Project Screenshots

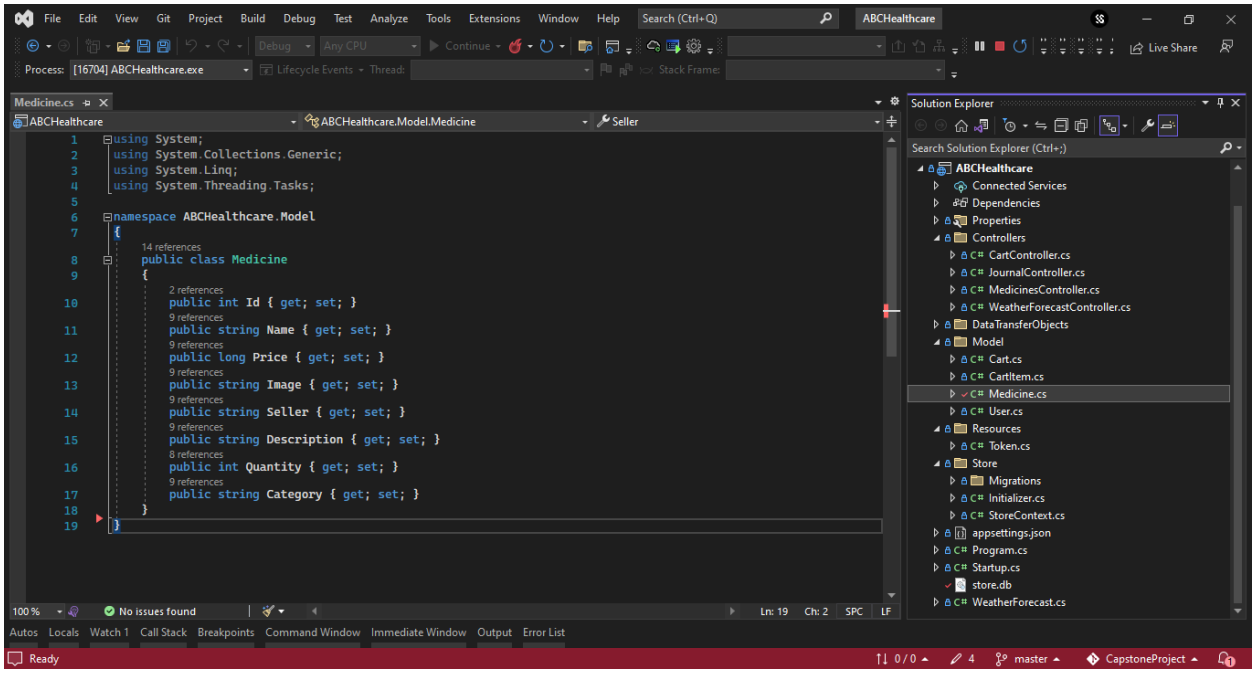


Fig. 1: Model/Medicine.cs

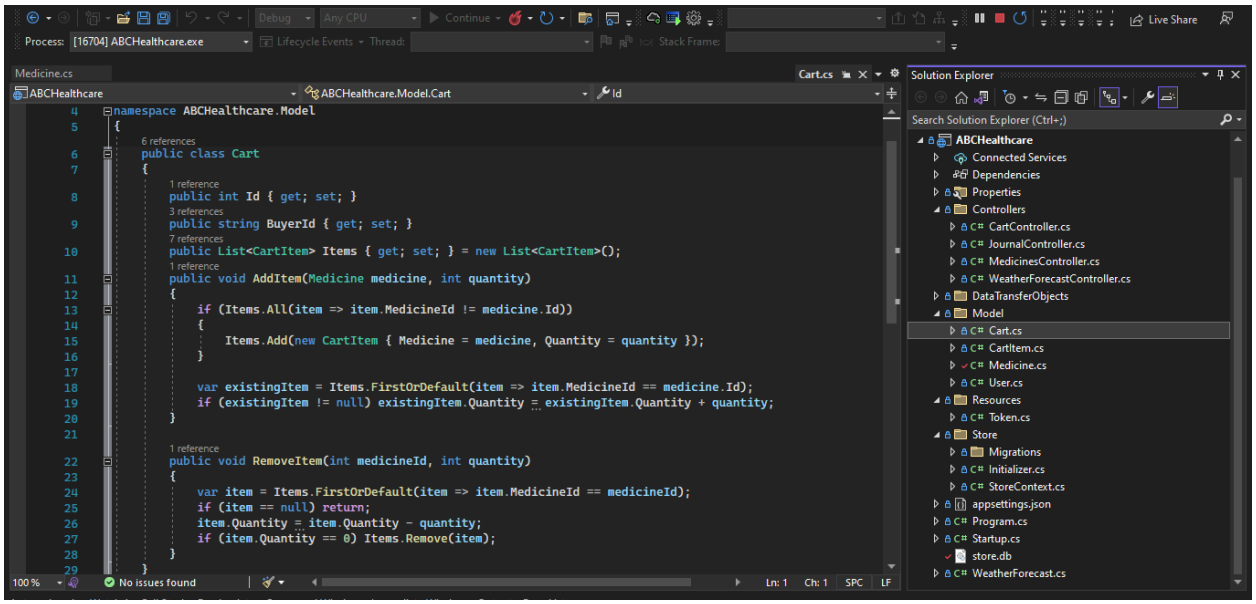


Fig. 2: Model/Cart.cs

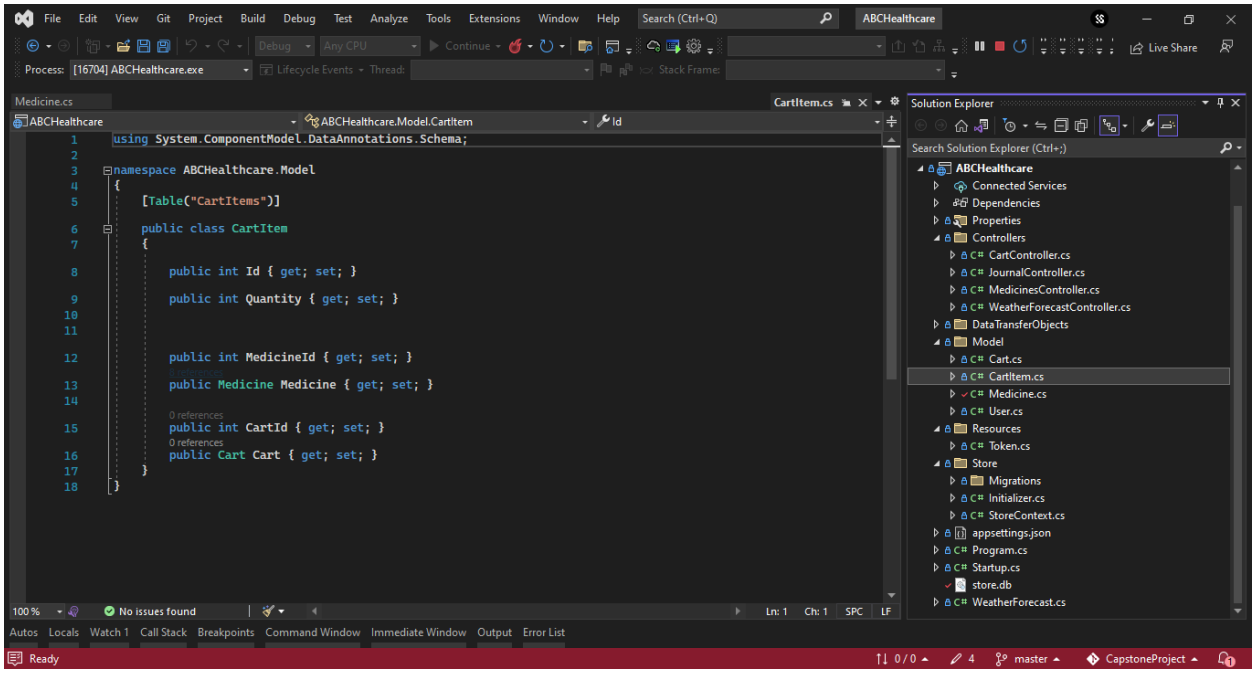


Fig. 3: Model/CartItem.cs

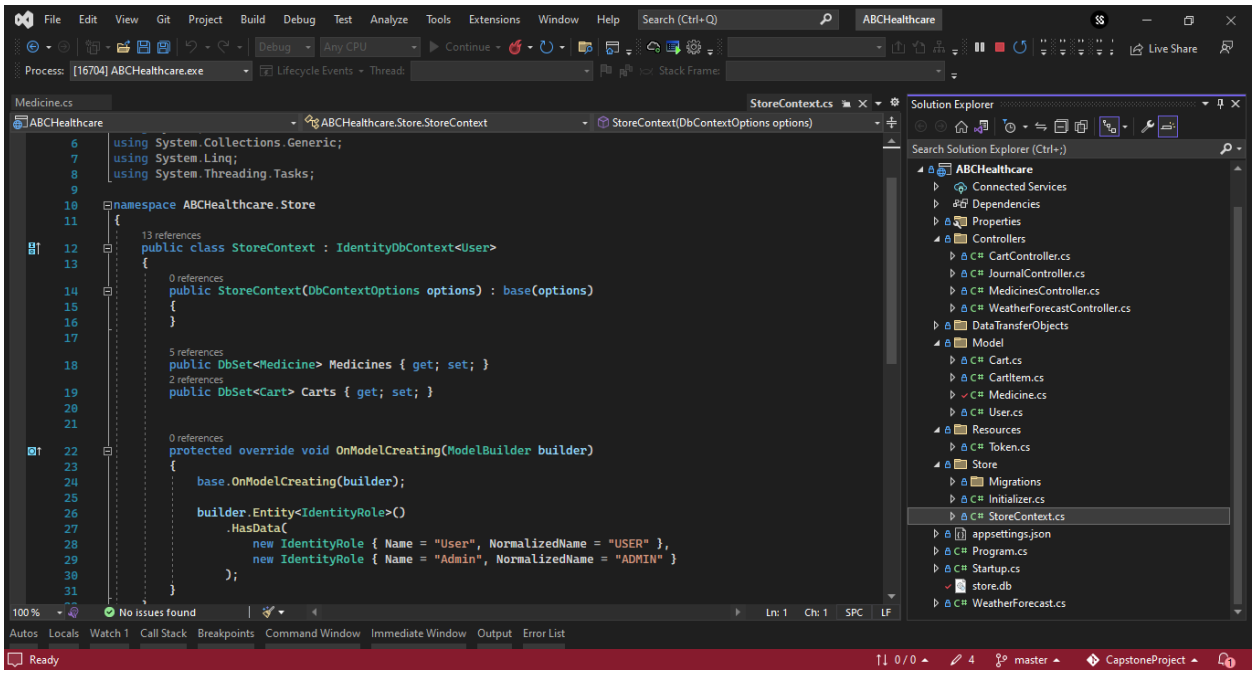


Fig. 4: StoreContext.cs

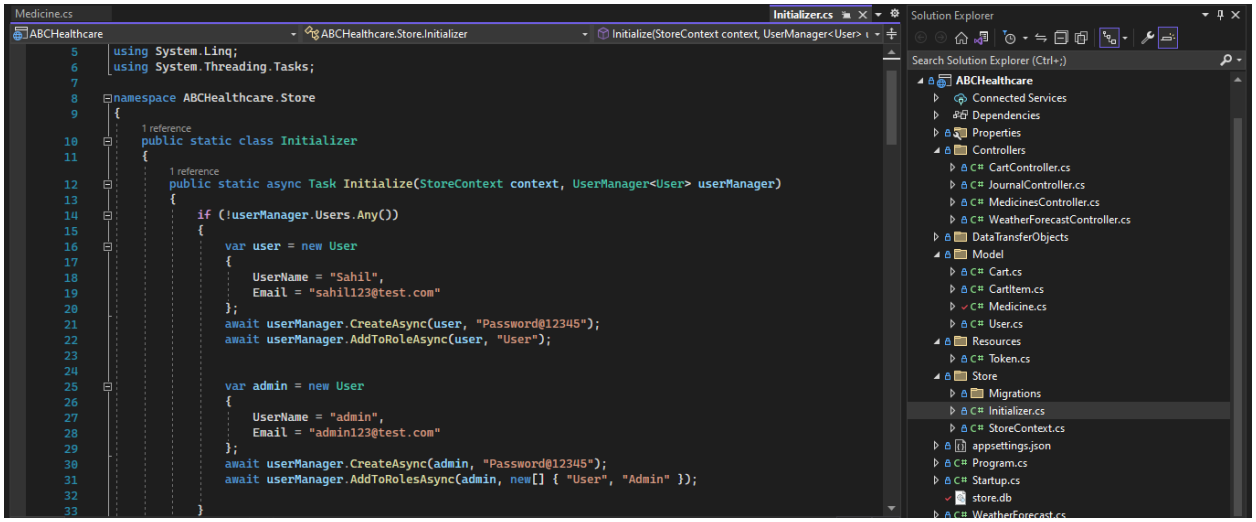


Fig. 5: Initializer.cs

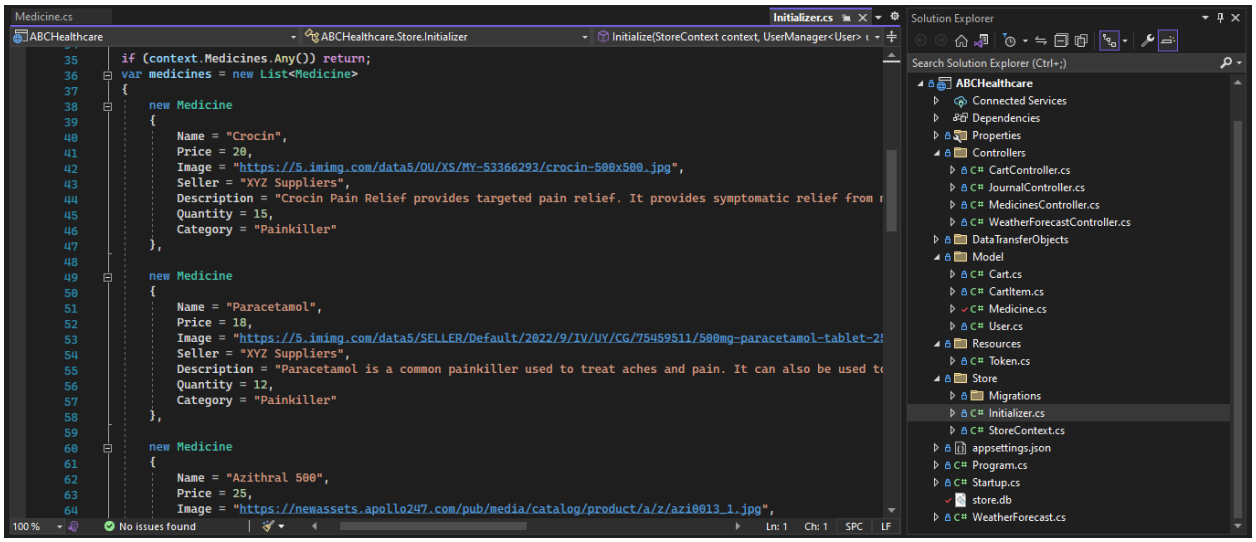
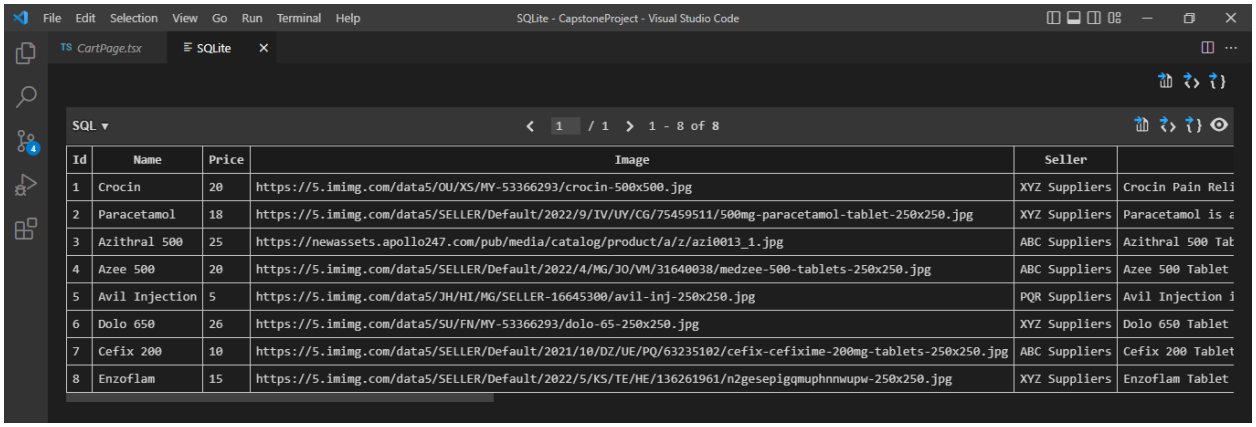


Fig. 6: Initializer.cs (contd…)



Id	Name	Price	Image	Seller	
1	Crocin	20	https://5.imimg.com/data5/OU/XS/MY-53366293/crocin-500x500.jpg	XYZ Suppliers	Crocin Pain Reli
2	Paracetamol	18	https://5.imimg.com/data5/SELLER/Default/2022/9/IV/UY/CG/75459511/500mg-paracetamol-tablet-250x250.jpg	XYZ Suppliers	Paracetamol is a
3	Azithral 500	25	https://newassets.apollo247.com/pub/media/catalog/product/a/z/azi0013_1.jpg	ABC Suppliers	Azithral 500 Tab
4	Azee 500	20	https://5.imimg.com/data5/SELLER/Default/2022/4/MG/JO/VM/31640038/medzee-500-tablets-250x250.jpg	ABC Suppliers	Azee 500 Tablet
5	Avil Injection	5	https://5.imimg.com/data5/JH/HI/MG/SELLER-16645300/avil-inj-250x250.jpg	PQR Suppliers	Avil Injection i
6	Dolo 650	26	https://5.imimg.com/data5/SU/FN/MY-53366293/dolo-65-250x250.jpg	XYZ Suppliers	Dolo 650 Tablet
7	Cefix 200	10	https://5.imimg.com/data5/SELLER/Default/2021/10/DZ/UE/PQ/63235102/cefix-cefixime-200mg-tablets-250x250.jpg	ABC Suppliers	Cefix 200 Tablet
8	Enzoflam	15	https://5.imimg.com/data5/SELLER/Default/2022/5/KS/TE/HE/136261961/n2gesepigqmuphnnwupw-250x250.jpg	XYZ Suppliers	Enzoflam Tablet

