B1-K3-W1 – Optimalisatie van het product – Pjotr Wisse – Video Tutorial Website

In dit deel van het document zal ik omschrijven hoe we ons product hebben geoptimaliseerd, met de verzoeken van de klant. We zullen een overzicht maken van hoe de applicatie in elkaar zit en welke talen en functies wij gebruiken om de applicatie zo veel mogelijk te optimaliseren.

Optimalisatie van de applicatie

Bij een applicatie wil je graag dat alles vlot, snel en flexibel is. Niemand wil graag een slome website gebruiken en daarom is het ook belangrijk om je webapplicatie te optimaliseren. Dit kan worden bereikt via verschillende manieren. Het gebruik maken van de juiste software, optimaliseren van programeer talen zoals javascript, PHP en SQL of het ontwerp van de website gemakkelijker en toegankelijker te maken. Bij het optimaliseren van de applicatie hebben we tools gebruikt uit onze andere projecten, omdat die het beste voor ons zouden werken.

Optimalisatie van de applicatie met software

Ten eerste, moesten we beslissen, met welke software gaan we werken? We hebben besloten om met XAMPP te werken voor onze server, omdat XAMPP alle noodzakelijke componenten heeft voor een normale webserver. XAMPP bevat:

- Een Apache server
- PHP
- Perl
- Een Mysql database
- Phpmyadmin als databasemanagement programma (meer informatie bij beheert gegevens).

We gaan ook gebruik maken van twee git software 's. De officiële Git software (https://git-scm.com) en GitKraken (https://www.gitkraken.com). De eerste voor als we kleine onderdelen willen toevoegen en aangezien we op Windows werken kunnen we deze git software makkelijk gebruiken aangezien we de command line kunnen gebruiken. De ander gebruiken we om te werken met verschillende platformen en computers aangezien we op school geen command line software kunnen installeren en wel Git Kraken.

Optimalisatie van de applicatie met editors

Ten tweede, met welke software/code editor gaan we de website maken? Het besluit was om Visual Studio Code te gebruiken voor een paar redenen:

- **Belangrijk:** Het is overal toegankelijk op elk apparaat (elk besturingssysteem ondersteunt Visual Studio code)
- Het heeft geïntegreerde git extensie en is makkelijk te combineren met GitHub
- Veel andere downloadbare extensies
- **Belangrijk**: De extensie Live Share een extensie die ervoor zorgt dat developers aan een gedeeld project kunnen werken in een sessie die wordt gehost op een lokale computer. Plus je kan ook je local server delen. Meer info: (https://visualstudio.microsoft.com/services/live-share/)

Het belangrijkste is dat Visual Studio code ook je code optimaliseert en het Visual is handig als development tool aangezien je veel functies en extensies kan gebruiken.

Pjotr heeft ook voor een gedeelte Sublime Text gebruikt voor kleine veranderingen of toevoegingen die in de applicatie pasten.

We hebben ook gebruik gemaakt van de tinymce editor, we hadden hier al eerder gebruik van gemaakt, maar we wilde nog meer functionaliteit bij toevoegen.

Optimalisatie van de applicatie met programeer talen.

Als laatste hoe gaan we onze applicatie optimaliseren met de programeer talen die we gaan gebruiken. In ieder geval gaan we gebruik maken van Javascript, PHP, HTML,

CSS en SQL als de talen die we gaan gebruiken in onze applicatie. Hier zijn een paar optimalisatie manieren die wij hebben gebruikt in onze applicatie.

Mysql is een heel vaak gebruikt door meerdere bedrijven omdat optimalisatie op mysql heel belangrijk is. In mysql heb je een paar opties om zelf je SQL-code te optimaliseren. We hebben een paar van deze optie gebruikt.:

- Gebruik geen * in je select clause, hierdoor zorg je ervoor dat je minder data moet inladen dat onnodig is.
- Zorg in PHP voor prepared statements, dit voorkomt SQL-injecties.
- Gebruik zo min mogelijk functies in je query, anders duurt de query langer.

In PHP hebben we ook een paar manier gevonden:

- Prepared statements, hiermee voorkom je niet alleen SQL-injecties, het zorgt er ook voor dat je code sneller wordt uitgevoerd.
- Gebruik bij het checken zo min mogelijk if statements, het zorgt ervoor dat het PHP sneller wordt geëxecuteerd.
- Bij select functies kunne we ook prepared statements uitvoeren en dan een store result functie toevoegen zo kunnen we het resultaat aan variabelen toevoegen.
- Voor de veiligheid hebben we er ook voor gezorgd dat we de strip tags en htmlspecialchars functies gebruikt, omdat het snel is en het zorgt ervoor dat er geen scripts kunnen worden uitgevoerd of iets dergelijks.

```php
if ($stmt = $conn->prepare("SELECT titel,text, image, video, leerlijn FROM subject WHERE id = ?")) {
    $stmt->bind_param("s", $_GET["id"]);
    $stmt->execute();
    $stmt->store_result();

    if ($stmt->num_rows > 0) {
        $stmt->bind_result($titel, $text, $image, $video, $leerlijn);
        $stmt->fetch();
    } else {
        header("Location: index.php");
    }
    $stmt->close();
}
```

Voorbeeld van een prepared statement.

- We hebben een nieuwe functie toegevoegd tot onze website en dat is een search engine. De search engine is niet bepaald slim, maar zoekt wel naar specifieke lesstoffen in de database. We gebruiken huidig hiervoor de LIKE functie in Mysql. Er is nog ook een andere manier om een search engine te maken, maar dat komt misschien in de toekomst wel.

```php
$sql = "SELECT `id`, `titel`, `image`, `subtext` FROM subject WHERE `titel` LIKE '%". $_POST['search'] ."
    %';";
$result = $conn->query($sql);
```

Voorbeeld van een search query.

