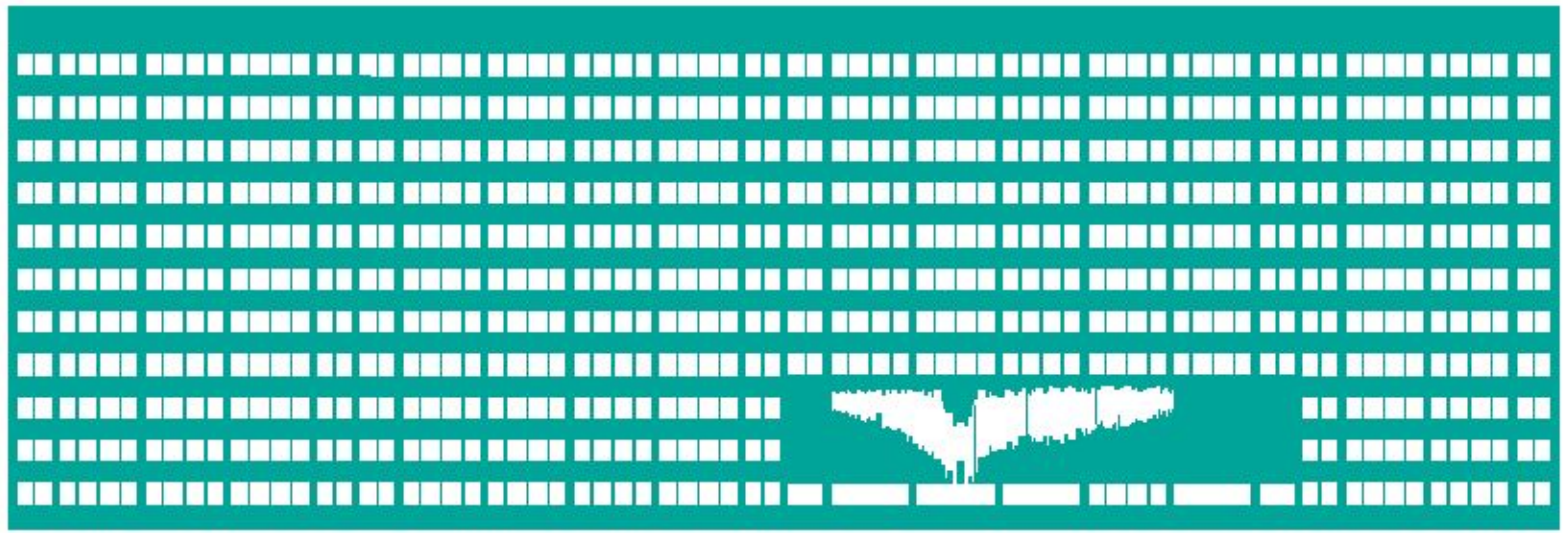
VŠB TECHNICKÁ UNIVERZITA OSTRAVA

VSB TECHNICAL UNIVERSITY OF OSTRAVA



www.vsb.cz

VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Development of Modern Web Applications

