***Cordova-project 2APP-BIT 2017-2018***

***1. Algemeen***

Voor het vak *Cordova* programmeer je individueel een *Android App*. Het onderwerp mag je zelf kiezen. Zorg er wel voor dat je een onderwerp kiest dat zinvol is als applicatie op een smartphone en/of tablet. Gebruik je fantasie en kies een onderwerp waarmee je creatief aan de slag kan. Dus liefst geen programma met veel gegevensinput, daarvoor is zo'n mobiel toestel sowieso niet geschikt. Maar wel een programma waarmee je de functionaliteiten van je *smartphone* of *tablet* ten volle benut. Je project moet bovendien voldoen aan een aantal minimumvereisten.

***2. Minimumvereisten***

***2. 1 User Interface***

Ontwerp een **SPA-app** met **minstens vier** tabs.

* Eén van die tabs is een info-scherm. In dit scherm komen je persoonlijke gegeven (naam, e-mail, GSM, …) en wat extra informatie over de app.
* Voorzie een SMS en TEL-link naar je GSM-nummer.
* Uiteraard heeft je app een aangepast start-icoon
* Maak ook een aantrekkelijke user interface. De CSS hiervoor kan je volledig zelf schrijven, maar je mag ook vertrekken van een bestaand framework ([Materialize](http://materializecss.com/), [Bootstrap](http://getbootstrap.com/), [Framework7](https://framework7.io/), [Kendo UI](http://www.telerik.com/kendo-ui/open-source-core), [jQuery Mobile](https://jquerymobile.com/), …
* Voor het menu mag je kiezen tussen een **sideNav** of een **knoppenbalk** met icoontjes.
* Gebruik ook **minstens één CORDOVA plug-in** die we **NIET** in de les gezien hebben.  
  <https://cordova.apache.org/plugins/>

***2. 2 Networking (AJAX)***

Heel belangrijk voor de app is dat je ook informatie/data uitwisselt met een **online dienst** of met een **Raspberry Pi** of met een **Arduino**.

* Verwerk in minstens één tabblad data die je **extern, bijvoorkeur json,** ophaalt. Gebruik ook hier een dienst die we NIET in de les hebben besproken.
* Verwerk deze data op een zinvolle manier in je app (maak een grafiek, toon locaties op een map, …).
* Heb je thuis nog een Raspberry Pi of Arduino-project draaien, dan mag je deze ook in je app verwerken (data ophalen en visualiseren, het toestel vanuit je app bedienen, …).

***2.3 Lokale Data***

Naast de gegevens ophalen, moet je app ook gegevens lokaal bewaren. Hiervoor gebruik je best [Local Storage](https://www.sitepoint.com/an-overview-of-the-web-storage-api/) omdat deze omdat deze op bijna alle platformen wordt ondersteund.

***2.4 Applicatie publiceren***

Host op SIN (of elders) een aantrekkelijke webpagina waarin je je **APK-file** aanbiedt. Zorg ervoor dat die pagina soortgelijke informatie aanbiedt als de pagina’s van de *Play Store*.

Je toont op je hosting dus minstens:

* De naam van de app en van de ontwerper.
* Enkele schermafbeelding.  
  Een handige tool hiervoor is <https://www.vysor.io/>
* Een korte omschrijving van de functionaliteiten van de app.
* Een link of QR-code naar APK-file.  
  Bijvoorbeeld: <http://nl.qr-code-generator.com/>

***3. Beoordeling en deadlines***

Bij de beoordeling van je programma wordt gelet op de originaliteit van je onderwerp, de complexiteit van je programma, de grootte van je project, de leesbaarheid/onderhoudbaarheid van je programmacode, de stabiliteit van je programma, de gebruiksvriendelijkheid, het ontwerp van je UI, het al dan niet voldoen aan de minimumvereisten, …

Ga op een doordachte manier na hoe je de online voorbeeldcode in je project verwerkt. Neem deze code niet blindelings over! Optimaliseer deze code voor je project en vertaal ze naar jQuery.

Dus niet: document.getElementById(“abc”).xxxx  
Maar wel: $(“#abc”).yyyy

**Voor een werkend project dat aan alle eisen voldoet, verdien je 50% van de punten. De overige 50% staat op de optimalisatie, structuur en leesbaarheid van je code.**

Daarnaast zijn er ook een aantal deadlines waarop je informatie i.v.m. je app tijdens de les aan je docent moet doorgeven. Een deel van de punten (PE) staat op het correct naleven van die deadlines. Zorg dus dat je in orde bent!

* Bewaar dit document met als titel: **Naam\_Voornaam\_klas\_groep.docx** in de **root van je project**.
* Vul in de koptekst van dit document uw **naam**, **klas, studentnummer, e-mail** en de **titel van het project** in.
* Voeg op pagina 4 een **foto** van jezelf toe.
* Naargelang het project vordert, vul je onderstaande pagina’s stapsgewijs verder aan.   
  Hou rekening met de vooropgestelde deadlines!

**Korte omschrijving (28 februari).**

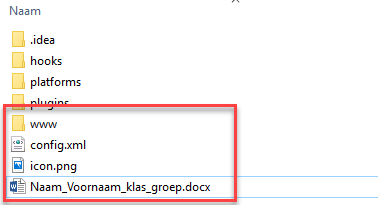
* Beschrijf op pagina 5 kort (10-tal lijntjes) de app die je gaat uitwerken.
* **Uitgeprinte versie** van **pagina 4** afgeven tijdens de les (28 februari).

**Schermafbeeldingen + plug-ins (21maart).**

* Maak van **elk tabblad** in je app een schermafdruk en plak dit op **pagina 5**.   
  Je kan een schermafdruk nemen vanuit Chrome (via chrome://inspect/#devices) of via de gratis versie van [Vysor](https://www.vysor.io/).
* Maak hier een lijstje met alle Cordova plug-ins die in je project verwerkt zijn.  
  Noteer de naam van de plug-in, de code om deze te integreren en beschrijf kort waarom je de plug-in nodig hebt.
* **LET OP**: een plug-in integreren die je altijd met de optie **--save** op het einde!  
  cordova plugin add <plug-in> **--save**Controleer zeker of alle plug-ins in het bestand **config.xml** staan vermeld.
* Pas de korte omschrijving eventueel verder aan.
* **Uitgeprinte versie** van **pagina 4 en pagina 5** afgeven tijdens de les (21 maart).

**Definitieve versie indienen (zondag 22 april)**

* Broncode droppen op ***Toledo*** ten laatste op zondag 22 april om 23h59.
* Vul in het opmerkingveld de URL in waar ik de applicatie kan downloaden (zie info *Applicatie publiceren*).
* Maak één zip-bestand (**Naam\_Voornaam\_Klas\_groep.zip**) met daarin **ENKEL** de bestanden **Naam\_Voornaam\_klas\_groep.docx,**  **config.xml**, **icon.png** en de volledige **www-map**.   
  De overige mappen NIET mee inpakken.



