Projectvoorstel eindwerk

# Fc Binkom

## Opdracht:

We gaan voor onze klant (een amateurvoetbal club) een nieuwe (moderne) site maken. Momenteel hebben zij een zeer ouderwetse site: <http://www.fcbinkom.be/>.

Op de site zullen een aantal verschillende dingen moeten staan:

* Home page met mededelingen
* Info over de club/bestuur/historiek/reglement/(pagina met bestanden?)
* Spelers en staf (naam/datum/foto)
* Voor zowel de eerste ploeg als de reserven een rangschikking en planning
* Sponsors (afbeeldingen)
* Evenementen (Binkom kermis/kermis bbq/Mosseldagen/Scampiavond/steakdagen/koers)
* Gastenboek

De bedoeling is dat de club zelf geen dingen meer moet aanpassen in de broncode/databank. Er zal dus een soort van CM (content management) moeten zijn. (lager beschreven wat allemaal)

Zoals je op de huidige pagina kan zien, maakt de club regelmatig gebruik van een pdf om iets op de site te zetten. Er moet dus ook gezorgd worden dat dit er deftig uit ziet.

Tot een tijdje geleden hadden ze reeds een gastenboek op hun pagina staan. Maar deze is eraf gehaald doordat iedereen onder eender welke naam er dingen op konden plaatsen naar eigen believen. Hier moet dus een controle op komen => registreren/aanmelden met facebook (of mail, indien mail moet iemand van het bestuur kunnen bevestigen dat ze dingen mogen posten). Zelfs wanneer een persoon geregistreerd zou deze niet alles mogen posten (daarmee de extra => ai controle scheldwoorden).

Indien er toch iets ongewenst gepost is, moet een bestuurslid dit ook kunnen verwijderen.

Dit betekent dat er dus ook een rollen systeem moet komen met (minstens) 3 verschillende rollen.

Rol 1: gebruiker => Mag posts plaatsen in het gastenboek  
Rol 2: bestuurslid => Mag gebruikers aanvaarden & dingen verwijderen uit gastenboek (+bovenstaand)  
Rol 3: paginabeheerder => Kan CM gebruiken om pagina aan te passen (+bovenstaande)

### CM:

* Mededelingen: Toevoegen, verwijderen, kiezen volgorde van mededelingen op home page. Een mededeling zou dus een titel moeten hebben, mogelijks een subtitel en de inhoud ervan moet dan geschreven kunnen worden met wysiwyg. (De auteur moet ook bijgehouden worden, maar niet getoond aan gewoon gebruikers, zodat andere bestuursleden achteraf kunnen zien wie wat heeft geplaatst)
* Alle gedeeltes onder info van club zou ook moeten aangepast kunnen worden. Implementatie nog te beslissen, bijvoorbeeld historiek één groot tekst bestand dat aanpasbaar is met wysiwyg, etc.
* Spelers & staf toevoegen/verwijderen => naam, (geboortedatum), foto uploaden indien geen foto => default gebruiken. (bij staf ook nog een gedeelte wat ze precies zijn, bijvoorbeeld: ‘trainer eerste’, ‘trainer tweede’, ‘keeperstrainer’)
* Sponsors => afbeeldingen toevoegen + mogelijks aan kunnen geven of het grote foto / kleine foto is zodat de styling zo afgehandeld kan worden en geen zooitje is na een tijdje.
* Evenementen => Toevoegen/Verwijderen hoe tonen => momenteel elk een eigen pagina. Herhaling: ze maken regelmatig gebruik van een pdf, deze moet dus zeker deftig getoond worden.

### Rangschikking/planning:

Dit gedeelte moet niet perse aanpasbaar zijn door de bestuursleden/paginabeheerder. Er gaat voor gezorgd moeten worden dat de gegevens opgehaald kunnen worden uit onze databank. Deze data gaat herhaaldelijk moeten geupdate worden, bijvoorbeeld elke week maandag of bv elke middag. Deze data gaat dan geupdate worden adhv een api, een python script of web scraping.

