



# Arte Tech Pro

Manage Like A Manager

---

Nawang Tendar  
3 New Media Development

---

Bachelor of Graphical And Digital Media  
Applied Science of University of Artevelde

---

CMS Dev  
2019- 2020



I.	<b>Discover</b> .....	<b>4</b>
	Briefing .....	4
II.	<b>Define</b> .....	<b>4</b>
	Analyse .....	4
	Planning .....	5
III.	<b>Design</b> .....	<b>6</b>
	Datamodel .....	6
	Sitemap Admin .....	7
	SITEMAP CLIENT .....	8
	SITEMAP REACT APP (Werknemers/Freelancers) .....	8
	Wireframes CMS .....	9
	WIREFRAMES REACT APP .....	9
	WIREFLOW MOBILE WEB APP .....	11
	Style Guide .....	12
	Visual Design .....	13
IV.	<b>Develop</b> .....	<b>16</b>
	Code Snippets .....	16
	Screenshots .....	19
V.	<b>Deliver</b> .....	<b>21</b>
	Handleiding .....	21
VI.	<b>Deploy</b> .....	<b>22</b>
	Deployment guide .....	22
VII.	<b>Schermopname</b> .....	<b>23</b>
VIII.	<b>Presentatie</b> .....	<b>23</b>
IX.	<b>Timesheet</b> .....	<b>23</b>



## DISCOVER

### BRIEFING

- Onze opdrachtgever wil 2 web platforms waar een(CMS systeem) voor manager en klanten bedoelden en een(Web App) voor werknemers/freelancers. Vanuit de CMS, manager kunt alle prestaties volgen van werknemers/freelancers. Vanuit de Webapp, kunnen de werknemers/freelancers hun prestaties doorsturen naar manager. Manager kunt een periode opstellen waar correspondentie klant moet bevestigen via website(een deel van CMS).
- Onze deadline is 5 januari 2020.

