



3-IT

IoT Advanced: Project parkingplace

IoT Advanced ITFactory

**Frederik Crauwels, Brent Druyts, Brandon
Morren, Milan Sterkens**

Academiejaar 2021-2022

Campus Geel, Kleinhoefstraat 4, BE-2440 Geel

Inhoud

1	IOT ADVANCED CHALLENGE	4
1.1	Introductie.....	4
1.1.1	Groepsindeling.....	4
2	PLAN VAN AANPAK.....	5
2.1	Hoe wij dit project gaan aanpakken	5
2.1.1	Introductie	5
2.1.2	Verdeling van taken	5
2.1.3	Projectplanning.....	6
3	DOCUMENTATIE	7
3.1	Verkenningfase	7
3.1.1	Visualisatie in DB ("virtualisatie in website" en DB).	7
3.1.2	Visualisatie op scherm aan de ingang.	10
3.1.3	Ultrasoon sensor met parkeerbarrière.....	10
3.1.4	Nummerplaatherkenning.....	12
3.1.5	Remote webinterface.....	15
3.1.6	Push message uitsturen wanneer 4 parkeerplaatsen vol zijn (LDR + Push).	25
4	ALGEMENE UITWERKING	26
4.1	Groepsbespreking: minimum requirements	26
4.1.1	Youtube link	26
4.1.2	Github link	26
5	EXTRA CODE.....	27
5.1	Remote webinterface	27
5.1.1	Index.php	27
5.1.2	license.php.....	29
5.1.3	Welcome.php	32
5.1.4	main.py	35
5.1.5	TestFile.py	36
6	BIBLIOGRAFIE	37

1 IoT ADVANCED CHALLENGE

1.1 Introductie

Voor het vak "IoT Advanced" krijgen we een interessante challenge voorgeschoteld. Het bedrijf 3-IT heeft een uitdaging voorzien om een oplossing te bekomen om een parking te monitoren, automatiseren en beheren.

Dit allemaal aan de hand van een totaaloplossing die voorziet in volgende aspecten:

- Parkeerbarrière met ultrasoon sensor.
- Analyse van de nummerplaat (nummerplaatherkenning).
- Visualisatie van de data in de database op een display.
- Remote webinterface voorzien.
- Push message sturen als er 4 plaatsen vol zijn.
- Professionele presentatie en demonstratie
 - Voorstellen groepsleden.
 - Plan van aanpak.
 - Aandeel per groepslid en verdelen van taken.
 - Demo en discussie.
- Project bundel (15 pagina's) met youtube link (2 minuten).
- Evaluatie moment (peer evaluation).

Hier hebben we enkele weken de tijd voor. Het project loopt van:

Vrijdag 29 oktober tot vrijdag 26 november (4 weken).

Op vrijdag 3 december wordt iedereen on campus verwacht. Dit wordt het jury moment!

Uploaden van bundel, presentatie en youtube filmpje moet voor woensdag 1 december 23:59.

1.1.1 Groepsindeling

Als groep hebben wij het nummer "8" gekregen. Wij zijn dus groep 8 met de volgende teamindeling:

R-nummer	Voornaam + naam	Klas
R0842921	Brandon Morren	2CCS01
R0831536	Brent Druyts	2CCS02
R0840939	Milan Sterkens	2CCS02
R0802241	Frederik Crauwels	2CCS01

Deze indeling is slechts beperkt tot de eerste bespreking. Naargelang de vorderingen in het project zullen deze overlappen waarbij het project als team aangepakt zal worden.

2.1.3 Projectplanning

In eerste instantie worden de taken verdeeld per groepslid. In dit hoofdstuk gaan we specifiek kijken voor de indeling per datum en de algemene planning opstellen voor het gehele project. Dit houdt in: deadlines, werkuren, algemene planning per dag,...

Datum	To-Do
Vrijdag 29 oktober	<p>Eerste “werkdag”. Opstellen van projectbundel, taakverdeling en verduidelijking individuele opdracht voor vakantie start. Planningen opmaken, deadlines bepalen en groepswerking uitzoeken.</p> <p>Eerste taak: algemene documentatie van elk individu zijn deelopdracht. Verkenningfase waarin elk lid van het team een onderdeel gaat onderzoeken en documenteren.</p> <p>Tweede taak: vakantie nemen... Tussentijdse vakantie moet ook dienen om rust te nemen. Voor de taken uit, maar neem ook tijd voor jezelf.</p>
Vrijdag 12 november	<p>Tussentijdse evaluatie hoe voor iedereen het project verloopt.</p> <p>Twee weken verder na vakantieperiode. Documentatie en verkenningfase worden afgesloten.</p> <p>ToDo: samen projectbundel opstellen en verder uitwerken. Verkenningfase documenteren en starten met overlappende groepswerking.</p> <p>Afspreken on-campus / op locatie P305-(4): 12:30 mag je langskomen of materiaal achterlaten (afgesloten kast).</p>
Vrijdag 19 november	<p>Tussentijdse evaluatie hoe voor iedereen het project verloopt.</p> <p>Finaliseren van werking systemen. Opstellen van basisopstelling en basiswerking van het systeem. De algemene werking van het gehele systeem moet in orde zijn: dit zijn de “basics”.</p> <p>ToDo: kijk na of basisopstelling in orde is.</p> <p>Werk een constructie uit (papier, karton,...) om de basisopstelling te faciliteren / voor te stellen.</p> <p>Extra content aanmaken voor bonuspunten.</p> <p>Afspreken on-campus / op locatie?</p>
Vrijdag 26 november	<p>Tussentijdse evaluatie hoe voor iedereen het project verloopt.</p> <p>Extra content toevoegen die zijn aangemaakt en afronden het project.</p> <p>Ten laatste zondag 28 november: finaliseren projectbundel en afwerken documentatie / powerpoint om woensdag af te geven.</p> <p>Afspreken on-campus / op locatie?</p>
Woensdag 1 december	<p>Afgeven van projectbundel en powerpoint presentatie.</p>
Vrijdag 3 december	<p>Jury waarbij iedereen aanwezig moet zijn. Voorstellen van project en demo voorzien.</p>

3 DOCUMENTATIE

3.1 Verkenningfase

Ieder lid van de groep gaat zijn deelstuk verkennen, documenteren en verder uitwerking in dit hoofdstuk. Dit noemen wij de verkenningfase. De bedoeling is vooral om nog niet in oplossingen te denken maar een brede visie te hebben en verkrijgen van mogelijke uitwerkingen. Dit kunnen wij vervolgens bekijken en bespreken wat voor onze de best mogelijke uitwerking lijkt. Kijk bij "verdeling van taken" om te weten wie waar aan heeft gewerkt.

3.1.1 Visualisatie in DB ("virtualisatie in website" en DB).

Als database hebben we gekozen voor MariaDB. Een makkelijke, eenvoudige database server die goed overweg kan met PHPMyAdmin. Onze database noemt "parkeergarage".

Voor de opstelling hebben we drie tabellen:

- Nrplaat: geeft een overzicht van onze gebruikers. Persoonlijke gegevens, betaalgegevens, aankomst- en vertrektijden,...
- Users: algemene login gegevens. Het wachtwoord is geëncrypteerd waardoor zelfs de admin hier géén toegang tot heeft.
- Parkingspot: het aantal parkeerplaatsen en hun beschikbaarheid.

3.1.1.1 Nrplaat

In deze tabel zien we een hoop persoonsgegevens verschijnen. Deze worden ook pas achteraf gelinkt met het account van de gebruiker. De extra gegevens worden ook pas achteraf voorzien door de gebruiker. In eerste instantie komt hier automatisch: een nummerplaat en aankomsttijd. Elke nummerplaat is een uniek record – indien de nummerplaat opnieuw voorkomt zal enkel de "aankomsttijd" veranderen.

```
CREATE TABLE IF NOT EXISTS `nrplaat` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(50) NULL,
  `voornaam` varchar(50) NULL,
  `achternaam` varchar(50) NULL,
  `straat_huisnummer` varchar(100) NULL,
  `gemeente_postcode` varchar(100) NULL,
  `bank` varchar(50) NULL,
  `nummerplaat` varchar(50) NOT NULL,
  `parkingspot` INT NULL,
  `aankomst` datetime DEFAULT CURRENT_TIMESTAMP,
  `vertrek` datetime NULL,
  `tebetalen` FLOAT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `nummerplaat` (`nummerplaat`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

Showing rows 0 - 1 (2 total, Query took 0.0004 seconds.)

SELECT * FROM `nrplaat`

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

	id	username	voornaam	achternaam	straat_huisnummer	gemeente_postcode	bank	nummerplaat	parkingspot	aankomst	vertrek	tebetalen
<input type="checkbox"/>	1	admin	tester	tester	aafazfaeae	gagagagaega	aegagaghaga	1-ABC-456	2	2021-11-28 22:08:28	2021-11-29 00:00:00	4
<input type="checkbox"/>	9	test	tester	tester	tester	tester	tester	1-ABZ-456	3	2021-11-28 22:40:15	2021-11-29 00:00:00	2

3.1.1.2 Users

Een tabel voorzien voor de webpagina. Hierin staan login gegevens – deze zullen eerst voorzien worden door de remote webinterface. Indien een gebruiker een account aanmaakt zal deze eerst hier verschijnen. Pas achteraf zal deze gelinkt worden aan een nummerplaat en met tabel "nrplaat" via de webinterface voor gebruikers / admin.

```
CREATE TABLE IF NOT EXISTS `users` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(50) NULL UNIQUE,
  `password` varchar(255) NULL,
  `created_at` datetime DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`)
);
```

Showing rows 0 - 1 (2 total, Query took 0.0003 seconds.)

SELECT * FROM `users`

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

	id	username	password	created_at
<input type="checkbox"/>	2	admin	\$2y\$10\$DqXGSR/z2MuxwWD5pASYMu.2aBi46CukPnwHpLCXBW...	2021-11-28 21:52:51
<input type="checkbox"/>	3	test	\$2y\$10\$/Vlbq8zxmzMqMZpJC1pae9K3P0cDvTTPePWNKrNeao...	2021-11-28 22:28:27

3.1.1.3 Parkingspot

Deze tabel zal gebruikt worden om onze parkeerplaatsen te monitoren. Per parkeerplaats heb je één rij waarin er staat of deze beschikbaar is of niet, en sinds wanneer deze niet meer beschikbaar is, of terug wel beschikbaar is. Deze parkeerplaatsen kunnen we al automatisch voorzien in ons script omdat deze vast staan – en dit gaat voor toekomstige uitbreidingen ook steeds op deze manier kunnen zonder manuele input.

