



Prácticas LHS

Documento de realización

Prácticas 2023-2024

Tymo Verhaegen 3APP02

Año académico 2022-2024

Campus Geel, Kleinhoefstraat 4, BE-2440 Geel

Índice

1 Gracias	5
2 Terminología	6
3 Asignación	7
3.1 Empresa	7
3.2 Equipo	7
3.3 Descripción del proyecto	7
4 Tecnologías y metodologías utilizadas	9
4.1 Tecnologías	9
4.1.1 Django / Python	9
4.1.2 HTML y Tailwind CSS	9
4.1.3 PostgreSQL	9
4.1.4 Javascript	9
4.1.5 Docker	9
4.2 Programas / Plataformas	10
4.2.1 Figma	10
4.2.2 UML (estrella)	10
4.2.3 Github	10
4.3 Metodologías	10
4.3.1 SCRUM	10
4.4 Tuberías	11
4.4.1 IC/CD	11
5 Realizaciones	12
5.1 Análisis	12
5.1.1 Casos prácticos	12
5.1.1.1 Descripciones de casos de uso (cliente)	13
5.1.1.2 Descripciones de casos de uso (personal)	16
5.1.1.3 Descripciones de casos de uso (empresa)	18
5.1.1.4 Descripciones de casos de uso (herramienta).	20
5.1.2 Diseño de bases de datos	21
5.1.3 Mapa del sitio	22
5.1.4 Prototipos	23
5.1.4.1 Calendario (vista semanal)	23
5.1.4.2 Calendario (vista por meses)	23
5.1.4.3 Nombramiento (detalles)	24
5.1.4.4 Estadísticas y registros	24
5.1.4.5 Estadísticas y registros (detalles)	25
5.1.4.6 Contactos	25
5.1.4.7 Contactos (crear)	26
5.1.4.8 Contactos (detalles)	26
5.1.4.9 Recursos	27
5.1.4.10 Recursos (detalles)	27
5.1.4.11 Recursos (crear sucursal)	28
5.1.4.12 Configuración de la empresa	28
5.1.4.13 Configuración de la empresa (detalles)	29

	A4	
5.1.4.14	Códigos promocionales	29
5.1.4.15	Códigos promocionales (socios)	30
5.1.4.16	Códigos de promoción (detalles)	30
5.1.4.17	Reservas	31
5.1.4.18	Reserva (cont.).	31
5.2	Front-end	32
5.2.1	Navegación	32
5.2.2	Inicio de sesión	33
5.2.3	Orden del día	33
5.2.4	Estadísticas y registros	34
5.2.5	Clientes	35
5.2.6	Empleados	36
5.2.7	Servicios y paquetes	38
5.2.8	Recursos	40
5.2.9	Promociones y ofertas	41
5.2.10	Reseñas	42
5.2.11	Configuración de la empresa	42
5.2.12	Reserva (usuario final)	43
5.3	Fondo	44
5.3.1	Navegación	44
5.3.2	Inicio de sesión	45
5.3.3	Orden del día	45
5.3.4	Estadísticas y registros	47
5.3.5	Clientes, empleados y socios	47
5.3.6	Revisiones	50
5.3.7	Servicios y paquetes	51
5.3.8	Recursos	54
5.3.9	Promociones y ofertas	57
5.3.10	Configuración de la empresa	58
5.3.11	Reservas (usuario final)	59
6	Conclusiones	60

1 GRACIAS

En este sentido, me gustaría dedicar un momento a dar las gracias a LHS by Spot Group por darme la oportunidad de hacer unas prácticas en la empresa. Esto no habría sido posible sin Steven. He aprendido mucho durante estas prácticas y he disfrutado mucho.

También me gustaría dar las gracias a mis colegas y compañeros de estudios, sobre todo a Thomas Verbruggen y Joppe Kerkhofs, que fueron mis compañeros de grupo en el proyecto y con los que disfruté trabajar.

Por supuesto, no hay que olvidar a mi mentor de prácticas, Patrick Dielens. Era un supervisor muy agradable y bueno que siempre estaba dispuesto cuando le necesitábamos.

2 TERMINOLOGÍA

PLAZO	RELACIONADO
Marco	Plataforma o estructura para construir software que proporciona herramientas y convenciones para simplificar el desarrollo.
Lenguaje de formato	Lenguaje que define la estructura de los documentos.
Respuesta	Un enfoque de diseño que garantiza el correcto funcionamiento y visualización del software en distintos dispositivos.
Tecnología de contenedores	Un método para empaquetar y ejecutar aplicaciones utilizando contenedores para una ejecución consistente en diferentes entornos.
UML	Lenguaje Unificado de Modelado: Lenguaje de modelado estandarizado para la visualización, el diseño y la documentación de proyectos de software.
CI/CD	Integración continua / Despliegue continuo: Práctica para automatizar procesos como las pruebas y el despliegue de código.
EAE/SEO	<ul style="list-style-type: none"> - Publicidad en buscadores Se trata de anuncios de pago en motores de búsqueda como Google Ads. - Optimización de motores de búsqueda Optimización de contenidos web para mejorar la posición en los resultados de búsqueda orgánicos (no de pago)
Casos prácticos	Describen cómo interactuarán los usuarios con un sistema o producto. Esbozan los pasos que puede dar un usuario y las posibles reacciones ante ellos.
Mapa del sitio	Representación visual de la estructura y jerarquía de las páginas de una aplicación o sitio web. Muestra la relación entre páginas.
Wireframes	Los wireframes son esquemas de un sitio web o una aplicación, centrados principalmente en el diseño y la funcionalidad.
Prototipos	Los prototipos son representaciones interactivas del aspecto y el funcionamiento del producto final.
MVC	<p>Modelo Vista Controlador es un patrón de diseño que divide el diseño de aplicaciones complejas en tres unidades con diferentes responsabilidades</p> <ul style="list-style-type: none"> - Modelo: modelo de datos - Vista: presentación de datos - Controlador: lógica de aplicación

3 ASIGNACIÓN

3.1 Empresa

LHS es una pequeña empresa especializada en la creación de sitios web y aplicaciones web, desde 2023 LHS forma parte de Spot Group. Spot Group es una empresa más grande y con más capacidad. Como resultado, LHS es ahora parte de una empresa que puede ofrecer más servicios a mayor escala, estos son servicios como SEA/SEO, hosting, marketing y mucho más.

3.2 Equipo

El equipo con el que desarrollé este proyecto estaba formado por Thomas Verbruggen, Joppe Kerkhofs y yo. Los tres somos estudiantes de desarrollo de aplicaciones en Thomas More, en Geel.

Este no es el primer proyecto que desarrollamos juntos, de hecho ya hemos creado proyectos escolares juntos, lo que ha supuesto ventajas para el proyecto de prácticas. Esto se debe a que conocemos los puntos fuertes y débiles de cada uno y podemos ayudarnos cuando sea necesario.

3.3 Descripción del proyecto

Una herramienta de reservas compleja y un sistema de gestión modular con una amplia gama de funcionalidades, entre las que se incluyen:

- **Sistema de reservas**

El aspecto central de la aplicación web, permite a los clientes concertar fácilmente citas, reservas o reservas en función de la disponibilidad.

- **Gestión de recursos**

El sistema incluirá funciones de gestión de activos, como salas, equipos y personal,..... Esto incluye el seguimiento de la disponibilidad de recursos, el personal asignaciones,

- **Integración del calendario**

Se ofrecerá la integración de un calendario, lo que permitirá tanto a los clientes como a los empleados ver, cambiar y hacer un seguimiento de las citas con facilidad.

- **Servicios y paquetes**

El sistema incluirá funciones de gestión de servicios y paquetes, para que los clientes puedan elegir fácilmente la opción adecuada a sus necesidades.

- **Sistema de pago**

Se implantará un sistema de pago seguro para permitir los pagos en línea de las reservas. Se implantará para ahorrar tiempo y realizar pagos por adelantado.

- **Gestión de clientes**

Se mantendrá algún tipo de base de datos de clientes, con información del cliente, datos personales, historial de reservas y preferencias.

- **Gestión de ofertas y promociones**

El sistema proporcionará la funcionalidad necesaria para gestionar y aplicar ofertas, promociones y descuentos a las reservas de los clientes.

- **Comunicación y notificaciones**

Los clientes reciben notificaciones automáticas por correo electrónico cuando hacen una reserva o cuando se producen cambios en ella.

- **Estadísticas y registros**

Los empleados podrán ver estadísticas como el número de reservas, el número de usuarios, etc. También podrán ver un historial detallado de todos los clientes y su historial de reservas.

- **Cronjobs**

El sistema realizará tareas y procesos automatizados, como la optimización de calendarios y reservas.

- ...

4 TECNOLOGÍAS

UTILIZADAS Y

METODOLOGÍAS

4.1 Tecnologías

4.1.1 Django / Python

Django es un framework web de alto nivel basado en Python y de código abierto que fomenta el desarrollo rápido y un diseño ajustado y pragmático. Django utiliza el principio MVC.

Razón: Django era la opción preferida de LHS para este proyecto.

4.1.2 HTML y CSS de Tailwind

HTML es un lenguaje de marcado que se utiliza para definir la estructura de las páginas web. Tailwind es un marco de trabajo que se utiliza para dar estilo a las páginas web y crear de forma rápida y eficaz interfaces de usuario personalizadas y con capacidad de respuesta.

Razón: HTML es obligatorio para crear una aplicación web, Tailwind era la preferencia personal del grupo. Gracias a su amplio uso para proyectos escolares, fue una elección fácil.

4.1.3 PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos relacionales (RDBMS) de código abierto utilizado para almacenar y gestionar datos.

Razón: PostgreSQL era la preferencia de LHS para este proyecto.

4.1.4 Javascript

Javascript es un lenguaje de programación utilizado para desarrollar páginas Web interactivas. Permite a los desarrolladores manipular el contenido de las páginas web.

Razón: Javascript es casi obligatorio para hacer una aplicación web que funcione.

4.1.5 Docker

Docker es una plataforma utilizada para desarrollar, enviar y ejecutar aplicaciones. Utiliza la tecnología de contenedorización para empaquetar aplicaciones con todo el código necesario.

Razón: Docker era una preferencia personal del grupo, de esa manera todos tendrían siempre los mismos datos de prueba y todo se ejecutaría un poco más fácil.

4.2 Programas / Plataformas

4.2.1 Figma

Figma es una herramienta de diseño utilizada para crear interfaces de usuario, wireframes, prototipos y otros elementos de diseño.

Razón: Figma era una preferencia personal del grupo para crear sitemaps, wireframes y prototipos. También es un estándar de la industria.

4.2.2 UML (estrella)

UML es un lenguaje de modelado estandarizado que se utiliza para visualizar, diseñar y documentar proyectos de software.

Razón: StarUML fue una preferencia personal del grupo para diseñar nuestra base de datos y casos de uso en, debido a su amplio uso durante los proyectos escolares.

4.2.3 Github

Github es una plataforma utilizada para el control de versiones y la colaboración durante el desarrollo de software. En Github, los desarrolladores pueden almacenar, editar y compartir su código con otros.

Razón: Github era obvio, era la preferencia de LHS y el grupo debido a su amplio uso durante los proyectos de la escuela y esto también es el estándar de la industria.

4.3 Metodologías

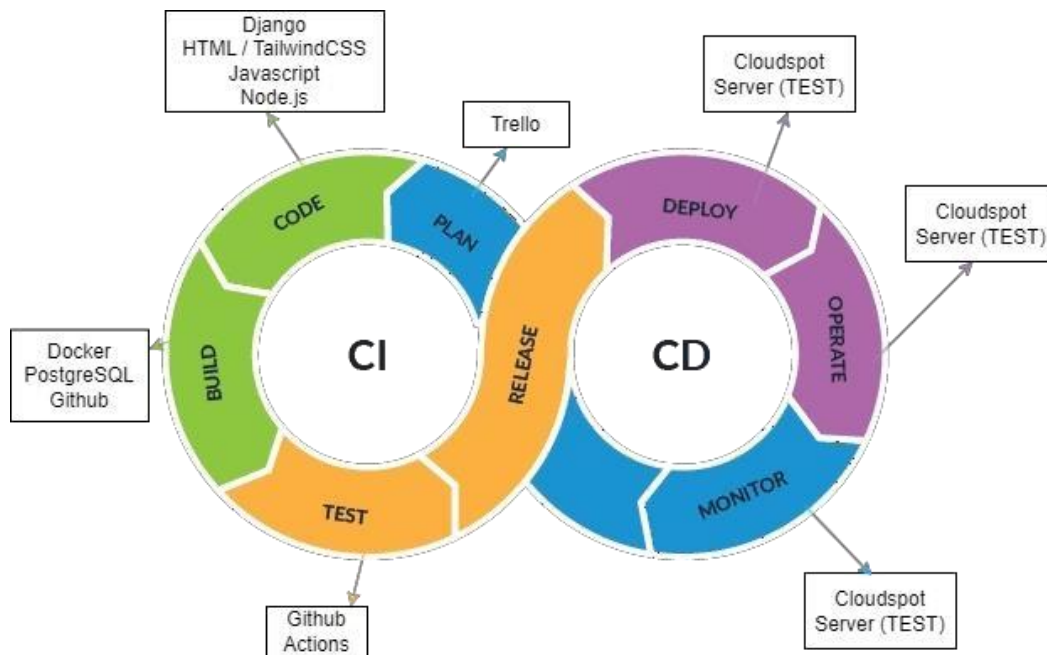
4.3.1 SCRUM

Metodología ágil que funciona a base de sprints, normalmente de dos a cuatro semanas. También incluye retrospectivas en las que el equipo echa la vista atrás al sprint anterior y se retroalimenta mutuamente sobre los aspectos positivos y negativos de ese sprint.

Razón: La metodología SCRUM fue el método de trabajo preferido por nuestro grupo, el uso extensivo de esta metodología para proyectos escolares hizo de esta una elección fácil.

4.4 Tuberías

4.4.1 CI/CD



5 REALIZACIONES

5.1 Análisis

El análisis del proyecto tuvo lugar en la semana 4. Durante esta fase, el equipo realizó un análisis en profundidad de las necesidades de los usuarios.