Fig. 7:.Medicine seed data reflected in database

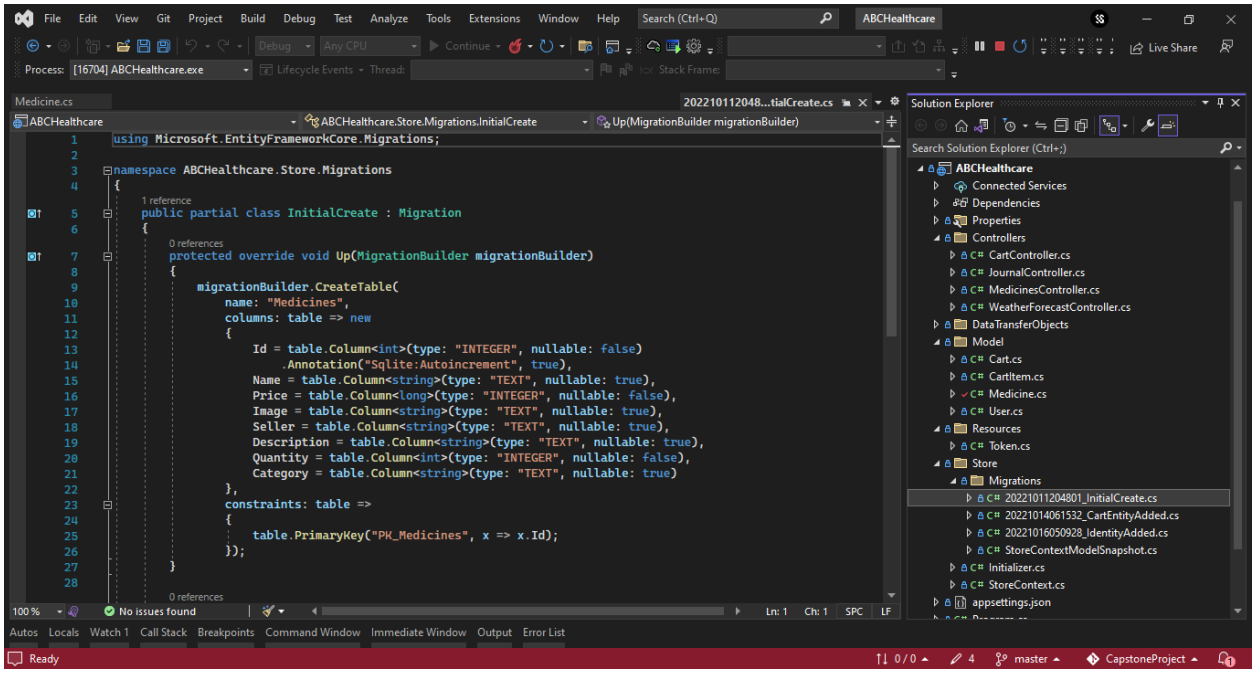


Fig. 8: InitialCreate.cs

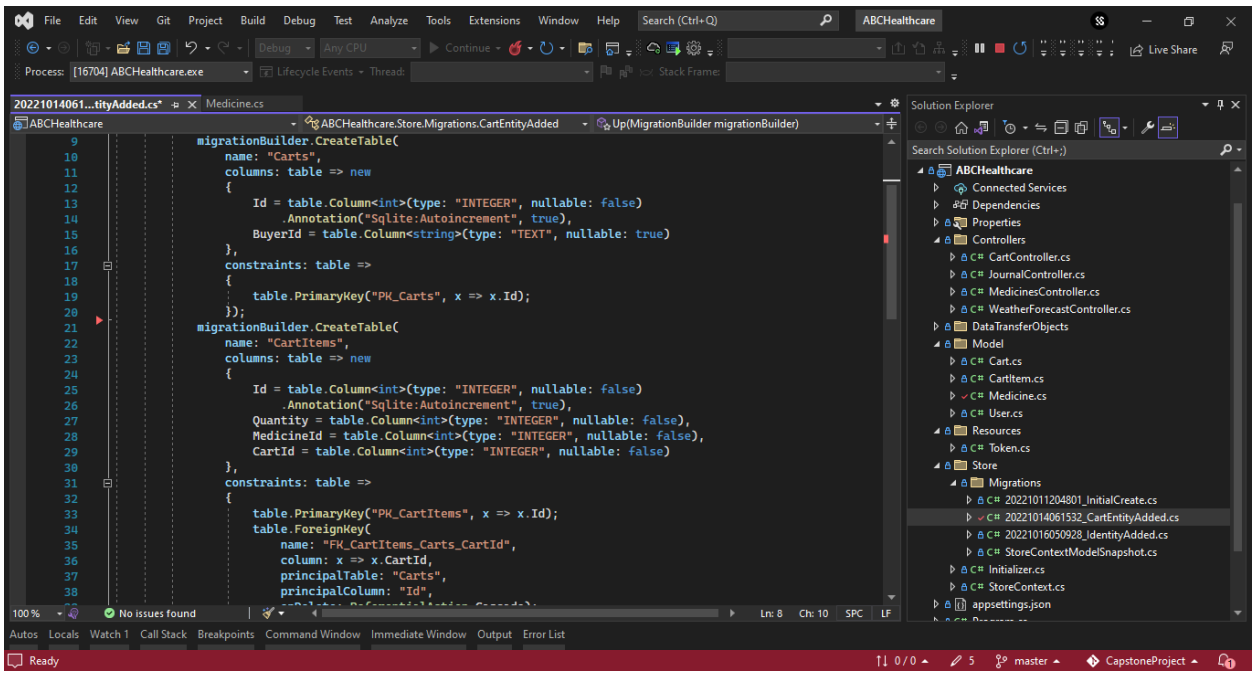


Fig. 9: InitialCreate.cs

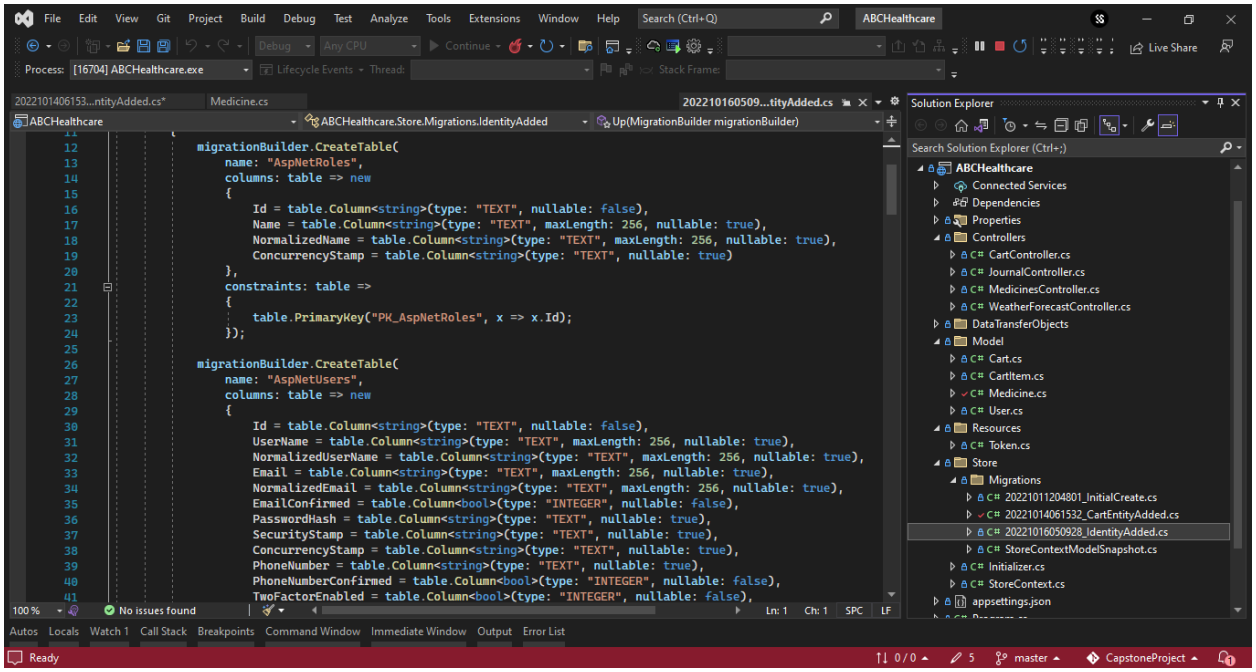
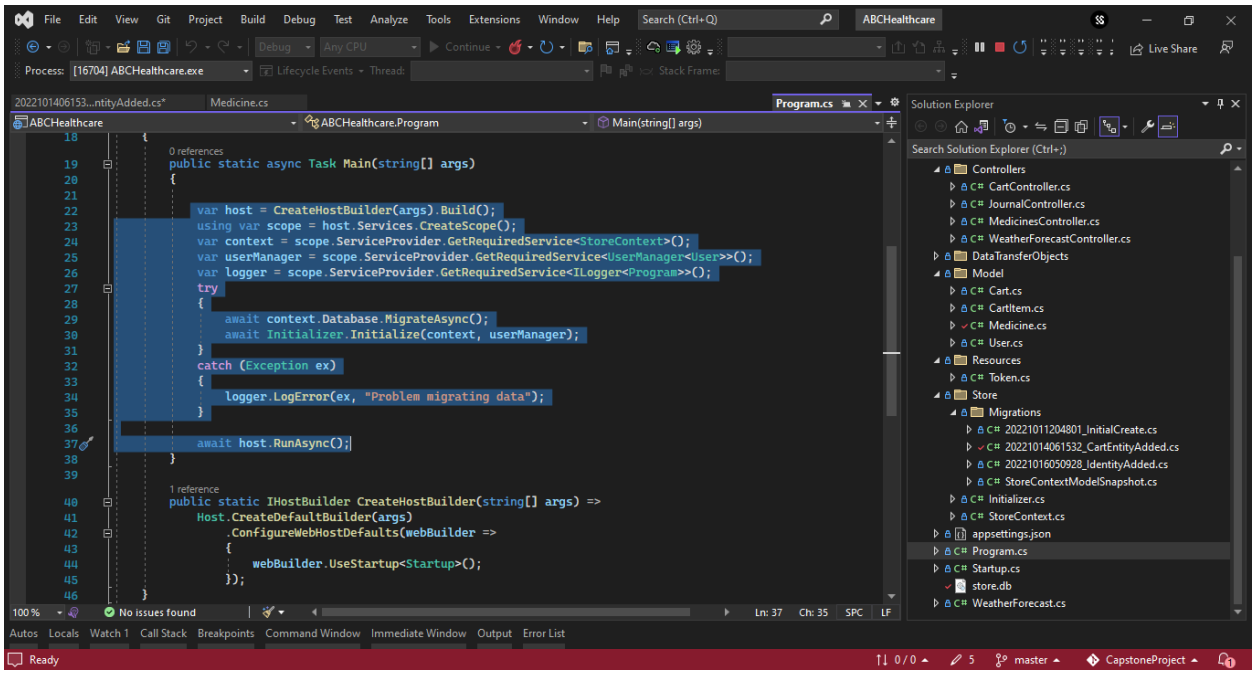


Fig. 10: IdentityAdded.cs



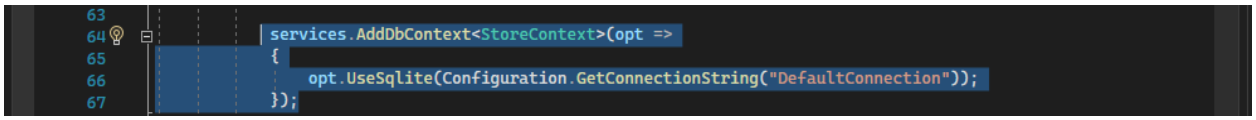


Fig. 11: Program.cs

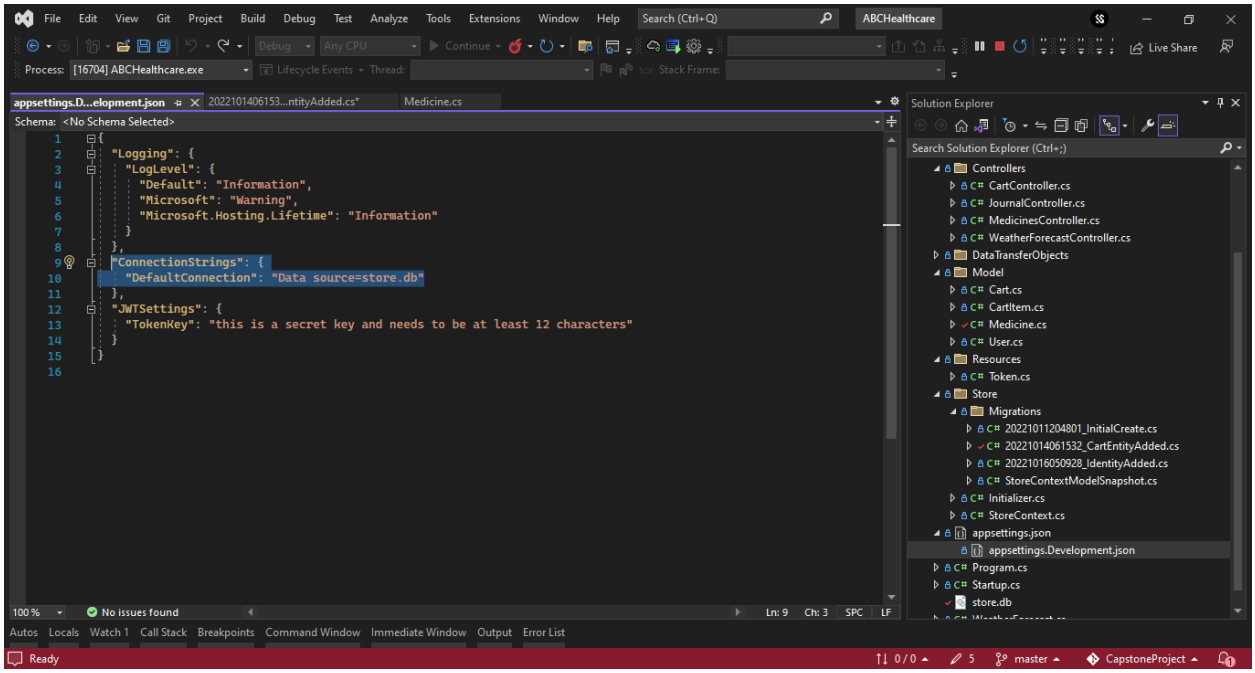


Fig. 12: appsettings.Development.json

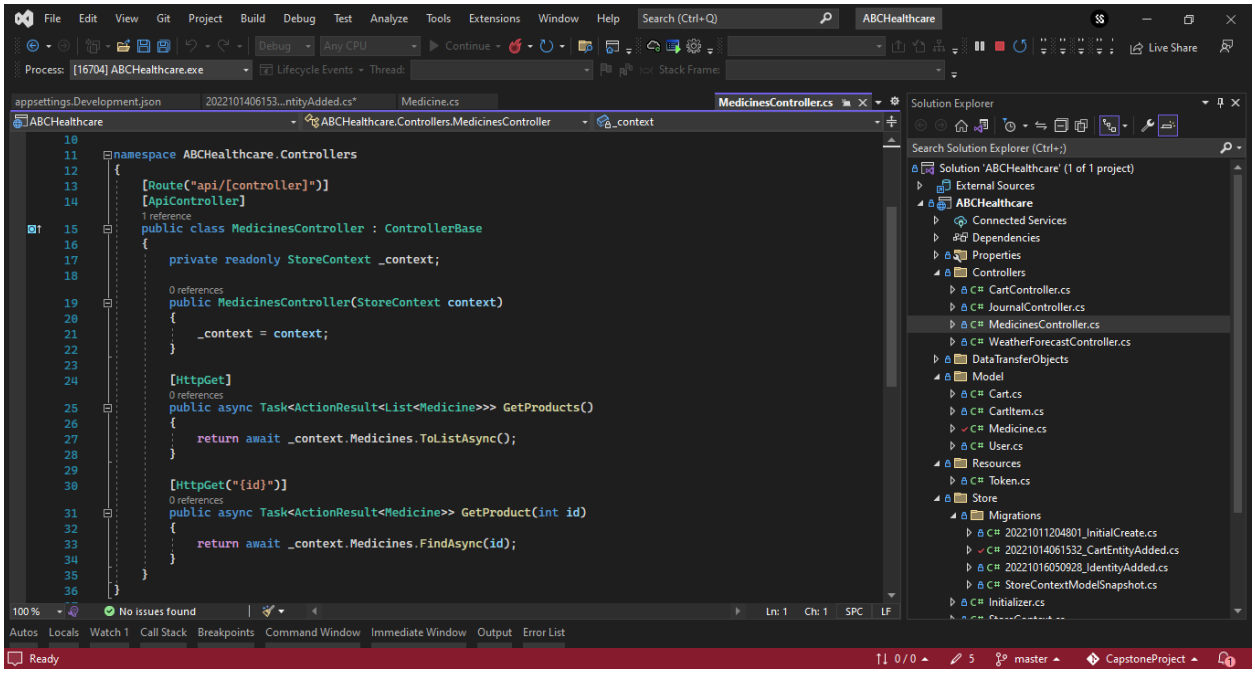
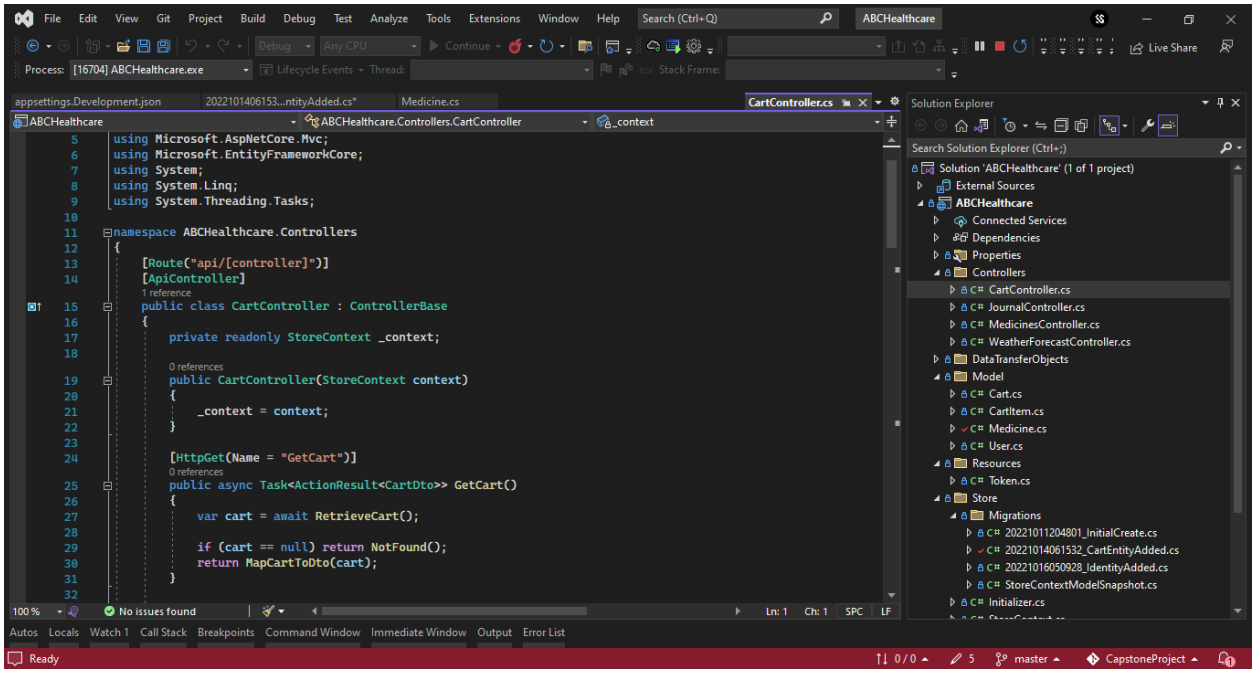
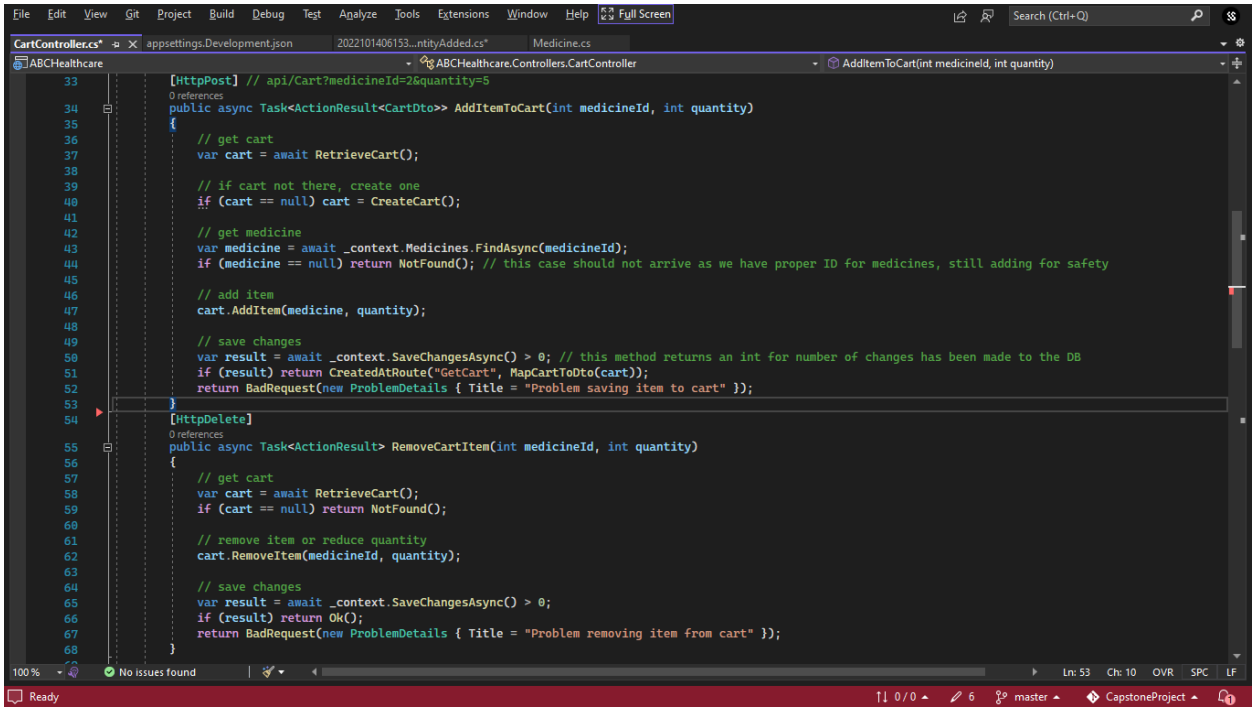


Fig. 13: Controllers/MedicineController.cs



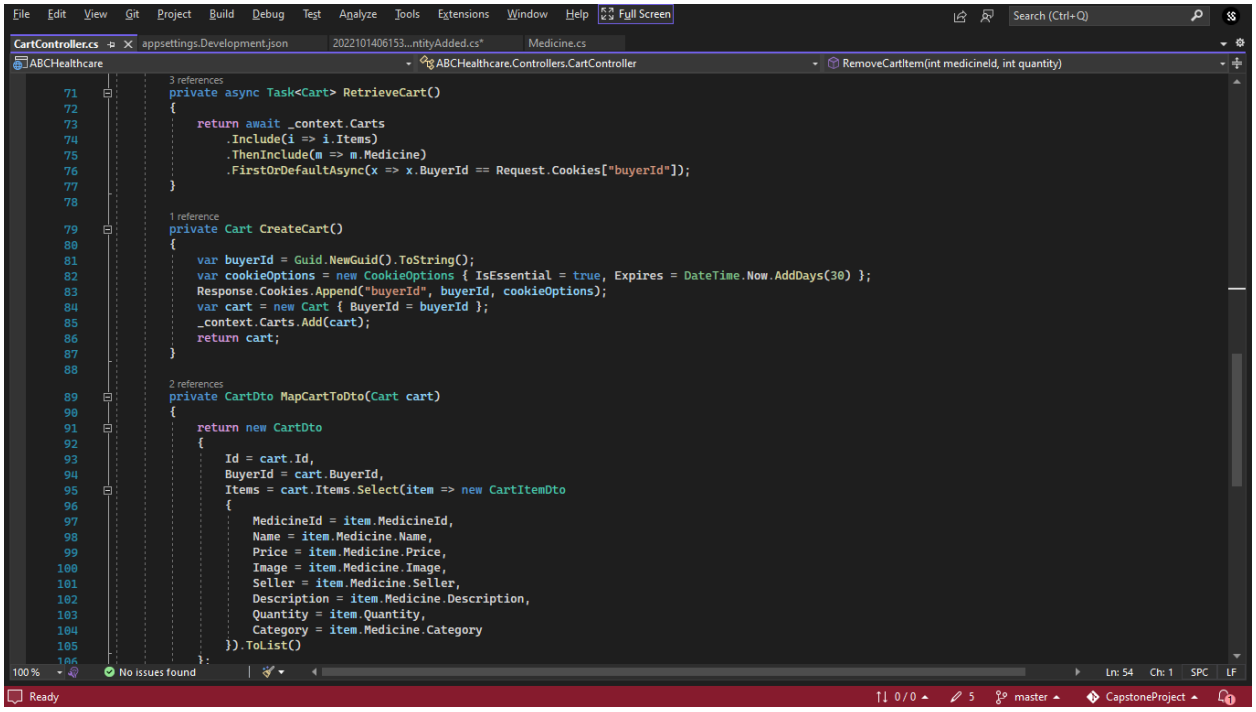
```
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using System;
using System.Linq;
using System.Threading.Tasks;

namespace ABCHealthcare.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    1 reference
    public class CartController : ControllerBase
    {
        private readonly StoreContext _context;

        0 references
        public CartController(StoreContext context)
        {
            _context = context;
        }

        [HttpGet(Name = "GetCart")]
        0 references
        public async Task<ActionResult<CartDto>> GetCart()
        {
            var cart = await RetrieveCart();

            if (cart == null) return NotFound();
            return MapCartToDto(cart);
        }
```

Fig. 14: Controllers/CartController.cs



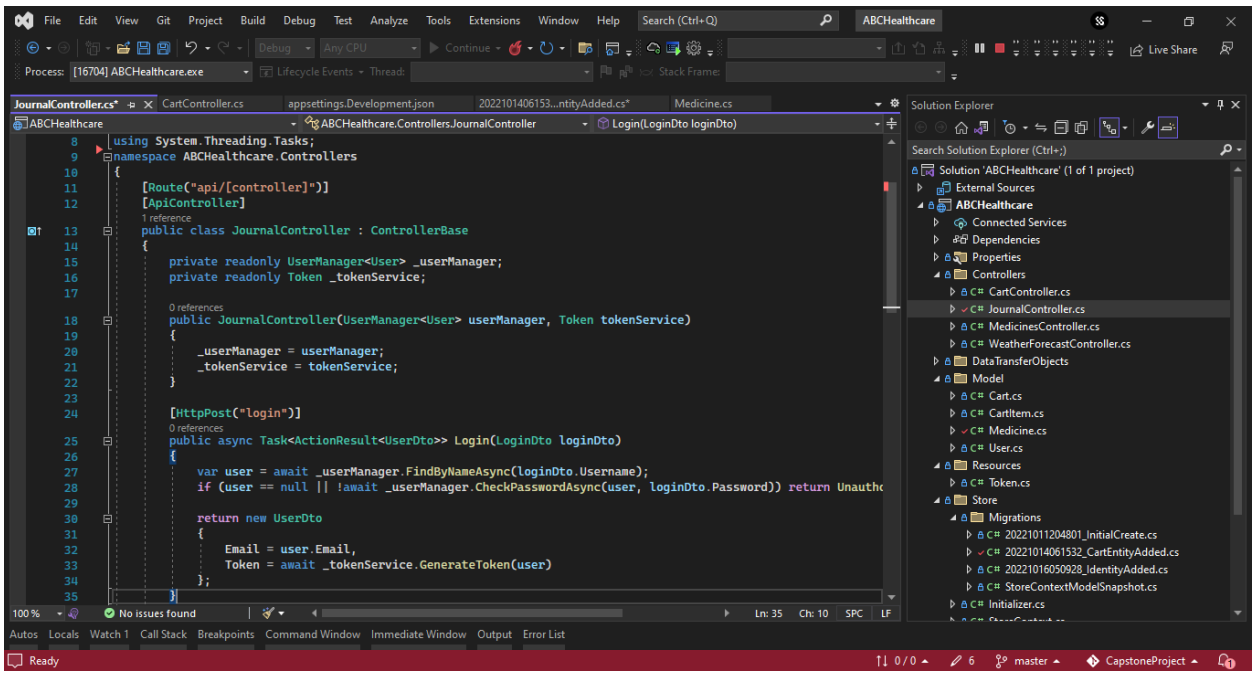
```
        [HttpPost] // api/Cart?medicineId=2&quantity=5
        0 references
        public async Task<ActionResult<CartDto>> AddItemToCart(int medicineId, int quantity)
        {
            // get cart
            var cart = await RetrieveCart();

            // if cart not there, create one
            if (cart == null) cart = CreateCart();

            // get medicine
            var medicine = await _context.Medicines.FindAsync(medicineId);
            if (medicine == null) return NotFound(); // this case should not arrive as we have proper ID for medicines, still adding for safety

            // add item
            cart.AddItem(medicine, quantity);

            // save changes
            var result = await _context.SaveChangesAsync() > 0; // this method returns an int for number of changes has been made to the DB
            if (result) return CreatedAtRoute("GetCart", MapCartToDto(cart));
            return BadRequest(new ProblemDetails { Title = "Problem saving item to cart" });
        }
        [HttpDelete]
        0 references
        public async Task<ActionResult> RemoveCartItem(int medicineId, int quantity)
        {
            // get cart
            var cart = await RetrieveCart();
            if (cart == null) return NotFound();

            // remove item or reduce quantity
            cart.RemoveItem(medicineId, quantity);

            // save changes
            var result = await _context.SaveChangesAsync() > 0;
            if (result) return Ok();
            return BadRequest(new ProblemDetails { Title = "Problem removing item from cart" });
        }
```