## DEFINE

### ANALYSE

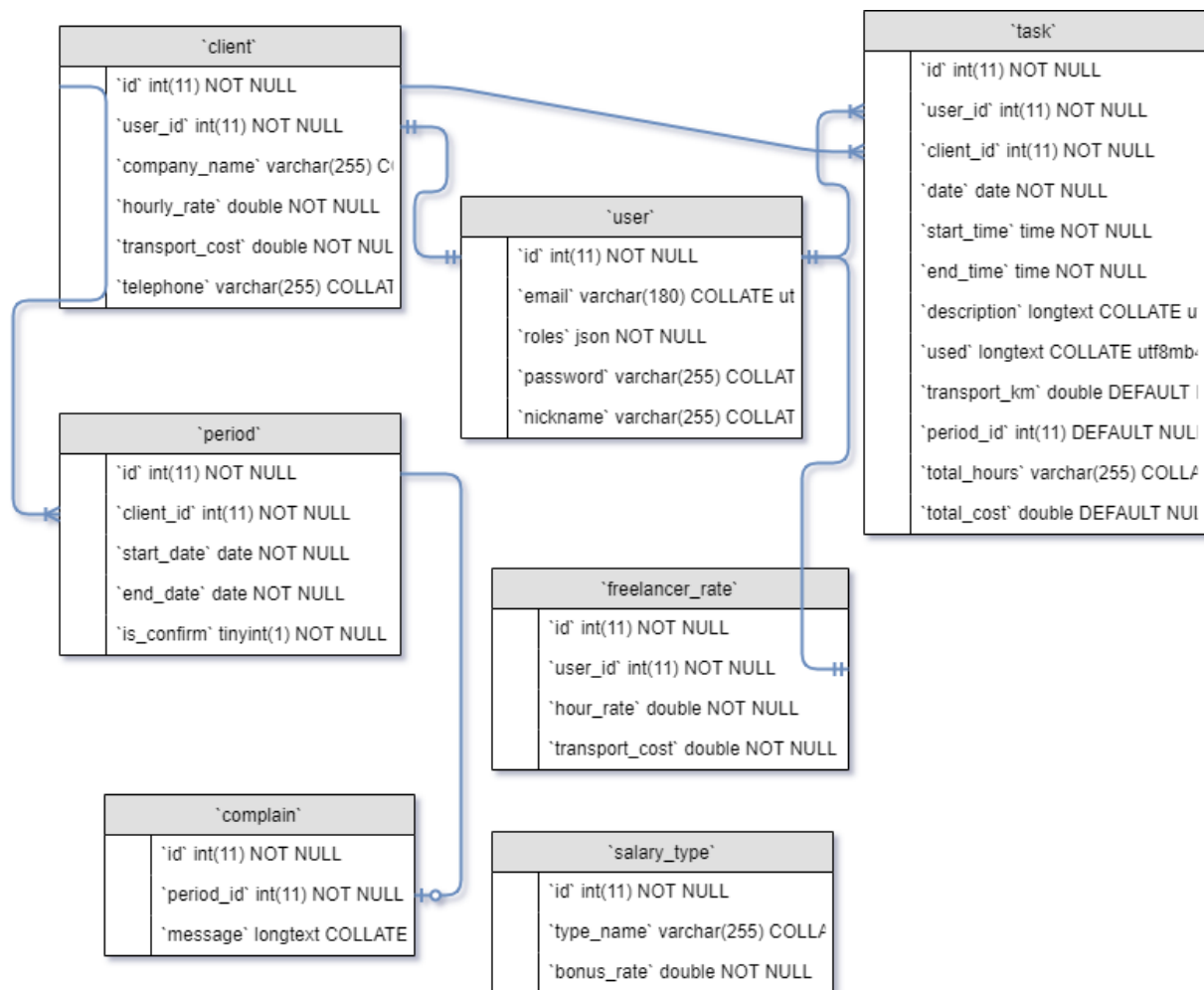
---

### CONCURRENTIE

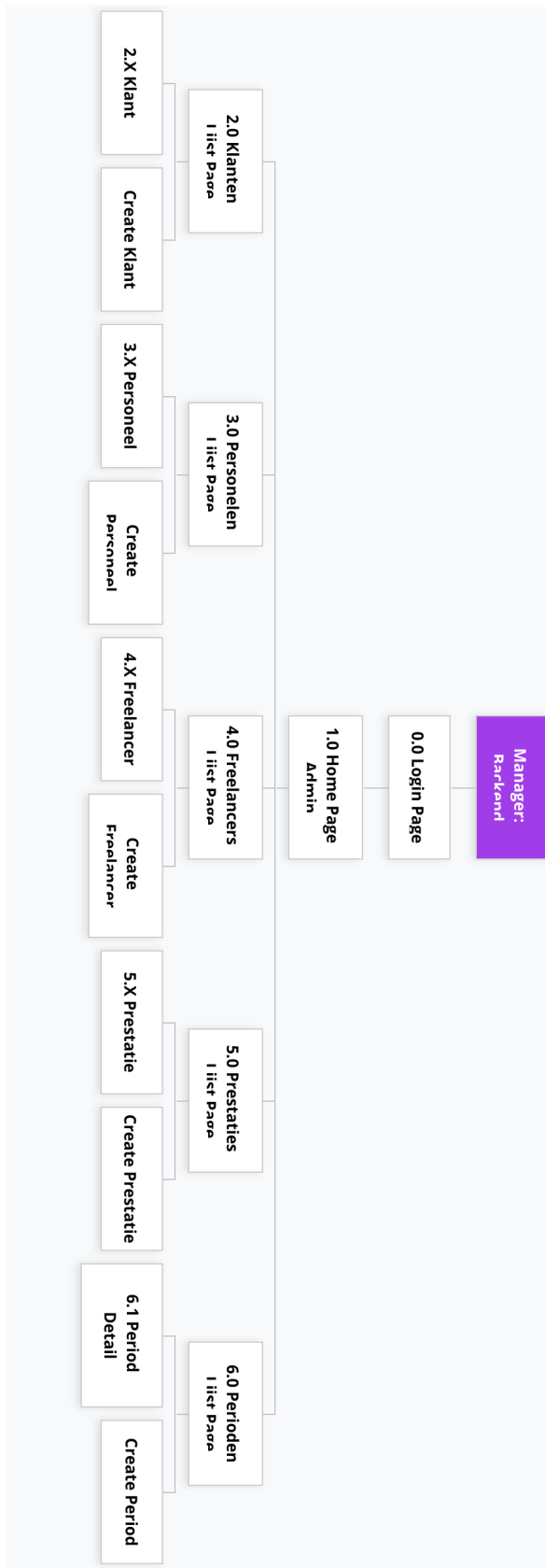
- <https://hubstaff.com/>
- <https://dribbble.com/nawangtendar/buckets/1695498-arte-tech-pro>