5.1.1 Casos prácticos

Se utilizó StarUML para crear los casos de uso



5.1.1.1 Descripciones de casos de uso (cliente)

- Caso práctico: Inicio de sesión
 - **Descripción:** El cliente puede iniciar sesión en el sistema introduciendo su correo electrónico y contraseña
 - **Condiciones posteriores:** El cliente ve la página de inicio desde la vista de cliente

- Caso práctico: Cierre de sesión
 - **Descripción:** El cliente puede salir del sistema haciendo clic en el botón de cierre de sesión.
 - **Condiciones:** El cliente ha iniciado sesión
 - **Condiciones posteriores:** Se cierra la sesión del cliente y se le envía a la pantalla de inicio de sesión

- Caso práctico: Concertación de citas
 - **Descripción:** El cliente puede concertar una cita seleccionando un servicio o paquete. A continuación, selecciona una hora y hace clic en confirmar. El cliente tendrá que pagar el 50% del importe por adelantado (*ver caso de uso pagar*)
 - **Condiciones:** El cliente ha iniciado sesión
 - **Condiciones posteriores:** El cliente es enviado a la página de inicio de la vista de cliente. Se envía un correo electrónico de confirmación al cliente
 - **Flujos alternativos:**
 - El personal está conectado
 - El funcionario puede concertar una cita a cualquier hora. Puede anular la propuesta del sistema

- Caso práctico: Edición de citas
 - **Descripción:** El cliente puede ajustar los servicios y la hora de su cita
 - **Condiciones:** El cliente ha iniciado sesión
 - **Condiciones posteriores:** La cita modificada se guarda en el sistema y se envía un correo electrónico de confirmación con los cambios al cliente

- Caso práctico: Pago
 - **Descripción:** El cliente es enviado al sistema de pago Stripe donde se completará el pago
 - **Condiciones:** El cliente ha iniciado sesión y tiene una cita válida en el sistema

- Caso práctico: uso de vales
 - **Descripción:** El cliente puede introducir un código de un vale o código de descuento durante el pago
 - **Condiciones:** El cliente ha iniciado sesión, tiene una cita válida en el sistema y está realizando el pago
 - **Condiciones posteriores:** El importe final tiene una reducción de precio

- Caso práctico: Dejar una opinión
 - **Descripción:** El cliente puede dejar una reseña después de su cita
 - **Condiciones:** El cliente ha iniciado sesión y tiene una cita concertada
 - **Condiciones posteriores:** La reseña se almacena en el sistema y puede utilizarse como "testimonio" en el sitio web de la empresa

- Caso práctico: compartir información médica
 - **Descripción:** El cliente puede introducir su información médica
 - **Condiciones:** El cliente ha iniciado sesión
 - **Condiciones posteriores:** La información médica del cliente se almacena en el sistema

- Caso práctico: Consultar lista de precios
 - **Descripción:** El cliente puede ver la lista de precios de la empresa

- Caso práctico: Consulta de horarios
 - **Descripción:** El cliente puede ver el horario de apertura del negocio

- Caso práctico: Dejar una opinión
 - **Descripción:** El cliente puede dejar una reseña después de su cita
 - **Condiciones:** El cliente ha iniciado sesión y tiene una cita concertada
 - **Condiciones posteriores:** La reseña se almacena en el sistema y puede utilizarse como "testimonio" en el sitio web de la empresa

5.1.1.2 Descripciones de casos de uso (personal).

El personal hereda todos los casos de uso del cliente

- Caso práctico: Gestionar las horas de trabajo
 - **Descripción:** El personal puede introducir sus horas de trabajo por día
 - **Requisitos previos:** El miembro del personal ha iniciado sesión
 - **Condiciones posteriores:** Las horas de trabajo se almacenan en el sistema

- Caso práctico: Consultar calendario
 - **Descripción:** El personal puede ver su calendario
 - **Requisitos previos:** El miembro de la prensa ha iniciado sesión

- Caso práctico: Gestión de pagos
 - **Descripción:** El personal puede registrar los pagos
 - **Requisitos previos:** El miembro del personal ha iniciado sesión

- Caso práctico: Gestión de facturas
 - **Descripción:** El personal puede crear una factura después de completar un tratamiento que se envía automáticamente al buzón del cliente
 - **Condiciones:** El funcionario ha iniciado sesión y el tratamiento ha finalizado
 - **Condiciones posteriores:** El cliente recibirá una factura en su buzón de correo electrónico

- Caso práctico: Consultar revisiones
 - **Descripción:** El personal puede ver las reseñas
 - **Requisitos previos:** El miembro del personal ha iniciado sesión

- Caso práctico: Consulta de información médica
 - **Descripción:** El personal puede consultar la información médica de un cliente
 - **Requisitos previos:** El miembro del personal ha iniciado sesión y el cliente ha cargado una ficha médica válida
 - **Condiciones posteriores:** Las horas de trabajo se almacenan en el sistema

- Caso práctico: Gestión de vales
 - **Descripción:** El personal puede crear, editar y eliminar comprobantes
 - **Requisitos previos:** El miembro del personal ha iniciado sesión
 - **Condiciones posteriores:** El cliente recibe las promociones en su buzón

- Caso práctico: Consulta de registros
 - **Descripción:** El personal puede ver los registros de todas las reservas
 - **Requisitos previos:** El miembro del personal ha iniciado sesión

5.1.1.3 Descripciones de casos de uso (empresa)

- Caso práctico: Gestión de personal
 - **Descripción:** La empresa puede crear, editar y eliminar personal
 - **Requisitos previos:** La empresa ha iniciado sesión
 - **Condiciones posteriores:** Las cuentas de personal se almacenan en el sistema. Después, el personal de la plataforma puede

- Caso práctico: Gestión de recursos
 - **Descripción:** La empresa puede crear, editar y eliminar recursos
 - **Requisitos previos:** La empresa ha iniciado sesión
 - **Condiciones posteriores:** Los recursos se almacenan en el sistema

- Caso práctico: Gestión de códigos promocionales
 - **Descripción:** La empresa puede crear, editar y eliminar códigos promocionales
 - **Requisitos previos:** La empresa ha iniciado sesión
 - **Condiciones posteriores:** Los códigos de promoción se almacenan en el sistema y pueden ser utilizados por el usuario final

- Caso práctico: Gestión de comités
 - **Descripción:** La empresa puede crear, editar y eliminar códigos de comisión
 - **Requisitos previos:** La empresa ha iniciado sesión y debe existir un interlocutor válido en el sistema al que se pueda vincular este código
 - **Condiciones posteriores:** Los códigos de comisión se almacenan en el sistema. Si un cliente utiliza estos códigos, el socio vinculado recibirá una parte del importe

- Caso práctico: gestión de servicios
 - **Descripción:** La empresa puede crear, editar y eliminar servicios
 - **Requisitos previos:** La empresa ha iniciado sesión
 - **Condiciones posteriores:** Los servicios se almacenan en el sistema. Las listas de precios de las empresas se actualizan automáticamente.

- Caso práctico: Gestión de paquetes
 - **Descripción:** La empresa puede crear, editar y eliminar paquetes.
 - **Condiciones:** La empresa ha iniciado sesión y deben existir servicios válidos en el sistema
 - **Condiciones posteriores:** Los paquetes se almacenan en el sistema

- Caso práctico: Gestión de los horarios de apertura
 - **Descripción:** La empresa puede crear o editar horarios de apertura
 - **Requisitos previos:** La empresa ha iniciado sesión
 - **Condiciones posteriores:** Los horarios de apertura se almacenan en el sistema

- Caso práctico: Gestión de una empresa
 - **Descripción:** La empresa puede modificar los datos de la empresa
 - **Requisitos previos:** La empresa ha iniciado sesión
 - **Condiciones posteriores:** La herramienta se adapta a las preferencias de la empresa

5.1.1.4 Descripciones de casos de uso (herramienta).

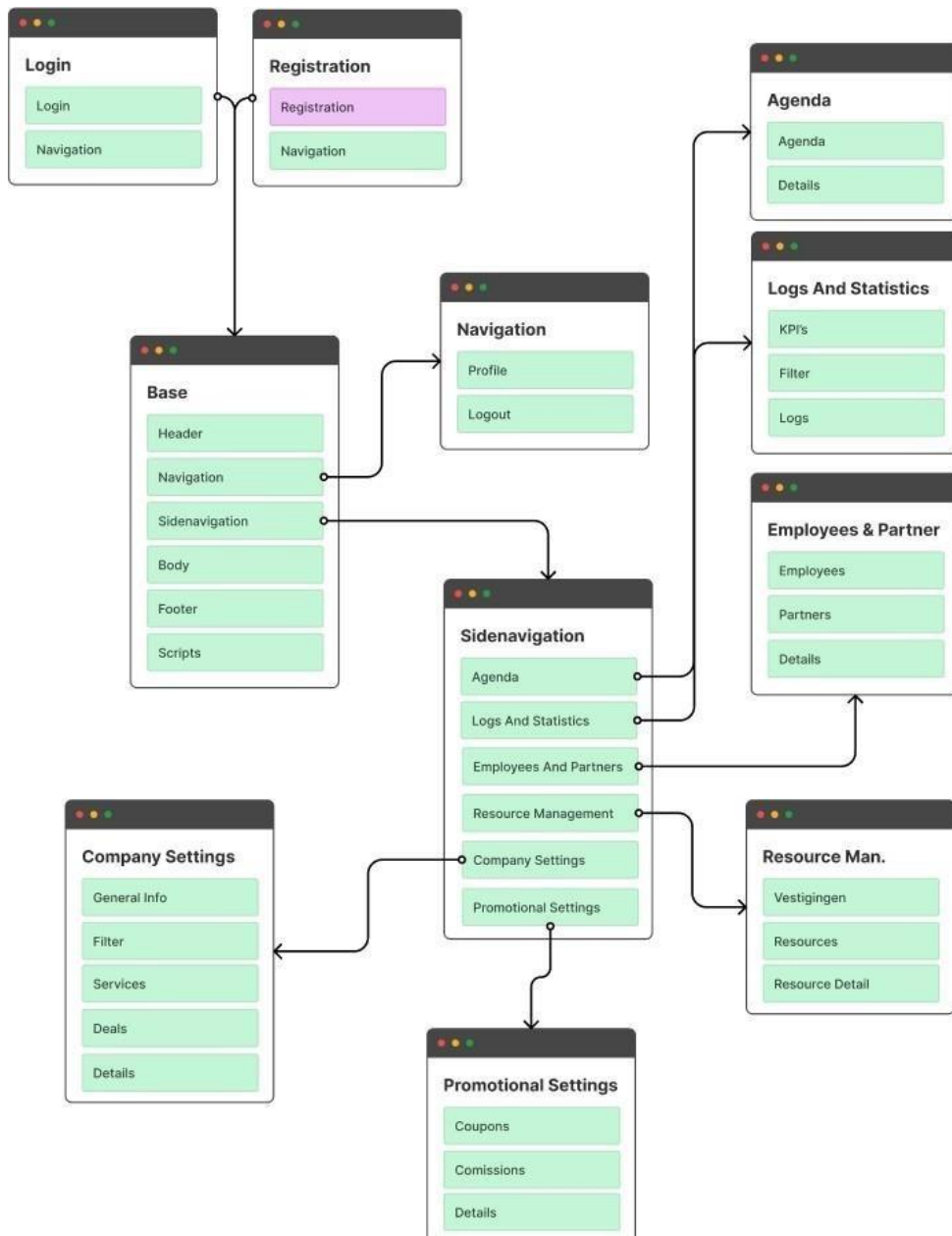
- Caso práctico: Optimización del calendario
 - **Descripción:** El sistema buscará automáticamente posibles optimizaciones en el calendario moviendo reservas. Si se encuentra una optimización, el sistema enviará automáticamente un correo electrónico al cliente correspondiente para ver si la reserva se puede mover. Si el cliente está de acuerdo, el sistema realizará las optimizaciones.
 - **Condiciones:** Existen entradas en el sistema
 - **Post-condiciones:** Se ha optimizado el orden del día
 - **Flujos alternativos:**
 - El cliente no acepta trasladar la reserva
 - El sistema enviará un correo al siguiente cliente, si está de acuerdo con el traslado, se reprogramará la cita de este cliente. El sistema continuará así indefinidamente.
 - El cliente acepta pero no se presenta
 - El miembro del personal lo introducirá en el sistema como "no presentado". Se enviará un correo electrónico al cliente que no se ha presentado. También se envía un correo electrónico al siguiente cliente para ver si desea reprogramar su cita.
 - Ausencia de un miembro del personal
 - Esto se indica en el sistema y todos los clientes del miembro del personal ausente son informados por correo electrónico. Se sigue optimizando para ver si otro miembro del personal puede hacerse cargo de la carga de trabajo.

5.1.2 Diseño de bases de datos



5.1.3 Mapa del sitio

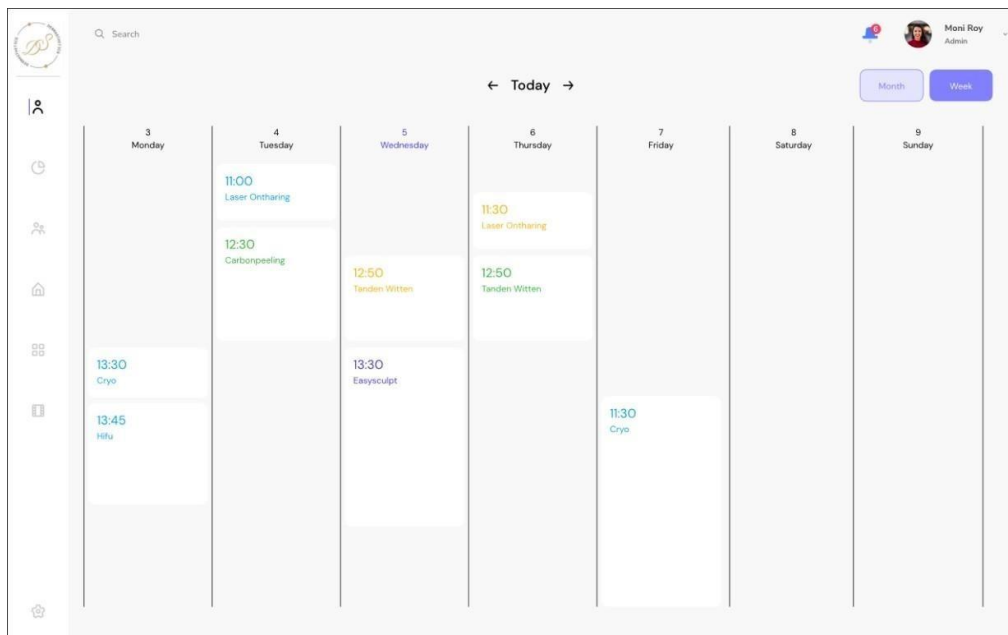
Para crear el mapa del sitio se utilizó Figma.



