Arte-Tech

Wouter Vlaeyen

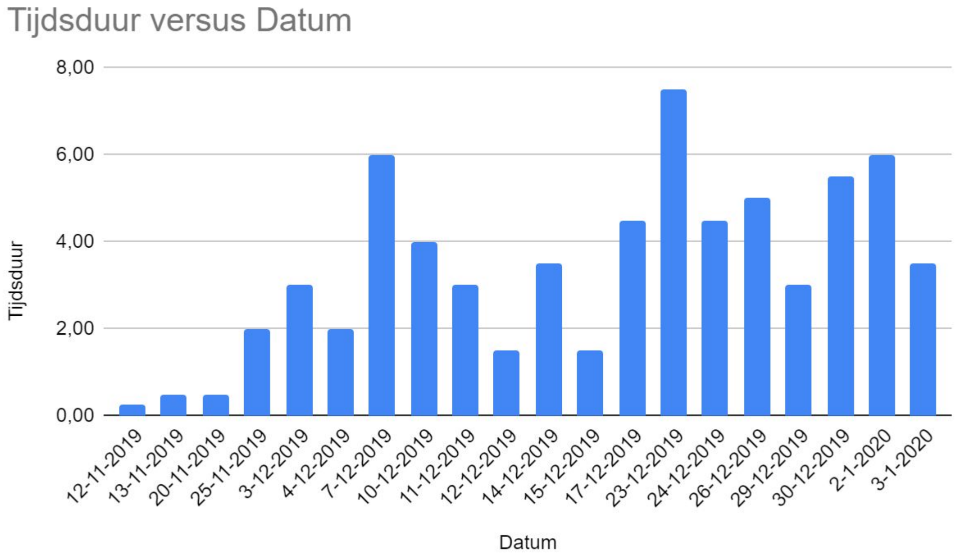
Inhoud

- Briefing
- Define
  - Planning & Timesheet
- Design
  - Sitemap
  - Wireframes & Wireflow
  - Visual Design
- Develop
  - Code snippets
- Demo

Briefing

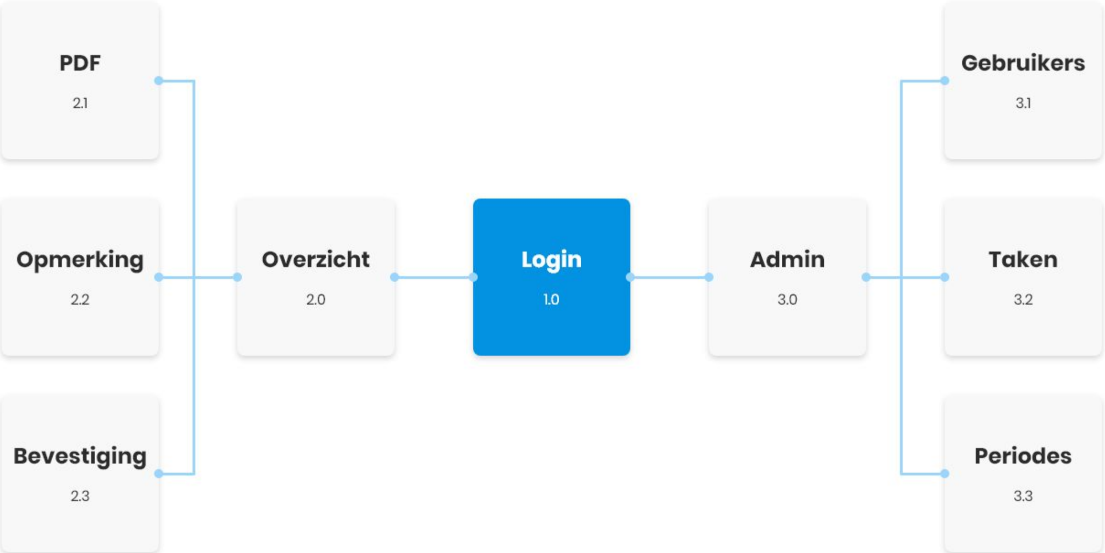
- CMS
- Web-app → Werknemer
- Website → Klant
- Symfony
- Vue.js