**Verdediging project (april - mei)**

Mee te brengen naar de verdediging:

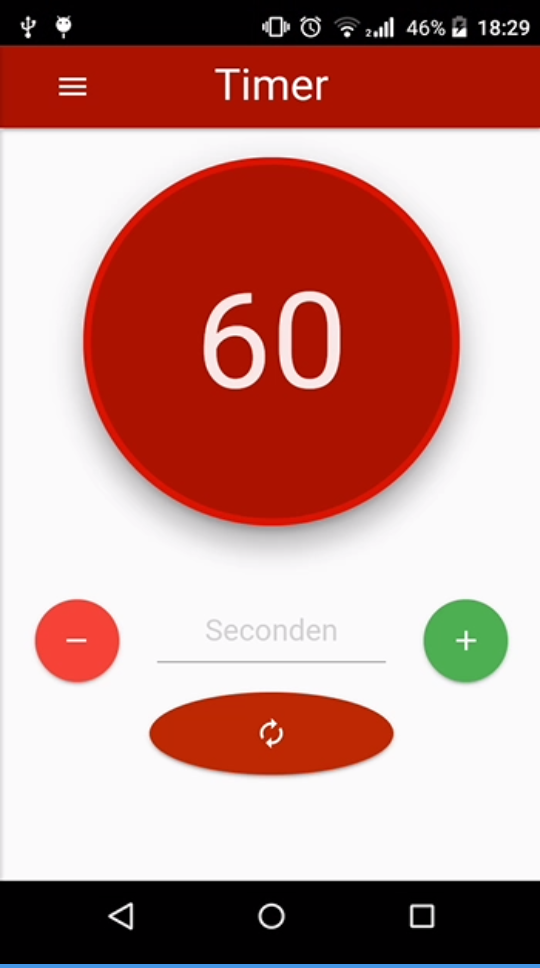
* **Definitieve, uitgeprinte versie** van dit document afgeven tijdens het examen.  
  Vul bovenstaande punten (korte omschrijving, schermafbeeldingen en plug-ins) verder aan.
* Voeg **de broncode** van **elke pagina** toe als bijlage. (Broncode ook uitprinten!)
* Breng alles mee wat je nodig hebt om de app te demonstreren.
* Breng ook je laptop mee zodat je de broncode kan tonen.
* We downloaden en installeren de app op een echt device en testen uit.

**Korte omschrijving van de app.**

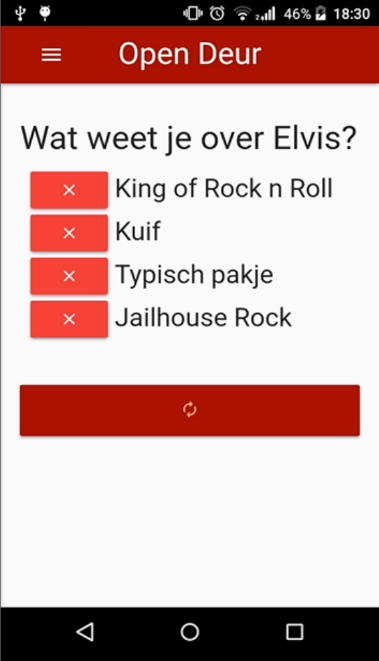
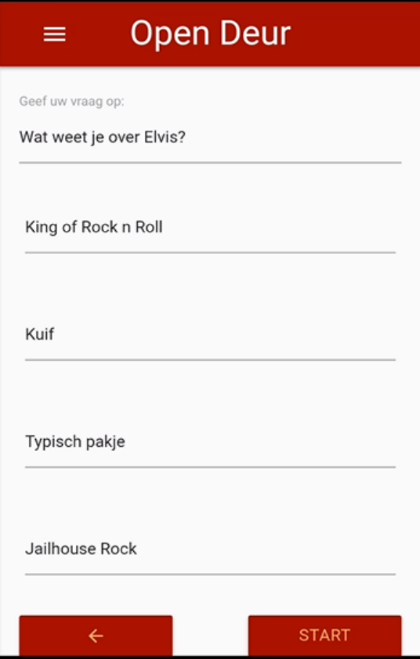
Ik heb onlangs het gezelschapsspel van De Slimste Mens gespeeld. Het spel werkt gewoon met kaartjes en dit zorgde soms voor problemen of dingen die niet efficiënt verliepen. Nu ga ik met mijn app een oplossing bieden hiervoor. Zo komt er voor elke ronde een interface waardoor het gemakkelijk is om punten e.d. bij te houden. Daarnaast komt er ook een timer in om de tijd bij te houden.

**Schermafbeeldingen.**

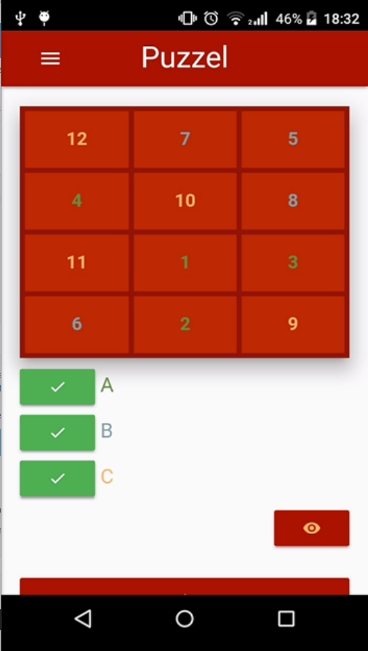
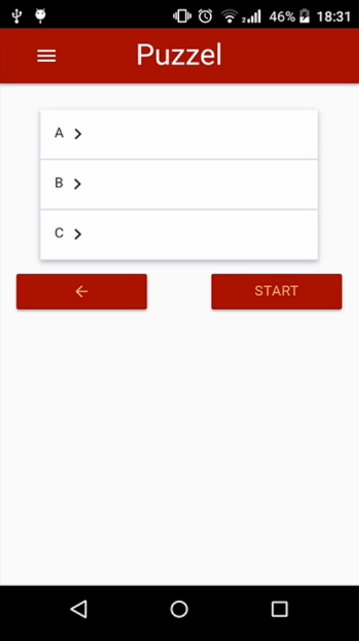
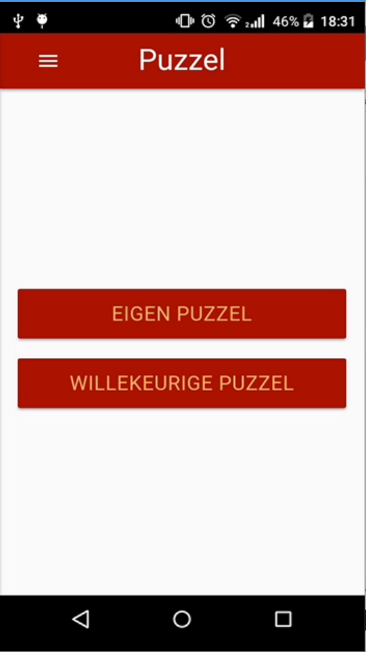
Timer



