Projekt 2

Indledning

Indholdsfortegnelse

Kort resume

I denne rapport gennemgår vi udviklingen af en webløsning som er skrevet i PHP med brugen af cURL og REST til at samarbejde med API'er og indlæse data til fremvisning for brugerne, vedrørende fortidsminder og forplejning muligheder i omegnen af disse.

Problemformulering

Problemstilling

Der skal udvikles et site som borgere med interesse i landets fortidsminder, kan bruge til at finde frem til et eller flere specifikke fortidsminder, som de ønsker at få en detaljeret beskrivelse om, samt en lokation på hvor disse kan findes. Ydermere skal borgerne oplyses om hvor henne nærmeste forplejning mulighed er, til de valgte fortidsminder.
Hjemmesiden skal indeholde en navigation af undersider, samt en brugervenlig udvælgelsesmetode af fortidsminder i de gængse byer eller regioner, som indlæses via cURL fra et autoritativt register, således at man er sikker på at adressen er korrekt beskrevet og noteret.

Problemformulering

Hvordan udvikler vi en brugervenlig hjemmeside løsning, som kan lokalisere fortidsminder ud fra et valg af by eller region, med forplejning muligheder i nærheden?

Afgrænsning

Projektbeskrivelsen lyder på at vi har mulighed for at vælge en løsningsmodel at arbejde ud fra. Vi vælger at udfordre os selv med at udvikle dette projekt for første gang i PHP, og afgrænser os derfor fra at bruge Node.js.

Metodeovervejelser

Teknologier

Vi benytter følgende teknologier til at løse følgende problemstillinger:

- HTML5, bruger vi til at bygge frontend skelettet for hele webapplikationen.
- CSS3/SASS/BOOTSTRAP, bruger vi til at designe udseendet på webapplikationen.
- PHP, bruger vi til at håndtere backenden af vores webløsning.
- cURL, bruger vi til at håndtere vores forbindelser og kald af data til diverse API'er som skal bruges i projektet.
- REST, bruger vi til at arbejde med og manipulere URL'ens query, til at indlæse og vise brugernes angivne data.

Research

API'er

Vi startede med at undersøge hvilke API'er vi skulle bruge for at finde den nødvendige data. Her var vi inde og finde et API fra DAWA som kunne give os fortidsminder for en bestemt kommune, hvorefter vi kunne bruge fortidsmindets koordinater og holde op i mod et andet API, såsom Google Maps, som kunne vise os forplejnings steder inden for en vis radius.

Google maps API

Tanken var at tage DAWA's fortidsminders lokation, og placere dette i google maps api med brug af deres Nearby metode, som finder alt der er registreret i google maps database, og som matcher de filtre som er angivet i søgningen, omkring de koordinater der blev plottet fra fortidsmindet i DAWA. Her ville vi bruge fortidsmindets koordinater i kombination med Google maps nearby metode, hvor man også kan angive en radius sammen med filtre på hvilke faciliteter der søges efter, og til slut danne en query streng som kunne parses i url'en og derved indlæse dataen på siden. Desværre kræver Google Maps Nearby metode en API nøgle fra en betalt konto, formentlig grundet annoncering og markedsføring af nærliggende faciliteter, så vi kunne ikke bruge denne metode alligevel.

AJaX

Ideen bag designet af denne hjemmeside løsning var oprigtigt at "live-opdatere" indholdet der vises på siden, for hver gang brugeren interagerer med dropdowns eller knapper. Sådan en løsning kræver brugen af AJaX, men dette har vi ikke formået at kunne få til at virke. Vi har derfor simplificeret vores tankegang ved at bruge formularer som POST'er til den samme side, hvortil nyt indhold som matcher formularens inputs, vises for brugeren i hhv. Næste kasse med data.

