Herkese merhaba! Ben,

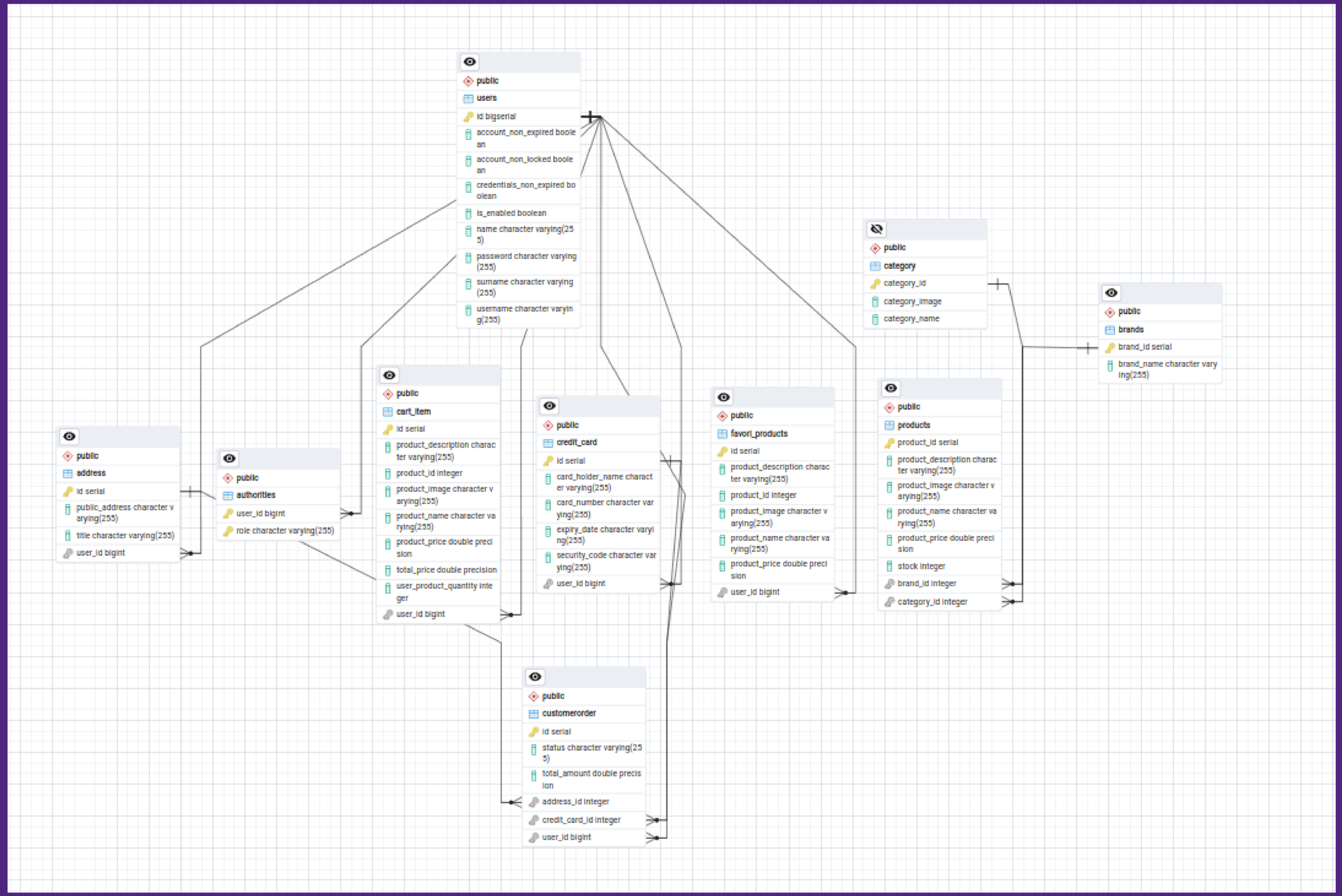
Deniz Süllü

KULLANDIĞIM TEKNOLOJİLER

Backend

Front-End

DATABASE TASARIMI



ENDPOİNT

user-controller

- PUT /auth/user/update
- POST /auth/register
- POST /auth/login
- GET /auth/user/{username}
- GET /auth/user/all

products-controller

- PUT /api/products/update
- POST /api/products/add
- POST /api/products/addMultiple
- GET /api/products/{id}
- DELETE /api/products/{id}
- GET /api/products/getAll
- GET /api/products/getAllByCategoryId/{id}

ENDPOİNT

order-controller

- PUT /api/orders/changeorderstatus/{orderId}/{status}
- POST /api/orders/create
- GET /api/orders/getallbyuserid/{userId}
- GET /api/orders/getall

categories-controller

- PUT /api/categories/update
- POST /api/categories/add
- POST /api/categories/addAll
- GET /api/categories/{id}
- GET /api/categories/getAll
- DELETE /api/categories/delete/{id}

brands-controller

- PUT /api/brands/update
- POST /api/brands/add
- GET /api/brands/{id}
- DELETE /api/brands/{id}
- GET /api/brands/getAll