Bachelor Thesis

Student: Thanh Tuan Nguyen
Supervisor: Ing. Michal Radecký, Ph.D., MBA

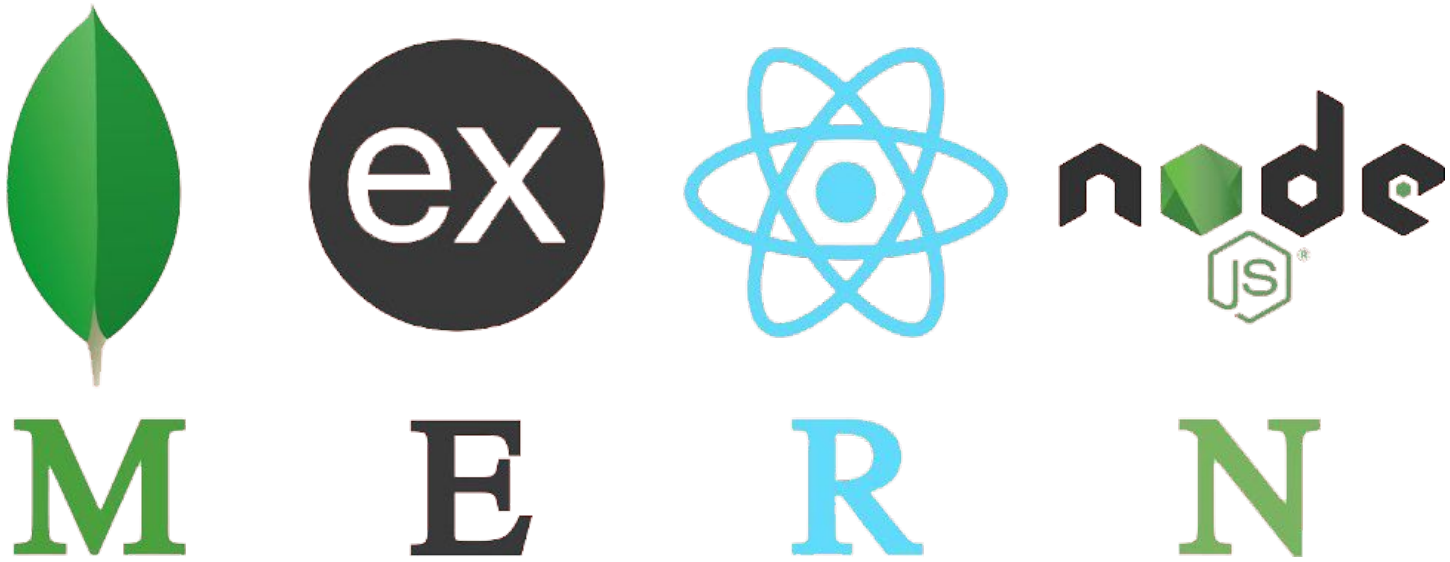
Purpose

- The purpose of the thesis is to create a modern website with any of today's popular and diverse web programming languages. With the implementation in the thesis, users can understand how to use this language.

Framework/Library

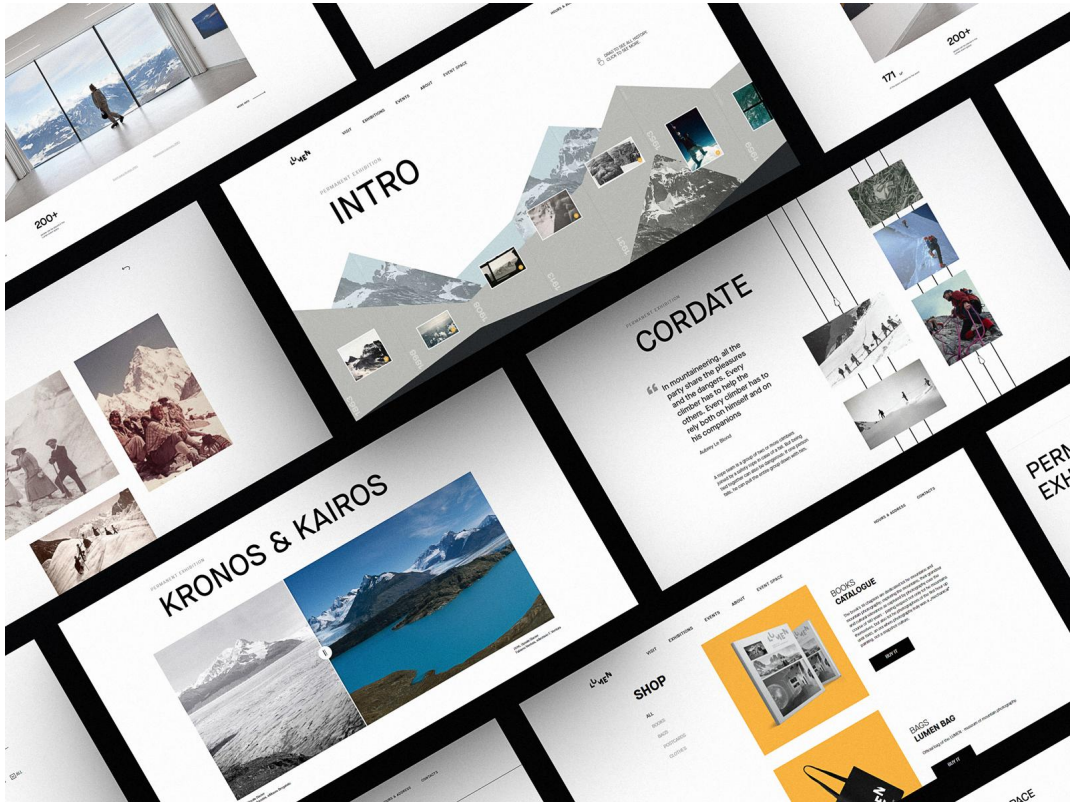
1. Front-end: React
2. Back-end: Node and Express
3. Database: MongoDB