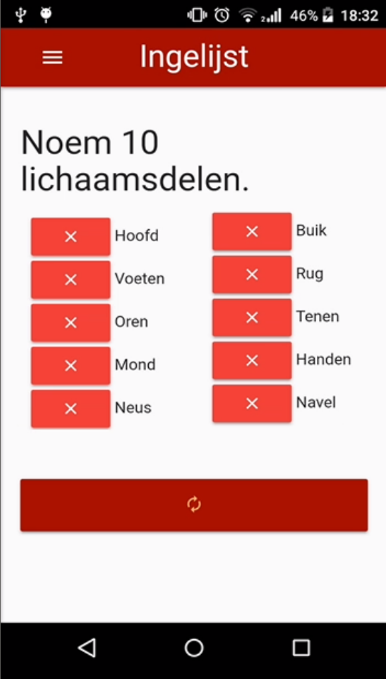
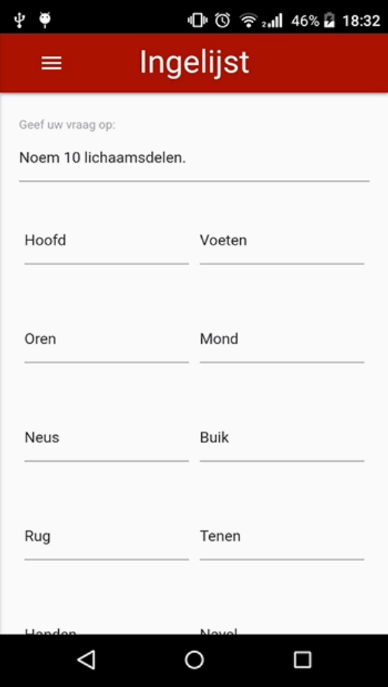
Open deur



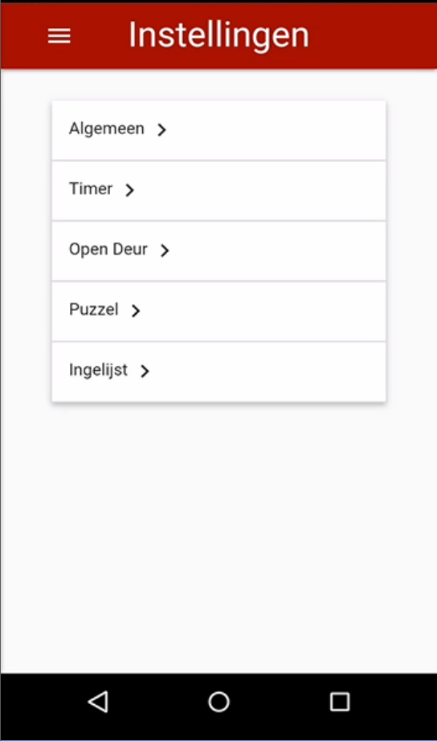
Puzzel



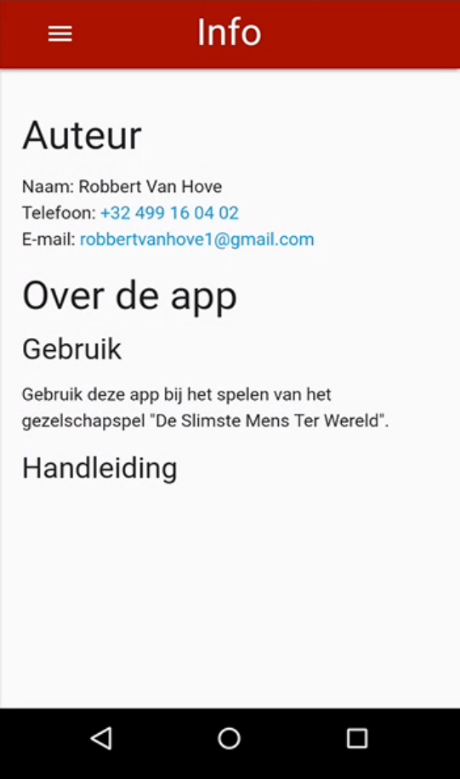
Ingelijst



Instellingen



Info



**Opsomming gebruikte Cordova plug-ins.**

-vibrate: om te trillen als de timer op 0 springt

-media: introsound spelen, geluidjes bij acties

**Bijlage.**

# Index.html

<!DOCTYPE html>

<html lang="nl">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1, width=device-width">

<link rel="stylesheet" href="http://fonts.googleapis.com/icon?family=Material+Icons" />

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/0.100.2/css/materialize.min.css" />

<link rel="stylesheet" href="css/style.css" />

<link href="css/toastr.css" rel="stylesheet" />

<title>DSMTW</title>

</head>

<body>

<!-- Fixed navbar -->

<div class="navbar-fixed">

<nav class="red-background">

<div class="nav-wrapper container">

<a href="#!" id="titel" class="brand-logo center">Timer</a>

<a href="#!" data-activates="nav-mobile" class="button-collapse show-on-large">

<i class="material-icons">menu</i>

</a>

</div>

</nav>

</div>

<!-- Fixed sideNav -->

<ul id="nav-mobile" class="side-nav">

<li>

<a href="#!" data-show="tabTimer" data-title="Timer">Timer

<i class="material-icons right">access\_time</i>

</a>

</li>

<li>

<div class="divider"></div>

</li>

<li>

<a href="#!" data-show="tabOpenDeur" data-title="Open Deur">

<i class="material-icons right">looks\_two</i>Open Deur</a>

</li>

<li>

<div class="divider"></div>

</li>

<li>

<a href="#!" data-show="tabPuzzel" data-title="Puzzel">

<i class="material-icons right">extension</i>Puzzel</a>

</li>

<li>

<div class="divider"></div>

</li>

<li>

<a href="#!" data-show="tabIngelijst" data-title="Ingelijst">

<i class="material-icons right">looks\_4</i>Ingelijst</a>

</li>

<li>

<div class="divider"></div>

</li>

<li>

<a href="#!" data-show="tabInstellingen" data-title="Instellingen">Instellingen

<i class="material-icons right">settings</i>

</a>

</li>

<li>

<div class="divider"></div>

</li>

<li>

<a href="#!" data-show="tabInfo" data-title="Info">Info

<i class="material-icons right">info</i>

</a>

</li>

</ul>

<!--*--------------------------------------------------------------------------*- -->

<!-- Pages -->

<div class="container">

<!-- Timer sectie -->

<div class="row spa" id="tabTimer">

<div class="col s12 center-align">

<!-- TODO:

[x] Timer visualiseren en stylen

[x] functionaliteit start/stop

[x] functionaliteit seconden +/-

-->

<div id="timer" class="valign-wrapper z-depth-5">

<span id="time" class="">60</span>

</div>

<div class="row" id="timerControls">

<div class="col s3">

<button id="timerMin" class="btn-floating btn-large waves-effect waves-light red ">

<i class="material-icons">remove</i>

</button>

</div>

<div class="col s6">

<input type="number" id="inputTijd" placeholder="Seconden" />

</div>

<div class="col s3">

<button id="timerPlus" class="btn-floating btn-large waves-effect green">

<i class="material-icons">add</i>

</button>

</div>

<div class="col s12">

<button id="timerReset" class="waves-effect waves-light btn-large btn-floating">

<i class="material-icons">autorenew</i>

</button>

</div>

</div>

</div>

</div>

<!--*------------------------------------------------------------------*- -->

<!-- Open deur sectie -->

<div class="row spa" id="tabOpenDeur">

<div class="col s12">

<div id="openDeurStart" class="row">

<div class="col s12">

<button id="openDeurEigenVraag" class="btn full-width red-background yellow-color">