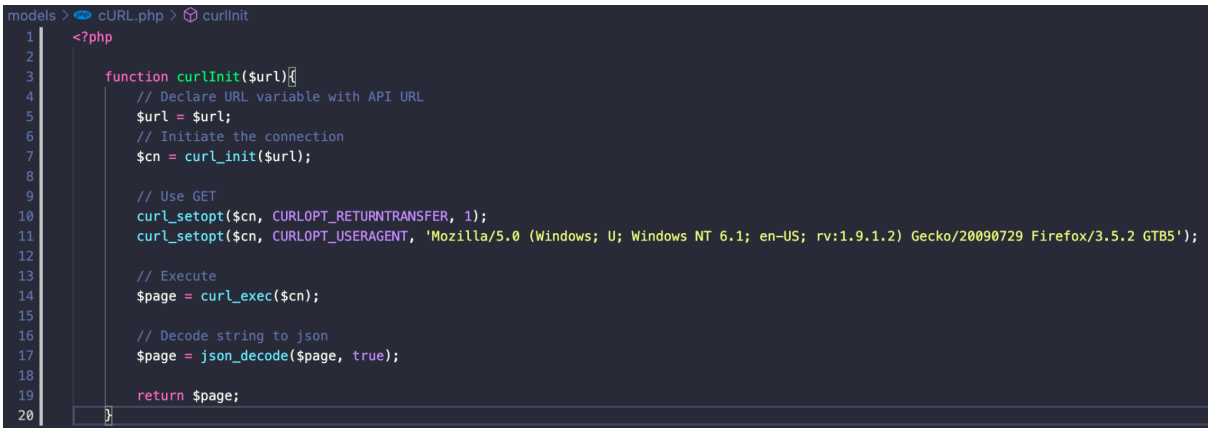
Beregning af afstande

Opgaven om at beregne afstanden fra et fortidsminde til en facilitet i form af forplejning, var tiltænkt at skulle løses via brugen af et ekstra API som kunne give os disse faciliteter, i kombination med at vi skulle omregne afstanden mellem faciliteten og fortidsmindets koordinater, til meter. En sådan formel eksistere nemlig, og ville kunne give os afstanden i fugleperspektiv. Samme formel ville kunne bruges til at lave en radius beregning, da vi så kunne holde to sæt koordinater op mod hinanden, og be- eller afkræfte om de ligger inden for radiussen som brugeren har angivet. Hvis de matcher kriterierne, skulle de så vises som ekstra detaljer omkring det valgte fortidsminde.

Analyse

cURL

For at skabe forbindelse til alle de API'er vi nu engang skulle arbejde med, benytter vi cURL til at skabe denne forbindelse. cURL står for Client URL, og bruges til at oprette en forbindelse til det valgte API, som er angivet som en URL.
Hernæst sætter vi de options som skal bruges, ud fra den metode man vælger at anvende, af hhv. POST, PUT, GET eller DELETE.
Når options er sat, eksekvere vi forbindelsen til API'en via curl_exec, som returnere data til os fra API'et, som vi hernæst gemmer i en variabel og decoder til JSON format.



Konstruktion

Vi ville gerne have delt alt koden op i bidder, og holdt det hele adskilt i form af classes for sig, views for sig etc., men vores approach til projektet handlede først og fremmest om at skrive noget som virkede, og derfor har vi ikke prioriteret tiden til at klippe i koden efterfølgende. Alt koden er derfor placeret i stort set samme fil, på nær vores curlInit funktion der skaber forbindelse til API'et.

// Forloop til at lave et array af kommuner hvor vi pusher kommunernes navne i vores array, efterfulgt af at fjerne duplicates.

```
18      // $kommuner = findKommune($page);
19      ######## Find all names of regions with Fortidsminder ############
20      $kommuneArray = [];
21
22      // Find regions with fortidsminder
23      for ($i = 0; $i < count($page); $i++) {
24
25          // declare value of $navne
26          $navne = $page[$i]["kommuner"][0]["navn"];
27
28          // push $navne to $kommuneArray
29          $kommuneArray[] = $navne;
30      }
31
32      // filter regions from dublicates
33      $kommuner = array_unique($kommuneArray, SORT_STRING );
34      // // print filtered regions
35      // // pre($kommuner);
36
37      ##########################################################################
```

