

# MobDev II

## Gdm Gent - Arteveldehogeschool

2018 - 2019

---

### **DOOR**

Wouter Vlaeyen, Owen De Waele, Bruno De Coene

### **DOCENT**

Phillipe De Pauw



# Inhoud

<b>1. Discover</b>	<b>5</b>
FUNCTIONELE SPECIFICATIES	5
TECHNISCHE SPECIFICATIES	5
WERKWIJZE	7
<b>2. Define</b>	<b>8</b>
PLANNING	8
PERSONA SHEETS	10
SCENARIOS	10
IDEABOARD	12
MOODBOARD	13
<b>3. Design</b>	<b>14</b>
SITEMAP	14
WIREFRAMES	15
STYLE TILES	23
VISUAL DESIGNS	24
<b>4. Develop</b>	<b>31</b>
FUNCTIONALITEIT	31
<b>5. Deliver</b>	<b>39</b>
HANDLEIDING (ENG.: MANUAL)	39
<b>6. Deploy</b>	<b>42</b>
DEPLOYMENT GUIDE	42



# 1. Discover

## FUNCTIONELE SPECIFICATIES

---

De functionele eisen zijn volledig afhankelijk van de noden van de frontoffice, en indien nodig ook de frontend van de backoffice. Wordt besproken in samenwerking met de docenten en opdrachtgever.

## TECHNISCHE SPECIFICATIES

---

### SERVER

De server wordt geschreven met JavaScript (esnext) in Node.js met het Express framework. Voor authenticatie en autorisatie gebruiken we het Passport framework. Binnen Passport zullen we minimaal een local-strategy, jwt-strategy en een extra strategy (bv: facebook-strategy, google-strategy, twitchtv-strategy, ...) aanbieden.

De server is voorzien van een API (version controlled). Voor ieder collectie uit de database maken we een corresponderende controller, schema, validatie en routes. De volgende actiemethoden zijn minimaal in de controller aanwezig: index (alle documenten uit een collectie tonen, inclusief pagination), show (GET), create, store (POST), edit, update (PUT), delete (DELETE), softdelete (PATCH) en softundelete (PATCH).

Alle API-endpoints worden gedocumenteerd via de OpenAPI-specificatie m.b.v swagger-jsdoc en swagger-ui-express (of ReDoc). Beveilig de swagger-ui-express en dus ook de API-endpoints met JWT. De API is afgeschermd voor niet-bevoegden en voldoet aan de normale eisen voor beveiliging.

Zorg voor een globale foutafhandeling, logging en custom 404 en Error pages (m.b.v. ejs template engine). De client wordt geïnjecteerd in de root route: `http://[your domain]:[your port]/`. De client wordt niet gerenderd op de server, maar wel de informatie in de <head>. Op deze manier wordt de app SEO-friendly en ook om te bookmarken via OpenGraph e.d.. SSR is ook mogelijk, maar combineren met PWA is niet eenvoudig.

## **DATABASE**

We gebruiken MongoDb als databasesysteem. Deze databank moet gehost worden op mLab of MongoDb Gebruik op de campus of binnen het netwerk van de Arteveldehogeschool de wifi "ArteveldeHS Open".

Voor een team dat bestaat uit:

- 1 lid, moeten minimaal 7 collecties (tabellen) aanwezig zijn, exclusief users en roles.
- 2 leden, moeten minimaal 9 collecties (tabellen) aanwezig zijn, exclusief users en roles..
- 3 leden, moeten minimaal 11 collecties (tabellen) aanwezig zijn, exclusief users en roles.

De volgende relaties tussen documenten (en dit zonder embedded documents) moeten minstens eenmaal voorkomen:

- 1..1 (één op één)
- 1..N (één op veel)
- N..N (veel op veel)

De volgende Unit Tests (Mocha, Chai, Sinon) moeten minimaal aanwezig zijn voor een team dat bestaat uit:

- 1 lid: tests voor een minimaal 1 controller, tests voor de auth controller, tests voor minimaal 1 mongoose model
- 2 leden: tests voor een minimaal 2 controllers, tests voor de auth controller, tests voor minimaal 2 mongoose models
- 3 leden: tests voor een minimaal 3 controllers, tests voor de auth controller, tests voor minimaal 3 mongoose models

## **CLIENT**

De cliënt die geïnjecteerd wordt in de server is een PWA React-client met gedeeltelijke SSR (volledig kan ook). De cliënt bevat ook state management via Redux. Met Redux middleware (Thunk, Sagas of Epics) . Het router-systeem dat we zullen gebruiken binnen React is react-router Styling wordt binnen React toegevoegd via scss en / of via styled components.

De volgende Unit Tests (Jest, fetch-mock, expect, Sinon) moeten minimaal aanwezig zijn voor een team dat bestaat uit:

- 1 lid: tests voor een minimaal 1 component, tests voor 1 redux store en middelware
- 2 leden: tests voor een minimaal 2 componenten, tests voor 2 redux stores en middelware
- 3 leden: tests voor een minimaal 3 componenten, tests voor 3 redux stores en middelware

## WERKWIJZE

Na het Algemeen Concept doorloop je deze stappen in volgorde:

- Requirements -> De opgegeven minimumeisen vul je zelf aan in functie van het concept
- Server
  - Data modeling Deze stap is gemeenschappelijk met de backoffice en API
  - Coderen
    - Functionaliteiten programmeren in Node.js (Express + Mongoose)
      - Migrations
      - Seeds
      - Models
      - Controllers
      - Routes
    - Tests runnen

- Client
  - Coderen
    - Functionaliteiten programmeren in React
      - Components
      - Pages
      - Routes
    - Tests runnen

## 2. Define

### PLANNING

---

Bepalen activiteiten

- Template downloaden
- MongoDB opzetten
- MongoDB linken
- API's schrijven voor MongoDB
- Structuur maken
- Dynamische content uit onze MongoDB halen
- CMS systeem
- Authenticatie
- Opstellen dossier
- Poster
- Presentatie

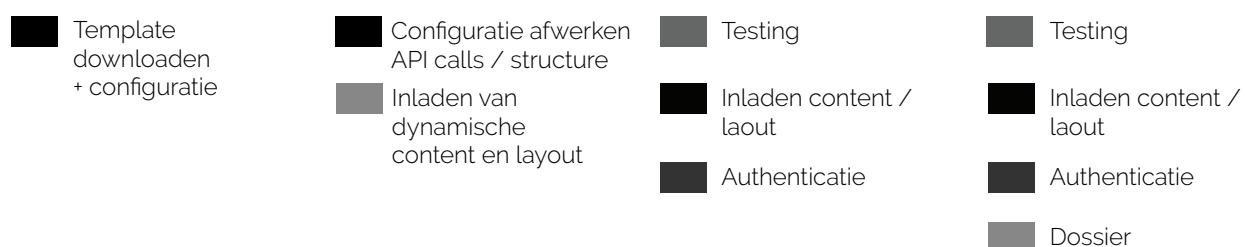
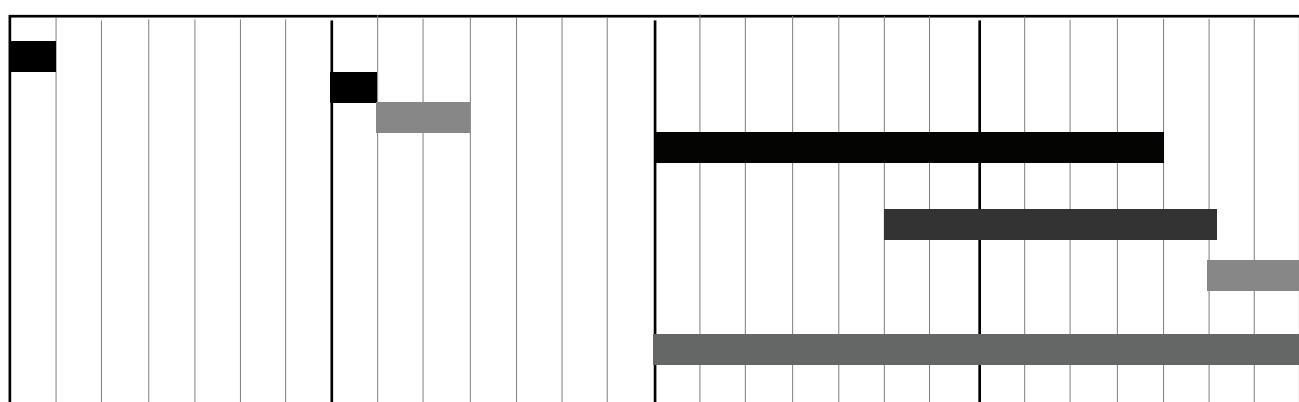
## Bepalen van de prioriteit