ENDPOİNT

favori-product-controller

- POST /api/favoriteProducts/add
- GET /api/favoriteProducts/getByUser/{userId}
- DELETE /api/favoriteProducts/delete/{favoriProductId}

credit-card-controller

- POST /api/creditcard/add
- GET /api/creditcard/getall/{userId}
- DELETE /api/creditcard/delete/{creditCardId}

cart-item-controller

- POST /api/cart/increase
- POST /api/cart/decrease
- POST /api/cart/add
- GET /api/cart/get/{username}
- DELETE /api/cart/{id}
- DELETE /api/cart/delete/{userid}

address-controller

- POST /api/address/save
- GET /api/address/getall/{userId}
- DELETE /api/address/delete/{addressId}

Endpoint security & Cache mechanism

```
deniz
@GetMapping("/get/{username}")
@PreAuthorize("hasAnyRole('ROLE_ADMIN','ROLE_USER')")
public List<GetAllCartItemResponse> getByUsername(@PathVariable("username") int userid){
    return cartService.getAllCartItems(userid);
}
```

```
deniz
@GetMapping("/getAllByCategoryId/{id}")
@Cacheable(value = "productsCategory")
public List<GetAllProductResponse> getAllByCategoryId(@PathVariable int id) {
    return this.productService.getAllByCategoryId(id);
}
```

Frond-End Kullanılan Yapılar

- Tailwind CSS
- Angular Material
- Reactive Forms
- NGX Owl Carousel
- NGX Toastr
- RxJS

Hangi Özellikler var?

- Guard Yapısı: Kullanıcı ve admin tabanlı erişim kontrolü.
- Interceptor: JWT için header kısmına token ekleme.
- Lazy Loading & Preloading: Performans ve hızlı yükleme.
- Angular 16 ve 17 Özellikleri: En son Angular özellikleri kullanıldı.

Login

```
1+ usages   deniz
login(user: LoginModel): Observable<UserDetail> {
    return this.httpClient.post<TokenModel>( url: environment.apiEndpoint + "auth/login", user).pipe(
        switchMap( project: tokenModel : TokenModel  => {
            this.setSession(tokenModel.token);
            return this.userService.getUserDetails(this.tokenService.getUserDetailsFromToken().sub);
        }),
        tap( observerOrNext: userData : UserDetail  => {
            this.updateUserState(userData);
        })
    );
}
```

LayzLoading

```
{
    path: 'user',
    loadChildren: () => import('./components/user/user.routes').then(m => m.routes),
    canActivate: [loginGuard]
},
{
    path: 'products',
    loadComponent: () => import("./components/products/pages/product.component").then(m => m.ProductComponent)
},
{path: 'product/:productId',
    loadComponent: () => import("./components/products/components/product-details/product-details.component").then(m => m.ProductDetails
{path: 'products/category/:categoryId',
    loadComponent: () => import("./components/products/pages/product.component").then(m => m.ProductComponent)},
{path: 'login',
    loadComponent: () => import("./components/auth/components/login/login.component").then(m => m.LoginComponent)},
{path: 'register',
    loadComponent: () => import("./components/auth/components/register/register.component").then(m => m.RegisterComponent)},
{path: 'cart',
    loadComponent: () => import("./components/cart/cart.component").then(m => m.CartComponent)},

{path: 'checkout', component: CheckoutComponent, canActivate: [loginGuard]},
```

Angular 16 Component Input Binding

```
  //withComponentInputBinding
@Input() productId:number;
  no usages  deniz
  ngOnInit(): void {
    this.getProductDetail(this.productId)
  }
  1+ usages  deniz
  getProductDetail(productId: number) : void  {
    this.productService.getProductDetail(productId).subscribe( observerOrNext: {
      next: (response : Product ) : void  => {
        this.product = response;
      },
      error: (error) : void  => {
        console.error('Ürün detayı yüklenirken bir hata oluştu:', error);
      },
      complete: () : void  => {
      }
    });
```

```
export const appConfig: ApplicationConfig = {
  providers: [
    provideRouter(routes,withComponentInputBinding()), provideClientHydration(),
    provideHttpClient(withInterceptors( interceptorFns: [authInterceptor])), provideAnimations(),
    provideToastr( config: {
      positionClass: "toast-top-right",
    }),
```

Angular 17 defer ve Control Flow

```
<div class=" grid grid-cols-1 gap-x-6 gap-y-10 sm:grid-cols-2 lg:grid-cols-4 xl:gap-x-8">
  @for (product of products; track product.productId) {
    @defer (on viewport) {
      <div class="group relative border rounded-lg">
        <div routerLink="/product/{{product.productId}}"
             class="aspect-h-1 aspect-w-1 w-full overflow-hidden rounded-md lg:aspect-none group-hover:opacity-75
              cursor-pointer">
          <img [src]="product.productImage" [alt]="product.productDescription"
               class="object-cover object-center h-full w-full">
        </div>
        <div class="mt-4 flex flex-col">
          <div class="flex justify-between px-4">
            <a class="truncate">
              {{ product.productName | titlecase }}
            </a>
            <p class="text-sm font-medium text-gray-900">{{ product.productPrice |
              currency: currencyCode: 'TL': display: 'symbol' }}</p>
          </div>
        </div>
        <button (click)="addToCart(product)"
                class="w-full bg-secondary-brand-color text-white p-3 rounded bottom-0 transition-all items-center
                 justify-center font-medium hover:bg-brand-color">
          Sepete Ekle
        </button>
```

