

OTECH

Productiedossier



Kevin Leemans
Dhr. Frederick Roegiers
CMS - Development
2019-2020

<u>Discover</u>	3
Briefing	3
<u>Define</u>	6
Analyse	6
Planning	7
<u>Design</u>	8
Sitemap Website	8
Wireframes Website	9
Wireflow Mobile webapp	11
Style guide	12
Visual Designs	15
<u>Develop</u>	18
Code Snippets	18
Schermafbeeldingen	21
<u>Deliver</u>	22
Handleiding	22
<u>Deploy</u>	23
Deployment guide	23

Discover

Briefing

Introduction

Jouw klant ARTE-TECH, een technisch onderhoudsbedrijf komt met de vraag om de opvolging van hun werken en opvolging van het personeel, aannemers, onderaannemers, ... te optimaliseren.

Activiteiten van de klant

ARTE-TECH biedt de dienst aan om technisch geschoold mensen gedurende een bepaalde periode ter plaatse naar bedrijven te sturen om daar gevraagde onderhoudswerken uit te voeren. Zowel tijdens een shutdown als voor het dagelijks onderhoud zorgen ze ervoor dat de juiste mensen snel op locatie zijn om assistentie te bieden of volledig autonoom een machine te repareren.

Probleemstelling

ARTE-TECH beschikt enerzijds over eigen personeel, maar beschikt anderzijds ook over een aantal onderaannemers (vergelijkbaar met freelancers) die ze - in naam van ARTE-TECH - kunnen uitsturen naar bedrijven. De opvolging hiervan is moeilijk want iedereen heeft een andere manier van werken.

De freelancers werken met eigen werkbonnen (startuur, einduur, activiteiten van de dag, verplaatsingen, ...).

Personnel houdt dan weer alle uren en activiteiten op papier bij en laat die wekelijks ondertekenen door hun klant (het bedrijf waar ze onderhoud doen).

Elke week verzamelt de manager van ARTE-TECH alle paperassen, werkbonnen en e-mails en stelt op basis daarvan de factuur op.

Maar er geraken al eens papieren, e-mails verloren en sommige zaken worden vergeten door gebrek aan uniformiteit.

Er is een oplossing nodig.

Opdracht

De klant heeft weinig kaas gegeten van computers, maar beseft wel dat zijn manier van werken een stuk gemakkelijker zou kunnen gemaakt worden door het maken van een webapplicatie. Hij had zelf het volgende idee:

Een all-in-one webapplicatie waarbij hij de uren niet meer zelf hoeft in te vullen, maar reeds gebeurd is door het personeel / ondераannemers die hij uitstuurt.

Mobiele WebApp

Het personeel/ondераannemers zullen via hun smartphone via een progressive web app - na inloggen - dagelijks het volgende kunnen invullen:

- klant
- datum
- startuur
- einduur
- pauze
- uitgevoerde activiteiten
- gebruikte materialen (vb schroeven, kabels, sensoren, ...)
- transport (aantal kilometers)
- Onderaannemers hebben ook een extra optie waarbij ze hun tarief kunnen wijzigen (uurtarief, transportkost)

Het zou handig zijn als de gebruikers van die app ook kunnen bekijken hoeveel uren ze in totaal al gepresteerd hebben (per week / per maand / per klant). Bedenk zelf een aantal praktische toevoegingen.

Backend

In de backend worden alle medewerkers van ARTE-TECH toegevoegd.

In de backend worden alle klanten van ARTE-TECH toegevoegd.

Het uurtarief van de techniekers en transportkost wordt per bedrijf op voorhand afgesproken en ingevoerd in de backend.

Volgende regel geldt altijd voor uurtarief bij uitzonderingen:

- overuren tijdens de week (> 8 uur): 20% meer
- zaterdag: 50% meerkost
- zondag: dubbel tarief

De manager kan op een gebruiksvriendelijke manier de gepresteerde uren opvolgen, die doorgegeven zijn door zijn personeel/ondераannemers (en zo nodig ook corrigeren of handmatig invullen).

De backend zou een functie moeten hebben dat een periode kan opgesteld worden en kan gemeld worden naar de klant (het bedrijf) per mail.

Deze klant moet de periode nakijken via de website (zie hieronder) en bevestigen. (tot nog toe gebeurde dat immers op papier)

Via de backend moet het mogelijk zijn om totaalprijs per periode te berekenen en dit te exporteren naar een overzichtelijke xls / pdf.

Je hoeft geen rekening te houden met BTW (alles excl. btw)

Facturatie is niet nodig (maar mag uiteraard)

Website Klant

Elke klant (de bedrijven dus) kan de gepresteerde uren nakijken via hun computer.

Zij volgen dit op per periode (na notificatie per mail), waarbij ze in detail alle prestaties kunnen nakijken en kunnen bevestigen.

Indien er iets zou mis zijn, hebben ze de mogelijkheid om een opmerking te versturen.

Er is ook een mogelijkheid om een periode als overzichtelijke pdf te printen voor eigen administratie.