Eigen vraag

</button>

</div>

<div class="col s12">

<button id="openDeurWillekeurigeVraag" class="btn full-width red-background yellow-color">

Willekeurige vraag

</button>

</div>

</div>

<div id="openDeurInput">

<label for="openDeurVraag">Geef uw vraag op:</label>

<input id="openDeurVraag" type="text" placeholder="Vraag" />

<table>

<tr>

<td>

<input type="text" placeholder="Antwoord 1" class="inputOD" />

</td>

</tr>

<tr>

<td>

<input type="text" placeholder="Antwoord 2" class="inputOD" />

</td>

</tr>

<tr>

<td>

<input type="text" placeholder="Antwoord 3" class="inputOD" />

</td>

</tr>

<tr>

<td>

<input type="text" placeholder="Antwoord 4" class="inputOD" />

</td>

</tr>

</table>

<button id="openDeurVorige" class="btn left navButton red-background yellow-color">

<i class="material-icons">arrow\_back</i>

</button>

<button id="openDeurVolgende" class="btn right navButton red-background yellow-color">

Start

</button>

</div>

<div id="openDeurToon">

<span id="openDeurOutput">

</span>

<div class="fixed-action-btn vertical click-to-toggle active">

<a class="btn-floating btn-large yellow-background red-color">

<i class="material-icons">menu</i>

</a>

<ul>

<li>

<button id="openDeurToonAntwoorden" class="btn-floating btn-large red-background yellow-color">

<i class="material-icons">remove\_red\_eye</i>

</button>

</li>

<li>

<button id="openDeurReset" class="btn-floating btn-large red-background yellow-color">

<i class="material-icons">autorenew</i>

</button>

</li>

</ul>

</div>

</div>

<!-- TODO:

[x] functionaliteit vraag + antwoorden invoeren

[x] (later) mogelijkheid random vraag uit database

[x] vraag tonen -> mogelijkheid afvinken

-->

</div>

</div>

<!--*------------------------------------------------------------------*- -->

<!-- Puzzel sectie -->

<div class="row spa" id="tabPuzzel">

<div class="col s12">

<!-- TODO:

[x] functionaliteit kernwoord + sleutelwoorden invoeren

[x] (later) mogelijkheid random uit database

[x] sleutelwoorden tonen -> kernwoordfvinken

[x] puzzel tonen

[x] knop om blurred antwoorden te verbergen/tonen

-->

<div id="puzzelStart" class="row">

<div class="col s12">

<button id="puzzelEigenVraag" class="btn full-width red-background yellow-color">

Eigen puzzel

</button>

</div>

<div class="col s12">

<button id="puzzelWillekeurigeVraag" class="btn full-width red-background yellow-color">

Willekeurige puzzel

</button>

</div>

</div>

<!-- input -->

<div id="puzzelInput">

<ul class="collapsible" data-collapsible="accordion" style="width: 100%">

<li>

<div class="collapsible-header">

<span id="s1" data-text="Sleutelwoord 1">Sleutelwoord 1</span>

<i class="material-icons">keyboard\_arrow\_right</i>

</div>

<div class="collapsible-body">

<ul class="">

<li class="">

<label for="puzzelSleutelwoord1">Geef het eerste sleutelwoord:</label>

<input id="puzzelSleutelwoord1" type="text" placeholder="Sleutelwoord 1" class="sleutelwoord" data-target="s1" />

</li>

<li class="">

<input type="text" placeholder="Kernwoord" class="answerP" />

</li>

<li class="">

<input type="text" placeholder="Kernwoord" class="answerP" />

</li>

<li class="">

<input type="text" placeholder="Kernwoord" class="answerP" />

</li>

<li class="">

<input type="text" placeholder="Kernwoord" class="answerP" />

</li>

</ul>

</div>

</li>

<li>

<div class="collapsible-header">

<span id="s2" data-text="Sleutelwoord 2">Sleutelwoord 2</span>

<i class="material-icons">keyboard\_arrow\_right</i>

</div>

<div class="collapsible-body">

<ul class="">

<li class="">

<label for="puzzelSleutelwoord2">Geef het tweed sleutelwoord:</label>

<input id="puzzelSleutelwoord2" type="text" placeholder="Sleutelwoord 2" class="sleutelwoord" data-target="s2" />

</li>

<li class="">

<input type="text" placeholder="Kernwoord" class="answerP" />

</li>

<li class="">

<input type="text" placeholder="Kernwoord" class="answerP" />

</li>

<li class="">

<input type="text" placeholder="Kernwoord" class="answerP" />

</li>

<li class="">

<input type="text" placeholder="Kernwoord" class="answerP" />

</li>

</ul>

</div>

</li>

<li>

<div class="collapsible-header">

<span id="s3" data-text="Sleutelwoord 3">Sleutelwoord 3</span>

<i class="material-icons">keyboard\_arrow\_right</i>

</div>

<div class="collapsible-body">

<ul class="">

<li class="">

<label for="puzzelSleutelwoord3">Geef het derde sleutelwoord:</label>

<input id="puzzelSleutelwoord3" type="text" placeholder="Sleutelwoord 3" class="sleutelwoord" data-target="s3" />

</li>

<li class="">

<input type="text" placeholder="Kernwoord" class="answerP" />

</li>

<li class="">

<input type="text" placeholder="Kernwoord" class="answerP" />

</li>

<li class="">

<input type="text" placeholder="Kernwoord" class="answerP" />

</li>

<li class="">

<input type="text" placeholder="Kernwoord" class="answerP" />

</li>

</ul>

</div>

</li>

</ul>

<button id="puzzelVorige" class="btn left navButton red-background yellow-color">

<i class="material-icons">arrow\_back</i>

</button>

<button id="puzzelVolgende" class="btn right navButton red-background yellow-color">

Start

</button>

</div>

<!-- puzzel tonen -->

<div id="puzzelToon">

<table id="puzzel" class="z-depth-5"></table>

<ul id="antwoordenPuzzel"></ul>

<div class="fixed-action-btn vertical click-to-toggle active">

<a class="btn-floating btn-large yellow-background red-color">

<i class="material-icons">menu</i>

</a>

<ul>

<li>

<button id="puzzelToonAntwoorden" class="btn-floating btn-large red-background yellow-color">

<i class="material-icons">remove\_red\_eye</i>

</button>

</li>

<li>

<button id="puzzelReset" class="btn-floating btn-large red-background yellow-color">

<i class="material-icons">autorenew</i>

</button>

</li>

</ul>

</div>

</div>

</div>

</div>

<!--*------------------------------------------------------------------*- -->

<!-- Ingelijst sectie -->

<div class="row spa" id="tabIngelijst">

<div class="col s12">

<!-- TODO:

[x] functionaliteit vraag + antwoorden invoeren

[x] (later) mogelijkheid random vraag uit database

[x] vraag tonen -> mogelijkheid afvinken

-->

<div id="ingelijstStart" class="row">

<div class="col s12">

<button id="ingelijstEigenVraag" class="btn full-width red-background yellow-color">

Eigen vraag

</button>

</div>

<div class="col s12">

<button id="ingelijstWillekeurigeVraag" class="btn full-width red-background yellow-color">