In HTML kun je de laadtijd van je webpagina optimaliseren. Je kan gebruik maken van:

- Zo min mogelijk inline CSS en javascript scripts in je document zetten, het is beter om externe bestanden te gebruiken voor je webpagina, omdat dan de gebruikers browser die scripts in cache opslaat. Zo hoeft de gebruiker niet nog een keer de scripts te laden als de pagina weer wordt herladen.
- Zorgen dat sommige gedeeltes van je websites code wordt opgeslagen op derde partijen websites. Dat hebben we bijvoorbeeld gedaan met onze Tinymce.
- Om SEO te gebruiken maken we ook gebruik van meta tags om in search engines terecht te komen.
- Zorg ook voor minder CDN HTTP request 's hierdoor kan de pagina sneller geladen worden.

Javascript heeft meerdere manieren om code te optimaliseren een van die opties is de library JQuery, hierdoor hoef je minder code te schrijven en is je bestand ook niet zo groot. Daarnaast is JQuery zo gemaakt dat de code al gelijk wordt geoptimaliseerd. We gebruiken minder javascript dan bijvoorbeeld PHP, maar dat is ter zake. We zouden JQuery kunnen gebruiken voor bijvoorbeeld AJAX, maar dat hebben we niet kunnen toevoegen. We waren vooral bezig met het optimaliseren van de PHP-code en de werkprocessen te voltooien.

Om CSS te optimaliseren hebben we een CSS framework gebruikt. Bootstrap, bootstrap heeft allemaal ingebouwde functies en stijlen dat we nauwelijks met CSS hoefde te werken. Bootstrap heeft ook ingebouwde JQuery tags die je kan gebruiken in je html pagina. Je kan ook zelfs CSS-variables veranderen in de bootstrap source als je de bestanden ook lokaal hebt, dat hebben we ook gebruikt.

Hieronder zijn nog een paar voorbeelden van optimalisatie bij de site met plaatjes.

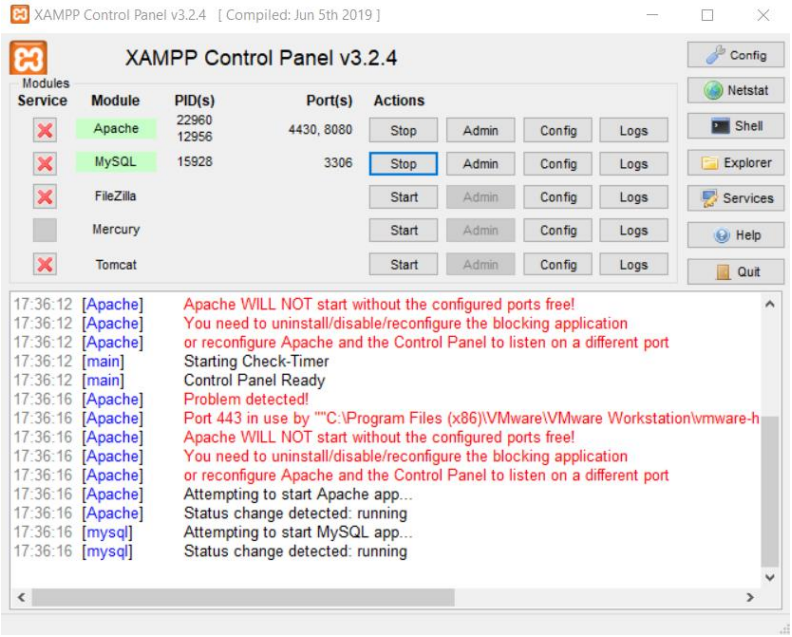
```
if ($stmt = $conn->prepare("SELECT titel,text, image, video, leerlijn FROM subject WHERE id = ?")) {
    $stmt->bind_param("s", $_GET["id"]);
    $stmt->execute();
    $stmt->store_result();

    if ($stmt->num_rows > 0) {
        $stmt->bind_result($titel, $text, $image, $video, $leerlijn);
        $stmt->fetch();
    } else {
        header("Location: index.php");
    }
    $stmt->close();
}
```

Hier zie je dat we bij select geen * gebruiken.

```
require('php/database.php');
```

Gebruik maken van require om gedeeltes van pagina's niet te herhalen.



XAMPP is heel makkelijk om te gebruiken en is een nog best wel flexibele server applicatie die je kan gebruiken voor een webserver.

```
tinymce.init({
    selector: '#text',
    height: '480',
    plugins: [
        'advlist autolink lists link image charmap print preview anchor',
        'searchreplace visualblocks code fullscreen',
        'insertdatetime media table paste code help wordcount'
    ],
    toolbar: 'undo redo | formatselect | ' +
        'bold italic forecolor backcolor | alignleft aligncenter ' +
        'alignright alignjustify | link unlink | bullist numlist outdent indent | ' +
        'removeformat | help',
    /* enable title field in the Image dialog*/
      image_title: true,
      /* enable automatic uploads of images represented by blob or data URIs*/
      automatic_uploads: true,
      /*
        URL of our upload handler (for more details check: https://www.tiny.cloud/docs/configure
        /file-image-upload/#images_upload_url)
        images_upload_url: 'postAcceptor.php',
        here we add custom filepicker only to Image dialog
      */
      file_picker_types: 'image',
      /* and here's our custom image picker*/
      file_picker_callback: function (cb, value, meta) {
        var input = document.createElement('input');
        input.setAttribute('type', 'file');
        input.setAttribute('accept', 'image/*');

        /*
          Note: In modern browsers input[type="file"] is functional without
          even adding it to the DOM, but that might not be the case in some older
          or quirky browsers like IE, so you might want to add it to the DOM
```

Tinymce om gemakkelijk tekst te creëren met javascript. We hebben ook lokale images upload toegevoegd als functie om ze toe te voegen in de tekst.

Dat was het einde van het document over hoe we onze website optimaliseren om gebruikers en de beheerders een snellere en fijnere ervaring te geven.