Planning & Timesheet

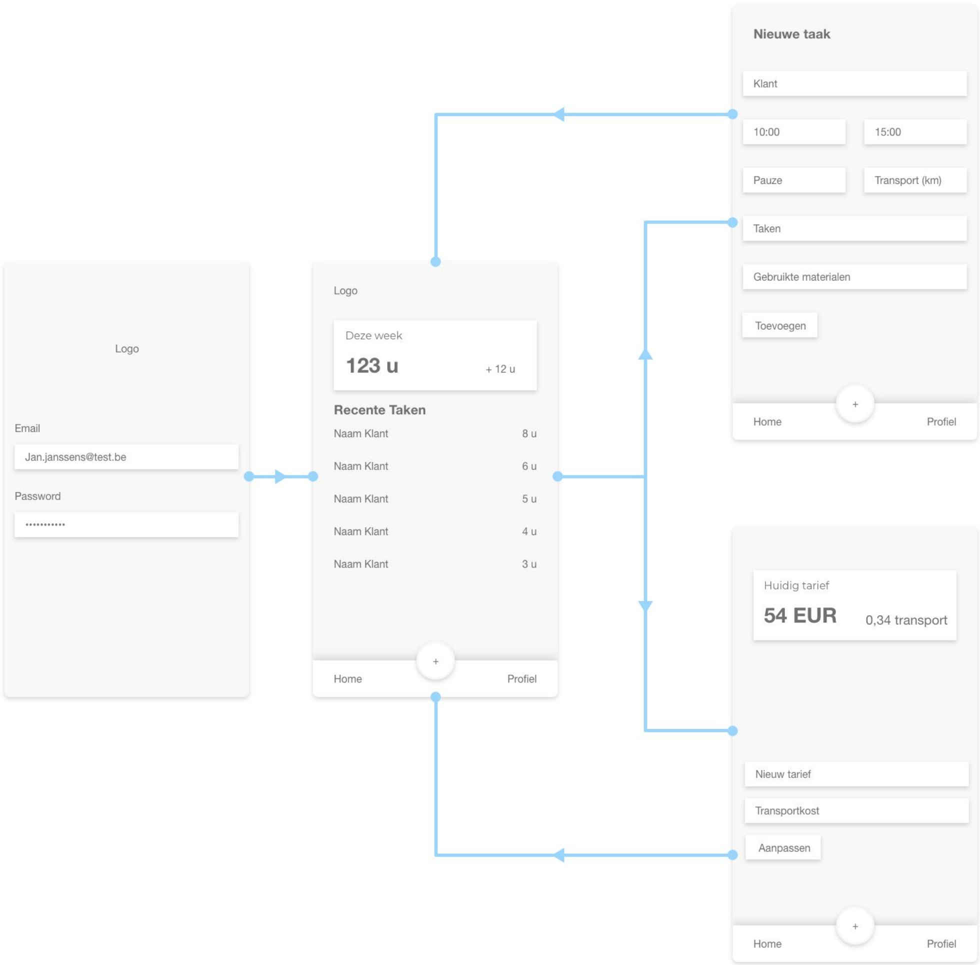


Taak	Planning										
	NOV					DEC					JAN
		4	11	18	25	2	9	16	23	30	6
Planning					🟦						
Wireframes				🟦	🟦						
Wireflow					🟦						
Visual Design					🟦	🟦					
Development CMS						🟨	🟨				
Development Web							🟨	🟨			
Development App								🟨	🟨		
Testing						🟩	🟩	🟩	🟩	🟩	
Schrijven dossier		🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	

Sitemap


PDF
2.1
Opmerking
2.2
Bevestiging
2.3
Overzicht
2.0
Login
1.0
Admin
3.0
Gebruikers
3.1
Taken
3.2
Periodes
3.3

Wireflow



Logo

Email

Jan.janssens@test.be

Password

•••••••••••

Logo

Deze week

123 u

+ 12 u

Recente Taken

Naam Klant	8 u
Naam Klant	6 u
Naam Klant	5 u
Naam Klant	4 u
Naam Klant	3 u

+

Home

Profiel

Nieuwe taak

Klant

10:00

15:00

Pauze

Transport (km)

