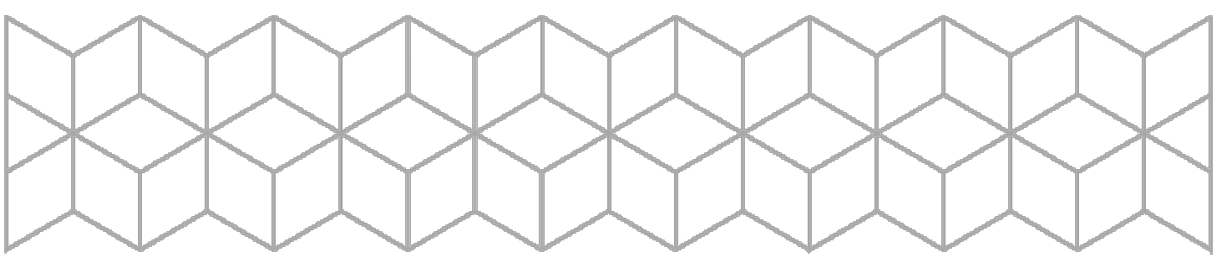
Høgskolen i Østfold

SENSORVEILEDNING

Emnekode:	ITM30617
Emnenavn:	**Utvikling av Interaktive Nettsteder**
Eksamensform:	**Mappeinnlevering og skriftlig eksamen**
Dato:	02.05.2019 (mappe) og 07.05.2019 (skriftlig)
Faglærer(e):	Tore Marius Akerbæk
Eventuelt: Mappeinnlevering teller 60% av samlet karakter. Skriftlig eksamen teller 40% av samlet karakter.	



Oppgaver mappeeksamen

For studenter som tar Utvikling av Interaktive Nettsteder (UIN) sammen med faget Informasjonsarkitektur (IA) skal mappeprosjektet i UIN utvikles på bakgrunn av prosjektet som arbeides med i IA. Det utarbeides en forprosjektrapport i IA som blir utgangspunktet for nettstedet. Studentene i gruppen leverer også en funksjonsbeskrivelse for funksjonen som skal utvikles i tillegg til nettstedet.

For studenter som tar UIN som valgfag uten å gå IA parallelt kan det velges mellom tre forhåndsdefinerte kravspesifikasjoner (disse kan ses her: http://www.it.hiof.no/~toremake/uin2018/?dir=kravspesifikasjoner), eller å utvikle eget prosjekt. Valgfagsstudenter lager også en kort forprosjektrapport for sitt prosjekt, samt en funksjonsbeskrivelse for funksjonen som skal utvikles i tillegg til nettstedet.

For studenter som ønsker å integrere funksjonen i WordPress, tillates dette, da dette krever kompetanse utover læringsmålene og kursets pensum som studentene tilegner seg på egen hånd (ofte med hjelp av litt veiledning).

I mappen skal det leveres

- Et nettsted utviklet i WordPress med tilsnitt av skreddersøm. Dette leveres som et gruppeprosjekt (evalueres som gruppe).
- En funksjon utviklet med minimum HTML, CSS, PHP og MySQL. Dette er eksempelvis en enkel bookingløsning for et hotellnettsted. Funksjonen skal utvikles individuelt (evalueres individuelt). Dersom en gruppe ønsker å utvikle en stor funksjon, kan funksjonaliteten deles opp og gruppemedlemmene arbeide med ulike deler av samme funksjon, så lenge alle viser at de møter læringsmål og minstekrav med sine deler av funksjonen.
- En prosessbeskrivelse som kort forklarer prosessen gjennomført og belyser eventuelle problemområder (eksempelvis om det er klare avvik fra funksjonsbeskrivelsen og den ferdig utviklede funksjonen).

Leveringen skal bestå av:

- URL til det ferdige nettstedet.
- En kodemappe med minimum det utviklede temaet (og eventuelt foreldretemaet)
- URL til den ferdig utviklede funksjonen.
- En mappe med filene som utgjør funksjonen.
- Et dokument som inneholder prosessbeskrivelsen.