```
CREATE TABLE IF NOT EXISTS `parkingspot` (
  `parkingspot` int(11) NOT NULL AUTO_INCREMENT,
  `status` varchar(50) NULL,
  `aankomst` datetime DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`parkingspot`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

INSERT INTO `parkingspot` (`parkingspot`, `status`, `aankomst`)
VALUES
(1,'beschikbaar',CURRENT_TIMESTAMP),
(2,'beschikbaar',CURRENT_TIMESTAMP),
(3,'beschikbaar',CURRENT_TIMESTAMP),
(4,'beschikbaar',CURRENT_TIMESTAMP);
```

✓ Showing rows 0 - 3 (4 total, Query took 0.0003 seconds.)

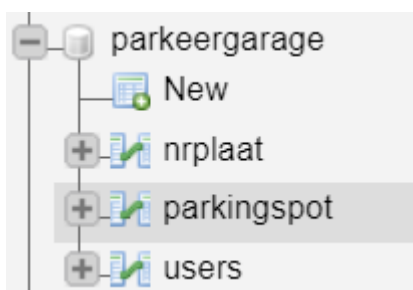
`SELECT * FROM `parkingspot``

☐ Show all | Number of rows: Filter rows:

+ Options

	parkingspot	status	aankomst
<input type="checkbox"/> Edit Copy Delete	1	beschikbaar	2021-11-28 22:08:11
<input type="checkbox"/> Edit Copy Delete	2	beschikbaar	2021-11-28 22:08:28
<input type="checkbox"/> Edit Copy Delete	3	beschikbaar	2021-11-28 22:40:15
<input type="checkbox"/> Edit Copy Delete	4	beschikbaar	2021-11-28 20:40:12

Zo voorzien wij onze gegevens in één database: parkeergarage, met drie tabellen die hierboven vermeld werden. Dankzij PhPMyAdmin hebben we hier steeds een handig overzicht – waardoor ook deze webinterface een onmisbaar deel wordt in dit project.



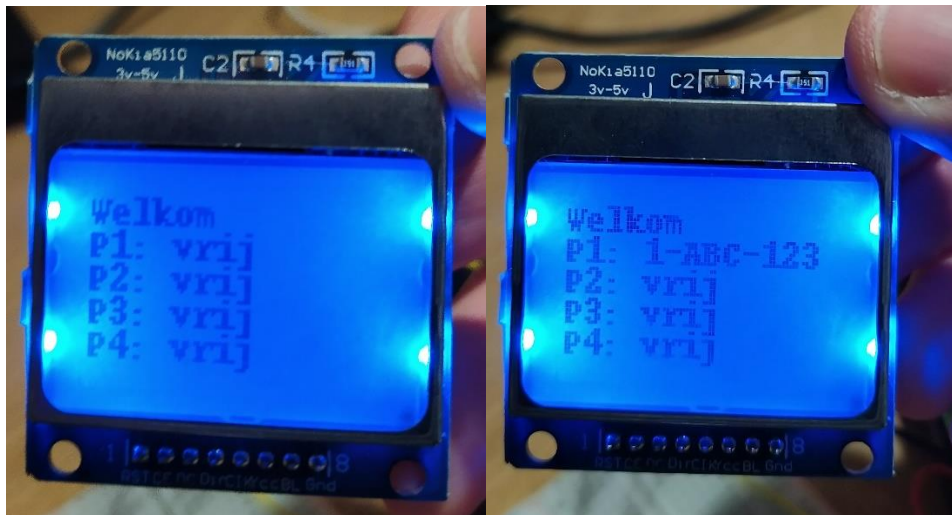
3.1.1.4 TestFile.py

Deze file werd gebruikt in ons project om de data van- en naar de database uit te voeren in een geïsoleerde opstelling. Hierbij werden gegevens van tevoren "voorzien" om te testen of de formules daadwerkelijk toegepast werden. Indien dit werkte – zal dit in onze main.py (basis van ons project) ook werken naargelang de gegevens parameters. Zie het als een soort "unit test" vooraleer deze werd toegepast in de praktijk. Deze code kan je helemaal achteraan bekijken bij de "extra code".

Hier staan vooral de berekeningen voor de prijzen en het versturen / updaten van allerlei gegevens in verschillende tabellen. Denk maar aan nummerplaat, aankomst en vertrek en parkeerplaats beschikbaarheid.

3.1.2 Visualisatie op scherm aan de ingang.

Aan de ingang van de parking hebben we een scherm (lcd) voorzien waarop je kan zien welke plaatsen bezet zijn en welke vrij zijn. Als er een auto binnenrijdt dan wordt er een foto genomen van de nummerplaat en dan wordt de nummerplaat opgeslagen in de database. Wanneer de auto dan op een parkeerplaats gaat staan dan wordt de nummerplaat van de auto gekoppeld aan de parkeerplaats. Vervolgens wordt het scherm en de web interface aangepast.

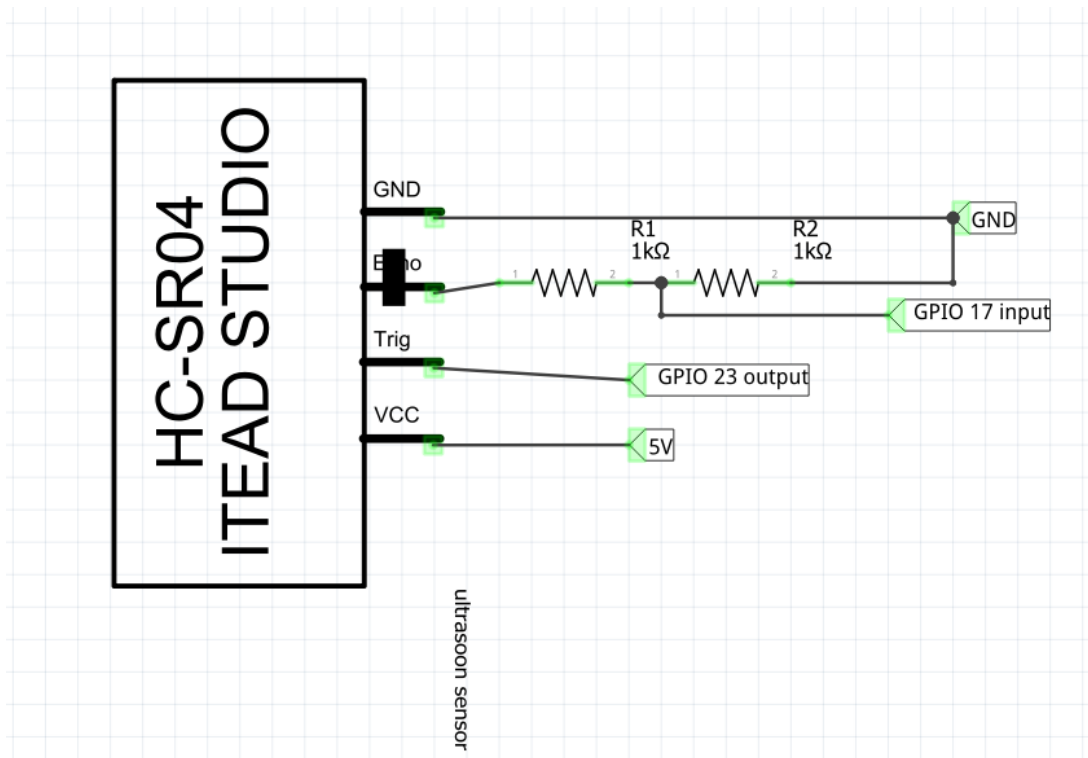
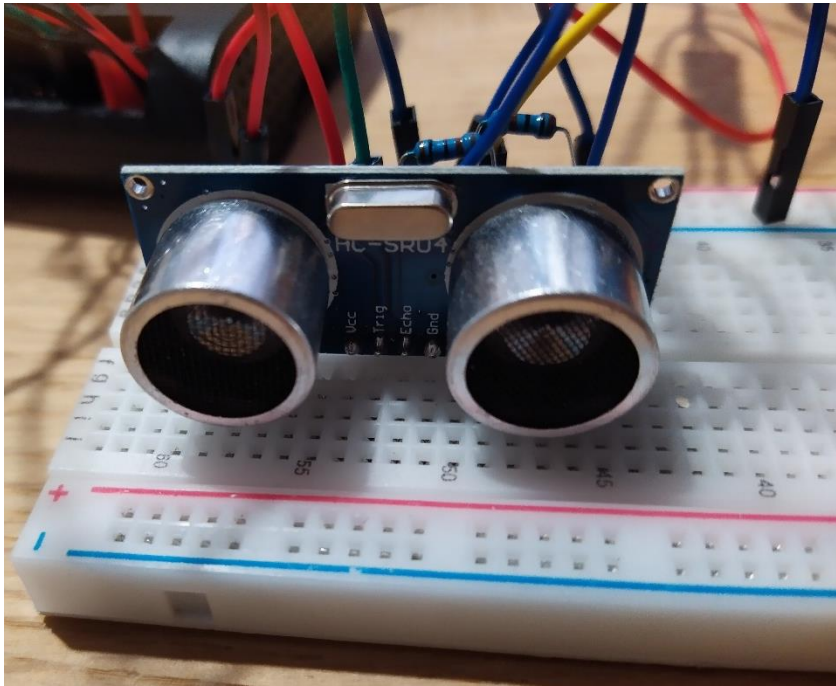


3.1.3 Ultrasoon sensor met parkeerbarrière.

Ik heb als deelopdracht gekozen om te werken met de ultrasoon sensor en de slagboom. Ik heb voor de ultrasoon sensor gekozen omdat we hier vorig schooljaar al mee gewerkt hebben en ik dit een zeer interessant onderwerp vind. Ik heb ook voor de slagboom gekozen omdat ik in het middelbaar al een beetje ervaring heb opgedaan met het programmeren van stappen, DC en servomotoren.

Ultrasoon sensor:

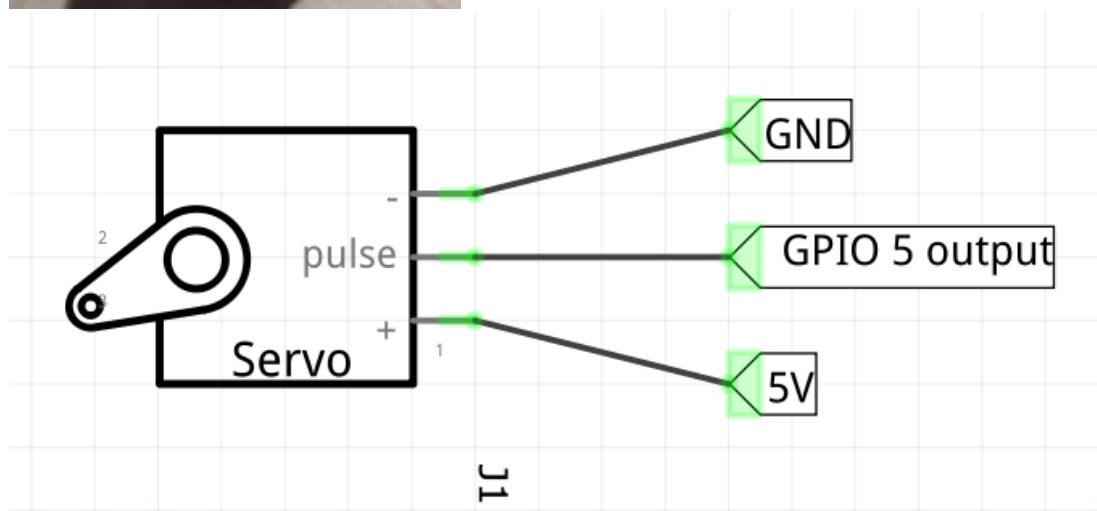
Om te weten of er een auto bij de ingang of uitgang staat heb ik de ultrasoon sensor van vorig jaar gebruikt.



Voor het programma voor de ultrasoon sensor heb ik het basisprogramma van IoT essentials genomen en dit aangepast.

Motor voor slagboom:

Voor de motor voor de slagboom had ik eerst een stappenmotor genomen. Maar deze motor kan alleen maar naar links of naar rechts draaien, dit is dus niet handig voor een slagboom. Deze motor kan gebruikt worden voor een slagboom in combinatie met een tandwielenkast, deze is er niet standaard bij. Bij een servomotor kan je de precieze stand van de motor achterhalen omdat hier een tandwielkast bij zit. Daarom heb ik voor dit project gekozen om te werken met een servomotor.



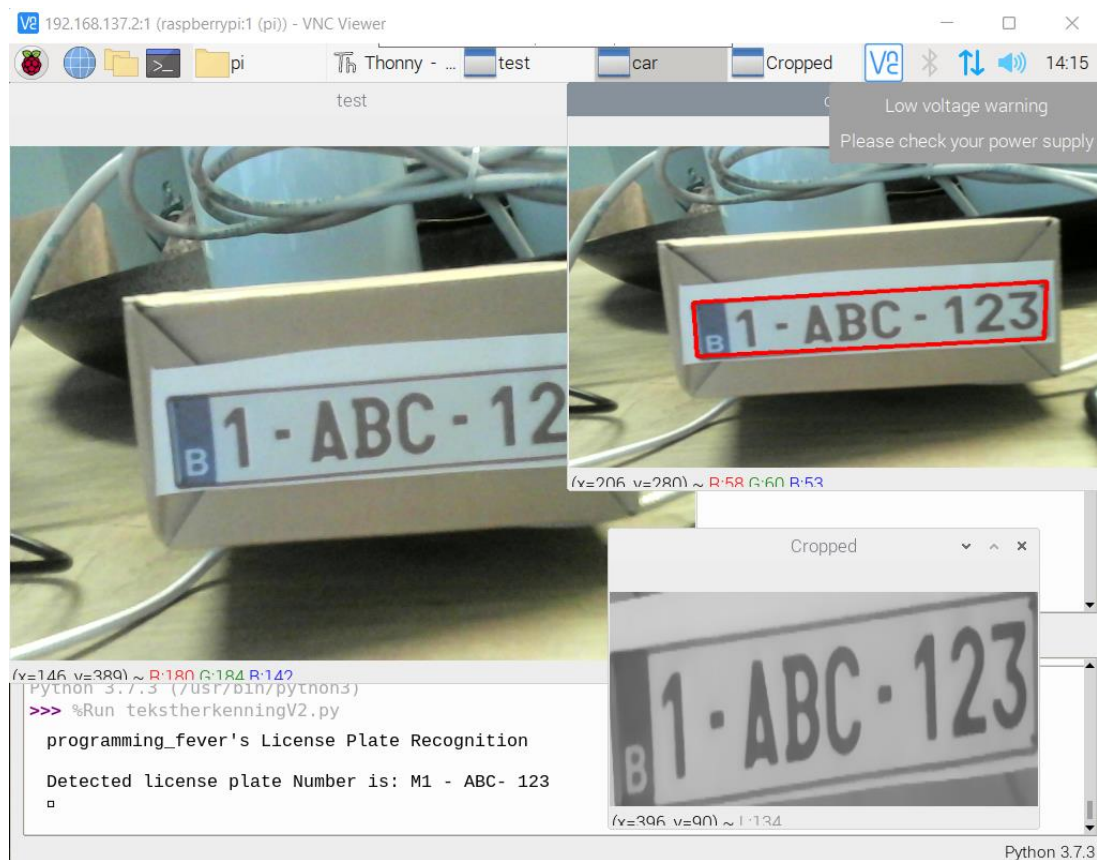
3.1.4 Nummerplaatherkenning.

De nummerplaatherkenning zal dienen voor het optisch encoderen van de nummerplaatgegevens naar karakters die in de database gezet zullen worden. Zo kunnen we later met die gegevens de wagen factureren voor de tijdspanne dat hij geparkeerd stond.

3.1.4.1 Werking

We gebruiken hiervoor natuurlijk als hoofdbestandsdeel de raspberry pi. De beelden worden gemaakt met een HD-usb camera. Met de software OpenCV en Tesseract sturen we een algoritme op het beeldmateriaal af die teksten kan aflezen en deze terugstuurt als een string. Deze string gaan we achteraf in onze database zetten.

Op de volgende foto van onze testopstelling is het resultaat duidelijk te zien:



3.1.4.2 De code

De herkenningsoftware met bijhorende code was op het internet slechts te vinden voor de dedicated raspberry pi camera. Daarom moest ik een groot deel van de code aanpassen zodat deze zou werken bij een usb-camera.

