COESIONE ITALIA 21-27 PIEMONTE
Cofinanziato dall'Unione europea
REGIONE PIEMONTE

web.config

Web development

Docente: Shadi Lahham

Intervento realizzato da
ITS TECNOLOGIE DELL'INFORMAZIONE E DELLA COMUNICAZIONE

web.config

configuration for in internet information services

web.config

The web.config file is a configuration file used in internet information services (IIS) for managing asp.net applications, offering similar functionalities to .htaccess but tailored for IIS, allowing control over various aspects of the website, including SEO settings, URL rewriting, and traffic redirection

1. Managing index/noindex

In SEO, the index directive allows search engines to include a page in search results, while noindex instructs search engines not to index a page, helping keep irrelevant or sensitive pages out of search results. This is useful for hiding outdated content, duplicate material, or private areas from search engines

Implementation in web.config
To prevent search engines from indexing pages, add the X-Robots-Tag header in web.config

```
<configuration>
  <system.webServer>
    <httpProtocol>
      <customHeaders>
        <add name="X-Robots-Tag" value="noindex, nofollow" />
      </customHeaders>
    </httpProtocol>
  </system.webServer>
</configuration>
```

Targeting Specific Pages or Directories
For specific pages or directories, use a separate web.config file within that directory

```
<configuration>
  <location path="folder/page.html">
    <system.webServer>
```

```
      <httpProtocol>
        <customHeaders>
          <add name="X-Robots-Tag" value="noindex, nofollow" />
        </customHeaders>
      </httpProtocol>
    </system.webServer>
  </location>
</configuration>
```

2. Managing HTTP/HTTPS

Switching all traffic to HTTPS is important for security and SEO. HTTPS is a ranking factor for Google, and mixed content can lead to duplicate content issues and security warnings

Enforces secure browsing and prevents issues with duplicate content from both HTTP and HTTPS versions of the site

Implementation in web.config
Force HTTPS using URL rewriting rules
The rule matches all URLs.
It checks if HTTPS is off and redirects to the HTTPS version with a 301 redirect.

```
<configuration>
  <system.webServer>
    <rewrite>
      <rules>
        <rule name="Redirect to HTTPS" stopProcessing="true">
          <match url="(.*)" />
          <conditions>
            <add input="{HTTPS}" pattern="off" ignoreCase="true" />
          </conditions>
          <action type="Redirect" url="https://{HTTP_HOST}/{R:1}"
redirectType="Permanent" />
        </rule>
      </rules>
    </rewrite>
  </system.webServer>
</configuration>
```

3. Managing www and non-www

Canonicalization involves choosing between WWW and non-WWW versions of a domain to avoid duplicate content and focus SEO efforts on a single domain version

Maintains a consistent URL structure, prevents duplicate content, and centralizes SEO efforts

Redirect Non-WWW to WWW
Use the following configuration to redirect non-WWW to WWW
{R:1} captures the URL path after the domain

```
<configuration>
  <system.webServer>
    <rewrite>
      <rules>
        <rule name="Redirect to WWW" stopProcessing="true">
          <match url="(.*)" />
          <conditions>
            <add input="{HTTP_HOST}" pattern="^example\.com$" />
          </conditions>
          <action type="Redirect" url="http://www.example.com/{R:1}"
redirectType="Permanent" />
        </rule>
      </rules>
    </rewrite>
  </system.webServer>
</configuration>
```

Redirect WWW to Non-WWW
Alternatively, to redirect WWW to non-WWW

```
<configuration>
  <system.webServer>
    <rewrite>
      <rules>
        <rule name="Redirect to non-WWW" stopProcessing="true">
          <match url="(.*)" />
          <conditions>
```

```
            <add input="{HTTP_HOST}" pattern="^www\.example\.com$" />
          </conditions>
          <action type="Redirect" url="http://example.com/{R:1}"
redirectType="Permanent" />
        </rule>
      </rules>
    </rewrite>
  </system.webServer>
</configuration>
```