- Template downloaden
- MongoDb opzetten
- MongeDb linken
- API's schrijven
- Stuur van de template ( Landing pages / detail pages / components)
- Inladen van content
- Authenticatie
- CMS systeem
- Opstellen van het dossier
- Presentatie
- Poster

## Tijdsinschatting

- 150 u

## Planning uitwerking MobDev2



## PERSONA SHEETS

**Thomas**  
Job: Student  
Locatie: Gent, België

**Jennifer**  
Job: Receptioniste  
Locatie: Wetteren, België

**Marco**  
Job: Grafisch Designer  
Locatie: Antwerpen, België

**Thomas**

**Beschrijving:** Thomas is een 19 jarige student uit Gent. In zijn vrije tijd houdt hij zich vooral bezig met sport en film, zijn twee grote passies. Nu de middelbare school achter de rug is, denkt hij er aan om een opleiding bij Artevelde te beginnen.

**Noden | Doelen:**

- Een opleiding vinden die voor hem geschikt is
- Creativiteit blijven combineren met werk.

**Uitdagingen / Frustraties:**

- Veel verschillende scholen en opleidingen
- Niet veel tijd om een keuze te maken
- Persoonlijke projecten nemen veel tijd in

**Skills:**

Algemene kennis	Video skills
83%	86%

**Jennifer**

**Beschrijving:** Jennifer is een 42 jarige receptioniste uit Wetteren. Haar zoon begint binnenkort aan een hogere opleiding. Ze heeft gehoord dat hij misschien een opleiding GDM wil doen, dus gaat ze op zoek naar meer uitleg over deze opleiding.

**Noden | Doelen:**

- Op een makkelijke en duidelijke manier informatie vinden
- Creativiteit blijven combineren met werk.

**Uitdagingen / Frustraties:**

- Computerbeleid van haar zoon is niet ideaal
- Geen idee waar te zoeken

**Skills:**

Algemene kennis	Internet skills
88%	62%

**Marco**

**Beschrijving:** Marco is een freelance grafisch designer. Na zijn opleiding aan Artevelde is hij begonnen bij een klein bedrijf in Antwerpen, maar is al snel als freelancer begonnen. Als oud-student kijkt hij regelmatig nog eens wat er binnen de opleiding gebeurd.

**Noden | Doelen:**

- Up to date blijven met nieuws binnen de opleiding
- Anderen informeren via getuigenissen

**Uitdagingen / Frustraties:**

- Geen duidelijk systeem om getuigenissen achter te laten
- Enkel nieuws voor oud-studenten in overvolle inbox

**Skills:**

Algemene kennis	Internet skills
74%	92%

## SCENARIOS

**1.**

Als bezoeker wil ik de aangeboden opleidingen bekijken

Stel ik ga met de browser naar <https://gdm.gent> en ik open het menu

Als ik op de link "Opleidingen" klik, dan zou ik een lijst met opleidingen moeten zien

**2.**

Als aangemelde beheerder wil ik een teamlid toevoegen

Stel ik ga met de browser naar <https://gdm.gent/admin>

En ik klik op Team

Als ik op "Teamlid toevoegen" klik

Dan zou ik een formulier om een teamlid toe te voegen moeten zien

### **3.**

Als bezoeker wil ik zien wanneer een open deurdag plaatsvindt

Stel ik open het menu en klik op "Events"

Als ik in de lijst kijk dan zou ik de eerstvolgende open deurdag moeten zien staan

### **4.**

Als bezoeker wil ik het laatste nieuws van de opleiding zien

Stel ik open het menu en ik op "Nieuws"

Dan zou ik de laatste nieuwe artikels uit de opleiding moeten zien

### **5.**

Als aangemelde beheerder wil ik nieuws kunnen verwijderen

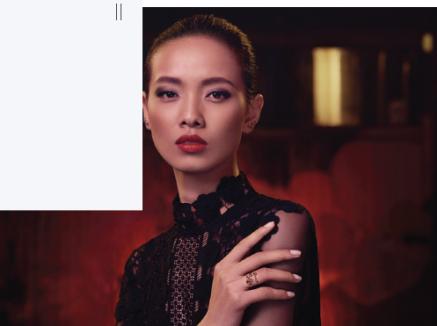
Stel ik ga in de browser naar <https://gdm.gent/admin>

En ik klik op nieuws en zoek een artikel in de lijst

Als ik op de knop "delete" klik dan zou het artikel verwijderd moeten worden

# Perspectives on the Digital World —

With a team of creative and development specialists, we consistently strive to be at the forefront of new media technology.



Iron-on Pockets  
Stitch to any apparel item.  
[Shop Now](#)

Brand  
Creative  
Code  
Growth

Naming / Identity /  
Visual language / Values /  
Guidelines / Tone of voice

UI/UX / Design / Direction /  
Print / Campaigns /  
Experience / Wayfinding /  
Animation

Front end / Back end /

ARTICLE



James Green  
Director



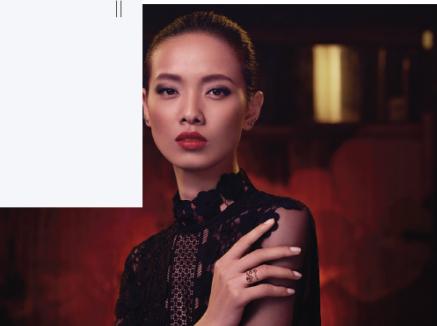
Charles Fisher: A Year at Chaptr in  
Digital Marketing.

2 days ago

## MOODBOARD

# Perspectives on the Digital World —

With a team of creative and development specialists, we consistently strive to be at the forefront of new media technology.



Brand Creative Code Growth

Naming / Identity / Visual language / Values / Guidelines / Tone of voice

UI/UX / Design / Direction / Print / Campaigns / Experience / Wayfinding / Animation