Sensorveiledning mappeeksamen

Under sensur skal det vektlegges

1. At sikkerhetsmessige tiltak er tatt
    - under installasjon og oppsett av WordPress
    - ved å ikke ha ubrukte plugins og temaer i den ferdige installasjonen
2. At studentene opererer med fornuftig temastruktur, altså enten et egetutviklet tema (kan også være utviklet med et starter theme ala underscores) eller et foreldre- og dattertema
3. At studentene viser beherskelse av innholdsarkitekturen (riktig bruk av statiske sider og dynamiske artikler, med egne innholdstyper der dette er fornuftig), og følger WordPress fil- og malhierarki fornuftig
4. At studentene klarer å manipulere tema gjennom egen CSS, ikke kun via temaets ferdige konfigurasjonsalternativer.
5. At minimumskravene til nettstedsinnlevering er møtt. Dette er minimum
    - en egen innholdstype
    - en egen page template (ikke standard page.php)
    - en egen single post template (ikke standard single.php)

    Veldig ofte følger disse minimumskravene hverandre, eksempelvis ved visning av referanser for et firma. Referanse er en egen innholdstype, en page template lages for å liste opp referanser, og en single post template lages for å vise en enkelt referanse.
6. At den utviklede funksjonen viser beherskelse av PHP-koding og databasekommunikasjon. Studenten skal kunne sette inn og hente ut og vise informasjon fra databasen i funksjonen sin.

Videre forsterkes karakteren av

1. At studenten viser kreativitet og initiativ for å utvikle et velfungerende nettsted, også fra administrasjonsperspektiv
2. At studenten viser hensyn til sikkerhet, søkemotoroptimalisering og universell utforming.
3. En god prosessbeskrivelse, som også viser innsikt i egen oppnådd kompetanse (NB: Det er ikke framstilt noen krav til akademisk skriving eller bestemt rapportstruktur).
4. At studenten går utover minstekrav der informasjonsarkitektur og innhold tilsier at det er fornuftig (eksempelvis med egne innholdstyper for både rom og matretter i hotellnettstedet skissert i en av de forhåndsviste kravspesifikasjonene)
5. At studenten har en god mappestruktur og viser god beherskelse av organisering av arbeid med større prosjekter.
6. At studenten tør utfordre seg selv i funksjonsutviklingen
7. At studenten viser god beherskelse av databasekommunikasjon, og viser hele CRUD (Create, Read, Update, Delete) i funksjonen sin.

Under sensur er det lagt fokus på å ikke vurdere smak – karakterer baseres ikke på hvor «pent» noe er, men hvor godt det følger prinsipper for brukeropplevelse, informasjonsarkitektur og teknisk utførelse.

Karakterer

A: teknisk tilnærmet komplett, alle minimumskrav er møtt, sikkerhet er tatt hensyn til, temastruktur er riktig. Funksjonen viser beherskelse av databasekommunikasjon som både setter inn, henter, viser, oppdaterer og sletter data, i tillegg til å løse komplekse oppgaver (eksempelvis datokontroller i booking av rom på hotell). Både nettsted og funksjon er velfungerende uten feil eller mangler, og tar hensyn til god brukeropplevelse.

C: Teknisk godt gjennomført, alle minstekrav er møtt. Funksjonen viser god beherskelse av databasekommunikasjon og fungerer som forventet (basert på funksjonens natur).

F: Teknisk svakt gjennomført, minstekravene ikke møtt. Funksjonen viser lite eller ingen beherskelse av databasekommuniksajon, svak teknisk gjennomføring og/eller ingen hensyn til god brukeropplevelse (eksempelvis å bruke felttypen «number» på et felt som kun skal inneholde tall). Obs: Automatisk F dersom sidebyggere (page builders) er benyttet til all innholdsstruktur.