Taken

Gebruikte materialen

Toevoegen

+

Home

Profiel

Huidig tarief

54 EUR

0,34 transport

Nieuw tarief

Transportkost

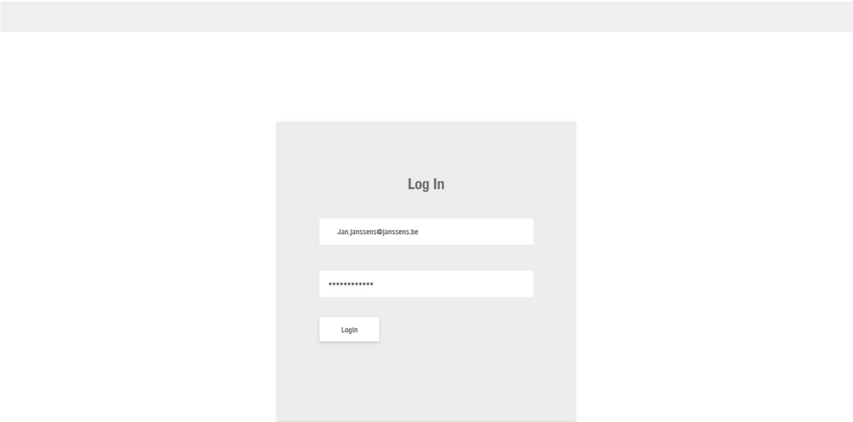
Aanpassen

+

Home

Profiel

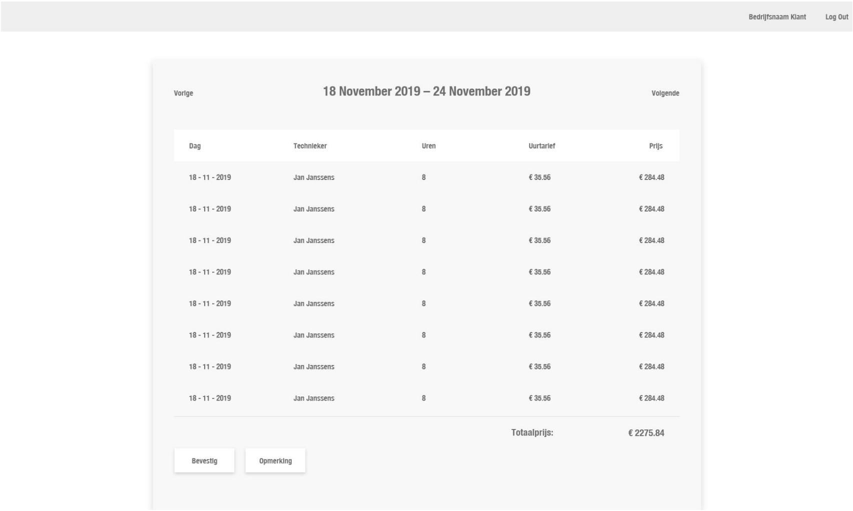
Wireframes



Log In

Jan.janssens@janssens.be

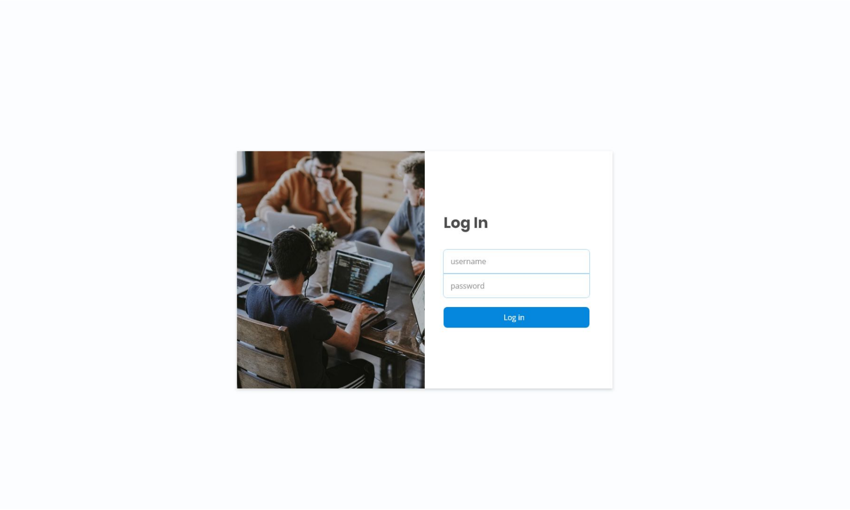
••••••••••••

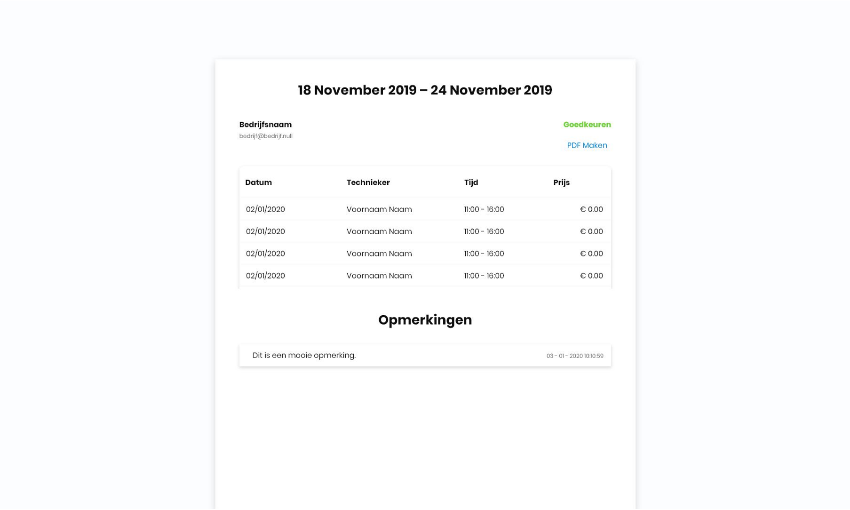


18 November 2019 – 24 November 2019

Dag	Technieker	Uren	Uurtarief	Prijs
18 - 11 - 2019	Jan Janssens	8	€ 35.56	€ 284.48
18 - 11 - 2019	Jan Janssens	8	€ 35.56	€ 284.48
18 - 11 - 2019	Jan Janssens	8	€ 35.56	€ 284.48
18 - 11 - 2019	Jan Janssens	8	€ 35.56	€ 284.48
18 - 11 - 2019	Jan Janssens	8	€ 35.56	€ 284.48
18 - 11 - 2019	Jan Janssens	8	€ 35.56	€ 284.48
18 - 11 - 2019	Jan Janssens	8	€ 35.56	€ 284.48
18 - 11 - 2019	Jan Janssens	8	€ 35.56	€ 284.48
			Totaalprijs:	€ 2275.84

Visual Design





18 November 2019 – 24 November 2019

Bedrijfsnaam
bedrijf@bedrijf.null

Goedkeuren

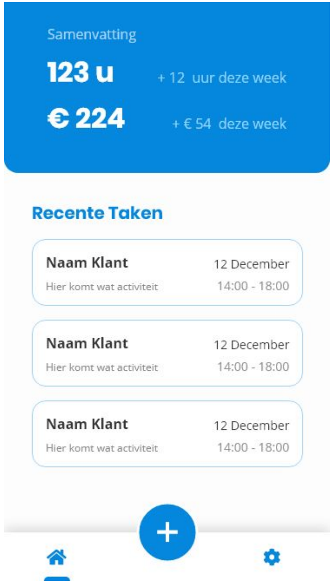
PDF Maken

Datum	Technieker	Tijd	Prijs
02/01/2020	Voornaam Naam	11:00 - 16:00	€ 0.00
02/01/2020	Voornaam Naam	11:00 - 16:00	€ 0.00
02/01/2020	Voornaam Naam	11:00 - 16:00	€ 0.00
02/01/2020	Voornaam Naam	11:00 - 16:00	€ 0.00

Opmerkingen

Dit is een mooie opmerking. 03 - 01 - 2020 10:10:59

Visual Design



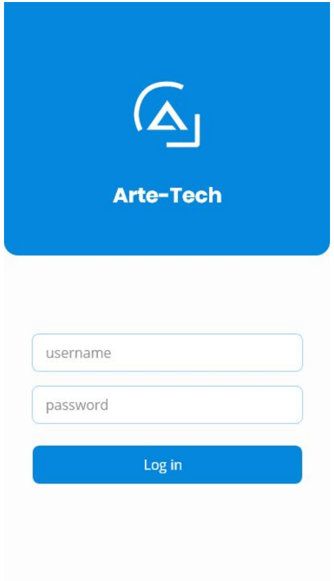
Samenvatting

123 u + 12 uur deze week

€ 224 + € 54 deze week

Recente Taken

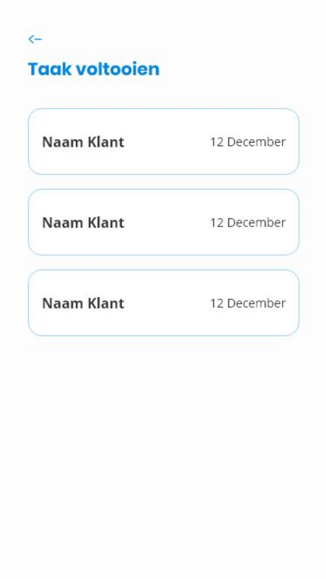
Naam Klant	12 December
Hier komt wat activiteit	14:00 - 18:00
Naam Klant	12 December
Hier komt wat activiteit	14:00 - 18:00
Naam Klant	12 December
Hier komt wat activiteit	14:00 - 18:00