## PLANNING

- Persoonlijke Milestones
  - Ideaboard
  - Data model
  - Moodboard
  - Sitemap
  - Wireframes – Wireflow
  - Style Guide
  - UI Design – Adobe XD – Prototype
  - Logo design
  - Authenticatie – Symfony
  - JWT – Symfony
  - Template – Symfony
  - Webpack – Symfony
  - CMS(maken controller, entity, template.twig) – Symfony
  - Init React App
  - SASS – React App
  - UI development
  - API Test
  - Post new tast – React App
  - Auth – React App
  - Task List – React App
  - Rapport – React App
  - Server form validatie
  - Export PDF - Symfony
  - Email – Symfony



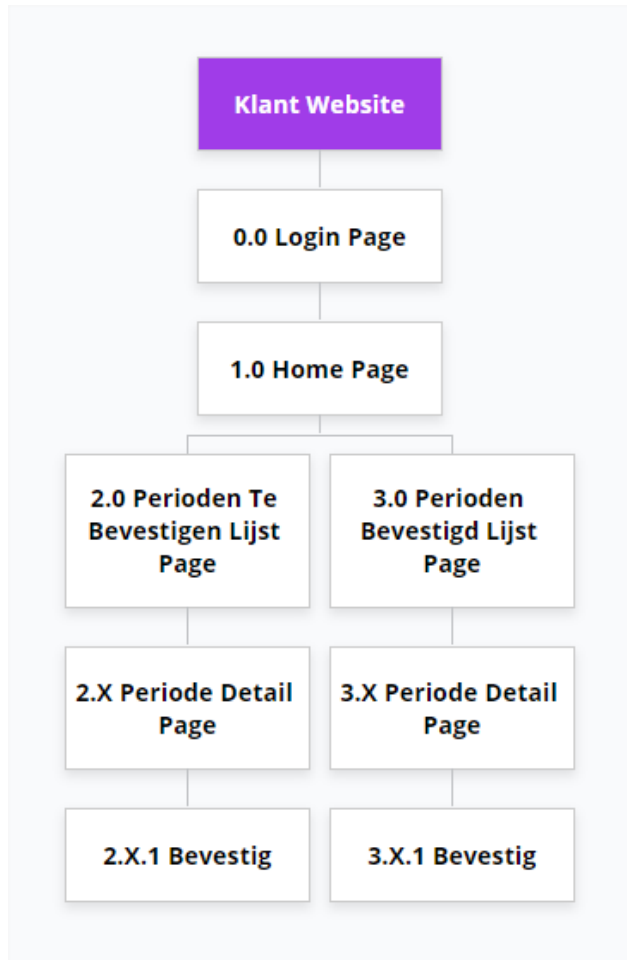
## SITEMAP ADMIN



<https://www.gloomaps.com/CrE7XsjgHz>

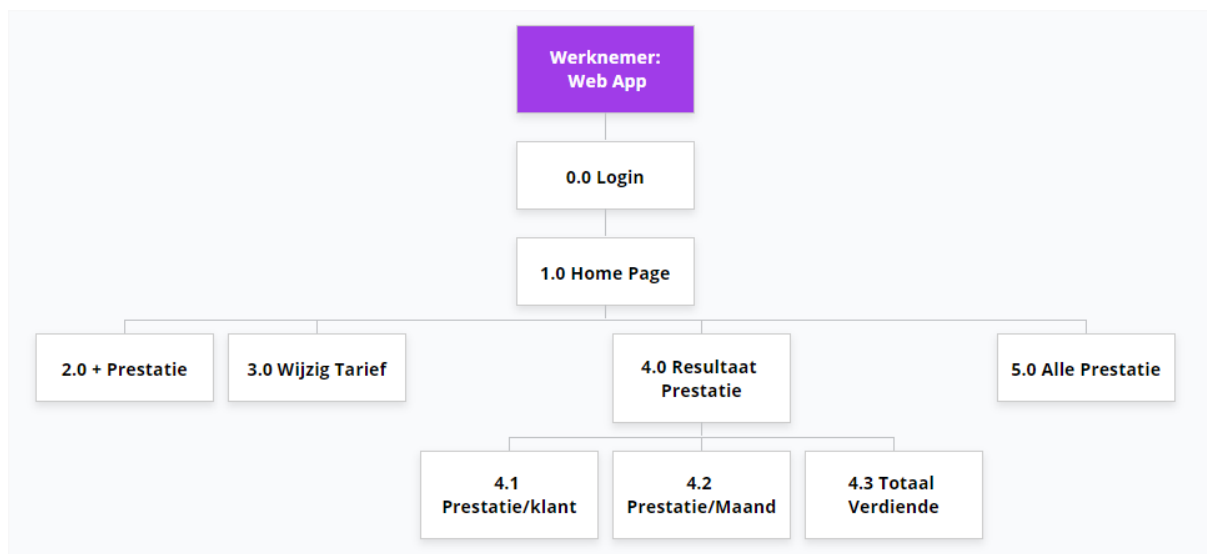


## SITEMAP CLIENT



<https://www.gloomaps.com/d3CmmKd7Mf>

## SITEMAP REACT APP(WERKNEMERS/FREELANCERS)



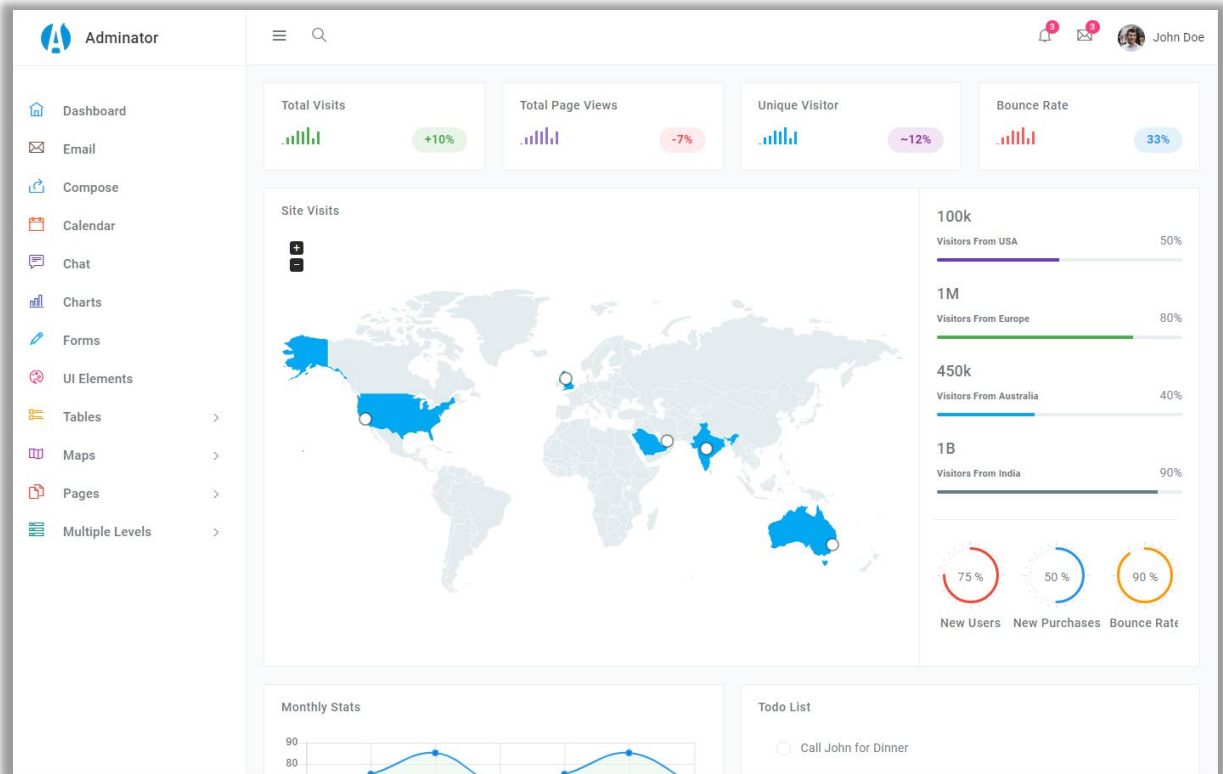
<https://www.gloomaps.com/ZRKG26lsqg>

## WIREFRAMES CMS

Template Based: Adminator

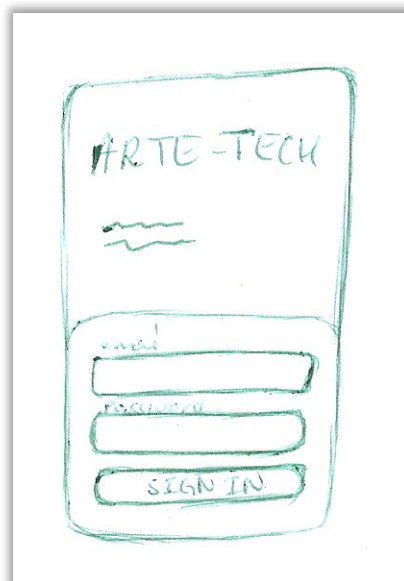
<https://github.com/puikinsh/Adminator-admin-dashboard>