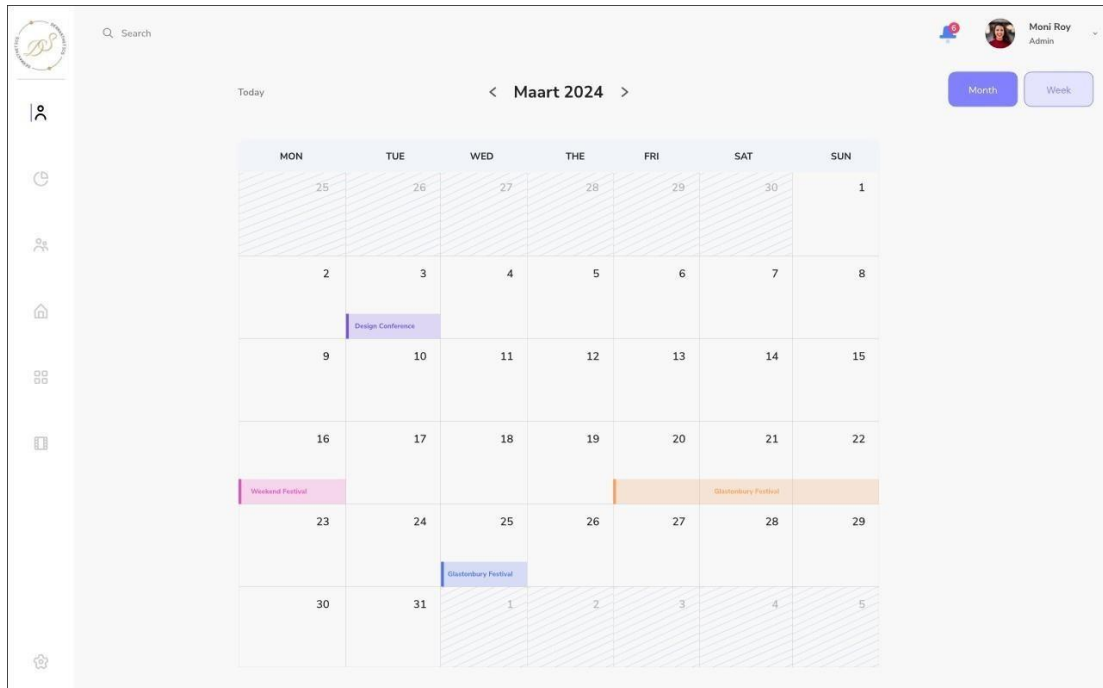
5.1.4 Prototipos

Dado que no recibimos el resumen del proyecto hasta el final de la tercera semana, decidí crear prototipos de inmediato para asegurarme de que no faltara tiempo. Para crear los prototipos se utilizó Figma

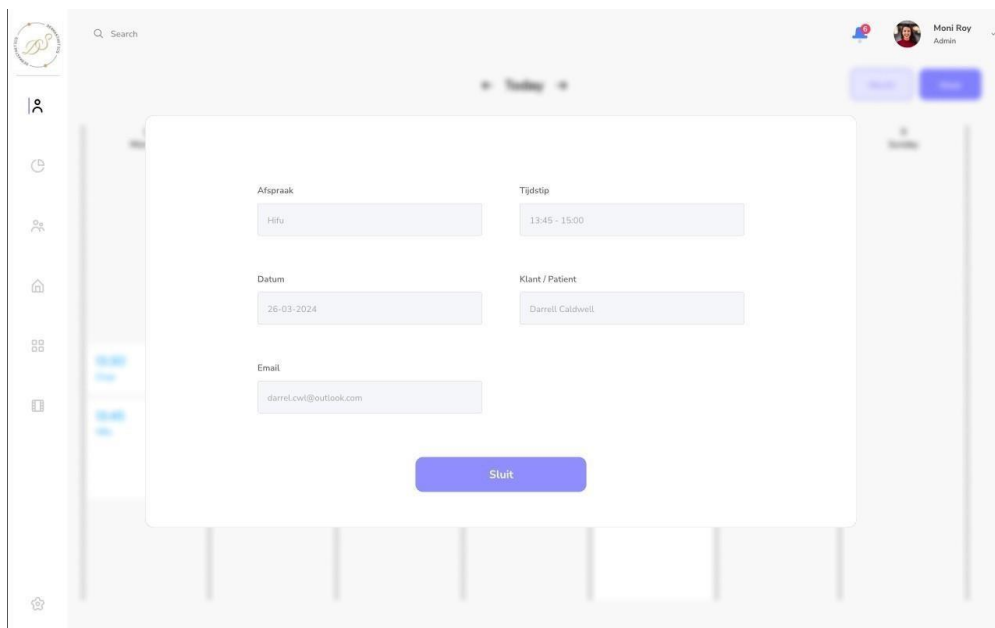
5.1.4.1 Calendario (vista semanal)



5.1.4.2 Calendario (vista mensual)



5.1.4.3 Nombramiento (detalles)



5.1.4.4 Estadísticas y registros

🏠

👤

📅

🔍

📊

📄

⚙️

Q Search

Gebruikers

3048

8.5% Tegenover gisteren

Boeking

5723

1.3% Tegenover gisteren

Pakketten

791

4.3% Tegenover gisteren

Onvoltooid

2040

1.8% Tegenover gisteren

Filter Op

Datum

Type

Status

Reset Filter

ID	NAAM	EMAIL	DATUM	TYPE	STATUS
00001	Christine Brooks	chri.brooks@telenet.be	04 Mrt 2024	Cryo	Voltooid
00002	Rosie Pearson	rosiepearson@gmail.com	04 Mrt 2024	Cryo	Voltooid
00003	Darrell Caldwell	darrel.cwl@outlook.com	26 Mrt 2024	Hifu	Onvoltooid
00004	Gilbert Johnston	gilber10002@gmail.com	14 Mrt 2024	Laser Ontharing	Voltooid
00005	Alan Cain	alain.cain.ac@gmail.com	29 Mrt 2024	Tanden Witten	Onvoltooid
00006	Alfred Murray	alfred-murray@telenet.be	28 Mrt 2024	Easysculpt	Onvoltooid
00007	Maggie Sullivan	maggievds@outlook.com	25 Mrt 2024	Carbonpeeling	Onvoltooid

Moni Roy

Admin

5.1.4.5 Estadísticas y registros (detalles)

🏠

👤

📅

🔍

📊

📄

⚙️

Q Search

Gebruikers

3048

8.5% Tegenover gisteren

Boeking

5723

1.3% Tegenover gisteren

Pakketten

791

4.3% Tegenover gisteren

Onvoltooid

2040

1.8% Tegenover gisteren

Filter Op

Datum

Type

Status

Reset Filter

ID	NAAM	EMAIL	DATUM	TYPE	STATUS
00001	Christine Brooks	chri.brooks@telenet.be	04 Mrt 2024	Cryo	Voltooid
00002	Rosie Pearson	rosiepearson@gmail.com	04 Mrt 2024	Cryo	Voltooid
00003	Darrell Caldwell	darrel.cwl@outlook.com	26 Mrt 2024	Hifu	Onvoltooid
00004	Gilbert Johnston	gilber10002@gmail.com	14 Mrt 2024	Laser Ontharing	Voltooid
00005	Alan Cain	alain.cain.ac@gmail.com	29 Mrt 2024	Tanden Witten	Onvoltooid
00006	Alfred Murray	alfred-murray@telenet.be	28 Mrt 2024	Easysculpt	Onvoltooid
00007	Maggie Sullivan	maggievds@outlook.com	25 Mrt 2024	Carbonpeeling	Onvoltooid

Moni Roy

Admin

Afspraak

Hifu

Tijdstip

13:45 - 15:00

Datum

26-03-2024

Klant / Patient

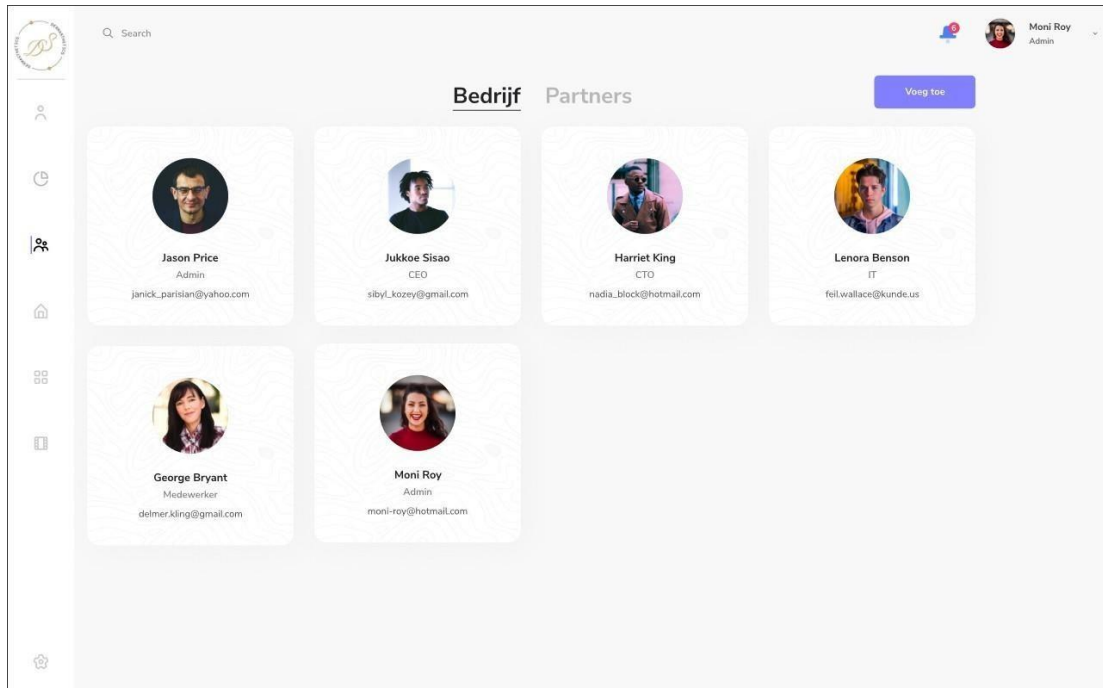
Darrell Caldwell

Email

darrel.cwl@outlook.com

Sluit

5.1.4.6 Contactos



5.1.4.7 Contactos (crear)

Search

Upload Photo

Voornaam
Voer voornaam in

Achternaam
Voer achternaam in

Email
Voer email in

Qualificaties / Rol
Admin

Geboortedatum
20-03-1980

Gender
Man

Voeg Toe

5.1.4.8 Contactos (detalles)

Search

Moni Roy Admin

Upload Photo

Voornaam: Jason

Achternaam: Price

Email: janick_pariisan@yahoo.com

Qualifications / Rol: Voer qualifications / rol in

Geboortedatum: Voer geboortedatum in

Gender: Man

Sluit

5.1.4.9 Recursos

Search

Moni Roy Admin

Voeg nieuwe vestiging toe

Vestigingen

Keerbergen Haachtsebaan 33

Voeg nieuwe resource toe

Resources

Cryomachine Haachtsebaan 33

Cryomachine Haachtsebaan 33

Cryomachine Haachtsebaan 33

Cryomachine

Machine / Apparatuur

Lorem ipsum dolor sit amet consectetur. Molestie ligula purus viverra cras ac pretium magna integer pellentesque. Semper sed est purus adipiscing pulvinar nibh. Erism in massa sed eu proin habitasse lectus. Elementum amet adipiscing sollicitudin scelerisque auctor. Auctor est velit in at viverra sed.

George Bryant

Lily French

5.1.4.10 Recursos (detalles)

Naam: Cryomachine

Type: Machine / Apparatuur

Beschrijving: Lorem Ipsum ...

