

# Technisch ontwerp

- Tech Stack
  - Framework
  - Packages
  - Versie beheer
  - Database
- Architectuur
  - Frontend
  - Backend
  - Database-structuur
  - API-ontwerp
    - REST API Endpoints
  - Authenticatie en Autorisatie
    - Werknemer
    - Manager
    - CEO
- Test Strategie
- Beveiliging
  - Veiligheidsmaatregelen
  - Logging en Monitoring
- Documentatie en code conventie

# Tech Stack

---

De volgende frameworks, libraries en software applicaties worden gebruikt.

## Framework

Het framework wat gebruikt gaat worden is Next.js. Next.js is een web compleet framework, wat gebaseerd is op React. Het heeft een makkelijk en snel ontwikkelproces.

Next.js kan geschreven worden in JavaScript of TypeScript. Voor dit project is TypeScript gekozen, vanwege het feit dat hier minder foutgevoelige code mee wordt geschreven.

## Packages

De packages worden gemanaged door NPM.

De packages die gebruikt worden zijn:

- Shadcn voor UI-componenten
- Tailwind voor CSS

## Versie beheer

GIT wordt gebruikt voor versiebeheer en deployment. De GIT-repo wordt op GitHub gehost.

## Database

De database die gebruikt wordt is Supabase. Voor nu is het nog niet duidelijk of deze zelf wordt gehost (lokaal) of op de servers van Supabase.

# Architectuur

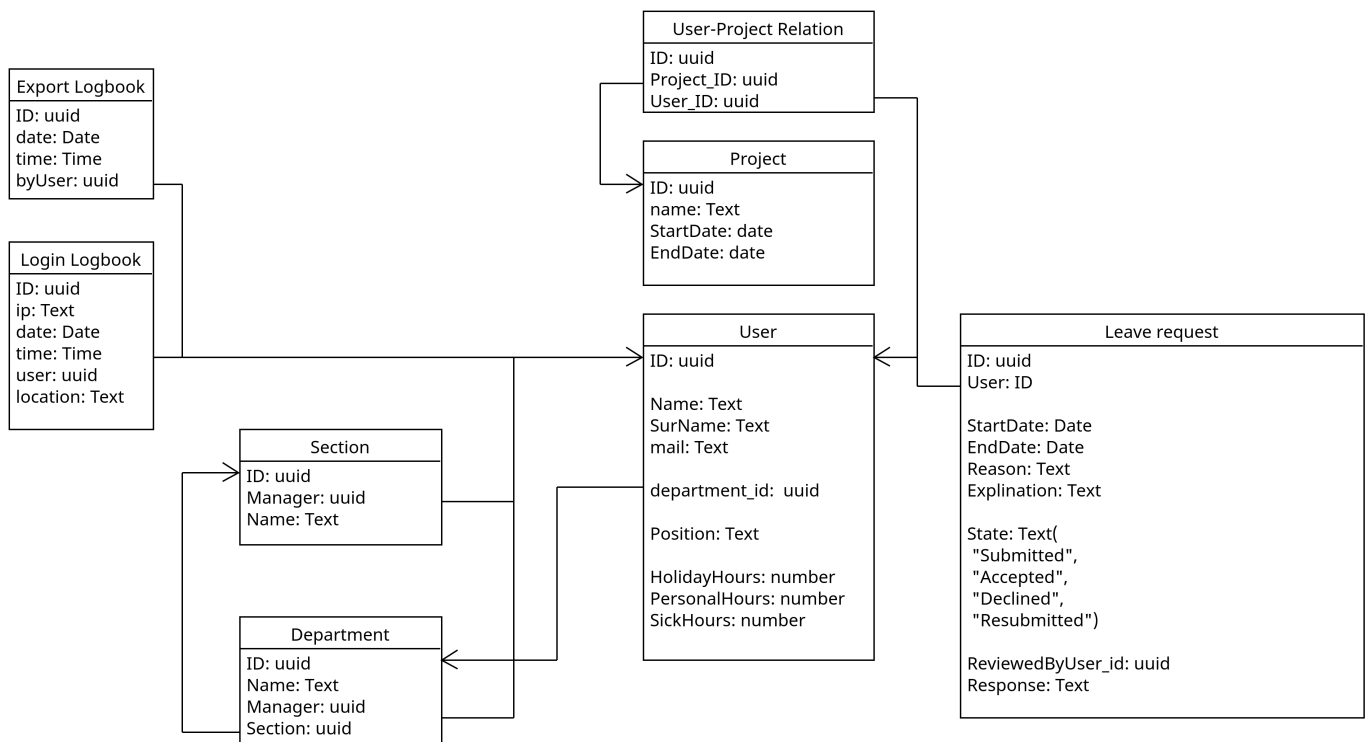
## Frontend

De front-end gebruikt shadcn als component library. NEXT maakt gebruik van server-side rendering (SSR) en client-side rendering (CSR).

## Backend

De backend bestaat uit Next.js API-routes, gecombineerd met Supabase als database-backend en authenticatiesysteem. Supabase biedt ingebouwde API's voor CRUD-operaties. Voor het ophalen van data zullen de ingebouwde API's van supabase gebruikt worden. Het aanroepen van functies zal via NEXT apis gaan.