Fig. 15: Controllers/CartController.cs (contd…)



```
3 references
private async Task<Cart> RetrieveCart()
{
    return await _context.Carts
        .Include(i => i.Items)
        .ThenInclude(m => m.Medicine)
        .FirstOrDefaultAsync(x => x.BuyerId == Request.Cookies["buyerId"]);
}

1 reference
private Cart CreateCart()
{
    var buyerId = Guid.NewGuid().ToString();
    var cookieOptions = new CookieOptions { IsEssential = true, Expires = DateTime.Now.AddDays(30) };
    Response.Cookies.Append("buyerId", buyerId, cookieOptions);
    var cart = new Cart { BuyerId = buyerId };
    _context.Carts.Add(cart);
    return cart;
}

2 references
private CartDto MapCartToDto(Cart cart)
{
    return new CartDto
    {
        Id = cart.Id,
        BuyerId = cart.BuyerId,
        Items = cart.Items.Select(item => new CartItemDto
        {
            MedicineId = item.MedicineId,
            Name = item.Medicine.Name,
            Price = item.Medicine.Price,
            Image = item.Medicine.Image,
            Seller = item.Medicine.Seller,
            Description = item.Medicine.Description,
            Quantity = item.Quantity,
            Category = item.Medicine.Category
        }).ToList()
    };
```

Fig. 16: Controllers/CartController.cs (contd…)



```
using System.Threading.Tasks;
namespace ABCHealthcare.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    1 reference
    public class JournalController : ControllerBase
    {
        private readonly UserManager<User> _userManager;
        private readonly Token _tokenService;

        0 references
        public JournalController(UserManager<User> userManager, Token tokenService)
        {
            _userManager = userManager;
            _tokenService = tokenService;
        }

        [HttpPost("login")]
        0 references
        public async Task<ActionResult<UserDto>> Login(LoginDto loginDto)
        {
            var user = await _userManager.FindByNameAsync(loginDto.Username);
            if (user == null || !await _userManager.CheckPasswordAsync(user, loginDto.Password)) return Unauth

            return new UserDto
            {
                Email = user.Email,
                Token = await _tokenService.GenerateToken(user)
            };
        }
```

Fig. 17: Controllers/JournalController.cs

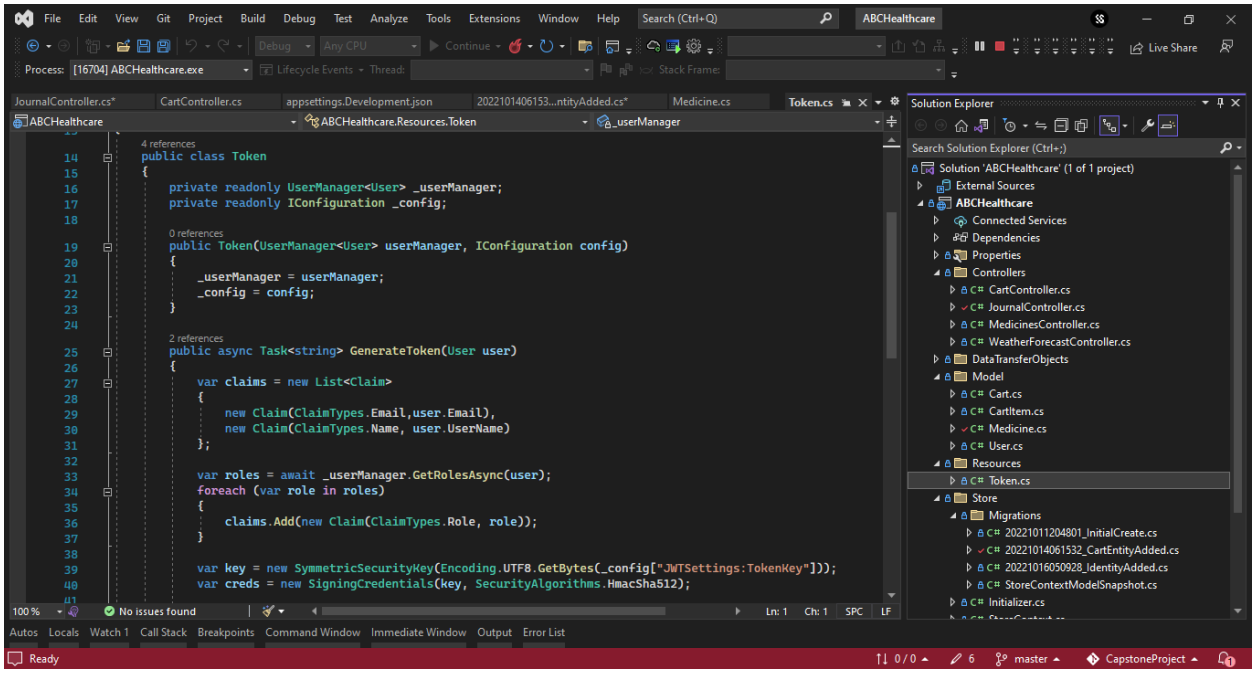


Fig. 18: Controllers/JournalController.cs

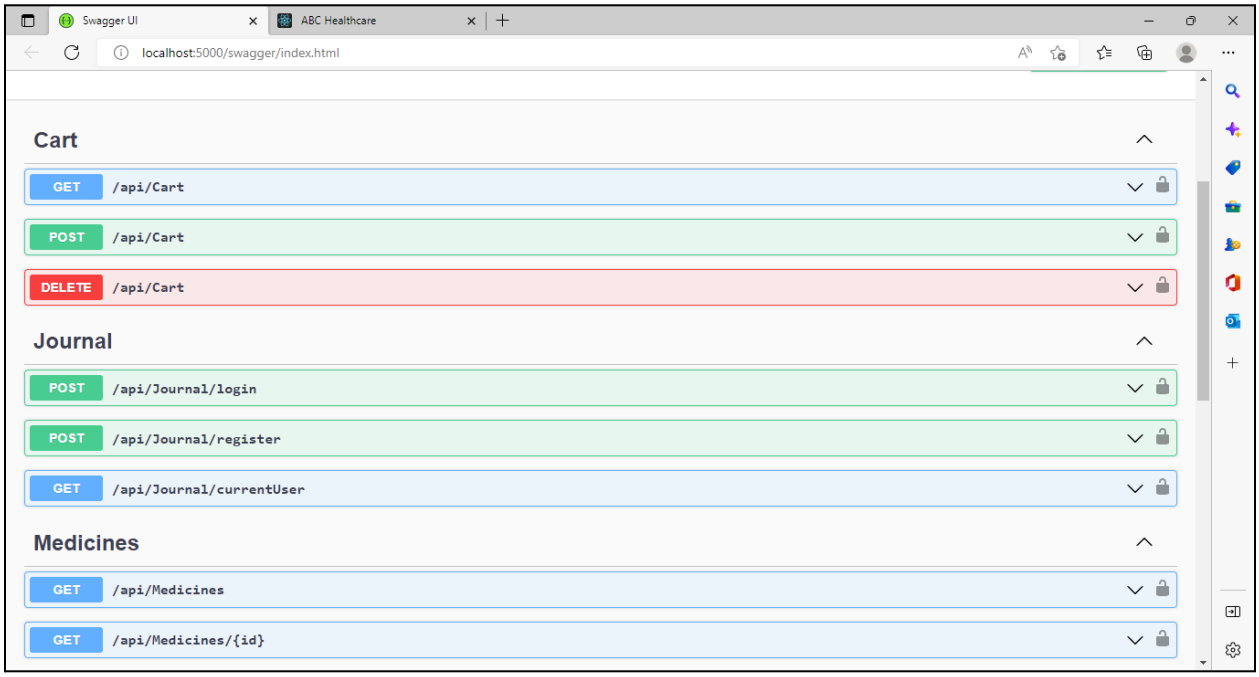


Fig. 19: Swagger/index.html

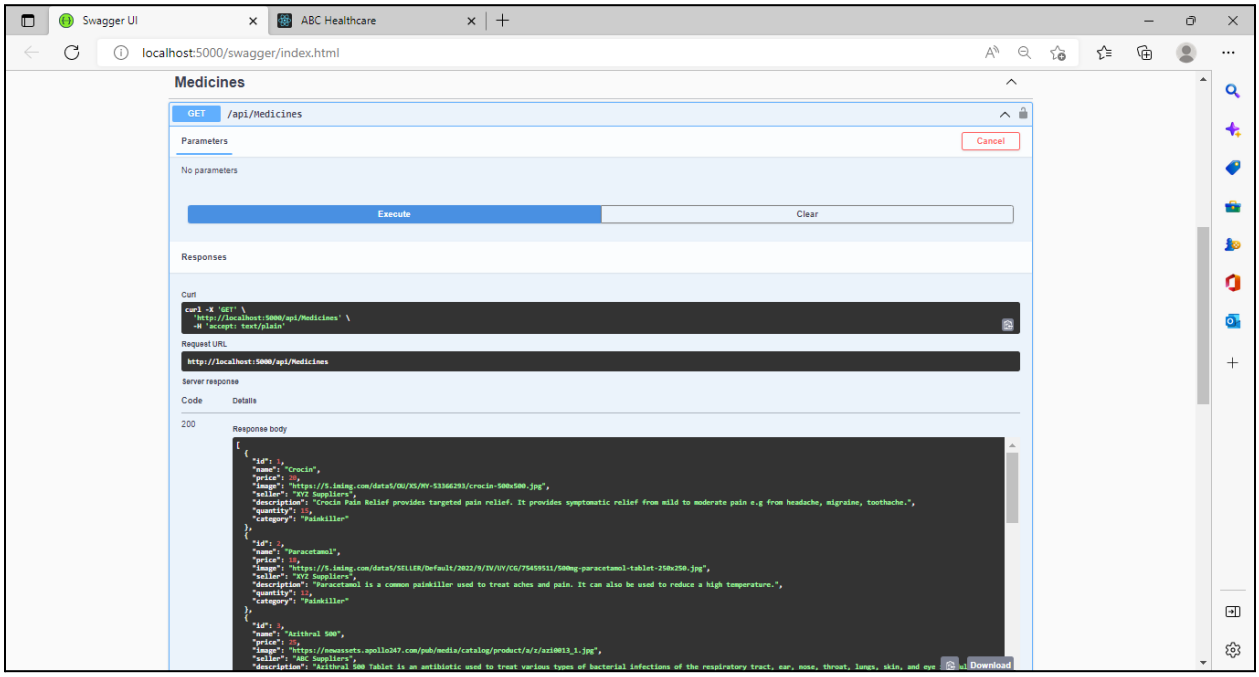


Fig. 20: Medicines GET method

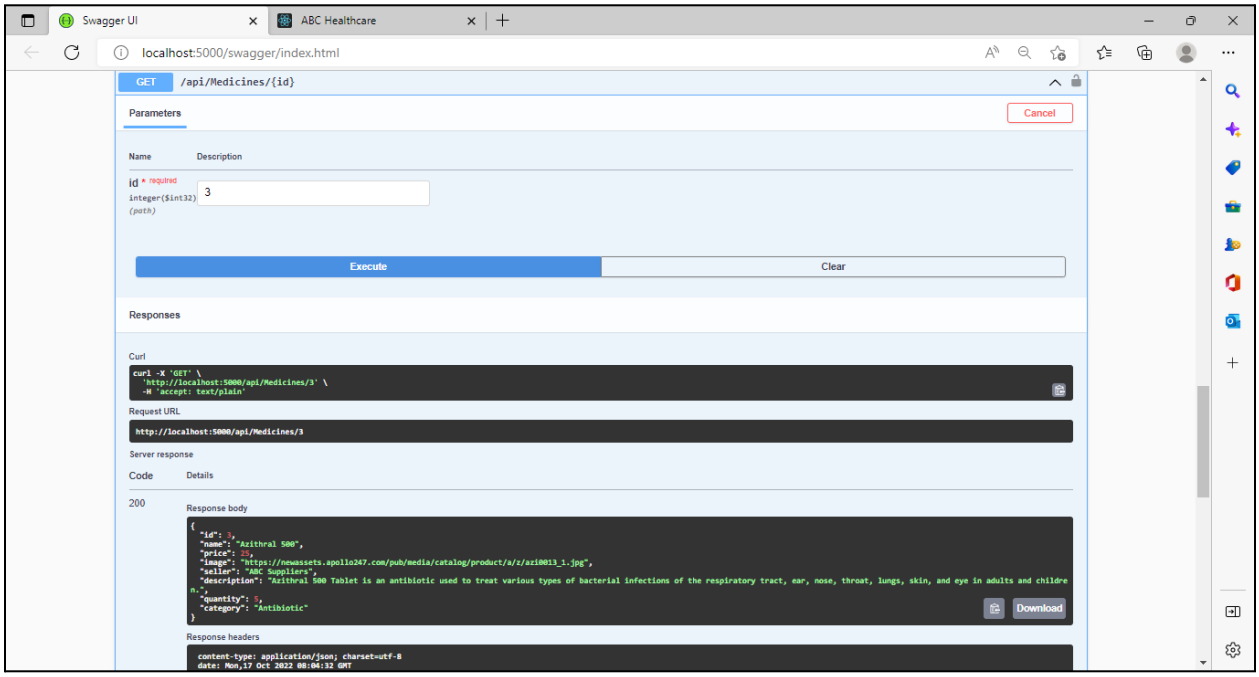


Fig. 21: Medicines GET method for a particular ID

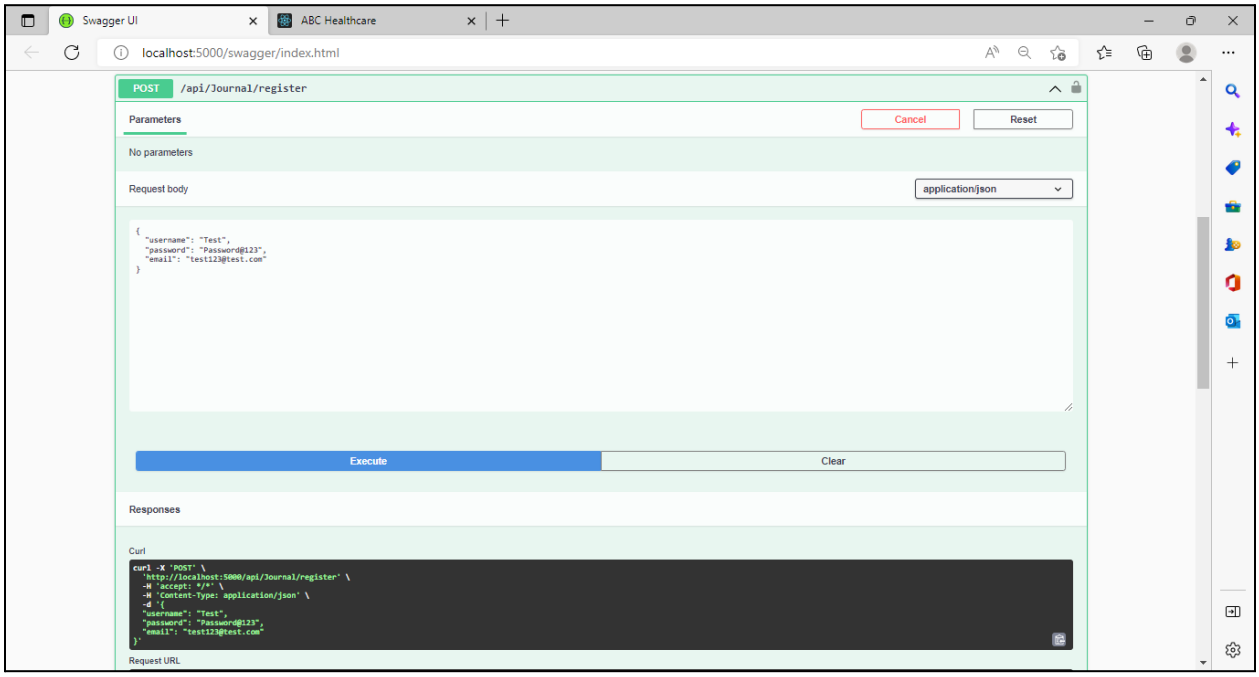
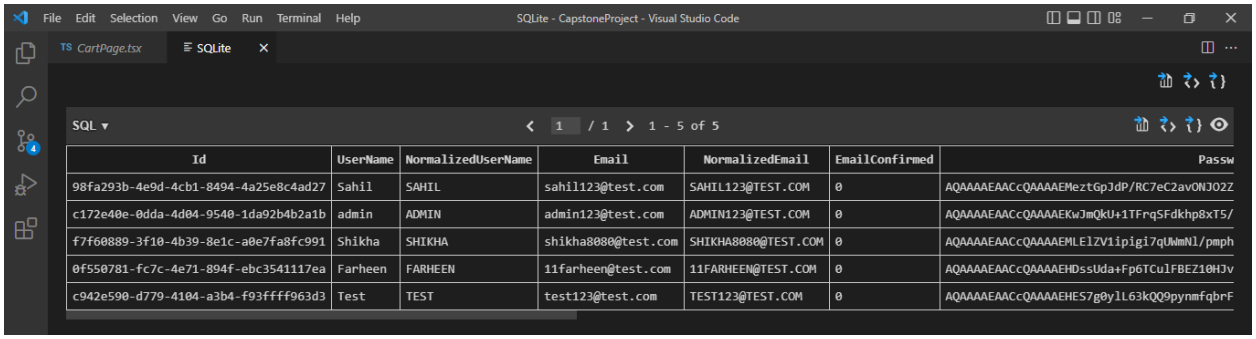


Fig. 22: Register POST method



Id	UserName	NormalizedUserName	Email	NormalizedEmail	EmailConfirmed	Passw
98fa293b-4e9d-4cb1-8494-4a25e8c4ad27	Sahil	SAHIL	sahil123@test.com	SAHIL123@TEST.COM	0	AQAAAAEAACcQAAAAEmztGpJdP/RC7eC2avONJO2Z
c172e40e-0dda-4d04-9540-1da92b4b2a1b	admin	ADMIN	admin123@test.com	ADMIN123@TEST.COM	0	AQAAAAEAACcQAAAAEKwJmQkU+1TFrqSFdkhp8xT5/
f7f60889-3f10-4b39-8e1c-a0e7fa8fc991	Shikha	SHIKHA	shikha8080@test.com	SHIKHA8080@TEST.COM	0	AQAAAAEAACcQAAAAEMLElZV1ipigi7qUWmNl/pmph
0f550781-fc7c-4e71-894f-ebc3541117ea	Farheen	FARHEEN	11farheen@test.com	11FARHEEN@TEST.COM	0	AQAAAAEAACcQAAAAEHDssUda+Fp6TCulfBEZ10HJv
c942e590-d779-4104-a3b4-f93ffff963d3	Test	TEST	test123@test.com	TEST123@TEST.COM	0	AQAAAAEAACcQAAAAEHES7g0yll63kQQ9pynmfqbrF

Fig. 23: Users table

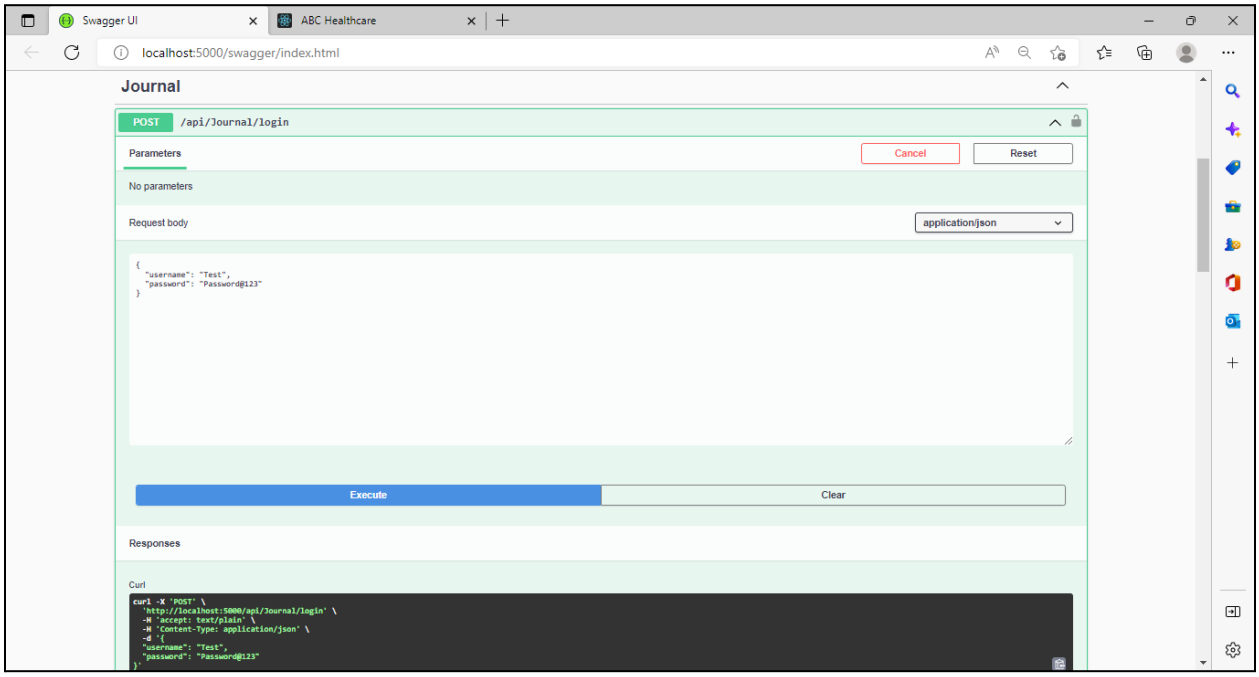


Fig. 24: Login POST method

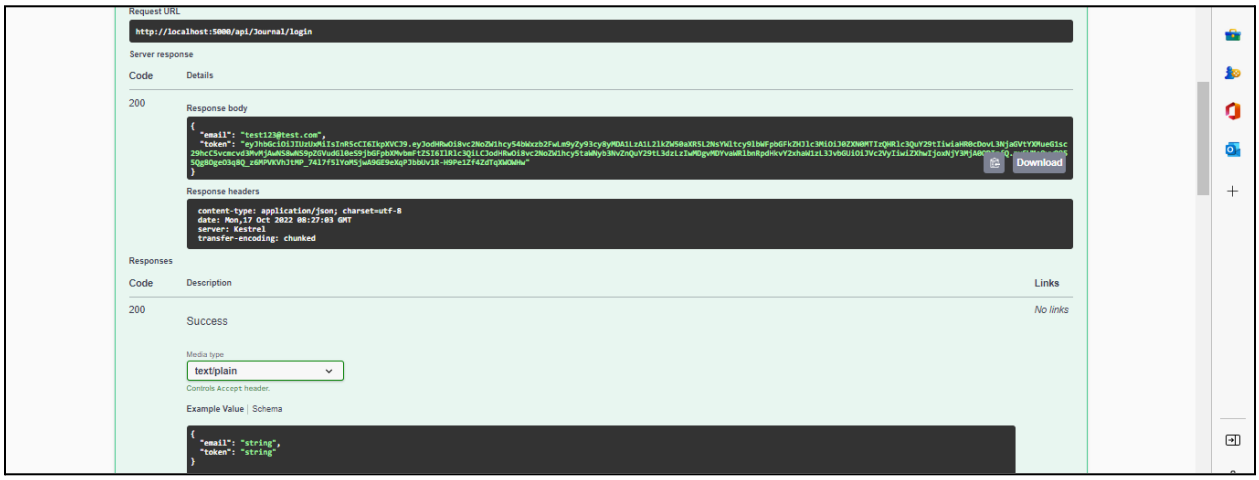


Fig. 25: Login POST method (contd…)