Define

Analyse

Mobiele Webapp

Eenvoudige UI waarbij er na inloggen op de homepagina alle mogelijke in te vullen info bevat van die dag:

- klant
- datum
- startuur
- einduur
- pauze
- uitgevoerde activiteiten
- gebruikte materialen (vb schroeven, kabels, sensoren, ...)
- transport (aantal kilometers)
- Onderaannemers hebben ook een extra optie waarbij ze hun tarief kunnen wijzigen (uurtarief, transportkost)

Met mogelijkheid om te navigeren naar een soort van persoonlijk overzicht van gepresteerde uren per week / maand /klant en misschien nog verdere mogelijkheden. Gewoon duidelijk overzicht waar je in verschillende secties alle gegevens kan zien en vandaar nog eens doorklikken naar een detail. Het lijkt mij het beste als het openen van de homepagina (eens aangemeld) direct het in te vullen van de gegevens voor die dag is aangezien je dit het meeste zal nodig hebben. Verder heeft de gebruiker niet veel meer nodig.

Backend

Dashboard cms-style back-office voor het toevoegen van medewerkers van ARTE-TECH en klanten van ARTE-TECH.

De layout zal een basic dashboard zijn met overzicht van de verschillende klanten en medewerkers met alle CRUD mogelijkheden.

Website

Layout wordt ook een dashboard style back-office waarin de klant dan alle gepresteerde uren kan bekijken en opvolgen per periode. Kunnen hierop verder doorgaan om alles in detail te bekijken en eventueel aanpassingen of opmerkingen doorsturen.

Planning

25 Nov – 1 Dec:

Productiedossier – Discover, Define

2 Dec – 8 Dec :

Productiedossier – Design

9 Dec – 15 Dec:

BackOffice

16 Dec – 22 Dec:

Website Klant

23 Dec – 29 Dec:

Mobiele web-app

30 Dec – 3 Jan:

Productiedossier – Develop

Productiedossier – Deliver

Productiedossier – Deploy

Design

Sitemap Website

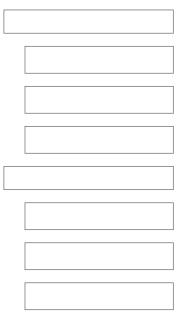
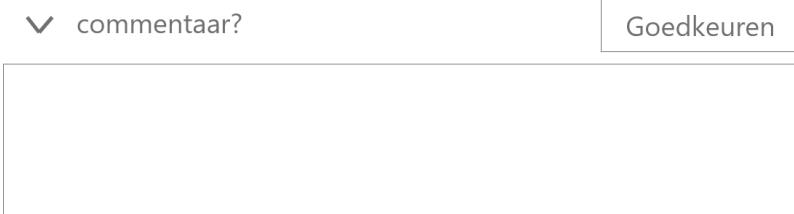
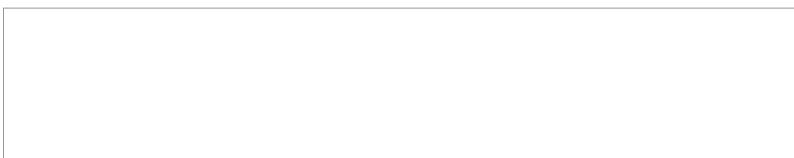