=> MERN Stack



1. Based on case study and design it

- Currently, with the development of the internet, people's needs to use the internet are increasing and websites are gradually becoming more diverse to attract users to access.
- So, to design a website, developers need to consult many websites to be able to create a website for themselves.



2. Implement Page and Feature

Page:

- Home Page
- Product Page
- Product Details Page
- About Page
- Contact Page
- Cart Page
- E-commerce, Size Guide, FAQ, Ship Policy, Privacy Policy, Return Policy Page
- Login and Register Page

Feature:

- Sign in, sign up, sign out.
- Search product.
- Dark/light mode.
- Localization
- Filter
- Change product details in the cart (Add or remove product in the cart).
- Enter payment details, including customer's information, and other required information.
- Quick Buy
- Responsive

3. Some Implementation Problem

- There are quite a few problem and difficulties in the implementation process, the important thing is shopping cart and login and registration because they have to use Redux to store data to LocalStorage. I'm not good at Redux, so I also tried to refer to many sources to fix and fix a lot to make them saveable. And with some problem about Dark Mode.
- And then I want to improve my product, so I use Backend, even though I'm not perfect because Backend I'm not good at and fully understand them even React I'm not confident that I'm perfect. But in order to work properly when executing, I also find it to be temporary.

```
import { createSlice } from '@reduxjs/toolkit';

const cartSlice = createSlice({
    name: 'cart',
    initialState: {
        cart: [],
        users: [],
        user: {},
        numberCart: 0,
    },
    reducers: {
        addToCart: (state, action) => {
            console.log(action.payload);
            const itemInCart = state.cart.find((item) => item.id === action.payload.id);
            if (itemInCart) {
                itemInCart.quantity += action.payload.quantity;
            } else {
                state.cart.push({ ...action.payload });
                state.numberCart++;
            }
        },
        incrementQuantity: (state, action) => {
            const item = state.cart.find((item) => item.id === action.payload);
            item.quantity++;
        },
        decrementQuantity: (state, action) => {
            const item = state.cart.find((item) => item.id === action.payload);
            if (item.quantity === 1) {
                item.quantity = 1;
            } else {
                item.quantity--;
            }
        },
        removeItem: (state, action) => {
            const removeItem = state.cart.filter((item) => item.id !== action.payload);
            state.cart = removeItem;
            state.numberCart--;
        },
```

