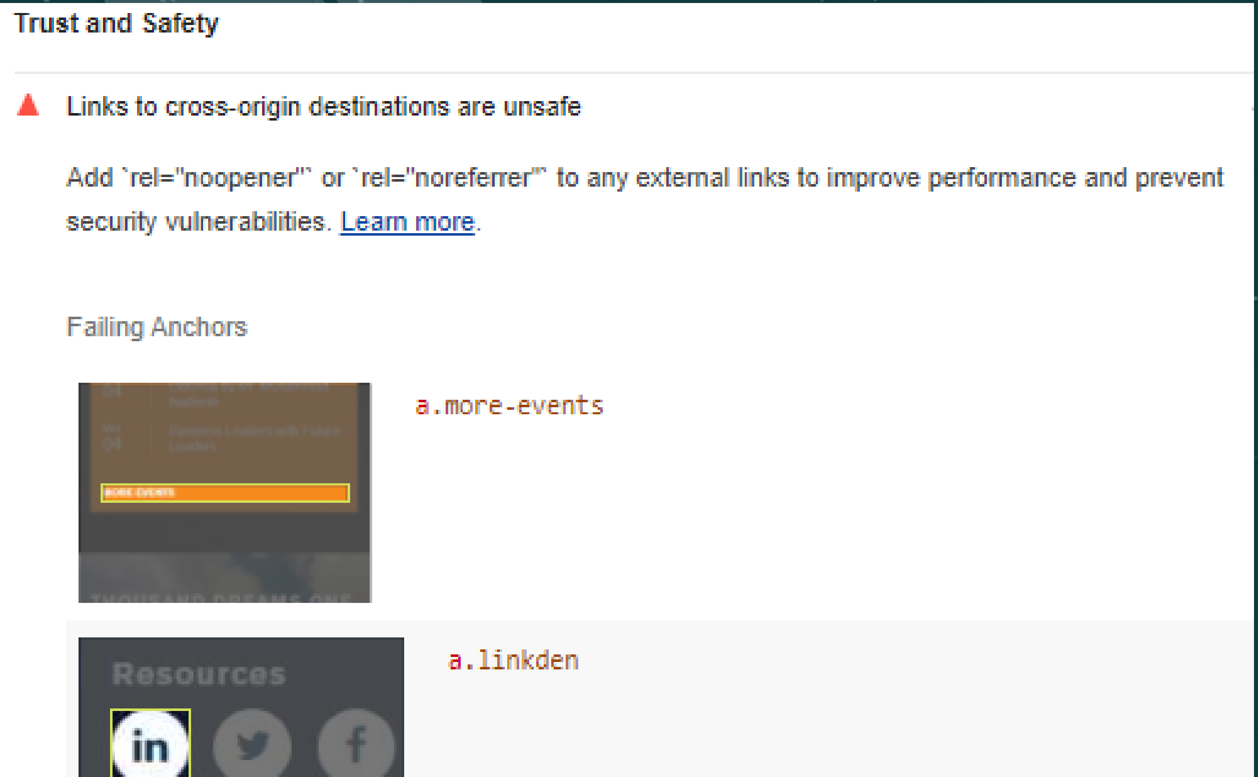


18	77	80	69
Performance	Accessibility	Best Practices	SEO

▲ 0-49 ■ 50-89 ● 90-100

- The report shows scores in different areas of UMT's homepage
- Lets focus on UMT homepage's security (click on "Best Practices")

Chrome-Lighthouse : UMT Homepage Security


Trust and Safety

Links to cross-origin destinations are unsafe

Add `rel="noopener"` or `rel="noreferrer"` to any external links to improve performance and prevent security vulnerabilities. Learn more.