Willekeurige vraag

</button>

</div>

</div>

<div id="ingelijstInput">

<label for="ingelijstVraag">Geef uw vraag op:</label>

<input id="ingelijstVraag" type="text" placeholder="Vraag" />

<table>

<tr>

<td>

<input type="text" placeholder="Antwoord 1" class="inputI" />

</td>

<td>

<input type="text" placeholder="Antwoord 2" class="inputI" />

</td>

</tr>

<tr>

<td>

<input type="text" placeholder="Antwoord 3" class="inputI" />

</td>

<td>

<input type="text" placeholder="Antwoord 4" class="inputI" />

</td>

</tr>

<tr>

<td>

<input type="text" placeholder="Antwoord 5" class="inputI" />

</td>

<td>

<input type="text" placeholder="Antwoord 6" class="inputI" />

</td>

</tr>

<tr>

<td>

<input type="text" placeholder="Antwoord 7" class="inputI" />

</td>

<td>

<input type="text" placeholder="Antwoord 8" class="inputI" />

</td>

</tr>

<tr>

<td>

<input type="text" placeholder="Antwoord 9" class="inputI" />

</td>

<td>

<input type="text" placeholder="Antwoord 10" class="inputI" />

</td>

</tr>

</table>

<button id="ingelijstVorige" class="btn left navButton red-background yellow-color">

<i class="material-icons">arrow\_back</i>

</button>

<button id="ingelijstVolgende" class="btn right navButton red-background yellow-color">

Start

</button>

</div>

<div id="ingelijstToon">

<span id="ingelijstOutput">

</span>

<div class="fixed-action-btn vertical click-to-toggle active">

<a class="btn-floating btn-large yellow-background red-color">

<i class="material-icons">menu</i>

</a>

<ul>

<li>

<button id="ingelijstToonAntwoorden" class="btn-floating btn-large red-background yellow-color">

<i class="material-icons">remove\_red\_eye</i>

</button>

</li>

<li>

<button id="ingelijstReset" class="btn-floating btn-large red-background yellow-color">

<i class="material-icons">autorenew</i>

</button>

</li>

</ul>

</div>

</div>

</div>

</div>

<!--*------------------------------------------------------------------*- -->

<!-- Info sectie -->

<div class="row spa" id="tabInfo">

<div class="col s12">

<!-- Auteur info -->

<h4>Auteur</h4>

<p>Naam: Robbert Van Hove

<br> Telefoon:

<a href="tel:+32499160402">+32 499 16 04 02</a>

<br> E-mail:

<a href="mailto:robbertvanhove1@gmail.com">robbertvanhove1@gmail.com</a>

</p>

<!-- *----------------------------------* -->

<!-- Over de app -->

<h4>Over de app</h4>

<h5>Gebruik</h5>

<p>

Gebruik deze app bij het spelen van het gezelschapspel "De Slimste Mens Ter Wereld".

</p>

<h5>Handleiding</h5>

</div>

</div>

<!--*------------------------------------------------------------------*- -->

<!-- Instellingen sectie -->

<div class="row spa" id="tabInstellingen">

<div class="col s12">

<!-- algemeen *--*-->

<!--Timer *----*-->

<ul class="collapsible" data-collapsible="accordion">

<li>

<div class="collapsible-header">Algemeen

<i class="material-icons">keyboard\_arrow\_right</i>

</div>

<div class="collapsible-body">

<table class="bordered">

<tr>

<td>Speel intro bij opstarten:</td>

<td>

<div class="switch right">

<label>

<input class="setting" data-setting="introSound" type="checkbox">

<span class="lever"></span>

</label>

</div>

</td>

</tr>

</table>

</div>

</li>

<li>

<div class="collapsible-header">Timer

<i class="material-icons">keyboard\_arrow\_right</i>

</div>

<div class="collapsible-body">

<table class="bordered">

<tr>

<td>Trillen als de timer op 0 springt:</td>

<td>

<div class="switch right">

<label>

<input class="setting" data-setting="vibrateOn0" type="checkbox">

<span class="lever"></span>

</label>

</div>

</td>

</tr>

</table>

</div>

</li>

<li>

<div class="collapsible-header">Open Deur

<i class="material-icons">keyboard\_arrow\_right</i>

</div>

<div class="collapsible-body">

<table class="bordered">

<tr>

<td>Stuur mijn ingevulde data door: </td>

<td>

<div class="switch right">

<label>

<input class="setting" data-setting="ODSendData" type="checkbox">

<span class="lever"></span>

</label>

</div>

</td>

</tr>

</table>

</div>

</li>

<li>

<div class="collapsible-header">Puzzel

<i class="material-icons">keyboard\_arrow\_right</i>

</div>

<div class="collapsible-body">

<table class="bordered">

<tr>

<td>Stuur mijn ingevulde data door: </td>

<td>

<div class="switch right">

<label>

<input class="setting" data-setting="PuzzelSendData" type="checkbox">

<span class="lever"></span>

</label>

</div>

</td>

</tr>

</table>

</div>

</li>

<li>

<div class="collapsible-header">Ingelijst

<i class="material-icons">keyboard\_arrow\_right</i>

</div>

<div class="collapsible-body">

<table class="bordered">

<tr>

<td>Stuur mijn ingevulde data door: </td>

<td>

<div class="switch right">

<label>

<input class="setting" data-setting="IngelijstSendData" type="checkbox">

<span class="lever"></span>

</label>

</div>

</td>

</tr>

<tr>

<td>Doe iets: </td>

<td>

<div class="switch right">

<label>

<input class="setting" data-setting="IngelijstDoe" type="checkbox">

<span class="lever"></span>

</label>

</div>

</td>

</tr>

<tr>

<td>Doe nog iets: </td>

<td>

<div class="switch right">

<label>

<input class="setting" data-setting="IngelijstDoeNog" type="checkbox">

<span class="lever"></span>

</label>

</div>

</td>

</tr>

</table>

</div>

</li>

</ul>

</div>

</div>

</div>

<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/materialize/0.100.2/js/materialize.min.js"></script>

<script src="cordova.js"></script>

<script defer src="js/app.js"></script>

<script src="js/Timer.js"></script>

<script src="js/Settings.js"></script>

<script src="js/Sound.js"></script>

<script src="js/Puzzel.js"></script>

<script src="js/Vraag.js"></script>

<script src="js/Vibrator.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/toastr.js/latest/toastr.min.js"></script>

</body>

</html>

# App.js

//vars

var ingelijst = new Vraag("#ingelijstStart", "#ingelijstInput", "#ingelijstToon", "#ingelijstVraag", ".inputI", "#ingelijstOutput", "outputIngelijst");

var opendeur = new Vraag("#openDeurStart", "#openDeurInput", "#openDeurToon", "#openDeurVraag", ".inputOD", "#openDeurOutput", "outputOD");

$(function () {

//general!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

/\*\*

\* Click link in navigation

\*/

$('.side-nav a').click(function () {

$('.spa').hide();

$('#' + $(this).data('show')).show();

$('.button-collapse').sideNav('hide');

$("#titel").html($(this).data("title"));

Timer.stopTimer(); //stopt timer bij het verlaten van een pagina

});

//html settings!!!!!!!!!!!!!!!!!!!!!!!!!!!!