<https://colorlib.com/polygon/adminator/index.html>

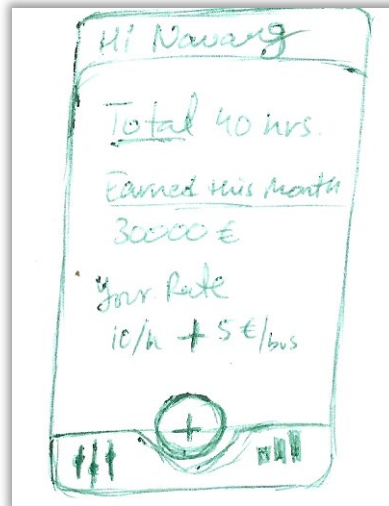


## WIREFRAMES REACT APP

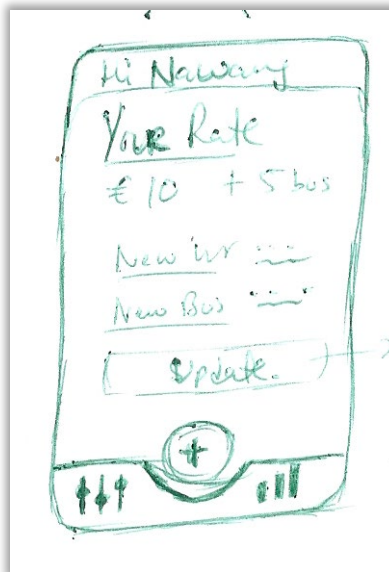
Login



Home Page



Rate Update



Rapport Overzicht





## Kleuren



#56349F



#7E6894



#FF772E



#F3F3F3

## Feather Icons



list



folder



plus



edit



log-out

## Font: Segeo UI



Segeo UI - 18px Bold



Segeo UI - 18px Regular



Segeo UI - 14px Regular

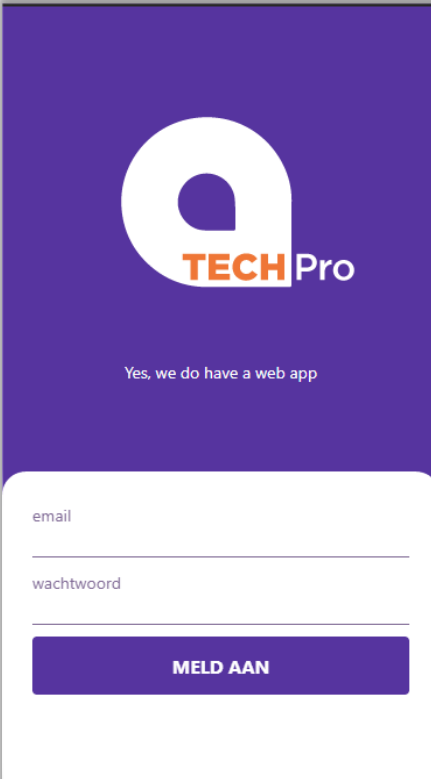


Segeo UI - 28px Semi Bold

## Buttons

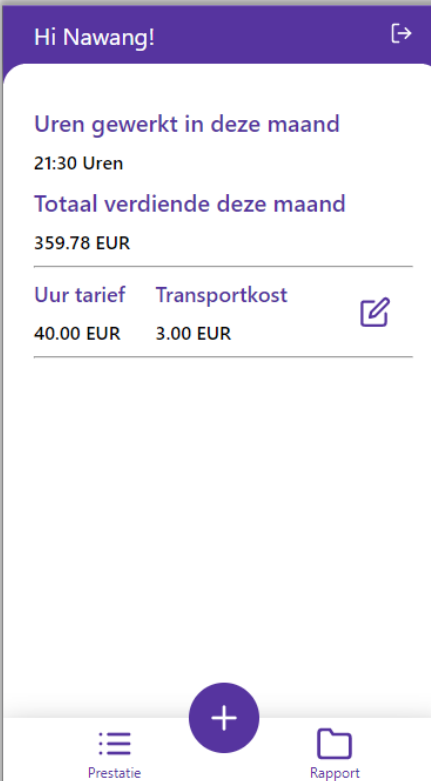


## Login



The login screen features a purple header with the ARTE TECH Pro logo. Below the logo, the text "Yes, we do have a web app" is displayed. The login form consists of two input fields: "email" and "wachtwoord". A purple button labeled "MELD AAN" is positioned below the password field.

## Home pagina



The home page displays a purple header with the greeting "Hi Nawang!" and a logout icon. The main content area shows the following information:


- Uren gewerkt in deze maand**  
21:30 Uren
- Totaal verdiende deze maand**  
359.78 EUR

Below this, there is a table with two columns: "Uur tarief" and "Transportkost".

