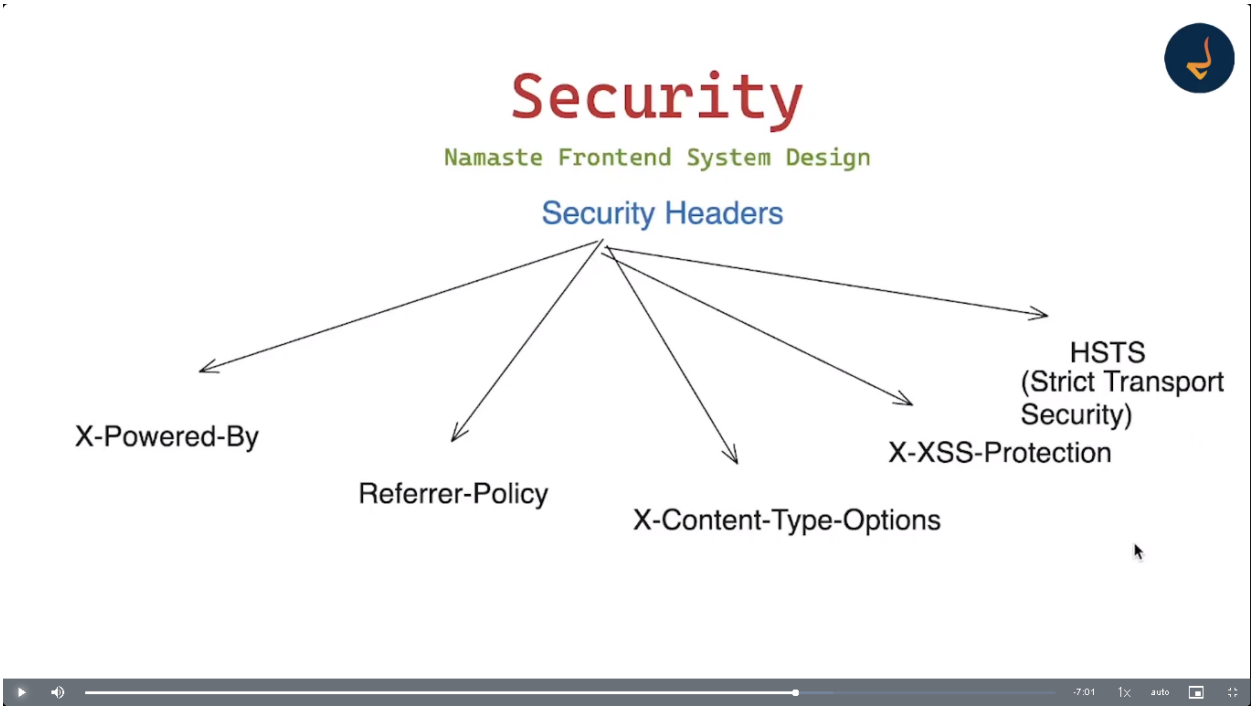
Security headers



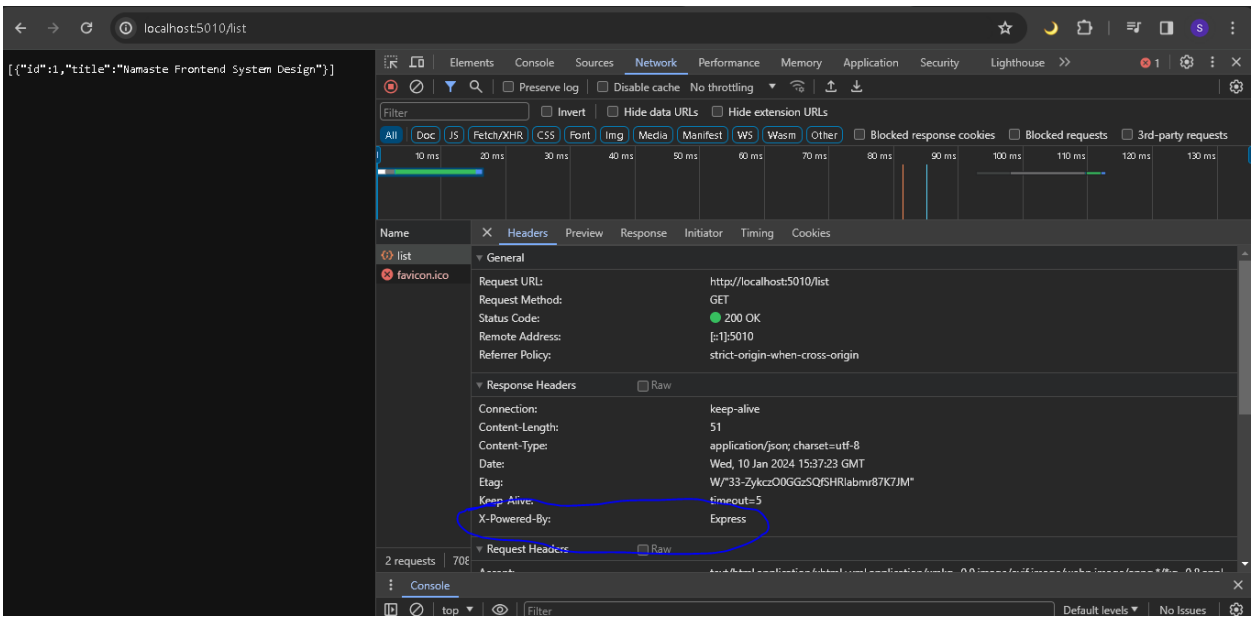
index js

```
const express = require('express');
const app = express();

app.get('/list', (req, res) => {
  res.send([{
    id: 1,
    title: "Namaste Frontend System Design"
  }])
});

const port = process.env.PORT || 5010;
```

```
app.listen(port, () => {
  console.log(`Server is running on port ${port}`);
});
```

X-Powered-By

This header tells you your application is built using what kind of server.



When we run the server, we can see in network tab, it shows our express server. This is not a good practice. We must never expose our server like this. What if there are some problems going on with express and attacker exploits those and attack your system.

Add below code - it will remove X-Powered-By header in network tab

```
app.use((req, res, next) => {
  res.removeHeader('X-Powered-By');
  next();
});
```

Referrer-Policy

The `Referrer-Policy` HTTP header controls how much referrer information (sent with the `Referer` header) should be included with requests. Aside from the HTTP header, you can set this policy in HTML.

Referrer-Policy: no-referrer
Referrer-Policy: no-referrer-when-downgrade
Referrer-Policy: origin
Referrer-Policy: origin-when-cross-origin
Referrer-Policy: same-origin
Referrer-Policy: strict-origin
Referrer-Policy: strict-origin-when-cross-origin
Referrer-Policy: unsafe-url