Front end / Back end /

ARTICLE



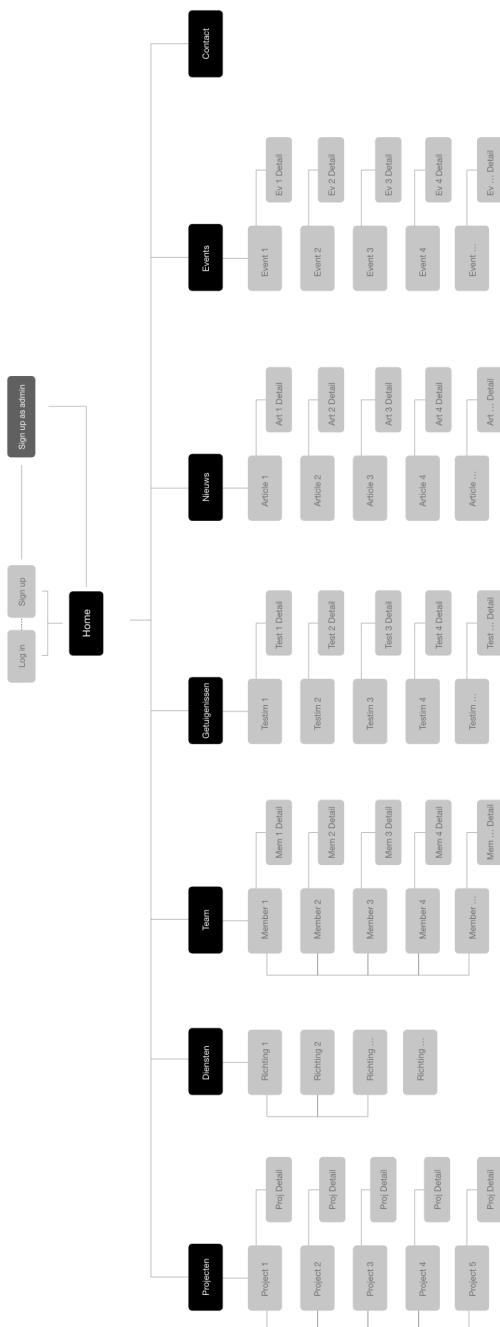
James Green  
Director

Charles Fisher: A Year at Chaptr in Digital Marketing.

2 days ago

# 3. Design

## SITEMAP



## WIREFRAMES

### **Homepage**



## PROJECTS



BUTTON



## Project page

MENU

### PROJECT PAGE



BUTTON

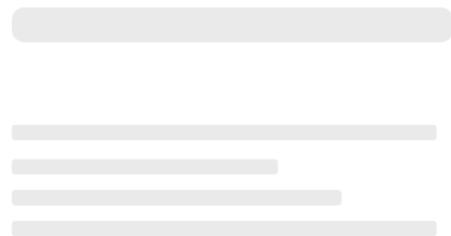


## Project detail

MENU

PROJECT TITLE

Project subtitle



## TEAM PAGE



Person name

|||||

BUTTON



Person name

|||||

BUTTON



Person name

|||||

BUTTON

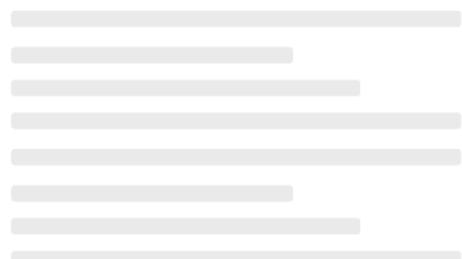


|||||

|||||

## TEAM DETAIL PAGE

Person name



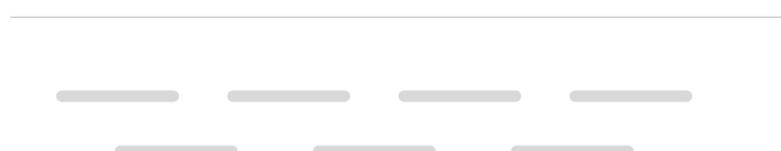
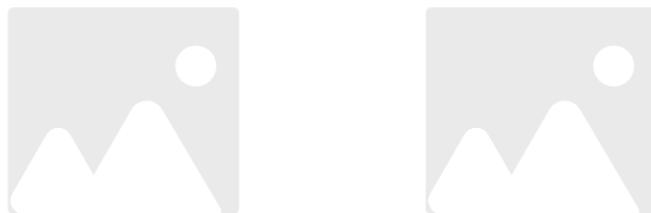
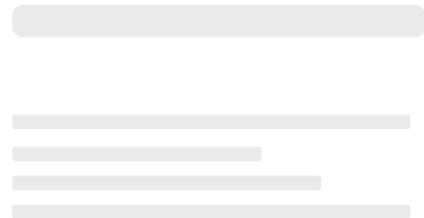
**News page**

MENU

## NEWS PAGE



## News Detail page



# STYLE TILES

---

## Colors

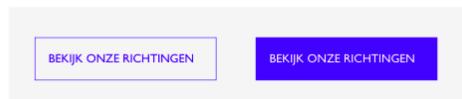


## Fonts

Gill Sans Std is our main font  
This is our main paragraph text  
**“A quote looks like this”**  
**Section title**  
PROJECT TITLE

## Buttons

Main call to action



Normal state

Hover state

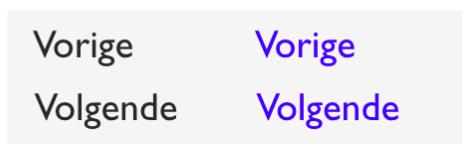
Secondary call to action



Normal state

Hover state

Main link button



Normal state

Hover state

Secondary link button

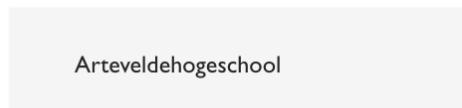


Normal state

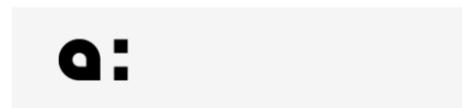
Hover state

## Logo

Text logo



Logo



## VISUAL DESIGNS

**Homepage**

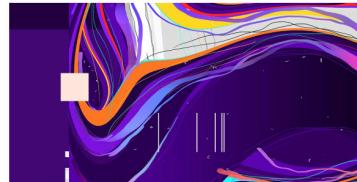
# Grafische en digitale media

Let's build the future together

scroll down

# Projecten

Thrive



Aftermovie GMB Café



Erasmusproject: Exchange Knowledge  
for Future Innovation

Op donderdag 6 december vond in de  
ontspanningsruimte het eerste GMB  
Café plaats

Aire project



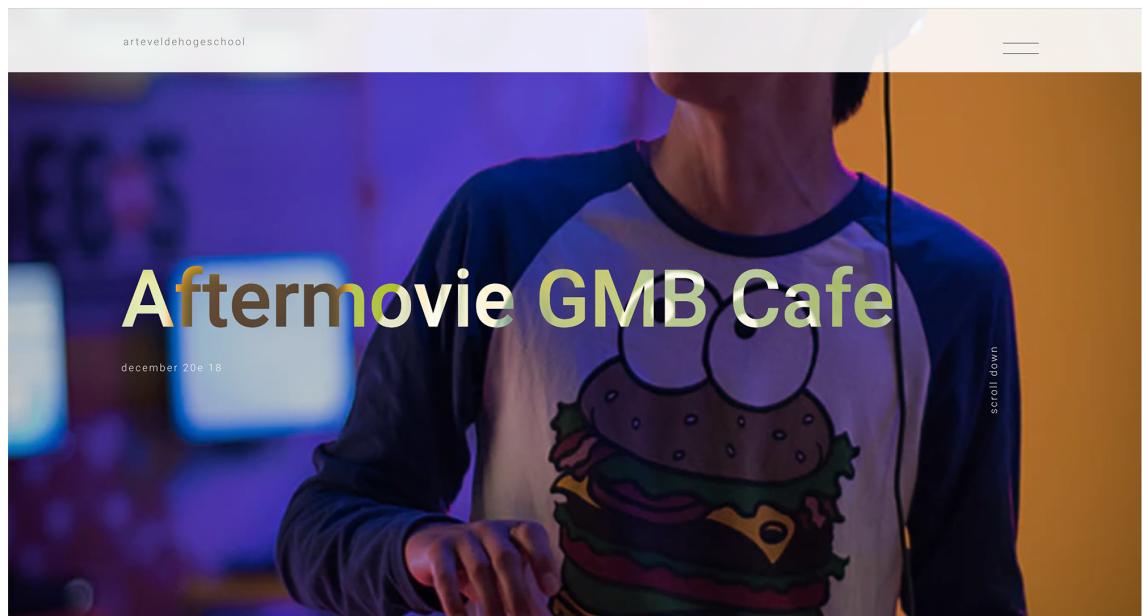
Aire is de online repository van de  
Arteveldehogeschool.