Wireframes Website

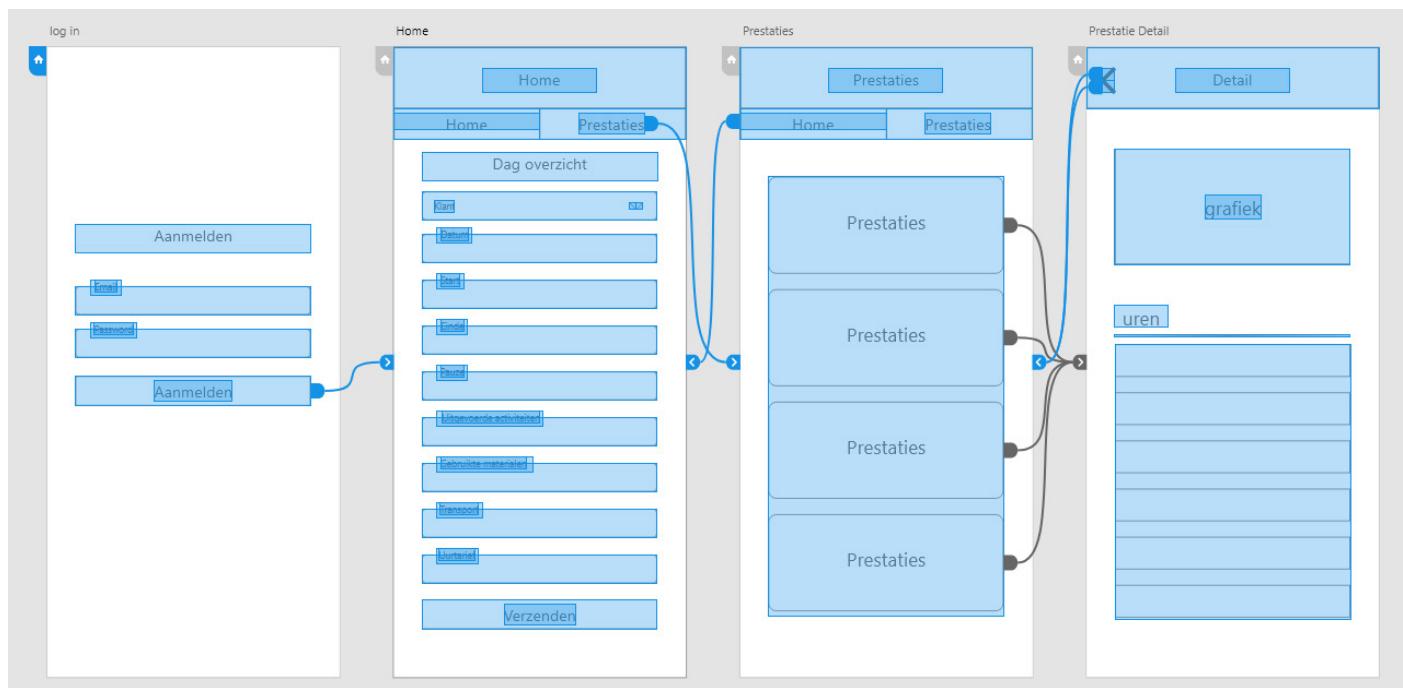
Aanmelden

logo	
logo	

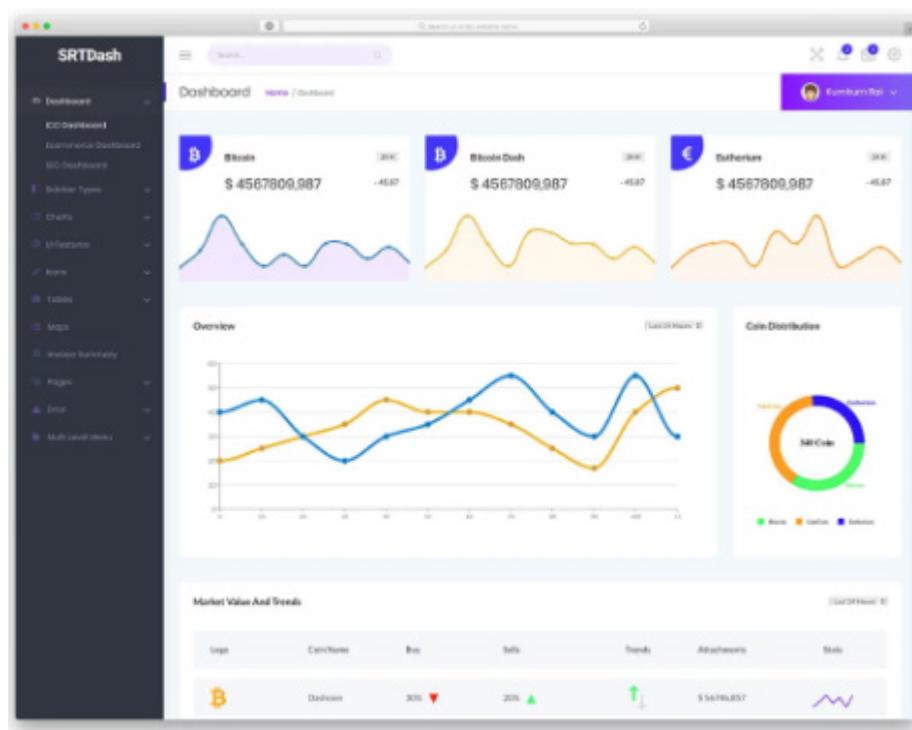
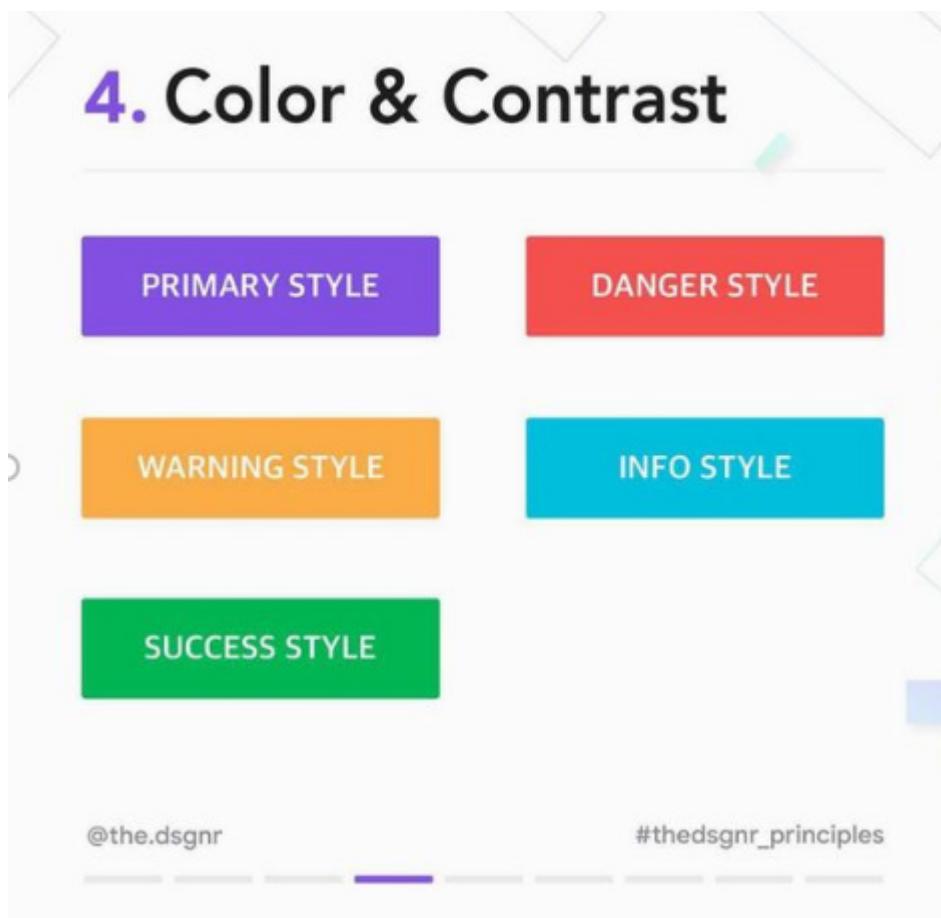
logo	Profiel			logo				
		Welkom						
		prestatie 1	prestatie 2					
		prestatie 3	prestatie 4					
<table border="1"><tr><td>logo</td><td>periode</td><td>▼</td></tr></table>						logo	periode	▼
logo	periode	▼						