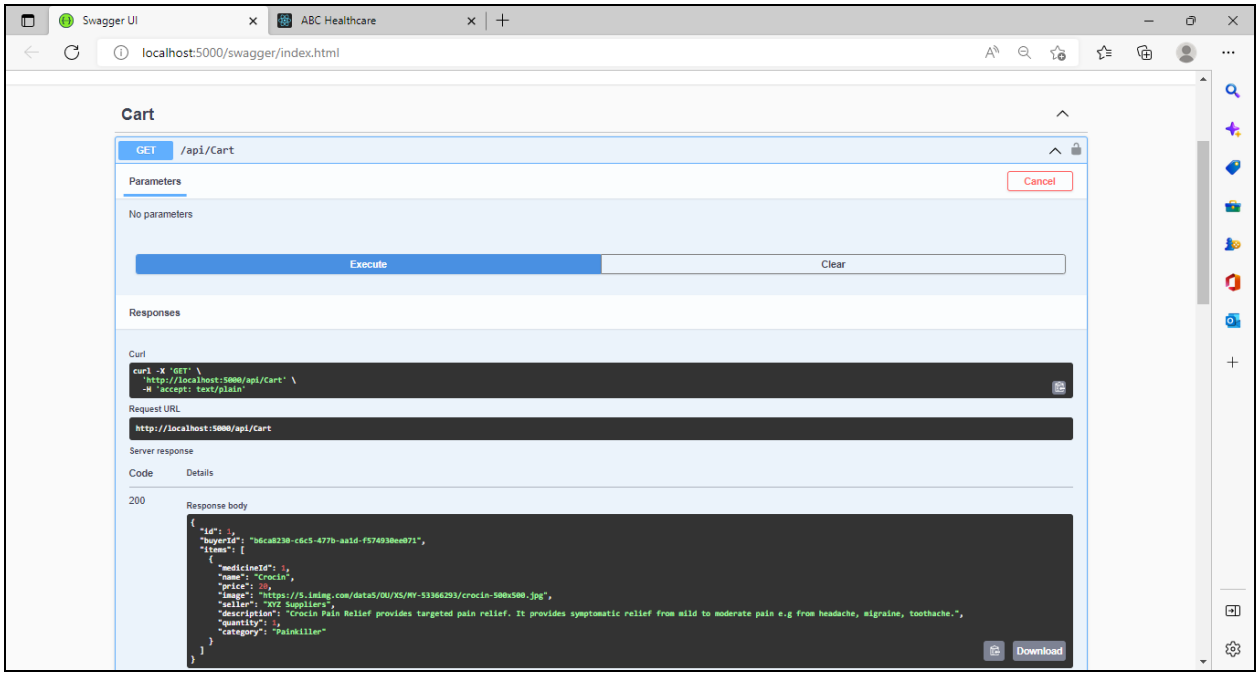


Fig. 26: Cart GET method

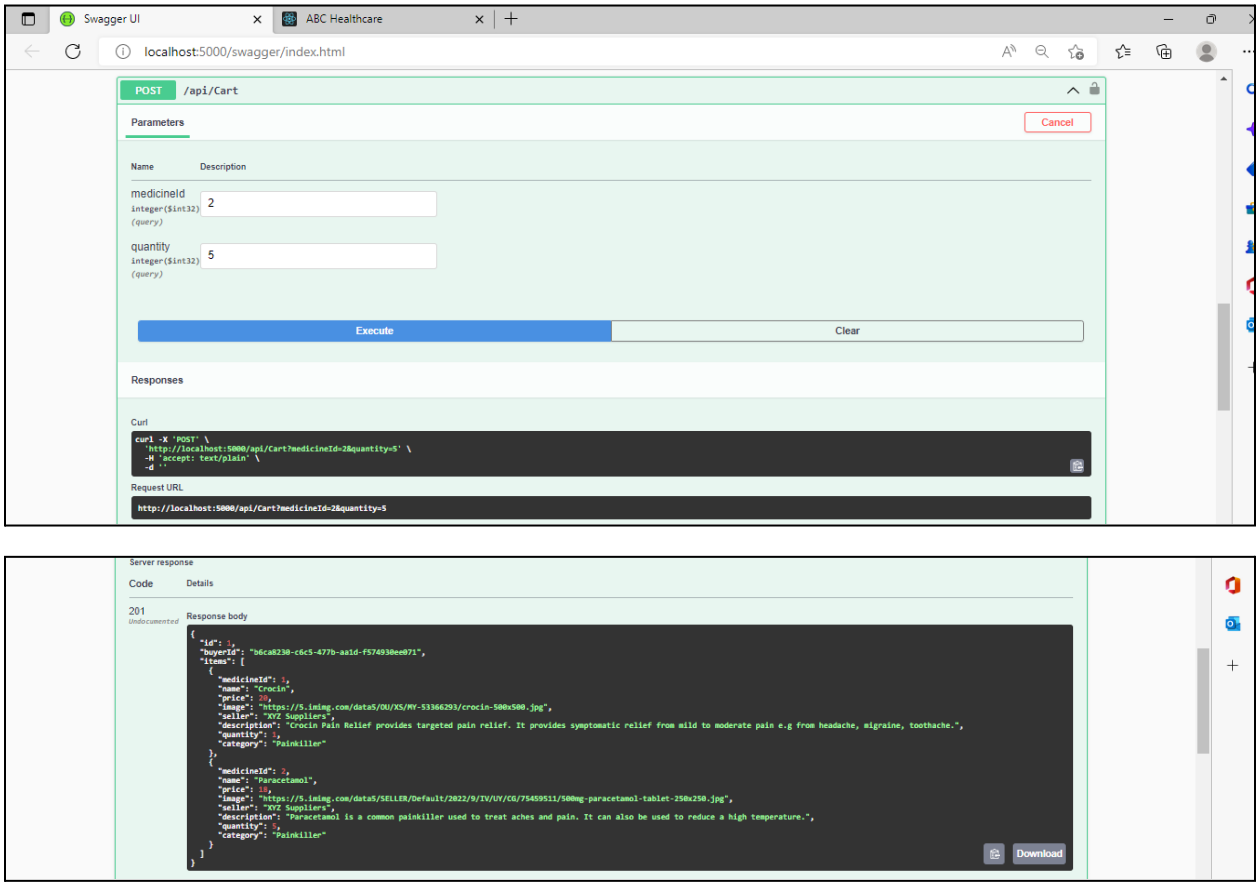


Fig. 27: Cart POST method

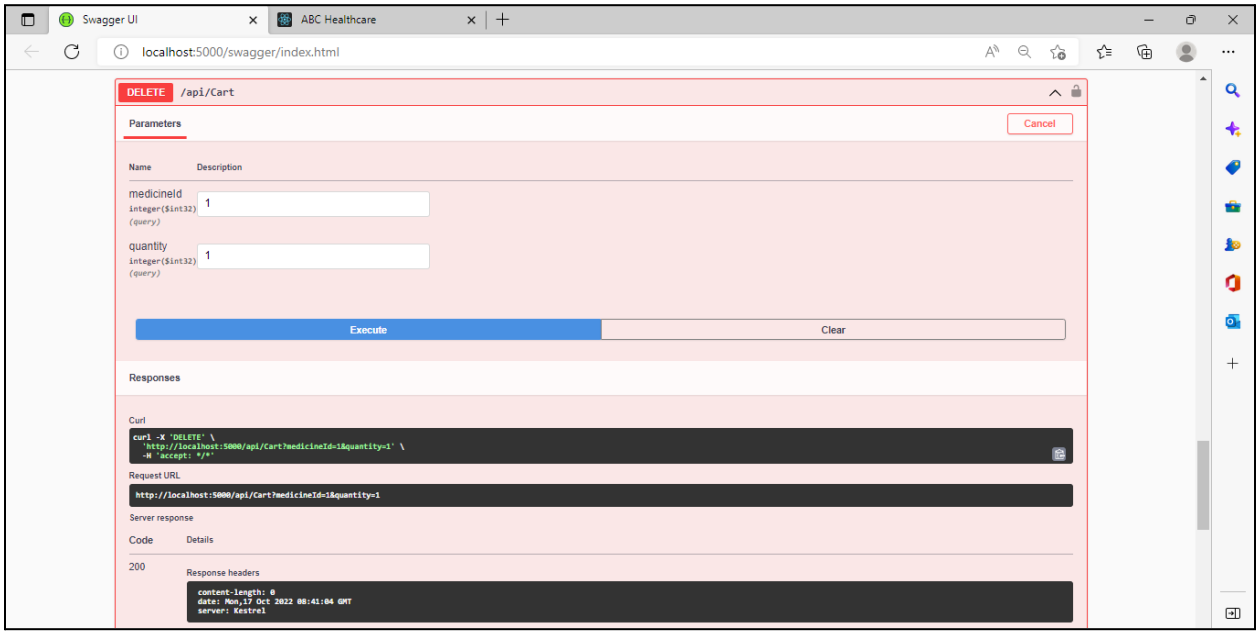


Fig. 28: Cart DELETE method

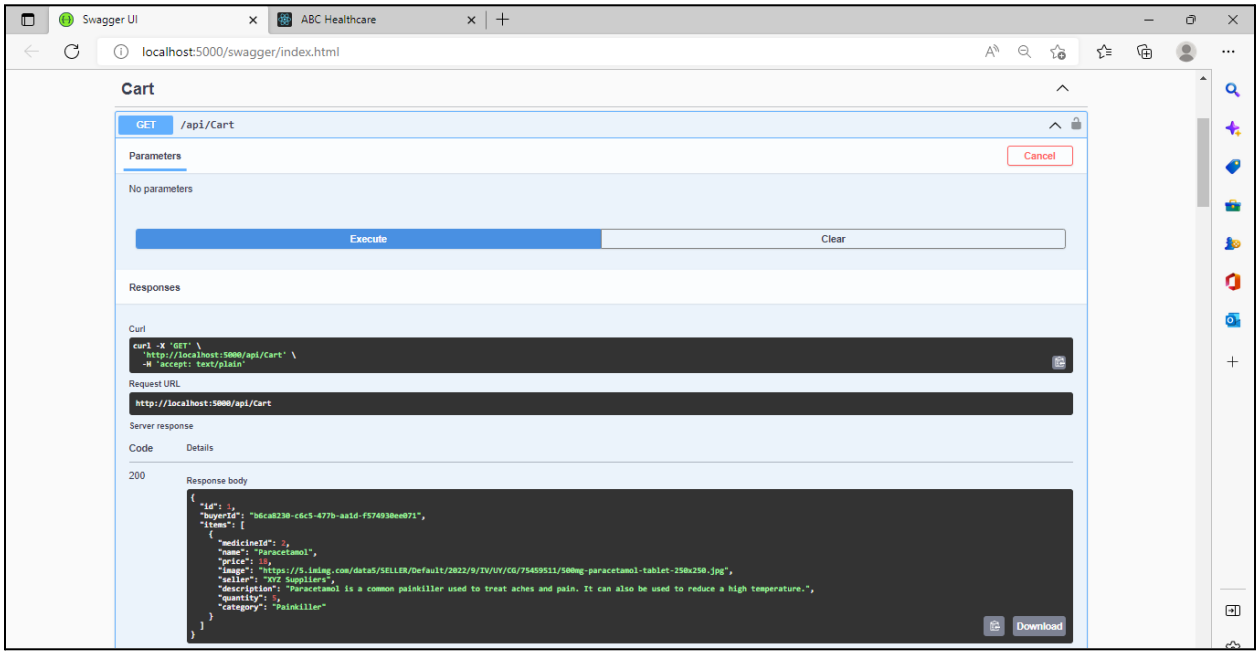


Fig. 29: Cart GET method

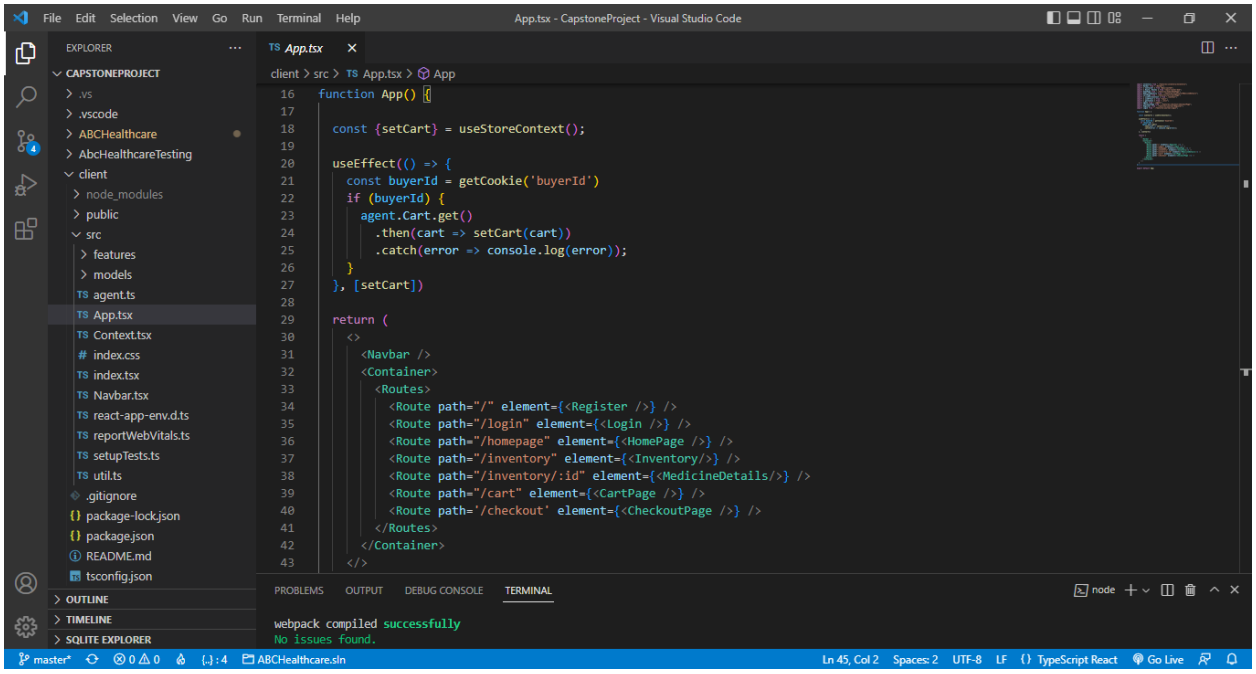


Fig. 30: App component

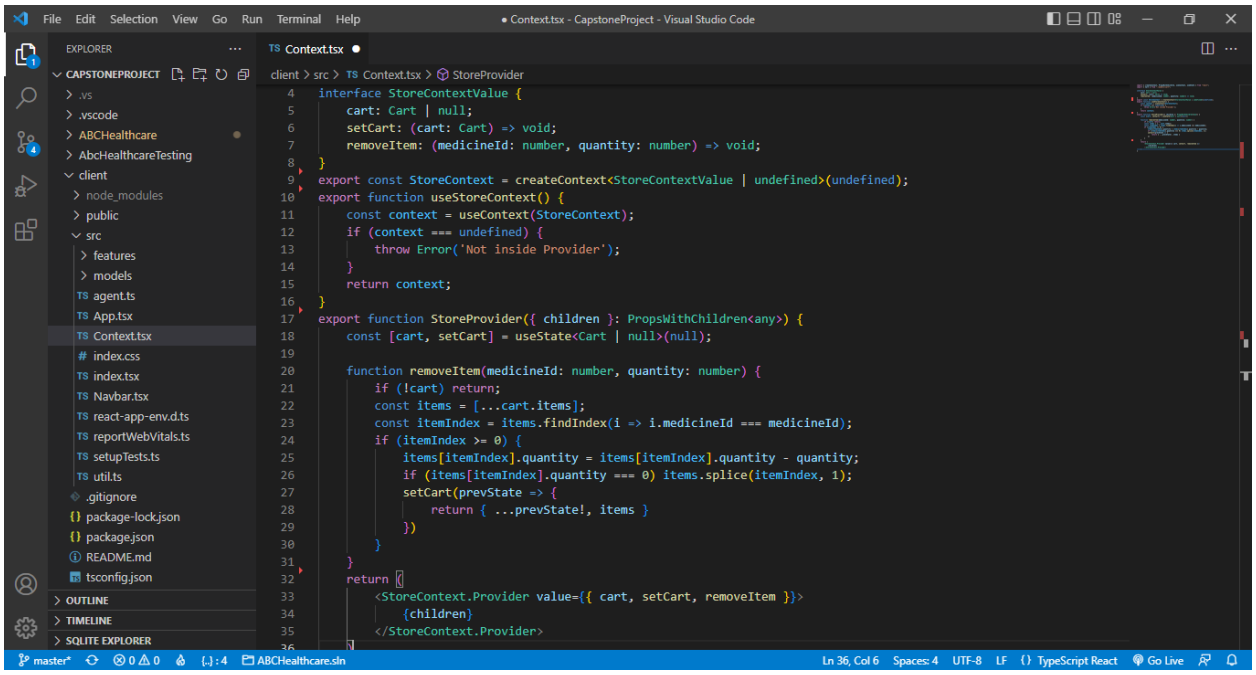


Fig. 31: Context.tsx

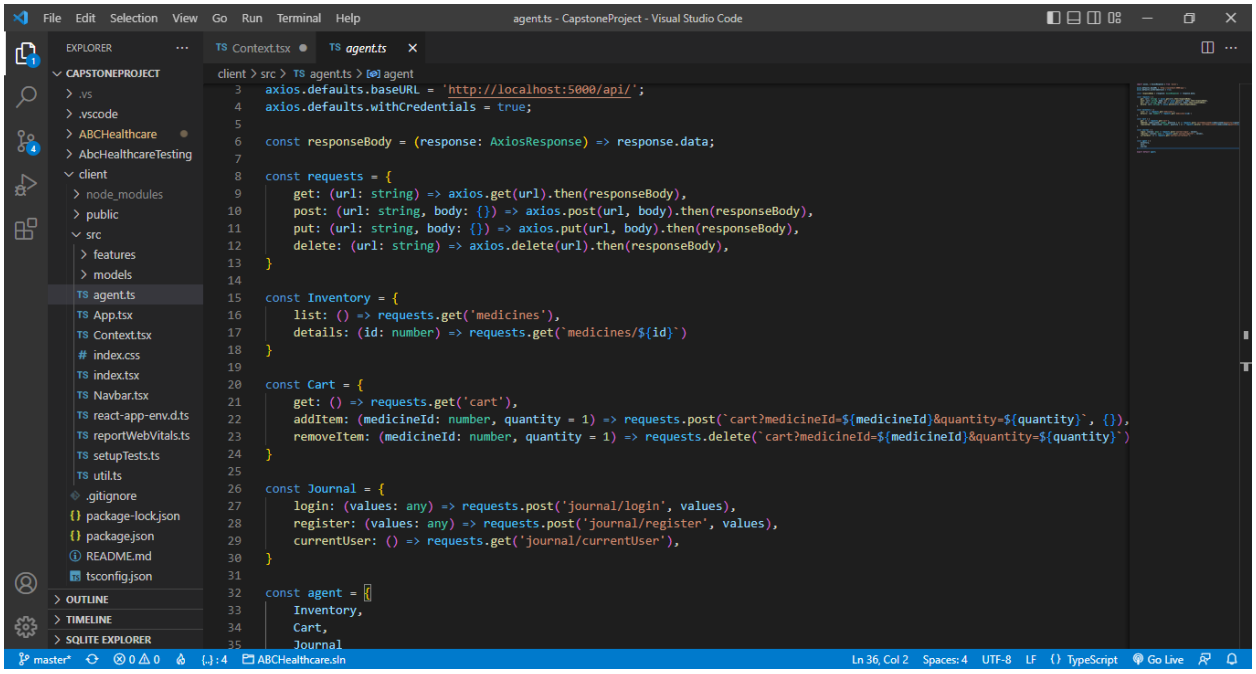


Fig. 32: agent.ts

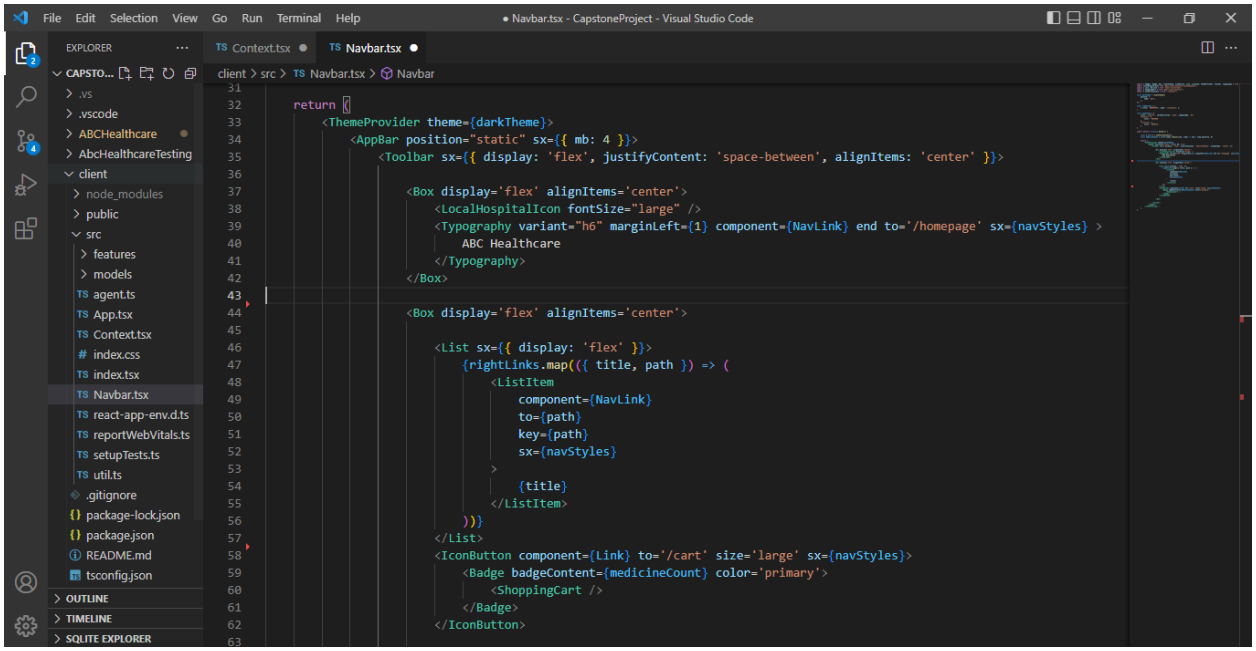


Fig. 33: Navbar component

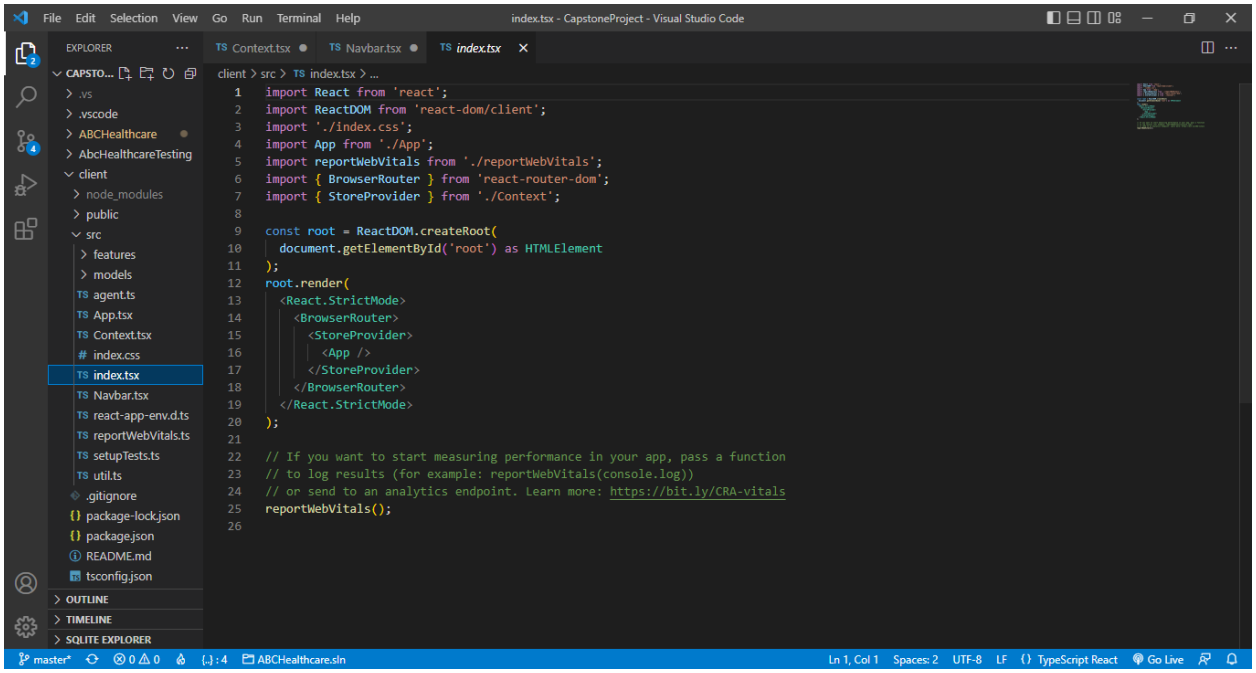


Fig. 34: index.tsx

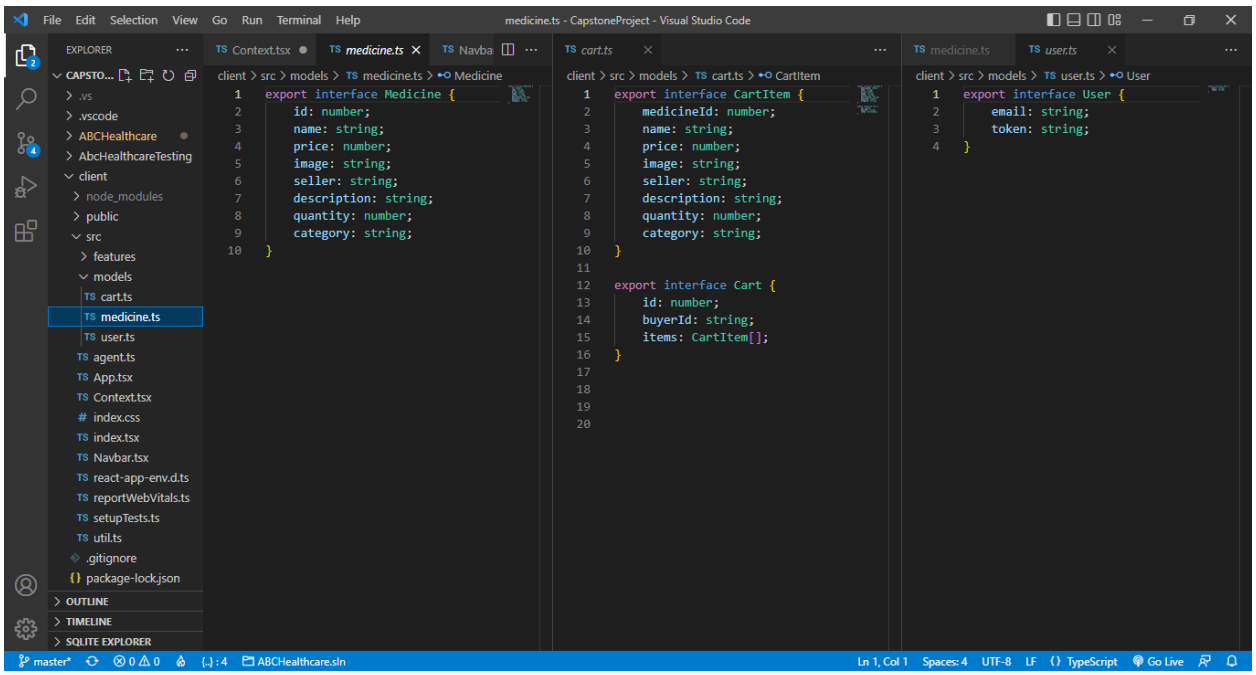


Fig. 35: medicine.ts, cart.ts, user.ts

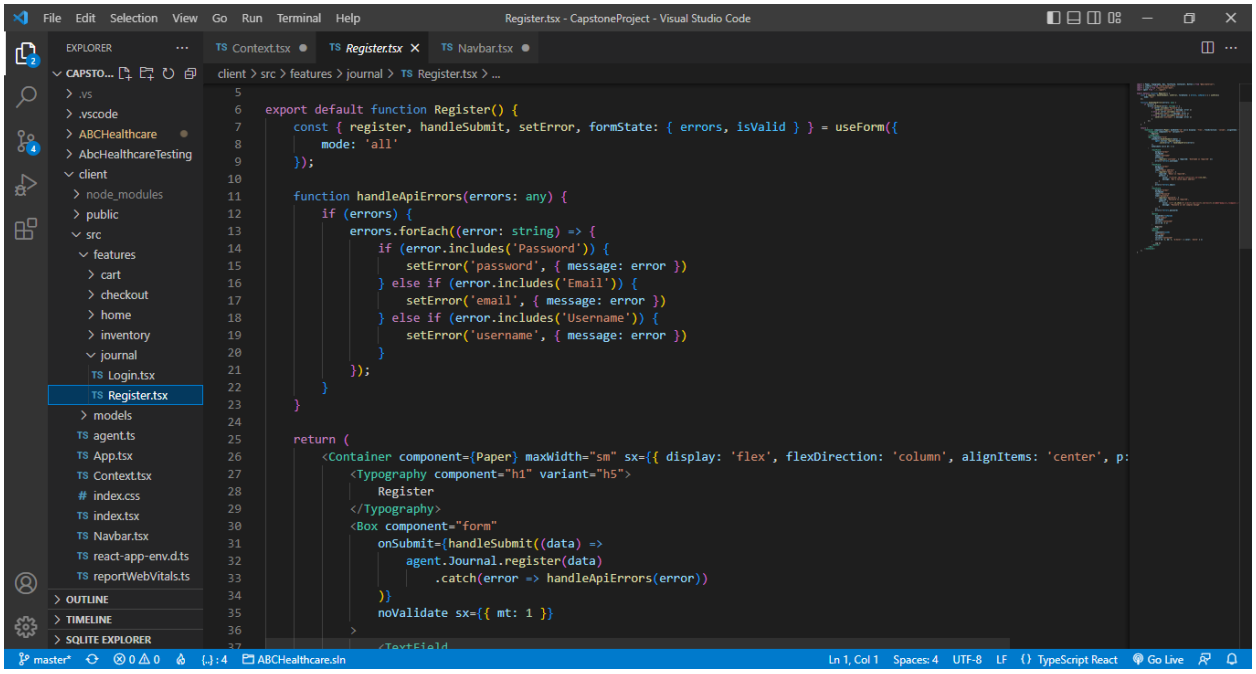


Fig. 36: Register.tsx

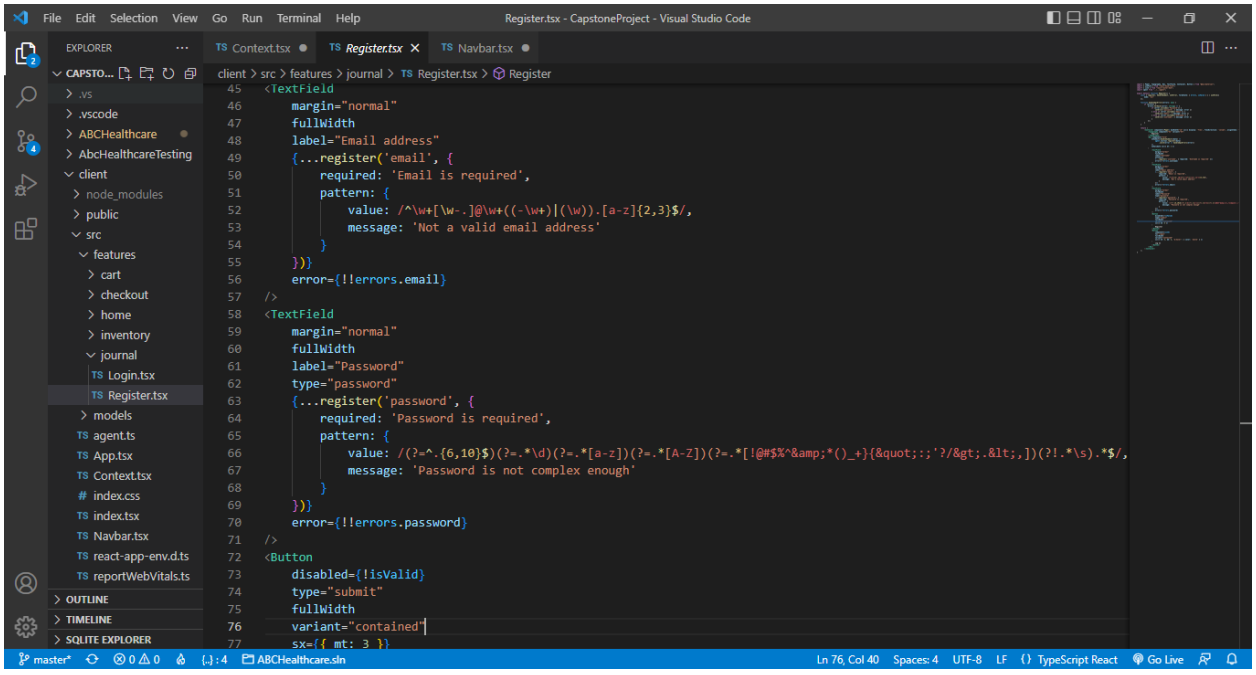


Fig. 37: Register.tsx (contd…)

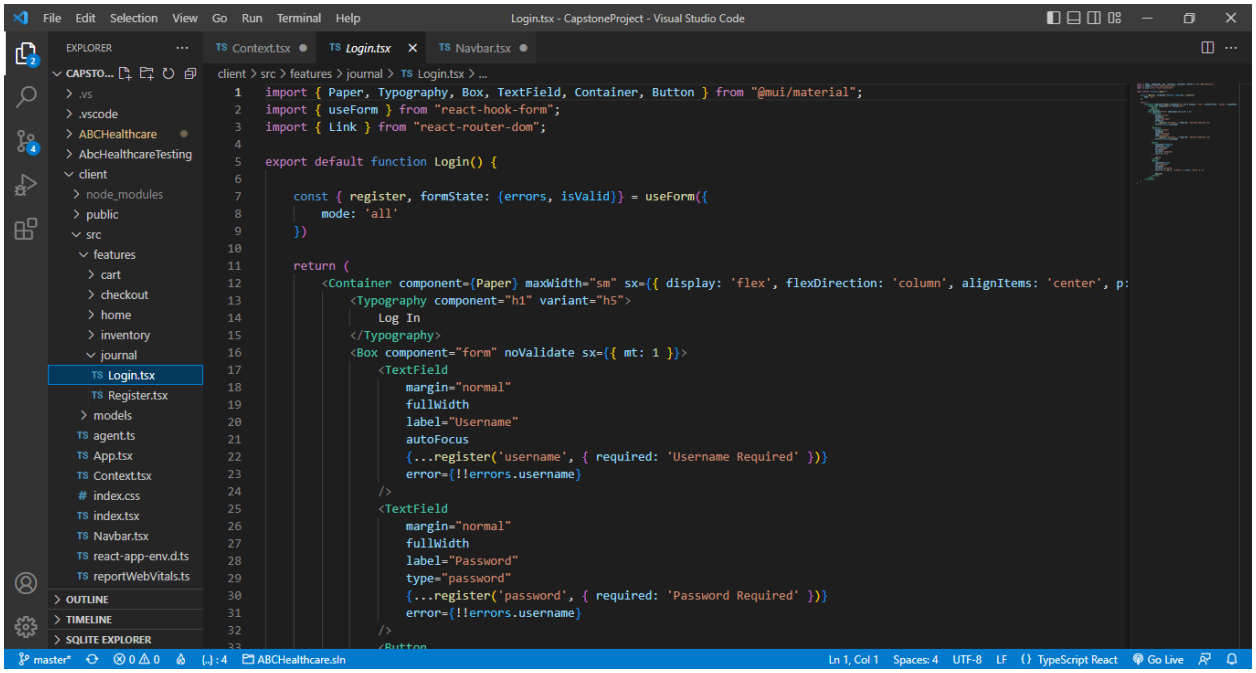


Fig. 38: Login.tsx