```
def scanNummerplaat():
    #while loop om scannen evt te herhalen
    i = 0
    while i < 1:
        #camera inlezen
        ret,frame=cam.read()
        time.sleep(0.5)
        img=frame

        #foto nummerplaat vorm herkennen
        img = cv2.resize(img, (600,400) )

        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        gray = cv2.bilateralFilter(gray, 13, 15, 15)
```

```

    edged = cv2.Canny(gray, 30, 200)
    contours = cv2.findContours(edged.copy(), cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
    contours = imutils.grab_contours(contours)
    contours = sorted(contours, key = cv2.contourArea, reverse =
True)[:10]
    screenCnt = None

    for c in contours:

        peri = cv2.arcLength(c, True)
        approx = cv2.approxPolyDP(c, 0.018 * peri, True)

        if len(approx) == 4:
            screenCnt = approx
            break

#als er geen nummerplaat is herkent
if screenCnt is None:
    detected = 0
    print ('No contour detected')
else:
    detected = 1

#nummerplaatcontouren tekenen
if detected == 1:
    cv2.drawContours(img, [screenCnt], -1, (0, 0, 255), 3)

mask = np.zeros(gray.shape,np.uint8)
new_image = cv2.drawContours(mask,[screenCnt],0,255,-1,)
new_image = cv2.bitwise_and(img,img,mask=mask)

(x, y) = np.where(mask == 255)
(topx, topy) = (np.min(x), np.min(y))
(bottomx, bottomy) = (np.max(x), np.max(y))
Cropped = gray[topx:bottomx+1, topy:bottomy+1]

#tekst van nummerplaat herkennen
text = pytesseract.image_to_string(Cropped, config='--psm 11')
print ('License Plate Recognition\n')
print('Detected license plate Number is:',text,'\n')
img = cv2.resize(img,(500,300))
Cropped = cv2.resize(Cropped,(400,200))
#foto's tonen om te debuggen:
#cv2.imshow('car',img)
#cv2.imshow('Cropped',Cropped)