Hybrid App - Kleen



Hybrid App dat het ecologische,  
economische en sociaal probleem van  
zwerfvuil en sluijkstorten kan helpen  
oplossen.

## Project detail



# Aftermovie GMB Cafe

december 20e 18

scroll down

## Aftermovie GMB Cafe

december 20e 18

In een gezellig aangeklede ontspanningsruimte verwelkomden de studenten 13 professionals die reeds hun GMB-skills dagelijks toepassen in de praktijk. Het event werd afgetrapt door een korte voorstelling van de professionals. Hierdoor konden de studenten al een keuze maken qua professionals waarmee ze verder contact wensten te leggen tijdens het formele gedeelte. Hierbij gingen de studenten in groepjes van 2 of 3 bij een professional aan tafel zitten en daar stellen ze een aantal vragen aan de hand van een spelletje Jenga, 4 op een rij of Uno. Na enkele vragen zag je al snel vlotte gesprekken ontstaan tussen de studenten en professionals. Terwijl er werd rondgegaan met zelfgemaakte hapjes door de studenten, schoven de studenten door naar de tweede professional waar ze wederom een hele reeks vragen voor hadden voorbereid. Na dit eerste formele gedeelte werden de gasten uitgenodigd om een glaasje te drinken tijdens het informele gedeelte. Hier konden de studenten nog wat extra vragen stellen aan de professionals of rustig napraten over de gesprekken die ze hadden. Enkele GMB-studenten uit het derde jaar waren ook aanwezig om "GMB doet mee" te promoten door het verkopen van gepersonaliseerde drinkflessen en tote bags. We kunnen terugkijken op een geslaagd evenement waar de studenten leerden netwerken, maar ook een gezellige avond met de professionals en docenten beleefden!

[Terug naar alle projecten](#)

## Lester Lievens

Consultant



## Dieter De Weirdt

Back end developer



## Frederik Roegiers

Back end developer



## Philip De Pauw

Full stack developer



[Team detail](#)

# Philip De Pauw

Full stack developer



**"Two things are infinite: the universe and human stupidity; and I'm not sure about the universe."**

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

[depauw@gmail.com](mailto:depauw@gmail.com)

[Terug naar teamleden](#)

## Nieuws

Ipsa ut dolor impedit earum.

---

At exercitationem fugiat ut earum et excepturi  
vel commodi.

---

Animi aliquam similique neque sunt et quam  
consectetur.

---

## Overlay menu

arteveldehogeschool



Projecten  
Diensten  
Team  
Getuigenissen  
Nieuws  
Events  
Contact

Github  
Facebook  
Twitter  
Vimeo  
Instagram

### FIND US HERE

Industrieweg 232, 9090 Mariakerke  
+32 9 234 86 00  
info.grafische.digitalemedia@arteveldehs.be

## 4. Develop

### FUNCTIONALITEIT

- Coding/Programmeren van
  - Frontend
  - Database
  - Backend
  - CMS
  - API
- Testing

```

import mongoose from 'mongoose';
import mongoosePaginate from 'mongoose-paginate';
import slug from 'slug';

const { Schema } = mongoose;

const ProjectSchema = new Schema(
  {
    title: { type: String, required: true, max: 128 },
    synopsis: { type: String, required: true, max: 1024 },
    body: { type: String, required: true },
    slug: {
      type: String, lowercase: true, unique: true, required: true,
    },
    creator: { type: String, max: 128 },
    published_at: { type: Date, required: false },
    deleted_at: { type: Date, required: false },
    categoryId: { type: Schema.Types.ObjectId, ref: 'Category', required: false },
  },
  {
    toJSON: { virtuals: true },
    toObject: { virtuals: true },
  },
  {
    timestamps: { createdAt: 'created_at', updatedAt: 'updated_at' },
  }
);

ProjectSchema.methods.slugify = function () {
  this.slug = slug(this.title);
};

Complexity is 3 Everything is cool!
ProjectSchema.pre('validate', function (next) {
  if (!this.slug) {
    this.slugify();
  }
  return next();
});

ProjectSchema.virtual('id').get(function () { return this._id });

ProjectSchema.virtual('category', {
  ref: 'Category',
  localField: 'categoryId',
  foreignField: '_id',
  justOne: true,
});
ProjectSchema.virtual('images', {
  ref: 'ProjectImage',
  localField: '_id',
  foreignField: 'projectId',
  justOne: false,
})

ProjectSchema.plugin(mongoosePaginate);
export default mongoose.model('Project', ProjectSchema)

```

```

import { Project } from '../database';
import { APIError, handleAPIError } from '../../../../../utilities';

Complexity is 14 You must be kidding
class ProjectController {
  Complexity is 14 You must be kidding
  index = async (req, res, next) => {
    try {
      const { limit, skip } = req.query;
      let projects = null;

      if (limit && skip) {
        const options = {
          page: parseInt(skip, 10) || 1,
          limit: parseInt(limit, 10) || 10,
          populate: ['images', 'category'],
          sort: { created_at: -1 },
        };
        projects = await Project.paginate({}, options);
      } else {
        projects = await Project.find().populate(['category', 'images']).sort({
          created_at: -1
        }).exec();
      }

      if (projects === undefined || projects === null) {
        throw new APIError(404, 'Collection for projects not found');
      }

      return res.status(200).json(projects);
    } catch (err) {
      return handleAPIError(500, err.message || 'Some error occurred while retrieving posts', next);
    }
  };
}

Complexity is 10 It's time to do something...
show = async (req, res, next) => {
  try {
    const { id } = req.params;
    const item = await Project.findById(id).populate(['category', 'images']).exec();

    if (item === undefined || item === null) {
      throw new APIError(404, `Project with id ${id} not found`);
    }

    return res.status(200).json(item);
  } catch (err) {
    return handleAPIError(err.status || 500, err.message || 'Something went wrong while retrieving posts', next);
  }
}

```

```
import { ProjectController } from '../controller';

const projectController = new ProjectController();

const initializeEndpoints = (parentRouter, authService) => {
  /**
   * @swagger
   * /api/v1/projects:
   *   get:
   *     tags:
   *       - Projects
   *     description: Returns all projects
   *     produces:
   *       - application/json
   *     responses:
   *       200:
   *         description: An array of projects
   */
  parentRouter.get('/projects', projectController.index);
  /**
   * @swagger
   * /api/v1/projects/create:
   *   get:
   *     tags:
   *       - Project
   *     description: Returns specific viewmodel such as projects
   *     produces:
   *       - application/json
   *     responses:
   *       200:
   *         description: Create project
   */
  parentRouter.get('/projects/create', projectController.create);
  /**
   * @swagger
   * /api/v1/projects/{id}:
   *   get:
   *     tags:
   *       - Project
   *     description: Returns specific project
   *     produces:
   *       - application/json
   *     parameters:
   *       - name: id
   *         description: Project id
   *         in: path
   *         required: true
   *         type: string
   *     responses:
   *       200:
   *         description: Get project by id
   */
  parentRouter.get('/projects/:id', projectController.show);
  /**
   * @swagger
   * /api/v1/projects:
   *   post:
   *     tags:
   *       - Projects
   *     description: Create a new project
   *     produces:
   *       - application/json
   *     responses:
   *       201:
   *         description: Created project
   */
  parentRouter.post('/projects', projectController.create);
}

export default initializeEndpoints;
```

```

Complexity is 6 It's time to do something...
initializeGithubStrategy = () => {
  passport.use(new GitHubStrategy(
    {
      clientID: config.auth.github.clientID,
      clientSecret: config.auth.github.clientSecret,
      callbackURL: "http://127.0.0.1:8080/api/v1/login/github/redirect",
    },
    Complexity is 5 Everything is cool!
    function(accessToken, refreshToken, profile, done) {
      Complexity is 4 Everything is cool!
      User.findOne({ 'githubProvider.id': profile.id }).then((currentUser) => {
        if(currentUser) {
          done(null, currentUser);
        } else {
          new User({
            email: profile.emails[0].value,
            githubProvider: {
              id: profile.id,
              token: accessToken,
            }
          }).save().then((newUser) => {
            done(null, newUser);
          });
        }
      });
      // User.findOrCreate({ githubId: profile.id }, function(err, user) {
      //   return cb(err, user);
      // });
    }
  ));
}

Complexity is 10 It's time to do something...
initializeLocalStrategy = () => {
  passport.use(new LocalStrategy({
    usernameField: 'email',
  },
  Complexity is 9 It's time to do something...
  async (email, password, done) => {
    const user = await User.findOne({email});
    if (!user) { return done(null, false); }
    Complexity is 5 Everything is cool!
    return user.comparePassword(password, (isMatch) => {
      if(isMatch) {
        return done(null, false);
      } else {
        return done(null, user);
      }
    });
  })
  // passport.use(new LocalStrategy(