B1-K2-W1 – Realisatie van het product – Pjotr Wisse – Video Tutorial Website

In dit deel van het document zal ik omschrijven hoe de realisatie zal gedaan worden. Hier zal omschreven worden welke software, talen, editors en databases wij gaan gebruiken om zo de website op te bouwen.

Software die we gaan gebruiken

In deze opdracht hebben we niet zoveel software nodig om deze applicatie mogelijk te maken. Eerst wat we gaan we een webserver gebruiken, hiervoor gaan we XAMPP gebruiken, omdat XAMPP gratis is en het is makkelijkst te gebruiken voor simpele websites en servers. Het bevat een apache server, mysql support, PHP-support en dat zorgt er ook voor dat we met de MYSQLi extensie kunnen werken in PHP. Hierop zullen we vooral de applicatie en de database testen, zodra we klaar zijn gaan we het hosten op een van onze servers. Met plesk kunnen we ervoor zorgen dat de databasen en de website toegankelijk zijn via een domainnaam.

Voor de database gebruiken we phpmyadmin als databasemanagement software. Hierdoor kunnen gebruikers makkelijker in de gaten houden hoe de database eruitziet en wat er allemaal op staat. Phpmyadmin ondersteund onze mariadb database die wij gaan gebruiken voor onze applicatie.

Programeer talen die we gaan gebruiken.

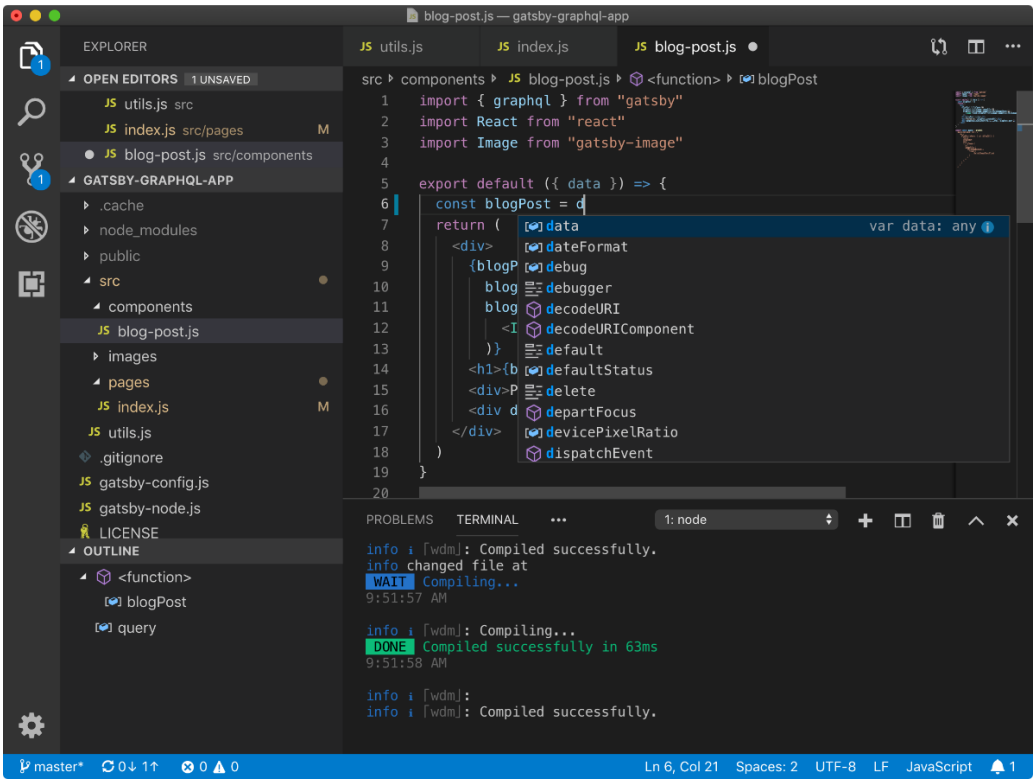
Met software zoals mysql en phpmyadmin kun je al verwachten dat we voor het grootste deel met PHP en SQL gaan werken. PHP en SQL zijn talen die makkelijk te gebruiken zijn als je een webapplicatie wilt maken met een database. PHP kan ervoor zorgen dat je query's en code veilig wordt verstuurd naar je mysql database, via checks, prepared statements en encryptie protocollen zoals sha1. SQL is ook handig omdat structureel de taal makkelijk te begrijpen is en het is gestructureerd aangegeven hoe je de data makkelijk kan krijgen. Naast PHP en SQL gaan we ook andere talen gebruiken zoals HTML, CSS en Javascript. Javascript is handig voor scripts bij bepaalde pagina's zoals de contact pagina en met HTML en CSS kan de site mooi gestileerd worden.

Dit zijn alle programeer talen die wij gaan gebruiken in ons project als je een overzicht wilt zien van hoeveel we van deze talen gebruikt hebben kun je het op onze GitHub pagina bekijken: https://github.com/Peter-The-Great/VideoTutorialBlogSite

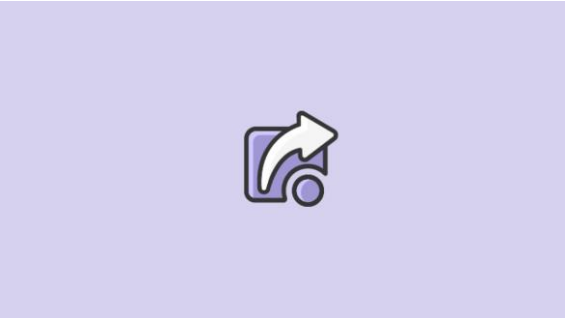
Tools die we gaan gebruiken.

We hebben al gekeken naar welke belangrijke software we gaan gebruiken voor de database, maar in dit gedeelte gaan we bespreken met welke tools we de applicatie gaan bouwen en welke tools we zullen gebruiken.

