XML3000R-1

Medlemmer:

Fornavn | Email | Studentnummer

Håvard Betten | 146565@usn.no | 146565

Ola Bredviken | 146975@usn.no | 146975

Fredrik Hulaas | 148258@usn.no | 148258

Fredrik Ravndal | 887698@usn.no | 887698

## Beskrivelse av applikasjonen

Applikasjon henter data fra to forskjellige XML-filer fra nettstedet utdanning.no. Utdanning inneholder alle utdanninger som er dokumentert i Norge, og yrker inneholder alle yrker. Hensikten med dette er å finne ut hvilken utdanning man trenger for hvilke yrker, eller så enkelt som hva et yrke eller utdanning innebærer.

Brukeren har tre muligheter for å se data på nettsiden.

* Oversikt over alle utdanninger
* Oversikt over alle yrker
* Søk på en spesifikk utdanning eller yrke.

Hovedsiden består av en presentasjon av formålet med applikasjonen og en søkefunksjon.  Søkefeltet bruker et javascript hentet ut fra w3schools som heter “autocomplete”. Dette dokumentet består av metoder og lyttere for å vise innholdet i listen som blir laget fra XML-filen utdanningogyrker.xml. Auto Complete brukes til å få opp forslag av alle utdanninger/yrker mens brukeren taster inn, og har egentlig ikke noe med søkefunksjonen å gjøre. Den viser bare mulige forslag for brukeren. Hvis brukeren skriver bokstaven “D”, vil alle yrker/utdanninger som starter med den bokstaven komme som forslag. Hvis man søker på for eksempel “ingeniør”, som er et utdanning og et yrke, kommer det opp alle utdanninger og yrker som har ordet ingeniør i seg. Det brukes regex i søkefunksjonen for å få riktig resultat av søk.

## Hvordan utnytter vi ekstern data?

Dataene blir hentet gjennom å lagre URL-en som referer til XML-dokumentene i variabler som videre blir “loadet”. Eks:

$profession = 'https://utdanning.no/data/atom/yrke';

$xml = **new** DOMDocument();  
$xml->load($profession);

Deretter brukes DOM for å «merge» de to forskjellige XML-dokumentene til ett felles XML-dokument hvor det er lagd ny struktur for XML’en så dataene er bedre å jobbe mot.

Etter dette ble dokumentene gjennomgått med for-løkker hvor alle dataene som trengtes fra de originale XML-dokumentene fra utdanning.no bla lagt i et nytt dokument med ny struktur.

Det brukes htmlspecialchars() for og «escape» spesielle html tegn for best resultat. Eks:

 $doc = **new** DOMDocument("1.0", "ISO-8859-1");  
$doc->formatOutput = **true**;

$root = $doc->createElement('utdanningogyrker');  
$doc->appendChild($root);

**for** ($i = 0; $i < $entryUtdanning->length; $i++) {

$element1 = $doc->createElement('utdanningTittel');  
 $element1->nodeValue=htmlspecialchars($mainTitle1->nodeValue);  
 $startTag->appendChild($element1);

Det passes på at det ikke skrives noe nytt XML-dokument unødig, det vil si at hvis et XML-dokument allerede er generert innen de siste 10 minuttene, vil det ikke gjøres igjen. Dette for å spare brukeren unødig «loade» tid av siden. Eks:

$file = 'utdanningogyrker.xml';  
$time = 600;  
**if**(file\_exists($file) && time() - filemtime($file) >= $time) {

## Hvordan velger vi å vise data?

Dataen vi henter viser vi i to forskjellige tabeller, en med alle yrker og en med alle utdanninger. Dette blir gjort ved bruk av XSL. Vi har stylet tabellene med bruk av bootstrap. Brukeren kan søke opp en utdanning/yrke som han vil se nærmere på, da henter vi ut informasjonen til den posten og viser den i tabell på hovedsiden. Dette gjøres med tanke på at brukeren skal kunne slippe å lete gjennom alle utdanningene i hele XML-fila. Informasjonen i søkefunksjonen blir hentet ut med bruk av PHP. I tabellene vises tittelen på yrke/utdanningen og beskrivelse av de. Vi valgte å bruke bootstrap med tanke på styling av nettstedet. Ettersom dette faget ikke fokuserer på CSS, valgte vi bootstrap til dette. Dette var noe vi ikke hadde brukt før, og tenkte derfor å lære oss dette.

Vi viser også data ved bruk av js i søkefeltet. Hvor alle utdanningstitler og yrkestitler blir lagt i en list og presentert ved input av brukeren.

## Refleksjonsnotat

Fredrik Ravndal:

Jeg har vært med å hente ut XML-data fra nettsiden utdanning.no fra to XML-dokumenter og lage det til ett XML-dokument. Jeg har også jobbet med søkefunksjonen og xsl på hovedsiden. Jeg har også jobbet med å style siden med bootstrap, for å lære dette.

Fredrik Hulaas:

Jeg jobbet med å «merge» de to forskjellige XML-dokumentene til ett felles XML-dokument. Dette gjordes først med fwrite, men ble byttet over til DOM. Jeg har også jobbet med noe XSl, men hovedsakelig PHP og XML.

Håvard Betten:

Jobbet hovedsakelig med XSL i dette prosjektet og litt PHP. Var også med å prøve å få til søkefunksjonen.

Ola Bredviken:

Under dette prosjektet har jeg jobbet mest med XSL og litt PHP, mens jeg også vært med på å style nettsiden med bootstrap.