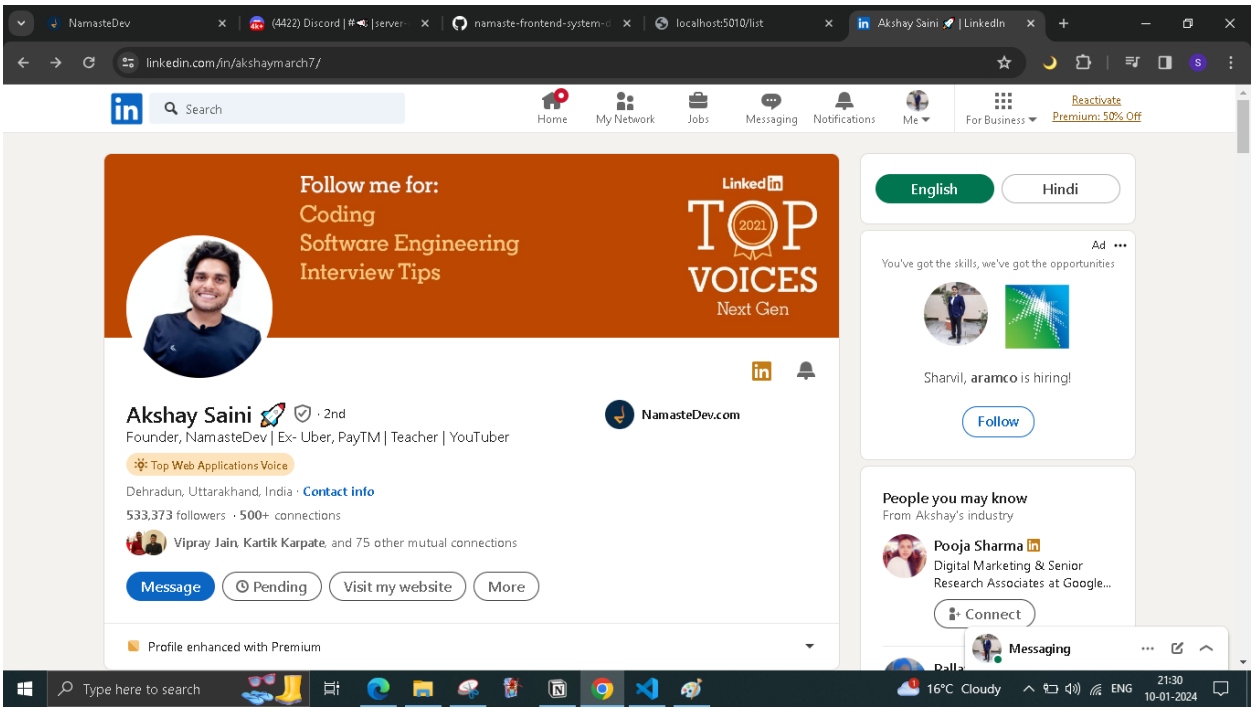
Read more

https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Referrer-Policy

Example

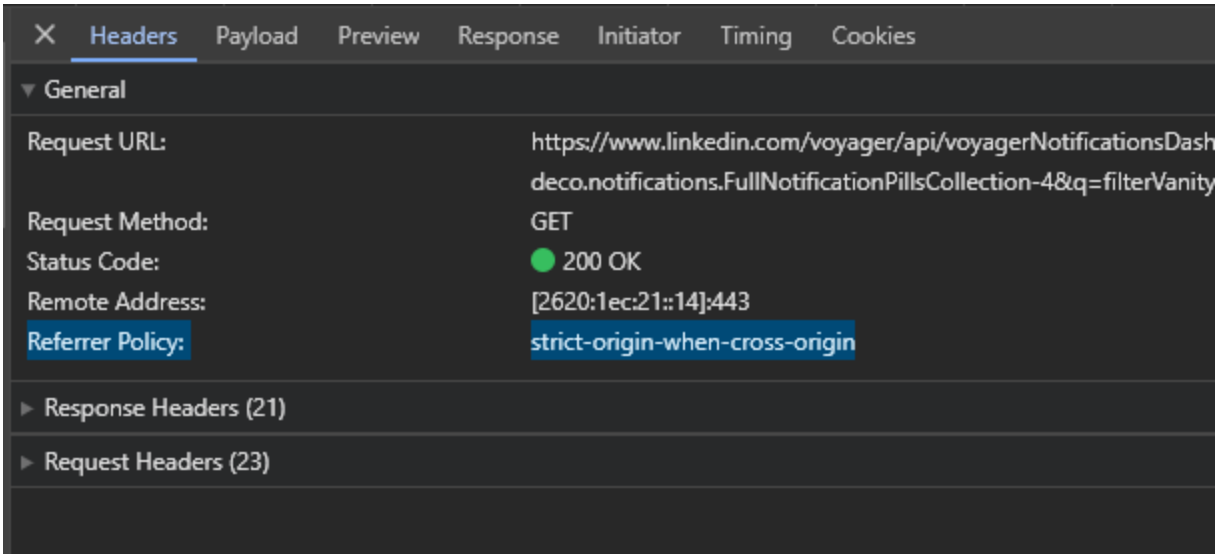
We are on Akshay’s linkedn profile url

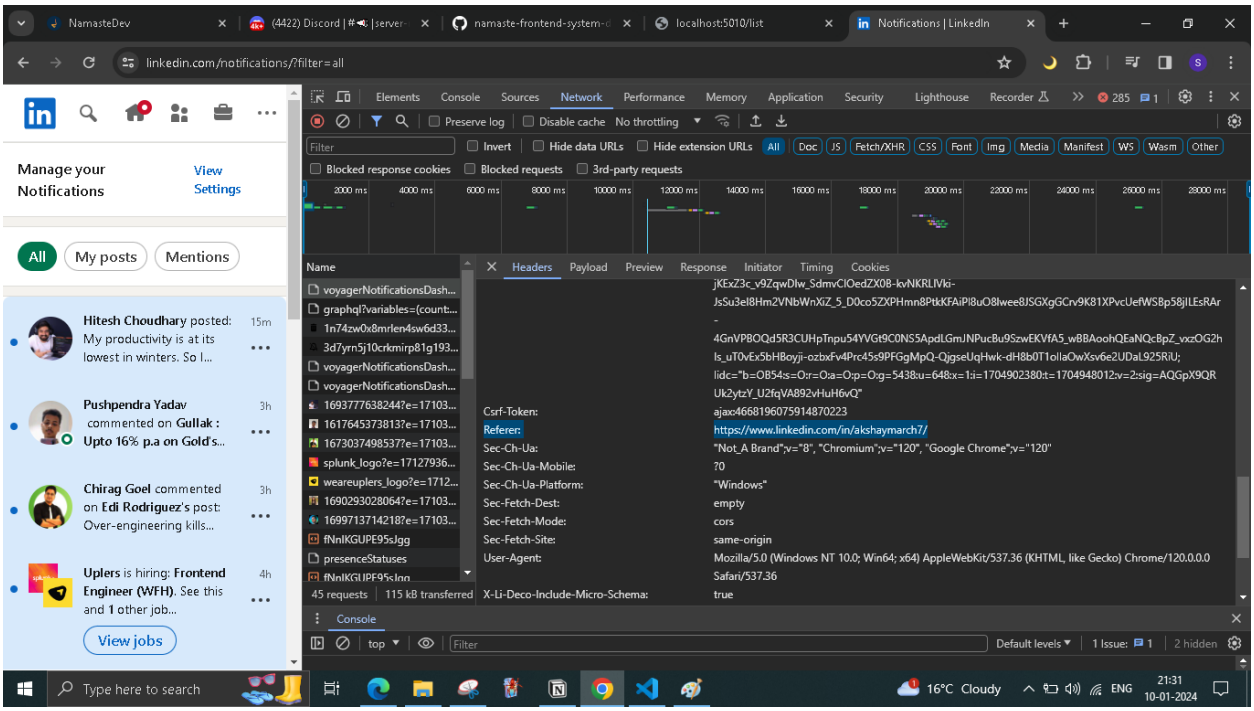
https://www.linkedin.com/in/akshaymarch7/



Now , we will click notification icon and see network tab

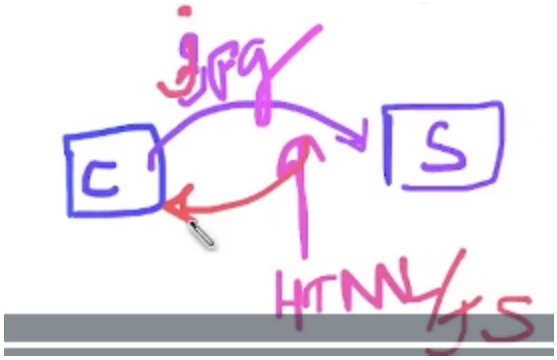
You will see Referrer Policy and referrer






```
app.use((req, res, next) => {
    res.setHeader('Referrer-Policy', 'no-referrer');
  res.removeHeader('X-Powered-By');
  next();
});
```

X-Content-Type-Options



Suppose client requested jpg image from server, but some man in the middle modified the request and injected some html/JavaScript . So, now server will send something else only.

The `X-Content-Type-Options` response HTTP header is a marker used by the server to indicate that the MIME types advertised in the `Content-Type` headers should be followed and not be changed. The header allows you to avoid MIME type sniffing by saying that the MIME types are deliberately configured.

```
app.use((req, res, next) => {
  res.setHeader('Referrer-Policy', 'no-referrer');
  res.removeHeader('X-Powered-By');
  res.setHeader('X-Content-Type-Options', 'nosniff');
  next();
});
```