#foto schrijven naar opslag met tijdstip in de naam

```

```

img_name = 'numberplate_image_{}.png'.format(tijdstip)
cv2.imwrite(img_name, frame)

time.sleep(2)
i=i+1
return text

```

3.1.5 Remote webinterface.

Voor deze opdracht moeten we een “remote webinterface” voorzien voor de admin (owner) van de parkeergarage. De admin kan hierop de status bekijken van bezoekers van de parkeergarage, als ook extra informatie.

Wij hebben gekozen voor een algemene **apache2** webinterface, met een **php** uitvoering om actief data op te halen uit een **mariaDB database**.

Buiten het feit dat we de website lokaal hosten op de raspberry pi, kan deze in allerlei vormen online worden voorzien via webhosting. Voor deze opdracht laten we de website lokaal, toegankelijk via apache2. Zo kan je steeds via het **ip adres van de raspberry pi** terecht komen op onze aanmeld pagina.

Een **extra toepassing of toegankelijkheid** kan gebeuren op verschillende manieren. **Toegang tot Raspberry Pi:**

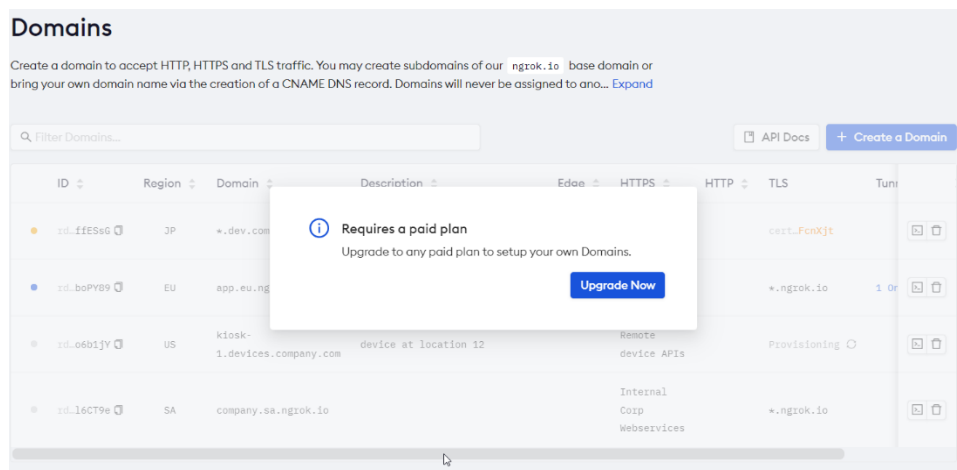
- Via **VNC Viewer:** met deze tool kan je steeds connectie maken naar je raspberry pi om het een en ander na te kijken.
- Via **NGROK:** deze tool gaan we verder gebruiken in dit hoofdstuk – al is het in dit project eerder primitief – het zal de webpagina “remote” toegankelijk maken (apache2), als ook de raspberry pi. Win=win, je hebt toegang tot een webinterface als ook rechtstreeks tot de Pi.

3.1.5.1 NGROK

NGROK is een gratis tool (of betalend) om toegang te krijgen tot jouw toestellen – zij het via port 80 (HTTP), 443 (HTTPS) (webinterface) of ook port 22 (SSH). Zo kan je zowel rechtstreeks als onrechtstreeks aan de informatie voor onze parkeergarage, als ook aan de raspberry pi in noodgevallen.

Alhoewel de apache2 client voldoende is, krijg je hier nog een extra stap om een beveiligde website te testen (HTTPS).

Het **nadeel van NGROK**: de gratis versie geeft elke keer opnieuw een random URL. Deze is ook heel “vaag” waardoor deze er niet netjes uit ziet. Alsnog biedt NGROK deze mogelijkheid aan om een vaste, statische URL aan te bieden mits betaling:



Voor deze opdracht is dit eerder irrelevant omdat het niets verandert aan onze uitwerking, buiten een “propere” link.

Als we NGROK dan opstarten met de nodige instellingen krijgen we dit te zien:

```

pi@raspberrypi: ~/Downloads
File Edit Tabs Help
ngrok by @inconshreveable (Ctrl+C to quit)

Session Status      online
Account             Frederik Crauwels (Plan: Free)
Version             2.3.40
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding            http://89a2-84-197-9-22.ngrok.io -> http://localhost:80
                    https://89a2-84-197-9-22.ngrok.io -> http://localhost:80

Connections          ttl    opn    rt1    rt5    p50    p99
                    3      0      0.00   0.00   5.77   9.95

HTTP Requests
-----
GET /favicon.ico     404 Not Found
GET /index.php       200 OK
GET /logout.php      302 Found
GET /favicon.ico     404 Not Found
GET /welcome.php     200 OK
POST /index.php      302 Found
GET /                200 OK
GET /favicon.ico     404 Not Found
GET /                200 OK

```

De locatie kan je ook kiezen – in mijn geval koos NGROK steeds voor een “US” link. Als we dan verder kijken op de website van NGROK krijg je dezelfde details:

Region	URL	Client IP	Established
US	http://89a2-84-197-9-22.ngrok.io	84.197.9.22	11m ago
<div> <div>ID</div> <div>tn_2127dVo8izZh1RcvGnTQpcusZ7N</div> </div> <div> <div>Protocol</div> <div>http</div> </div> <div> <div>URL</div> <div>http://89a2-84-197-9-22.ngrok.io</div> </div> <div> <div>Client IP</div> <div>84.197.9.22</div> </div> <div> <div>Region</div> <div>us</div> </div> <div> <div>Established</div> <div>Nov 28th, 2021 22:00:43 UTC</div> </div> <div> <div>Metadata</div> <div>0 bytes</div> </div>			
US	https://89a2-84-197-9-22.ngrok.io	84.197.9.22	11m ago
<div> <div>ID</div> <div>tn_2127dUIm1vqeEFQeJ3V2MgVdzn7w</div> </div> <div> <div>Protocol</div> <div>https</div> </div> <div> <div>URL</div> <div>https://89a2-84-197-9-22.ngrok.io</div> </div> <div> <div>Client IP</div> <div>84.197.9.22</div> </div> <div> <div>Region</div> <div>us</div> </div> <div> <div>Established</div> <div>Nov 28th, 2021 22:00:42 UTC</div> </div> <div> <div>Metadata</div> <div>0 bytes</div> </div>			

Zo creëren we een directe tunnel naar onze raspberry pi om allerlei toepassingen te testen en uit te voeren. In ons geval wordt dit een remote webinterface die je nu overal kan bereiken:

← → ↻ <https://89a2-84-197-9-22.ngrok.io/index.php>

Login

Please fill in your credentials to login.

Username

Password

Login

Don't have an account? [Sign up now.](#)

Uiteraard om het hoofdstuk “NGROK” verder af te sluiten, krijg je exact dezelfde pagina zoals die via localhost. Hier hebben we wel géén keuze uit HTTPS als extra beveiliging / test:

← → ↻ ⚠ Not secure | 192.168.60.254/index.php

Login

Please fill in your credentials to login.

Username

Password

Login

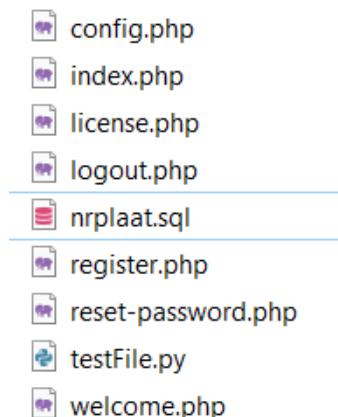
Don't have an account? [Sign up now.](#)

Hiermee willen we graag aantonen dat we ook verder hebben gekeken dan enkel apache2. De optie via NGROK geeft als extra dat je deze ook al kan hosten online en toegankelijk maken voor iedereen. Voor onze opdracht blijft het relatief hetzelfde; via NGROK of via apache2 lokaal. Het extra voordeel blijft vooral dat je als owner ook extra toegang hebt met SSH naar je raspberry pi; voor testing en toegang "indien er iets fout loopt". NGROK biedt ook nog een tal van extra voordelen aan die in de toekomst veel opties bieden:

- IP restrictions & password protection: je kan de webpagina extra beveiligen met een extra aanmeld pagina – specifiek om aan de webpagina te geraken (dus niet in-browser). Daarnaast kan je ook IP-restricties toepassen – bv om je eigen parkeergarage WiFi enkel te laten toestaan om de webpagina te betreden.
- SSH, TLS certificaten, APIs, "Events",...: extra beveiliging en zelfs API toegang voor bepaalde doeleinden zijn mogelijk in de toekomst. Events kunnen ook "in de cloud" aangestuurd worden om zo bv. Login log informatie bij te houden (wanneer meld iemand zich aan, met welk IP,...). Veel opties dus, waar wij nu jammer genoeg geen toegang tot hebben (beta of betalend).

3.1.5.2 Remote webinterface

Voor onze remote webinterface hebben we gekozen voor een **PHP website / uitvoering**. PHP biedt de mogelijkheid om consistent nieuwe data aan te leveren terwijl die wordt verstuurd via een python programma. Vandaar ziet onze webinterface er zo uit:



- **Index.php:** dit is de "startpagina" zoals iedereen deze wel kent. De "home" pagina waarop je als eerst komt. Hier zie je een "login" pagina staan, en indien je nog geen login hebt kan je deze aanmaken met de link onderaan.
- **Config.php:** deze file dient voor de basisgegevens aan te leveren. Dit gaat voornamelijk over de instellingen voor de database, bijvoorbeeld.
- **Nrplaat.sql:** deze SQL file staat ten dienste van onze database (zie vorige hoofdstuk). Met deze file kunnen we automatisch onze gehele setup opzetten in PHPmyadmin.
- **Welcome.php:** Je zal na het aanmelden op deze pagina verschijnen. Hier staan een tal van informatie voor de gebruiker. Afhankelijk van welke login (admin of "user") zal je een ander layout en gegevens te zien krijgen.

- **Register.php:** deze pagina dient om je te registreren indien je nog geen account hebt. Username + password is voldoende! Er zijn validatieregels voorzien die de lengte van een wachtwoord bepalen.
- **License.php:** op deze pagina ga je jouw persoonlijke account linken met een nummerplaat die geregistreerd is via onze nummerplaatherkenning. Als jouw nummerplaat in onze DB voorkomt zullen de gegevens ingevuld worden en krijg je deze ook te zien op welcome.php.
- **Reset-password.php:** wachtwoord vergeten? Geen probleem! Je kan deze makkelijk en snel zelf wijzigen – zonder tussenkomst van admin.
- **Logout.php:** een standaard doorverwijspagina die, indien je op welcome.php (of andere pagina) wilt afmelden – je automatisch doorstuurt naar index.php. Ook indien een gebruiker een andere pagina wil betreden zal dit script in werking treden om de gebruikers terug naar index.php te verwijzen.
- **PHPmyadmin** (niet zichtbaar): een algemene admin pagina voor onze database. Je kan de database op twee manieren bekijken: de gegevens staan (voor admin) op de welcome.php als ook in PHPmyadmin. Voor specifieke uitwerkingen (resetten van ww, veranderen van records) zal je steeds hier terecht moeten. Om alle gegevens te bekijken van gebruikers kan je de welcome.php pagina gebruiken.

De code voor elke pagina apart zal te zien zijn helemaal achteraan. De volgende hoofdstukken focussen op wat je te zien zal krijgen op elke pagina.

3.1.5.3 Index.php

De standaard login pagina, zoals je deze al gezien hebt voordien.

← → ↻ ⚠ Not secure | 192.168.60.254/index.php

Login

Please fill in your credentials to login.

Username

Password

Login

Don't have an account? [Sign up now.](#)

Aanmelden met username en password. Heb je deze niet? Klik op “Sign up now” om te registreren – foute gegevens? Dan laat de browser je niet door!

Invalid username or password.

Username

3.1.5.4 Config.php

Basis configuratie voor onze database.