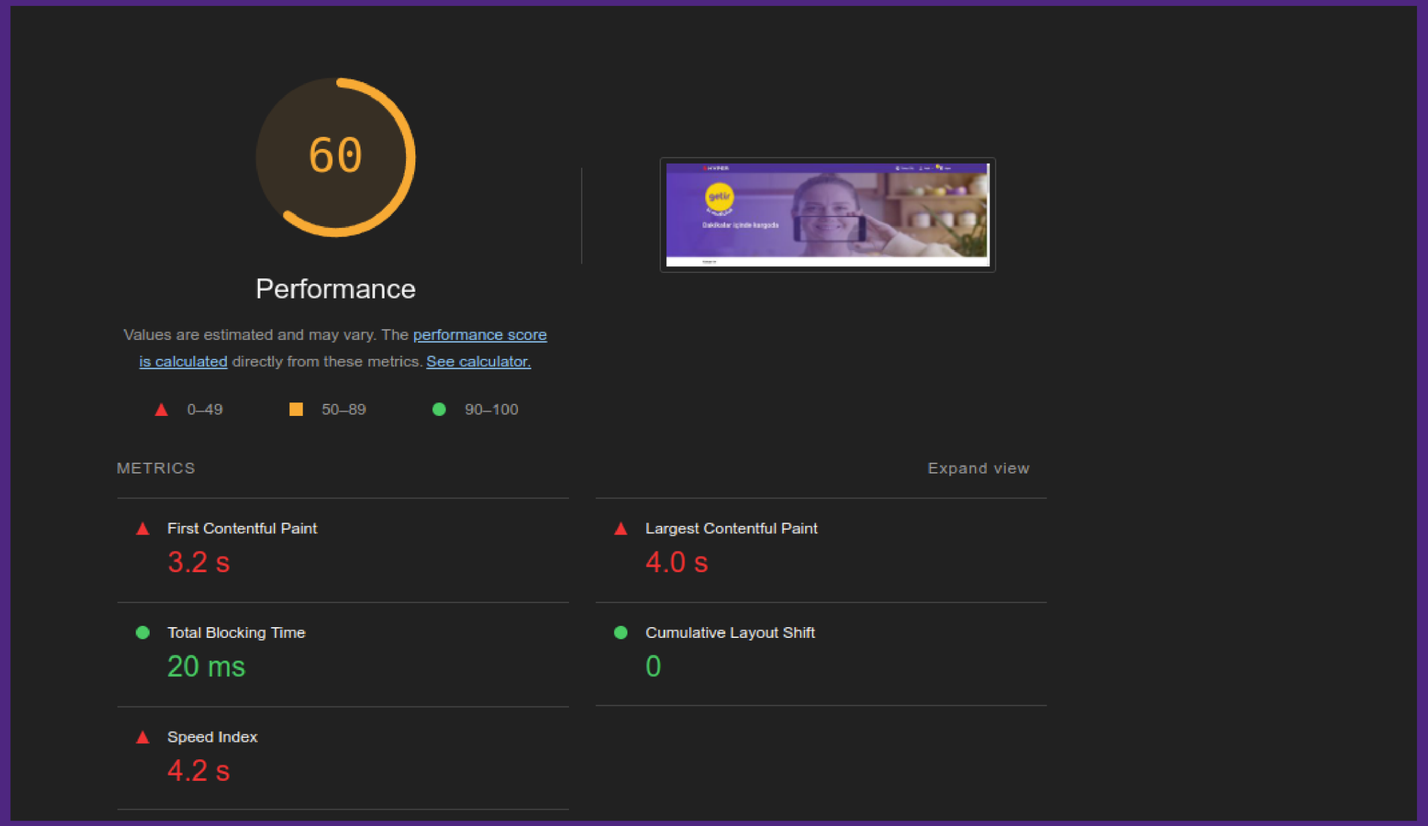
Interceptor

```
export const authInterceptor: HttpInterceptorFn = (req : HttpRequest<?> , next : HttpHandlerFn ) => {

    let token : string  = localStorage.getItem( key: 'token');
    if (token) {
      req = req.clone( update: {
        headers: req.headers.set('Authorization', 'Bearer ' + token)
      });
    }


  return next(req);
};
```

Neyi farklı yapardım?

- Bootstrap Kullanımı
- Temiz Kod Yazımı
- Signal kullanımı
- NgRx kullanımı

Performans



- SSR
- Büyük Boyutlu Görseller
- Test ortamı

Projede bundan sonra neler değişecek?

- NgRx
- Search Api(Elasticsearch)

Teşekkür ederim.