Sensorveiledning skriftlig eksamen

Oppgavene vurderes i henhold til vedlagt løsningsforslag. Det er foreslått en maksimal poengsum som kan oppnås per oppgave. Karakterer gis i henhold til oppnådd poengsum, etter følgende skala:

- A: 95 - 100%
- B: 89 - 94%
- C: 78 - 88%
- D: 65 - 77%
- E: 51 - 64%
- F: < 50%

Løsningsforslag:

1 Variabler

1.1 Vi bruker i hovedsak to metoder for å sende data fra et HTML-skjema til PHP-scriptene våre. Hvilke to metoder?
$_GET og $_POST (1 poeng)

1.2 Hva er forskjellene på disse metodene
$_POST sender data usynlig for brukeren via serveren gjennom et HTTP REQUEST. $_GET sender data som en query-string via URLen, og er synlig i adressefeltet i nettleseren. $_GET kan kun sende 255 tegn videre, mens $_POST begrenses kun av hvor stor belastning serveren klarer å prosessere. (1 poeng)

1.3 For å sende data på tvers av PHP-filer på et nettsted, hvilken metode brukes for å mellomlagre data?
$_SESSION (1 poeng)

1.4 Hva må alltid være på plass i PHP-filene for at metoden fra spørsmål 1.3 skal fungere?
et kall til funksjonen session_start(); helt i starten av koden (før HTMLen) (1 poeng)

1.5 Vi har URLen https://www.domene.com/index.php?p=ansatt&id=2. Skriv PHP-koden som henter id'en fra URLen og skriver den ut til skjerm.
<?php echo $_GET['id']; ?> (1 poeng)

1.6 Vi har URLen https://www.domene.com/index.php?p=produkt&prodnr=23&prodcat=4. Hva vil skrives til skjerm dersom vi skriver følgende PHP-kode: <?php echo «Produktnummer: «.$_GET['prod']; ?>
Kun teksten «Produktnummer:». Det finnes ikke noen variabel i query-stringen som heter prod. (1 poeng)

2 Løkker

2.1 Vi bruker i hovedsak tre typer løkker til ulike formål. Redegjør for de tre typene løkker og beskriv kort typisk bruk av hver av dem.
for-løkker; brukes typisk for å liste opp tall eller løpe gjennom verdier som identifiseres med tallverdier.

foreach-løkker; brukes typisk for å løpe gjennom assosiative arrayer
while-løkker; brukes typisk for å løpe gjennom alle verdiene av et datasett, eksempelvis resultater fra en SQL-spørring

Alle de tre løkkene baserer seg på samme prinsipp; å løpe gjennom et sett med data og utføre en prosess for hver gjennomgang (iterasjon)
(6 poeng: 3 for hver av løkkene, 3 for hver riktige beskrivelse)

2.2 Beskriv de tre variablene vi trenger for å kjøre en for-løkke.
Vi må vite 1) startpunktet/startverdien, 2) stoppunkt/forutsetning for avslutning og 3) utregning som skal skje hver iterasjon/repetisjon.
(3 poeng)

2.3 Vi har hentet noen rader fra en databasetabell gjennom spørringen $sql = mysqli_query($db, «SELECT navn, adresse FROM folkeregisteret»); Skriv starten på løkken du ville brukt for å liste opp resultatene som ligger i $sql.

While($row = mysqli_fetch_array($sql)) { … } (3 poeng)

3 Arrayer

3.1 Hva er forskjellene på en indexert og en assosiativ array?
indexerte arrayer tildeler automatisk en nummeret index til hvert element av en array, mens i en assosiativ array består av par med en nøkkel (som programmereren kan sette selv) og en verdi. Indexerte arrayer bruker tall som nøkler, hvor første nøkkel er 0 (dersom ikke programmeren overstyrer dette ved å manuelt fylle den indexerte arrayen med nøkler og verdier).
(2 poeng)

3.2 Kan en multidimensjonal array kun bestå av flere indexerte arrayer?
Nei, mulitdimensjonale arrayer kan bestå av både enkeltverdier, indexerte og assosiative arrayer. (1 poeng)

3.3 Med koden på bildet under; hvilke navn vil bli skrevet ut?

```php
<?php
    //navn.php

    $navn = array('Trine', 'Jostein', 'Ray', 'Gunnstein', 'Vinnjard', 'Leona');

    for($i = 2; $i < 5; $i++) {
        echo $navn[$i];
    }
?>
```

Siden løkken starter med verdien 2 og kjører så lenge $i er mindre enn 5, vil navnene med index 2, 3 og 4 bli skrevet ut. Dette er Ray, Gunnstein og Vinnjard. (1 poeng for å skrive de korrekte navnene)