Specificaties: Medewerker

Sluit

5.1.4.11 Recursos (crear rama)

Naam: Keerbergen

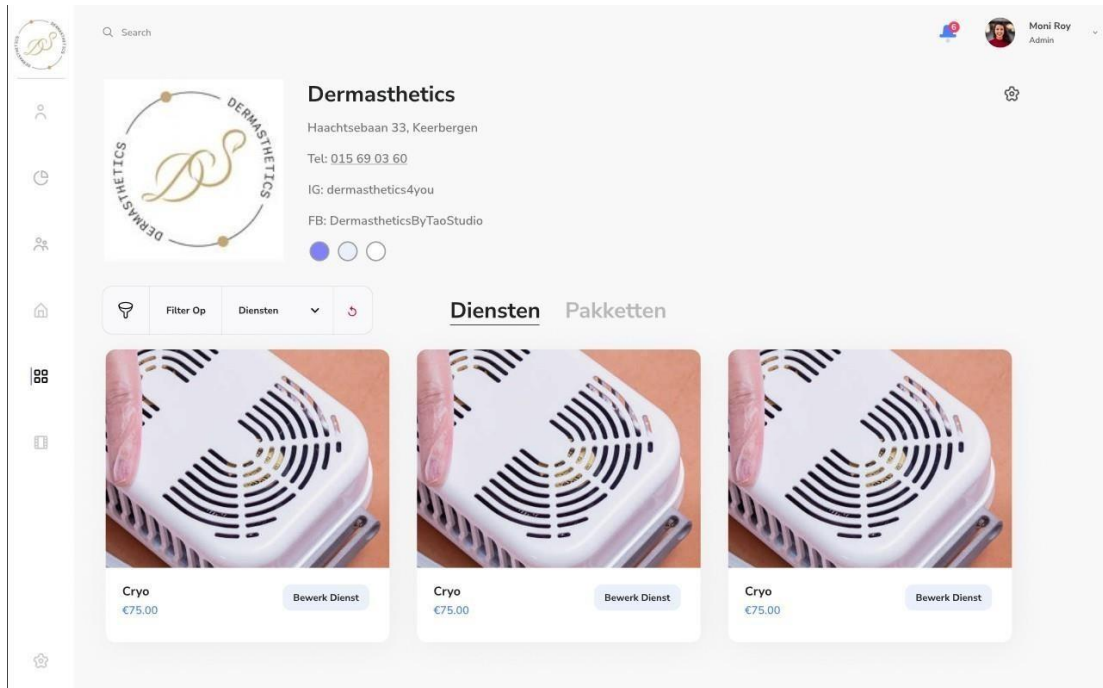
Adres: Haachtsebaan 33, Keerbergen

Openingsuren: Maandag, Dinsdag, Woensdag, Donderdag

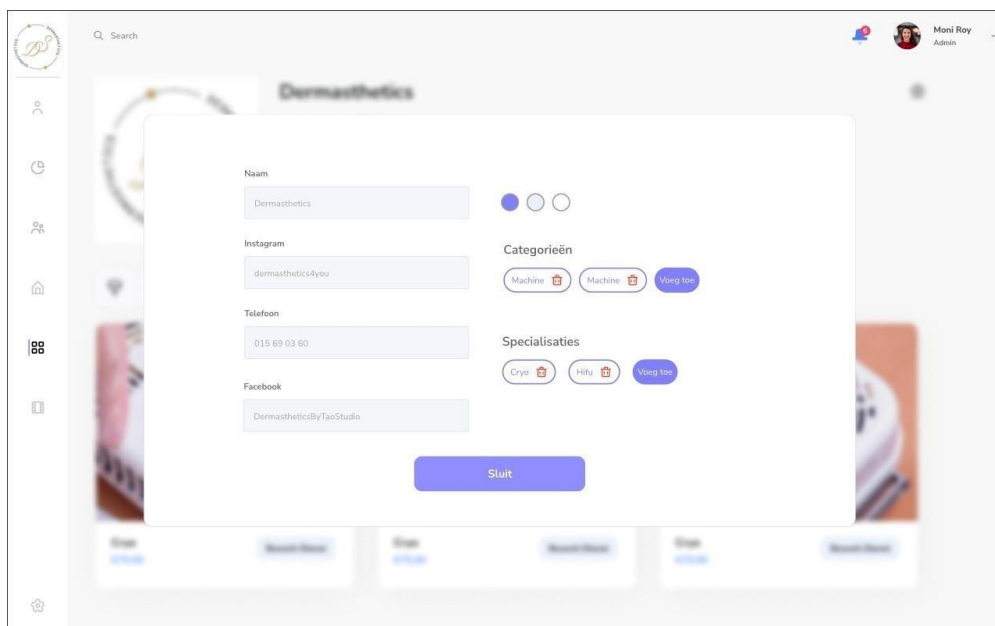
Vrijdag, Zaterdag, Zondag

Sluit


5.1.4.12 Configuración de la empresa



5.1.4.13 Configuración de la empresa (detalles)



5.1.4.14 Códigos promocionales



🏠

🕒

🔍



🏠

📦

📺

⚙️

🔍 Search

  **Moni Roy**
Admin


Coupons

Partners

Voeeg toe

ID	HOEEVEELHEID	CODE	BEDRAG	DATUM	TYPE	STATUS
00001	5	9280148220	15€	14 Apr 2024	Coupon	Actief
00002	1	802021029302	100€	20 Apr 2024	Giftcard	Actief
00003	20	3930202230	20%	14 Feb 2019	Coupon	Inactief

5.1.4.15 Códigos promocionales (socios)



🏠

🕒

🔍



🏠

📦

📺

⚙️

🔍 Search

  **Moni Roy**
Admin

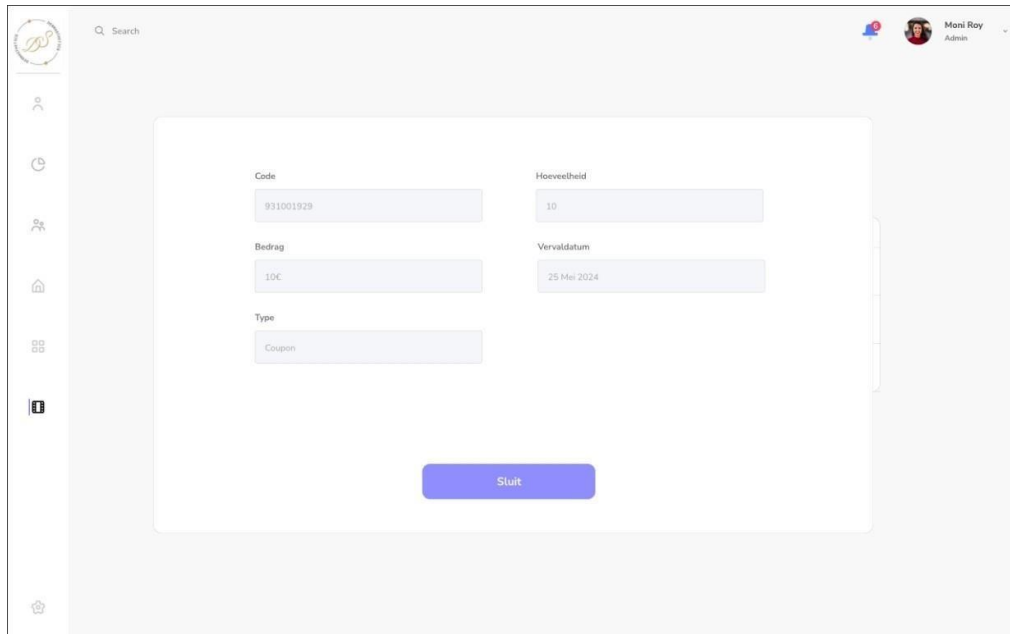
Coupons

Partners

Voeeg toe

ID	PARTNER	CODE	KORTING	DATUM	COMMISSIE	STATUS
00004	Spot Group	2003944920	20%	14 Apr 2024	20%	Actief
00005	LHS	220209120	15%	22 May 2024	10%	Actief
00006	Darrell Caldwell	3930202230	40%	14 Feb 2019	20%	Inactief

5.1.4.16 Códigos de promoción (detalles)




The screenshot shows a web application interface for managing promotional codes. The interface includes a sidebar with navigation icons, a search bar, and a user profile in the top right corner. The main content area displays a form for editing a promotional code. The form fields are as follows:

Field	Value
Code	931001929
Hoeveelheid	10
Bedrag	10C
Vervaldatum	25 Mei 2024
Type	Coupon

At the bottom of the form is a blue button labeled "Stuit".

5.1.4.17 Reservas



Afspraak


Hifu

← Today →

Month Week

3 Monday	4 Tuesday	5 Wednesday	6 Thursday	7 Friday	8 Saturday	9 Sunday
11:00 Boek	11:00 Bezet		11:30 Bezet			
11:30 Boek	12:30 Bezet		12:50 Bezet			
12:00 Boek		12:50 Bezet	12:50 Bezet			
		13:30 Bezet				
13:45 Bezet				11:30 Bezet		

5.1.4.18 Reservas (cont.)



Voornaam

Danell

Achternaam

Caldwell

Email

danell.cwi@outlook.com

Invoice email

danell.cwi@outlook.com

Telefoon

+32 479 80 86 39

Adres

Gelderweg 1

Post code

3304

Stad

Aarschot

Opmerking

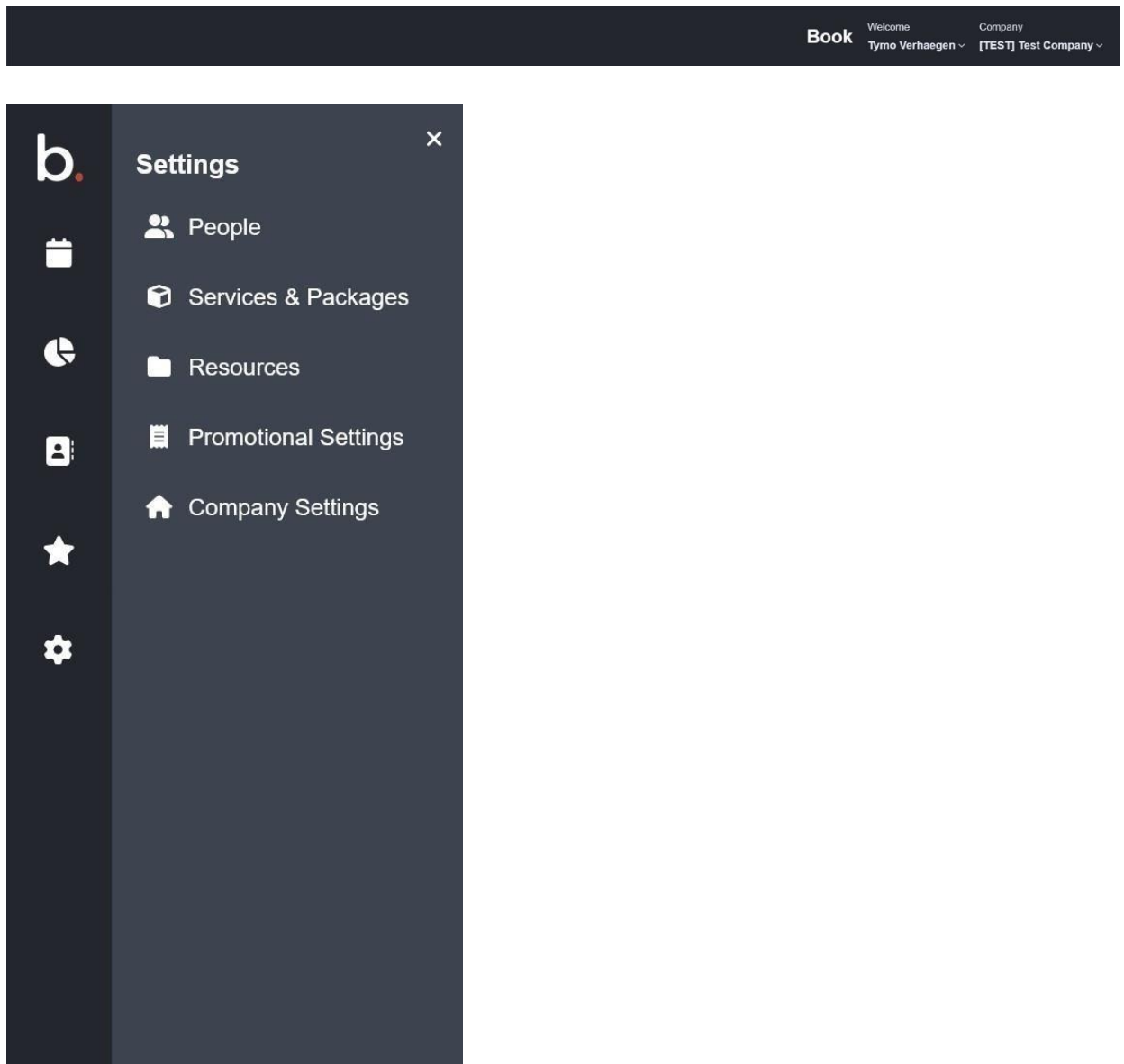
Bevestig booking

5.2 Parte delantera

La parte front-end de nuestra aplicación fue creada en Django y HTML, Tailwind CSS se utilizó para el estilo y Javascript se utilizó para animaciones, ventanas emergentes y más. La aplicación web tiene modos oscuro y claro.