```
parkingiot.be > [ ] config.php
1  <?php
2  /* Database credentials. Assuming you are running MySQL
3  server with default setting (user 'root' with no password) */
4  define('DB_SERVER', 'localhost');
5  define('DB_USERNAME', 'pi');
6  define('DB_PASSWORD', 'raspberry');
7  define('DB_NAME', 'parkeergarage');
8
9  /* Attempt to connect to MySQL database */
10 $link = mysqli_connect(DB_SERVER, DB_USERNAME, DB_PASSWORD, DB_NAME);
11
12 // Check connection
13 if($link === false){
14     die("ERROR: Could not connect. " . mysqli_connect_error());
15 }
16 ?>
```

3.1.5.5 Nrplaat.sql

Basis configuratie voor onze database (MariaDB op raspberry pi lokaal).

```
parkingiot.be > [ ] nrplaat.sql
1  SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
2  SET time_zone = "+00:00";
3
4  --
5  -- Table structure for table `users`
6  --
7
8  CREATE TABLE IF NOT EXISTS `nrplaat` (
9      `id` int(11) NOT NULL AUTO_INCREMENT,
10     `username` varchar(50) NULL,
11     `voornaam` varchar(50) NULL,
12     `achternaam` varchar(50) NULL,
13     `straat_huisnummer` varchar(100) NULL,
14     `gemeente_postcode` varchar(100) NULL,
15     `bank` varchar(50) NULL,
16     `nummerplaat` varchar(50) NOT NULL,
17     `parkingspot` int NULL,
18     `aankomst` datetime DEFAULT CURRENT_TIMESTAMP,
19     `vertrek` datetime NULL,
20     `tebetalen` float NULL,
21     PRIMARY KEY (`id`),
22     UNIQUE KEY `nummerplaat` (`nummerplaat`)
23 ) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
24
25
26 CREATE TABLE IF NOT EXISTS `users` (
27     `id` int(11) NOT NULL AUTO_INCREMENT,
28     `username` varchar(50) NULL UNIQUE,
29     `password` varchar(255) NULL,
30     `created_at` datetime DEFAULT CURRENT_TIMESTAMP,
31     PRIMARY KEY (`id`)
32 );
33
34
35 CREATE TABLE IF NOT EXISTS `parkingspot` (
36     `parkingspot` int(11) NOT NULL AUTO_INCREMENT,
37     `status` varchar(50) NULL,
38     `aankomst` datetime DEFAULT CURRENT_TIMESTAMP,
39     PRIMARY KEY (`parkingspot`)
40 ) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
41
42
43 INSERT INTO `parkingspot`(`parkingspot`, `status`, `aankomst`)
44 VALUES
45 (1,'beschikbaar',CURRENT_TIMESTAMP),
46 (2,'beschikbaar',CURRENT_TIMESTAMP),
47 (3,'beschikbaar',CURRENT_TIMESTAMP),
48 (4,'beschikbaar',CURRENT_TIMESTAMP);
49
50
```

In totaal telt onze database "parkeergarage" 3 tabellen: users om geregistreerde gebruikers te bekijken, nrplaat voor algemene gegevens per gebruiker en parkingspot om de beschikbaarheid van parkeerplaatsen weer te geven.

3.1.5.6 Welcome.php

Dit is de pagina waar je op terecht komt indien je succesvol bent aangemeld. Je krijgt hier **twee soorten pagina's te zien**: een ADMIN pagina, of een gebruiker pagina.

ADMIN: krijg je alleen te zien indien je als gebruikersnaam "admin" aanmeldt.

Hi, **admin**. Welcome to our parking garage.

[Reset Your Password](#)
[Sign Out of Your Account](#)
[Register your license plate](#)

Available parking spots:

Parkingspot	Status	Bezet / beschikbaar sinds
1	beschikbaar	2021-11-28 22:08:11
2	beschikbaar	2021-11-28 22:08:28
3	beschikbaar	2021-11-28 22:40:15
4	beschikbaar	2021-11-28 20:40:12

Your personal information:

voornaam	achternaam	straat_huisnummer	gemeente_postcode	bank	nummerplaat	parkingspot	aankomst	vertrek	tebetalen
tester	tester	aafazfaaeafae	gagagagaega	aegagaghaga	1-ABC-456	2	2021-11-28 22:08:28	2021-11-29 00:00:00	4
tester	tester	tester	tester	tester	1-ABZ-456	3	2021-11-28 22:40:15	2021-11-29 00:00:00	2

User information

id	username	password	created at
2	admin	\$2y\$10\$DqXGSR/z2MuxwWD5pASYMu.2aB46CukPnwHpLCXBW.iwmJNdo3u	2021-11-28 21:52:51
3	test	\$2y\$10\$/Vbq8zmxzMQmZpJC1pae9K3P0cDvTTPePWNKrNeaoZwJFDdVHF2	2021-11-28 22:28:27

Alleen de admin kan alle gegevens zien van gebruikers. Zowel persoonlijke als aanmeldgegevens. Wachtwoorden zijn nooit zichtbaar – deze worden tijdens het aanmelden geëncrypteerd zodat ook de admin niet zomaar alles kan zien. Je ziet hier duidelijk alle gegevens die je moet weten van de parking, als ook van de gebruikers.

USER: users kunnen enkel hun persoonlijke gegevens bekijken. De gegevens voor andere gebruikers, als ook de "user information" tabel zijn niet zichtbaar.

Hi, **test**. Welcome to our parking garage.

[Reset Your Password](#)
[Sign Out of Your Account](#)
[Register your license plate](#)

Available parking spots:

Parkingspot	Status	Bezet / beschikbaar sinds
1	beschikbaar	2021-11-28 22:08:11
2	beschikbaar	2021-11-28 22:08:28
3	beschikbaar	2021-11-28 22:40:15
4	beschikbaar	2021-11-28 20:40:12

Your personal information:

voornaam	achternaam	straat_huisnummer	gemeente_postcode	bank	nummerplaat	parkingspot	aankomst	vertrek	tebetalen
tester	tester	tester	tester	tester	1-ABZ-456	3	2021-11-28 22:40:15	2021-11-29 00:00:00	2

User information

id	username	password	created at
2	admin	\$2y\$10\$DqXGSR/z2MuxwWD5pASYMu.2aB46CukPnwHpLCXBW.iwmJNdo3u	2021-11-28 21:52:51
3	test	\$2y\$10\$/Vbq8zmxzMQmZpJC1pae9K3P0cDvTTPePWNKrNeaoZwJFDdVHF2	2021-11-28 22:28:27

Enkel jouw eigen gegevens (indien gevalideerd via "register your license plate" kan de gebruiker zien.

Reset your password is om je wachtwoord te veranderen. Sign out om af te melden en register your license plate om jouw aanmeldgegevens

(gebruikersnaam) te linken aan jouw nummerplaat gegevens en extra persoonlijke informatie.

3.1.5.7 Register.php

Op deze pagina kan een nieuwe gebruiker zich aanmelden. Er wordt telkens gevalideerd of er al een soortgelijke gebruikersnaam in onze database zit.

← → ↻

⚠ Not secure | 192.168.60.254/register.php

Sign Up

Please fill this form to create an account.

Username

Password

Confirm Password

Submit

Reset

Already have an account? [Login here.](#)

Username

This username is already taken.

3.1.5.8 License.php

Op deze pagina moet de gebruiker extra informatie toevoegen. Allereerst om de nummerplaat te linken met dit account – als ook extra informatie voor de toekomst. Voor betalingen, persoonlijk adres... Zo kan je steeds een gebruiker in de "user" tabel linken met een persoon in de "nrplaat" tabel. Ook hier wordt aan gegevensvalidatie gedaan zodat gegevens zeker lang genoeg zijn om in de tabel geplaatst te worden. De nummerplaat wordt niet nagekeken – dit zou een extra stap kunnen zijn om te zien of deze herkend wordt (zoals hierboven met username). Indien foutief, krijg je gewoon géén gegevens te zien bij welcome.php.

← → ↻ ⚠ Not secure | 192.168.60.254/license.php

Create your license plate

Please fill out this form to insert your license plate into our database, with additional information.

License plate

Achternaam

Voornaam

Straat + huisnummer

Gemeente + postcode

Bankrekeningnummer

[Submit](#) [Cancel](#)

3.1.5.9 Reset-password.php

Deze pagina staat er om je wachtwoord te veranderen indien je het bent vergeten. Zoals eerder aangehaald zijn wachtwoorden geëncrypteerd. Een admin kan dus niet zomaar jouw wachtwoord vertellen. Alleen jij weet het – en anders pas je het hier makkelijk zelf aan.

← → ↻ ⚠ Not secure | 192.168.60.254/reset-password.php

Reset Password

Please fill out this form to reset your password.

New Password

Confirm Password

[Submit](#) [Cancel](#)

3.1.5.10 Logout.php

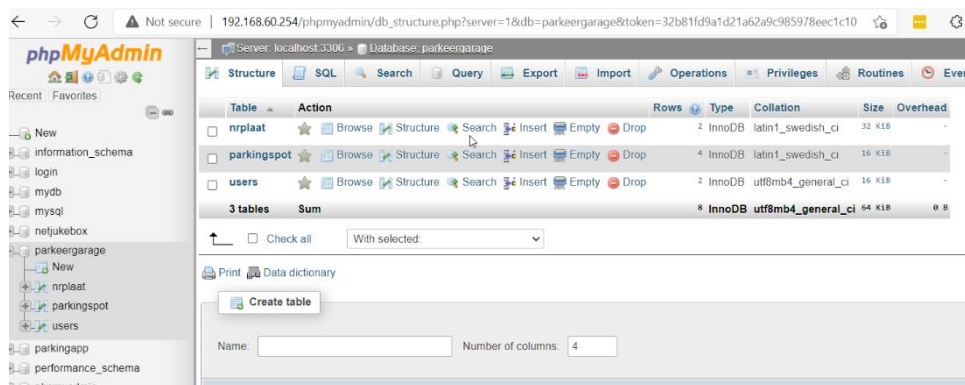
Logout.php dient enkel om door te linken naar index.php (login pagina). Telkens je “sign out” of niet juist bent aangemeld zal je door dit PHP script worden teruggestuurd naar index.php.