Failing Anchors

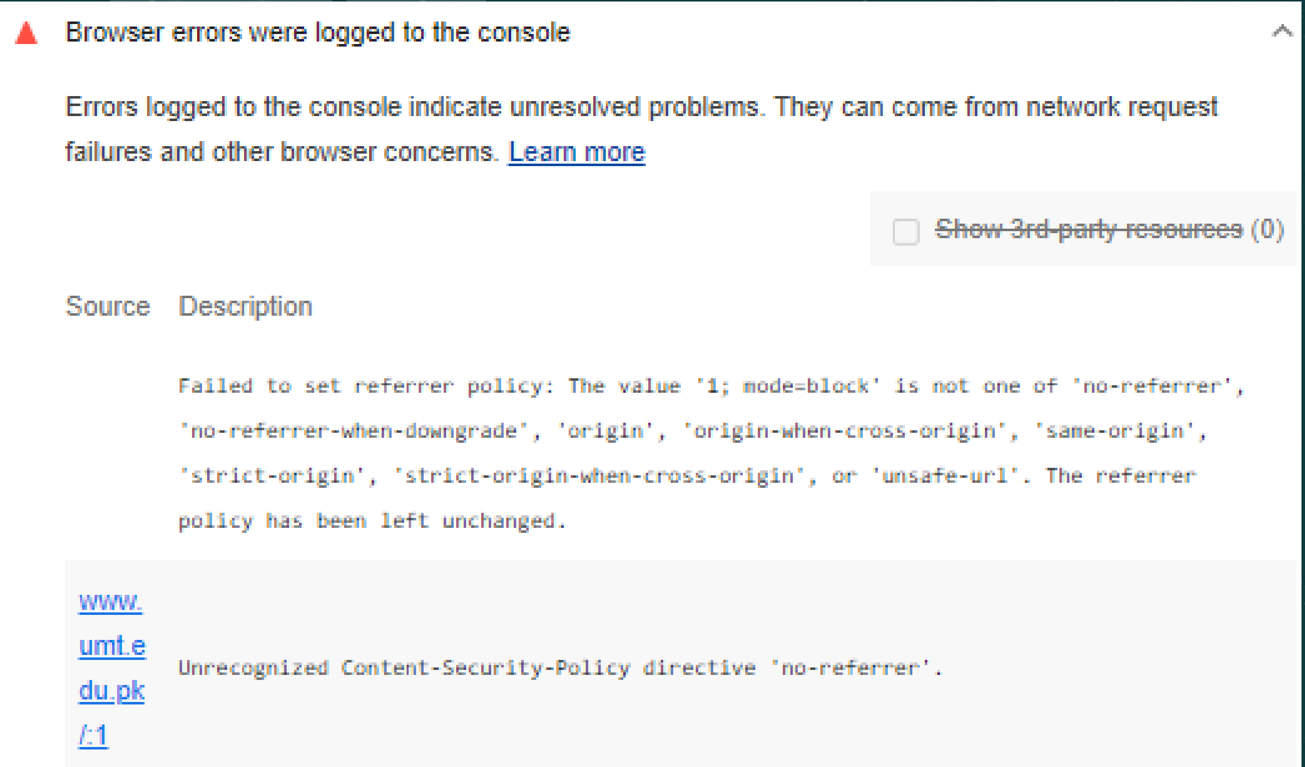
a.more-events

a.linkden

The noreferrer keyword for the rel attribute of the <a>, <area>, and <form> elements instructs the browser, when navigating to the target resource, to omit the Referer header and otherwise leak no referrer information

The noopener keyword for the rel attribute of the <a>, <area>, and <form> elements instructs the browser to navigate to the target resource without granting the new browsing context access to the document that opened it

Chrome-Lighthouse : UMT Homepage Security



Browser errors were logged to the console

Errors logged to the console indicate unresolved problems. They can come from network request failures and other browser concerns. Learn more

~~Show 3rd-party resources~~ (0)