$(".sleutelwoord, answerP").attr("maxlength", "40"); //sets maxlength in puzzel inputs

document.addEventListener("deviceready", onDeviceReady, false); //ondeviceready

$('.button-collapse').sideNav(); //set sidenav

//Timer!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

/\*\*

\* Click on the timer

\*/

$("#timer").click(function () {

Timer.toggle();

Sound.playSound("../assets/start.mp3")

});

/\*\*

\* Click on plus button

\*/

$("#timerPlus").click(function () {

var timeToCalculate = parseInt($("#inputTijd").val());

Timer.add(timeToCalculate);

});

/\*\*

\* Click on minus button

\*/

$("#timerMin").click(function () {

var timeToCalculate = parseInt($("#inputTijd").val());

Timer.remove(timeToCalculate);

});

/\*\*

\* Click on reset button

\*/

$("#timerReset").click(function () {

if (confirm('Bent u zeker?')) {

Timer.reset();

Timer.stopTimer();

}

});

//Open deur!!!!!!!!!!!!!!!!!!!!!!!!!!!!

/\*\*

\* Click on "Eigen vraag"

\*/

$("#openDeurEigenVraag").click(function () {

opendeur.navigate(false, true, false);

});

/\*\*

\* Click on "Willekeurige vraag"

\*/

$("#openDeurWillekeurigeVraag").click(function () {

opendeur.navigate(false, true, false);

opendeur.selectRandomInput("OD");

});

/\*\*

\* Click on "Start" in inputDiv

\*/

$("#openDeurVolgende").click(function () {

if (opendeur.controleInput()) {

opendeur.navigate(false, false, true);

opendeur.start();

} else {

Materialize.toast("Gelieve alles in te vullen!", 4000);

}

});

/\*\*

\* Click on reset/previous

\*/

$(document).on("click", "#openDeurReset, #openDeurVorige", function () {

opendeur.navigate(true, false, false);

opendeur.reset();

});

/\*\*

\* Click on eye-button in output

\*/

$("#openDeurToonAntwoorden").click(function(){

opendeur.showToggle();

});

//Puzzel!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

/\*\*

\* Click on "Eigen puzzel"

\*/

$("#puzzelEigenVraag").click(function () {

$("#puzzelStart").hide();

$("#puzzelInput").show();

});

/\*\*

\* Click on "Willekeurige puzzel"

\*/

$("#puzzelWillekeurigeVraag").click(function () {

Puzzel.selectRandomPuzzel();

$("#puzzelStart").hide();

$("#puzzelInput").show();

});

/\*\*

\* Click on "Start"

\*/

$("#puzzelVolgende").click(function () {

Puzzel.start();

});

/\*\*

\* Click on button next to an answer

\*/

$(document).on("click", "button.antwoordBtn", function () {

Puzzel.checkAnswer($(this));

});

/\*\*

\* Click on "Vorige" or resetbutton

\*/

$(document).on("click", "#puzzelVorige, #puzzelReset", function(){

Puzzel.reset();

});

/\*\*

\* Click on button with eye in output

\*/

$("#puzzelToonAntwoorden").click(function () {

Puzzel.viewAnswers();

});

/\*\*

\* When typing in "sleutelwoord" -> collapsible-header's text changes

\*/

$(document).on("change paste keyup", ".sleutelwoord", function () {

var target = "#" + $(this).data("target");

var inhoud = $(this).val();

if (!inhoud) { //check empty

inhoud = $(this).attr("placeholder");

}

$(target).html(inhoud); //colapse-title = input

});

//Ingelijst!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

/\*\*

\* Click on "Eigen vraag"

\*/

$("#ingelijstEigenVraag").click(function () {

ingelijst.navigate(false, true, false);

});

/\*\*

\* click on Volgende

\* First checks for any emty inputs

\*/

$("#ingelijstVolgende").click(function () {

if (ingelijst.controleInput()) {

ingelijst.navigate(false, false, true);

ingelijst.start();

} else {

Materialize.toast("Gelieve alles in te vullen!", 000);

}

});

/\*\*

\* Click on "Willekeurige vraag"

\*/

$("#ingelijstWillekeurigeVraag").click(function () {

ingelijst.navigate(false, true, false);

ingelijst.selectRandomInput("ingelijst");

});

/\*\*

\*Click on "Vorige" or resetbutton

\*/

$(document).on("click", "#ingelijstReset, #ingelijstVorige", function () {

ingelijst.navigate(true, false, false);

ingelijst.reset();

});

/\*\*

\* Click on button with eye in output

\*/

$("#ingelijstToonAntwoorden").click(function () {

ingelijst.showToggle();

});

//Open Deur + Ingelijst!!!!!!!!!!!!!!!!!!!!

/\*\*

\* Click on button next to an answer

\*/

$(document).on("click", "button.check", function () {

console.log("checked");

$(this).removeClass("red").addClass("green"); //verander achtergrond naar groen

$(this).find("i").html("check"); // verander kruis in checkmark

$(this).siblings().removeClass("blurry").addClass("checked");

});

//settings!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

/\*\*

\* Klik on checkbox from a setting

\* changes the setting in true/false -> localstorage

\*/

$(".setting").click(function () {

var setting = $(this).data("setting");

var checked = false;

console.log(setting);

if ($(this).is(":checked")) {

checked = true;

}

localStorage.setItem(setting, checked);

});

});

function onDeviceReady() {

console.log('Device is ready');

opendeur.init();

Puzzel.init();

ingelijst.init();

Settings.init();

}

# Style.css

/\* General \*/

.btn:active,

.btn:focus {

background-color: rgba(250, 182, 101, 0.5);

}

.red-background {

background-color: #AA1300;

}

.yellow-background {

background-color: rgba(250, 182, 101, 0.5);

}

table {

font-size: 15px;

}

.navButton {

width: 40%

}

.yellow-color {

color: rgb(250, 182, 101);

}

.full-width {

width: 100%;

}

.collection>li.collection-item,

#antwoordenPuzzel>li {

margin-top: 10px;

}

.collection {

border-style: none;

}

.transparentBG {

background-color: rgba(0, 0, 0, 0);

}

.split-list {

column-count: 2;

}

.split-list>li {

font-size: 15px;

}

.collapsible-header.active>i {

visibility: hidden;

}

/\*navigation and pages \*/

.spa {

display: none;

margin-top: 20px;

}

.collapsible-header.active {

background-color: #921405;

color: white;

}

body {

background-color: rgba(250, 250, 250, 0.9);

}

#tabTimer {

display: block;

}

.side-nav {

background-color: #BD2903;

}

.side-nav>li>a {

color: white;

}

/\* Tmer \*/

#timer {

width: 250px;

height: 250px;

background-color: #AA1300;

border-radius: 50%;

margin: 0 auto;

border-color: rgba(230, 19, 0, 0.8);

border-style: solid;

border-width: 5px;

}

#time {

margin: 0 auto;

font-size: 90px;

color: rgba(255, 255, 255, 0.9);

}

#timerControls {

margin-top: 50px;

}

#tabTimer input {

text-align: center;

font-size: 20px;

}

#timerReset {

width: 50%;

background-color: #BD2903;

font-size: 20px;