```
parkingiot.be > [?] logout.php
1  [?]php
2  // Initialize the session
3  session_start();
4
5  // Unset all of the session variables
6  $_SESSION = array();
7
8  // Destroy the session.
9  session_destroy();
10
11 // Redirect to login page
12 header("location: index.php");
13 exit;
14 [?]
```

3.1.5.11 PHPMyadmin (DB)

De standaard phpmyadmin is ook beschikbaar en helemaal onmisbaar voor dit project. Wil je makkelijk de gegevens van de database nakijken? Dan gaat dit eenvoudig via deze pagina. Hier voeren we ook de SQL query door die voordien ter sprake kwam om alle gegevens automatisch te voorzien zonder al te veel manuele input. Dit is ook handig voor testing purposes of als admin om iets aan te passen – een foutieve nummerplaat, verkeerde gegevens ingevoerd / doorgestuurd / ...

Zelfs met een remote webinterface kom je eigenlijk niet toe als admin. De PHPMyAdmin is hier onmisbaar en moet als tweede remote webinterface behandeld worden zodat de owner op alle vlakken in controle is.



Extra code die nog niet te zien was in bovenstaande screenshots vind je achteraan terug voor de bronvermeldingen.

In de bronnen vind je ook het grootste deel van de basisopstelling van deze PHP pagina's terug. De basis setup komt van een online bron – waarbij telkens op verschillende pagina's extra's zijn voorzien door ons. Deze extra's gaan van: een compleet nieuw script (license.php) als ook het voorzien van data voor de gebruiker en admin (allebei aparte view). Veel gegevens worden ook doorgestuurd naar de database – die opnieuw door ons is voorzien (nrplaat en parkingspot) buiten de “users” tabel.

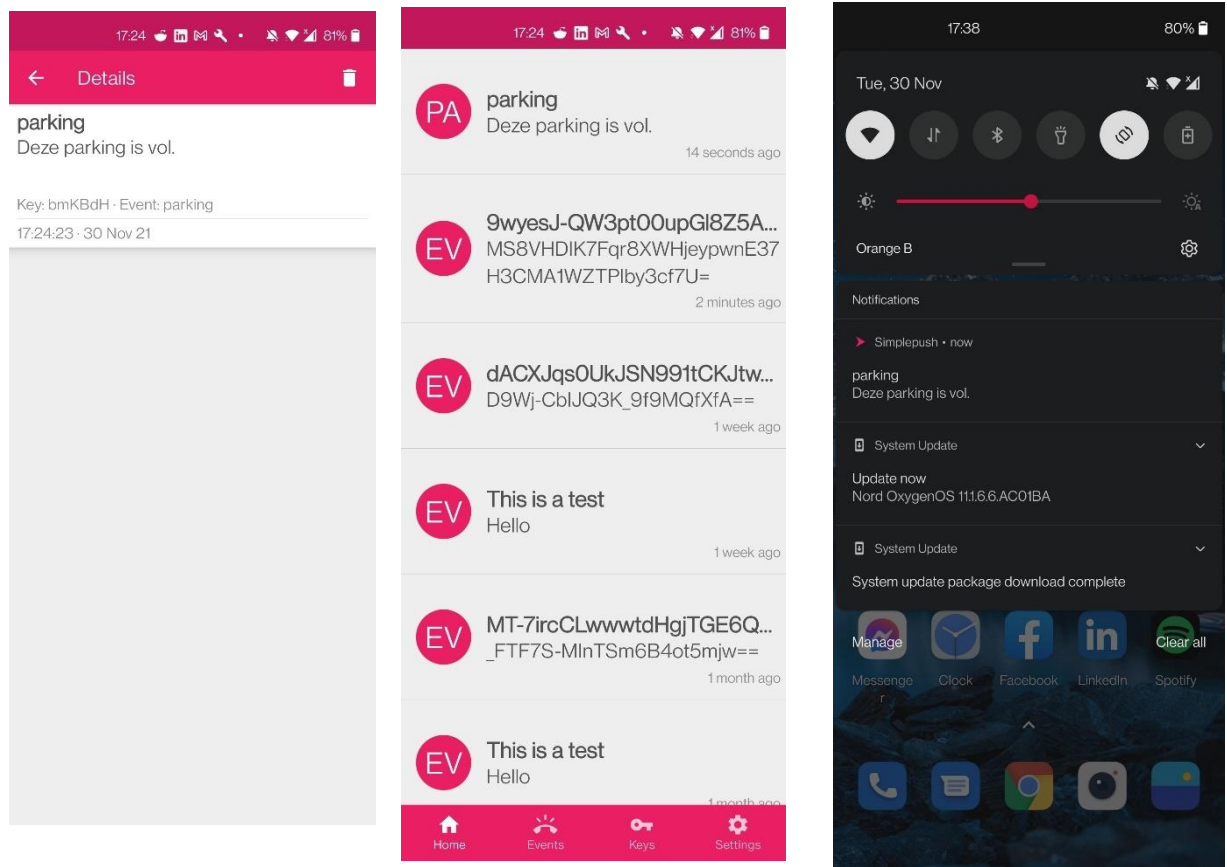
3.1.6 Push message uitzenden wanneer 4 parkeerplaatsen vol zijn (LDR + Push).

LDR:

Op elke parkeerplaats in de ondergrond zit een LDR die de hoeveelheid licht meet. Wanneer er een auto op de parkeerplaats staat dat meet de LDR veel minder licht en dan weet het programma dat er een auto op die parkeerplaats staat.

Push:

De owner krijgt een push bericht indien de parking volledig vol is. Deze ziet er als volgt uit:



4 ALGEMENE UITWERKING

4.1 Groepsbespreking: minimum requirements

Parking barrier	Ultrasoon sensor (servo, slagboom). Led: parkeergarage is vol.
Display entrance parking spaces	4 led lichtjes die aangeven welke parkeerplaats vol is of niet (a.h.v. LDR).
Photo USB cam en license plate analyze	Full HD 1080p webcam Nummerplaatherkenning.
Display (webinterface) + remote access	Lokaal (Ngrok) client kunnen bekijken. Website / interface fixen met data uit DB. Basis login pagina voorzien (Php MyLogin).
Push message + LDR	Licht voorzien voor parkeergarage (indien we in donkere hoek zitten). LDR op breadboard (1x) -> in echte setup 4x (groep voorziet andere 3). Push message kunnen aanleveren.
Database	Haal gegevens uit alle programma's (zie GitHub). -> aanmaken github en centrale plaats voor de "programma's". Test bestanden voor aanleveren en ontvangen gegevens. Van python bestanden naar MariaDB naar PHP webclient.

Tussendeadline: woensdag november 17 om 20:00. Tussentijdse update.

4.1.1 Youtube link

<https://youtu.be/CzePaAIzDDs>

4.1.2 Github link

[brandonmorren/parkeergarage \(github.com\)](https://github.com/brandonmorren/parkeergarage)

5 EXTRA CODE

5.1 Remote webinterface

Meer informatie per hoofdstuk kan je terugvinden in het bovenstaande hoofdstuk "remote webinterface".

5.1.1 Index.php

```
parkingiot.be > index.php
1  <?php
2  // Initialize the session
3  session_start();
4
5  // Check if the user is already logged in, if yes then redirect him to welcome page
6  if(isset($_SESSION["loggedin"]) && $_SESSION["loggedin"] === true){
7      header("location: welcome.php");
8      exit;
9  }
10
11 // Include config file
12 require_once "config.php";
13
14 // Define variables and initialize with empty values
15 $username = $password = "";
16 $username_err = $password_err = $login_err = "";
17
18 // Processing form data when form is submitted
19 if($_SERVER["REQUEST_METHOD"] == "POST"){
20
21     // Check if username is empty
22     if(empty(trim($_POST["username"]))){
23         $username_err = "Please enter username.";
24     } else{
25         $username = trim($_POST["username"]);
26     }
27
28     // Check if password is empty
29     if(empty(trim($_POST["password"]))){
30         $password_err = "Please enter your password.";
31     } else{
32         $password = trim($_POST["password"]);
33     }
34
35     // Validate credentials
36     if(empty($username_err) && empty($password_err)){
37         // Prepare a select statement
38         $sql = "SELECT id, username, password FROM users WHERE username = ?";
39
40         if($stmt = mysqli_prepare($link, $sql)){
41             // Bind variables to the prepared statement as parameters
42             mysqli_stmt_bind_param($stmt, "s", $param_username);
43
44             // Set parameters
45             $param_username = $username;
46
47             // Attempt to execute the prepared statement
48             if(mysqli_stmt_execute($stmt)){
49                 // Store result
50                 mysqli_stmt_store_result($stmt);
51
52                 // Check if username exists, if yes then verify password
53                 if(mysqli_stmt_num_rows($stmt) == 1){
54                     // Bind result variables
55                     mysqli_stmt_bind_result($stmt, $id, $username, $hashed_password);
56                     if(mysqli_stmt_fetch($stmt)){
57                         if(password_verify($password, $hashed_password)){
58                             // Password is correct, so start a new session
59                             session_start();
60                         }
59
60
```