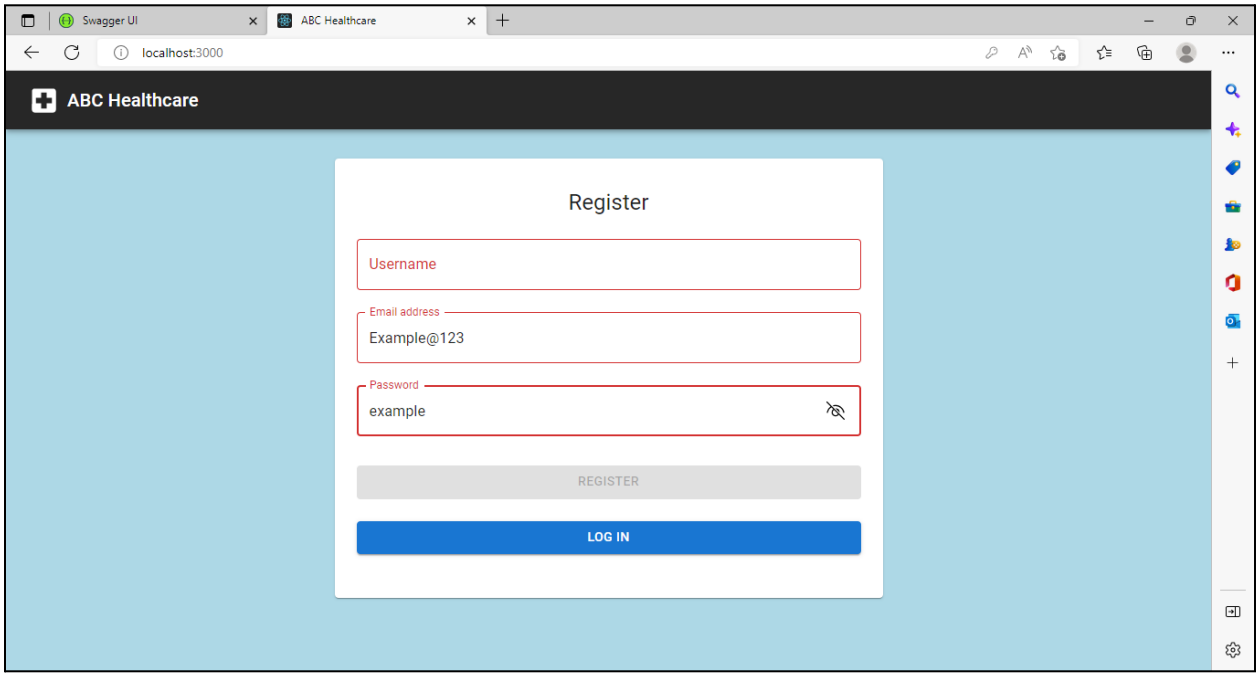


Fig. 39: Register Page

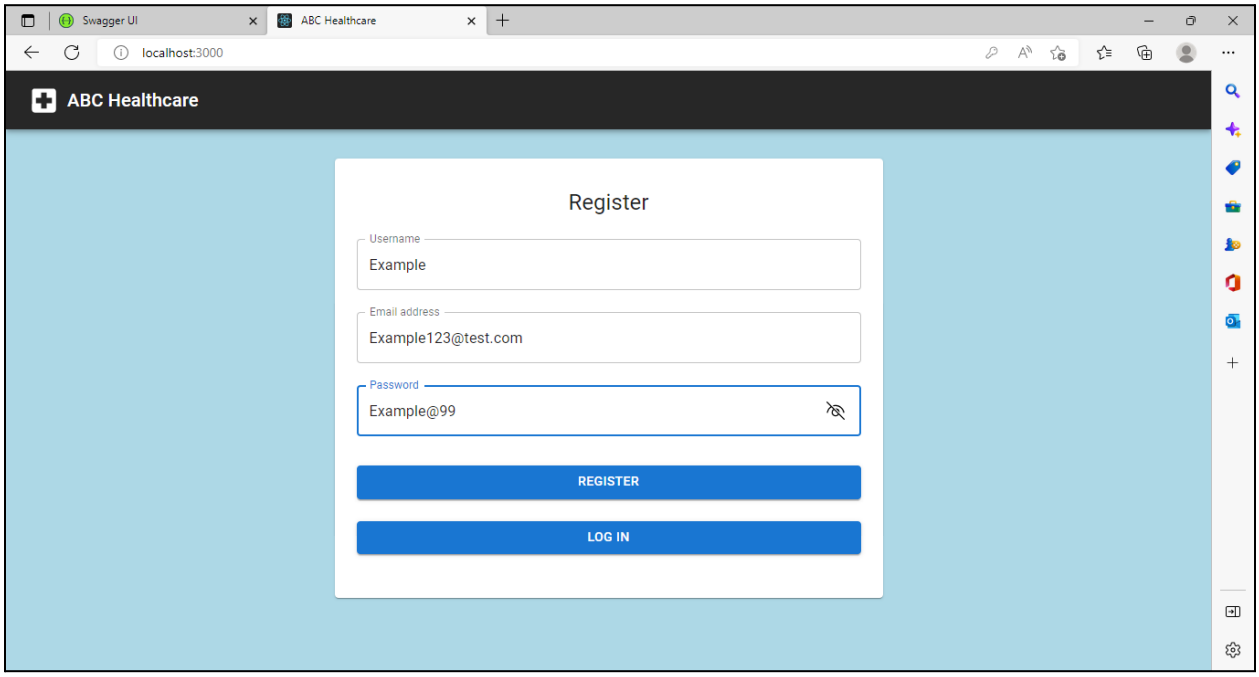
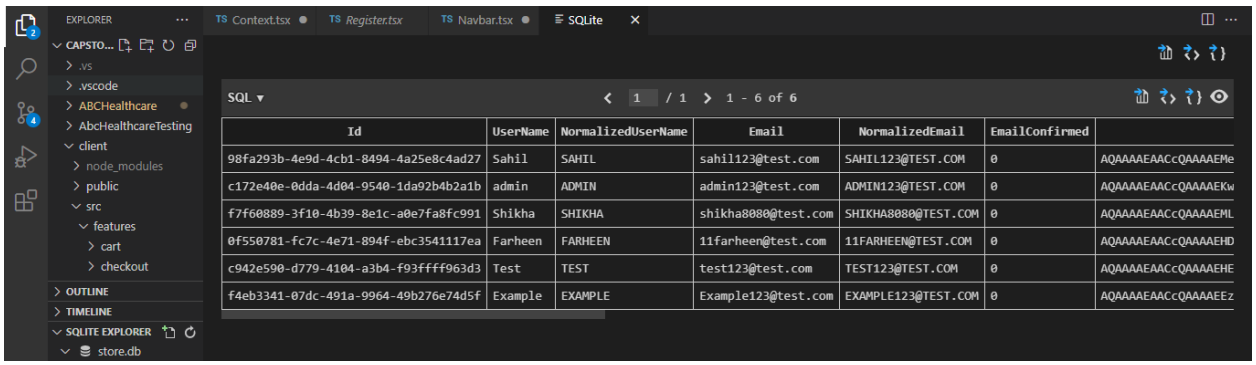


Fig. 40: Register Page (contd…)



Id	UserName	NormalizedUserName	Email	NormalizedEmail	EmailConfirmed	
98fa293b-4e9d-4cb1-8494-4a25e8c4ad27	Sahil	SAHIL	sahil123@test.com	SAHIL123@TEST.COM	0	AQAAAAEAACcQAAAAEMe
c172e40e-0dda-4d04-9540-1da92b4b2a1b	admin	ADMIN	admin123@test.com	ADMIN123@TEST.COM	0	AQAAAAEAACcQAAAAEKw
f7f60889-3f10-4b39-8e1c-a0e7fa8fc991	Shikha	SHIKHA	shikha8080@test.com	SHIKHA8080@TEST.COM	0	AQAAAAEAACcQAAAAEML
0f550781-fc7c-4e71-894f-ebc3541117ea	Farheen	FARHEEN	11farheen@test.com	11FARHEEN@TEST.COM	0	AQAAAAEAACcQAAAAEHD
c942e590-d779-4104-a3b4-f93ffff963d3	Test	TEST	test123@test.com	TEST123@TEST.COM	0	AQAAAAEAACcQAAAAEHE
f4eb3341-07dc-491a-9964-49b276e74d5f	Example	EXAMPLE	Example123@test.com	EXAMPLE123@TEST.COM	0	AQAAAAEAACcQAAAAEEz