3. Some Implementation

```
function Tshirt() {
    const {t} = useTranslation()
    const [sizes, setSizes] = useState([
        { symbol: 'S', cm: 73, title: '1m55 - 1m59', weight:'48kg - 54kg' },
        { symbol: 'M', cm: 74, title: '1m60 - 1m65' , weight:'55kg - 61kg'},
        { symbol: 'L', cm: 75, title: '1m66 - 1m72' , weight:'62kg - 68kg'},
        { symbol: 'XL', cm: 76, title: '1m72 - 1m77', weight:'69kg - 75kg' },
        { symbol: '2XL', cm: 77, title: '1m77 - 1m84', weight:'76kg - 84kg' },
        { symbol: '3XL', cm: 77, title: '1m85 - 1m92', weight:'85kg - 92kg' }

    ]);
    const [measurements, setMeasurements] = useState([
        { label: t('size.content7'), sizes: [65, 67, 69, 71, 73, 75] },
        { label: t('size.content5'), sizes: [48, 50, 52, 54, 56, 58] },
        { label: t('size.content6'), sizes: [59, 60, 61, 62, 63, 64] },
        { label: t('size.content8'), sizes: [41.5, 43, 44.5, 46, 47.5, 49] }
    ]);

    return (
        <div className="size-item ">
            <div className="size-table">
                <table className="size-tablechart">
                <thead>
                        <tr>
                            <td><span>SYMBOL/<br/>PARAMETER(cm)</span></td>
                            {sizes.map((size, index) => (
                                <td key={index}>
                                    <span>{size.symbol}</span>
                                    <span className="table-title">
                                        {size.title}<br/>
                                        {size.weight}
                                    </span>
                                </td>
                            ))}
                        </tr>
                    </thead>
                    <tbody>
```

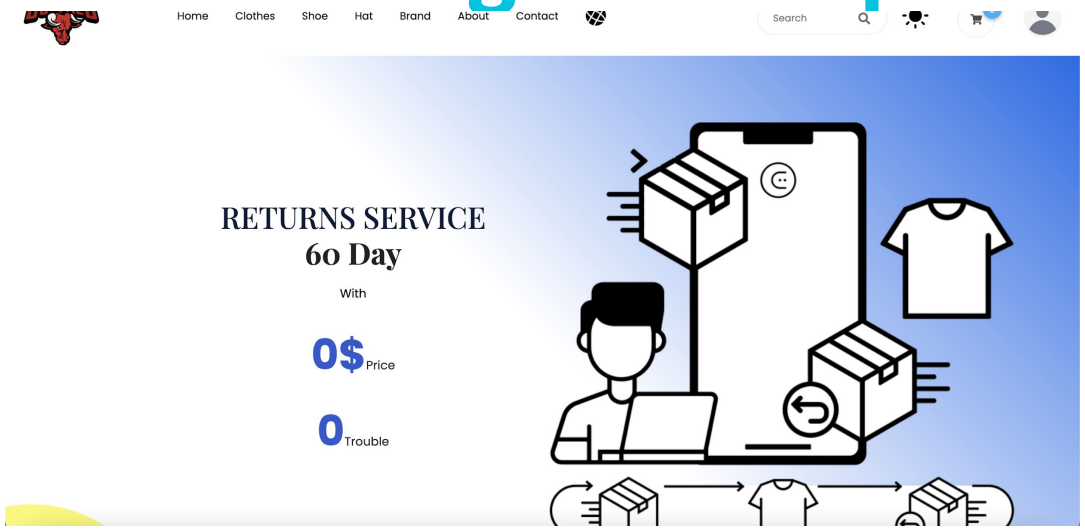
```
useEffect(() => {
    localStorage.setItem('theme', theme);
    document.body.className = theme;
    if (theme === 'dark') {
        toggleDarkMode(true);
    } else {
        toggleDarkMode(false);
    }
    document.getElementById('navigation').className = theme;
    // add class in class name_product
    const products = document.getElementsByClassName('name_product');
    for (let i = 0; i < products.length; i++) {
        products[i].className = 'name_product ' + theme;
    }
    document.getElementById('header-sticky').className = 'row align-items-center ' + theme;
    document.getElementById('header1').style.backgroundColor = theme === 'light' ? '#fff' : '#333';
    document.getElementsByClassName('slicknav_menu').className = theme;
    document.getElementById('search-submit').className = 'form-box f-right ' + theme;
    document.getElementById('dark-mode-btn').className = theme;
    document.getElementById('shopping-card-color').className = 'shopping-card ' + theme;
    const contentDown = document.getElementsByClassName('content_down');
```