3.4 Med URLen navn.php?i=4, hva vil bli skrevet ut fra koden på bildet under?

```php
<?php
    //navn.php

    $familien_Hansen = array('Trine', 'Jostein', 'Ray', 'Gunnstein', 'Vinnjard', 'Leona');

    echo "Mitt navn er " . $familien_Hansen[$_GET['i']] . " Hansen.";
?>

```

Mitt navn er Vinnjard Hansen. (1 poeng)

3.5 Med URLen navn.php?i=6, hva vil bli skrevet ut fra koden på bildet under?

```php
<?php
    //navn.php

    $familien_Hansen = array('Trine', 'Jostein', 'Ray', 'Gunnstein', 'Vinnjard', 'Leona');

    echo "Mitt navn er " . $familien_Hansen[$_GET['i']] . " Hansen.";
?>
```

Mitt navn er Hansen. (Det finnes ikke noen index 6 i arrayen, ergo blir ikke noe navn fra arrayen skrevet ut, kun teksten som allerede er statisk plassert i echo-setningen.) (1 poeng)

3.6 Med koden på bildet under; hvordan ser HTMLen som blir skrevet ut ut?
(Tips: Skriv koden med kommentarer for å følge prosessen i koden)

```php
<?php

    $severdigheter = array(
        'england' => array(
            'london' => array('Tower Bridge', 'Madame Tussauds Wax Cabinet', 'O2 Arena'),
            'liverpool' => array('Abbey Road music studio', 'Liverpool Institute of Permorming Arts')
        ),
        'spania' => array(
            'madrid' => array('Prado-museet', 'Det kongelige palass', 'Thyssen Museer'),
            'barcelona' => array('Park Güell', 'L\'Aquarium Barcelona'),
            'lanzarote' => array('Lavahulene i Jameos del Agua')
        ),
        'italia' => array(
            'milan' => array('Duomo di Milano', 'La Scala'),
            'roma' => array('Det Sixtinske kapell', 'St. Peterskirken')
        )
    );


    foreach($severdigheter as $land => $by) {

        $bc = count($by);

        $all_s = 0;

        foreach($by as $b => $s) {
            $sc = count($s);
            $all_s = $all_s + $sc;

        }

        echo "<p>".$land." (".$bc." byer, ".$all_s." severdigheter)</p>";

    }
?>
```

england (2 byer, 5 severdigheter) spania (3 byer, 6 severdigheter) italia (2 byer, 4 severdigheter)
(2 poeng; oppgaven etterspør HTML som skrives ut, altså skal -tagene inkluderes som en del av svaret for full pott)

4 Databaser

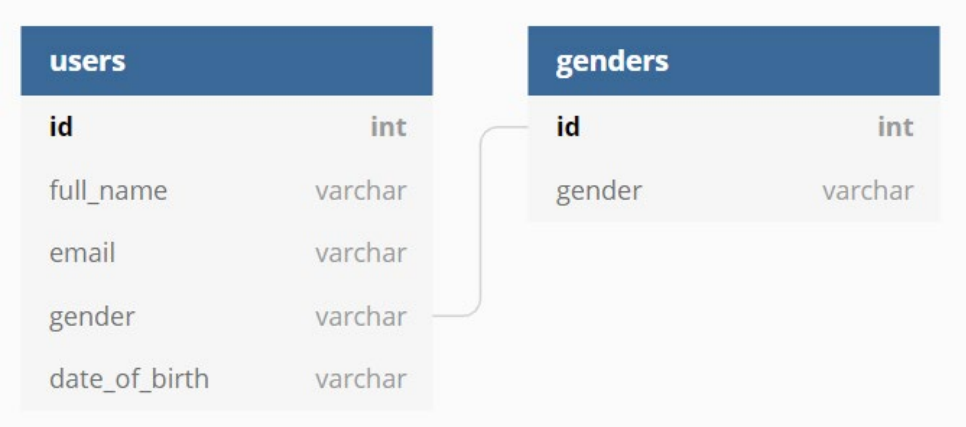
4.1 Hvilke to typer identifikatorer brukes for å beskrive relasjoner mellom tabeller i en database? Forklar forskjellene mellom dem.

