

FRONTEND 1

Tattoo Studio



Bootcamp Presencial

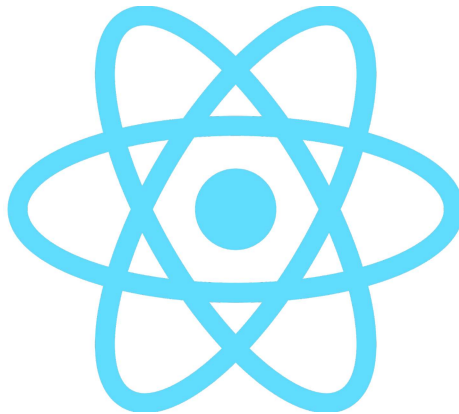
Full Stack Developer



GeeksHubs
academy__

Descripción

En esta ocasión se nos requiere generar la parte frontal que conecte con nuestra API encargada de gestionar el modelo de negocio de un estudio de tatuajes.



Para ello, generaremos unas vistas sencillas que nos permitan registrarnos y logearnos como usuarios, ver nuestro perfil de usuario y modificarlo, así como ver nuestras citas. Además, necesitaremos poder ver los servicios que ofrece el estudio y poder concertar nuevas citas.

Por último, debemos tener una vista de administrador desde la que se podrá acceder a la información de la aplicación y modificarla según proceda.



En definitiva, tendremos que ser capaces de crear una parte frontal de la app lo suficientemente versátil como para que clientes, profesionales y un administrador puedan acceder y hacer uso de ella.

Entregable esperado:

Teniendo en cuenta los requisitos, llegamos a la conclusión de que estas serían las vistas convenientes:

- Vista de inicio / home / landing page
- Vista de registro de usuarios.
- Vista de login de usuarios.
- Vista del perfil de usuario editable.
- Vista que como usuario me permita ver y borrar mis citas.
- Vista de los servicios disponibles
- Vista que como usuario me permita crear nuevas citas.
- Vista como ADMIN que me permita ver y borrar usuarios

Extras:

- Vista de una cita en detalle.
- Vista de un usuario como ADMIN
- Modificación de un usuario como ADMIN.
- Acceso a las citas de un usuario como ADMIN.
- CRUD de servicios como ADMIN.
- Validación de datos (comprobar que los datos introducidos son correctos *antes* de enviarlos al backend, etc).
- Gestión de errores (mensajes informativos desde el formato de contraseña esperado, hasta errores procedentes de backend).

Tecnologías

- Para el desarrollo de esta parte frontal usaremos ReactJs instalado mediante Vite, junto con componentes funcionales, asumiendo buenas prácticas en la medida de lo posible (max 400 líneas por componente y 75 por función).
- Será obligatorio el uso de hooks, props, y para compartir información entre vistas se accederá anualmente a localStorage a través de los métodos `localStorage.setItem('clave', valor)` y `localStorage.getItem('clave')`
- El proyecto se subirá a un repositorio público de GitHub y se valorará la existencia de ramas o trabajo en features (Git Flow), así como diversos commits con la evolución del proyecto.

Mejoras

Se valorará la estética de las vistas, la buena explicación del readme/s con ilustraciones y animaciones (GIF, svg, etc.), y la calidad y cantidad de los commits realizados..

Requisitos indispensables para la corrección

Será requisito indispensable que los repositorios dispongan de un README completo.

Dudas

Para cualquier duda puedes dirigirte al Slack de Geekshubs. En el grupo formativo al que perteneces.