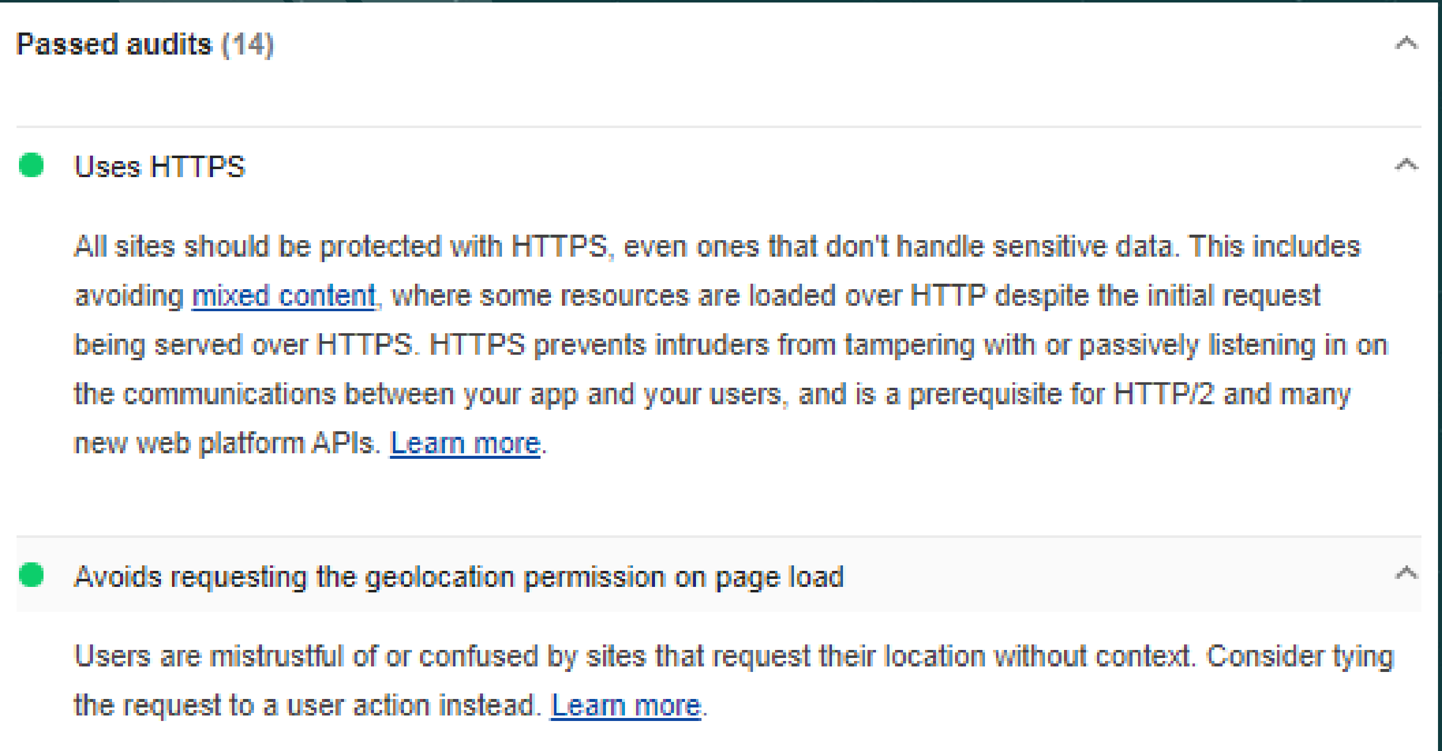
Source	Description
	`Failed to set referrer policy: The value '1; mode=block' is not one of 'no-referrer', 'no-referrer-when-downgrade', 'origin', 'origin-when-cross-origin', 'same-origin', 'strict-origin', 'strict-origin-when-cross-origin', or 'unsafe-url'. The referrer policy has been left unchanged.`
www.umt.edu.pk/:1	`Unrecognized Content-Security-Policy directive 'no-referrer'.`

Due to some error, an incorrect value was passed to:

Content-Security-Policy: referrer <referrer-policy>;

Browser therefore ignored the directive

Chrome-Lighthouse : UMT Homepage Security



Passed audits (14)