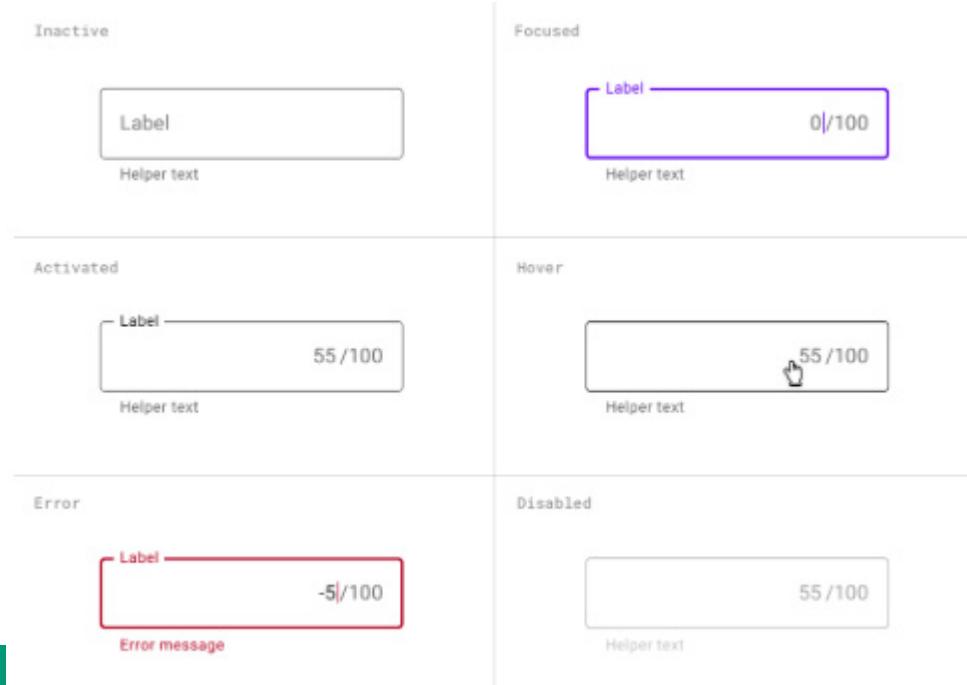
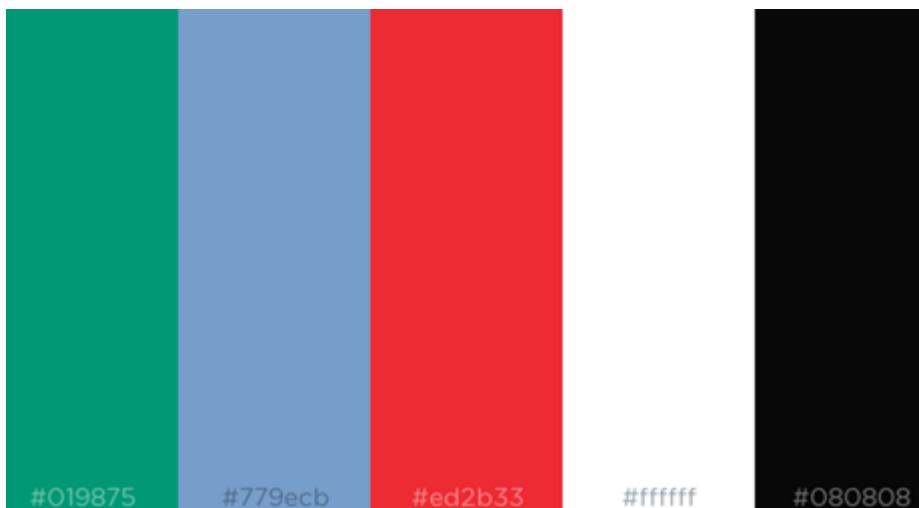
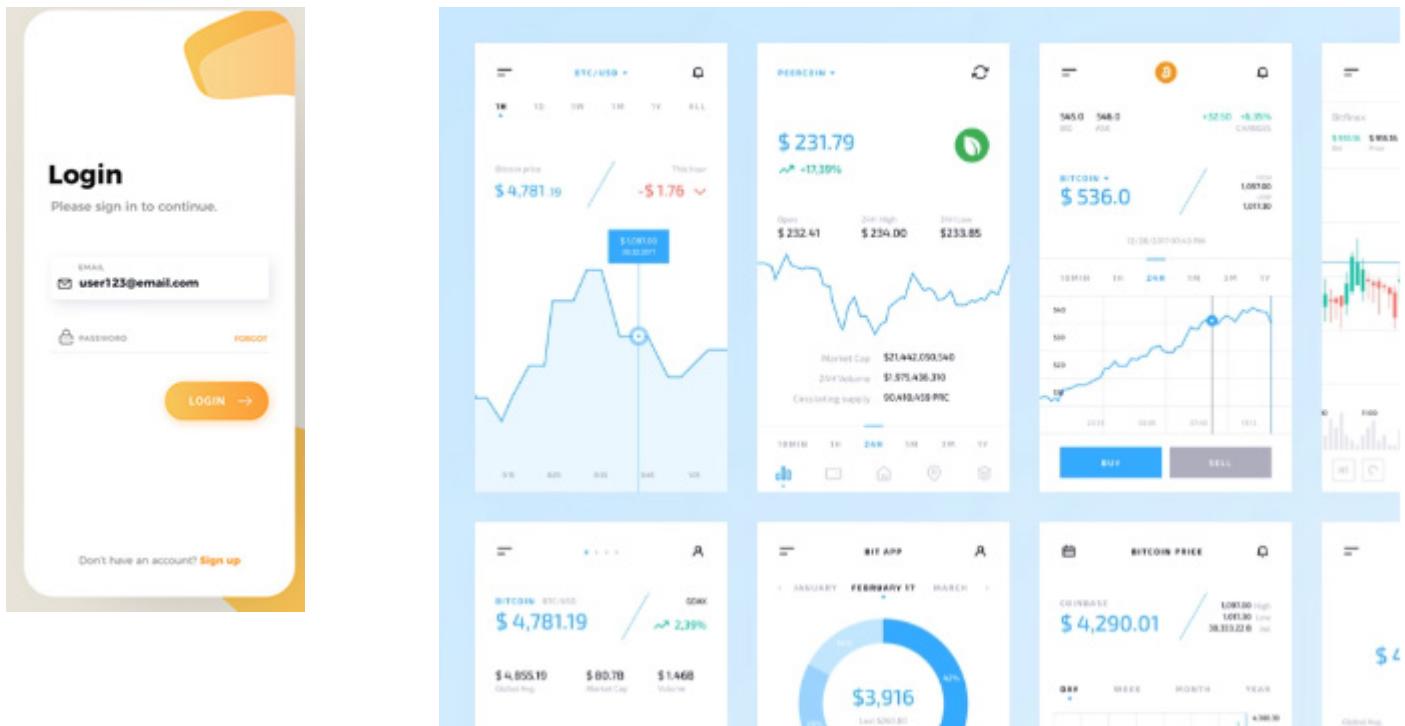
logo		Profiel	logo
		Prestatie	
		grafiek	
		Detail	
			