```

The screenshot shows the Postman application interface. On the left, there's a sidebar with a history of requests and a collection named 'Collections'. The main area has a search bar at the top. Below it, a URL field contains 'GET http://127.0.0.1:8080/api/v1/projects'. To the right of the URL are 'Send' and 'Save' buttons. The request type is set to 'GET' and the URL is 'http://127.0.0.1:8080/api/v1/projects'. The 'Body' tab is selected, showing a JSON response. The response body is a list of projects, with one specific project highlighted in red. The highlighted project's details are as follows:

```

1  [
2    {
3      "_id": "5ceba948864a75d4acef2be5",
4      "title": "Thrive",
5      "synopsis": "Gedurende 3 jaar zal de opleiding 'Grafische en Digitale Media' van de Arteveldehogeschool samen werken met 5 buitenlandse partners in het kader van Erasmus+ voor het project 'EKI!' (Exchange Knowledge for Future Innovation). Dit project heeft als doel om een platform te ontwikkelen om vooral innovatief lesmateriaal uit te wisselen tussen verschillende organisaties in de creatieve en crossmediale sector, zoals bv. secundaire en hogeschool, bedrijven & organisaties die belang hebben in de ontwikkeling van jonge talenten. De belangrijkste doelgroep bestaat uit docenten en studenten die interesse hebben in het uitwisselen van digitale lesmateriaal in het domein van crossmedia (print & multimedia), marketing en design. Heb je interesse, neem dan snel contact met Inge Sintobin mail us.",
6      "body": "Gedurende 3 jaar zal de opleiding 'Grafische en Digitale Media' van de Arteveldehogeschool samen werken met 5 buitenlandse partners in het kader van Erasmus+ voor het project 'EKI!' (Exchange Knowledge for Future Innovation). Dit project heeft als doel om een platform te ontwikkelen om vooral innovatief lesmateriaal uit te wisselen tussen verschillende organisaties in de creatieve en crossmediale sector, zoals bv. secundaire en hogeschool, bedrijven & organisaties die belang hebben in de ontwikkeling van jonge talenten. De belangrijkste doelgroep bestaat uit docenten en studenten die interesse hebben in het uitwisselen van digitale lesmateriaal in het domein van crossmedia (print & multimedia), marketing en design. Heb je interesse, neem dan snel contact met Inge Sintobin mail us.",
7      "category": "5cd816742bb6276aa9b9340d",
8      "slug": "this-is-a-test-project",
9      "img": "5cd816742bb6276aa9b9340d",
10     "id": "5ceba948864a75d4acef2be5",
11     "category": {
12       "_id": "5cd816742bb6276aa9b9340d",
13       "name": "imedit",
14       "description": "Omnis qui vero.",
15       "slug": "imedit",
16       "img": "5cd816742bb6276aa9b9340d"
17     },
18     "images": [
19       {
20         "id": "5cd816742bb6276aa9b9340d"
21       }
22     ]
23   }
24 ]

```

```

33      <Componenf={Бонфігурація}|
34      |   яглонеf={Бадерялоне}|
35      |   бету="\\b1o\есfс\:\Tq"|
36      |   exscf|
37      |<Бонфігурація|
38      |   \|
39      |   componenf={Б1o\есfс\:\Tq}|
40      |   яглонеf={Бадерялоне}|
41      |   бету="\\b1o\есfс"|
42      |   exscf|
43      |<Бонфігурація|
44      |   \|
45      |   componenf={Немсб986}|
46      |   яглонеf={Бадерялоне}|
47      |   бету="\\u6m8\:\Tq"|
48      |   exscf|
49      |<Бонфігурація|
50      |   \|
51      |   componenf={Номе986}|
52      |   яглонеf={Бадерялоне}|
53      |   бету="\\u"|
54      |   exscf|
55      |<Бонфігурація|
56      |   \|
57      |<Спільні|
58      |   Ленгел() {|
59      |   Спільні Мети експорт componenf .{|
60      |   тмовеf contract986 flow "\\\\"\\p986\contract\contract986"|
61      |   тмовеf { 20000 } flow "02"|
62      |   тмовеf "\\\\"\\p986\c88"|
63      |

```

```

39 //Events api calls
40 static findAllEvents = async (queryParams = null) => {
41   let url = `${this.URL}/events`;
42   if (queryParams !== null) {
43     url += `?${url.indexOf("?") === -1 ? "?" : "&"}${this.queryParams(queryParams)}`;
44   }
45   const response = await fetch(`${url}`);
46   return await response.json();
47 };
48 static findOneEvent = async id => {
49   const response = await fetch(`${this.URL}/events/${id}`);
50   return await response.json();
51 };
52 //Team api calls
53 static findTeam = async (queryParams = null) => {
54   let url = `${this.URL}/team`;
55   if (queryParams !== null) {
56     url += `?${url.indexOf("?") === -1 ? "?" : "&"}${this.queryParams(queryParams)}`;
57   }
58   const response = await fetch(`${url}`);
59   return await response.json();
60 };
61 static findOneMember = async id => {
62   const response = await fetch(`${this.URL}/team/${id}`);
63   return await response.json();
64 };
65 //Testimonial api calls
66 static findTestimonials = async (queryParams = null) => {
67   let url = `${this.URL}/testimonials`;
68   if (queryParams !== null) {
69     url += `?${url.indexOf("?") === -1 ? "?" : "&"}${this.queryParams(queryParams)}`;
70   }
71   const response = await fetch(`${url}`);
72   return await response.json();
73 };
74 static findOneTestimonial = async id => {
75   let response = await fetch(`${this.URL}/testimonials/${id}`);
76   return response.json();
77 };
78 //Testimonial api calls
79 static findOneTestimonial = async id => {
80   let response = await fetch(`${this.URL}/testimonials/${id}`);
81   return response.json();