Uur tarief	Transportkost
40.00 EUR	3.00 EUR

A purple icon of a document with a pencil is located to the right of the table. At the bottom, there is a navigation bar with three icons: a list icon labeled "Prestatie", a plus icon, and a folder icon labeled "Rapport".

## Tarief aanpassen

Hi Nawang! 

Oude uurtarief  
40.00 EUR


Oude transportcost  
3.00 EUR


Nieuwe tarief


Uur tarief

transportkost /km


OPSLAAN

Prestatie




Rapport


## lijst van prestaties


Hi Nawang! 

Alle Prestaties

Datum	Bedrijf	Uren
2020		
14/1	Very Nice Company	10:30
13/1	In The Pocket	11:00
2019		
31/12	Very Nice Company	07:48
31/12	In The Pocket	03:15
30/12	In The Pocket	03:45
30/12	Nawang Production	07:15
29/12	In The Pocket	03:30

Prestatie



Rapport

Totaal rapport van elke maand

Hi Nawang!

Selecteer bedrijf

Alle bedrijven

Selecteer maand

Alle tijden

MAGIC

Resultaat

Indienst uren

21:30 Uren

Inkomst

359.78 EUR

Prestatie

Rapport



## API – Fetch alle eigen prestaties

```
/**
 * @Route("/api/tasksbyuser", name="api_getTasksByUser", methods={"POST"})
 * @param Request $request
 * @return JsonResponse
 * @throws AnnotationException
 * @throws ExceptionInterface
 */
public function getTasksByUser(Request $request)
{
    $userManager = $this->getDoctrine()->getRepository( persistentObject: User::class);
    $taskManager = $this->getDoctrine()->getRepository( persistentObject: Task::class);
    $classMetadataFactory = new ClassMetadataFactory(new AnnotationLoader(new AnnotationReader()));
    $norm = [new DateTimeNormalizer(), new ObjectNormalizer($classMetadataFactory),];
    $encoders = [new JsonEncoder()];
    $serializer = new Serializer($norm, $encoders);

    if ($request->isMethod( method: 'POST')) {
        $postData = json_decode($request->getContent());
        $user = $userManager->findOneBy(['email' => $postData->email]);
        $tasks = $taskManager->findBy(['user' => $user->getId()], ["date" => "DESC"]);
        $tasks = $serializer->normalize($tasks, format: 'json', ['groups' => 'taskInfo']);

        return $this->json($tasks);
    }
}
```

## API – Opslaan nieuwe prestatie

```
/**
 * @Route("/api/savetask", name="api_saveTask", methods={"POST"})
 * @param Request $request
 * @return JsonResponse
 * @throws \Exception
 */
public function saveTask(Request $request)
{
    $em = $this->getDoctrine()->getManager();
    $clientRepo = $this->getDoctrine()->getRepository( persistentObject: Client::class);
    $userRepo = $this->getDoctrine()->getRepository( persistentObject: User::class);
    $newTask = new Task();
    $helper = new Helper();

    if ($request->isMethod( method: "POST")) {

        $postData = json_decode($request->getContent());

        $client = $clientRepo->find($postData->clientId);
        $employeeUser = $userRepo->find($postData->workerId);

        //get DateTime from timeStamp
        $startTime = $helper->getTimeFromTimestamp($postData->startTime);
        $endTime = $helper->getTimeFromTimestamp($postData->endTime);

        //fill in new data for new task
        $newTask->setClient($client);
        $newTask->setUser($employeeUser);
        $newTask->setDate(new DateTime($helper->getDateFromTimestamp($postData->date)));

        $newTask->setStartTime($startTime);
        $newTask->setEndTime($endTime);
        $newTask->setTotalHours($helper->getHoursDifference($startTime, $endTime));

        $newTask->setTotalCost($helper->calculateTaskTotalCost($client->getHourlyRate()
            , $helper->getHoursDifference($startTime, $endTime), $client->getTransportCost(), $postData->km));
        $newTask->setDescription($postData->description);
        $newTask->setUsed($postData->used);
        $newTask->setTransportKM($postData->km);

        $em->persist($newTask);
        $em->flush();
    }

    return $this->json( data: "Saved");
}
```