```

61 // Store data in session variables
62 $SESSION["loggedin"] = true;
63 $SESSION["id"] = $id;
64 $SESSION["username"] = $username;
65
66 // Redirect user to welcome page
67 header("location: welcome.php");
68 } else{
69 // Password is not valid, display a generic error message
70 $login_err = "Invalid username or password.";
71 }
72 }
73 } else{
74 // Username doesn't exist, display a generic error message
75 $login_err = "Invalid username or password.";
76 }
77 } else{
78 echo "Oops! Something went wrong. Please try again later.";
79 }
80
81 // Close statement
82 mysqli_stmt_close($stmt);
83 }
84
85 // Close connection
86 mysqli_close($link);
87 }
88 }
89 }
90
91 <!DOCTYPE html>
92 <html lang="en">
93 <head>
94 <meta charset="UTF-8">
95 <title>Login</title>
96 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
97 <style>
98 body{ font: 14px sans-serif; }
99 .wrapper{ width: 360px; padding: 20px; }
100 </style>
101 </head>
102 <body>
103 <div class="wrapper">
104 <h2>Login</h2>
105 <p>Please fill in your credentials to login.</p>
106
107 <?php
108 if(!empty($login_err)){
109 echo '<div class="alert alert-danger">' . $login_err . '</div>';
110 }
111 }>
112
113 <form action=""<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>" method="post">
114 <div class="form-group">
115 <label>Username</label>
116 <input type="text" name="username" class="form-control <?php echo (empty($username_err)) ? 'is-invalid' : ''; ?>" value=""<?php echo $username;
117 <span class="invalid-feedback"><?php echo $username_err; ?></span>
118 </div>
119 <div class="form-group">
120 <label>Password</label>
121 <input type="password" name="password" class="form-control <?php echo (empty($password_err)) ? 'is-invalid' : ''; ?>"
122 <span class="invalid-feedback"><?php echo $password_err; ?></span>
123 </div>
124 <div class="form-group">
125 <input type="submit" class="btn btn-primary" value="Login">
126 </div>
127 <p>Don't have an account? <a href="register.php">Sign up now</a></p>
128 </form>
129 </div>
130 </body>
131 </html>

```

5.1.2 license.php

```

parkingiot.be > license.php
1  <?php
2  // Initialize the session
3  session_start();
4
5  // Check if the user is logged in, otherwise redirect to login page
6  if(!isset($_SESSION["loggedin"]) || $_SESSION["loggedin"] !== true){
7      header("location: index.php");
8      exit;
9  }
10
11 // Include config file
12 require_once "config.php";
13
14 // Define variables and initialize with empty values
15 $license = $license_err = "";
16 $achternaam = $achternaam_err = "";
17 $voornaam = $voornaam_err = "";
18 $straat_huis = $straat_huis_err = "";
19 $gemeente_post = $gemeente_post_err = "";
20 $bank = $bank_err = "";
21 $username = ($_SESSION["username"]);
22
23
24 // Check connection
25 if($link === false){
26     die("ERROR: Could not connect. " . mysqli_connect_error());
27 }
28
29 // header("location: welcome.php");
30 if($_SERVER["REQUEST_METHOD"] == "POST"){
31
32     // Validate password
33     if(empty(trim($_POST["license"]))) {
34         $license_err = "Please enter a password.";
35     } elseif(strlen(trim($_POST["license"])) < 6){
36         $license_err = "Password must have atleast 6 characters.";
37     } else{
38         $license = trim($_POST["license"]);
39     }
40
41     // Validate password
42     if(empty(trim($_POST["achternaam"]))) {
43         $achternaam_err = "Please enter a password.";
44     } elseif(strlen(trim($_POST["achternaam"])) < 6){
45         $achternaam_err = "Password must have atleast 6 characters.";
46     } else{
47         $achternaam = trim($_POST["achternaam"]);
48     }
49
50     // Validate password
51     if(empty(trim($_POST["voornaam"]))) {
52         $voornaam_err = "Please enter a password.";
53     } elseif(strlen(trim($_POST["voornaam"])) < 6){
54         $voornaam_err = "Password must have atleast 6 characters.";
55     } else{
56         $voornaam = trim($_POST["voornaam"]);
57     }
58
59     // Validate password
60     if(empty(trim($_POST["straat_huis"]))) {

```

```

60 if(empty(trim($_POST["straat_huis"]))) {
61     $straat_huis_err = "Please enter a password.";
62 } elseif(strlen(trim($_POST["straat_huis"])) < 6) {
63     $straat_huis_err = "Password must have atleast 6 characters.";
64 } else {
65     $straat_huis = trim($_POST["straat_huis"]);
66 }
67
68 // Validate password
69 if(empty(trim($_POST["gemeente_post"]))) {
70     $gemeente_post_err = "Please enter a password.";
71 } elseif(strlen(trim($_POST["gemeente_post"])) < 6) {
72     $gemeente_post_err = "Password must have atleast 6 characters.";
73 } else {
74     $gemeente_post = trim($_POST["gemeente_post"]);
75 }
76
77 // Validate password
78 if(empty(trim($_POST["bank"]))) {
79     $bank_err = "Please enter a password.";
80 } elseif(strlen(trim($_POST["bank"])) < 6) {
81     $bank_err = "Password must have atleast 6 characters.";
82 } else {
83     $bank = trim($_POST["bank"]);
84 }
85
86 // Check input errors before inserting in database
87 if(empty($license_err) && empty($achternaam_err) && empty($voornaam_err) && empty($straat_huis_err) && empty($gemeente_post_err) && empty($bank_err)) {
88
89     // Prepare an insert statement
90     $sql = "UPDATE nrplaat SET username='$username', achternaam='$achternaam', voornaam='$voornaam', straat_huisnummer='$straat_huis', gemeente_postcode='$gemeente_post', bank='$bank'";
91
92     if($stmt = mysqli_prepare($link, $sql)) {
93         // Bind variables to the prepared statement as parameters
94         mysqli_stmt_bind_param($stmt, "ss");
95
96
97         // Attempt to execute the prepared statement
98         if(mysqli_stmt_execute($stmt)) {
99             // Redirect to login page
100             header("location: welcome.php");
101         } else {
102             echo "Oops! Something went wrong. Please try again later.";
103             echo "ERROR: Could not able to execute $sql. " . mysqli_error($link);
104         }
105
106         // Close statement
107         mysqli_stmt_close($stmt);
108     }
109 }
110
111 // Close connection
112 mysqli_close($link);
113
114 }
115 ?>
116

```

```

parkingnot.be > license.php
111 // Close connection
112 mysqli_close($link);
113
114 }
115 ?>
116
117 <!DOCTYPE html>
118 <html lang="en">
119 <head>
120     <meta charset="UTF-8">
121     <title>Reset Password</title>
122     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
123     <style>
124         body{ font: 14px sans-serif; }
125         .wrapper{ width: 360px; padding: 20px; }
126     </style>
127 </head>
128 <body>
129     <div class="wrapper">
130         <h2>Create your license plate</h2>
131         <p>Please fill out this form to insert your license plate into our database, with additional information.</p>
132         <form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>" method="post">
133             <div class="form-group">
134                 <label>License plate</label>
135                 <input type="text" name="license" class="form-control <?php echo (empty($license_err)) ? 'is-invalid' : ''; ?>" value="<?php echo $license">
136                 <span class="invalid-feedback"><?php echo $license_err; ?></span>
137             </div>
138             <div class="form-group">
139                 <label>Achternaam</label>
140                 <input type="text" name="achternaam" class="form-control <?php echo (empty($achternaam_err)) ? 'is-invalid' : ''; ?>" value="<?php echo $achternaam">
141                 <span class="invalid-feedback"><?php echo $achternaam_err; ?></span>
142             </div>
143             <div class="form-group">
144                 <label>Voornaam</label>
145                 <input type="text" name="voornaam" class="form-control <?php echo (empty($voornaam_err)) ? 'is-invalid' : ''; ?>" value="<?php echo $voornaam">
146                 <span class="invalid-feedback"><?php echo $voornaam_err; ?></span>
147             </div>
148             <div class="form-group">
149                 <label>Straat + huisnummer</label>
150                 <input type="text" name="straat_huis" class="form-control <?php echo (empty($straat_huis_err)) ? 'is-invalid' : ''; ?>" value="<?php echo $straat_huis">
151                 <span class="invalid-feedback"><?php echo $straat_huis_err; ?></span>
152             </div>
153             <div class="form-group">
154                 <label>Gemeente + postcode</label>
155                 <input type="text" name="gemeente_post" class="form-control <?php echo (empty($gemeente_post_err)) ? 'is-invalid' : ''; ?>" value="<?php echo $gemeente_post">
156                 <span class="invalid-feedback"><?php echo $gemeente_post_err; ?></span>
157             </div>
158             <div class="form-group">
159                 <label>Bankrekeningnummer</label>
160                 <input type="text" name="bank" class="form-control <?php echo (empty($bank_err)) ? 'is-invalid' : ''; ?>" value="<?php echo $bank">
161                 <span class="invalid-feedback"><?php echo $bank_err; ?></span>
162             </div>
163             <input type="submit" class="btn btn-primary" value="Submit">
164             <a class="btn btn-link ml-2" href="welcome.php">Cancel</a>
165         </div>
166     </form>
167 </div>
168 </body>
169 </html>

```

5.1.3 Welcome.php

```
parkingiot.be > welcome.php
1  <?php
2  // Initialize the session
3  session_start();
4
5  require_once "config.php";
6
7  // Check if the user is logged in, if not then redirect him to login page
8  if(!isset($_SESSION["loggedin"]) || $_SESSION["loggedin"] !== true){
9      header("location: login.php");
10
11
12      exit;
13  }
14  ?>
15
16  <!DOCTYPE html>
17  <html lang="en">
18  <head>
19      <meta charset="UTF-8">
20      <title>Welcome</title>
21      <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
22      <style>
23          body{ font: 14px sans-serif; text-align: center; }
24      </style>
25  </head>
26  <body>
27      <h1 class="my-5">Hi, <b><?php echo htmlspecialchars($_SESSION["username"]); ?></b>. Welcome to our parking garage.</h1>
28      <p>
29          <a href="reset-password.php" class="btn btn-warning">Reset Your Password</a>
30          <a href="logout.php" class="btn btn-danger ml-3">Sign Out of Your Account</a>
31          <a href="license.php" class="btn btn-warning ml-3">Register your license plate</a>
32      </p>
33      <?php
34
35
36      if ($link->connect_error) {
37          die("Database connection failed: " . $link->connect_error);
38      }
39
40  ?>
41
42
43
44
45  <table border="1" align="center">
46      <h3>Available parking spots: </h2>
47      <tr>
48          <td>Parkingspot</td>
49          <td>Status</td>
50          <td>Bezet / beschikbaar sinds</td>
51      </tr>
52
53
54
55  <?php
56
57  $query2 = mysqli_query($link, "SELECT * from parkingspot")
58      or die (mysqli_error($link));
59
60  while ($row2 = mysqli_fetch_array($query2)) {
```