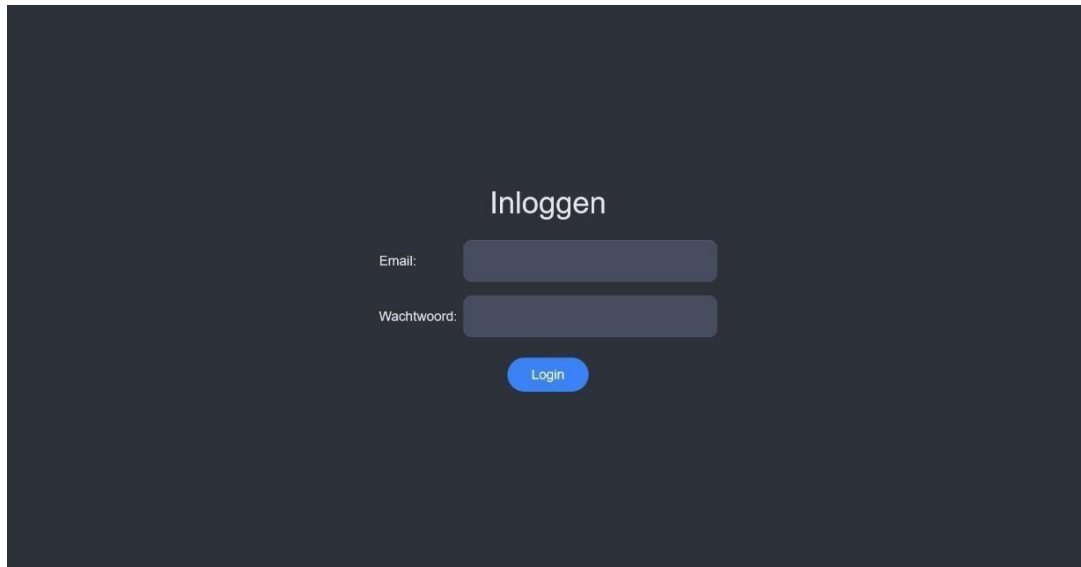
5.2.1 Navegación

Desarrollado por mí y Joppe Kerkhofs



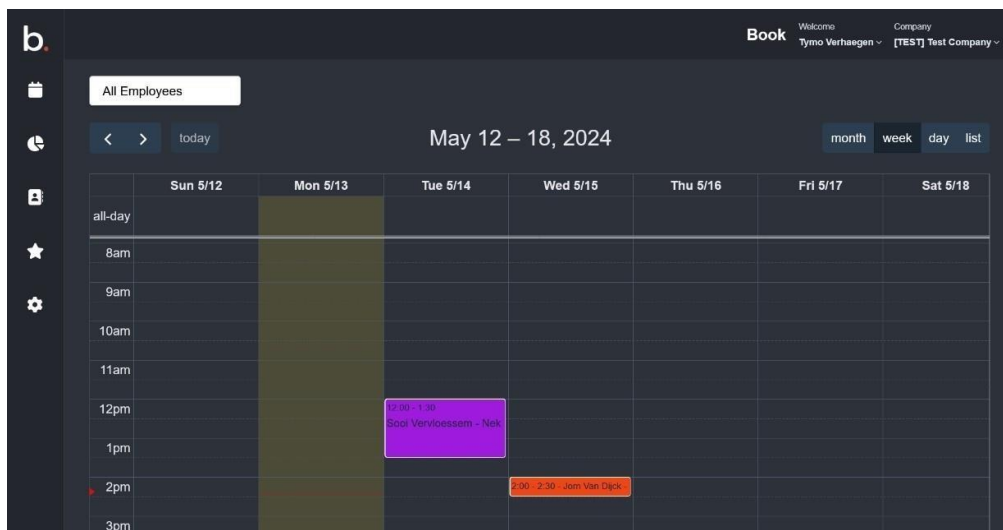
5.2.2 Inicio de sesión

Desarrollado por Joppe Kerkhofs



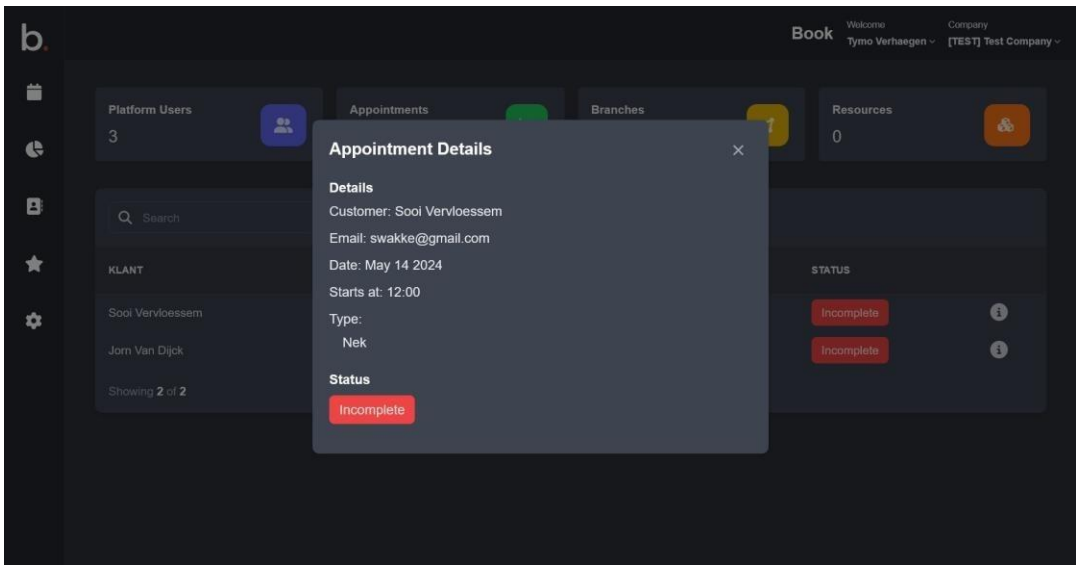
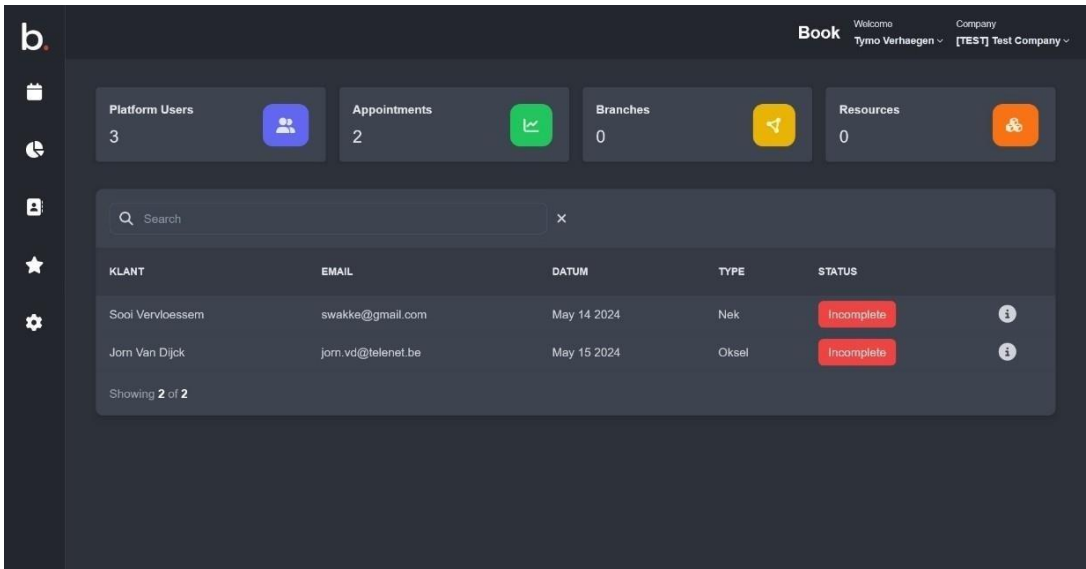
5.2.3 Agenda

Desarrollado por Thomas Verbruggen



5.2.4 Estadísticas y registros

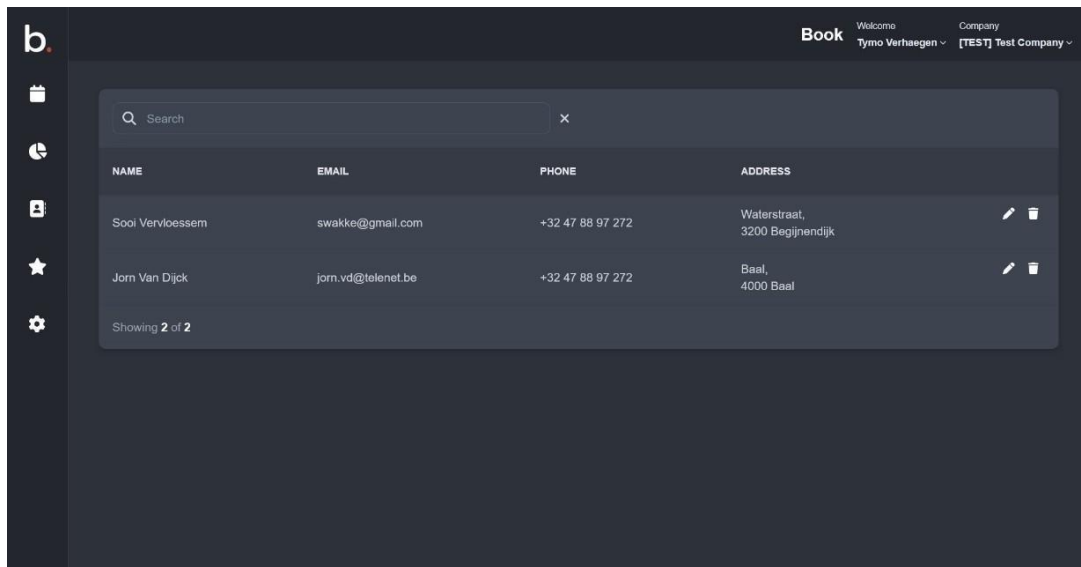
He desarrollado toda esta página, tanto el front-end como el back-end.



5.2.5 Clientes

Desarrollado por Joppe Kerkhofs y yo mismo

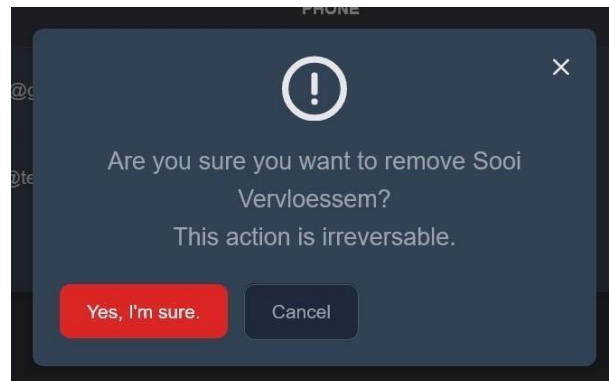
Vista del cliente



Actualizar cliente

The screenshot shows a modal form titled 'Update - Sooi Vervloessem'. The form has fields for: FIRST NAME (Sooi), LAST NAME (Vervloessem), EMAIL (swakke@gmail.com), PHONE (+32 47 88 97 272), ADDRESS (Waterstraat), ZIPCODE (3200), and CITY (Begijnendijk). At the bottom, there is a blue button labeled 'Update Customer'.

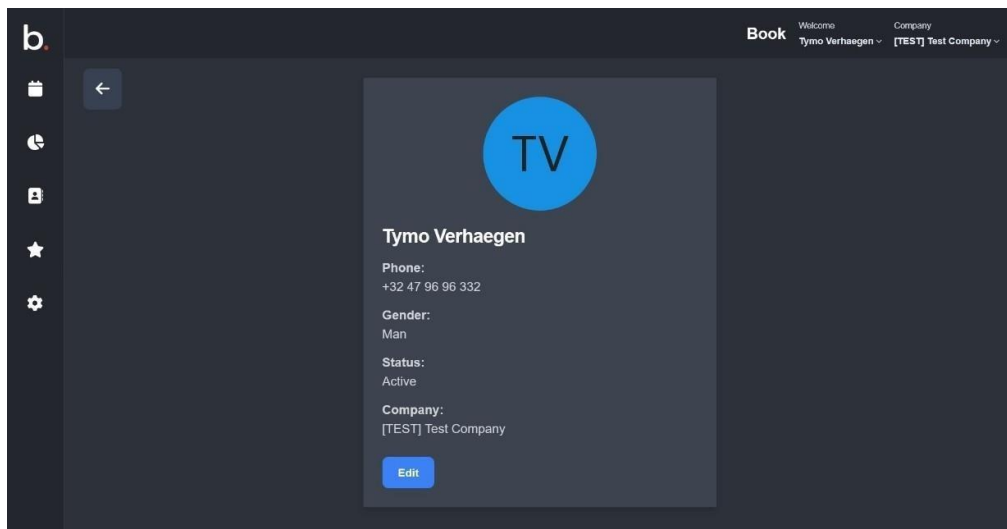
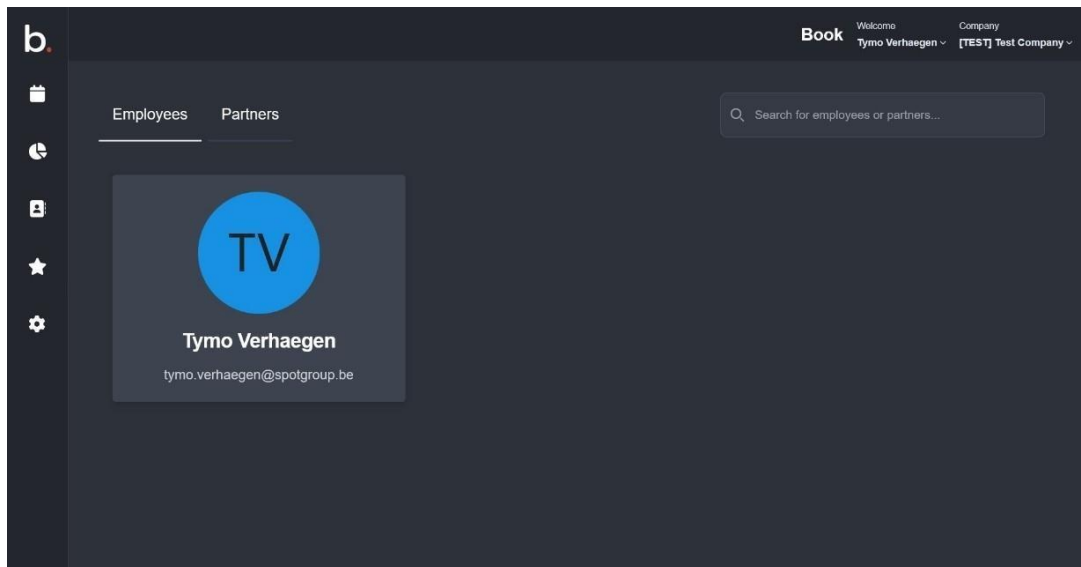
Borrar cliente