Fig. 41: Users table

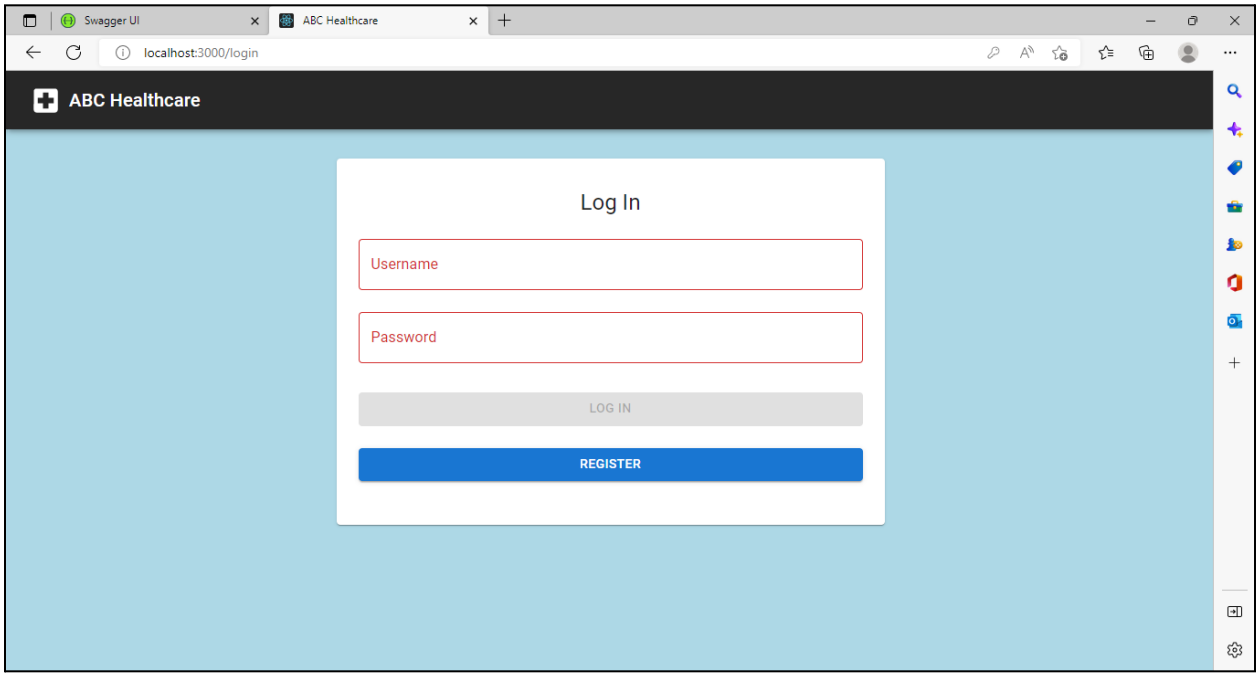


Fig. 42: Login Page

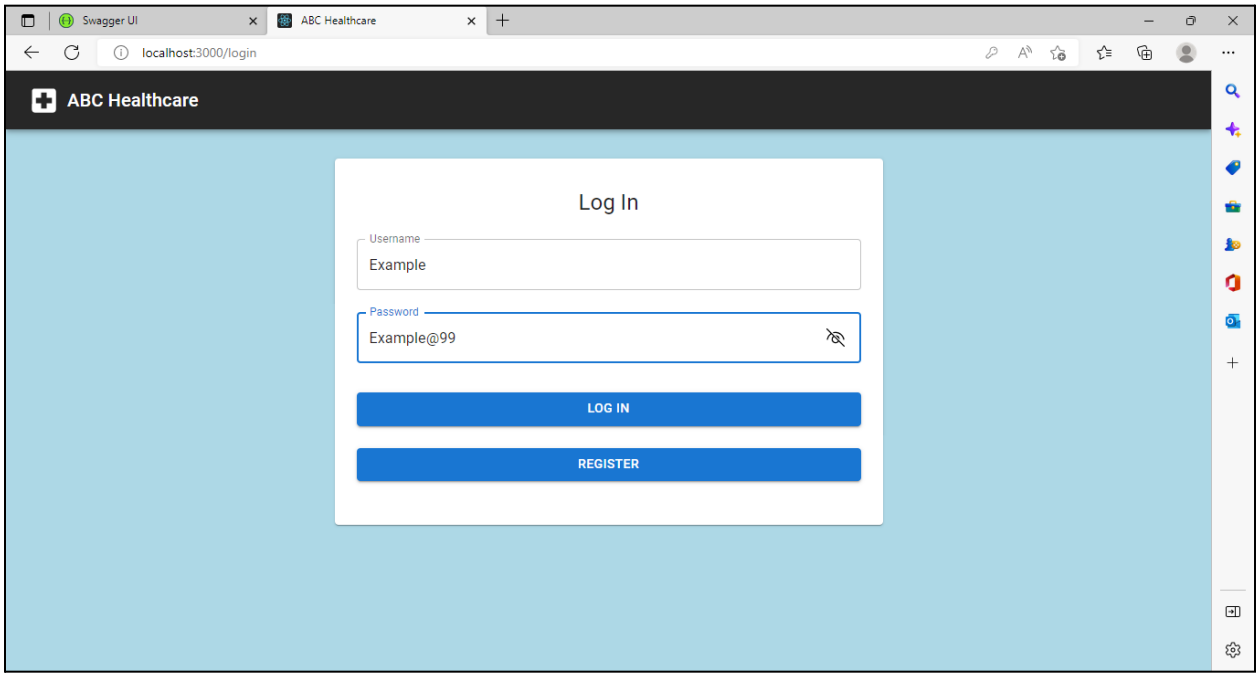


Fig. 43: Login Page (contd…)

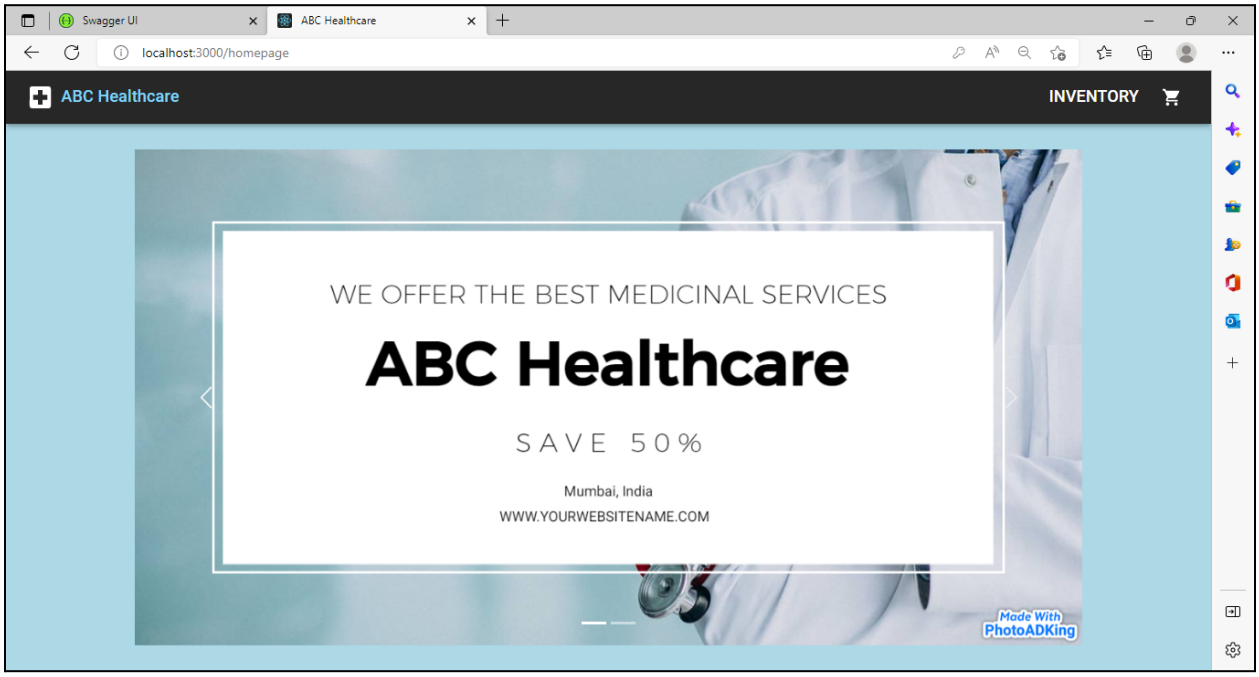


Fig. 44: HomePage

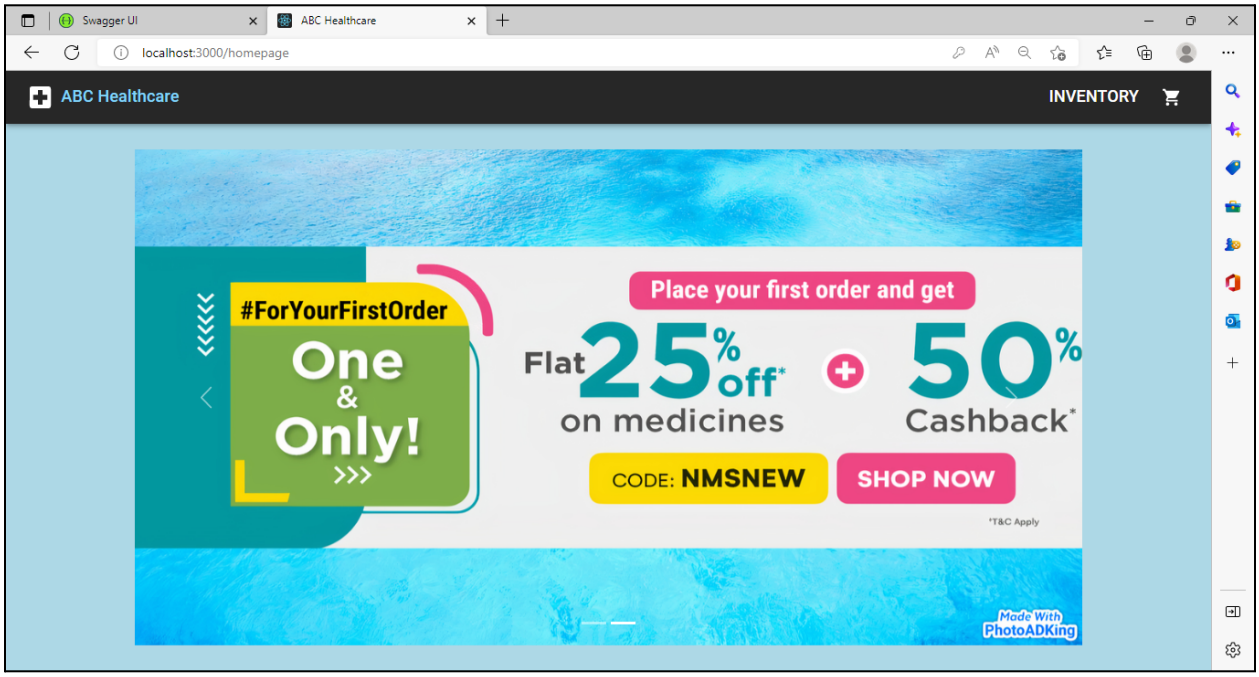


Fig. 45: HomePage (contd…)

