Onderwijsgroep Professionele Opleidingen

Handelswetenschappen en bedrijfskunde

|  |
| --- |
|  |
| **Thomas Wyns**  **Eilish Van Der Snickt** |
| Derde bachelor - Toegepaste Informatica  Choose an item. |
|  |
|  |
| **Mobile & Internet 5**  Steven Ophalvens |
|  |
| Academiejaar 2019-2020 |

|  |  |
| --- | --- |
| Documentatie project | |
| AniME |

# Inhoudsopgave

Inhoudsopgave 3

Documentatie AniME 4

Project omschrijving 4

Over AniME 4

Functionaliteit 4

Project structuur 4

React componenten 4

Databank 4

Applicatie flow 4

# Documentatie AniME

## Project omschrijving

### Over AniME

Kijk je graag anime of zoek je goede anime om te leren kennen? AniME geeft je de mogelijkheid om de populairste anime van het moment in detail te leren kennen. Ook kan je je anime kennis testen doormiddel van een quiz.

Ook kan je andere anime opzoeken via onze app en hiervan de details bekijken

(=> nog niet goed dit moet de gebruiker warm maken om de app te installeren)

### Functionaliteit

AniME is een progressive web app (PWA), dit wil zeggen dat je AniME kan installeren via de browser op het startscherm van je mobiele toestellen.

AniME staat je toe alle anime op te zoeken via een zoekmachine en haalt alle data van de gekozen anime op via de kitsu.io API. Zo kan je de anime details in een bepaald formaat bekijken. Verder worden standaard de 10 meest populaire anime van het moment getoond. Doordat deze anime lokaal opgeslagen worden, zijn deze ook beschikbaar wanneer je geen internetconnectie ter beschikking hebt.

Doordat er info lokaal wordt opgeslagen is de applicatie altijd bruikbaar in hoeverre de applicatie offline gebruik toelaat. Dit wil zeggen dat je zonder internetconnectie wel de populairste anime te zien krijgt maar bijvoorbeeld geen toegang hebt tot de specifieke details van deze anime.

Ook zijn er quizzen over de meest populaire anime ter beschikbaar. De gebruiker kan zo op een speelse manier anime leren kennen of de kennis testen die de gebruiker heeft over de meest populaire anime.

## Project structuur

Het project is gemaakt doormiddel van het de javascriptbibliotheek React. React zorgt ervoor dat javascript projecten op een efficiëntere manier kunnen worden gecreëerd.

React werkt met componenten, dit zorgt ervoor dat je voor specifieke zaken een apart javascript bestand hebt. En je bijvoorbeeld meerdere javascript bestanden kunt hebben die samen op een pagina terecht komen in je uiteindelijke webapplicatie.

React heeft ook een eigen routeringsmethode en verschillende andere functionaliteiten die het simpel maken sommige zaken toe te voegen aan je project. Bijvoorbeeld een offline/online component die zelf checkt of je applicatie wordt uitgevoerd met of zonder internetconnectie.

Eerst en vooral hebben we het homescherm van de applicatie gemaakt. Op dit scherm worden de populairste anime van het moment opgehaald uit de API en getoond op dit scherm. Deze info werd dan daarna weggeschreven naar de Firebase Firestore.

Daarna hebben we doormiddel van de data in de Firestore de verschillende quizzen ontworpen en uitgewerkt. Het puntensysteem van deze quizzen wordt lokaal bijgehouden en aangezien we nog geen accountbeheer toepassing hebben toegevoegd aan dit project is het ook nog niet nodig om de behaalde scores bij te houden van de verschillende spelers.

Daarna is de detail pagina van elke anime uitgewerkt. Hierop kunnen gebruikers zien welke score de anime krijgt, waarover de anime gaat, welke personages eraan meedoen, … .

Daarna werd de applicatie helemaal PWA vriendelijk gemaakt. Doormiddel van het gebruiken van lokale opslagmogelijkheden, ervoor zorgen dat de applicatie offline kan gebruikt worden, het omzetten van HTTP verkeer naar HTTPS verkeer, … .

## React componenten

Er zijn een heleboel React componenten gebruikt in dit project. Elk component heeft zijn eigen functies. Ze zijn zo opgebouwd dat er zo min mogelijk componenten meerdere functionaliteiten bevatten die los staan van elkaar.

**De meest speciale componenten zijn de App.js, FirebaseInit.js, AnimeList.js en de DataToFirebase.js componenten.**

De **App.js component** bevat eigenlijk alle routering die zich in het pro-ject bevindt. Het is erg belangrijk voor de overzichtelijkheid dat er 1 component alle routering bevat en dat dit niet verspreid zit over veschillende componenten.

De **FirebaseInit.js component** is eigenlijk een component die Firebase inlaadt in het project en zo kunnen alle Firebase functies worden gebruikt in het hele project door het aanroepen van deze component.

De **AnimeList.js component** is eigenlijk het hard van de applicatie hierop staat de code die de benodigde data ophaald uit de API en andere componenten aanstuurt wat er met deze data moet gebeuren. Bijvoorbeeld de AnimePoster.js component wordt aangestuurd vanuit deze component om doormiddel van de opgehaalde data alle posters op het homescherm te zetten.

En als laatste de **DataToFirebase.js component** is ook een component die is aangestuurd vanuit AnimeList.js om alle opgehaalde data in de Firestore op te slaan doormiddel van te kijken of er al data bevindt in deze Firestore en welke moet worden aangepast of toegevoegd vergeleken met de reeds opgeslagen data.

## Databank

De databank is gemaakt in Firebase met de Cloud Firestore. Dit is een NoSQL databank. Dit wil zeggen dat er geen ERD kan van gemaakt worden. We hebben echter wel structuur in onze databank gestoken.

Eerst en vooral hebben we verschillende collecties gemaakt, wat je een beetje kunt vergelijken met tabellen. De collecties bevatten verschillende documenten. Documenten kan je wat vergelijken met rijen en elk document bevat verschillende velden wat je kan vergelijken met de kolommen. Elk document ingevuld met data kan je vergelijken met 1 record.

We hebben 4 verschillende collecties:

1. houdt alle info van de populairste anime bij;

2. houdt alle titels van de anime bij;

3. houdt alle korte inhouden van de anime bij;

4. houdt alle posters van de anime bij.

Deze laatste 3 collecties worden vooram gebruikt voor de data gemakkelijker op te halen voor de quizzen.

## Applicatie flow