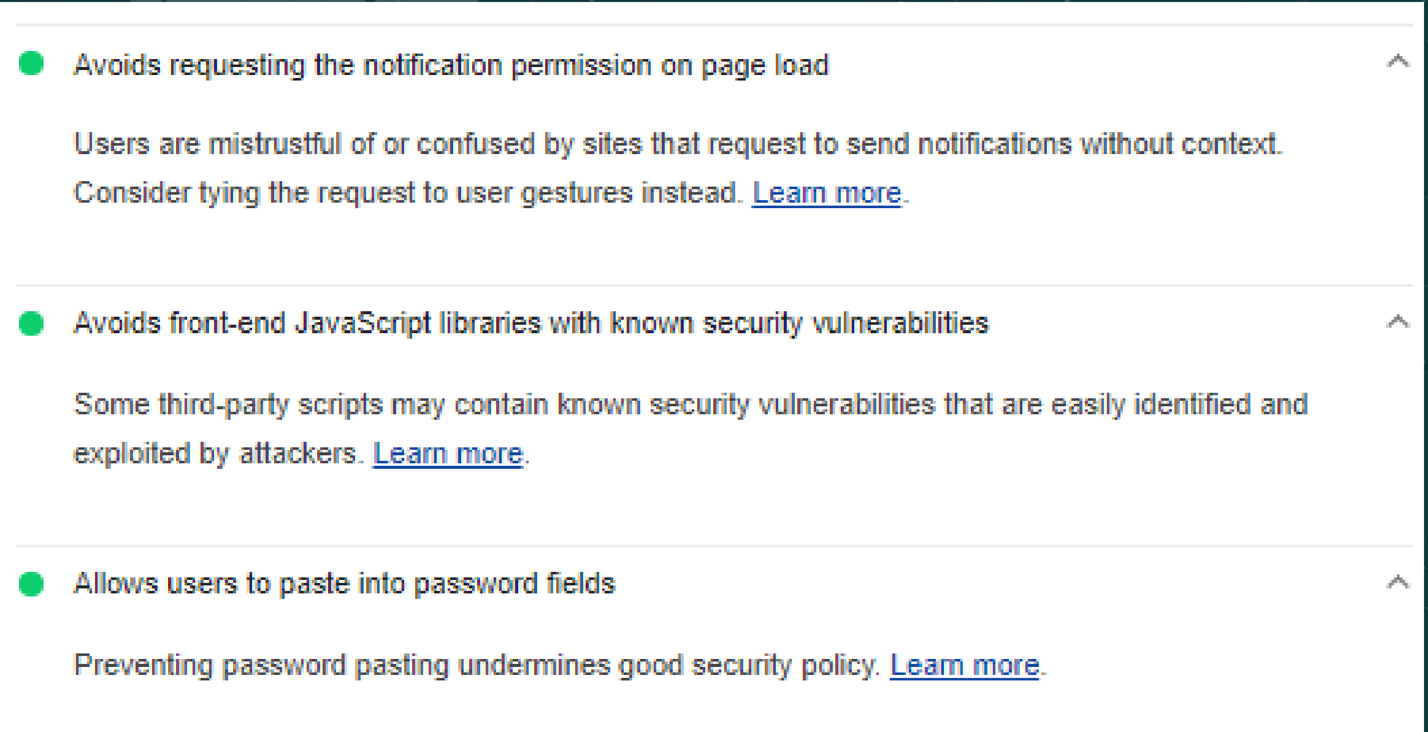
● Uses HTTPS

All sites should be protected with HTTPS, even ones that don't handle sensitive data. This includes avoiding mixed content, where some resources are loaded over HTTP despite the initial request being served over HTTPS. HTTPS prevents intruders from tampering with or passively listening in on the communications between your app and your users, and is a prerequisite for HTTP/2 and many new web platform APIs. Learn more.

● Avoids requesting the geolocation permission on page load

Users are mistrustful of or confused by sites that request their location without context. Consider tying the request to a user action instead. Learn more.

Chrome-Lighthouse : UMT Homepage Security



● Avoids requesting the notification permission on page load

Users are mistrustful of or confused by sites that request to send notifications without context. Consider tying the request to user gestures instead. Learn more.