Arte-Tech

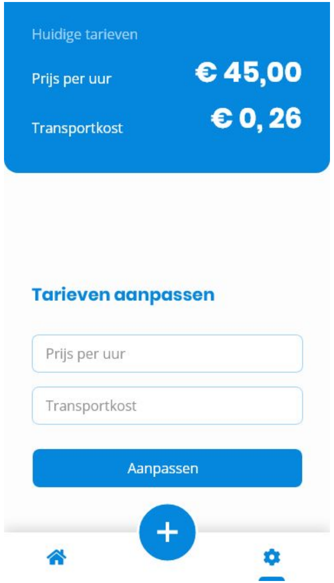
username

password



Taak voltooien

Naam Klant	12 December
Naam Klant	12 December
Naam Klant	12 December



Huidige tarieven

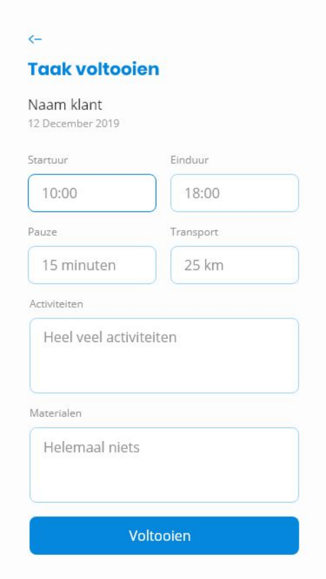
Prijs per uur € 45,00

Transportkost € 0, 26

Tarieven aanpassen

Prijs per uur

Transportkost



Taak voltooien

Naam klant

12 December 2019

Startuur: 10:00

Einduur: 18:00

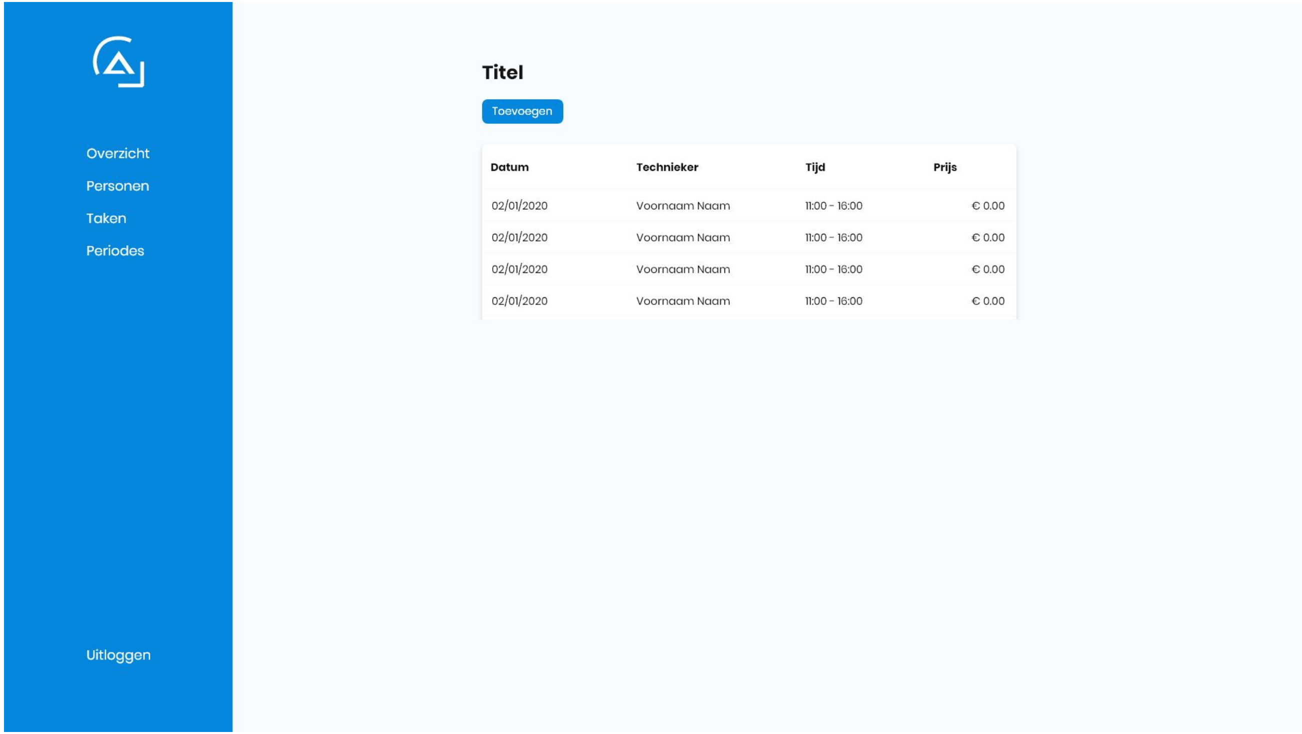
Pauze: 15 minuten

Transport: 25 km

Activiteiten: Heel veel activiteiten

Materialen: Helemaal niets

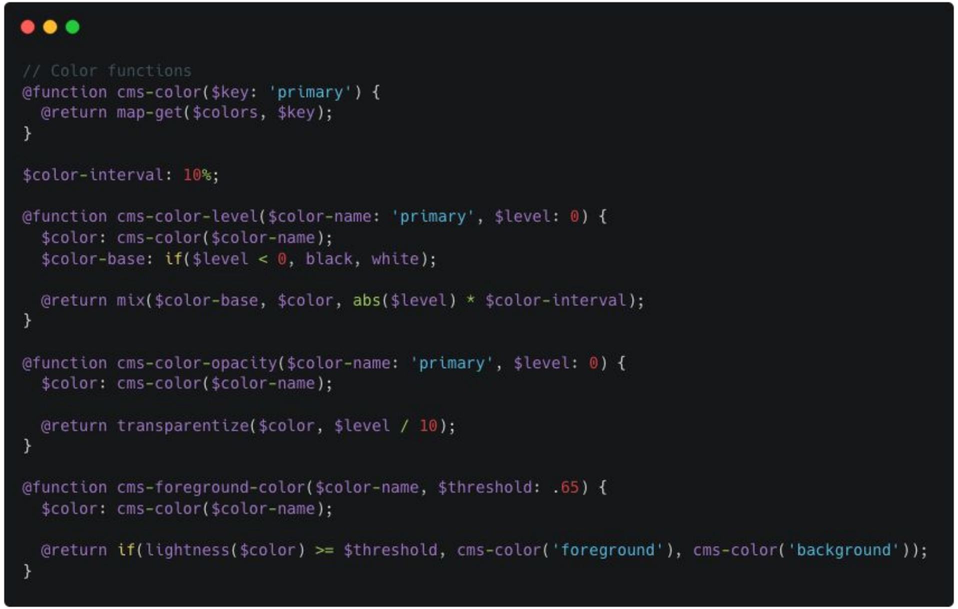
Visual Design



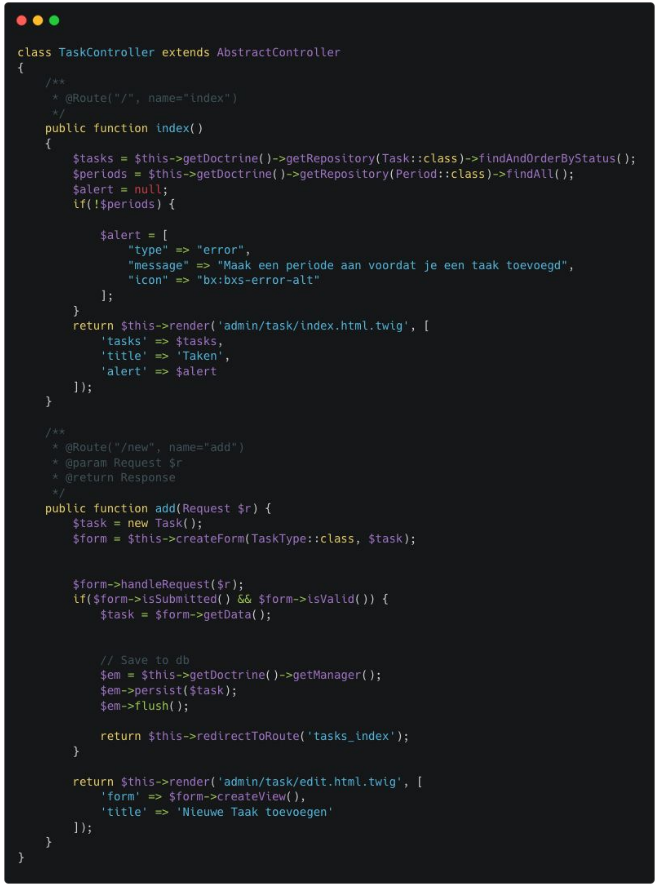
Titel

Datum	Technieker	Tijd	Prijs
02/01/2020	Voornaam Naam	11:00 - 16:00	€ 0.00
02/01/2020	Voornaam Naam	11:00 - 16:00	€ 0.00
02/01/2020	Voornaam Naam	11:00 - 16:00	€ 0.00
