SPS Solutions

En esta guía se proporciona información específica para los equipos con el objetivo de ayudarte a comenzar de forma efectiva y productiva el entendimiento de Eagle Wear. A continuación está el índice donde podrán visualizar las implementaciones realizadas.

Dependencias

Versiones

Para el proyecto de **Eagle Wear** requieren las siguientes versiones para su ejecución. Este proyecto está creado a través de React con la implementación de Next.js, en este caso se utilizó esto ya que el sistema debía cumplir con las características de una Web App, por lo cual recomiendan que para ese tipo de ejecuciones se utilice esta integración.

1. Node.js: **^20.17.0 o superior**
2. NPM: **^10.9.2 o superior**
3. React: **^19.0.0**
4. Next: **^15.2.2**

Dependencias

Tenemos un listado de dependencias que se requirieron para la ejecución de funcionalidades dentro de Eagle Wear.

1. **@heroicons**: Se utilizó para la inserción de iconos.
2. **@mui/material**: Utilización para componentes específicos dentro del UI.
3. **@hookform/resolvers**: Utilizado para la validación y creación de formularios.
4. **axios**: Permite el posicionamiento y ejecución de servicios HTTP.
5. **dotenv**: Se utilizó para la configuración de environments.
6. **jest**: Permite la ejecución de las pruebas unitarias.
7. **@testing-library/react**: Permite la ejecución de las pruebas unitarias.
8. **Tailwind**: Permitió la implementación del diseño responsivo y adaptativo.

Pantallas UI&UX

Visuales

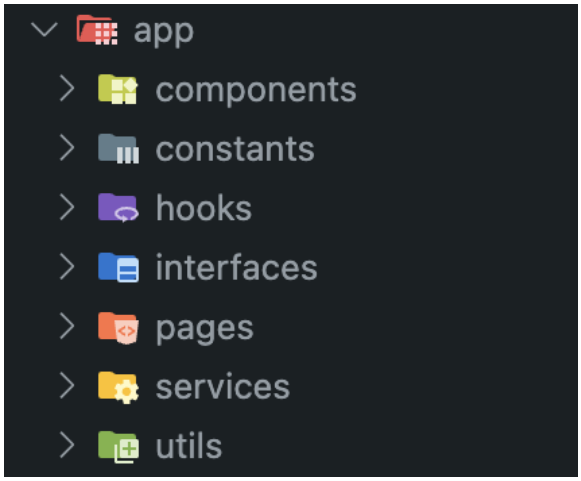
Dentro de Eagle Wear encontraremos las siguientes pantallas:

1. Dashboard
2. Detalle del Producto
3. Carrito de Compras
4. Autenticación:
    a. Iniciar Sesión
    b. Registro

Estructura del Proyecto

La estructura del proyecto está definida por documentación de React o documentaciones externas que nos explican que tener una buena estructura dentro del proyecto permite generar una mayor organización y poder expandir el proyecto.

La estructura implementada es la siguiente:



Deployment

Environments

La configuración de ambientes fue gracias a la implementación de @**dotnev**, el cual permite ejecutar diferentes variables de entorno, pero en la práctica real estas variables son configuradas a través del proveedor de servicios en la Nube o tomadas desde un archivo .env. A continuación nuestro los environments pre-configurados para la ejecución.

Deploy

La web está deployada en un ambiente **AWS Amplify** para la ejecución rápida de servicios Frontend, el ambiente se encuentra en la siguiente URL por lo cual tendrá un tiempo de uso ya que la ejecución y los gastos corren por parte del candidato.

AWS Amplify URL: **https://main.d1vzatn206s6ma.amplifyapp.com/**

NOTA: Por favor de notificar cuanto antes haya sido revisado ya que el ambiente el costo corre por parte mía.