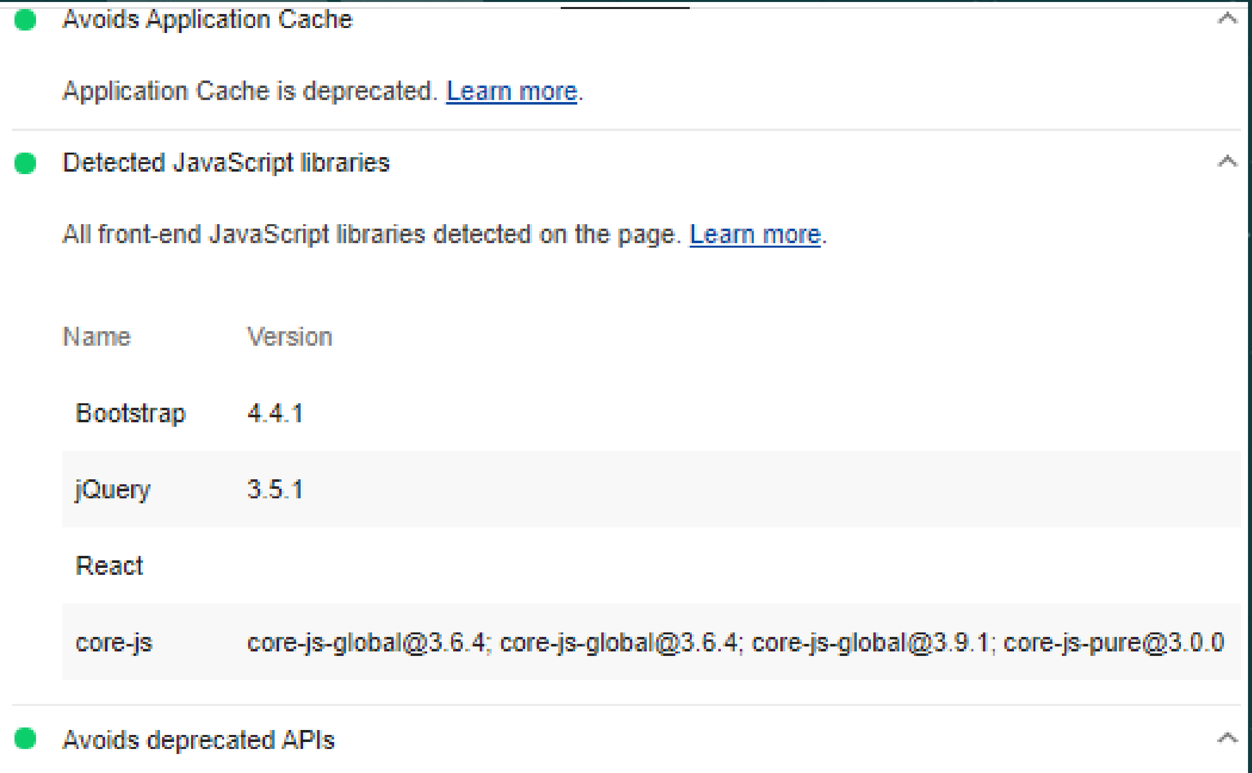
● Avoids front-end JavaScript libraries with known security vulnerabilities

Some third-party scripts may contain known security vulnerabilities that are easily identified and exploited by attackers. Learn more.

● Allows users to paste into password fields

Preventing password pasting undermines good security policy. Learn more.

Chrome-Lighthouse : UMT Homepage Security



● Avoids Application Cache

Application Cache is deprecated. Learn more.