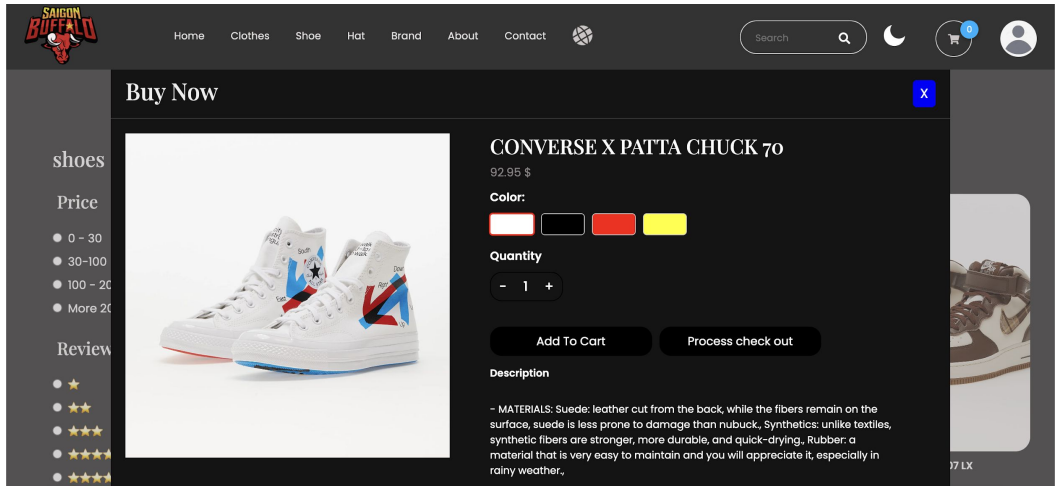
3. Some Implementation

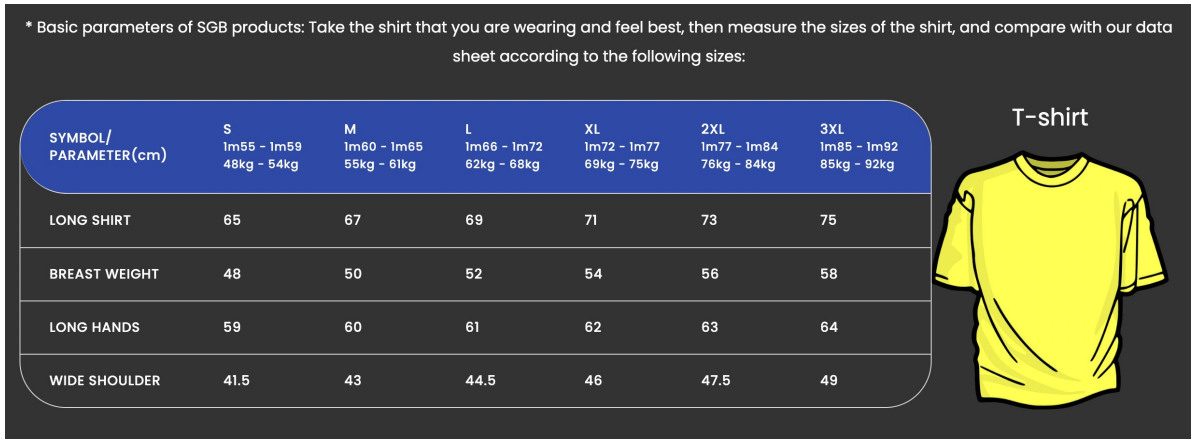
```
const UserController = {
    register: (req, res) => {
        try {
            // get the data from the request
            const { fullname, phone, email, password } = req.body;

            // create a new user
            const user = new User({
                fullname,
                phone,
                email,
                password
            });

            // save the user to the database
            user.save()
            // send api response created
            .then(() => res.status(200).json({message: 'User created'}))
        } catch (error) {
            res.status(500).json({message: 'Error creating user'});
        }
    },
    login: (req, res) => {
        // get the data from the request
        const { email, password } = req.body;

        // find the user in the database
        User.findOne({email: email})
        .then(user => {
            // check if the user exists
            if(!user) {
                res.status(404).json({message: 'User not found'});
            }
```

```
const [openModal, setOpenModal] = useState(false);
return (
    <>
        <div className="slider-area " style={{ textAlign: 'center' }}>
            <h1 style={{ color: 'black', paddingTop: 50 }}>{t('cart.list')}</h1>
        </div>

        <section className="cart_area section_padding" style={{ paddingTop: 50 }}>
            <div className="container">
                <div className="cart_inner">
                    <div className="table-responsive">
                        <CartTable />

                        <div className="checkout_btn_inner float-right pos-link">
                            <Link className="btn_1" to="/">
                                {t('cart.continue')}
                            </Link>
                                {user ? (
                                    <div
                                        className="btn_1 checkout_btn_1"
                                        style={{ cursor: 'pointer' }}
                                        onClick={() => setOpenModal(true)}
                                    >
                                        {t('cart.process')}
                                    </div>
                                ): (
                                    <Link className="btn_1 checkout_btn_1" to="/login">
                                        {t('cart.login')}
                                    </Link>
                                )}
                        </div>
                    </div>
                </div>
            </div>
        </section>
        {openModal && cart.length > 0 && <Modal onOpen={setOpenModal} />}
    </>
```