Modularización

Dentro del proyecto se utiliza una modularización de rápido acceso para **Components, Hooks, Services, Utils e interface**s, con el propósito de tener un fácil acceso y tener una buena práctica de código limpio al momento de importar algún componente de estos.

```
You, hace 27 segundos | 1 author (You)
export { default as Layout } from './Layout/Layout';
export { default as Navbar } from './Navbar/Navbar';
export { default as Product } from './Product/Product';
export { default as ProductItem } from './Product/Product';
export { default as SnackbarAlert } from './Snackbar/Snackbar';
```

```
"paths": {
  "@/*": [
    "./*"
  ],
  "@Components/index": [
    "./app/components/index.ts"
  ],
  "@Hooks/index": [
    "./app/hooks/index.ts"
  ],
  "@Interfaces/*": [
    "./app/interfaces/*"
  ],
  "@Utils/*": [
    "./app/utils/*"
  ],
  "@Services/index": [
    "./app/services/index.ts"
  ]
}
```

Fake Store API

Implementación

En el consumo de la API de Fake Store no hubo ninguna complicación, sin embargo carece de funcionalidades para el requerimiento. El consumo se ubica dentro de **Services/AuthService**, donde permite el posicionamiento a través de **Axios.**

```
const AuthService = {
    setInformation: (key: string, request: any) => localStorage.setItem(key, JSON.stringify(request)),
    Complexity is 9 It's time to do something...
    login: async (request: IAuthLoginUser) => {  …
    },
    Complexity is 9 It's time to do something...
    register: async (request: IAuthRegisterUser) => {  …
    },      You, hace 24 horas • feat: register and logout done
    logout: () => {…
    },
}
```

Testing

Implementación

Las pruebas unitarias corren a través de Jest & React Testing Library, estas mismas están ya pre configuradas sin embargo no logré ejecutarlas de manera exitosa ya que tengo un bloqueante que al momento de yo llamar algún componente, jest no lo encuentra y truena todo el proyecto. Sin embargo si se ejecuta alguna prueba unitaria que no importe ningún componente pasa de manera limpia.

Recursos

Links

Aqui dejare los links de apoyo tanto de dependencias, librerías, documentación de la cual me fui apoyando para ir solucionando los errores y implementaciones, así como recursos externos e internos que pudiera encontrar en internet.

1. https://react.dev/
2. https://tailwindcss.com/plus/ui-blocks
3. https://fakestoreapi.com/
4. https://www.npmjs.com/package/react-dotenv
5. https://headlessui.com/
6. https://heroicons.com/
7. https://mui.com/
8. https://www.npmjs.com/package/axios
9. https://react-hook-form.com/
10. https://www.npmjs.com/package/@testing-library/dom
11. https://www.npmjs.com/package/@testing-library/jest-dom
12. https://testing-library.com/docs/react-testing-library/intro/
13. Amazon Web Services
14. Github
15. Chat GPT
16. Visual Studio Code
17. Postman

Best Greetings

Mis mejores saludos a todo el equipo de SPS Solutions. Agradezco la oportunidad de continuar en el proceso de selección y valoro el tiempo y la atención que han dedicado a evaluar mi perfil. Del mismo modo, espero poder cumplir con las expectativas técnicas y profesionales que buscan para esta posición, aportando mis conocimientos y experiencia para contribuir al éxito del equipo. Quedo atento a cualquier retroalimentación o paso siguiente en el proceso, con la mejor disposición para continuar avanzando. Esperando que la calidad del proyecto, código, documentación y demás pudieran cumplir pero sobre todo disipar las dudas respectivas a la tecnología, aún me queda mucho por aprender y que más encantado de poder aprender a lado de ustedes. Siempre abierto al aprendizaje nuevo.

Por tiempo y disposición mía, en el proyecto hubo cosas que carecieron de funcionalidad y sobre todo de dedicación sin embargo puse mi mayor esfuerzo y dedicación para poder desarrollar e implementar la mayor parte de este pequeño proyecto.

Aprecio mucho esta oportunidad y agradezco de antemano su tiempo y consideración.

¡Muchas gracias y quedo a su disposición!

- Cesar Candia