

BACKEND 1

Bootcamp Presencial

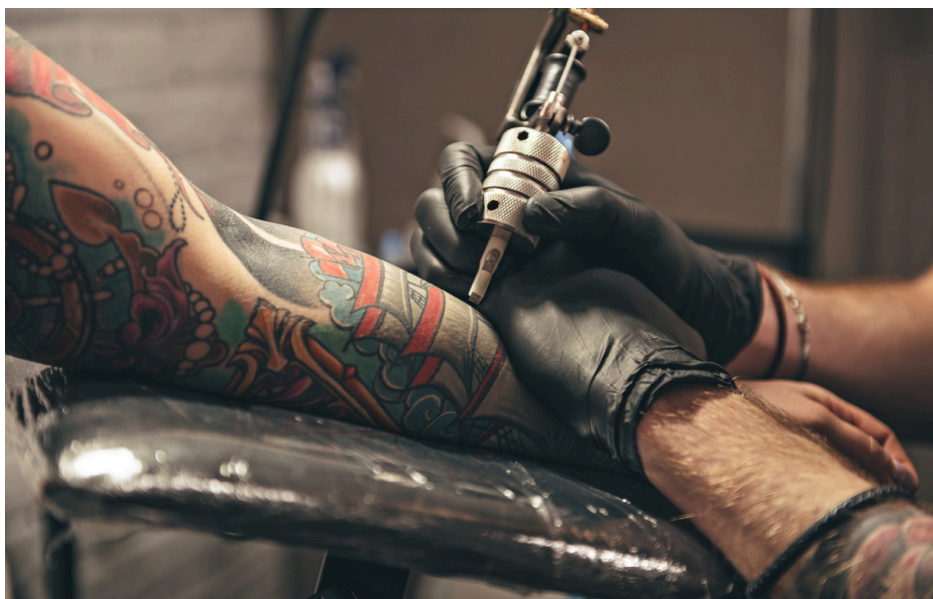


Full Stack Developer

Descripción



GeeksHubs
academy_



Descripción

El departamento de producto nos ha encomendado desarrollar el backend correspondiente al sistema de gestión de citas para un estudio de tatuajes.

Gestión de citas para un estudio de tatuajes

Los usuarios podrán registrarse en la aplicación, iniciar sesión y acceder a su área personal. Dentro de su área, podrán visualizar una lista de las citas programadas para servicios de tatuajes y piercing, así como crear nuevas citas para diversos servicios que ofrece el estudio, tales como:

1. Tatuajes personalizados:

Descripción: Los clientes tendrán la libertad de seleccionar motivos y diseños únicos, personalizando completamente su experiencia de tatuaje de acuerdo a sus preferencias y gustos.

2. Tatuajes del catálogo:

Descripción: Ofrecemos la realización de tatuajes basados en diseños predefinidos en nuestro catálogo. Los clientes pueden elegir entre una variedad de opciones estilizadas y probadas.

3. Restauración y rejuvenecimiento de trabajos:

Descripción: Nos especializamos en la restauración y rejuvenecimiento de tatuajes existentes. Nuestros expertos trabajan para mejorar y renovar tatuajes antiguos, devolviéndoles su vitalidad.

4. Colocación de piercings y dilatadores:

Descripción: Ofrecemos servicios profesionales para la colocación de piercings y dilatadores. Nuestro equipo garantiza procedimientos seguros y estilos variados para satisfacer las preferencias individuales de nuestros clientes.

5. Venta de piercings y otros artículos:

Descripción: Además de nuestros servicios de aplicación, ofrecemos una selección de piercings y otros artículos relacionados con el arte corporal. Los clientes pueden adquirir productos de calidad para complementar su estilo único.

Además, se contará con una sección de usuario donde podrán acceder a sus datos personales, visualizables únicamente por el propio usuario (perfil).

Entregable esperado

Considerando los requisitos, se ha determinado la siguiente lista de endpoints como el mínimo necesario, con la posibilidad de endpoints adicionales, indicados en la columna correspondiente.

Endpoints:

Autenticación

Método	URI	Acción	Extra
POST	/api/auth/register	Registro de usuarios	
POST	/api/auth/login	Login de usuarios	

Usuarios

Método	URI	Acción	Extra
GET	/api/users	Ver todos los usuarios (super_admin)	
GET	/api/users/profile	Ver perfil de usuario	
PUT	/api/users/profile	Modificar datos del perfil (al menos un campo)	
GET	/api/users?email=ejemplo@ejemplo.com	Filtrar usuario por email (super_admin)	SI
DELETE	/api/users/{id}	Eliminar usuario (super_admin)	SI
PUT	/api/users/{id}/role	Cambio de role (super_admin)	SI

Citas

Método	URI	Acción	Extra
POST	/api/appointments	Crear cita	
PUT	/api/appointments	Actualizar mi cita	
GET	/api/appointments/{id}	Recuperar cita	
GET	/api/appointments	Ver mis propias citas	

Servicios

Método	URI	Acción	Extra
GET	/api/services	Ver todos los servicios	
POST	/api/services	Crear servicio (super_admin)	SI
PUT	/api/services/{id}	Modificar servicio (super_admin)	SI
DELETE	/api/services/{id}	Eliminar servicio (super_admin)	SI

Otros aspectos requeridos:

- Migraciones y Seeders para todas las tablas.
- Todos los endpoints no públicos deben tener sus correspondientes middlewares.

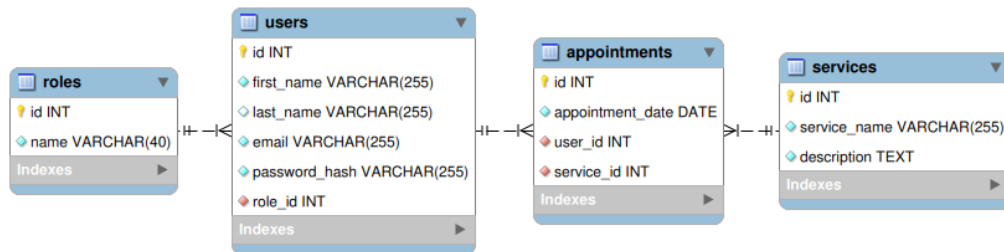
Otros extras

- Validación de la fecha de la cita, evitando solicitar citas con fechas anteriores a la actual.
- Opción para seleccionar tatuador.

Tecnologías

- ORM: TypeOrm.
- Lenguaje: JavaScript ES6 o superior / TypeScript.
- Se valorará el uso avanzado de promesas y async/await.

Estructura y diseño mínimo de la base de datos relacional



Requisitos indispensables para la corrección

- El proyecto debe ser subido a un repositorio público de GitHub, valorándose la existencia de ramas o trabajo en features (Git Flow) y diversos commits que muestren la evolución del proyecto.
- Se requiere un README completo en los repositorios.
- El proyecto debe estar correctamente desplegado en producción.
- El uso de Git requerirá la presencia de ramas y un número adecuado de commits, siendo obligatorio contar con al menos dos ramas principales: "develop" y "master".
- Además, se exigirá que el código a corregir esté ubicado exclusivamente en la rama "MASTER" o "MAIN".

Mejoras

- Se valorará una explicación detallada en el README con ilustraciones y animaciones (GIF, SVG, etc.).
- La legibilidad y número de commits también serán considerados.

Dudas

Para cualquier consulta, dirígete al Slack de Geekshubs en el grupo formativo al que perteneces.