## Database-structuur



# API-ontwerp

## REST API Endpoints

<b>URL</b>	<b>/api/leave-request?reason=TEXT&amp;explanation=TEXT&amp;start-date=DATE&amp;end-date=DATE</b>
<b>Method</b>	POST
<b>Uitleg</b>	Voor het indienen van een verlofaanvraag.
<b>Toegang</b>	Alle ingelogde gebruikers
<b>Return</b>	ActionStatus

<b>URL</b>	<b>/api/leave-requests</b>
<b>Method</b>	GET
<b>Uitleg</b>	Voor het ophalen van alle verlofaanvragen van een gebruiker en/of department
<b>Toegang</b>	Alle ingelogde gebruikers
<b>Return</b>	array met leave_requests

**Parameters:**

*?user=ID*: De geselecteerde gebruiker, standaard is dit de huidig ingelogde gebruiker.

*?department=ID* De geselecteerde department, geeft nu alle leave\_requests binnen een department terug.\*\*

*?sectie=ID* De geselecteerde sectie, geeft nu alle leave\_requests binnen een sectie terug.\*\*

<b>URL</b>	<b>/api/leave-requests/ID/accept</b>
<b>Method</b>	PUT
<b>Uitleg</b>	Voor het accepteren van een leave request.
<b>Toegang</b>	Alle managers welke de gebruiker in een van zijn secties of departments heeft.
<b>Return</b>	array met leave requests

<b>URL</b>	<b>/api/leave-requests/ID/decline?response=TEXT</b>
<b>Method</b>	PUT
<b>Uitleg</b>	Voor het afwijzen van een leave request.
<b>Toegang</b>	Alle managers welke de gebruiker in een van zijn secties of departments heeft.
<b>Return</b>	array met leave requests

<b>URL</b>	<b>/api/users</b>
<b>Method</b>	GET
<b>Uitleg</b>	Voor het ontvangen van alle user data.
<b>Toegang</b>	Alle ingelogde gebruikers.
<b>Return</b>	array met leave requests

---

<b>URL</b>	<b>/api/users/me</b>
<b>Method</b>	GET
<b>Uitleg</b>	Voor het ontvangen van alle beveiligde data van de ingelogde gebruiker.
<b>Toegang</b>	Alle ingelogde gebruikers.
<b>Return</b>	array met leave requests

---

<b>URL</b>	<b>/api/calender/YEAR/MONTH</b>
<b>Method</b>	GET
<b>Uitleg</b>	Voor het ontvangen van alle leave_requests van de gegeven maand.
<b>Toegang</b>	Alle ingelogde gebruikers. *
<b>Return</b>	array met leave requests.

De leave\_requests zijn niet volledig gevuld gebaseerd op de rechten van de gebruiker, zo krijgt een werknemer alleen te zien wie er binnen zijn department afwezig zijn, maar niet reden.

---

# Authenticatie en Autorisatie

De applicatie maakt gebruik van Supabase-authenticatie voor inlog- en toegangsbeheer. Toegang tot de API wordt verleend op basis van rollen en ownership. De rollen zijn als volgt:

## Werknemer

Een werknemer kan:

- zijn eigen gebruikers data inzien via *GET /api/users/me*
- Leave aanvragen via *PUT /api/leave\_request*
- Afwezigheid zijn via *GET /api/calendar/MONTH*

## Manager

Een manager kan het volgende:

- Alle leave\_requests ophalen binnen zijn/haar bevoegdheid via: *GET /api/leave-requests*
- leave\_requests accepteren binnen zijn/haar bevoegdheid via: *PUT /api/leave-requests/ID/accept*
- leave\_requests afwijzen binnen zijn/haar bevoegdheid via: *PUT /api/leave-requests/ID/decline*
- Afwezigheid zijn, en de redenen hiervan via: *GET /api/calendar/MONTH*
- De gebruikers data van alle gebruikers binnen zijn/haar bevoegdheid via: *GET /api/users*

Een manager kan ook alles wat een werknemer kan.

## CEO

Een CEO kan het volgende:

- Het exporteren van alle afwezigheid binnen periode naar .CSV via: *GET /api/export*

# Test Strategie

---

Testen zal plaatsvinden op zowel frontend als backend. Supabase wordt niet getest, aangezien het een beheerde service is.

Er wordt gebruik gemaakt van:

- **Jest** voor unit testing
- **Cypress** voor end-to-end testing

# Beveiliging

---

## Veiligheidsmaatregelen

Bescherming tegen veelvoorkomende kwetsbaarheden zoals SQL-injecties, Cross-Site Scripting (XSS), en Cross-Site Request Forgery (CSRF).

SQL-Injectie is niet mogelijk, de supabase API is hier namelijk voor beschermd.

Voor cross site scripting zullen we elk request "schoonmaken" door middel van de js package DOMPurify.

Voor Cross-site request forgery zullen wij de js package csrfProtection gebruiken.

## Logging en Monitoring

Bepaalde acties van de applicatie worden gelogd. Hierin wordt datum en tijd van de actie gelogd, de gebruiker welke deze actie uitvoert en de actie zelf. Deze logs zullen verborgen zijn voor de gebruiker, en kunnen opgevraagd worden wanneer nodig.



# Documentatie en code conventie

---

Documentatie van de code zal plaatsvinden in de code. Er zal gebruik gemaakt worden van een Doxygen-documentatievorm.

```
class Foo {  
    /**  
     * @brief bar function of foo  
     *  
     * @todo nog geen implementatie  
     * @return void  
     */  
    public Bar() {  
    }  
}
```