		 <p>▼ commentaar?</p> <p>Goedkeuren</p>	
			
		 <p>verstuur opmerking</p>	

Wireflow Mobile webapp



Style guide







Quicksand



Open Sans



Q TECH

Visual Designs

The screenshot shows a teal-colored login page. At the top is the 'o TECH' logo, where the 'o' is orange and outlined, and 'TECH' is in green. Below the logo is the text 'Aanmelden'. There are two input fields: one for 'Email' and one for 'Password', both with placeholder text. A large green button labeled 'Aanmelden' is at the bottom.

The screenshot shows a blue header bar with the word 'Home'. Below it is a navigation bar with 'Home' and 'Prestaties' buttons. The main content area is titled 'Dag overzicht' (Daily Overview). It contains several input fields with placeholder text: 'Klant' (Client) with a dropdown arrow, 'Datum' (Date), 'Start' (Start), 'Einde' (End), 'Pauze' (Break), 'Uitgevoerde activiteiten' (Performed activities), 'Gebruikte materialen' (Used materials), and 'Transport' (Transport). At the bottom is a blue button labeled 'Verzenden' (Send).

Prestaties

Home

Prestaties

Prestaties

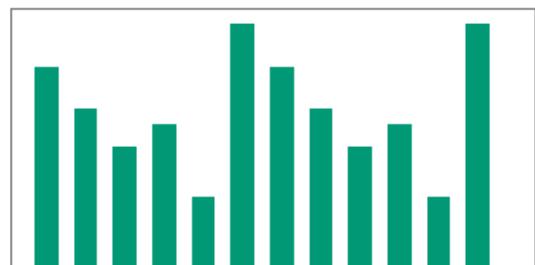
Prestaties

Prestaties

Prestaties

Details

Prestaties



Uren

04/12/2109 9u00 15euro Designen



Aanmelden

<input type="text"/>	<input type="password"/>
<input type="checkbox"/>	<input type="password"/>
aanmelden	

Bedrijf klant Profile Logout

Welkom

periode

prestatie 1 sub-prestatie sub-prestatie sub-prestatie sub-prestatie	Prestatie 1
prestatie 2 sub-prestatie sub-prestatie sub-prestatie sub-prestatie	Prestatie 3
prestatie 3 sub-prestatie sub-prestatie sub-prestatie sub-prestatie	Prestatie 2
prestatie 4 sub-prestatie sub-prestatie sub-prestatie sub-prestatie	Prestatie 4

Bedrijf klant Profile Logout

Prestatie

prestatie 1 sub-prestatie sub-prestatie sub-prestatie sub-prestatie	Detail
prestatie 2 sub-prestatie sub-prestatie sub-prestatie sub-prestatie	
prestatie 3 sub-prestatie sub-prestatie sub-prestatie sub-prestatie	
prestatie 4 sub-prestatie sub-prestatie sub-prestatie sub-prestatie	

Detail

Werknemer	Datum	Gepresteerde uren	Activiteiten	Materialen	Transport	Uur Tarief
Kevin Leemans	04/12/2019	9u00	Designen	adobe Xd	Auto	15 euro
Kevin Leemans	05/12/2019	9u00	Designen	adobe Xd	Auto	15 euro
Kevin Leemans	06/12/2019	9u00	Designen	adobe Xd	Auto	15 euro

commentaar? Goedkeuren

verstuur opmerking

Develop

Code Snippets

The screenshot shows a Symfony project structure in a code editor. The current file is `base.html.twig` located in the `templates/partials` directory. The code defines a base template with includes for `headNavBar.html.twig`, `sideNav.html.twig`, and `scripts.css`. It also includes partials for `headNavBar`, `sideNav`, and `scripts`.

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>% block title %>Welcome!% endblock %</title>
        {% include 'partials/stylesheets' %}
    </head>
    <body>
        {% include 'partials/headNavBar' %}
        {% include 'partials/sideNav.html.twig' %}
        {% block body %}
        {% endblock %}
        {% block javascripts %}{% include 'partials/scripts' %}{% endblock %}
    </body>
</html>
```

The screenshot shows a Symfony project structure in a code editor. The current file is `SecurityController.php` located in the `Controller` directory. The code defines a `SecurityController` class that extends `AbstractController`. It contains methods for `login` and `logout`. The `login` method checks if a user is logged in and redirects them if they are. It also handles login errors and remembers the last username. The `logout` method throws an exception.

```
<?php
namespace App\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;
use Symfony\Component\Security\Http\Authentication\AuthenticationUtils;

class SecurityController extends AbstractController
{
    /**
     * @Route("/login", name="app_login")
     * @param AuthenticationUtils $authenticationUtils
     * @return Response
     */
    public function login(AuthenticationUtils $authenticationUtils): Response
    {
        if ($this->getUser()) {
            return $this->redirectToRoute('app_index');
        }

        // get the login error if there is one
        $error = $authenticationUtils->getLastAuthenticationError();
        // last username entered by the user
        $lastUsername = $authenticationUtils->getLastUsername();

        return $this->render('security/login', ['last_username' => $lastUsername, 'error' => $error]);
    }

    /**
     * @Route("/logout", name="app_logout")
     */
    public function logout()
    {
        throw new \Exception('This method can be blank - it will be intercepted by the logout key on your firewall');
    }
}
```

```

/* The side navigation menu */
.sidenav {
    height: 100%; /* 100% Full-height */
    width: 250px; /* 0 width - change this with JavaScript */
    position: fixed; /* Stay in place */
    z-index: 1; /* Stay on top */
    top: 80px; /* Stay at the top */
    left: 0;
    background-color: #029875; /* Black*/
    color: white;
    overflow-x: hidden; /* Disable horizontal scroll */
    padding-top: 40px; /* Place content 60px from the top */
    box-shadow:
        0 2.8px 2.2px rgba(0, 0, 0, 0.034),
        0 6.7px 5.3px rgba(0, 0, 0, 0.048),
        0 12.5px 10px rgba(0, 0, 0, 0.06),
        0 22.3px 17.9px rgba(0, 0, 0, 0.072),
        0 41.8px 33.4px rgba(0, 0, 0, 0.086),
        0 100px 80px rgba(0, 0, 0, 0.12);
}