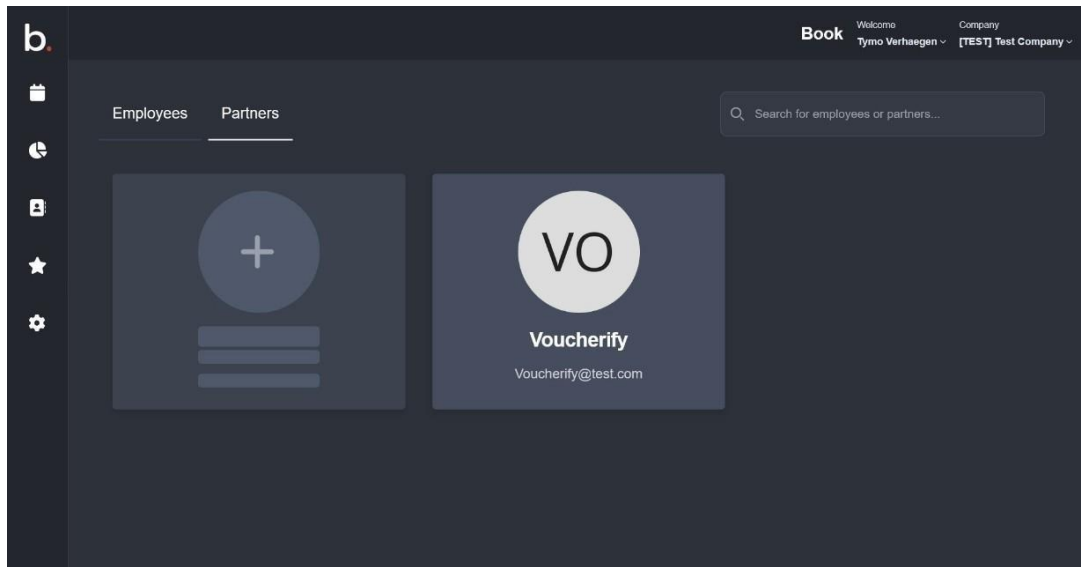
5.2.6 Empleados

Desarrollado por Joppe Kerkhofs y yo mismo

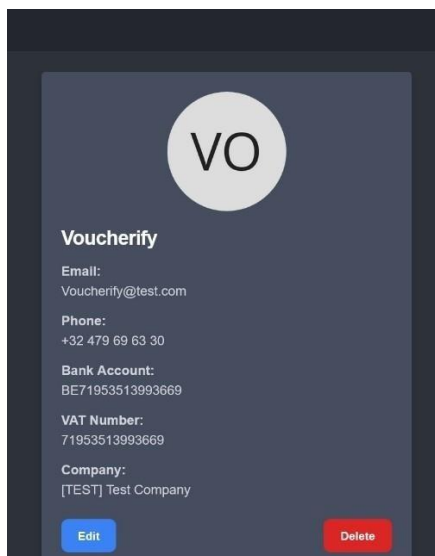
Ficha Empleado



Ficha Socio



Información para socios

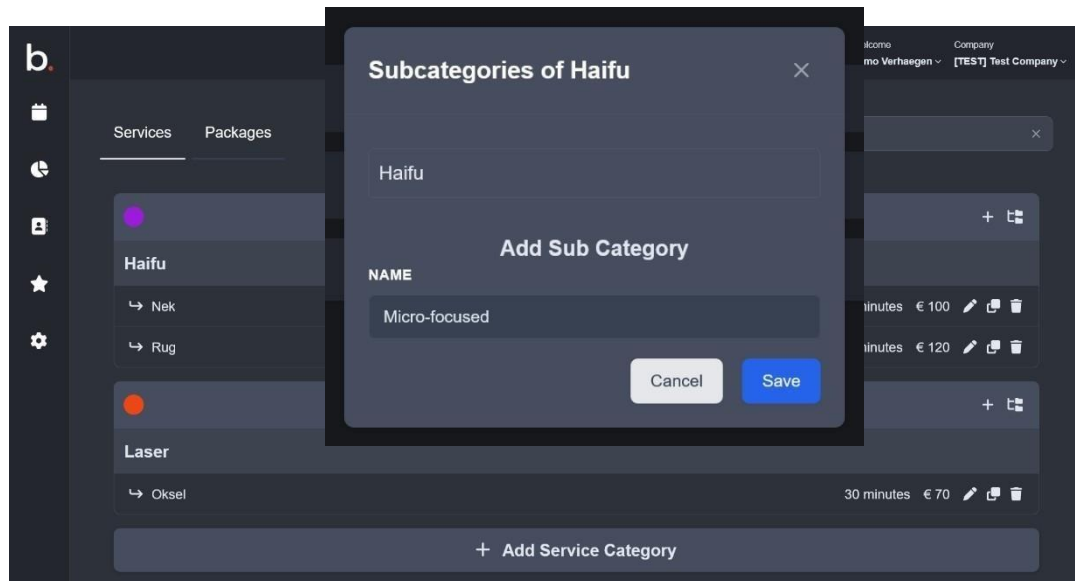


Añadir socio

5.2.7 Servicios y paquetes

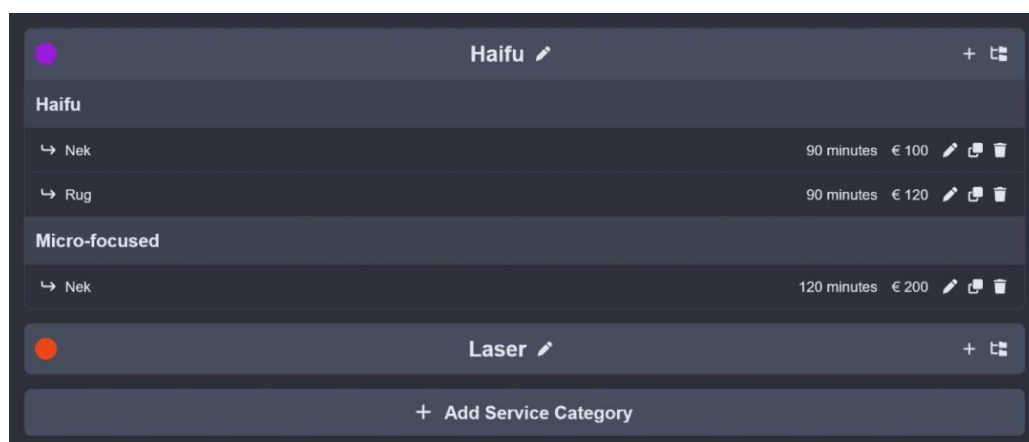
Desarrollado por Joppe Kerkhofs y por mí

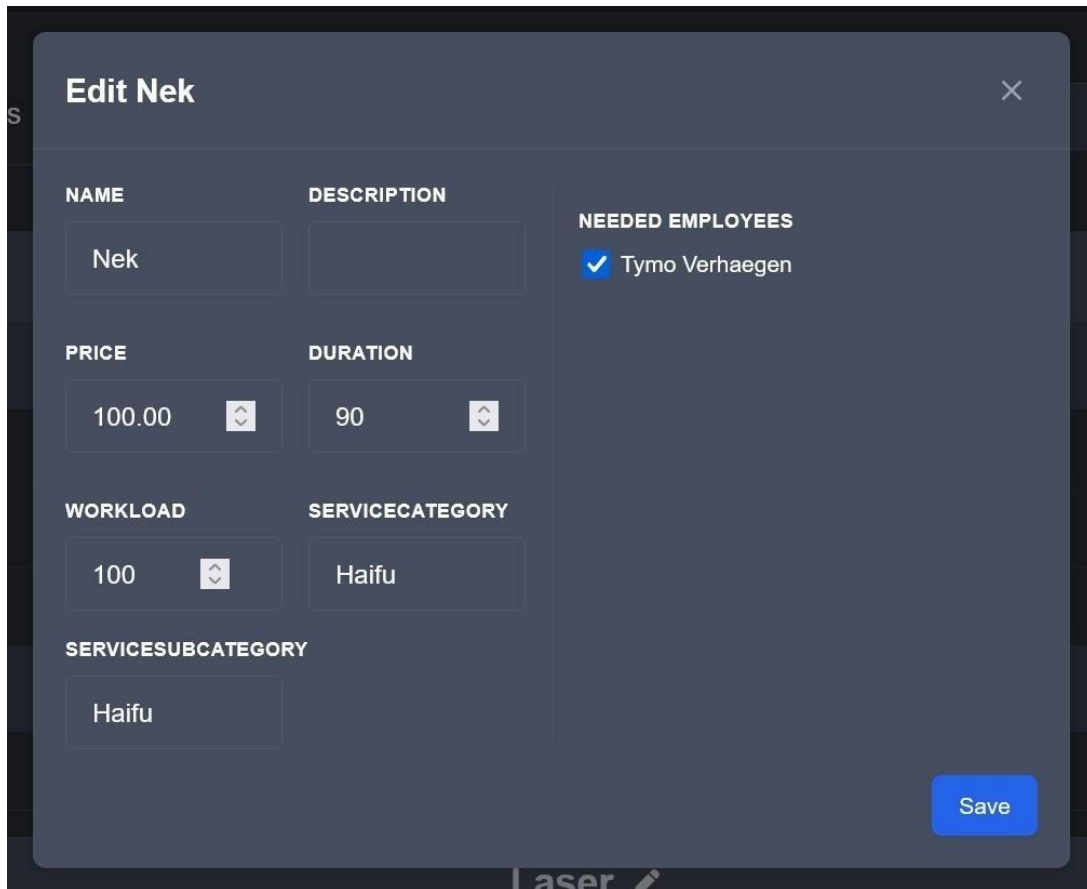
Vista principal



Crear subcategoría

Múltiples subcategorías

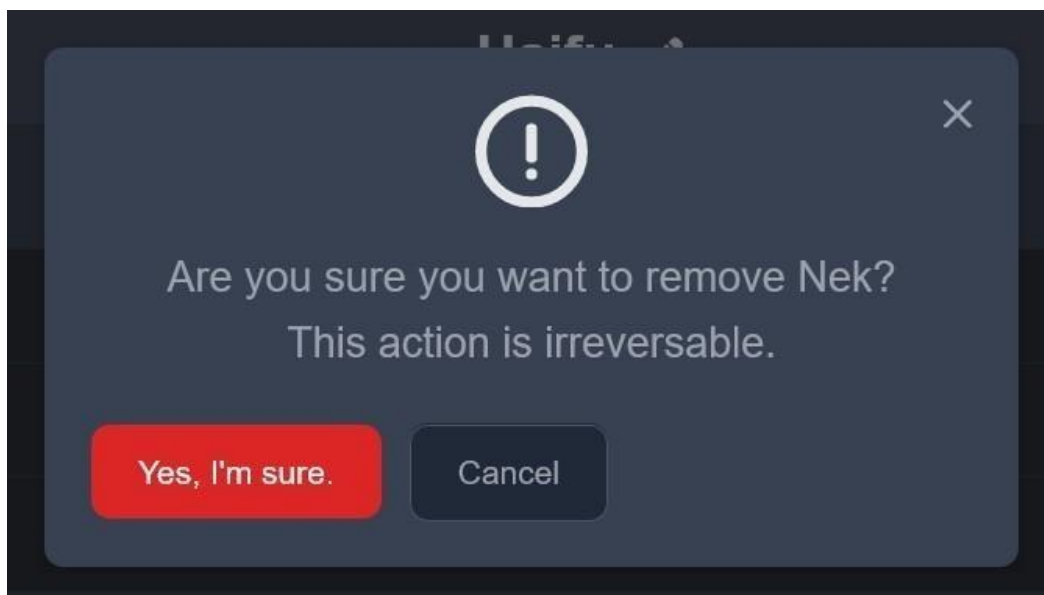


Editar en servicio

The 'Edit Nek' dialog box is a dark-themed window with a title bar containing the text 'Edit Nek' and a close button (X). The main content area is divided into several sections for editing service details:

- NAME:** A text input field containing 'Nek'.
- DESCRIPTION:** An empty text input field.
- NEEDED EMPLOYEES:** A section with a blue checkmark icon and the text 'Tymo Verhaegen'.
- PRICE:** A numeric input field showing '100.00' with a small up/down arrow icon.
- DURATION:** A numeric input field showing '90' with a small up/down arrow icon.
- WORKLOAD:** A numeric input field showing '100' with a small up/down arrow icon.
- SERVICECATEGORY:** A text input field containing 'Haifu'.
- SERVICSUBCATEGORY:** A text input field containing 'Haifu'.

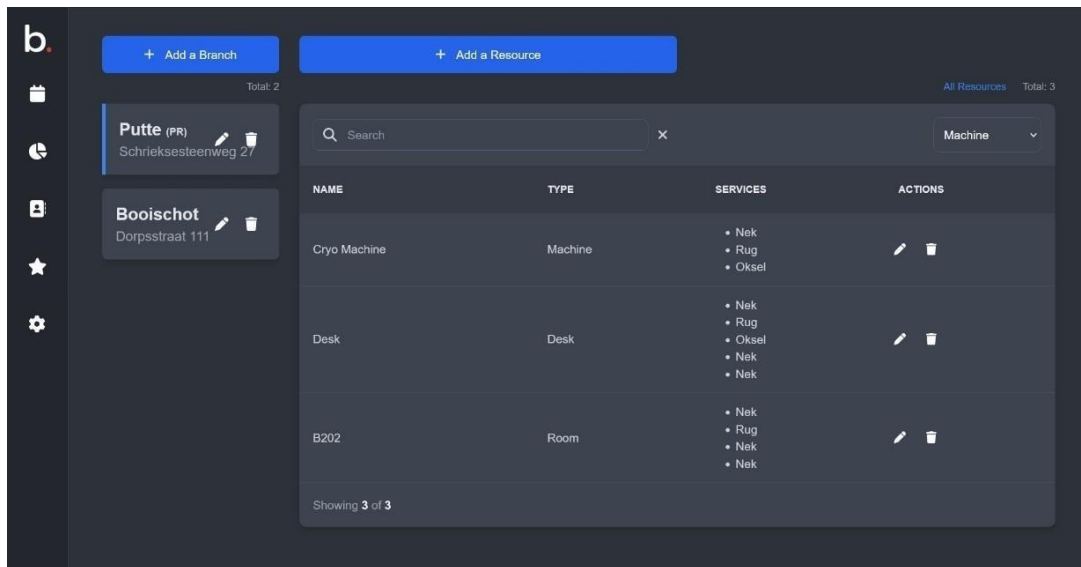
A blue 'Save' button is located in the bottom right corner of the dialog.

Supresión del servicio

5.2.8 Recursos

Desarrollado por todo el grupo, Thomas Verbruggen, Joppe Kerkhofs y yo mismo.

Vista principal





Añadir recurso

The screenshot shows the 'Add Resource' modal form. It has a title bar with 'Add Resource' and a close button. The form is divided into several sections: 'NAME' and 'DESCRIPTION' (with a plus icon between them), 'RESOURCE TYPE' and 'BRANCH', and a 'SERVICE' section with five checkboxes labeled 'Nek', 'Rug', 'Oksel', 'Nek', and 'Nek'. At the bottom is a blue button labeled 'Add Resource'.

Vista filtrada

Machine






NAME	TYPE	SERVICES	ACTIONS
Desk	Desk	<ul style="list-style-type: none">NekRugOkselNekNek	 

Showing 1 of 1

5.2.9 Promociones y ofertas

Desarrollado por Thomas Verbruggen y yo mismo

b.