```

parkingiot.be > welcome.php
44
45 <table border="1" align="center">
46 <h3>Available parking spots: </h3>
47 <tr>
48 <td>Parkingspot</td>
49 <td>Status</td>
50 <td>Bezet / beschikbaar sinds</td>
51 </tr>
52
53
54
55 <?php
56
57 $query2 = mysqli_query($link, "SELECT * from parkingspot")
58 or die (mysqli_error($link));
59
60 while ($row2 = mysqli_fetch_array($query2)) {
61     echo
62     "<tr>
63         <td>{$row2['parkingspot']}</td>
64         <td>{$row2['status']}</td>
65         <td>{$row2['aankomst']}</td>
66     </tr>\n";
67 }
68 ?>
69
70
71
72 <div>
73 <table border="1" align="center">
74 <h3>Your personal information: </h3>
75 <tr>
76 <td>voornaam</td>
77 <td>achternaam</td>
78 <td>straat_huisnummer</td>
79 <td>gemeente_postcode</td>
80 <td>bank</td>
81 <td>nummerplaat</td>
82 <td>parkingspot</td>
83 <td>aankomst</td>
84 <td>vertrek</td>
85 <td>tebetalen</td>
86 </tr>
87 </div>
88
89
90
91 <?php
92
93 if ($_SESSION["username"] != "admin") {
94
95     $query = mysqli_query($link, "SELECT * FROM nrplaat WHERE username='".$_SESSION["username"]."'")
96     or die (mysqli_error($link));
97
98     while ($row = mysqli_fetch_array($query)) {
99         echo
100         "<tr>
101             <td>{$row['voornaam']}</td>
102             <td>{$row['achternaam']}</td>
103             <td>{$row['straat_huisnummer']}</td>

```



```

parkingiot.be > welcome.php
97
98 while ($row = mysqli_fetch_array($query)) {
99     echo
100     "<tr>
101         <td>{$row['voornaam']}

```

```

157  <?php } ?>
158
159
160  <?php
161
162  if ($_SESSION["username"] == "admin") {
163
164
165      $query3 = mysqli_query($link, "SELECT * FROM users")
166      or die (mysqli_error($link));
167
168      while ($row3 = mysqli_fetch_array($query3)) {
169          echo
170          "<tr>
171              <td>{$row3['id']}</td>
172              <td>{$row3['username']}</td>
173              <td>{$row3['password']}</td>
174              <td>{$row3['created_at']}</td>
175          </tr>\n";
176      }
177  }
178
179  }
180
181
182  ?>
183
184
185
186  </body>
187  </html>
188
189
190

```

De overige pagina's (register.php, reset-password) blijven zo goed als onveranderd van de basis setup. Deze kan je steeds bekijken door in bibliografie te gaan kijken naar de link in verwijzing.

5.1.4 main.py

Onze algemene code van het programma met de slagboom, LDR,... kan je terugvinden via de Github link die eerder vermeld werd:

[brandonmorren/parkeergarage \(github.com\)](https://github.com/brandonmorren/parkeergarage)

5.1.5 TestFile.py

Testfile die wordt gebruikt bij het voorzien van data van webinterface / python programma naar de database / webinterface visualisatie. Deze wordt geïntegreerd met onze main.py file (de basis voor ons project).

```

petengotche> testfile.py ~
1
2 #Log toevoegen indien nodig
3 import MySQLdb
4 from datetime import datetime
5 from datetime import date
6
7 #Definitie voor laatste toevoeging TE BETALEN. Mag "ergens in file" bij andere functies
8 def calculate_betalen(extro):
9     hours = extro.hour
10    minutes = extro.minute
11    if minutes >= 30:
12        hours_total = hours + 1
13        day = (extro.day) + 24
14        hours_total = hours_total + day
15        tebetalen = hours_total * parkeertarief
16        tebetalen = round(tebetalen, 2)
17        tebetalen = str(tebetalen)
18        return tebetalen
19    else:
20        day = (extro.day) + 24
21        tebetalen = (day + hours) * parkeertarief
22        tebetalen = round(tebetalen, 2)
23        tebetalen = str(tebetalen)
24        return tebetalen
25
26
27 # Start al in de file : is extra voor testing
28 database = MySQLdb.connect(host="localhost", user="pi", passwd="raspberry", db="parkeergarage")
29 cursor = database.cursor()
30
31 # Test data. Nummerplaat en tijdstip STAAN AL IN DE PYTHON FILE! Enkel "vertrek" en parkeertarief toevoegen voor een "realistische tijd" die langer dan 2 minuten duurt
32 nummerplaat = "1-ABZ-456"
33
34 #Parkeerspot moet niet toegevoegd worden, - de waarde die komt vanuit de "LDK" -> LDW -> parking nummer # (enkel INT, geen STR. Parking = 1, of 2, of 3, maar niet "parking 1")
35 parkingspot = "3"
36 tijdstip = datetime.now()
37
38 #Enkel deze twee toevoegen aan python file: parkeertarief = te betalen bedrag en vertrek = "doen alsof auto later vertrekt"
39 parkeertarief = 2
40 vertrek = datetime(2021,11,29)
41 status_aankomst = "bezet"
42 status_vertrek = "beschikbaar"
43
44
45
46 #Eerste aanroeping in onze webpagina: voegt aankomst aan slagboom + nummerplaat toe aan DB
47 cursor.execute("INSERT INTO nrplaat(aankomst,nummerplaat) VALUES(%s,%s) ON DUPLICATE KEY UPDATE aankomst = CURRENT_TIMESTAMP", (tijdstip,nummerplaat))
48 database.commit()
49
50
51 # PER PARKEERPLAATS: voegt parkeerplaats toe in DB bij nummerplaat gescannede slagboom, daarnaast ook in de "parkingspot" DB om beschikbaarheid / parkeerplaats weer te geven
52 cursor.execute("UPDATE nrplaat SET parkingspot=%s WHERE nummerplaat=%s", (parkingspot, nummerplaat))
53 cursor.execute("UPDATE parkingspot SET status=%s, aankomst=CURRENT_TIMESTAMP WHERE parkingspot=%s", (status_aankomst, parkingspot))
54 database.commit()
55
56
57 # Einde parkeerbezoek - bij vertrek slagboom. Voegt vertrektijd bij nummerplaat toe in DB
58 cursor.execute("UPDATE nrplaat SET vertrek=%s WHERE nummerplaat=%s", (vertrek, nummerplaat))
59 cursor.execute("UPDATE parkingspot SET status=%s, aankomst=CURRENT_TIMESTAMP WHERE parkingspot=%s", (status_vertrek, parkingspot))
60 database.commit()
61
62
63 # WERKT ENKEL ALS AANKOMST EN VERTREK IN DB STAAN (zie boven): berekent het "te betalen" bedrag ; helemaal achteraan
64 cursor.execute("SELECT aankomst, vertrek FROM nrplaat WHERE nummerplaat=%s", [nummerplaat])
65 data = cursor.fetchall()
66
67 try:
68     for row in data:
69         aankomen = row[0]
70         vertrekken = row[1]
71         print("aankomst = ", row[0], )
72         print("vertrek = ", row[1])
73         test = vertrekken - aankomen
74         print(test)
75         print("-----")
76         newDate = datetime.strptime(str(test), "%H:%M:%S")
77         hours = newDate.hour
78         minutes = newDate.minute
79         if minutes >= 30:
80             hours_total = hours + 1
81             print(hours_total)
82             tebetalen = hours_total * parkeertarief
83             tebetalen = round(tebetalen, 2)
84             tebetalen = str(tebetalen)
85             print(tebetalen)
86         else:
87             tebetalen = hours * parkeertarief
88             tebetalen = round(tebetalen, 2)
89             tebetalen = str(tebetalen)
90             print(tebetalen)
91
92 except ValueError:
93     try:
94         oneday = datetime.strptime(str(test), "%d day, %H:%M:%S")
95         print("There is one additional day.")
96         tebetalen = calculate_betalen(oneday)
97         print(tebetalen)
98
99 except ValueError:
100     days = datetime.strptime(str(test), "%d days, %H:%M:%S")
101     print("There are more than 1 day.")
102     tebetalen = calculate_betalen(days)
103     print(tebetalen)
104
105 cursor.execute("UPDATE nrplaat SET tebetalen=%s WHERE nummerplaat=%s", (tebetalen, nummerplaat))
106 database.commit()
107
108

```

6 BIBLIOGRAFIE

- Docenten IoT essentials. (2021, November 2). *Chapter5_Python3_GPIO*.
Opgehaald van IoT essentials:
https://thomasmore.instructure.com/courses/15211/files/2240798?module_item_id=680300
- Documentation*. (sd). Opgehaald van ngrok.com: ngrok.com/docs
- licence plate recognition using openCV & python*. (sd). Opgehaald van medium.com: <https://medium.com/programming-fever/license-plate-recognition-using-opencv-python-7611f85cdd6c>
- OCR with OpenCV & Tesseract*. (sd). Opgehaald van maker.pro: <https://maker.pro/raspberry-pi/tutorial/optical-character-recognizer-using-raspberry-pi-with-opencv-and-tesseract>
- OCR with tesseract*. (sd). Opgehaald van nanonets.com: <https://nanonets.com/blog/ocr-with-tesseract/>
- php mysql login system*. (sd). Opgehaald van tutorialrepublic.com: <https://www.tutorialrepublic.com/php-tutorial/php-mysql-login-system.php>
- Python how to capture image from wabcam*. (sd). Opgehaald van stackoverflow.com: <https://stackoverflow.com/questions/34588464/python-how-to-capture-image-from-webcam-on-click-using-opencv/34588758#34588758>
- Raspberry Pi Tutorials. (2021, November 2). *Raspberry Pi Servo Motor control*.
Opgehaald van Raspberry Pi Tutorials: <https://tutorials-raspberrypi.com/raspberry-pi-servo-motor-control/>