```

```

1  /*
2  * Import external libraries
3  */
4  import React, { Component } from 'react';
5  import { Redirect, Route } from 'react-router-dom';
6
7  /*
8  Layout
9  */
10 import { ContentLayout } from '../../../../../layouts';
11
12 /*
13 Pages
14 */
15 import TeamsTablePage from '../team-table';
16 import TeamFormPage from '../team-form';
17
18 const tabs = [
19   { id: 'List', link: '/admin/team' },
20   { id: 'Create new member', link: '/admin/team/create' },
21 ];
22
23 class TeamsOverviewPage extends Component {
24   render() {
25     const { children } = this.props;
26
27     return (
28       <ContentLayout title="Team Overview" tabs={tabs}>
29         { children }
30       <Route exact path="/admin/team" component={ TeamsTablePage }/></Route>
31       <Route path="/admin/team/create" component={ TeamFormPage }/></Route>
32       <Route path="/admin/team/:id/edit" component={ TeamFormPage }/></Route>
33     </ContentLayout>
34   )
35 }
36
37 }
38
39 export default (TeamsOverviewPage);
```

```

10 /*
11 * Import components
12 */
13 import GridWrapper from "../../components/structural-components/grid-wrapper/GridWrapper";
14 import Title from "../../components/text-components/title>Title";
15 import GeneralNav from "../../components/header/general-nav/GeneralNav";
16 import BlankDiv from "../../components/styled-components/blank-div/BlankDiv";
17 import Footer from "../../components/footer";
18 import OverlayMenu from "../../components/structural-components/overlay-menu/OverlayMenu";
19 import EasyFlexRow from "../../components/structural-components/flexbox/easy-flex-row/EasyFlexRow";
20 import EasyFlexCol from "../../components/structural-components/flexbox/easy-flex-col/EasyFlexCol";
21
22 import "./TeamDetailPage.scss";
23
24 class TeamDetailPage extends Component {
25   state = {
26     member: null
27   };
28
29   componentWillMount() {
30     this.Member(this.props.match.params.id);
31   }
32
33   Member = id => {
34     Api.findOneMember(id)
35       .then(data => {
36         this.setState(prevState => ({
37           ...prevState,
38           member: data
39         }));
40       })
41       .catch(error => {});
42   };
43
44   toggleMenu = e => {
45     e.preventDefault();
46     this.setState(state => ({ showMenu: !state.showMenu }));
47   };
48
49   getParentState = e => {
50     let parentState = this.state.showMenu;
51     return parentState;
52   };
53 }
```

```

131     componentWillMount() {
132       this.loadTeams();
133     }
134   }
135 
136   loadTeams = () => {
137     fetch('/api/v1/team')
138       .then(response => response.json())
139       .then(item => this.setState({ teams: item }));
140   }
141 
142   render() {
143     const { classes } = this.props;
144     const { teams } = this.state;
145 
146     return (
147       <Paper className={classes.root}>
148         <div className={classes.tableWrapper}>
149           <Table className={classes.table} aria-labelledby="tableTitle">
150             <TableHead>
151               <TableRow>
152                 <TableCell>Firstname</TableCell>
153                 <TableCell>Lastname</TableCell>
154                 <TableCell>Job</TableCell>
155                 <TableCell>Email</TableCell>
156                 <TableCell>Image</TableCell>
157                 <TableCell>Bio</TableCell>
158                 <TableCell>Quote</TableCell>
159               </TableRow>
160             </TableHead>
161             <TableBody>
162               {team && teams.map( (team, index) => (
163                 <TableRow key={team.id}>
164                   <TableCell>{team.first_name}</TableCell>
165                   <TableCell>{team.last_name}</TableCell>
166                   <TableCell>{team.job}</TableCell>
167                   <TableCell>{team.email}</TableCell>
168                   <TableCell>{team.image_path}</TableCell>
169                   <TableCell>{team.bio}</TableCell>
170                   <TableCell>{team.quote}</TableCell>
171                   <TableCell>
172                     <IconButton
173                       component={Link} to={`/admin/team/${team.id}/edit`}>
174                       <IconCreate />
175                     </IconButton>
176 
177                     <IconButton
178                       onClick={() => this.handleDialogOpen(team.id, POSTACTIONSENUM.DELETE)}>
179                       <IconDeleteForever />
180                     </IconButton>
181                   </TableCell>
182                 </TableRow>
183               )));
184             </TableBody>
185           </Table>
186         </div>
187       <Dialog

```

```

38  */
39  import "./Main.css";
40  import { homedir } from "os";
41  import ContactPage from "./pages/contact/ContactPage";
42 
43  class Main extends Component {
44    render() {
45      return (
46        <div className="App">
47          <Switch>
48            <RouteWithLayout
49              exact
50              path="/"
51              layout=(PageLayout)
52              component={HomePage}
53            />
54            <Redirect from="/home" to="/" />
55            <RouteWithLayout
56              exact
57              path="/news"
58              layout=(PageLayout)
59              component={NewsPage}
60            />
61            <RouteWithLayout
62              exact
63              path="/news/:id"
64              layout=(PageLayout)
65              component={NewsDetailPage}
66            />
67            <RouteWithLayout
68              exact
69              path="/projects"
70              layout=(PageLayout)
71              component={ProjectsPage}
72            />
73            <RouteWithLayout
74              exact
75              path="/projects/:id"
76              layout=(PageLayout)
77              component={ProjectDetailPage}
78          </Switch>
79        </div>
80      );
81    }
82  }
83 
84  export default Main;

```

```

67     render() {
68       const { team } = this.state;
69       this.items = team.map(item => (
70         <div key={item.id} className="col-space-between team-card-wrapper">
71           <h2 className="team-name-title">
72             {item.first_name} {item.last_name}
73           </h2>
74           <p className="team-jobssubscription">{item.job}</p>
75           <div className="card-zoom">
76             <a href={"/team/" + item.id}>
77               <img src={item.image_path} className="team-thumbnail" />
78             </a>
79           </div>
80           <a href={"/team/" + item.id} className="team-read-more-button">
81             i
82           </a>
83         </div>
84       );
85     );
86     return (
87       <React.Fragment>
88         <OverlayMenu menustate={this.state.showMenu} />
89         <GridWrapper style="html-wrapper">
90           <GridWrapper style="main-page-wrapper">
91             <BlankDiv style="blank-div-lg" />
92             <GeneralNav
93               toggleMenu={this.toggleMenu}
94               menustate={this.getParentState()}>
95             />
96             <Title style="section-title" text="Team" />
97             <Paragraph
98               text="Wat als je drie studententeams in verschillende landen dezelfde briefing geeft? En ze vervolgens parallel aan hun eigen op"
99               style="standard-text-paragraph par-pos-1 paragraph-mb-med" />
100            </Paragraph>
101            <EasyFlexRow style="team-general-page-wrapper">
102              {this.items}
103            </EasyFlexRow>
104            <Footer />
105          </GridWrapper>
106        </GridWrapper>
107      </React.Fragment>
108    );
109  }
110}
111
112 export default TeamPage;
113 var imageStyle = {
114   width: "300px",
115   height: "300px",
116   objectFit: "cover"
117 };
118

```

```

27 class TeamPage extends Component {
28   state = {
29     team: [],
30     post: null
31   };
32   async componentWillMount() {
33     await this.loadTeam(1);
34   }
35
36   loadTeam = pageIndex => {
37     Api.findTeam({ limit: 4, skip: pageIndex })
38     .then(data => {
39       const prevMember = this.state.team;
40       const newMember = [...prevMember, ...data.docs];
41       this.setState(prevState => ({
42         ...prevState,
43         team: newMember,
44         pagination: {
45           limit: data.limit,
46           page: data.page,
47           pages: data.pages,
48           total: data.total
49         }
50       }));
51     })
52     .catch(error => {
53       console.log(error);
54     });
55   };
56
57   toggleMenu = e => {
58     e.preventDefault();
59     this.setState(state => ({ showMenu: !state.showMenu }));
60   };
61
62   getParentState = e => {
63     let parentState = this.state.showMenu;
64     return parentState;
65   };
66
67

```

# 5. Deliver

## HANDLEIDING (ENG.: MANUAL)

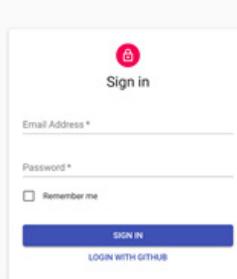
---

### 1. Gebruik van de applicatie

Deze handleiding is voor het gebruik van het administrator-gedeelte van de gdm.gent applicatie. De verschillende onderdelen van de CMS en hun gebruik zullen hierin besproken worden.