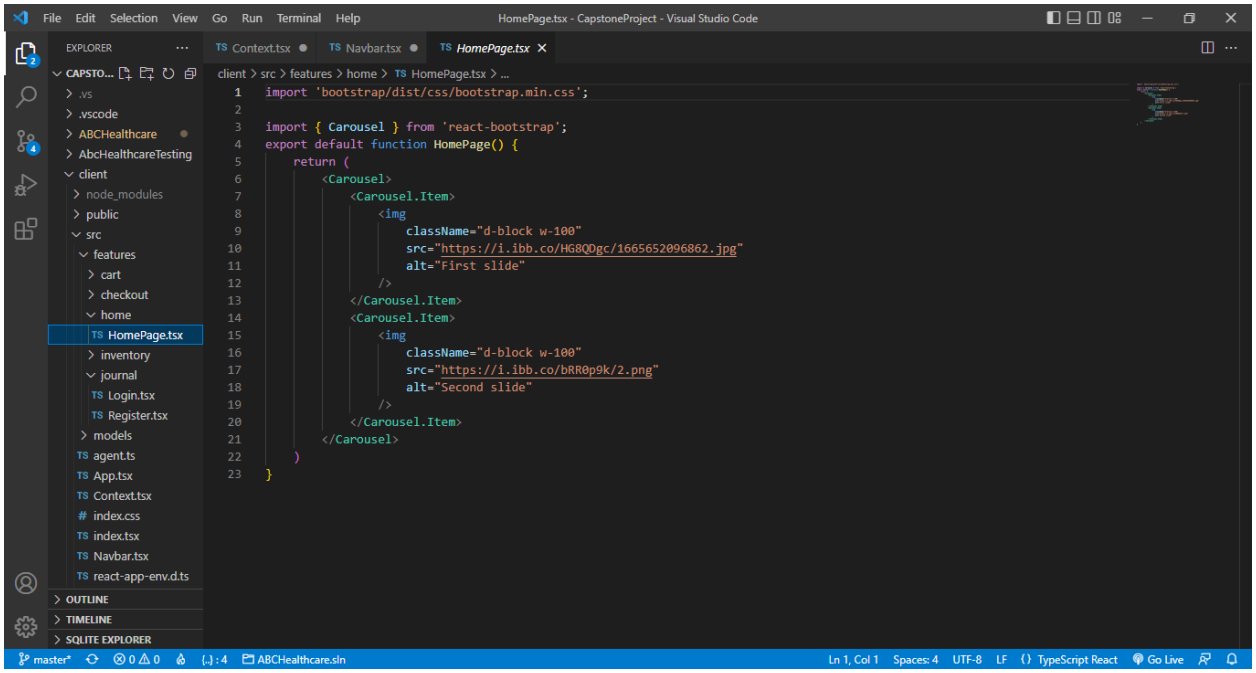


Fig. 46: HomePage.tsx

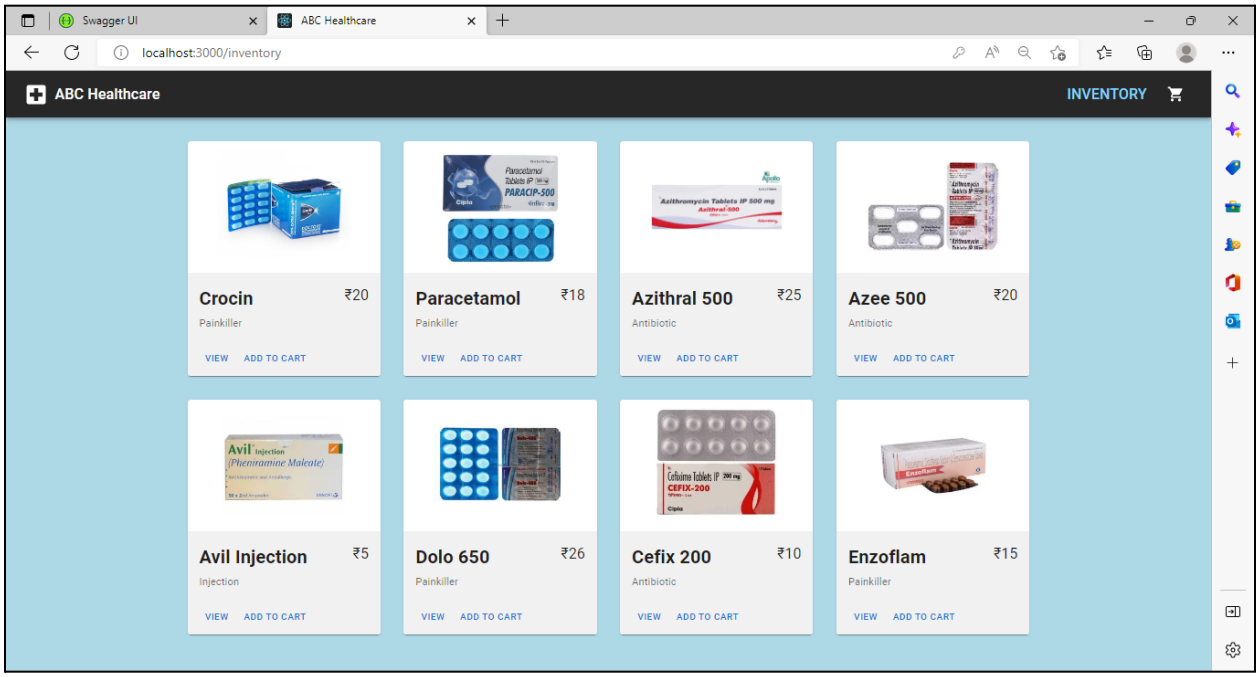


Fig. 47: Inventory Page

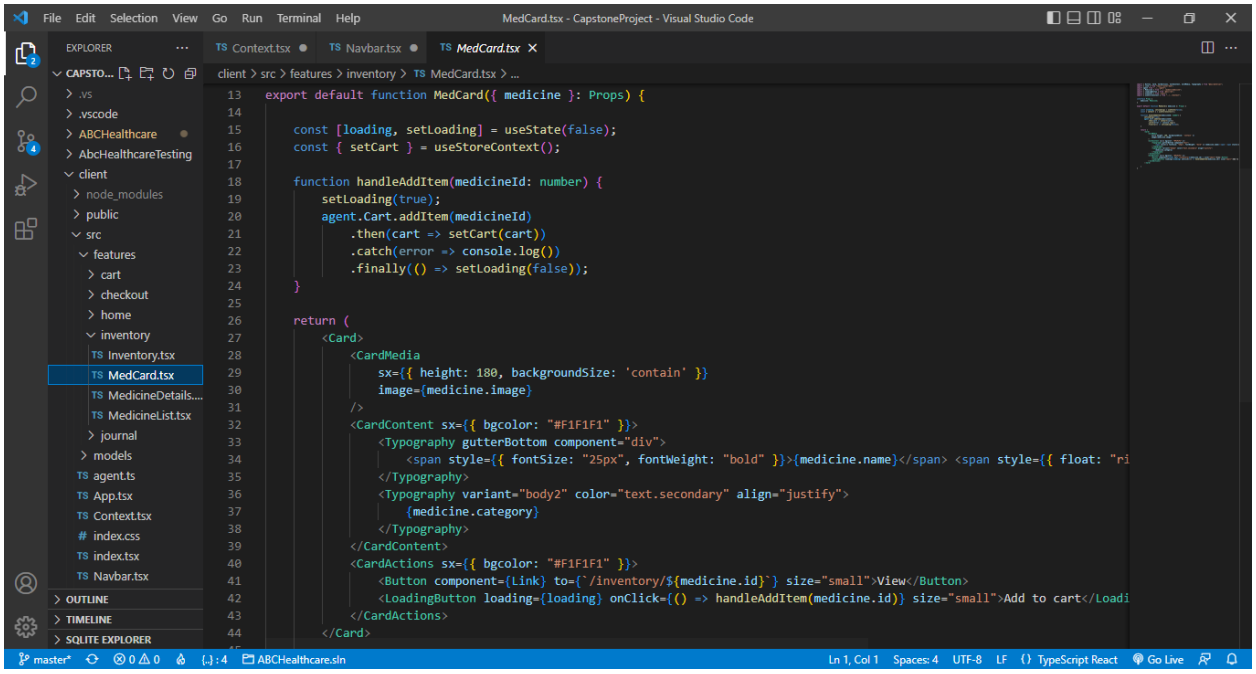


Fig. 48: inventory/MedCard.tsx

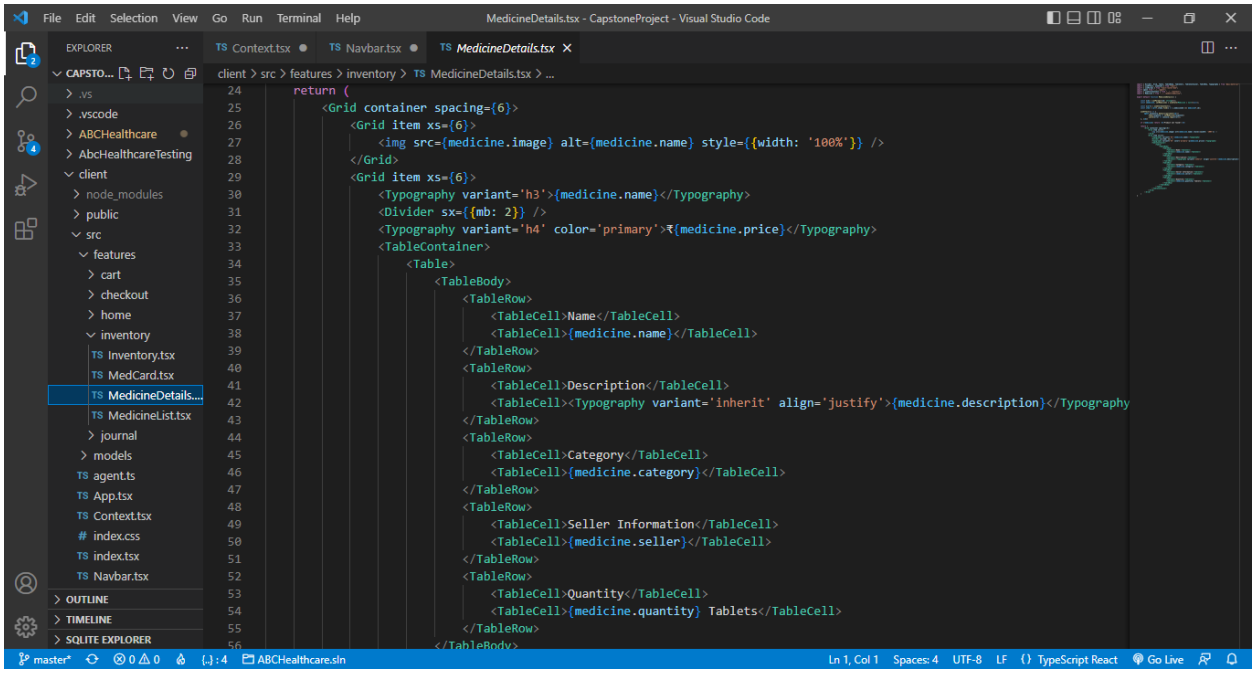


Fig. 49: inventory/MedicineDetails.tsx

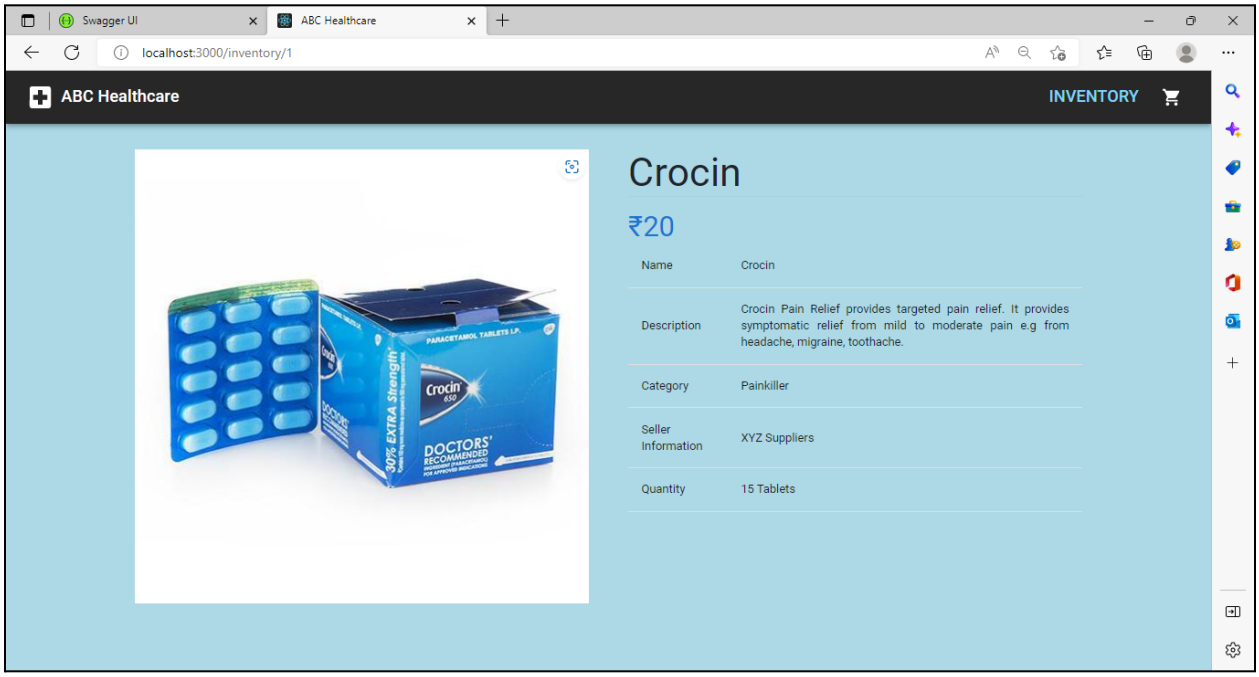


Fig. 50: Medicine-Details Page

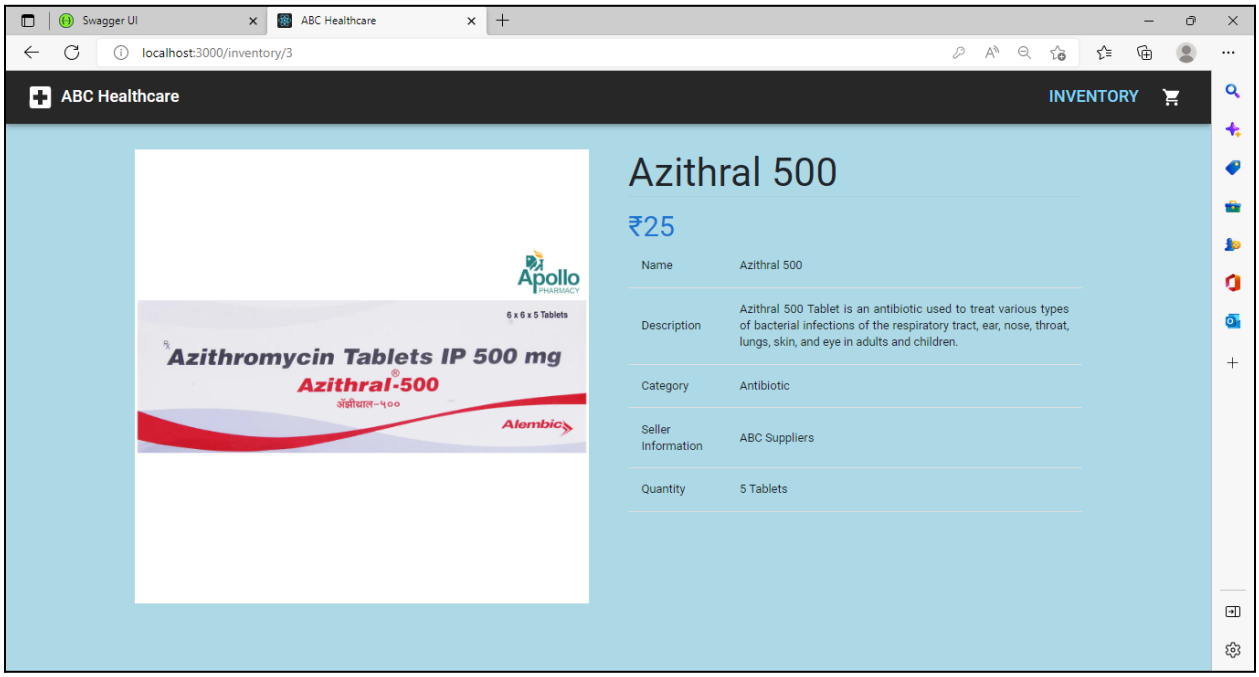


Fig. 51: Medicine-Details Page

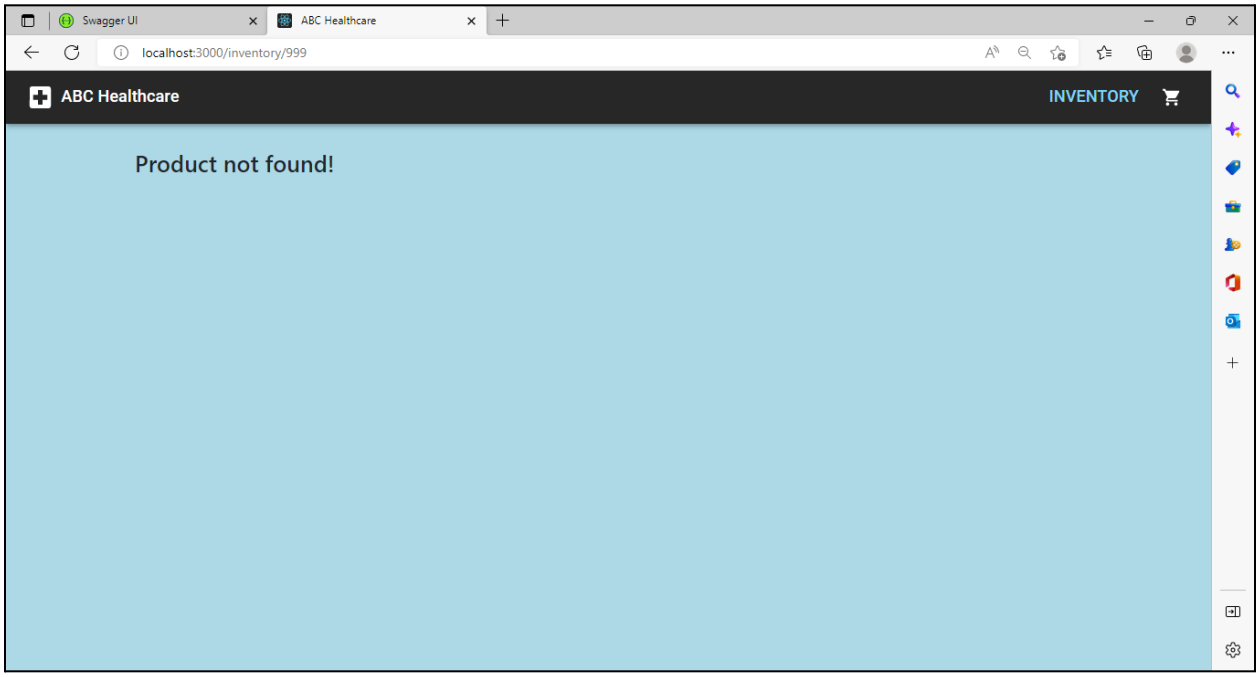


Fig. 52: Wrong Medicine-Details Page

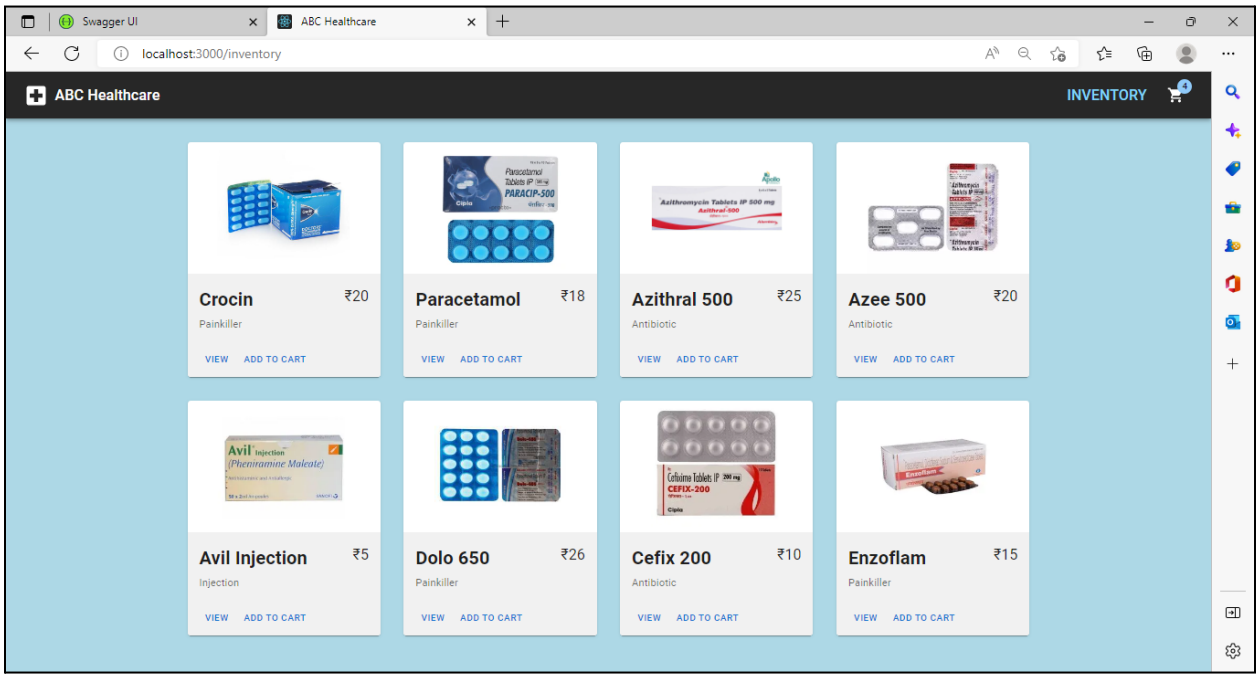


Fig. 53: Adding medicines to Cart

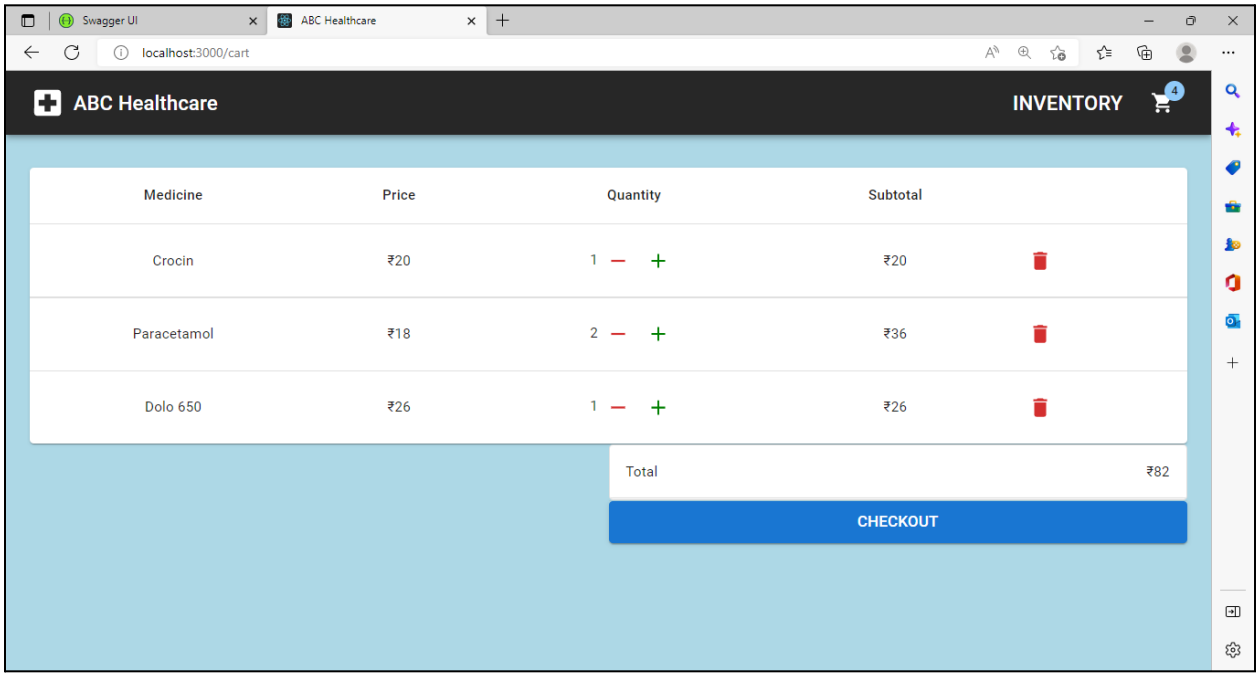


Fig. 54: Cart Page

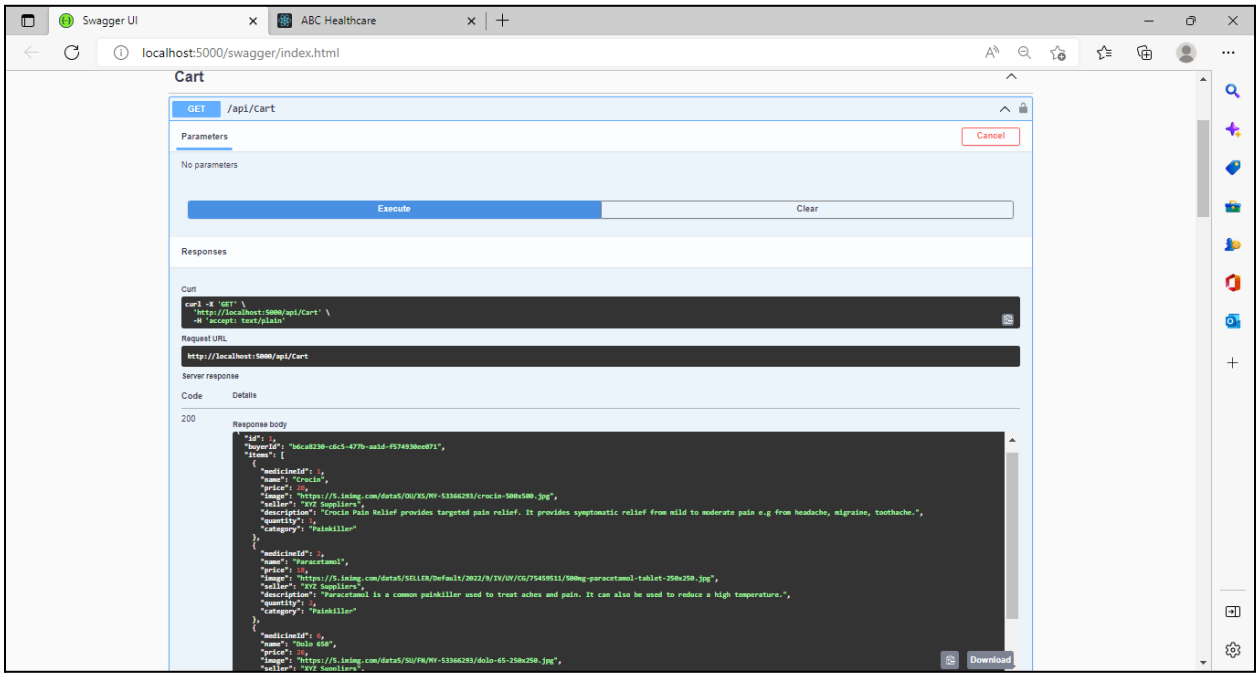


Fig. 55: Cart GET method

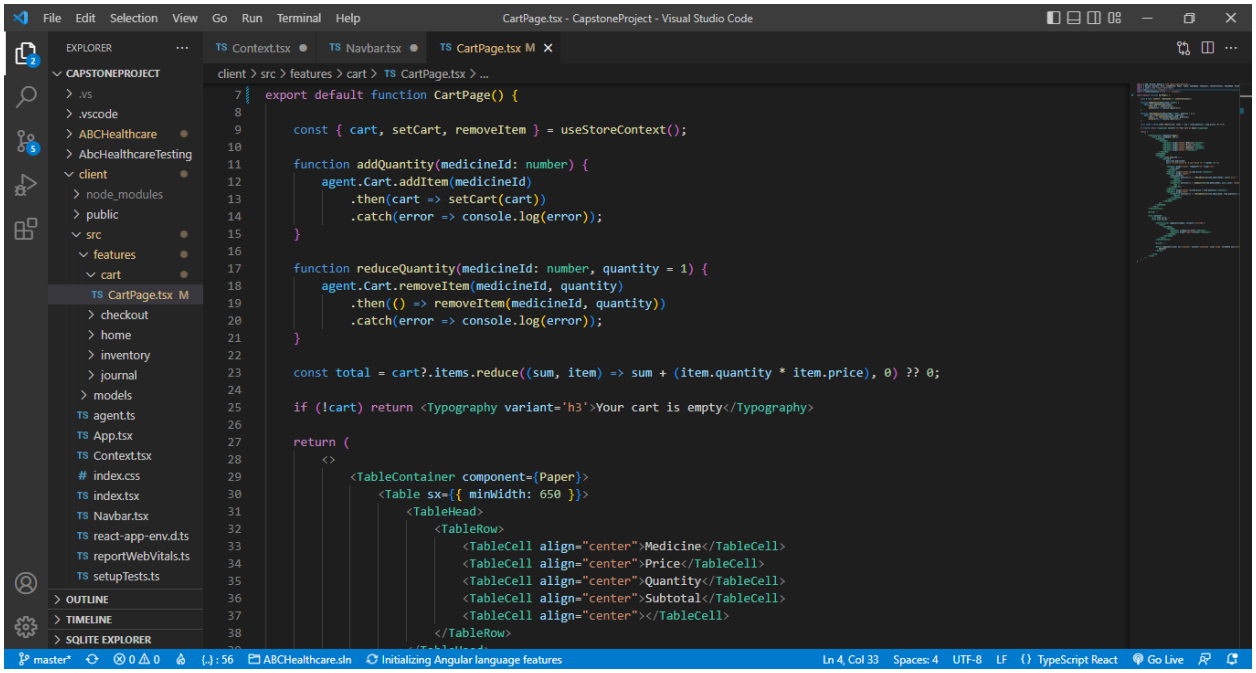


Fig. 56: CartPage.tsx

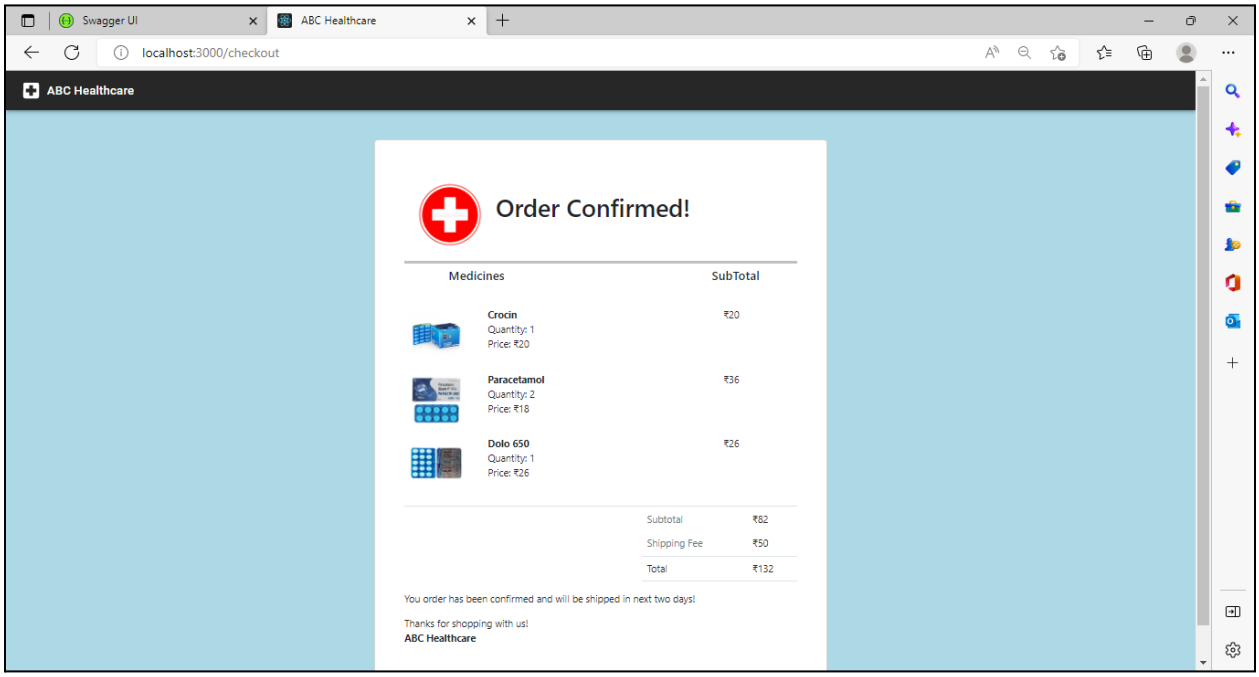


Fig. 57: Checkout Page

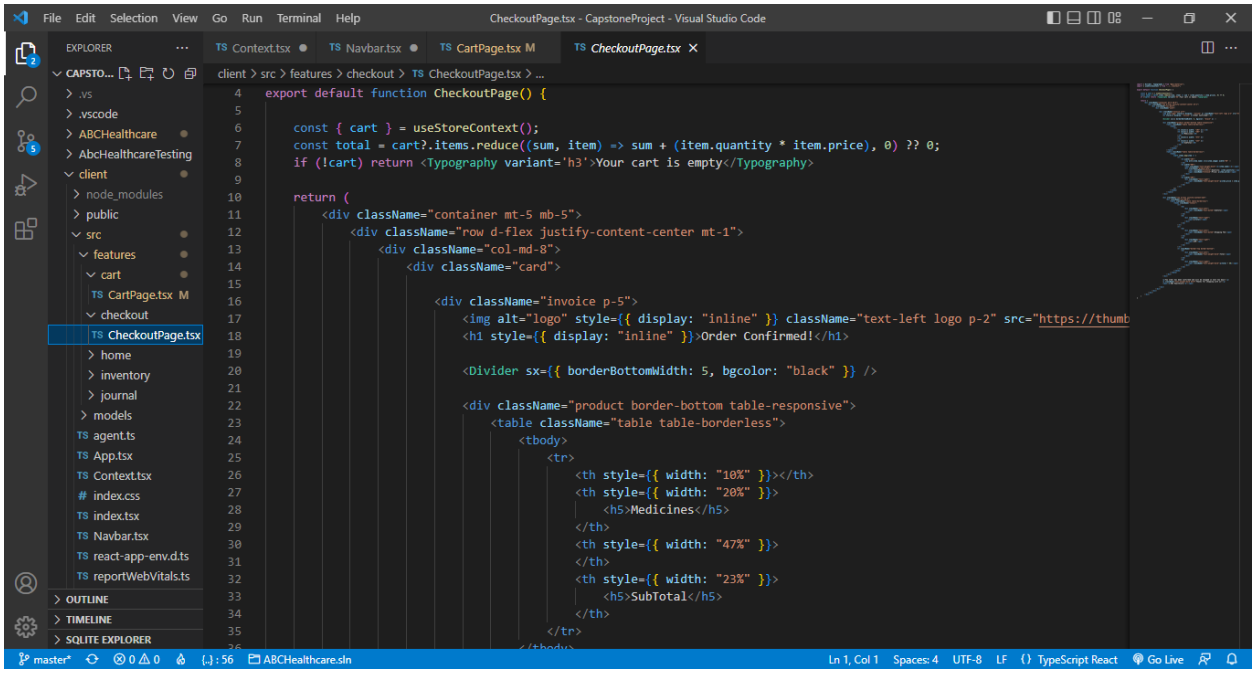


Fig. 58: CheckoutPage.tsx

Testing

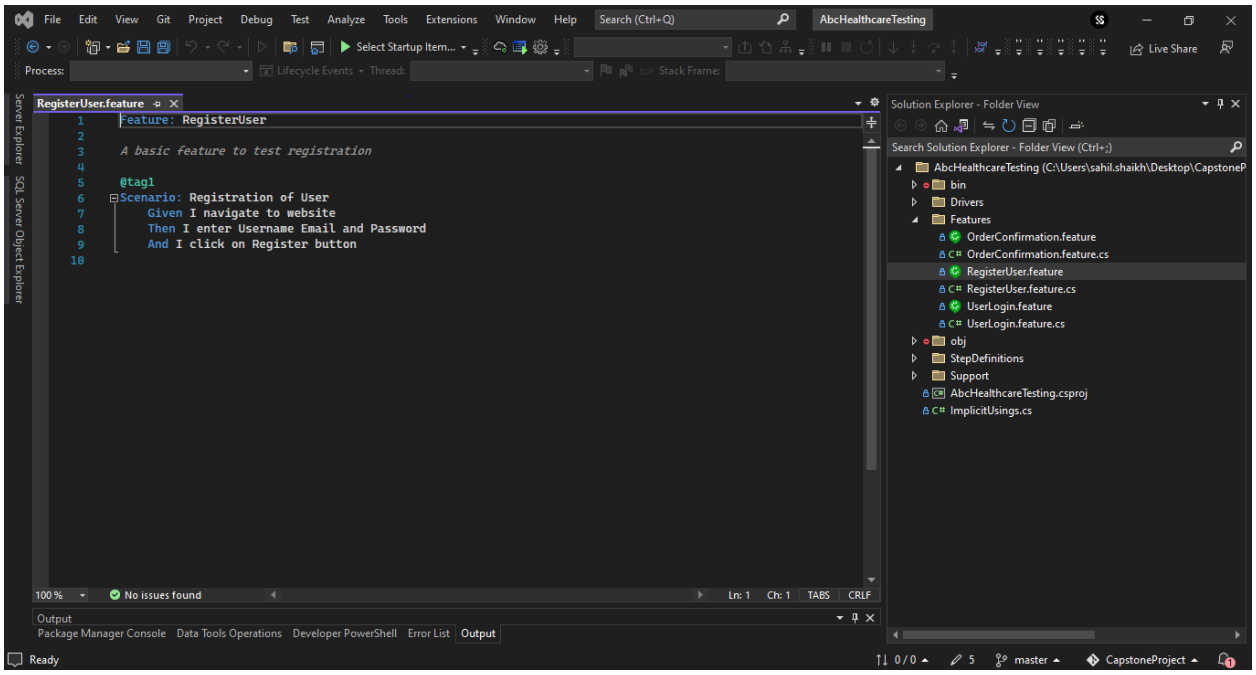


Fig. 59: RegisterUser.feature

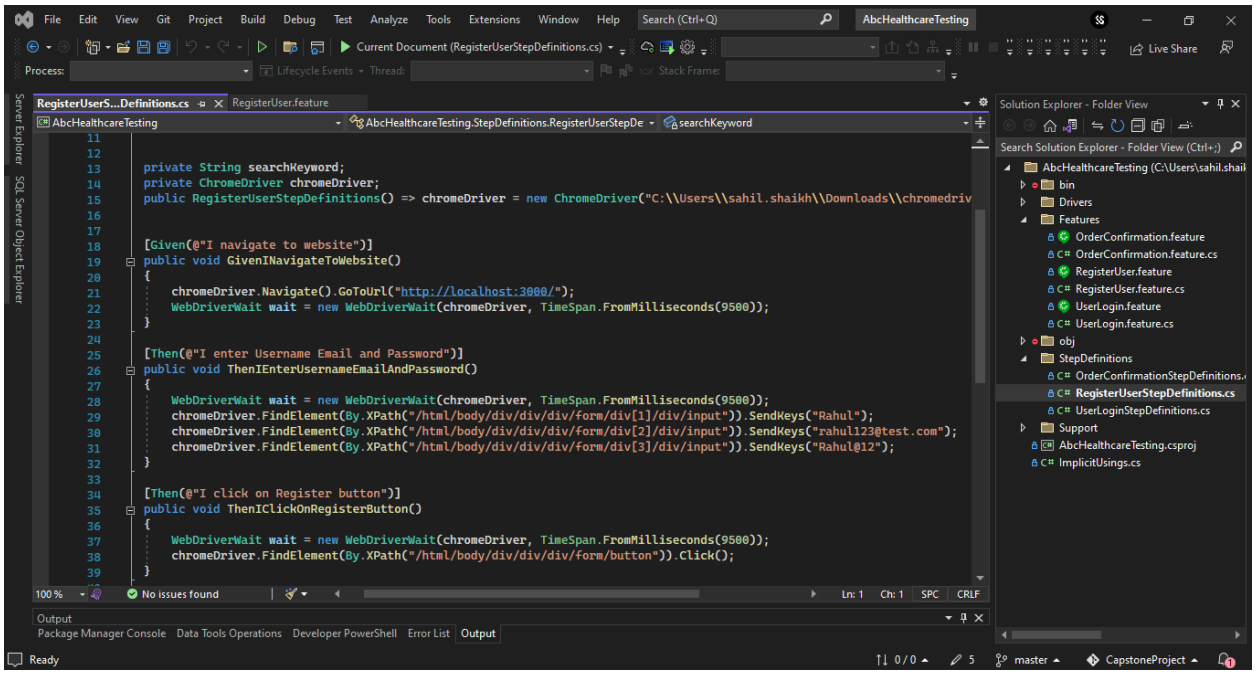


Fig. 60: RegisterUserStepDefinitions.cs

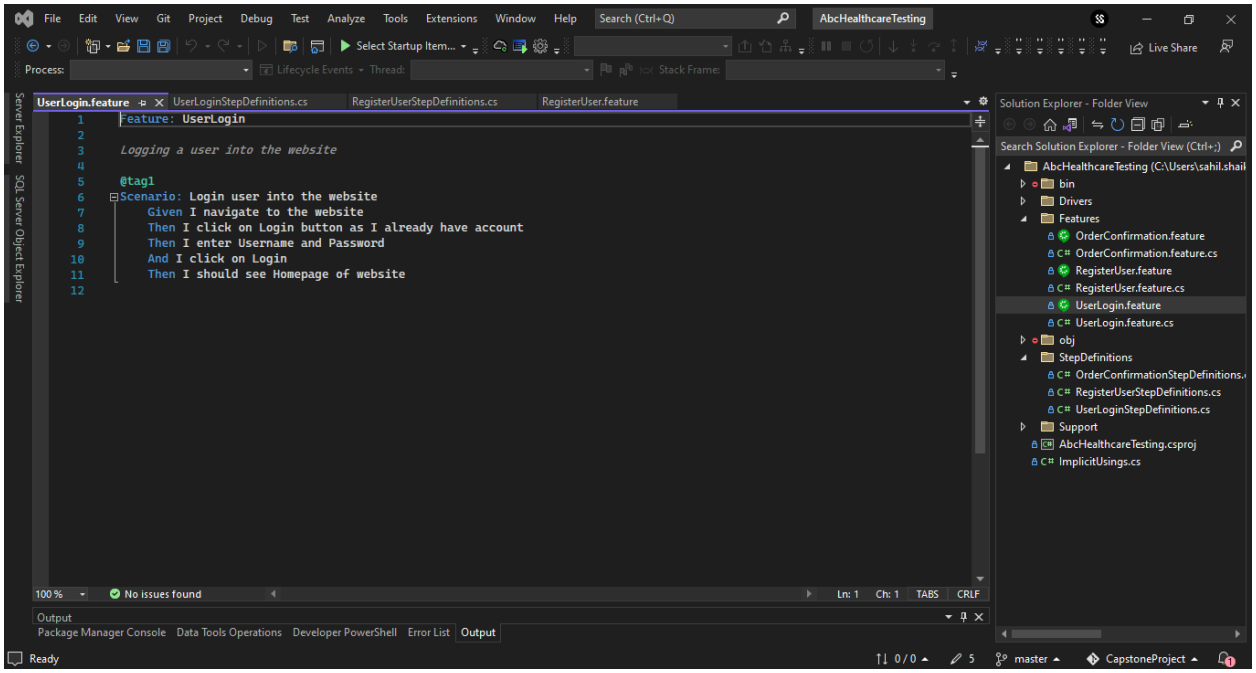


Fig. 61: UserLogin.feature

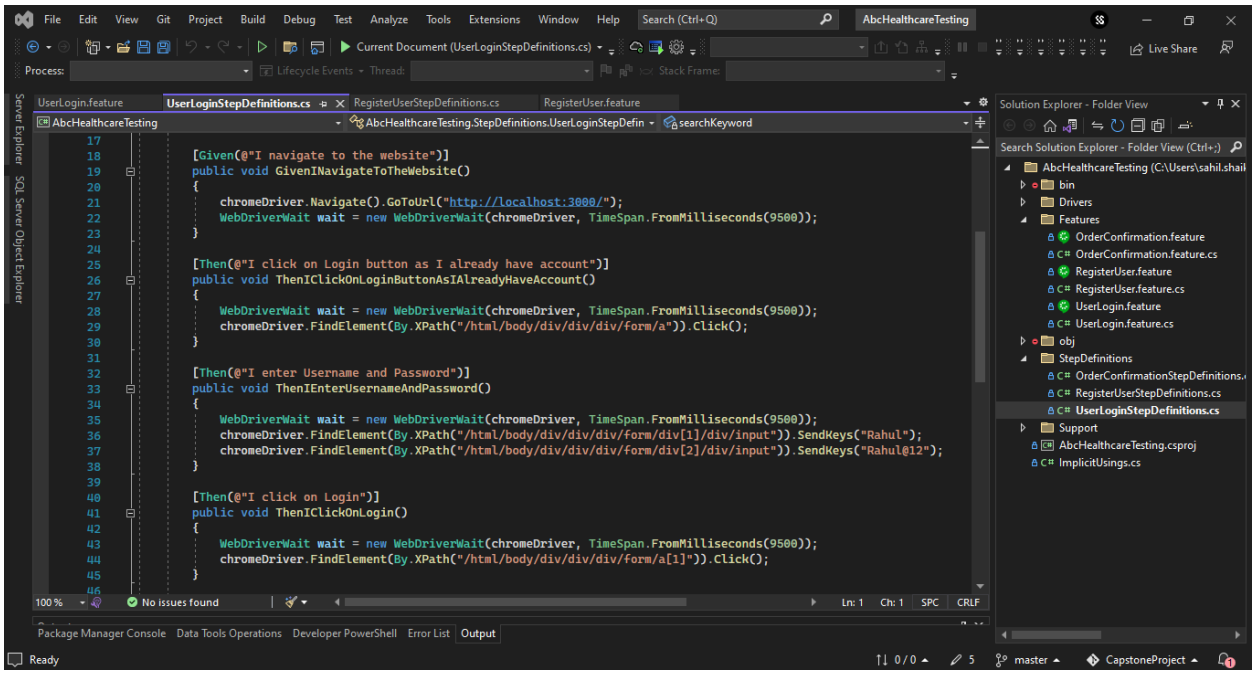


Fig. 62: UserLoginStepDefinitions.cs

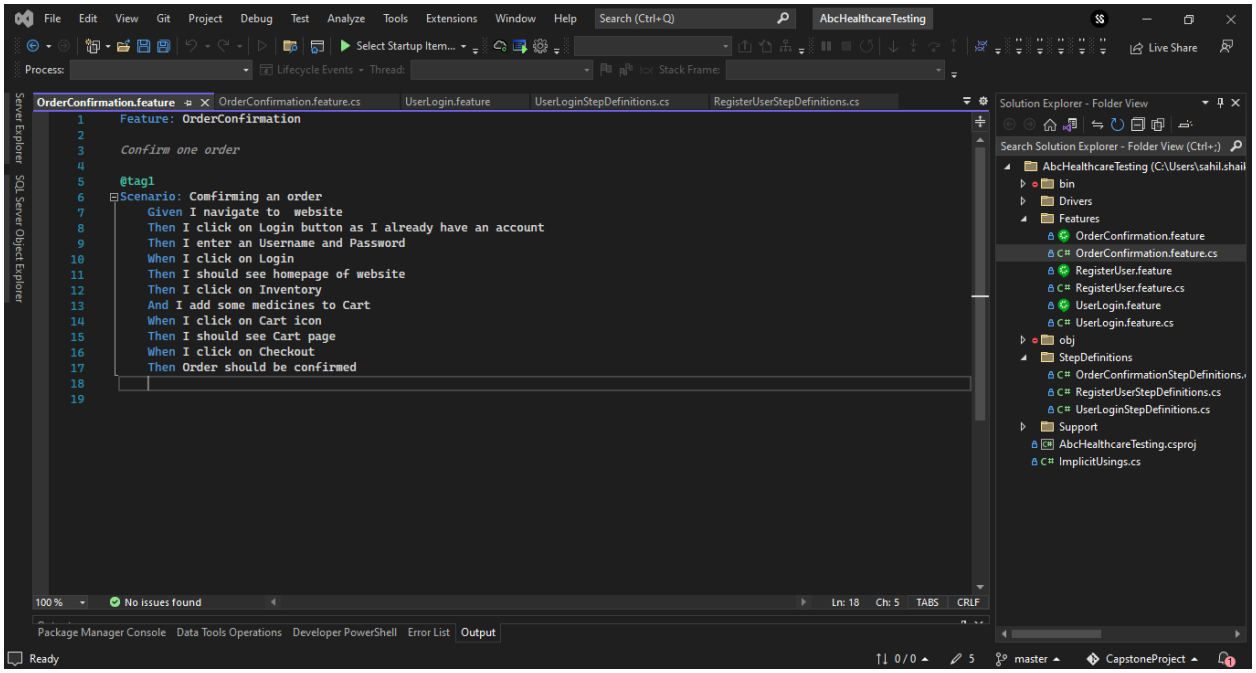


Fig. 63: OrderConfirmation.feature

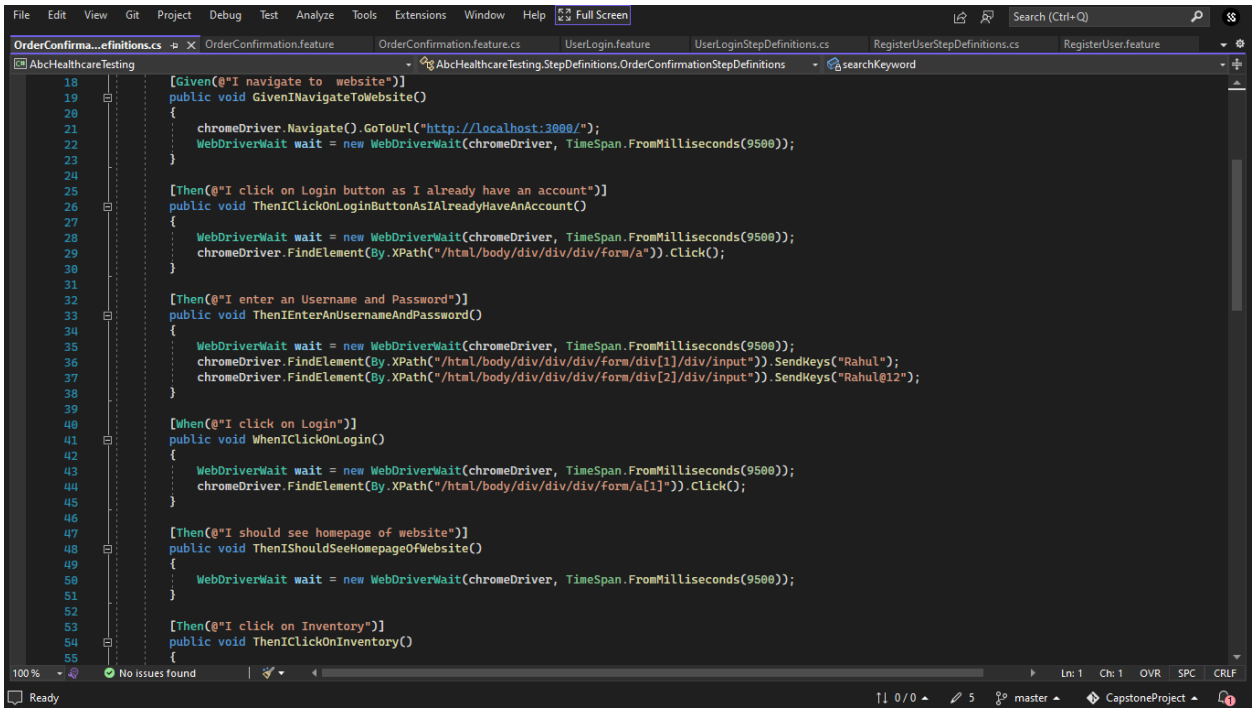
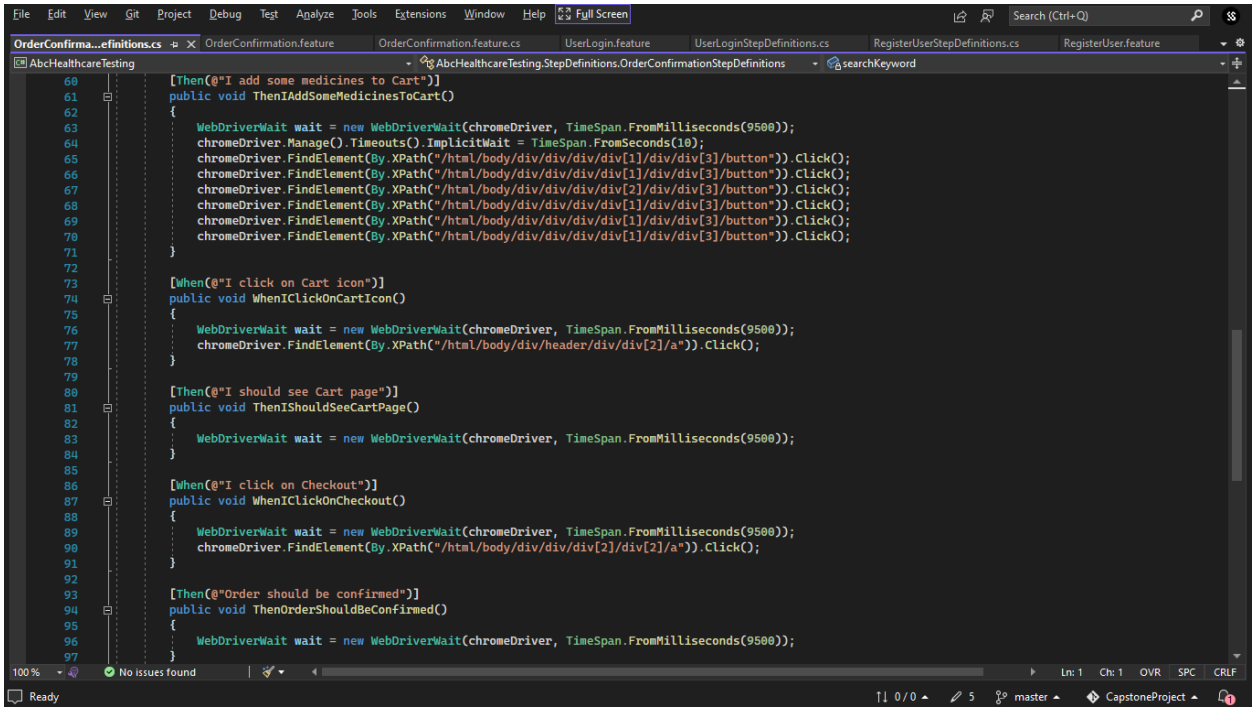


Fig. 64: OrderConfirmationStepDefinitions.cs



```
[Then(@"I add some medicines to Cart")]
public void ThenIAddSomeMedicinesToCart()
{
    WebDriverWait wait = new WebDriverWait(chromeDriver, TimeSpan.FromMilliseconds(9500));