In dit project zullen we gebruik maken van Visual Studio Code met visual studio code kunnen we een mooi overzicht creëren van ons project, visual studio code heeft ook een geïntegreerde GitHub en git ondersteuning (komt deels omdat het van Microsoft is visual en GitHub).



Voorbeeld van visual studio code in actie



Met visual kunnen we ook live sharen, dat is een extension waarmee developers aan een project kunnen werken die alleen maar op een computer hoeft te staan. Je deelt je project met andere via een sessie en iedereen kan code toevoegen, veranderen en verwijderen en ze kunnen je webserver bezoeken. Visual studio code is ook handig voor debuggen, syntax correctie en automatische suggesties om zo de code sneller te laten werken. Er zijn op visual studio code nog allemaal een boel andere extensie die we kunnen gebruiken om ons project beter te maken, maar die zijn niet zo belangrijk als deze.

We maken ook gebruik van Git software, waaronder GitKraken. Met GitKraken kunnen we op elke computer onze git repository beheren. Het heeft een duidelijke GUI en is makkelijk te verbinden met je GitHub account. We gebruiken daarnaast ook nog de normale Git Command Line software op onze Windows besturingssystemen.

Back-up en versiebeheer

Als het gaat om versiebeheer en back-up gebruiken we altijd onze vertrouwelijke git, GitKraken en GitHub services. GitHub geeft ons mogelijk om onze code op te slaan als back-up en als toegankelijkheid, zodat we er altijd toegang tot hebben. GitHub heeft een heel makkelijke manier repositories te maken en geeft je instructies om hoe je een remote server adres kan instellen via de git command line. Het is voor ons belangrijk en voor de klant belangrijk om altijd toegang te hebben tot de code op een veilige en toegankelijke manier.

Voor de database hadden we bijna een grootte fout gemaakt, alles op de database was kwijtgeraakt, maar gelukkig was er wel een back-up gemaakt van de server zelf en alle informatie in de database alleen niet de database van de opdracht. Gelukkig konden we nog een test database overzetten naar de database en toen hadden we alles weer teruggezet naar hoe het hoorde. Om deze gebeurtenis is het voor ons dus heel belangrijk om altijd van onze database een back-up te maken. Meestal maken we op een hard kopie van de gehele XAMPP-server of alleen een SQL-bestand van de hele database.

```
CREATE TABLE `subject` (
  `id` varchar(64) NOT NULL,
  `titel` varchar(64) NOT NULL,
  `text` longtext NOT NULL,
  `subtext` text NOT NULL,
  `video` varchar(48) NOT NULL,
  `leerlijn` varchar(64) NOT NULL,
  `date` datetime NOT NULL DEFAULT current_timestamp(),
  `uitgelicht` int(1) NOT NULL,
  `image` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Gegevens worden geëxporteerd voor tabel `subject`
--

INSERT INTO `subject` (`id`, `titel`, `text`, `subtext`, `video`, `leerlijn`, `date`, `uitgelicht`, `ima
('02b95889-5448-47e4-8ee1-bff1d5574483', 'PHP is niet PHP', '<p>In dit document ga ik het hebben over ha
('5b59a25d-4ec6-4fc2-8e9a-d246100d69cf', 'Cool', '<p>Top gear</p>', '<p>Top gear</p>', 'cDoRmT0iRic', '6
('64f8385d-958a-494a-b916-e0a62027ce67', 'PHP is niet PHP', '<p>In dit document ga ik het hebben over ha
('cac348f9-4492-46ec-9a22-c492d7d97e98', 'Top gear', '<p>Gearsssssssssssss</p>', '<p>Top gearsssssssssss
```

Voorbeeld van SQL back-up van de tabellen in de database

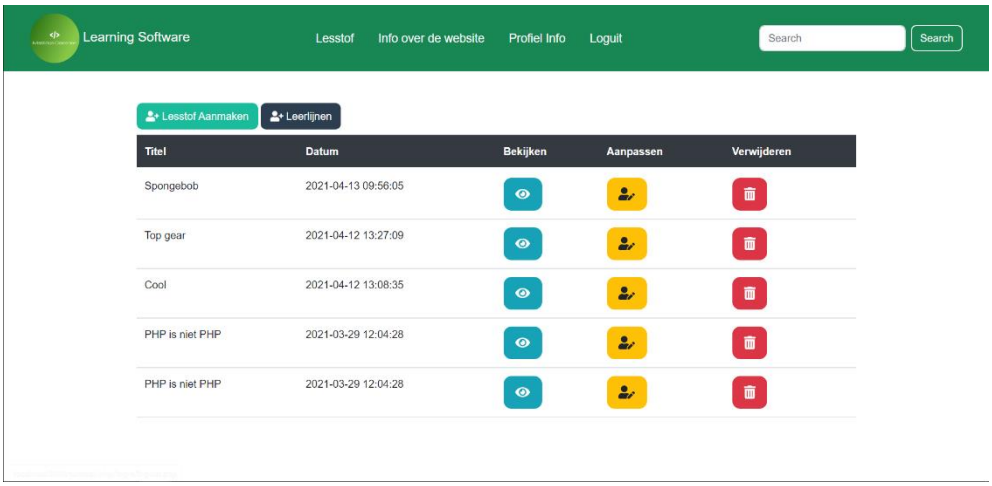
In kort we hebben vooral GitHub gebruikt als versiebeheer en Back-up. Daarnaast gebruiken we ook SQL-bestanden om al onze database informatie op te slaan, zodat we het later kunnen gebruiken en kunnen zetten op andere database.

B1-K2-W2 – Test het ontwikkelde product – Pjotr Wisse – Video Tutorial Website

In dit deel van het document zal ik omschrijven hoe we het product hebben getest. Hier wordt per onderdeel omschreven, wat we getest hebben van het product en hoe we dat hebben gecreëerd. Verder zal ik een gedetailleerde omschrijving geven van elk gedeelte wat we hebben getoetst. Hier worden 4 belangrijke onderdelen beschreven.