### 2. Login

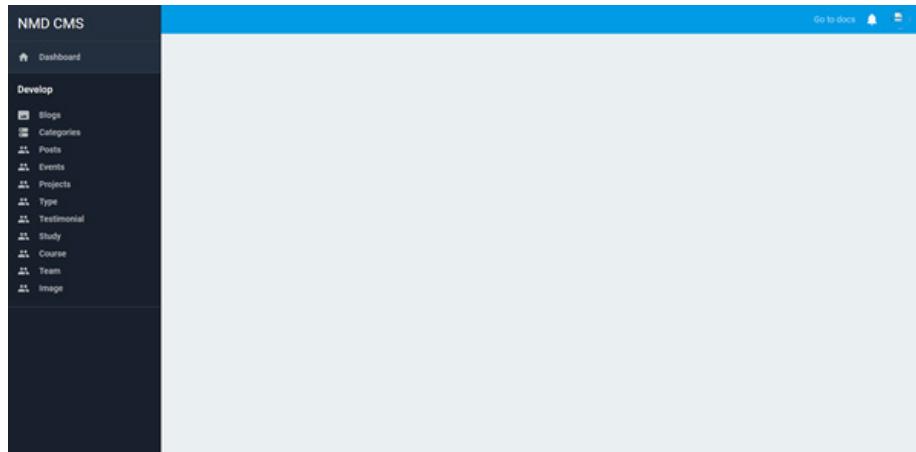
Voor gebruik te maken van de applicatie moet de gebruiker eerst inloggen. De link naar de loginpagina staat niet vermeld in het zichtbare deel van de applicatie. Om in te loggen in het administrator-gedeelte moet de gebruiker navigeren naar de volgende URL: <https://gdm.gent/login>.



De gebruiker kan hier kiezen om ofwel lokaal ofwel via een Github account in te loggen. Nadat de gebruiker is ingelogd komt deze in het admin-gedeelte van de applicatie.

### 3. CMS

Het admin-gedeelte van de applicatie is een CMS (Content Management Systeem). Hierin kan de content van de website aangemaakt en aangepast worden. Het belangrijkste voor het gebruik van dit CMS bevindt zich aan de linkerkant van het scherm. Hier staan de verschillende categorieën van de content die kan aangepast worden op de website. In het geval van deze applicatie bestaat dit uit de volgende soorten:



Categorieën: Categorieën voor posts en projecten

Posts: Nieuwsberichten

Events: Alle events georganiseerd voor en door de opleiding.

Projects: Verschillende projecten gemaakt in de opleiding

Types: Verschillende types voor testimonials (bv. oud-leerling)

Testimonials: Getuigenissen over de opleiding

Study: Verschillende afstudeerrichtingen en keuzetrajecten

Course: Vakken aangeboden door de opleiding

Team: Teamleden binnen de opleiding/campus

Image: Afbeeldingen voor projecten

#### 4. Content

Content aanpassen kan door op de linkerkant van het scherm op een van de opties te klikken. De gebruiker komt uit op het volgende scherm

Title	Body	Date	Actions
Exchange knowledge for future innovation	<p>Gedurende 3 jaar zal de opleiding 'Grafische en Digitale Media' van de Artesia Hogeschool samen werken met 5 buitenlandse partners in het kader van Erasmus+ aan het project TKKF (Exchange Knowledge for Future Innovation). Dit project heeft als doel om een platform te ontwikkelen om vooruit innovatief leeraanbod uit te wisselen tussen verschillende organisaties in de creatieve en creatimedialen sectie, zoals bv. secundaire en hogeschools, universiteiten en bedrijven. Het project begon in september 2018 en zal eindigen in december 2019 in Schiphol in Nederland. Meer info over dit project is te vinden op EFKE-project. Het onderzoeksteam op zoek naar kleinere bedrijven en onderwijsinstellingen die interesse hebben in het uitwisselen digitaal leeronderzoek in het gebied van grafische en digitale media (print &amp; multimedia), marketing en design. Heb je interesse, neem dan snel contact met Inge Simóbin mail us.</p> <p>Het project THRIVE organiseerde een internationale 'Train the trainer'-week op onze campus. Deelnemende collega's konden hier hun lesgevenstrategieën voor het aankomende semester (onderwijsperiode) verfijnen door middel van verschillende workshops. Daarnaast was er een bezoek aan de Universitat de València in Spanje en het Vrije Kunsthuis, aangevuld met onze eigen medewerkers. Het project THRIVE (Erasmus+) heeft het afgelopen jaar lespakketten ontwikkeld voor ondernemerschap in de creatieve industrie voor drie specifieke doelgroepen: jonge ondernemende professionals, docenten en studenten. Tijdens de week hadden de docenten een dubbele rol: ze gaven feedback op de lerarmaterialen, maar moesten ze ook als 'lecturer' ondergaan. Zo wisten ze in groepjes van drie design challenge volgens de 'Human-centered design' methode op te lossen. De studenten waren deelnemers aan de workshop voor het ontwerp van een speciale managementgame. Daarnaast kregen ze een presentatie waarin het werk van Idea Factory en Artpreneur op onze school besproken werd. - inspirerend, want niet iedere school heeft zo'n programma om ondernemende studenten te ondersteunen! Meer informatie over het project THRIVE is te vinden op de website <a href="https://www.thriveproject.eu/">https://www.thriveproject.eu/</a></p>	2019-09-04T00:00:00.000Z	<a href="#">Edit</a> <a href="#">Delete</a>
Thrive: Is ondernemerschap iets dat je kan leren?	<p>Tijdens de week hadden de docenten een dubbele rol: ze gaven feedback op de lerarmaterialen, maar moesten ze ook als 'lecturer' ondergaan. Zo wisten ze in groepjes van drie design challenge volgens de 'Human-centered design' methode op te lossen. De studenten waren deelnemers aan de workshop voor het ontwerp van een speciale managementgame. Daarnaast kregen ze een presentatie waarin het werk van Idea Factory en Artpreneur op onze school besproken werd. - inspirerend, want niet iedere school heeft zo'n programma om ondernemende studenten te ondersteunen! Meer informatie over het project THRIVE is te vinden op de website <a href="https://www.thriveproject.eu/">https://www.thriveproject.eu/</a></p>	2019-09-04T00:00:00.000Z	<a href="#">Edit</a> <a href="#">Delete</a>
Opendag 2019	<p>Kom alles te weten over de verschillende afstudeerrichtingen en bijkomende specialisaties op onze mediacampus tijdens de opendag op zaterdag 29 juni 2019. THIS IS WHO WE DO. Benieuwd hoe de studie aanbieding eruitziet? Kom langs en ontdek de mogelijkheden. We vertellen je alvast hoe onze studenten deze leuke sector ervaren. Ga in interactie met studenten, docenten en projectbegeleiders. Blader even door de cursussen en aanschouw de vele werken van de studenten, verspreid over de mediacampus. Kiebelt het na de introductie om aan de slag te gaan met je creativiteit? Dan zien we je graag terug in september!</p>	2019-06-30T00:00:00.000Z	<a href="#">Edit</a> <a href="#">Delete</a>

Dit is een overzicht van de verschillende data binnen een van de opties. Deze kan hier bekijken worden. In de rechterkolom van de tabel vinden we de acties die de gebruiker kan ondernemen voor de data.