Book

Welcome

Tymo Verhaegen

Company

[TEST] Test Company



Discount

Gift Cards

Commission

+ Create Code

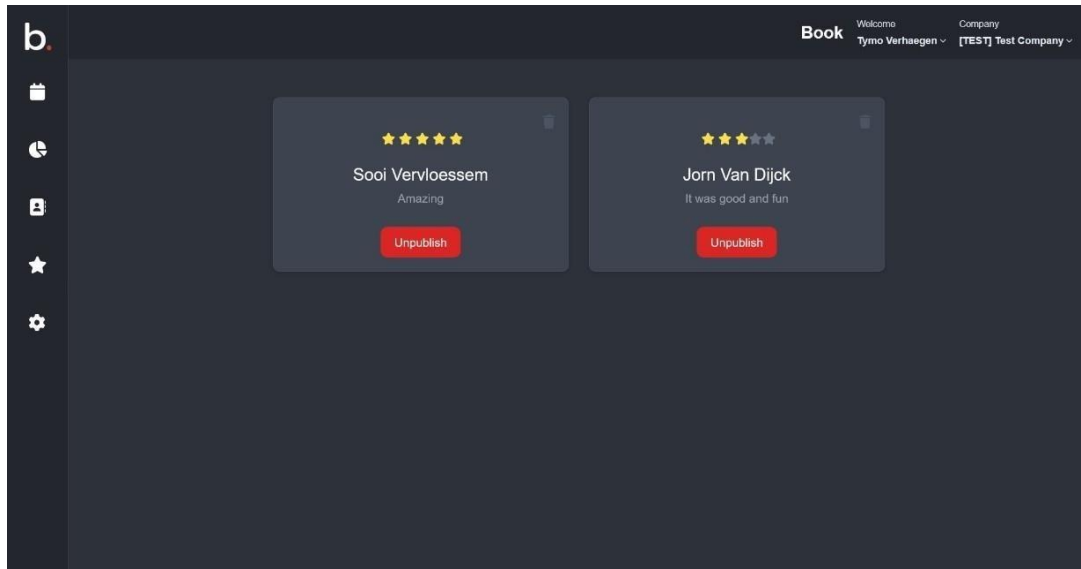
X

CODE	TYPE	PARTNER	PERCENTAGE	AMOUNT	USES	COMMISSION	EXPIRE DATE	
KAI98T91	Discount	Voucherify	100	40.00	1	100	2024-05-15	 

Showing 1 of 1

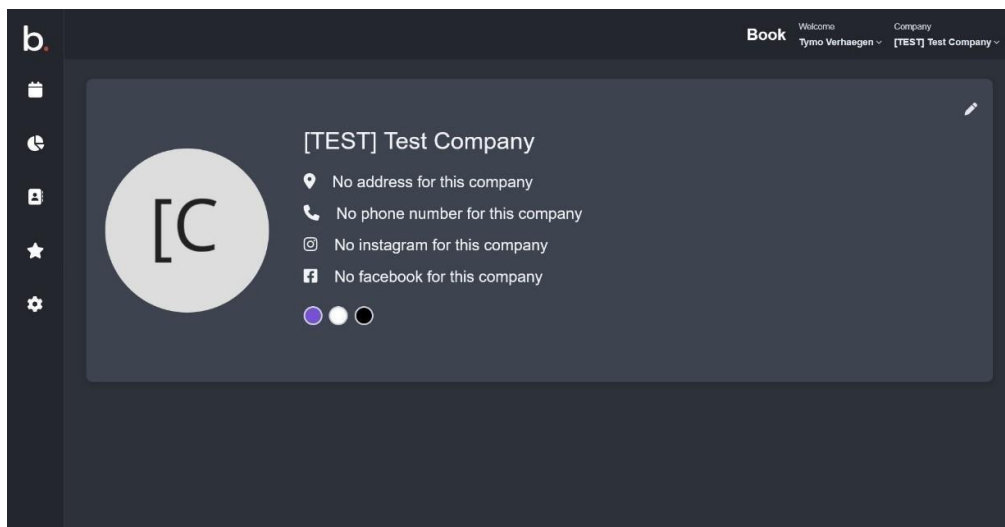
5.2.10 Reseñas

Desarrollado por Thomas Verbruggen



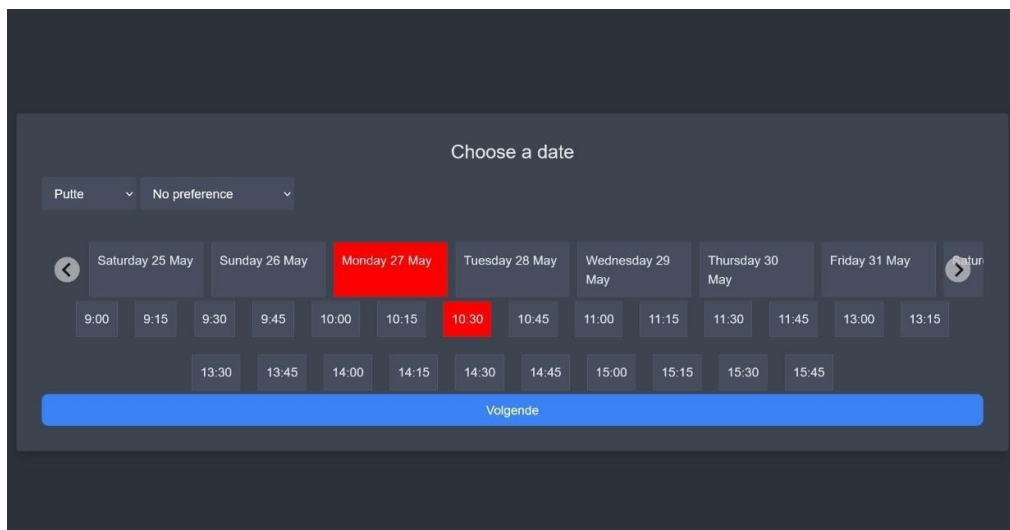
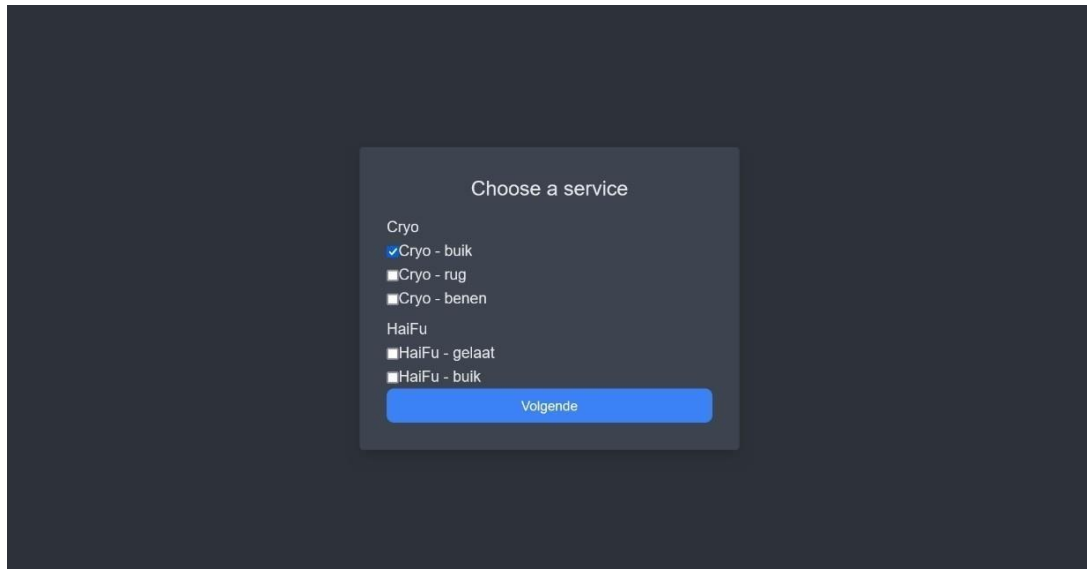
5.2.11 Configuración de la empresa

Desarrollado por Thomas Verbruggen y yo mismo



5.2.12 Reserva (usuario final)

Desarrollado por todo el grupo, Thomas Verbruggen, Joppe Kerkhofs y yo mismo.



Persoonlijke informatie

Tes

test

test@test.com

test

Factuur E-mail

test

3000

test

Opmerking

Betalen

5.3 Backend

La parte backend de nuestra aplicación web está diseñada con vistas Django y todos los datos se escriben y almacenan en una base de datos PostgreSQL.

5.3.1 Navegación

La navegación se controla mediante un archivo `urls.py`, que contiene todos los enlaces posibles que un usuario final puede visitar

Desarrollado por Joppe Kerkhofs y yo mismo

```
urlpatterns = [
    # Redirects to agenda page
    path('', RedirectView.as_view(url='/agenda/', permanent=False), name='home'),

    # First login page
    path('first_login/', views.FirstLoginView.as_view(), name='firstlogin'),

    # AGENDA
    path('agenda/', views.AgendaView.as_view(), name='agenda'),
    path('event/<uid:event_id>', views.EventDetailsView.as_view(), name='event_details'),
    path('update_appointment_field/<uid:event_id>', views.update_appointment_field.as_view(), name='updateAppointment'),

    # COMPANY
    path('company/', views.CompanyView.as_view(), name='company'),
    path('company/edit/', views.EditCompanyView.as_view(), name='updateCompany'),

    # PEOPLE
    path('people/', views.PeopleView.as_view(), name='people'),
    # Employee CRUD operations
    path('people/user/info/<uid:user_id>', views.UserInfoView.as_view(), name='updateEmployee'),
    # Partner CRUD operations
    path('people/partner/add/', views.PartnerAddView.as_view(), name='addPartner'),
    path('people/partner/info/<uid:id>', views.PartnerView.as_view(), name='updatePartner'),
    path('people/partner/remove/<uid:id>', views.PartnerRemoveView.as_view(), name='removePartner'),
    # Customer CRUD operations
    path('customers/', views.CustomersView.as_view(), name='customers'),
    path('customer/update/<uid:pk>', views.UpdateCustomerView.as_view(), name='updateCustomer'),
    path('customer/remove/<uid:pk>', views.DeleteCustomerView.as_view(), name='removeCustomer'),

    # RESOURCES
    path('resources/', views.ResourcesView.as_view(), name='resources'),
    # Branch CRUD operations
    path('resources/<uid:branch_id>', views.EditBranchView.as_view(), name='updateBranch'),
    path('resources/', views.ResourcesView.as_view(), name='addBranch'),
```

```
path('resources/<uid:branch_id>', views.EditBranchView.as_view(), name='updateBranch'),
path('resources/', views.ResourcesView.as_view(), name='addBranch'),
path('resources/branch/remove/<uid:id>', views.RemoveBranchView.as_view(), name='removeBranch'),
# Resource CRUD operations
path('resources/info/<uid:id>', views.EditResourceView.as_view(), name='updateResource'),
path('resources/', views.ResourcesView.as_view(), name='addResource'),
path('resources/remove/<uid:id>', views.RemoveResourceView.as_view(), name='removeResource'),
path('resources/name/add', views.ResourceNameView.as_view(), name='resourceName'),
path('resources/name/update/<uid:id>', views.UpdateResourceNameView.as_view(), name='updateResourceName'),
path('resources/name/remove/<uid:id>', views.RemoveResourceNameView.as_view(), name='removeResourceName'),

# SERVICES
# Service CRUD operations
path('services/', views.ServicesView.as_view(), name='services'),
path('services/update/<uid:id>', views.EditServiceView.as_view(), name='updateService'),
path('services/remove/<uid:id>', views.DeleteServiceView.as_view(), name='removeService'),
path('services/duplicate/<uid:serviceId>', views.ServiceDuplicateView.as_view(), name='duplicateServices'),
# Service Category
path('services/category/', views.ServiceCategoryView.as_view(), name='serviceCategory'),
path('services/subcategories/', views.ServiceSubcategoryView.as_view(), name='serviceSubCategory'),
path('services/subcategories/update/<uid:id>', views.UpdateServiceSubcategoryView.as_view(), name='updateServiceSubCategory'),
path('services/subcategories/remove/<uid:id>', views.RemoveServiceSubcategoryView.as_view(), name='removeServiceSubCategory'),
# Package CRUD operations
path('packages/add/', views.AddPackageView.as_view(), name='addPackage'),
path('packages/update/<uid:id>', views.EditPackageView.as_view(), name='updatePackage'),
path('packages/remove/<uid:id>', views.DeletePackage.as_view(), name='removePackage'),

# PROMOTIONS
path('promotions/', views.PromotionsView.as_view(), name='promotions'),
path('promotions/create/', views.CreatePromotionView.as_view(), name='createCode'),
path('promotions/update/<uid:id>', views.EditPromotionView.as_view(), name='updateCode'),
path('promotions/remove/<uid:id>', views.DeletePromotionView.as_view(), name='removeCode'),

# STATISTICS
path('statistics/', views.StatisticView.as_view(), name='statistics'),
```

5.3.2 Inicio de sesión

Desarrollado por Joppe Kerkhofs

FirstLoginView:

- Funcionalidad:
 - Presentación de la vista cuando el usuario se conecta por primera vez
 - Una solicitud POST procesa los datos introducidos por el usuario
 - Si el formulario es válido
 - Comprueba si UserInfo ya existe
 - Para los UserInfo existentes, se actualiza la información preexistente
 - Con new UserInfo, se crea un nuevo usuario
 - Las empresas se cargan mediante el método *cargar_empresas()*
 - El usuario es redirigido a la página del calendario
 - El método *load_companies()* carga empresas en la base de datos comprobando si hay empresas de CloudSpot y añadiéndolas al modelo *CompanyInfo*
 - Cuando se recuperan los datos contextuales, se añade un formulario al contexto

5.3.3 Agenda

Desarrollado por Thomas Verbruggen

AgendaView:

- Funcionalidad:
 - Vista de renderizado de la página de calendario
 - En una petición GET, comprueba si el usuario se está conectando por primera vez utilizando la función *check_first_login()*
 - Recoge empleados para la empresa
 - Construye un elemento de selección de empleados
 - Recoge las citas y los servicios asociados de la base de datos y aplica cálculos para determinar la duración y la hora de finalización de las citas.

- Filtra las citas de los empleados seleccionados y prepara los datos contextuales para enviarlos a la plantilla.

EventDetailsView:

- Funcionalidad:
 - Esta vista muestra la página detallada de un evento específico (cita) dentro del calendario
 - En una solicitud GET, recupera la información sobre el evento seleccionado y el cliente asociado
 - Recoge información sobre los servicios asociados a la cita y calcula el precio total
 - Remite los datos recogidos a la plantilla para que los muestre

update_appointment_field:

- Funcionalidad:
 - Esta vista procesa las solicitudes POST para actualizar el estado de una cita, como "informó tarde" y "no se presentó".
 - Comprueba que se han incluido los campos correctos en la solicitud y actualiza el estado de la cita
 - Guarda los cambios en la base de datos y redirige al usuario a la página de detalles.

5.3.4 Estadísticas y

registros Desarrollado

por mí mismo

StatisticView:

- Funcionalidad:
 - Esta vista muestra estadísticas relacionadas con citas, usuarios, sucursales y recursos.
 - Con una petición GET, recoge los datos necesarios para las estadísticas.
 - Recupera el número de usuarios, citas, sucursales y recursos.
 - Realiza una función de búsqueda en las citas basándose en la consulta de búsqueda de la URL.
 - Actualiza el estado de las citas en función de la hora actual, marcando las citas como "finalizadas" cuando ha pasado la hora de finalización prevista.
 - Se incluye la función de paginación para que la mayor parte de los registros sean más manejables.