Admin Pagina en Nieuwe Functies

We hebben onze admin pagina voor het grootste deel nieuwe functies gegeven. Het design blijft voor het grootste gedeelte hetzelfde, omdat we niet echt vonden dat het nog beter komt.



Je krijgt nu wel wat extra informatie zien op het dashboard zoals de lesstof en andere gedeeltes van je posts, maar we hebben wel iets nieuws toegevoegd aan de code van de website.

```
//here we are making sure that the image is moved to its right location.
$afbeelding = $map.$imagenaam;
$new_str = str_replace(' ', '', $afbeelding);
$fileExt = pathinfo($_FILES['image']['name'], PATHINFO_EXTENSION);
$new_str = $map . uniqid() . "_" . uniqid() . "." . $fileExt;
if (in_array($type,$Toegestaan)){
    move_uploaded_file($Tijdelijk, "../".$new_str);
}else{
    header("Location: createpost.php?error=nietgeupload");
}
```

Wanneer je nu een background afbeelding uploadt krijg je een automatische UUID (uniek ID) toegewezen. Hier wordt dus via de UUID-functie in PHP een ID van gemaakt.

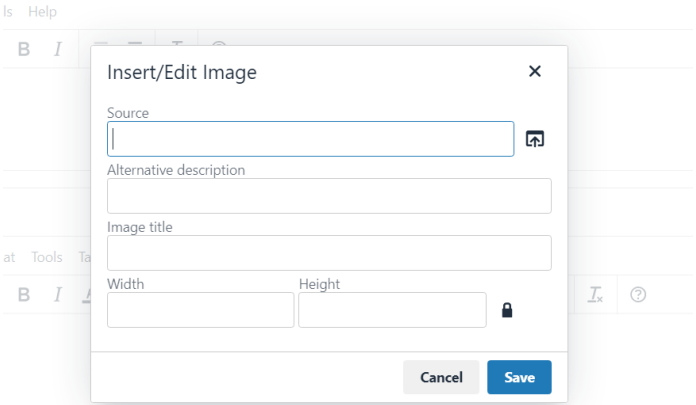
```
function uuidv4(){
    $data = openssl_random_pseudo_bytes(16);

    $data[6] = chr(ord($data[6]) & 0x0f | 0x40);
    $data[8] = chr(ord($data[8]) & 0x3f | 0x80);

    return vsprintf('%s%s-%s-%s-%s-%s%s%s', str_split(bin2hex($data), 4));
}
```

We doen de ID voor de post nog wel zelf met onze custom UUID-code. Alles werkt als het gaat om een UUID aanwijzen aan een afbeelding of property in een database.

Ook bij het toevoegen en veranderen van een post kun je nu ook lokale afbeeldingen toevoegen in de tekst, bij de Tinymce editor. Dit geeft de gebruiker iets meer vrijheid over wat hij in de tekst wil zetten en is dus niet gelimiteerd tot afbeeldingen linken.



Vlak bij de source balk zie je een upload icon, hierbij kun je vanaf je eigen computer afbeeldingen uploaden.

We hebben ook meer beveiligingsfuncties toegevoegd tot de admin website. Waaronder een tijd sessie variabele. Hiermee wordt na een bepaald stuk tijd je sessie afgebroken om zo de veiligheid van de website in stand te houden.

```
if($_SESSION['ip'] !==  $_SERVER['REMOTE_ADDR']){
    session_unset();
    session_destroy();
header('Location: index.php?error=notthesameipadress');
}
if((time() - 1200) > $_SESSION['timestamp']){
    $_SESSION['loggedin'] = FALSE;
}else{
    $_SESSION['timestamp'] = time();
}
if($_SESSION['loggedin'] !== TRUE){
    session_unset();
    session_destroy();
    header("Location: index.php?telangnietdepaginagebruikt");
}
```

We zien hier dat er een sessie is voor je IP dus je IP wordt ook gecheckt en op je timestamp. Als de tijd langer duurt dan de aangegeven tijd dan moet je opnieuw inloggen.

We hebben daarnaast een token toegevoegd die checkt of je wel op de website een post aan het maken, wijzigen of verwijderen bent. We maken hier gebruik van een 32 lange karakter dat wordt gegeneerd van willekeurige bytes.

```
$token = bin2hex(openssl_random_pseudo_bytes(32));
$_SESSION['token'] =  $token;
```

We vinden dat de veiligheid en de stabiliteit van de website verbeterd is. Via het gebruik maken van de sessie tijd, token checks en de UUIDS kunnen wij zelfverzekerd zeggen dat we een veilige omgeving hebben gecreëerd voor de klant.

YouTube Integratie

We hebben de klant de mogelijkheid gegeven om zijn YouTube video's te integreren in zijn posts. Het moet alleen wel een YouTube link hebben van de video.

YouTube links kunnen heel erg verschillen van lengte en informatie. Het enige wat we echt nodig hebben van de link is het ID. Dus hoe halen we het ID uit de link? Heel simpel, Eerst zorgen we ervoor dat het eerste gedeelte van de link (https://www.youtube.com/watch?v=) verwijderd wordt via string_replace. Daarna splitsen we de query op (explode) als er nog iets naast de string van het ID is en dan neemt PHP die op in een array. Als laatst halen we alleen het ID uit de array.