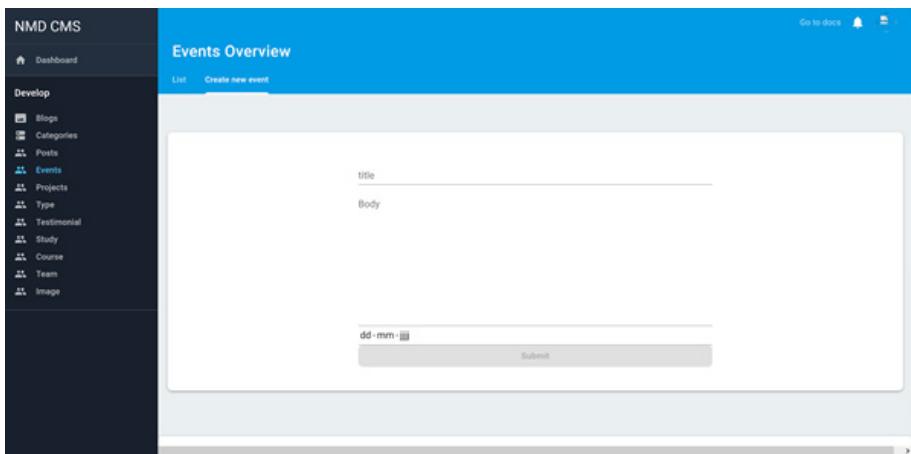
#### 4.1. Content verwijderen

Content verwijderen kan op twee manieren gebeuren. De eerste manier is een "soft delete". Dit zorgt ervoor dat de data niet meer op de website beschikbaar is, maar wel nog in het CMS beschikbaar is en kan aangepast worden.

De tweede manier is dan een definitieve delete. Dit verwijderd de data uit de CMS en dus ook de database. Dit is permanent en de data kan dus niet worden teruggehaald.

#### 4.2. Content toevoegen

Content kan toegevoegd worden via de tweede tab. Hiervoor klikt de gebruiker op "Create new ... ". De gebruiker zal dan dit formulier zien.

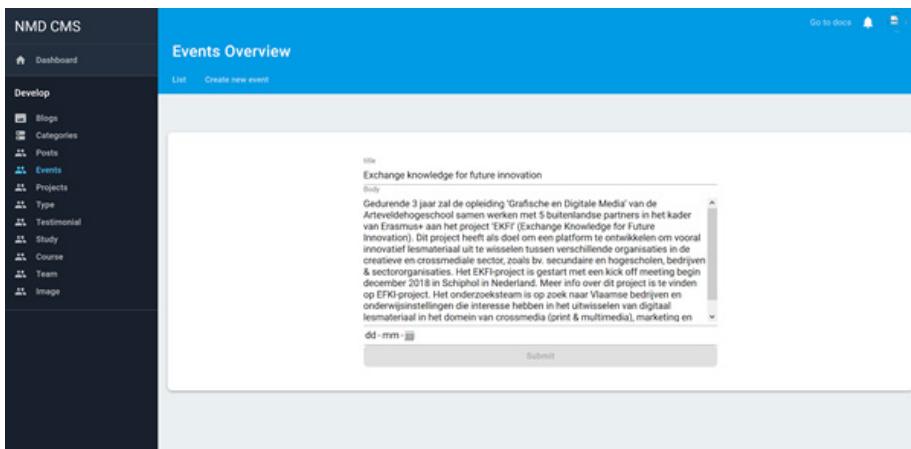


The screenshot shows the 'Events Overview' page in the NMD CMS. On the left, there's a sidebar with a dark background and white text, listing categories like 'Dashboard', 'Develop', 'Blogs', 'Categories', 'Posts', 'Events' (which is currently selected and highlighted in blue), 'Projects', 'Type', 'Testimonial', 'Study', 'Course', 'Team', and 'Image'. The main content area has a light blue header with the title 'Events Overview' and a 'Create new event' button. Below the header is a large input form. It has three text input fields: 'title' (with placeholder 'title'), 'Body' (with placeholder 'Body'), and a date input field 'dd-mm-yy' (with placeholder 'dd-mm-yy'). At the bottom of the form is a grey 'Submit' button.

Wanneer dit formulier met de overeenkomstige data is ingevuld, wordt dit toegevoegd aan de database en kan de gebruiker deze verwijderen zoals hierboven beschreven.

#### 4.3. Content aanpassen

Content kan ook bewerkt worden door op het potlood-icoon te klikken. Dit werkt op dezelfde manier als het aanmaken van content, met het verschil dat hier al content aanwezig is in de verschillende velden.



The screenshot shows the 'Events Overview' page in the NMD CMS. The sidebar on the left is identical to the previous screenshot. The main content area shows an existing event entry. The event has a title 'Exchange knowledge for future innovation' and a body text block. The body text is a long paragraph about a project called 'EKFF' (Exchange Knowledge for Future Innovation) from Erasmus University. It mentions the project's goal, partners, and duration. At the bottom of the event card is a grey 'Submit' button.

# 6. Deploy

## DEPLOYMENT GUIDE

---

Het deployment process gaat als volgt.

Om te beginnen moet de klant / user het bestand downloaden.

Daarna moet deze persoon 2 .env files aanmaken

1 zal zich bevinden in de hoofdmap en 1 zal zich bevinden in de client map

De .env file in de hoofmap heeft volgende inhoud:

```
NMD_BASELINE='Like Graphics Love Code'  
NODE_ENV=Development  
NODE_SERVER_HOSTNAME=[ Give your hostname ex. 127.0.0.1]  
NODE_SERVER_PORT= { Give your port number example:8080 }  
MONGODB_CONNECTION= { Put in your MongoDB connection string }  
SKIP_PREFLIGHT_CHECK=true  
// login true github  
AUTH_GITHUB_CLIENT_ID= { your id }  
AUTH_GITHUB_CLIENT_SECRET={your secret key }  
AUTH_FACEBOOK_CLIENT_ID={ your Facebook Client id }  
AUTH_FACEBOOK_CLIENT_SECRET={ your Facebook Client secret }
```

Voor onze tweede .env file moeten we naar de src/client folder gaan.

Daar maken we opnieuw een .env file aan en deze geven we volgende inhoud.

```
SKIP_PREFLIGHT_CHECK=true  
SASS_PATH=./node_modules;./src/app
```

Nadat deze is aangemaakt. Gaan we terug naar de hoofdmap.

In deze map moeten we yarn installeren als je yarn nog niet globaal hebt

geinstalleerd zal je dit met brew moeten doen. op mac met volgend command.

[brew install yarn](#)

Via Windows kan je gewoon volgende comand gebruiken ( als npm globaal is geinstalleerd)

[npm install yarn](#)

Nadat dit is gedaan kan yarn worden geinstalleerd worden via

[yarn install](#)

Als we yarn hebben geinstalleerd gaan we naar onze src/client folder hier moeten we twee commando's uitvoeren.

[npm install](#)

[yarn install](#)

Als dit is gedaan kunnen we builden met onderstaande command.

[yarn build:all](#)

Nu gaan we onze server zowel op client niveau als op het hoofdniveau ( in de hoofdmap). Open twee terminals en voer volgende commands uit.

[terminal 1: yarn start:all](#)

[terminal 2: yarn client:start](#)

Nu zijn beide servers actief in terminal 1 zal u op de getoonde link moeten duwen ( deze word generate uit de .env file in de hoofdmap). Terminal 2 zal automatisch een dev server openen.

Beide servers moeten actief zijn om via de development server onze api aan te spreken.

Nadat onze servers opgestart zijn kan u beginnen met het aanpassen van de site. Alle aanpassingen worden direct getoond op de development server ( opgestart in terminal 2).

Als u deze aanpassingen wilt doorvoeren naar de online staande server ( uit terminal 1) voert u volgend command uit in terminal 1. of een optionele derde terminal.

[yarn client:build](#)

Nadat dit is uitgevoerd zullen de aanpassingen zichtbaar zijn in de online server uit terminal 1.