}

/\* Open Deur, Puzzel, Ingelijst \*/

#openDeurStart,

#puzzelStart,

#ingelijstStart {

margin-top: 50%;

}

#openDeurStart>div,

#puzzelStart>div,

#ingelijstStart>div {

margin-top: 20px;

}

#openDeurStart>div>button,

#puzzelStart>div>button,

#ingelijstStart>div>button {

height: 50px;

font-size: 20px;

}

.blurry {

color: transparent;

text-shadow: 0 0 15px rgba(0, 0, 0, 0.5);

}

/\* Open Deur \*/

.answer-check {

margin-top: 5px;

margin-left: 10px;

font-size: 25px;

}

/\* puzzel \*/

#puzzel {

background-color: #BD2903;

table-layout: fixed;

}

#puzzel td {

padding: 0;

width: 33%;

text-align: center;

color: white;

font-weight: bold;

border-width: 5px;

border-color: #921405;

border-style: solid;

font-size: 14px;

word-wrap: break-word;

height: 50px;

}

#antwoordenPuzzel>li {

font-size: 20px;

}

/\*

# Kleuren:

\* Hoofdkleur (rood) : AA1300

\* Secundaire kleur (rood) : BD2903

\* Geel: rgb(250, 182, 101)

\*/

# Puzzel.js

var Puzzel = function () {

var puzzelstukken = new Array();

var keywords = new Array();

var aantalJuist = 0;

//public

var init = function () {

$("#puzzelInput").hide();

$("#puzzelToon").hide();

$("#puzzelStart").show();

};

var start = function () {

\_getPuzzel();

if (\_validateInput()) {

//puzzel en antwoorden in html steken

\_showPuzzel();

\_showAntwoorden();

// puzzel

$("#puzzelInput").hide();

$("#puzzelToon").show();

} else {

Materialize.toast("Gelieve alles in te vullen!") //melding tonen om alles in te vullen

puzzelstukken = new Array();

keywords = new Array();

}

};

var reset = function () { //reset alle arrays + laat startscherm zien

init(); //startscherm laten zien

//variabelen resetten

puzzelstukken = new Array();

keywords = new Array();

aantalJuist = 0;

//html leegmaken

$("#puzzel").html("");

$("#antwoordenPuzzel").html("");

$(".answerP, .sleutelwoord").val("");

$("#s1, #s2, #s3").each(function () {

$(this).html($(this).data("text"));

});

};

/\*\*

\* @param {} knop

\*/

var checkAnswer = function (knop) {

var href = knop.attr("href");

id = knop.data("id");

var puzzelstuk = puzzelstukken[id];

var kleur;

console.log(puzzelstuk);

//sleutels aanduiden

puzzelstuk.kernwoorden.forEach(function (key) {

$("#puzzel").find('td').each(function () {

var antwoord = $(this).text();

if (antwoord == key) {

kleur = knop.data("color");

$(this).css("color", kleur);

}

});

}, this);

//antwoord tonen

knop.siblings().removeClass("blurry");

knop.siblings().css("color", kleur);

knop.removeClass("red").addClass("green");

knop.find("i").html("check");

aantalJuist++;

};

var viewAnswers = function () {

if ($(".answerPuzzel").hasClass("blurry")) {

$(".answerPuzzel").removeClass("blurry");

} else {

$(".answerPuzzel").addClass("blurry");

}

};

var selectRandomPuzzel = function () {

var counter = 0;

var question;

var answer = [];

var vraagInput = this.inputVraag;

var antwoordInput = this.inputAntwoorden;

function getData() {

return $.ajax({

url: "http://robbertvanhove.ddns.net/dsmtw/randomPuzzel.php",

dataType: 'json',

});

}

getData().done(function (data) {

$(".collapsible > li > div > ul").each(function () {

var oplossing = data[counter].oplossing;

$(this).find("li > input.sleutelwoord").val(oplossing);

$("#s" + (counter + 1)).html(oplossing);

var antwoorden = new Array();

data[counter].antwoorden.forEach(function (i) {

antwoorden.push(i);

}, this);

console.log(antwoorden);

counterAntwoorden = 0;

$(this).find("li > input.answerP").each(function () {

$(this).val(antwoorden[counterAntwoorden].stuk);

counterAntwoorden++;

});

counter++;

});

});

};

//private

var \_getPuzzel = function () { //haalt alle sleutelwoorden + kernwoorden op

$(".collapsible > li > div > ul").each(function () {

var sleutel = $(this).find("li > input.sleutelwoord").val(); //alle sleutelwoorden ophalen

console.log(sleutel);

var kernwoorden = new Array();

$(this).find("li > input.answerP").each(function () { // alle kernwoorden ophalen

var kernwoord = $(this).val();

kernwoorden.push(kernwoord);

keywords.push(kernwoord);

console.log(kernwoord);

});

var puzzelstuk = { //object met sleutelwoord en bijbehorende sleutelwoorden

sleutel: sleutel,

kernwoorden: kernwoorden // array

};

puzzelstukken.push(puzzelstuk); // puzzelstuk toevoegen aan puzzelstukken

});

console.log(puzzelstukken);

console.log(keywords);

};

var \_showPuzzel = function () { //toont de puzzel in de tabel

var html = "<tr>";

var teller = 1;

var keys = \_shuffle(keywords); //kernwoorden randomizen

keys.forEach(function (key) { //tabel opvullen

//kleur selecteren

html += "<td>" + key + "</td>";

if (teller % 3 == 0) { // elke 3 woorden nieuwe rij

html += "</tr><tr>";

}

teller++;

}, this);

html += "</tr>";

$("#puzzel").html(html);

};

var \_showAntwoorden = function () { //toont antwoorden in ul

var teller = 0;

var kleuren = ["#609140", "#7ca1aa", "#fab665"]

puzzelstukken.forEach(function (e) {

$("<li></li>").html("<button data-id='" + teller +

"' class='check btn red antwoordBtn' data-color='"

+ kleuren[teller]

+ "'><i class='material-icons'>clear</i></button> <span class='answerPuzzel blurry'>" +

e.sleutel +

"</span>").appendTo("#antwoordenPuzzel");

teller++;

}, this);

};

var \_shuffle = function (o) { //randomizet array

for (var j, x, i = o.length; i; j = parseInt(Math.random() \* i), x = o[--i], o[i] = o[j], o[j] = x);

return o;

};

var \_validateInput = function () { // valideert alle input fields; geeft false terug als niet alles is ingevuld

var valid = true;

var inputs = new Array();

//controle alles ingevuld

$("#puzzelInput").find("input").each(function () {

console.log($(this).val());

if ($(this).val() == "") {

valid = false;

}

});

//controle dubbele waardes

return valid;

};

return {

init: init,

start: start,

reset: reset,

checkAnswer: checkAnswer,

viewAnswers: viewAnswers,

selectRandomPuzzel: selectRandomPuzzel

}

}();

# Setting.js

var Settings = function () {

/\*\*

\* Gets all checkboxes with the .setting class

\* Makes new setting in localstorage if not exists

\*

\*/

var init = function () {

$(".setting").each(function () {

var setting = $(this).data("setting");

//check exists

if (localStorage.getItem(setting)) {

if (localStorage.getItem(setting) == "true") {

$(this).prop("checked", true);