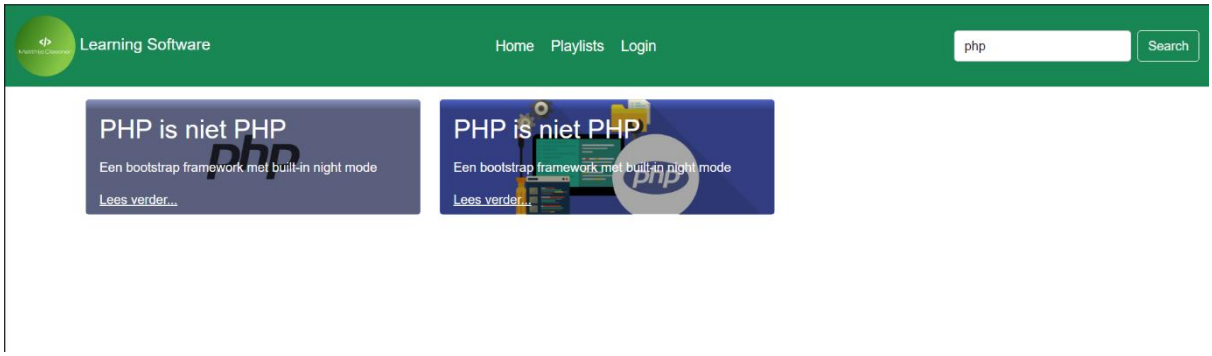
```
//Get the video string
$videostring = str_replace('https://www.youtube.com/watch?v=', '', $_POST['video']);
$videostring1 = explode('&', $videostring);
$vidstring = $videostring1[0];
```

Voorbeeld van het pakken van de string.

Uiteindelijk wordt de link opgeslagen in de database. Zo wordt er makkelijk een YouTube video in de post gezet. We zijn er zelf heel tevreden mee. We bedanken Floris voor het helpen met dit gedeelte van de website.

Zoek functie

In de website hebben we met iets totaal nieuws geëxperimenteerd. We hebben de zoek functie uitgebracht. Hiermee kan de gebruiker een specifieke taal of woord zoeken en dan kan zien of het resultaten oplevert.



We hebben dit gemaakt via gebruik van de SQL-functie LIKE. Hiermee kunnen we kijken naar verschillende woorden die misschien in de titel staan van de post. Het is

niet een zoekmachine op hetzelfde niveau als Google, maar het is voor ons een echte stap in de juiste richting. We kunnen hieronder zien dat we zoeken via de LIKE functie met de %%. Hier zoek je dus voor een woord of zin die voorkomt in de database. Het is heel handig voor kleine databases, maar niet bedoeld voor Google-websites.

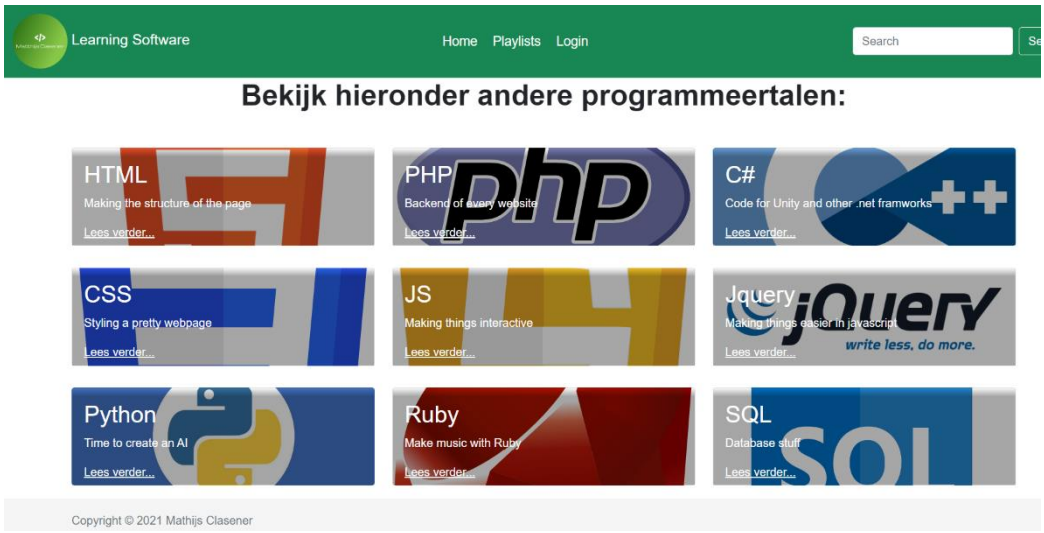
```
$sql = "SELECT `id`, `titel`, `image`, `subtext` FROM subject WHERE `titel` LIKE '%". $_POST['search'] ."%';";
$result = $conn->query($sql);
```

Dit is een voorbeeld van de zoek query.

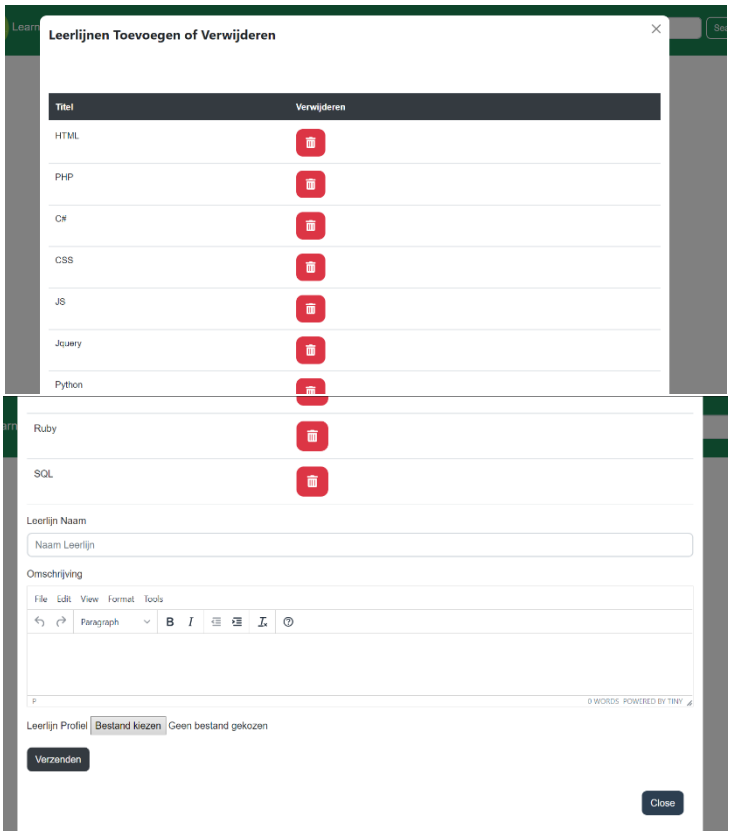
Dit was een van onze favoriete gedeeltes om aan te werken. We vonden het best een leerzaam gedeelte van de opdracht en we hebben ervoor gezorgd dat het nog werkt ook.