Het makkelijkste zou zijn gebruik makend van een api, voetbal vlaanderen heeft er geen ter beschikking. Dus we zouden mogelijks gebruik kunnen maken van een api van foot24.be, momenteel nog niet helemaal geregeld, wel reeds contact met webbeheerder van hun.  
Indien dit niet geregeld kan worden met foot24 zullen we moeten overgaan naar het schrijven van een scriptje waarmee je naar de site (<https://www.rbfa.be/nl/competities/downloads>) gaat, zip download en dan uitpakt. Daarmee krijg je een csv bestand waarin je dan kan zoeken naar fc Binkom. De data die hieruit komt, kan je dan in je databank steken om op te vragen.  
Indien dat ook niet geregeld kan worden, zullen we moeten over gaan naar webscraping om zo de data te kunnen opslaan in databank.

## Technologieën:

### Front end (links down below):

* Next.js (react) => typescript
* Webpack
* OAuth of Auth0
* Wysiwyg: react-draft-wysiwyg
* Lazy load images

### Ertussen:

* Gewoon rest? (=> anders eens kijken naar GraphQL mogelijks duidelijker/makkelijker)

### Back end:

* Java spring boot api

### Database:

* PostgreSQL (=> indien we rest api gebruiken en niet GraphQL => MongoDB ipv PostgreSQL)

## Extra’s:

* AI grove taal voor gastboek
* Mobile app (Android) (Met mededelingen, gastenboek en rangschikking/planning)

# Links:

* Next.js => <https://nextjs.org/docs/basic-features/typescript>
* Next.js & bootstrap => <https://dev.to/theallegrarr/add-bootstrap-to-next-js-app-35jp>
* Webpack => check if necessary and how to implement
* OAuth => <https://oauth.net/2/>
* Auth0 => <https://auth0.com/>
* Wysiwyg => <https://www.npmjs.com/package/react-draft-wysiwyg>
* Wysiwyg typescript => <https://www.npmjs.com/package/@types/react-draft-wysiwyg>
* Lazy load images => <https://www.npmjs.com/package/react-lazy-load-image-component>
* Lazy load images typescript => <https://www.npmjs.com/package/@types/react-lazy-load-image-component>
* MongoDB => <https://www.mongodb.com/2>
* MongoDB => <https://docs.spring.io/spring-data/mongodb/docs/current/reference/html/#referenc>

## Recap keuzes:

next.js => server side pre rendering + client side rendering (onderliggend react)

client side rendering => mogelijks onveiliger (mogelijks api key lekken, access token onderscheppen etc.) & seo problemen

typescript => makkelijker ondersteuning lane termijn (minder fouten, failproof => IDE meer ondersteuning)

=> npm package importeren @types npm package? Anders zelf typings maken => anders problemen bij importeren

Webpack => bundler gebruiken voor typescript naar js (compiler & bundler)

oauth of auth0 => link leggen via account id of email want access token blijft veranderen, spring boot security koppelen aan auth

<https://www.detroitlabs.com/blog/2018/02/28/adding-custom-type-definitions-to-a-third-party-library/>

<https://www.npmjs.com/package/@types/jquery>

wysiwyg => <https://www.npmjs.com/package/react-draft-wysiwyg>

=> <https://www.npmjs.com/package/@types/react-draft-wysiwyg>

Lazy load images

=> <https://www.npmjs.com/package/react-lazy-load-image-component>

=> <https://www.npmjs.com/package/@types/react-lazy-load-image-component>

## OAuth vs Auth0:

**OAuth** 2.0 is a protocol that allows a user to grant limited access to their resources on one site, to another site, without having to expose their credentials.   
**Auth0** is an organisation, who manages Universal Identity Platform for web, mobile and IoT can handle any of them