02/01/2020	Voornaam Naam	11:00 - 16:00	€ 0.00

Code Snippets

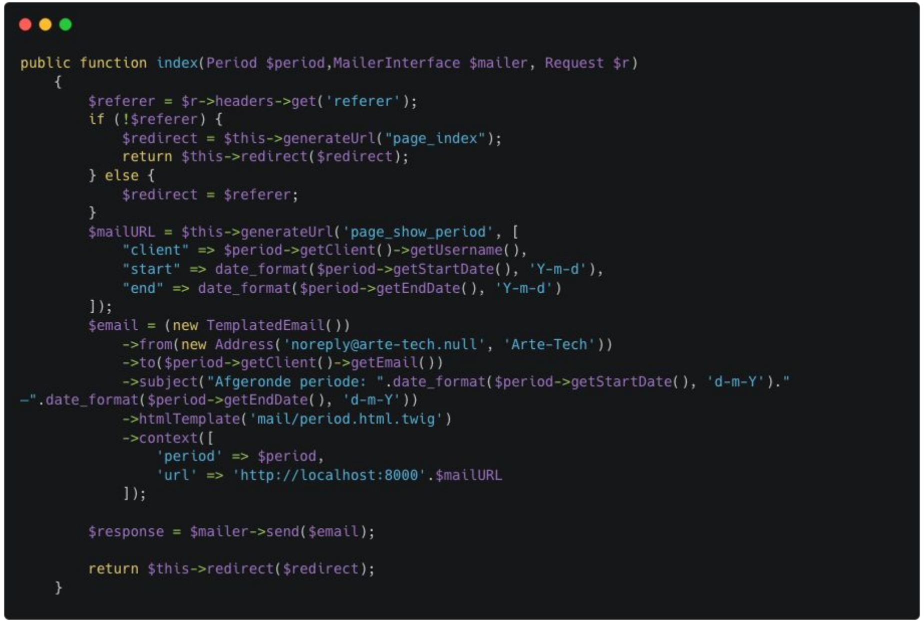


```scss
// Color functions
@function cms-color($key: 'primary') {
  @return map-get($colors, $key);
}

$color-interval: 10%;

@function cms-color-level($color-name: 'primary', $level: 0) {
  $color: cms-color($color-name);
  $color-base: if($level < 0, black, white);

  @return mix($color-base, $color, abs($level) * $color-interval);
}

@function cms-color-opacity($color-name: 'primary', $level: 0) {
  $color: cms-color($color-name);

  @return transparentize($color, $level / 10);
}

@function cms-foreground-color($color-name, $threshold: .65) {
  $color: cms-color($color-name);

  @return if(lightness($color) >= $threshold, cms-color('foreground'), cms-color('background'));
}
```