Playlist lijsten

Met de playlists pagina krijg je playlists te zien waarin alle lesstoffen staan van allemaal verschillende programeer talen.



We verdelen onze lesstoffen in playlists door ze een specifieke categorie te geven. Dit kan een programmeertaal zijn of een ander vak wat de klant wil toevoegen aan de website. De gebruiker kan via de admin pagina een nieuwe leerlijn toevoegen, simpel genoeg via een formulier. Deze kan de gebruiker ook heel erg makkelijk verwijderen.

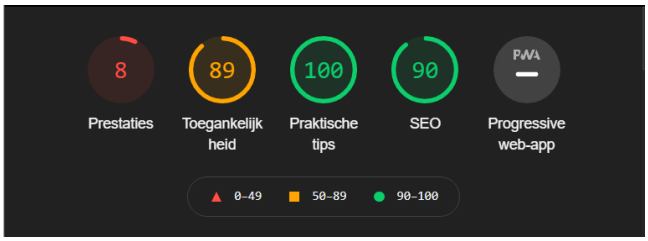


Hier zien we dus de gehele leerlijnen model, waar je leerlijnen kan toevoegen en verwijderen. Deze categorie worden opgeslagen in een tabel op de database genaamd categorie (CAT). Hier wordt ook gebruik gemaakt van de UUID-image en ID zelf.

Dit was ons test document van allemaal verschillende gedeeltes van onze website. We zijn heel erg trots dat we allemaal nieuwe functies konden toevoegen tot onze website en hopen dat u als klant wij een mooi product aan u hebben opgeleverd. Zelf hoop ik dat we samen als leverancier en klant, we samen verder kunnen komen en een betere relatie tussen ons kunnen vormen, bedankt en heb nog een fijne dag.

Extra SEO en light house Experiment

Hier laten we nog een stukje test materiaal van de overal snelheid van de website. We kunnen zien dat we goed gebruik maken van toegankelijkheid, praktische tips en SEO. De prestatie heeft nogal laag gescoord, omdat we dit geëxperimenteerd hadden op een slome computer en zelfs dat lade nog snel. Dus meeste gebruikers zullen geen problemen herkennen.



Floris Werkprocessen

In dit document ga ik mijn werkprocessen bewijzen voor de videotutorialsite.

De werkprocessen zijn:

1. Levert een bijdrage aan het projectplan.
2. Levert een bijdrage aan het ontwerp
3. Bereidt de realisatie voor
4. Onderhoudt een applicatie
5. Beheert gegevens

1. **Levert een bijdrage aan het projectplan:**
   Voor elke werksessie hebben we besproken wie wat gaat doen, dit heb ik vervolgens in de planning gezet en daarna toen we gebeld werden door de opdrachtgever hebben we doorgeven wie welke taken gaat doen die dag.
   Zie hieronder een compleet overzicht van de planning die we voor dit project hebben gebruikt.
   Van te voren hebben we afgesproken dat we elke week bespreken wat de voortgang is en wie wat gaat doen die week.
   Zie hieronder planning die ik elke week heb bijgehouden. Dit heb ik voor/ na elke werksessie bijgewerkt. Dit vond ik een zeer gestructureerde manier om alle taken bij te houden.

Planning Video TutorialSite

Week	TASK NAME	SUB TASK NAME	STATUS	ASSIGNED TO	START DATE	END DATE	DURATION in days	COMMENTS
Week 1	Klantengesprek, offerte maken		Complete	Fjotr	03-07	03-08	1	Fjotr heeft het klantengesprek gevoerd en de offerte gemaakt. Ook besproken wie welke taken gaat doen.
Week 1	IPP maken		Complete	Floris	03-08	03-10	2	Floris heeft het ipp gemaakt. Ook besproken wie welke taken gaat doen.
Week 2	Database aanmaken, offerte/IPP opsturen.		Complete	Fjotr en Floris	03-14	03-15	1	We hebben de database aangemaakt en de offerte/ipp opgestuurd. Ook besproken wie welke taken gaat doen.
Week 2	Files klaarmaken		Complete	Fjotr en Floris	03-15	03-16	1	We hebben de files klaargemaakt zodat we kunnen beginnen aan de code. Ook besproken wie welke taken gaat doen.
Week 3	Home pagina aan beginnen	Cards toevoegen voor de programmeertalen.	In Progress	Fjotr en Floris	03-22	03-23	1	We zijn beginnen aan de homepagina. Ook besproken wie welke taken gaat doen.
Week4	Toevoegen aan de database of alles wordt gerenereert	Naar de post pagina kijken.	In Progress	Fjotr en Floris	03-29	03-30	1	We hebben data toegevoegd aan de database en styling veranderd op sommige pagina's. Ook besproken wie welke taken gaat doen.
Week 4			In Progress					
Week 5	Dashboard maken/css toevoegen		In Progress	Floris	04-05	04-06	1	We hebben de dashboard pagina gemaakt en wat css aangespast van verschillende pagina's. Ook besproken wie welke taken gaat doen.
Week 5	Database updaten		In Progress	Fjotr	04-06	04-07	1	We hebben wat updates gemaakt in de database (tabellen, veldnamen). Ook besproken wie welke taken gaat doen.
Week 6	Dasboard pagina afmaken		In Progress	Fjotr en Floris	04-12	04-13	1	We hebben de dashboard pagina afgemaakt en kunnen nu toevoegen, updaten en verwijderen. Ook besproken wie welke taken gaat doen.
Week6	Comments toevoegen. Video's toevoegen oplossen.		Not Started	Fjotr en Floris	04-13	04-14	1	We hebben wat fouten ontdekt bij het toevoegen van de video's. Dit hebben we ook verholpen die dag.
Week 7	Individuele werkprocessen invullen		In progress	Fjotr en Floris	04-21	04-21	1	We hebben aan de werprocessen gewerkt om die kort na de oplevering te kunnen inleveren.
	Task 11		Not Started		02-11	02-14	3	
	Task 12		Not Started		02-15	02-17	2	
	Task 13		Not Started		02-15	02-17	2	
	Task 14		Not Started		02-15	02-17	2	
	Task 15		Not Started		02-15	02-17	2	
	Task 16		Not Started		02-15	02-17	2	
	Task 17		Not Started		02-15	02-17	2	