}

} else {

localStorage.setItem(setting, false); //put setting in localstorage

}

});

};

return{

init: init

};

}();

# Sound.js

var Sound = function () {

var init = function() {

};

var playSound = function (soundfile) {

var sound = new Media(soundfile);

sound.play();

console.log(soundfile + "playing");

};

return {

init:init,

playSound: playSound,

};

}();

# Timer.js

var Timer = function () {

var time;

var started;

var interval;

//public

var init = function () {

time = 60; //default time

started = false;

};

var setTime = function (tijd) {

if ($.isNumeric(tijd)) { //numeric check

time = parseInt(tijd);

\_updateTimer();

}

};

var toggle = function () {

if (!started) {

startTimer();

} else {

stopTimer();

}

};

var add = function (addedTime) {

if ($.isNumeric(addedTime)) { //numeric check

var added = parseInt(addedTime);

console.log(added);

time += added;

\_updateTimer();

}

};

var remove = function (timeToRemove) {

if ($.isNumeric(timeToRemove)) { //numeric check

var added = parseInt(timeToRemove);

console.log(added);

time -= added;

\_updateTimer();

}

};

var stopTimer = function () {

clearInterval(interval); //stops interval

started = false; //remember timer has stopped

$("#timer").css("background-color", "#AA1300"); //make timer red again when stopped

$("#time").css("color", "rgba(255, 255, 255, 0.9)");

};

var startTimer = function () {

started = true; //remember timer has started

time = parseInt($("#time").html());; //get time from span in timer div

console.log("timer gestart");

console.log(time);

$("#timer").css("background-color", "rgba(200, 19, 0, 0.95)"); //make timer green when running

$("#time").css("color", "rgb(250, 182, 101) ")

interval = setInterval(\_countdown, 1000); //run the loop

};

var reset = function(){

setTime(60);

stopTimer();

};

//private

var \_countdown = function () {

console.log("min 1 sec");

console.log(time);

time -= 1; //new time

\_updateTimer();

};

var \_updateTimer = function () {

if (time < 0) {

time = 0;

stopTimer();

if(localStorage.getItem("vibrateOn0") == "true"){

Vibrator.vibrate(1000);

console.log("vibreert nu voor 1s");

}

}

$("#time").html(time.toString()); //show time

};

return {

init: init,

toggle: toggle,

add: add,

setTime: setTime,

remove: remove,

stopTimer: stopTimer,

startTimer: startTimer,

reset: reset

}

}();

# Vibrator.js

var Vibrator = new function() {

var init = function() {

};

var vibrate = function(time) {

navigator.vibrate(time);

};

return {

vibrate: vibrate,

init: init

}

}();

# Vraag.js

/\*\*

\* @param {string} startDiv - ID of the div that is shown at the start

\* @param {string} inputDiv - ID of the div where input is done

\* @param {string} showDiv - ID of the div where the questing is shown

\* @param {string} inputVraag - ID of the input for the question

\* @param {string} inputAntwoorden - Classname of the input for answers

\* @param {string} output - ID of the element where the question and answers are show

\*/

function Vraag(startDiv, inputDiv, showDiv, inputVraag, inputAntwoorden, output, outputKlasse) {

this.startDiv = startDiv;

this.inputDiv = inputDiv;

this.showDiv = showDiv;

this.inputVraag = inputVraag;

this.inputAntwoorden = inputAntwoorden;

this.output = output;

this.outputKlasse = outputKlasse;

this.vraag;

var antwoorden = new Array();

/\*\*

\* Shows the startDiv

\*/

this.init = function () {

this.navigate(true, false, false);

};

/\*\*

\* Decides to show/hide the start-/input-/showDiv based on a boolean

\* @param {boolean} start

\* @param {boolean} input

\* @param {boolean} show

\*/

this.navigate = function (start, input, show) {

if (start) { //hide/show start

$(this.startDiv).show();

} else {

$(this.startDiv).hide();

}

if (input) { //input tonen/verbergen

$(this.inputDiv).show();

} else {

$(this.inputDiv).hide();

}

if (show) { //output tonen/verbergen

$(this.showDiv).show();

} else {

$(this.showDiv).hide();

}

};

/\*\*

\* Gets the value of the input with the "inputVraag" ID

\* Gets the values of the inputs which contain the class "inputAntwoorden"

\* Shows the question and answers in the element with the "output" ID

\*/

this.start = function () {

//get input

this.vraag = $(this.inputVraag).val(); // get question

$(this.inputAntwoorden).each(function () { //get answers

console.log($(this).val());

antwoorden.push($(this).val());

});

console.log(this.vraag);

console.log(antwoorden);

//fill up output

$("<h4></h4>").html(this.vraag).appendTo(this.output);

var lijstAntwoorden = $("<ul></ul>");

if (antwoorden.length > 4) lijstAntwoorden.addClass("split-list");

var outputKlasse = this.outputKlasse;

antwoorden.forEach(function (antwoord) {

var html = "<button class='check btn red'><i class='material-icons'>clear</i></button> <span class='blurry "

+ outputKlasse

+ "'>"

+ antwoord + "</span>";

$("<li class='answer-check'></li>").html(html).appendTo(lijstAntwoorden);

});

lijstAntwoorden.appendTo(this.output);

console.log(this.outputKlasse);

};

/\*\*

\* Resets all necessary variables

\* Resets all inputs

\*/

this.reset = function () {

//reset variables

this.vraag = "";

antwoorden = new Array();

//reset input vals

$(this.output).empty();

$(this.inputVraag).val("");

$(this.inputAntwoorden).val("");

};

/\*\*

\* Checks all inputs whit the "inputVraag" ID or the "inputAntwoorden" class

\* Retuns true if all inputs have a value

\* @return {boolean} filledIn

\*/

this.controleInput = function () {

var filledIn = true;

if ($(this.inputVraag).val() == "") filledIn = false; //checks emptyness of question input

$(inputAntwoorden).each(function () { //checks if all inputAntwoorden are empty

if ($(this).val() == "") filledIn = false;

});

return filledIn;

};

/\*\*

\* Gets a random question and its answers from the database and filles in the inputs automatically

\* @param {string} soort - the kind of question you want to be returned from the database (OD/ingelijst)

\*/

this.selectRandomInput = function (soort) {

var counter = 0;

var question;

var answers = [];

var vraagInput = this.inputVraag;

var antwoordInput = this.inputAntwoorden;

function getData() {

return $.ajax({

url: "http://robbertvanhove.ddns.net/dsmtw/randomVraag.php",

dataType: 'json',

data: {soort: soort}

});

}

getData().done(function (data) {

//vraag opvullen

question = data.vraag;

$(vraagInput).val(question);

//antwoorden opvullen

answers = data.antwoorden;

console.log(answers);

$(antwoordInput).each(function () { //antwoorden ophalen

$(this).val(answers[counter].antwoord);

counter++;

});

});

};

this.showToggle = function(){

var outputKlasse = this.outputKlasse;

$("." + outputKlasse).each(function(){

if($(this).hasClass("blurry")){

$(this).removeClass("blurry");

}else {

if(!$(this).hasClass("checked")){

$(this).addClass("blurry");

}

}

});

};

}