Primærnøkkel; som er en unik identifikator i en tabell, og Fremmednøkkel; som er en referanse til en primærnøkkel i en annen tabell for å beskrive relasjonen mellom tabellene. (2 poeng)

4.2 Hva er viktig å ta hensyn til når det gjelder data man tar imot fra et HTML-skjema som skal settes inn i databasen?

At et skjemafelt kan inneholde SQL-kode, og man må sørge for å «vaske» data før de settes inn i databasen. Man kan bruke funksjonen mysqli_real_escape_string for å fjerne potensiell SQL-kode fra dataene man tar imot, mens best practice er å bruke prepared statements ved innsetting av data i en database.
(2 poeng: 1 for å nevne vasking av data, 1 poeng for å ha med en eller begge av mysqli_real_escape_string-funksjonen eller metoden prepared statements)

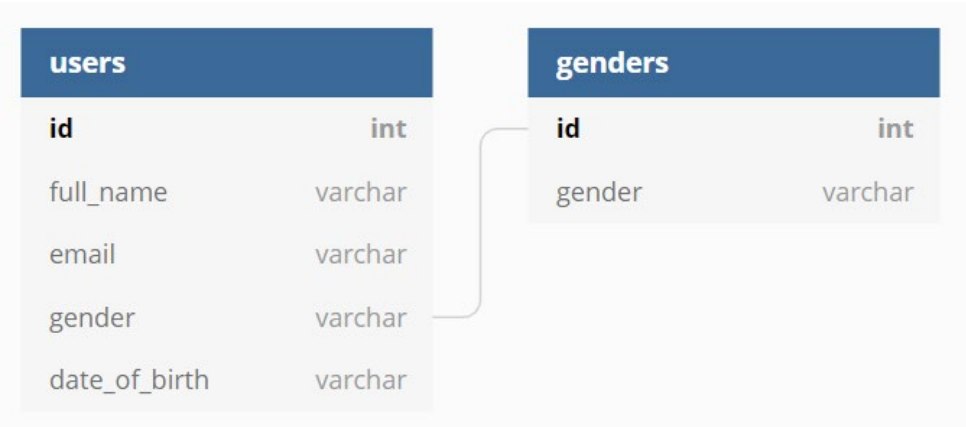
4.3 Med databasetabelloppsettet på bildet under; beskriv hvilke felter i hvilke tabeller som er nøkler, og hvilken type nøkkel de er.



Feltet id er primærnøkkel i de respektive tabellene. I tabellen users er feltet gender en fremmednøkkel til tabellen genders.

Dette vil fungere dersom vi bruker programmatiske relasjoner – at vi knytter sammen tabellene gjennom WHERE-klausulen i spørringen. Siden users.gender er av typen VARCHAR og genders.gender er av typen INT, vil ikke MySQL kunne opprette en index (relasjon) mellom dem.
(3 poeng: 2 poeng for å identifisere riktige nøkler, 1 poeng for å ha med typeforskjellene for users.gender og genders.id)

4.4 Med databasetabelloppsettet på bildet under; Skriv en spørring som henter ut navnet på alle brukere hvor kjønn er «female» hvis «female» har id 1 i tabellen «genders».



SELECT full_name FROM users, genders WHERE users.gender = genders.id AND users.gender = 1 (2 poeng; 1 poeng for riktig format på statement, 1 poeng for korrekt where-klausul. NB: en join-spørring vil også være korrekt her)

5 Kode

5.1 Finn alle feilene/manglene i koden på bildet under, og beskriv hva som er feil. Henvis til linjenummer. (Det er til sammen 7 feil eller mangler).

```php
<php

    $valid == true;

    $product_name = "iPad Mini;

    $price = 6000;

    $vat = $price - ($price * 0.8);

    if($valid = true) {

        echo "<p>You have put " $product_name . " in the cart. Total price is " . price . " NOK (inlcuding VAT: " . $vat . ")</p>"

    }
?>
```

Linje 1: Det mangler et ? i <?php
Linje 3: Det er et = for mye
Linje 5: Det mangler en dobbeltfnutt
Linje 11: Det mangler (minst) et = for at if-testen skal kunne prosessere
Linje 13: Det mangler et . foran $product_name. Det mangler et $ foran price. Det mangler et ; på slutten av utskriften.

Ordet «including» er stavet feil, men dette er ikke avgjørende for om koden kjører eller ikke.