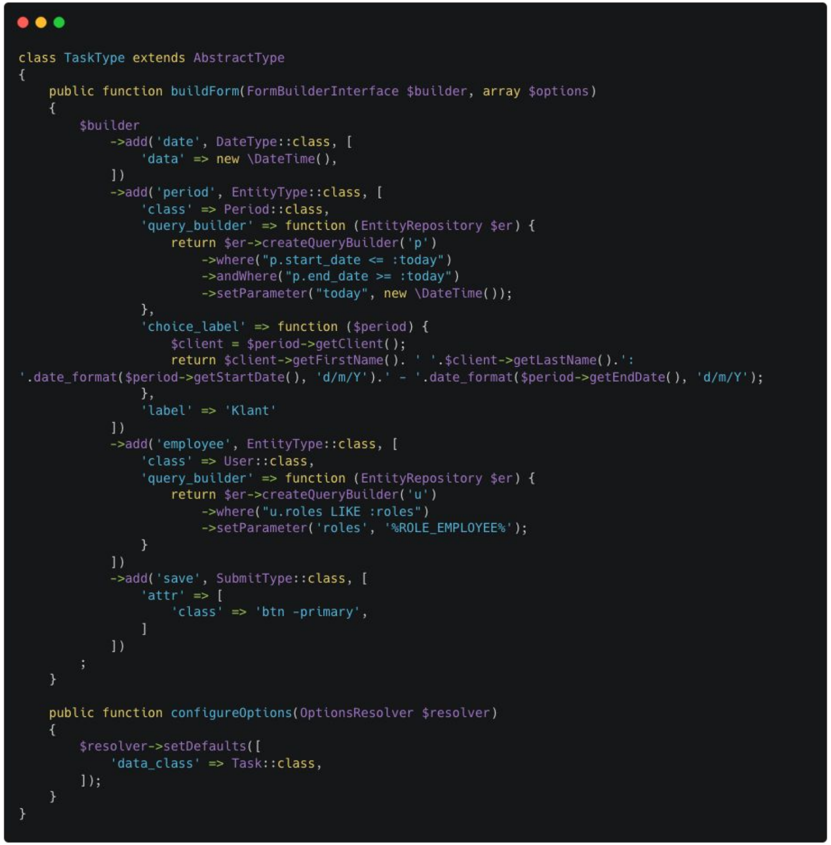


```php
class TaskController extends AbstractController
{
    /**
     * @Route("/", name="index")
     */
    public function index()
    {
        $tasks = $this->getDoctrine()->getRepository(Task::class)->findAndOrderByStatus();
        $periods = $this->getDoctrine()->getRepository(Period::class)->findAll();
        $alert = null;
        if(!$periods) {

            $alert = [
                "type" => "error",
                "message" => "Maak een periode aan voordat je een taak toevoegd",
                "icon" => "bx:bxs-error-alt"
            ];
        }
        return $this->render('admin/task/index.html.twig', [
            'tasks' => $tasks,
            'title' => 'Taken',
            'alert' => $alert
        ]);
    }

    /**
     * @Route("/new", name="add")
     * @param Request $r
     * @return Response
     */
    public function add(Request $r) {
        $task = new Task();
        $form = $this->createForm(TaskType::class, $task);


        $form->handleRequest($r);
        if($form->isSubmitted() && $form->isValid()) {
            $task = $form->getData();


            // Save to db
            $em = $this->getDoctrine()->getManager();
            $em->persist($task);
            $em->flush();

            return $this->redirectToRoute('tasks_index');
        }

        return $this->render('admin/task/edit.html.twig', [
            'form' => $form->createView(),
            'title' => 'Nieuwe Taak toevoegen'
        ]);
    }
}
```

Code Snippets

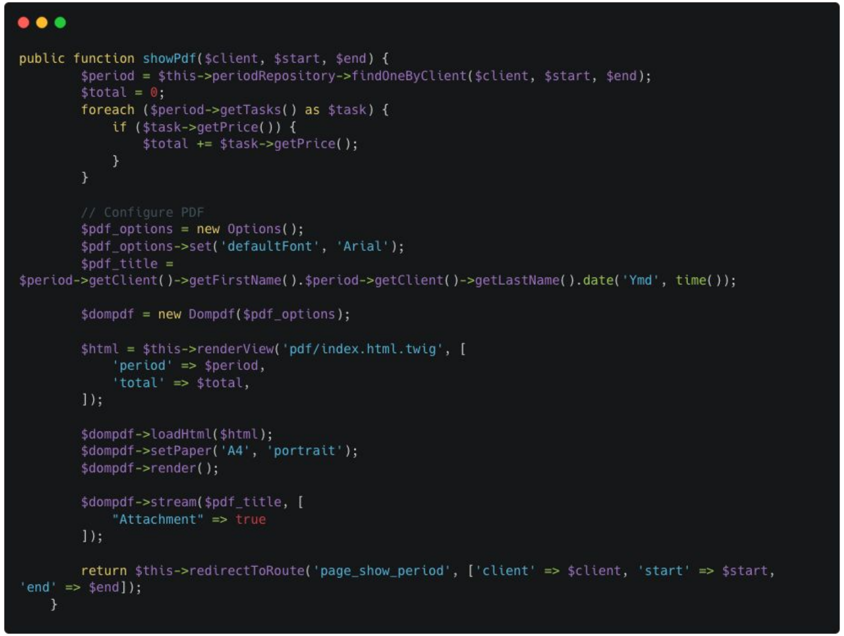


```php
public function index(Period $period,MailerInterface $mailer, Request $r)
    {
        $referer = $r->headers->get('referer');
        if (!$referer) {
            $redirect = $this->generateUrl("page_index");
            return $this->redirect($redirect);
        } else {
            $redirect = $referer;
        }
        $mailURL = $this->generateUrl('page_show_period', [
            "client" => $period->getClient()->getUsername(),
            "start" => date_format($period->getStartDate(), 'Y-m-d'),
            "end" => date_format($period->getEndDate(), 'Y-m-d')
        ]);
        $email = (new TemplatedEmail())
            ->from(new Address('noreply@arte-tech.null', 'Arte-Tech'))
            ->to($period->getClient()->getEmail())
            ->subject("Afgeronde periode: ".date_format($period->getStartDate(), 'd-m-Y')."
-".date_format($period->getEndDate(), 'd-m-Y'))
            ->htmlTemplate('mail/period.html.twig')
            ->context([
                'period' => $period,
                'url' => 'http://localhost:8000'.$mailURL
            ]);

        $response = $mailer->send($email);

        return $this->redirect($redirect);
    }
```