/* The navigation menu links */
.sidenav a {
    padding: 8px 8px 8px 32px;
    text-decoration: none;
    font-size: 25px;
    color: #0DD;
    display: block;
    transition: 0.3s;
}

/* When you mouse over the navigation links, change their color */
.sidenav a:hover {
    color: white;
}

/* Position and style the close button (top right corner) */
.sidenav .closebtn {
    position: absolute;
    top: 0;
    right: 25px;
    font-size: 36px;
}

```

```

security:
    # https://symfony.com/doc/current/security.html#where-do-users-come-from-user-providers
    access_control:
        - { path: ^/login$, roles: IS_AUTHENTICATED_ANONYMOUSLY }
    encoders:
        App\Entity\User:
            algorithm: bcrypt
    providers:
        our_db_provider:
            entity:
                class: App\Entity\User
                property: email
    firewalls:
        dev:
            pattern: ^/_(profiler|wdt)|css|images|js/
            security: false
        main:
            anonymous: lazy
            guard:
                authenticators:
                    - App\Security\LoginFormAuthenticator
            logout:
                path: app_logout
                # where to redirect after logout
                target: app_login

            # activate different ways to authenticate
            # https://symfony.com/doc/current/security.html#firewalls-authentication

            # https://symfony.com/doc/current/security/impersonating_user.html
            # switch_user: true

```

Screenshot of the PhpStorm IDE showing the code editor with the file `LoginFormAuthenticator.php` open. The code implements a `LoginFormAuthenticator` class that extends `AbstractFormLoginAuthenticator` and implements `PasswordAuthenticatedInterface`. It uses `TargetPathTrait` and various Symfony services like `$entityManager`, `$urlGenerator`, `$csrfTokenManager`, `$passwordEncoder`, and `$security`.

```

namespace App\Security;

use ...;

class LoginFormAuthenticator extends AbstractFormLoginAuthenticator implements PasswordAuthenticatedInterface
{
    use TargetPathTrait;

    private $entityManager;
    private $urlGenerator;
    private $csrfTokenManager;
    private $passwordEncoder;
    private $security;

    public function __construct(EntityManagerInterface $entityManager, UrlGeneratorInterface $urlGenerator, CsrfTokenManagerInterface $csrfTokenManager, PasswordEncoderInterface $passwordEncoder, SecurityInterface $security)
    {
        $this->entityManager = $entityManager;
        $this->urlGenerator = $urlGenerator;
        $this->csrfTokenManager = $csrfTokenManager;
        $this->passwordEncoder = $passwordEncoder;
        $this->security = $security;
    }

    public function supports(Request $request)
    {
        return 'app_login' === $request->attributes->get('key: '_route')
            && $request->isMethod('POST');
    }

    public function getCredentials(Request $request)
    {
        $credentials = [
            'email' => $request->request->get('key: 'email'),
            'password' => $request->request->get('key: 'password'),
            'csrf_token' => $request->request->get('key: '_csrf_token'),
        ];
        $request->getSession()->set(
            Security::LAST_USERNAME,
            $credentials['email']
        );
    }