Gruppe 3

// Forloop til at lave et array af fortidsmindernes undertype kategori, hvor vi pusher undertyperne ind i vores array, efterfulgt af at fjerne duplicates.

```php
    ########################################################################

    ######## Find all names of regions with Fortidsminder ############
    $undertypeArray = [];

    // Find regions with fortidsminder
    for ($i = 0; $i < count($page); $i++) {

        // declare value of $navne
        $undertype = $page[$i]["undertype"];

        // push $navne to $kommuneArray
        $undertypeArray[] = $undertype;
    }

    // filter regions from dublicates
    $undertyper = array_unique($undertypeArray, SORT_STRING );
    // print filtered regions
    // pre($kommuner);

    ########################################################################
```

// Angiver brugernes valgte værdier fra POST som variabler, og tjekker om disse er defineret eller ej, for at undgå warnings

```php

    $selectedKommune = isset($_POST["selectedKommune"]) ? $_POST["selectedKommune"] : '';
    $selectedUndertype = isset($_POST["selectedUndertype"]) ? $_POST["selectedUndertype"] : '';
    $selectedMatch = isset($_POST["primærtnavn"]) ? $_POST["primærtnavn"] : '';

```

Gruppe 3

// Inline javascript
// Scriptet indeholder vores koordinator fra et fortidsminder (hvis disse er sat), som vi plotter ind i et object ved navn coordinates. Disse koordinator bruger vi til at lave en ny instance af et Google Maps, som placere en marker der passer på det valgte fortidsminde.

```
<!-- Async script executes immediately and must be after any DOM elements used in callback. -->
<script
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyAPvo3EAiQ21sCVzDqhIcnOjvKjAF8-VU8&callback=initMap&v=weekly&channel=2" async>
</script>

<script>
    if (lng === null && lat === null){
        console.log('this is null');
    }
    lng = parseFloat(document.getElementById("lng").innerHTML) ? parseFloat(document.getElementById("lng").innerHTML) : lng;
    lat = parseFloat(document.getElementById("lat").innerHTML) ? parseFloat(document.getElementById("lat").innerHTML) : lat;

    console.log(lng);
    console.log(lat);

    // Initialize and add the map
    function initMap() {
    // The location of coordinates
    const coordinates = {
        lat: lat,
        lng: lng
    };
    // The map, centered at coordinates
    const map = new google.maps.Map(document.getElementById("map"), {
        zoom: 6,
        center: coordinates,
    });
    // The marker, positioned at coordinates
    const marker = new google.maps.Marker({
        position: coordinates,
        map: map,
    });
    }
</script>
```

Evaluering af proces

Vi alle 3 har været meget engageret og interesseret i at arbejde og udvikle vores kompetencer inden for programmerings feltet, og derfor har gruppedynamikken fungeret upåklageligt. Det har været nemt at samarbejde med hinanden både med opgavefordeling samt fælles opgaver, hvor vi løbende har brainstormet og live coded foran hinanden, med henblik på at problemløse i fællesskab. Vi valgte at arbejde med PHP, som viste sig at være lidt udfordrende for os, pga. at vi skulle vænne os til sproget igen.
Projektet har desuden været drilsk at gennemfører, da vi har haft problemer med at finde et API som kunne give os data over forplejnings muligheder, som vi kunne holde op mod geo lokationen på fortidsminderne.
Alt i alt, så har vi nået at lave en præsentation af vores tankegang bag hele projektet, hvor der er plads til forbedring og videreudvikling, som var noget vi kunne arbejde videre på hvis vi havde mere tid.

Konklusion

Vores kompetencer vedrørende programmering i PHP, samt brugen af data integration er forbedret, grundet vores fokus på at prioritere det fremfor et design. Vi har i fællesskab opnået bedre forståelse for behandlingen af data. Vi har dog ikke nået at løse opgaven i dens fulde helhed, da vi stødt panden mod muren med hensyn til brugen af AJaX. Vi er dog blevet klogere på hvad man skal gøre for at løse en sådan fejl, som vil bruge til fremtidige opgaver.

Referencer

https://www.kulturarv.dk/fundogfortidsminder/
https://dataforsyningen.dk/
https://dawadocs.dataforsyningen.dk/