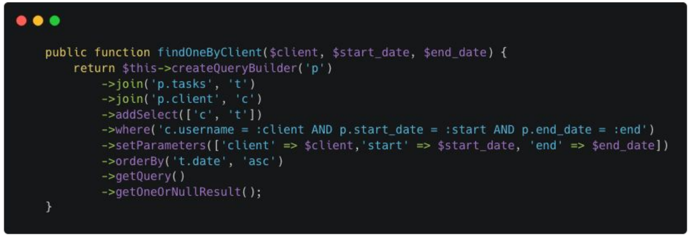


```php
class TaskType extends AbstractType
{
    public function buildForm(FormBuilderInterface $builder, array $options)
    {
        $builder
            ->add('date', DateType::class, [
                'data' => new \DateTime(),
            ])
            ->add('period', EntityType::class, [
                'class' => Period::class,
                'query_builder' => function (EntityRepository $er) {
                    return $er->createQueryBuilder('p')
                        ->where("p.start_date <= :today")
                        ->andWhere("p.end_date >= :today")
                        ->setParameter("today", new \DateTime());
                },
                'choice_label' => function ($period) {
                    $client = $period->getClient();
                    return $client->getFirstName(). ' '.$client->getLastName().':
'.date_format($period->getStartDate(), 'd/m/Y').' - '.date_format($period->getEndDate(), 'd/m/Y');
                },
                'label' => 'Klant'
            ])
            ->add('employee', EntityType::class, [
                'class' => User::class,
                'query_builder' => function (EntityRepository $er) {
                    return $er->createQueryBuilder('u')
                        ->where("u.roles LIKE :roles")
                        ->setParameter('roles', '%ROLE_EMPLOYEE%');
                }
            ])
            ->add('save', SubmitType::class, [
                'attr' => [
                    'class' => 'btn -primary',
                ]
            ])
        ;
    }

    public function configureOptions(OptionsResolver $resolver)
    {
        $resolver->setDefaults([
            'data_class' => Task::class,
        ]);
    }
}
```

Code Snippets

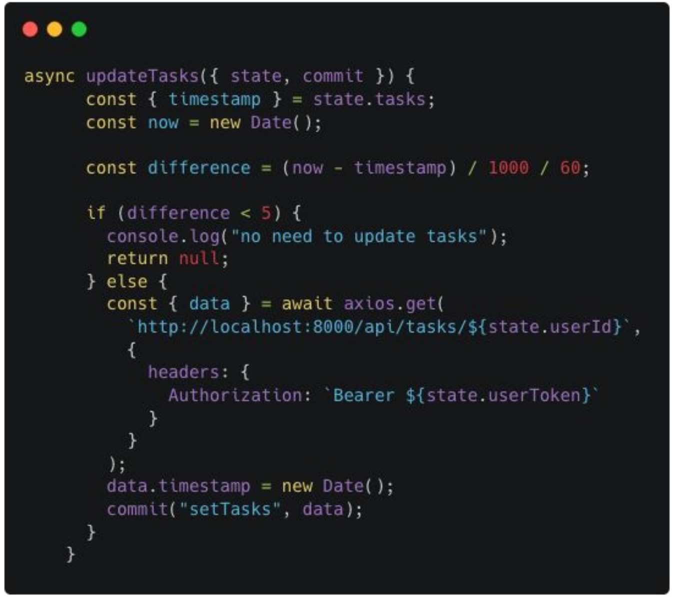


```php
public function showPdf($client, $start, $end) {
        $period = $this->periodRepository->findOneByClient($client, $start, $end);
        $total = 0;
        foreach ($period->getTasks() as $task) {
            if ($task->getPrice()) {
                $total += $task->getPrice();
            }
        }

        // Configure PDF
        $pdf_options = new Options();
        $pdf_options->set('defaultFont', 'Arial');
        $pdf_title =
$period->getClient()->getFirstName().$period->getClient()->getLastName().date('Ymd', time());

        $dompdf = new Dompdf($pdf_options);

        $html = $this->renderView('pdf/index.html.twig', [
            'period' => $period,
            'total' => $total,
        ]);

        $dompdf->loadHtml($html);
        $dompdf->setPaper('A4', 'portrait');
        $dompdf->render();

        $dompdf->stream($pdf_title, [
            "Attachment" => true
        ]);

        return $this->redirectToRoute('page_show_period', ['client' => $client, 'start' => $start,
'end' => $end]);
    }
```

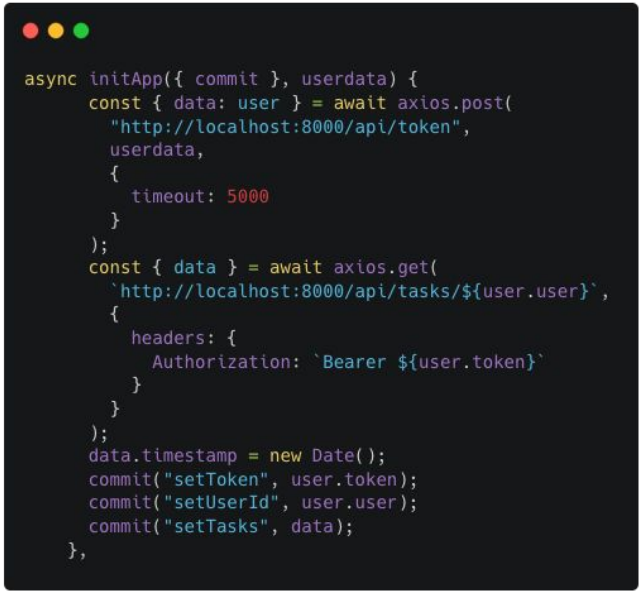


```php
public function findOneByClient($client, $start_date, $end_date) {
    return $this->createQueryBuilder('p')
        ->join('p.tasks', 't')
        ->join('p.client', 'c')
        ->addSelect(['c', 't'])
        ->where('c.username = :client AND p.start_date = :start AND p.end_date = :end')
        ->setParameters(['client' => $client,'start' => $start_date, 'end' => $end_date])
        ->orderBy('t.date', 'asc')
        ->getQuery()
        ->getOneOrNullResult();
}
```