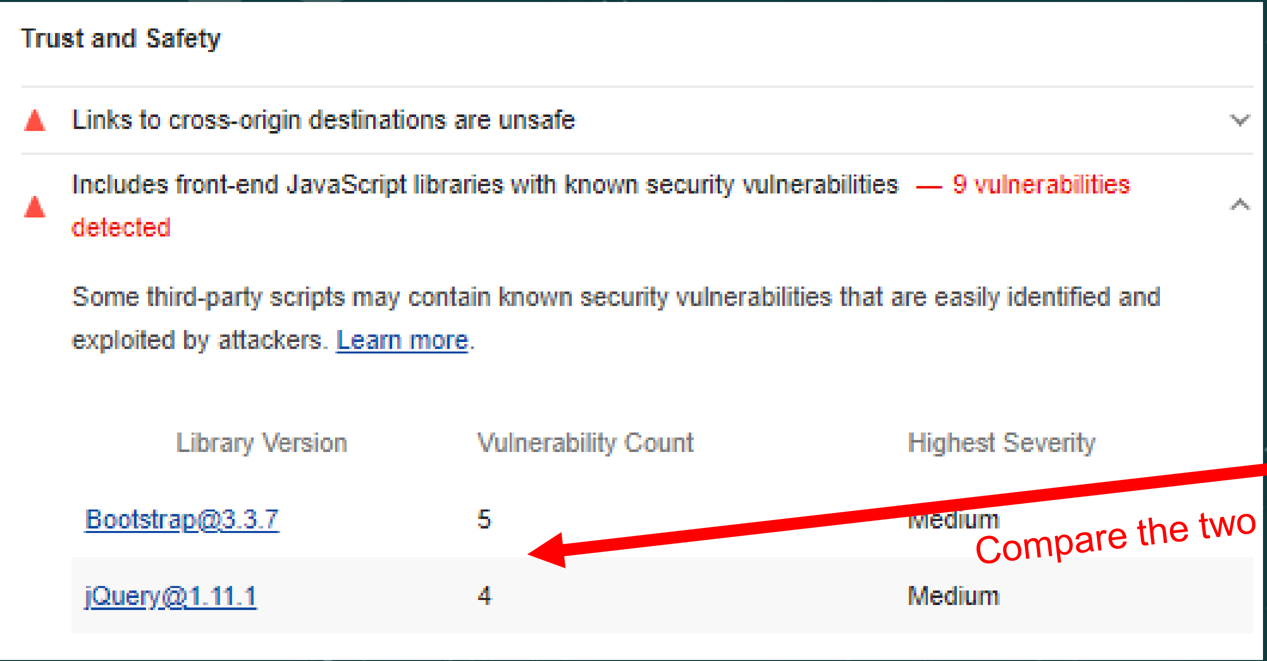
● Detected JavaScript libraries

All front-end JavaScript libraries detected on the page. Learn more.

Name	Version
Bootstrap	4.4.1
jQuery	3.5.1
React	
core-js	core-js-global@3.6.4; core-js-global@3.6.4; core-js-global@3.9.1; core-js-pure@3.0.0

● Avoids deprecated APIs

Chrome-Lighthouse : geo.tv Homepage Security



Trust and Safety

▲ Links to cross-origin destinations are unsafe

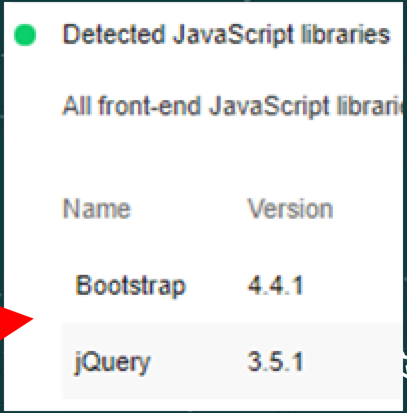
▲ Includes front-end JavaScript libraries with known security vulnerabilities — 9 vulnerabilities detected

Some third-party scripts may contain known security vulnerabilities that are easily identified and exploited by attackers. Learn more.

Library Version	Vulnerability Count	Highest Severity
Bootstrap@3.3.7	5	Medium
jQuery@1.11.1	4	Medium