    chromeDriver.Manage().Timeouts().ImplicitWait = TimeSpan.FromSeconds(10);
    chromeDriver.FindElement(By.XPath("/html/body/div/div/div/div[1]/div/div[3]/button")).Click();
    chromeDriver.FindElement(By.XPath("/html/body/div/div/div/div[1]/div/div[3]/button")).Click();
    chromeDriver.FindElement(By.XPath("/html/body/div/div/div/div[2]/div/div[3]/button")).Click();
    chromeDriver.FindElement(By.XPath("/html/body/div/div/div/div[1]/div/div[3]/button")).Click();
    chromeDriver.FindElement(By.XPath("/html/body/div/div/div/div[1]/div/div[3]/button")).Click();
    chromeDriver.FindElement(By.XPath("/html/body/div/div/div/div[1]/div/div[3]/button")).Click();
}

[When(@"I click on Cart icon")]
public void WhenIClickOnCartIcon()
{
    WebDriverWait wait = new WebDriverWait(chromeDriver, TimeSpan.FromMilliseconds(9500));
    chromeDriver.FindElement(By.XPath("/html/body/div/header/div/div[2]/a")).Click();
}

[Then(@"I should see Cart page")]
public void ThenIShouldSeeCartPage()
{
    WebDriverWait wait = new WebDriverWait(chromeDriver, TimeSpan.FromMilliseconds(9500));
}

[When(@"I click on Checkout")]
public void WhenIClickOnCheckout()
{
    WebDriverWait wait = new WebDriverWait(chromeDriver, TimeSpan.FromMilliseconds(9500));
    chromeDriver.FindElement(By.XPath("/html/body/div/div/div[2]/div[2]/a")).Click();
}

[Then(@"Order should be confirmed")]
public void ThenOrderShouldBeConfirmed()
{
    WebDriverWait wait = new WebDriverWait(chromeDriver, TimeSpan.FromMilliseconds(9500));
}
```

Fig. 65: OrderConfirmationStepDefinitions.cs (contd…)

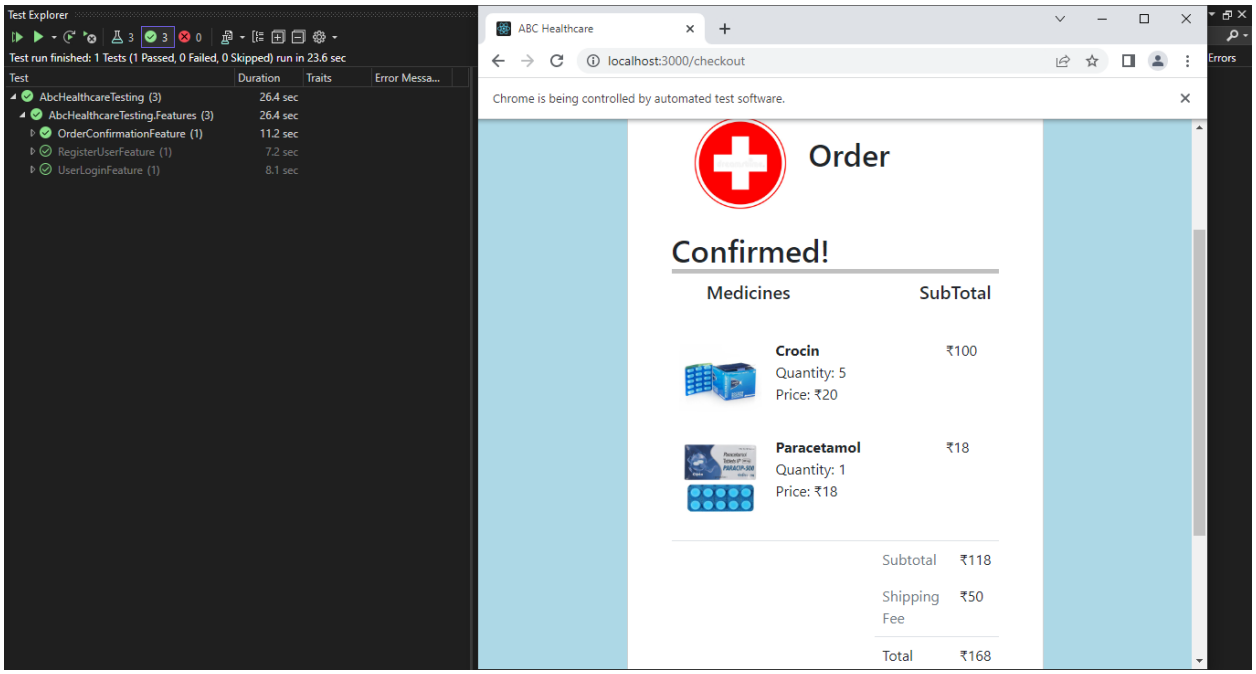


Fig. 66: All testcases passed