5.3.5 Clientes, empleados y socios

Desarrollado por Joppe Kerkhofs y yo mismo

PeopleView:

- Funcionalidad:
 - Esta vista gestiona la información relacionada con "personas" u organismos como socios
 - Al recuperar datos contextuales, recupera datos de la base de datos, incluidos usuarios, clientes, empresas y socios.
 - Crea un formulario para añadir nuevos socios
 - Con una solicitud POST, procesa el formulario enviado para crear un nuevo socio
 - Si el formulario es válido, el nuevo socio se guarda y el usuario es redirigido a la página Gestionar personas con un mensaje de éxito

UserInfoView:

- Funcionalidad:
 - Esta vista muestra y actualiza la información del usuario
 - Al recuperar los datos contextuales, recupera la información del usuario y añade un formulario para actualizar la información
 - Para una solicitud POST, procesa el formulario enviado para actualizar la información del usuario
 - Si el formulario es válido, la información del usuario se actualiza con los nuevos datos

PartnerView:

- Funcionalidad:
 - Esta vista muestra y actualiza la información sobre los socios
 - Al recuperar los datos contextuales, recupera la información de los socios y añade un formulario para actualizar la información
 - Con una solicitud POST, procesa el formulario enviado para actualizar la información del socio
 - Si el formulario es válido, la información del socio se actualizará con los nuevos datos

PartnerRemoveView:

- Funcionalidad:
 - Este punto de vista elimina a un socio
 - En una solicitud POST, el interlocutor se identifica y se elimina de la base de datos

PartnerAddView:

- Funcionalidad:
 - Esta vista añade un nuevo interlocutor

- Con un formulario válido, valida los datos y crea un nuevo interlocutor
- Si el interlocutor ya existe, aparece un mensaje

CustomerView:

- Funcionalidad:
 - Esta vista muestra una lista de clientes
 - En una petición GET, recupera los clientes asociados a la empresa del usuario actual
 - Permite buscar clientes en función de la consulta de búsqueda introducida.
 - Se ha añadido la paginación para facilitar la gestión de un gran número de clientes.

ActualizarVistaCliente:

- Funcionalidad:
 - Esta vista actualiza los datos del cliente
 - Utiliza un formulario para actualizar los datos del cliente
 - Con un formulario válido, valida los datos y actualiza los datos del cliente

BorrarVistaCliente:

- Funcionalidad:
 - Esta vista elimina un cliente
 - Muestra una página de confirmación para eliminar al cliente
 - Al confirmar la eliminación, el cliente se elimina y el usuario es redirigido a la lista de clientes

5.3.6 Reseñas

Desarrollado por Thomas Verbruggen

RevisionesVer:

- Funcionalidad:
 - Esta vista muestra las revisiones
 - Recupera un conjunto de reseñas filtradas por citas de la empresa del usuario

BorrarRevisión:

- Funcionalidad:
 - Esta vista borra una revisión.
 - Una solicitud POST elimina la revisión.
 - Después de borrar la revisión se muestra un mensaje de éxito.

AlternarRevisiónVistaPública:

- Funcionalidad:
 - Esta vista alterna el estado público de una revisión.
 - Con una solicitud POST, el estado público de la revisión es

5.3.7 Servicios y paquetes

Desarrollado por todo el grupo, Thomas Verbruggen, Joppe Kerkhofs y yo mismo.

ServiciosVer:

- Funcionalidad:
 - Recupera los datos de la base de datos y prepara el contexto para la representación.
 - Procesa las solicitudes POST para añadir un nuevo servicio
 - Proporciona formularios para añadir nuevos servicios y/o paquetes

VistaCategoríaServicio:

- Funcionalidad:
 - Gestiona la creación o actualización de una categoría de servicio
 - Procesa las solicitudes POST para añadir una nueva categoría de servicio o actualizar una existente.
 - Muestra un mensaje de éxito después de añadir o actualizar correctamente la categoría de servicio.

VistaSubcategoríaServicio:

- Funcionalidad:
 - Gestiona la creación de una subcategoría para los servicios
 - Procesa las solicitudes POST para añadir una nueva subcategoría a una categoría de servicio determinada.
 - Verifica que los datos requeridos están presentes en las peticiones POST y luego añade la nueva subcategoría

UpdateServiceSubCategoryView:

- Funcionalidad:
 - Procesa las solicitudes POST para actualizar una subcategoría de servicio existente
 - Comprueba que los datos requeridos, como el ID y el nombre de la subcategoría, están presentes en la solicitud POST.
 - Actualiza el nombre de la subcategoría de servicio y guarda los cambios.

EliminarServicioSubCategoríaView:

- Funcionalidad:
 - Procesa las solicitudes POST para eliminar una subcategoría de servicio
 - Comprueba si la subcategoría tiene servicios antes de eliminarla
 - Eliminar la subcategoría de servicios si no tiene servicios

ServicioDuplicarVista:

- Funcionalidad:
 - Procesa las peticiones POST para duplicar un servicio
 - Crea un nuevo servicio con los mismos datos que el servicio original
 - También duplica las *instancias* UserAvailableService para el nuevo servicio
 -

EditarServicioVista:

- Funcionalidad:
 - Procesa las solicitudes POST para actualizar un servicio
 - Actualizar las propiedades de un servicio existente con los datos del formulario
 - Actualice también las *instancias* UserAvailableService para el servicio actualizado

BorrarVistaServicio:

- Funcionalidad:
 - Procesa las solicitudes POST para eliminar un servicio
 - Elimina el servicio seleccionado junto con todas las *instancias* UserAvailableService relacionadas.

AñadirVistaPaquete:

- Funcionalidad:
 - Procesa las peticiones POST para añadir un nuevo paquete
 - Crea un nuevo paquete con los datos enviados y vincula los servicios seleccionados al paquete.

EditarVistaPaquete:

- Funcionalidad:
 - Procesa las peticiones POST para actualizar un paquete
 - Actualiza las propiedades de un paquete existente con los datos del formulario y actualiza los servicios asociados.

BorrarVistaPaquete:

- Funcionalidad:
 - Procesa las solicitudes POST para eliminar un paquete
 - Elimina el paquete seleccionado de la base de datos

5.3.8 Recursos

Desarrollado por todo el grupo, Thomas Verbruggen, Joppe Kerkhofs y yo mismo.

RecursosVer:

- Funcionalidad:
 - Gestiona solicitudes GET y POST para administrar recursos
 - Recupera recursos y aplica filtros en función de las búsquedas, la rama y el tipo de recurso.
 - Muestra los recursos en paginación y proporciona formularios para añadir nuevas ramas y recursos.
 - Gestiona la adición de nuevas sucursales y recursos, y vincula los servicios a los recursos.
 - Formularios para la creación de recursos y sucursales

EditarVistaRama:

- Funcionalidad:
 - Permite editar una sucursal existente
 - Muestra un formulario para cambiar la información del afiliado, como nombre, dirección, información de contacto, etc.
 - Al enviar el formulario, la rama se actualiza con los nuevos datos y el usuario es redirigido a la página de recursos con un mensaje de éxito

QuitarVistaRama:

- Funcionalidad:
 - Permite eliminar un afiliado existente
 - Procesa las solicitudes POST para borrar un afiliado eliminándolo de la base de datos
 - Tras la eliminación, el usuario es redirigido a la página de recursos con una notificación de que la rama se ha eliminado correctamente.

EditarVistaRecurso:

- Funcionalidad:
 - Permite editar un recurso existente
 - Muestra un formulario para editar los datos de un recurso, como el nombre, el tipo, la disponibilidad, etc.
 - Procesa el formulario y actualiza el recurso con los nuevos datos
 - Permite elegir y añadir o eliminar los servicios de recursos seleccionados
 - Una vez realizada la edición, se devuelve al usuario a la página de recursos con un mensaje de éxito

QuitarResourceView:

- Funcionalidad:
 - Permite eliminar un recurso existente
 - Gestiona las peticiones POST de borrado de un recurso eliminándolo de la base de datos
 - Una vez eliminado el recurso, se devuelve al usuario a la página de recursos con una notificación de que el recurso se ha eliminado correctamente.

ResourceNameView:

- Funcionalidad:
 - Permite añadir un nuevo nombre de fuente.
 - Procesa las solicitudes POST para añadir un nuevo nombre de fuente a la base de datos.
 - Comprueba si el nombre especificado ya existe y lo añade si es único.
 - Muestra un mensaje de éxito si el nombre se ha añadido correctamente, y un mensaje de error si el nombre ya existe o no se ha especificado.
 - Después de añadir, el usuario vuelve a la página de recursos.

ActualizarResourceNameView:

- Funcionalidad:
 - Permite actualizar un nombre de fuente existente.
 - Procesa las solicitudes POST para actualizar un nombre de fuente existente en la base de datos.
 - Recupera el nuevo nombre de la solicitud y actualiza el nombre de origen.
 - Muestra un mensaje de éxito después de actualizar correctamente el nombre.
 - Tras la actualización, el usuario vuelve a la página de recursos.

QuitarResourceNameView:

- Funcionalidad:
 - Permite eliminar un nombre de fuente existente.
 - Procesa las solicitudes POST para eliminar un nombre de fuente existente de la base de datos.
 - Elimina el nombre de la fuente especificada de la base de datos.
 - Muestra un mensaje de éxito después de eliminar con éxito el nombre.
 - Tras la eliminación, se devuelve al usuario a la página de recursos

5.3.9 Promociones y ofertas

Desarrollado por Thomas Verbruggen y yo

mismo **PromotionsView**:

- Funcionalidad:
 - Mostrar promociones en una página
 - Recupera todos los códigos promocionales de la base de datos
 - Establece la posibilidad de buscar códigos promocionales en función de la consulta de búsqueda
 - Muestra la lista de códigos promocionales mediante paginación
 - Ofrece un formulario para añadir nuevos códigos promocionales

CrearVistaPromoción:

- Funcionalidad:
 - Permite a los usuarios añadir un nuevo código promocional
 - Procesa la solicitud POST para crear un nuevo código promocional
 - Muestra un mensaje de éxito después de añadir con éxito el código de promoción

EditarVistaPromoción:

- Funcionalidad:
 - Permite a los usuarios editar un código promocional existente
 - Recupera el código promocional existente para mostrarlo en el formulario de edición
 - Procesa la solicitud POST para actualizar un código promocional existente
 - Muestra un mensaje de éxito tras actualizar correctamente el código promocional

BorrarVistaPromoción:

- Funcionalidad:
 - Permite eliminar un código promocional
 - Procesa la solicitud POST para eliminar un código promocional de la base de datos
 - Muestra un mensaje de éxito después de eliminar correctamente el código promocional

5.3.10 Configuración de la empresa

Desarrollado por Thomas Verbruggen y yo mismo

Ver empresa:

- Funcionalidad:
 - Muestra al usuario la página de información de la empresa
 - Recupera la información de la empresa de la cuenta de usuario que ha iniciado sesión.
 - Muestra información sobre la empresa en el contexto de la página

EditarVistaEmpresa:

- Funcionalidad:
 - Permite a los usuarios editar la información de la empresa
 - Cargar formulario de edición de información de la empresa
 - Recupera la información actual de la empresa para mostrarla en el formulario
 - Procesa la solicitud POST para actualizar la información de la empresa
 - Comprueba y formatea el número de teléfono antes de almacenarlo
 - Muestra un mensaje de éxito después de actualizar correctamente la información de la empresa
 - Redirige a los usuarios a la página de información de la empresa tras una actualización correcta

5.3.11 Reserva (usuario final)

Desarrollado por todo el grupo, Thomas Verbruggen, Joppe Kerkhofs y yo mismo.

addServiceToAppointment:

- Funcionalidad:
 - Verifica que la cita y el servicio especificados existen
 - Comprueba si ya existe un AppointmentService con el mismo servicio
 - Comprueba si el nuevo servicio se ajusta a los servicios ya vinculados de la cita
 - Crea un nuevo AppointmentService si las comprobaciones anteriores tienen éxito
 - Devuelve el AppointmentService creado

endAppointment:

- Funcionalidad:
 - Marca una cita como finalizada
 - Devuelve la cita actualizada

checkWorkloadInTimeFrame:

- Funcionalidad:
 - Calcula la carga de trabajo de un empleado en un plazo determinado
 - Recoge todas las citas y servicios asociados de los empleados
 - Cálculo de la carga de trabajo en función de la duración y la carga de trabajo de los servicios
 - Devuelve la carga de trabajo total

checkIfResourceAvailable:

- Funcionalidad:
 - Comprueba si un recurso está disponible en un plazo determinado
 - Recoge las citas dentro de ese plazo
 - Comprueba si el recurso es utilizado por un AppointmentService dentro de ese periodo de tiempo
 - Devuelve True si el recurso está disponible, False en caso contrario

checkIfResourceAvailable:

- Funcionalidad:
 - Comprueba si el nuevo servicio se ajusta a los servicios ya vinculados de una cita
 - Compara categorías y subcategorías de servicios
 - Devuelve True si el nuevo servicio encaja, en caso contrario False

6 CONCLUSIÓN

Durante mis prácticas en LHS, adquirí valiosos conocimientos sobre la creación de sitios y aplicaciones web. También adquirí un conocimiento más profundo de todo el proceso, que empieza con el análisis y termina con un proyecto totalmente implementado.

También aprendí lo que es trabajar en una empresa y cuántas cosas pueden salir mal aquí, basta pensar en la falta de comunicación o la mala interpretación de temas y asuntos. Esto es algo muy diferente de los proyectos escolares y algo con lo que realmente tuve que aprender a lidiar.

Una de las cosas más importantes para el proyecto fue también trabajar de forma independiente, ya que LHS es una empresa relativamente pequeña, algo que se nos dejó claro desde el principio. Al final, creo que todo salió bien y podemos estar orgullosos del proyecto que hicimos.

En definitiva, mis prácticas en LHS no sólo me han proporcionado conocimientos y habilidades técnicas, sino que también han perfeccionado las que ya tenía. Piensa en la comunicación profesional y la gestión de plazos, son cosas que se enseñan en la escuela, pero que realmente se perfeccionaron para mí durante el periodo de prácticas. Estas experiencias s e r á n muy valiosas para mi futura carrera.

7 POLÍTICA DE IA GENERATIVA

Durante las prácticas se utilizaron herramientas de IA generativa como ChatGPT, Gemini y Phind.

Estos ayudaron durante los escollos que surgieron durante el desarrollo del proyecto, piensa en la resolución de errores, la depuración, También fue útil utilizarlas para

En la fase analítica, estas herramientas nos ayudaron a resolver los dilemas que se nos plantearon, por ejemplo, al diseñar la base de datos o al idear los wireframes.

El uso de estas herramientas nos permitió trabajar con más eficacia y dar mayores saltos en el desarrollo del proyecto. También nos ayudó a desarrollar nuevas habilidades y aumentar nuestros conocimientos al proporcionarnos apoyo y orientación directos en la resolución de problemas.

Para crear mi documento de realización, también utilicé IA generativa, por ejemplo, para la terminología o para corregir errores ortográficos y gramaticales.

En resumen, la IA generativa fue una gran herramienta que me ayudó mucho pero también me enseñó mucho durante este periodo de prácticas.