Code Snippets

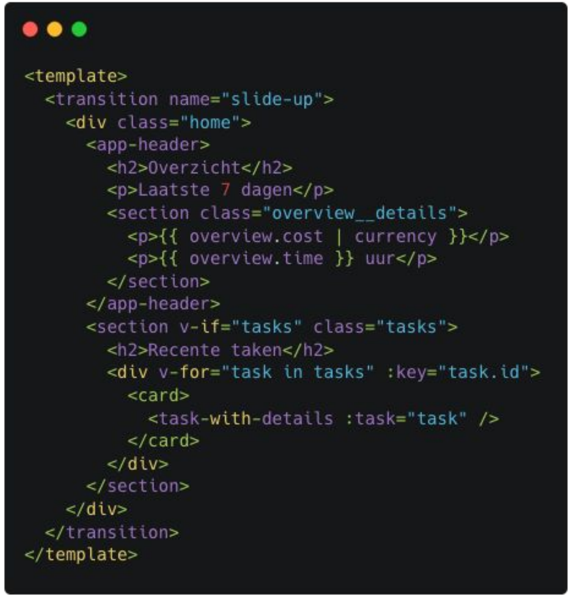


```
async updateTasks({ state, commit }) {
      const { timestamp } = state.tasks;
      const now = new Date();

      const difference = (now - timestamp) / 1000 / 60;

      if (difference < 5) {
        console.log("no need to update tasks");
        return null;
      } else {
        const { data } = await axios.get(
          `http://localhost:8000/api/tasks/${state.userId}`,
          {
            headers: {
              Authorization: `Bearer ${state.userToken}`
            }
          }
        );
        data.timestamp = new Date();
        commit("setTasks", data);
      }
    }
```

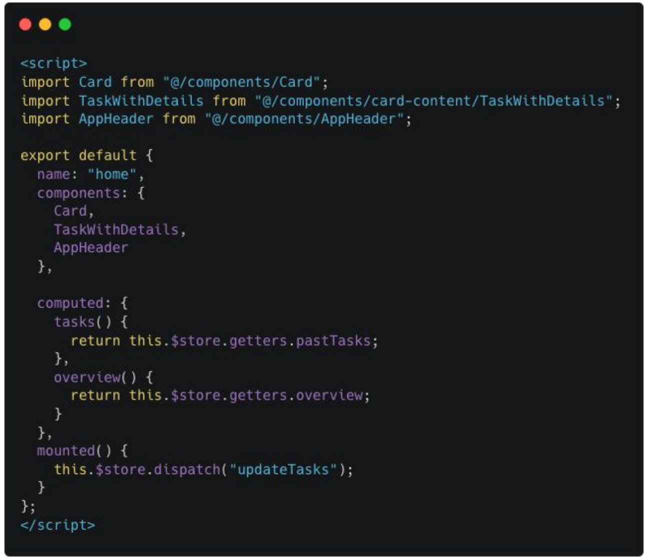


```
async initApp({ commit }, userdata) {
      const { data: user } = await axios.post(
        "http://localhost:8000/api/token",
        userdata,
        {
          timeout: 5000
        }
      );
      const { data } = await axios.get(
        `http://localhost:8000/api/tasks/${user.user}`,
        {
          headers: {
            Authorization: `Bearer ${user.token}`
          }
        }
      );
      data.timestamp = new Date();
      commit("setToken", user.token);
      commit("setUserId", user.user);
      commit("setTasks", data);
    },
```

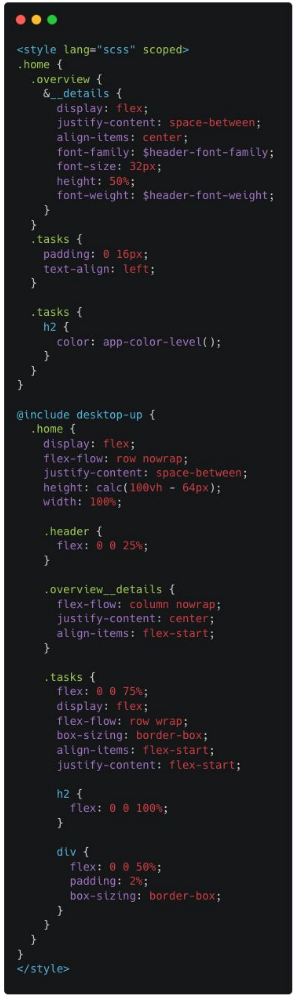
Code Snippets



```
<template>
  <transition name="slide-up">
    <div class="home">
      <app-header>
        <h2>Overzicht</h2>
        <p>Laatste 7 dagen</p>
        <section class="overview__details">
          <p>{{ overview.cost | currency }}</p>
          <p>{{ overview.time }} uur</p>
        </section>
      </app-header>
      <section v-if="tasks" class="tasks">
        <h2>Recente taken</h2>
        <div v-for="task in tasks" :key="task.id">
          <card>
            <task-with-details :task="task" />
          </card>
        </div>
      </section>
    </div>
  </transition>
</template>
```



```
<script>
import Card from "@/components/Card";
import TaskWithDetails from "@/components/card-content/TaskWithDetails";
import AppHeader from "@/components/AppHeader";

export default {
  name: "home",
  components: {
    Card,
    TaskWithDetails,
    AppHeader
  },

  computed: {
    tasks() {
      return this.$store.getters.pastTasks;
    },
    overview() {
      return this.$store.getters.overview;
    }
  },
  mounted() {
    this.$store.dispatch("updateTasks");
  }
};
</script>
```



```
<style lang="scss" scoped>
.home {
  .overview {
    &__details {
      display: flex;
      justify-content: space-between;
      align-items: center;
      font-family: $header-font-family;
      font-size: 32px;
      height: 50%;
      font-weight: $header-font-weight;
    }
  }
  .tasks {
    padding: 0 16px;
    text-align: left;
  }

  .tasks {
    h2 {
      color: app-color-level();
    }
  }
}

@include desktop-up {
  .home {
    display: flex;
    flex-flow: row nowrap;
    justify-content: space-between;
    height: calc(100vh - 64px);
    width: 100%;

    .header {
      flex: 0 0 25%;
    }

    .overview__details {
      flex-flow: column nowrap;
      justify-content: center;
      align-items: flex-start;
    }

    .tasks {
      flex: 0 0 75%;
      display: flex;
      flex-flow: row wrap;
      box-sizing: border-box;
      align-items: flex-start;
      justify-content: flex-start;

      h2 {
        flex: 0 0 100%;
      }

      div {
        flex: 0 0 50%;
        padding: 2%;
        box-sizing: border-box;
      }
    }
  }
}
</style>
```

Demo