(7 poeng, en for hver feil/mangel)

5.2 Beskriv med tekst hva følgende kode (se bildet under) gjør. Henvis gjerne til linjenummer.

```php
<?php
if(isset($_GET['postnummer'])) {

    $postnummer = array(
        array( 'start' => 1750, 'stopp' => 1789, 'sted' => 'Halden' ),
        array( 'start' => 1790, 'stopp' => 1793, 'sted' => 'Tistedal' ),
        array( 'start' => 1794, 'stopp' => 1794, 'sted' => 'Sponvika' ),
        array( 'start' => 1796, 'stopp' => 1796, 'sted' => 'Kornsjø' )
    );

    foreach($postnummer as $ps) {

        if( ($ps['start'] <= $_GET['postnummer']) && ($_GET['postnummer'] <= $ps['stopp'])) {
            echo "<p>Postnummeret ".$_GET['postnummer']." hører til poststed ".$ps['sted']."</p>";
        }

    }

}
?>
<h2>Sjekk postnummer</h2>
<form action="" method="get">
    <p>
        <label for="postnummer">Postnummer</label>
        <input type="number" name="postnummer" id="postnummer" />
    </p>
    <p><button type="submit">Send</button></p>
</form>
```

På linje 2 kontrolleres om det er satt en variabel «postnummer» i query-stringen når denne filen åpnes. Hvis den er det, opprettes en array i varabelen «postnummer» på linje 4. Denne arrayen

består av 4 assosiative arrayer, hver med 3 sett nøkkel/verdi-par. Nøklene start og stopp inneholder et tall hver, mens nøkkelen sted inneholder et stedsnavn som tekst.

På linje 11 startes en foreach-løkke som går gjennom alle verdiene i arrayen $postnummer, og hver verdi lagres i variabelen $ps. I løkken, på linje 13, utføres en if-test som kontrollerer at postnummeret fra query-stringen er større eller lik start-verdien og mindre eller lik stopp-verdien. Hvis if-testen slår inn, skrives setningen Postnummerer XXXX hører til posted POSTSTED, hvor XXXX vil være postnummeret som finnes i query-stringen, og POSTSTED er verdien fra nøkkelen 'sted' i postnummer-arrayen.

På linje 21 til 28 er et HTML-skjema som sender verdier til filen vi befinner oss på med metoden «get». Det er et felt brukeren kan fylle ut kalt «postnummer», samt en knapp. Når knappen trykkes, lastes filen på nytt, og skjemaet sender det som er skrevet i feltet postnummer som en verdi til variabelen postnummer i en query-string.

(5 poeng, vurderes etter tydelighet og om de viktigste programstegene er forklart; verdi fra query-stringen, array-struktur, foreach-løkken, if-setningen og hvordan den oppnår true-status, utskrift og skjemafunksjon).

5.3 Med koden og funksjonaliteten fra oppgave 5.2; redegjør for om det er greit å bruke metoden GET i skjemaet og eventuelle fordeler og ulemper dette medfører.

Siden ingen sensitive data sendes via skjemaet, er det uproblematisk å bruke GET som metode her. En fordel er at man også kan gjøre oppslag mot filen via en Url, altså ikke være avhengig av å sende postnummer via skjemaet. Hadde vi brukt post som metode måtte vi gått via skjemaet og trykket på knappen for å resende postdataene for hver nye postnummer. En URL kan også bokmerkes, altså kan man unngå å måtte gjøre flere oppslag for verdier man bruker ofte.

(2 poeng; 1 for å godkjenne GET som metode i dette tilfellet, 1 for begrunnelse (bokmerking, ikke sensitive data)

5.4 Med koden fra oppgave 5.2; Vi ønsker å gjøre om arrayen $postnummer til en databasetabell for å effektivisere postnummersjekken vår. Beskriv hvilke felter som må være med, hvilke felttyper de må ha, og hvilke felt som bør brukes som primærnøkkel. Skriv gjerne forklaring/tankerekke for hvordan du har kommet fram til resultatet. (Tabellen skal kun inneholde norske postnummer/poststeder).