## React App – Create New Task met API

```
let handleSubmit = async () => {  
  //post new task  
  
  let newTask = {  
    clientId: clientId,  
    workerId: workerId,  
    date: startDate,  
    startTime: startTime,  
    endTime: endTime,  
    description: description,  
    used: used,  
    pauzeMinutes: pauzeMinutes,  
    km: km  
  }  
  
  await axios.post(process.env.REACT_APP_API_URL + "/api/savetask", newTask  
    , { headers: { "Authorization": `Bearer ${token}` } })  
    .then((response) => {  
      if (response.status === 200) {  
        console.log(response.data)  
        NotificationManager.success('Succes', 'Prestatie Toegevoegd', 2300)  
        setTimeout(() => { setTaskSaved(true) }, 2400);  
      }  
    }).catch((error) => {  
      setError("Fout! doe opnieuw")  
      console.log(error)  
    })  
}
```

## Migrations

```

Controller
├── .gitignore
├── ApiController.php
├── ClientController.php
├── ClientFrontendController.php
├── EmployeeController.php
├── ErrorController.php
├── FreelancerController.php
├── PageController.php
├── PeriodController.php
├── SecurityController.php
└── TaskController.php

```

## Entities

```

Entity
├── .gitignore
├── Client.php
├── Complain.php
├── FreelancerRate.php
├── Helper.php
├── Period.php
├── SalaryType.php
├── Task.php
└── User.php

```

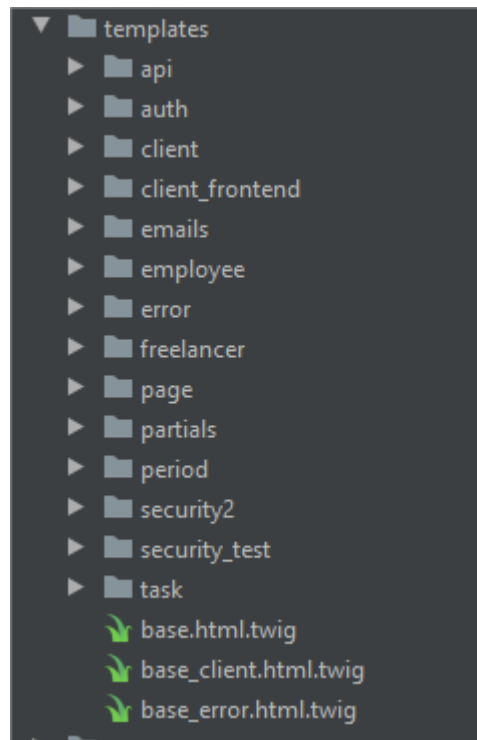
Access Controls  
– CMS

```

access_control:
- { path: ^/login$, roles: IS_AUTHENTICATED_ANONYMOUSLY }
- { path: ^/api/login, roles: IS_AUTHENTICATED_ANONYMOUSLY }
- { path: ^/api, roles: IS_AUTHENTICATED_FULLY }
- { path: ^/create_admin, roles: ROLE_ADMIN }
- { path: ^/clients, roles: ROLE_ADMIN }
- { path: ^/tasks, roles: ROLE_ADMIN }
- { path: ^/freelancers, roles: ROLE_ADMIN }
- { path: ^/employees, roles: ROLE_ADMIN }
- { path: ^/periods, roles: ROLE_ADMIN }
- { path: ^/edit_period[0-9]+, roles: ROLE_ADMIN }
- { path: ^/edit_client[0-9]+, roles: ROLE_ADMIN }
- { path: ^/edit_employee[0-9]+, roles: ROLE_ADMIN }
- { path: ^/edit_task[0-9]+, roles: ROLE_ADMIN }
- { path: ^/create_client, roles: ROLE_ADMIN }
- { path: ^/create_employee, roles: ROLE_ADMIN }
- { path: ^/create_freelancer, roles: ROLE_ADMIN }
- { path: ^/create_period, roles: ROLE_ADMIN }
- { path: ^/create_task, roles: ROLE_ADMIN }
- { path: ^/client_home, roles: ROLE_CLIENT }
- { path: ^/client_confirmed_periods, roles: ROLE_CLIENT }
- { path: ^/client_confirmed_periods, roles: ROLE_CLIENT }
- { path: ^/client_detail_period_[0-9]+, roles: ROLE_CLIENT }

```

## Twig Templates - CMS



---

#### MANAGER - CMS

- Kan nieuwe manager/admin aanmaken
- Kan nieuwe klant, personeel, freelancer, prestatie en periode aanmaken
- Kan alle bovenstaande gegevens wijzigen of verwijderen
- Overzicht van alle prestaties, klanten, personeel, freelancer, periode

---

#### KLANT – CMS

- Heeft overzicht van alle niet & wel bevestigd perioden
- Kan periode bevestigen en opmerking sturen

