

WRITEUP

Frontend:

- The frontend of the web page is built with **React**, a JavaScript library for creating user interfaces.
- The frontend consists of four main components: **Navbar**, **Card**, **CartPage**, and **AdminPanel**. Each component has its own JSX file and CSS file for rendering and styling.
- The **Navbar** component displays the logo, the name, the cart button, and the sign in or sign out buttons. It also renders the **LoginModal** and **SignInModal** components for authentication.
- The **Card** component displays the medical items in a grid layout. Each card shows the image, name, description, price, and add to cart button of an item. It also dispatches the **addToCart** action to add an item to the cart.
- The **CartPage** component displays the cart items in a table. It also shows the total cost and the confirm and clear buttons. It dispatches the **getCartItems**, **addToCart**, and **deleteCartItem** actions to manage the cart state.
- The **AdminPanel** component allows the admin to manage the categories and items. It can add, delete, or edit a category or an item. It dispatches the **getCategories**, **addCategory**, **deleteCategory**, **getItems**, **addItem**, and **deleteItem** actions to interact with the backend.

Backend:

- The backend of the web page is built with **ASP.NET Core**, a cross-platform framework for building web applications.
- The backend consists of four main controllers: **MedCartsController**, **MedCategoriesController**, **MedItemsController**, and **UserControlsController**. Each controller handles the HTTP requests for a specific entity.

- The **MedCartsController** handles the requests for the med carts. It can get, add, update, or delete a med cart. It also includes the med items navigation property in the response.
- The **MedCategoriesController** handles the requests for the med categories. It can get, add, update, or delete a med category.
- The **MedItemsController** handles the requests for the med items. It can get, add, update, or delete a med item. It also links the item to the existing category before creating it..
- The **UserControlsController** handles the requests for the user controls. It can get, add, update, or delete a user control. It also has an authentication endpoint that checks the username and password of a user.

The backend uses an ApplicationDbContext class that inherits from DbContext to manage the database connection and operations. The backend defines four models: MedCart, MedCategory, MedItems, and UserControl. Each model has its own properties and annotations.