Siden det kun skal være norske postnummer i tabellen kan feltene som inneholder startnummeret og stoppnummeret være av felttypen smallint (int vil også fungere) med en lengde på 4 siffer. Feltet sted kan være en varchar med en lengde på ca 120 tegn (ideell størrelse vil være antall bokstaver/tegn i det lengste stedsnavnet i Norge, siden vi ikke vet det må vi ta en størrelse som sannsynligvis dekker dette).

Hvis alle stedsnavn har en (og kun en) suksessiv tallrekke med tilhørende postnummer (eks: 1750-1789 er Halden), kan feltet sted være primærnøkkel, siden poststedet ikke trenger å ha to tallrekker knyttet til seg. Hvis vi ikke vet dette, vil det være fornuftig å lage en konstruert id i et eget felt som primærnøkkel, slik at vi har muligheten til å ha flere tallrekker knyttet til et poststed. (Hvis eks. både tallrekkene 1750-1789 og 1794-1796 har Halden som poststed, kan ikke sted være primærnøkkel)

Optimal løsning med den informasjonen vi vet fra koden i 5.2 og teksten i 5.4 vil da se slik ut:

poststeder	
id	int
start	tinyint(4)
stopp	tinyint(4)
sted	varchar(120)

(3 poeng: vurderes ihht begrunnelse og fornuftig feltnavn og -type)

5.5 Med koden fra oppgave 5.2 og databasetabellen fra oppgave 5.4; Skriv om koden fra oppgave 5.2 slik den ville se ut dersom vi skulle bruke databasetabellen i stedet for arrayen. Få med eventuelle avhengigheter. Hvis du ikke husker nøyaktig hvordan en kode eller funksjon skrives, skriv kommentarer i koden som forteller hva du ønsker å oppnå.

```php
<?php
    if(isset($_GET['postnummer'])) {
        //Hent databaseoppkobling
        require("db.php");

        //Vask data fra query string
        $postnummer = mysqli_real_escape_string($db, $_GET['postnummer']);

        //Sett opp spørring etter postnummer fra query string
        $sql = "SELECT sted FROM postnummer WHERE '".$postnummer."' >= start AND '".$postnummer."' <= stopp";

        //Kjør spørring mot databasen
        $run = mysqli_query($db, $sql);

        //Hent resultater i en array
        $res = mysqli_fetch_array($run);

        if($res) {
            echo "<p>Postnummeret " . $postnummer . " hører til poststed " . $res['sted'] . "</p>";
        }
    }
?>
<h2>Sjekk postnummer</h2>
<form action="" method="get">
    <p>
        <label for="postnummer">Postnummer</label>
        <input type="number" name="postnummer" id="postnummer" />
    </p>
    <p><button type="submit">Send</button></p>
</form>
```

(3 poeng: vurderes ihht om de viktigste programstegene er inkludert (Databaseoppkobling må inkluderes, ok om dette gjøres via fil. Vaske data (eventuelt bruke PDO/prepared statements). Riktig spørring basert på resultatet fra oppgave 5.4). Ved start/stopp-felter bør sjekken om postnummeret befinner seg i riktig tallområde gjøres i SQL-setningen for å spare ressurser, ikke i while-løkken som skriver ut resultatene.)

Læringsutbytte

Studentene har gjennom 10 uker fått opplæring i

- Grunnleggende PHP
- Grunnleggende MySQL
- Innføring i jQuery
- Funksjonsplanlegging- og utvikling
- WordPress som CMS; arkitektur og oppbygning

For å møte følgende læringsutbytte fra emnebeskrivelsen:

Kunnskaper:

Studenten kan anvende

- Databaser
- Grunnleggende prinsipper og metoder innen informasjonsarkitektur og interaksjonsdesign
- HTML / CSS
- JavaScript / PHP
- Publiseringssystemer
- Søkemotoroptimalisering og universell utforming

Ferdigheter:

Studenten kan

- planlegge og redegjøre for utviklingen av et nettstedprosjekt
- sette opp og tilpasse publiseringsløsninger
- lage tilpasset funksjonalitet med kommunikasjon mot databaser

Generell kompetanse:

Studenten har

- kompetanse til å implementere publiseringsløsninger
- erfaring med prosjektarbeid i team

(https://www.hiof.no/studier/emner/it/2019/var/itm30617.html)