---

#### PERSONEEL – WEB APP

- Toegang tot alle eigen prestaties
- Krijgt overzicht van totale verdiende en gewerkte prestatie
- Kan nieuwe prestatie aanmaken

---

#### FREELANCER – WEB APP

- Toegang tot alle eigen prestaties
- Krijgt overzicht van totale verdiende en gewerkte prestatie
- Kan nieuwe prestatie aanmaken
- Kan hun uurtarief/transportkost aanpassen

## DEPLOY

### DEPLOYMENT GUIDE

1. Maak een folder op uw toestel
2. Navigeer naar uw reeds gemaakte folder via terminal en voer 'git clone [https://github.com/nawatend/arte\\_tech\\_pro.git](https://github.com/nawatend/arte_tech_pro.git)' uit

#### CMS

1. Vanaf zelfde terminal voer: 'cd cms'
2. Voer 'yarn install'
3. Voer 'composer install'
4. Pas aan .env file

```
DATABASE_URL=mysql://USERNAME:PASSWORD@127.0.0.1:PORT/arte_tech_pro?serverVersion=5.7
```

De rest van gegevens zit al in voor test.  
**SQL file** zit in /docs.

5. In terminal voer 'symfony serve' uit

#### App

1. Van zelfde terminal: 'cd app'
2. Voer 'yarn install'
3. Maak .env file en voeg

```
REACT_APP_API_URL= 'YOUR IP/PORT OF SYMFONY SERVER'
```

Voorbeeld: 'http://127.0.0.1:8000'

4. In terminal voer 'yarn start' uit en enjoy

## SCHERMOPNAME

- Links:
  - Worker\_app:  
[https://github.com/nawatend/arte\\_tech\\_pro/blob/master/docs/screen%20video/app\\_worker.mp4](https://github.com/nawatend/arte_tech_pro/blob/master/docs/screen%20video/app_worker.mp4)
  - Admin\_cms:  
[https://github.com/nawatend/arte\\_tech\\_pro/blob/master/docs/screen%20video/app\\_worker.mp4](https://github.com/nawatend/arte_tech_pro/blob/master/docs/screen%20video/app_worker.mp4)
  - Client\_cms:  
[https://github.com/nawatend/arte\\_tech\\_pro/blob/master/docs/screen%20video/cms\\_client.mp4](https://github.com/nawatend/arte_tech_pro/blob/master/docs/screen%20video/cms_client.mp4)

## PRESENTATIE

Link: [https://github.com/nawatend/arte\\_tech\\_pro/blob/master/docs/presentatie.pptx](https://github.com/nawatend/arte_tech_pro/blob/master/docs/presentatie.pptx)

## TIMESHEET

Datum	Start	eind	totaal uren			
18-11-19	20:00:00	20:50:00	0:50:00			
25-11-19	16:00:00	18:30:00	2:30:00			
26-11-19	9:00	12:01:00	3:00:00			
30-11-19	16:30:00	22:40:00	6:00:00			
12-01-19	17:00:00	20:00:00	3:00:00			
12-03-19	9:00:00	12:01:00	3:01:00			
12-03-19	20:00:00	22:00:00	2:00:00			
12-04-19	10:00:00	12:00:00	2:00:00			
12-07-19	11:30:00	23:30	12:00:00			
12-08-19	21:00:00	23:00:00	2:00:00			
12-09-19	20:00	23:59:59	3:59:59			
12-10-19	7:05	7:40	0:35:00			
12-10-19	9:00:00	13:00:00	4:00:00		<b>Totaal uren</b>	<b>169:05:59</b>
12-17-19	9:00:00	12:00:00	3:00:00		echte totaal uren	845:29:55
12-17-19	14:30:00	17:00:00	2:30:00			
12-18-19	8:00:00	8:30:00	0:30:00			
12-21-19	15:00	21:00	6:00:00			
12-22-19	15:30	20:00	4:30:00			
12-23-19	16:00	21:00	5:00:00			
12-24-19	12:30	23:00	10:30:00			
12-25-19	12:00	24:00:00	12:00:00			
12-26-19	0:00	0:00	0:00:00	PPD		
12-27-19	11:30	23:50	12:20:00			
12-28-19	11:00	23:30	12:30:00			
12-29-19	11:40	23:00	11:20:00			
12-30-19	10:30	22:00	11:30:00			
12-31-19	13:00	18:30	5:30:00			
01-01-20	15:00	21:00	6:00:00			
01-02-20	10:30	22:30:00	12:00:00			
01-03-20	10:30	19:30	9:00:00			