X-XSS-Protection

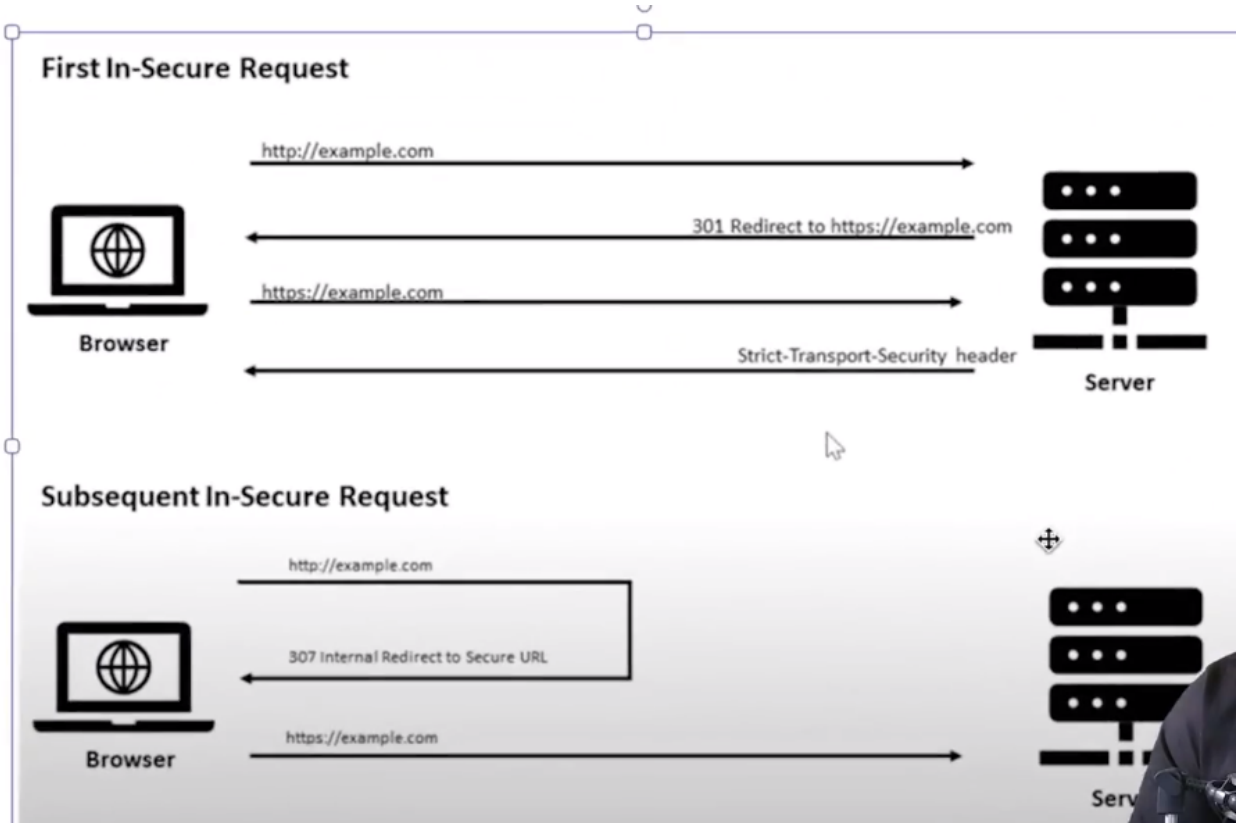
The HTTP `X-XSS-Protection` response header is a feature of Internet Explorer, Chrome and Safari that stops pages from loading when they detect reflected cross-site scripting (XSS) attacks. These protections are largely unnecessary in

modern browsers when sites implement a strong `Content-Security-Policy` that disables the use of inline JavaScript (`'unsafe-inline'`).

X-XSS-Protection: 0
X-XSS-Protection: 1
X-XSS-Protection: 1; mode=block
X-XSS-Protection: 1; report=<reporting-uri>

Strict-Transport-Security(HSTS)

The HTTP `Strict-Transport-Security` response header (often abbreviated as HSTS) informs browsers that the site should only be accessed using HTTPS, and that any future attempts to access it using HTTP should automatically be converted to HTTPS.



It happens in 2 steps as you can see in above image-

1. For first request, we manually redirect to https. Then browser calls https and we set strict-transport-security header.
2. For further request(subsequent in-secure request), browser calls https

```
//not working in localhost
const redirectToHttps = (req, res, next) => {
  if (req.headers['x-forwarded-proto'] !== 'https') {
    // Redirect to HTTPS
    return res.redirect(['https://', req.get('Host'), req.ur
l].join(''));
  }
  next();
};

app.use(redirectToHttps);

app.use((req, res, next) => {
  res.setHeader('Referrer-Policy', 'no-referrer');
  res.removeHeader('X-Powered-By');
  res.setHeader('X-Content-Type-Options', 'nosniff');
  res.setHeader('Strict-Transport-Security', 'max-age=3153600
0; includeSubDomains; preload');
  next();
});
```

If you want above step to happen in single step, then you need to register your domain in

hstspreload.org