    return $credentials;
}

```

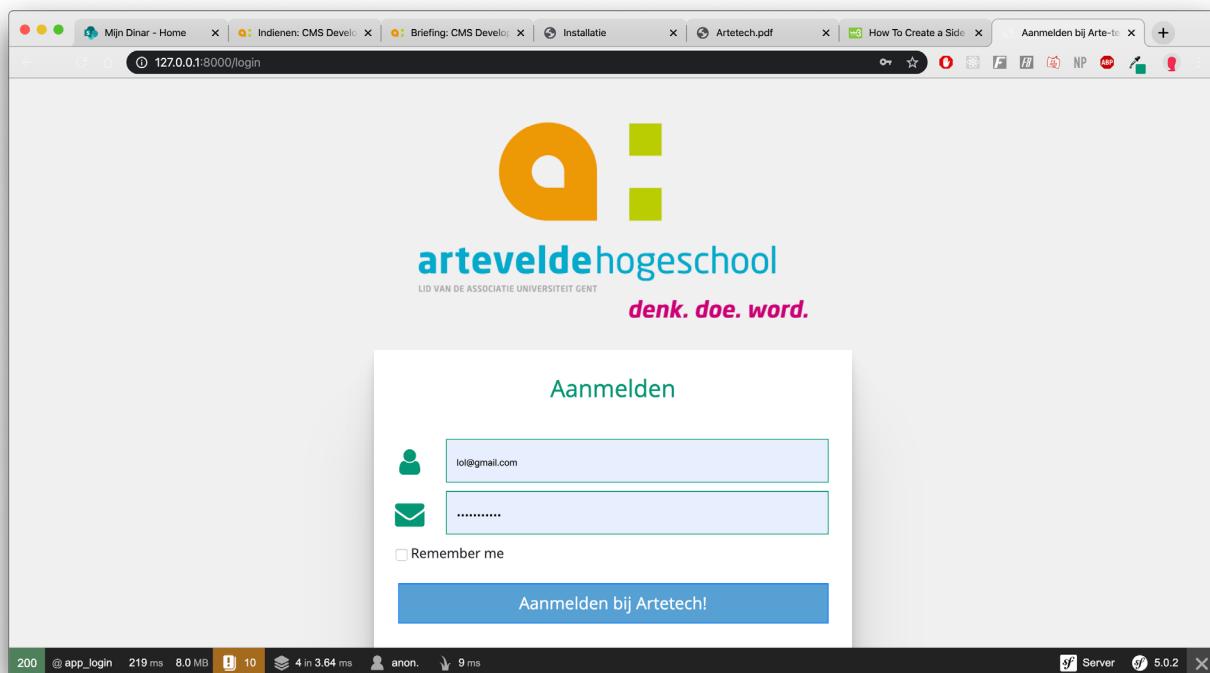
Screenshot of the PhpStorm IDE showing the code editor with the file `login.html.twig` open. This is a Twig template for a login form. It includes a header with meta tags and a title. It checks for errors and displays them in an alert box. The main content is a container with two columns: one for the logo and one for the form. The form itself has two rows: one for the email input and one for the password input, both with accompanying icons. It also includes a csrf token field and a remember me checkbox.

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>{% block title %}Anmelden bei Arte-tech CMS!{% endblock %}</title>
        {% include 'partials/stylesheet' %}
    </head>
    <body>
        {% if error %}
            <div class="alert alert-danger">{{ error.messageKey|trans(error.messageData, 'security') }}</div>
        {% endif %}
        <div class="container">
            <div class="row">
                <div class="col-md-2 col-lg-6">
                    
                <div class="col-md-12 col-lg-6 authentication-form">
                    <div class="row input-row">
                        <div class="icon-authentication-form">
                            <i class="fa fa-user icon"></i>
                        </div>
                        <input type="text" value="{{ last_username }}" name="email" id="inputEmail" class="form-control" required="" autofocus="autofocus">
                    </div>
                    <div class="row input-row">
                        <div class="icon-authentication-form">
                            <i class="fa fa-envelope icon"></i>
                        </div>
                        <input type="password" name="password" id="inputPassword" class="form-control" required="required">
                    </div>
                    <input type="hidden" name="csrf_token" value="{{ csrf_token('authenticate') }}>
                </div>
                <div class="checkbox mb-3">
                    <label>
                        <input type="checkbox" name="remember_me"> Remember me
                    </label>
                </div>
            </div>
        </div>
    </body>

```

Schermafbeeldingen



Deliver

Handleiding

Mobiele webapp

De werknemer kan via de mobiele webapp aanmelden via zijn email adres en eens aangemeld kan hij alle gegevens van die gewerkte dag invullen en versturen naar de database. Hij kan ook zijn eigen prestaties raadplegen van de verschillende klanten waarvoor hij heeft gewerkt. Doormiddel van een gemakkelijke navigatie bovenaan de applicatie kan hij switchen tussen zijn gegevens invullen van die dag en de prestaties bekijken. Enkel de werknemer met geldige log in gegevens kan zich authenticeren in de app en de app gebruiken.

Website

De website kan worden gebruikt door klanten van Artetech. Klanten kunnen met geldige log in gegevens aanmelden bij deze website en eens geauthenticeerd kunnen ze alle periodes bekijken en de bijhorende opgelverde prestaties van de werknemers. Ze kunnen deze goedkeuren of een opmerking versturen. De layout is een dashboard style. Ze kunnen via de grote blokken op de homepagina navigeren naar de verschillende opgeleverde prestaties. Via een handige knop bovenaan kunnen ze de periode in pdf afprinten of veranderen van periode.

CMS

CMS is enkel toegankelijk voor een admin rol. Geen andere persoon kan zich aanmelden bij de dashboard van artetech. De layout is een zelfde layout als de dashboard van de klant enkel met andere functionaliteiten. Zo kunnen hier Werknemers en Klanten worden toegevoegd net als periodes voor klanten en de gepresterde uren van werknemers.

Deploy

Deployment guide

Mobiele webapp

1. Clone de github repo van <https://github.com/kevileem2/artetech> lokaal op uw machine.
2. Zorg dat u node package manager heeft geïnstalleerd (of yarn).
3. Open de locatie van de app in uw terminal en doe npm install (of yarn install). Dit zal alle node modules installeren die u nodig heeft.
4. Daarna voert u npm start uit en u kan de app opstarten.

Website & CMS

1. Clone de github repo van <https://github.com/kevileem2/artetech> lokaal op uw machine.
2. Zorg dat u MAMP geïnstalleerd heeft staan
3. Zorg dat u PHP geïnstalleerd heeft staan
4. Zorg dat u symfony geïnstalleerd heeft staan.
5. Open de file locatie van het CMS met uw terminal.
6. Voer de nodige migrations uit:
7. php bin/console make:migration
8. php bin/console doctrine:create:migration
9. Start daarna de server op MAMP
10. Voer symfony server:start uit in uw terminal om de applicatie op te starten.