Compare the two

UMT's JS libraries



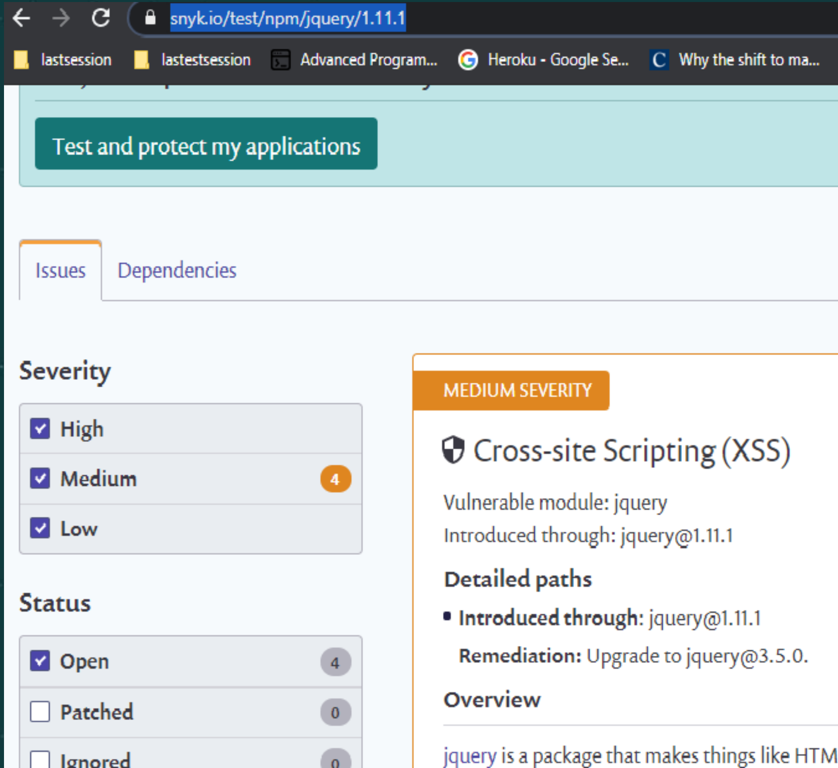
● Detected JavaScript libraries

All front-end JavaScript librari

Name	Version
Bootstrap	4.4.1
jQuery	3.5.1

Vulnerability / Exploit Databases

- Does knowing the library version help the attacker?
- Many websites maintain a vulnerability database online
  - E.g., https://snyk.io/test/npm/jquery/1.11.1
  - E.g., https://snyk.io/test/npm/bootstrap/3.3.7
  - There are others as well



Severity

✓ High	
✓ Medium	4
✓ Low	

Status

✓ Open	4
Patched	0
Ignored	0

MEDIUM SEVERITY

Cross-site Scripting (XSS)

Vulnerable module: jquery
Introduced through: jquery@1.11.1

Detailed paths

- **Introduced through:** jquery@1.11.1
  Remediation: Upgrade to jquery@3.5.0.

Overview

jquery is a package that makes things like HTM

Other Tools

- There are many other tools that can be used to assess the security (and performance) of a webpage
  - E.g., https://webpagetest.org/
  - E.g., https://securityheaders.com

Next Stop - Implementing Security Headers

Server Side Code For HTTP Security Headers

- There are many middleware used to automatically add HTTP security headers to response packets

- Lets focus on Helmet.js

Server Side Code For HTTP Security Headers

- Helmet. js is a useful Node. js module that helps you secure HTTP traffic for Express.js apps
  - To many names – you'll cover these in the Web Dev courses
- It sets up various HTTP headers to prevent attacks like Cross-Site-Scripting(XSS), clickjacking, etc.
- Lets look at some code snippets

We are scratching the surface – details will be covered in the Web Dev courses

Server Side Code For HTTP Security Headers

Telling your application to use HTTP security headers

```
const express = require("express");
const helmet = require("helmet");

const app = express();

app.use(helmet());
```

Same as

```
app.use(helmet.contentSecurityPolicy());
app.use(helmet.dnsPrefetchControl());
app.use(helmet.expectCt());
app.use(helmet.frameguard());
app.use(helmet.hidePoweredBy());
app.use(helmet.hsts());
app.use(helmet.ieNoOpen());
app.use(helmet.noSniff());
app.use(helmet.permittedCrossDomainPolicies());
app.use(helmet.referrerPolicy());
app.use(helmet.xssFilter());
```

Server Side Code For HTTP Security Headers

Setting custom options – example of helmet.referrerPolicy()

```
const express = require("express");
const helmet = require("helmet");

const app = express();
//app.use(helmet());      //don’t use this

app.use(
          helmet.referrerPolicy({
                    policy: "no-referrer",
          })
); // Sets "Referrer-Policy: no-referrer"
```

```
const express = require("express");
const helmet = require("helmet");

const app = express();
//app.use(helmet());      //don’t use this

app.use(
          helmet.referrerPolicy({
                    policy: ["origin", "unsafe-url"],
          })
); // Sets "Referrer-Policy: origin,unsafe-url"
```

Practice

- Select any website of your choice (ideally an old site that you believe would be weak)
- Evaluate it's security using
  - Lighthouse
  - WebPageTest
  - SecurityHeaders