2. **Levert een bijdrage aan het ontwerp:**
   Het functioneel, technisch en grafisch ontwerp staat in het aangeleverde IPP. Dit is zo gemaakt dat het voldoet aan de eisen van de klant.

3. **Bereidt de realisatie voor**
   Voor deze website hebben we de volgende programmeertalen gebruikt:
   - HTML5 (dit is nieuwste versie van HTML)
   - CSS3(dit is nieuwste versie van CSS)
   - PHP7.3.27(dit is de versie die ook de schoolserver gebruikt)
   - Javscript ES6 (dit is de nieuwste versie van Javascript).

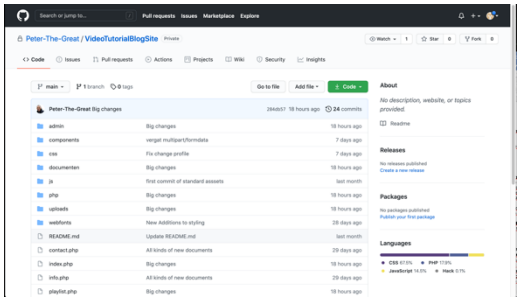
   Daarbij hebben we ook libraries gebruikt zoals:

- Bootstrap 5.0 (dit is de nieuwste beta versie). Hierbij hebben we ook popper gebruikt).
- Font-awesome 5.15.3 (dit is de nieuwste versie)
- Jquery 3.5.1 (dit is de nieuwste versie). Dit is later nog aangepast naar een nieuwere versie.

Verder hebben we ook verschillende software gebruikt zoals:

- Plesk (voor het opslaan van alle bestanden) Dit was de back-up omdat we het ook op een lokale server hadden staan.
- Visual Liveshare (voor het tegelijkertijd werken aan de bestanden).
- Xampp (voor het runnen van de website op een lokale server).

Als laatste hebben we ook github/gitkraken gebruikt op het project in een repository op te slaan en er makkelijk bij te kunnen.



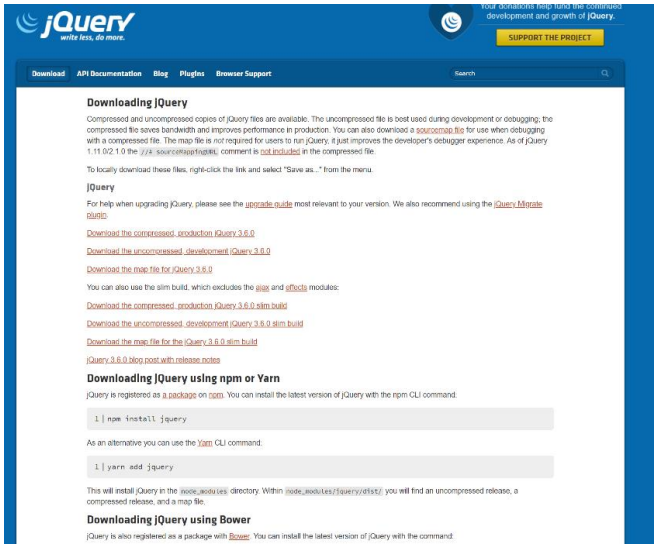
We hebben gewoon de nieuwste versie gebruikt van alle programma's, hardware en software. Zie hieronder de link naar de GitHub pagina waar we alle bestanden hebben opgeslagen. Dit is ook gelijk de back-up van alle bestanden.

https://github.com/Peter-The-Great/VideoTutorialBlogSite
Daarbij hebben we ook alle bestanden nogmaals op de ssd staan.

4. **Onderhoudt een applicatie:**
   Voor dit project hebben we een scripts folder gemaakt waarin we libraries/frameworks toevoegen. Daarna hebben we deze op alle andere pagina's erbij toegevoegd. Voor bootstrap, popper en fontawesome hebben we gewoon de files gedownload van de website en die toegevoegd aan het scripts bestand. Dit geldt ook voor jquery.
   Om dit bij te houden heb ik op de websites gekeken en zag dat er een 3.6.0 versie is dit hebben we dus aangepast in ons project.

4 lines (4 sloc) | 172 Bytes

```
<script src="js/jquery-3.5.1.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/popper.min.js"></script>
<script src="js/fontawsome.js"></script>
```



5. **Beheert gegevens:**
   Zodat de gebruiker gebruik kan maken van de website is het handig dat die de inloggevens heeft. Die zijn als volgt:
   Username: admin
   Password: admin
   Je moet ingelogd zijn om de video's/ afbeeldingen te kunnen toevoegen. Via bovenstaande gegevens heeft de gebruiker de mogelijkheid om dit te doen. Op de website zelf is er een knop rechtsboven in om voor de gebruiker in te loggen.

   Dit was mijn werkprocessen document voor de videotutorialsite.

Einde Document