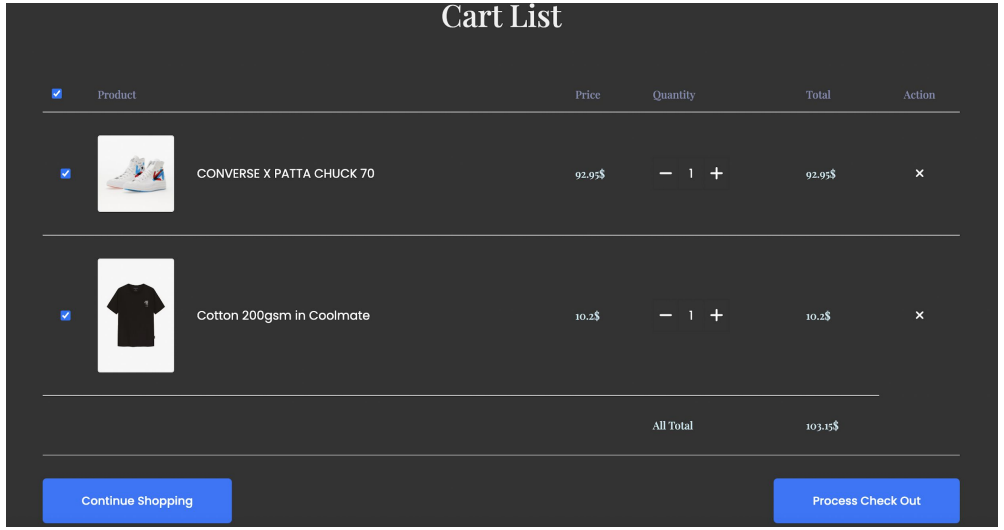
4. Some Image for implementation



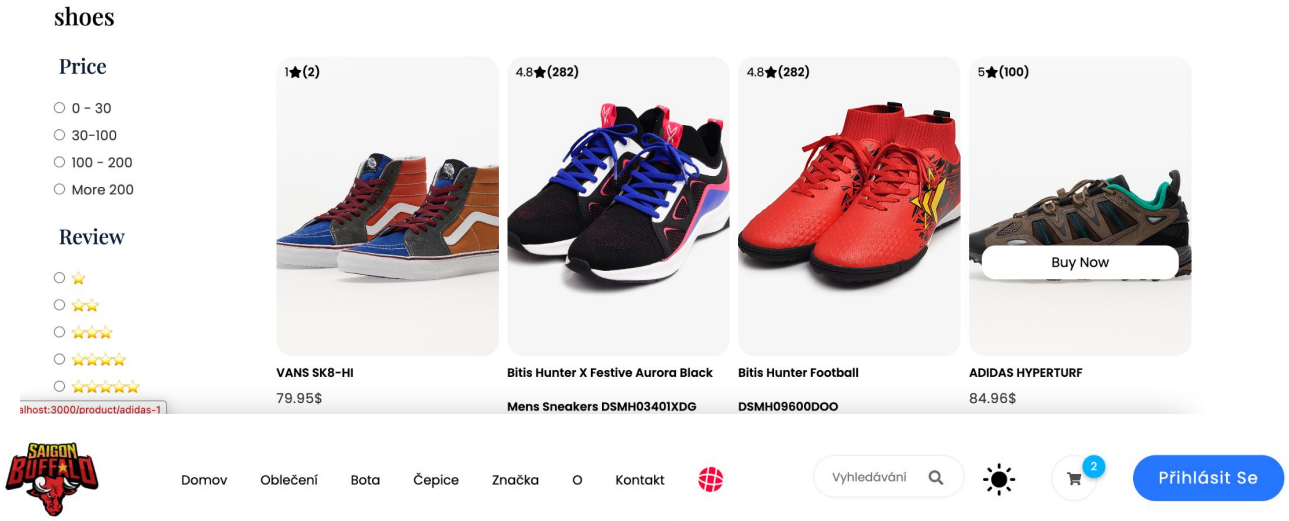


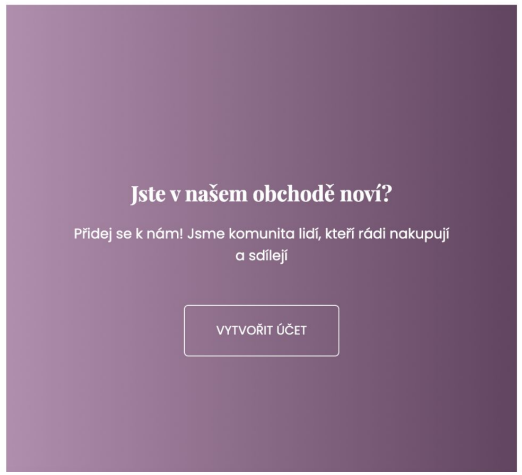


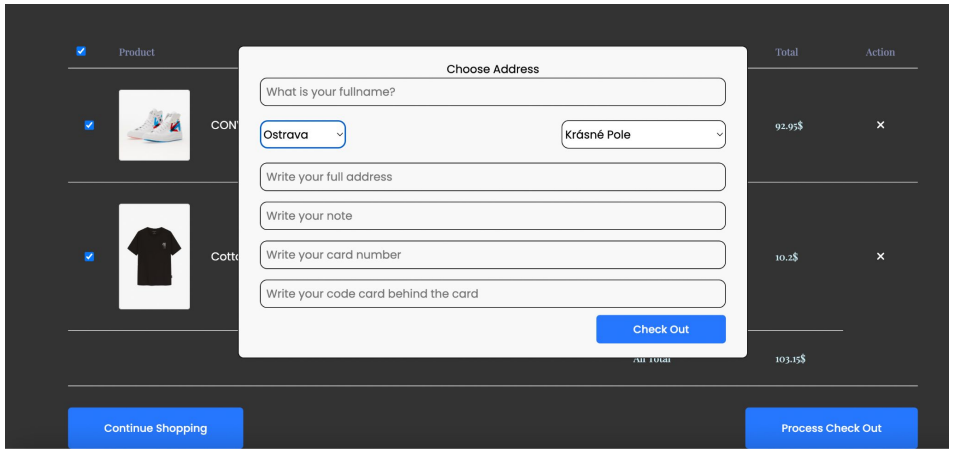
SYMBOL/ PARAMETER(cm)	S 1m55 - 1m59 48kg - 54kg	M 1m60 - 1m65 55kg - 61kg	L 1m66 - 1m72 62kg - 68kg	XL 1m72 - 1m77 69kg - 75kg	2XL 1m77 - 1m84 76kg - 84kg	3XL 1m85 - 1m92 85kg - 92kg
LONG SHIRT	65	67	69	71	73	75
BREAST WEIGHT	48	50	52	54	56	58
LONG HANDS	59	60	61	62	63	64
WIDE SHOULDER	41.5	43	44.5	46	47.5	49



4. Some Image for implementation

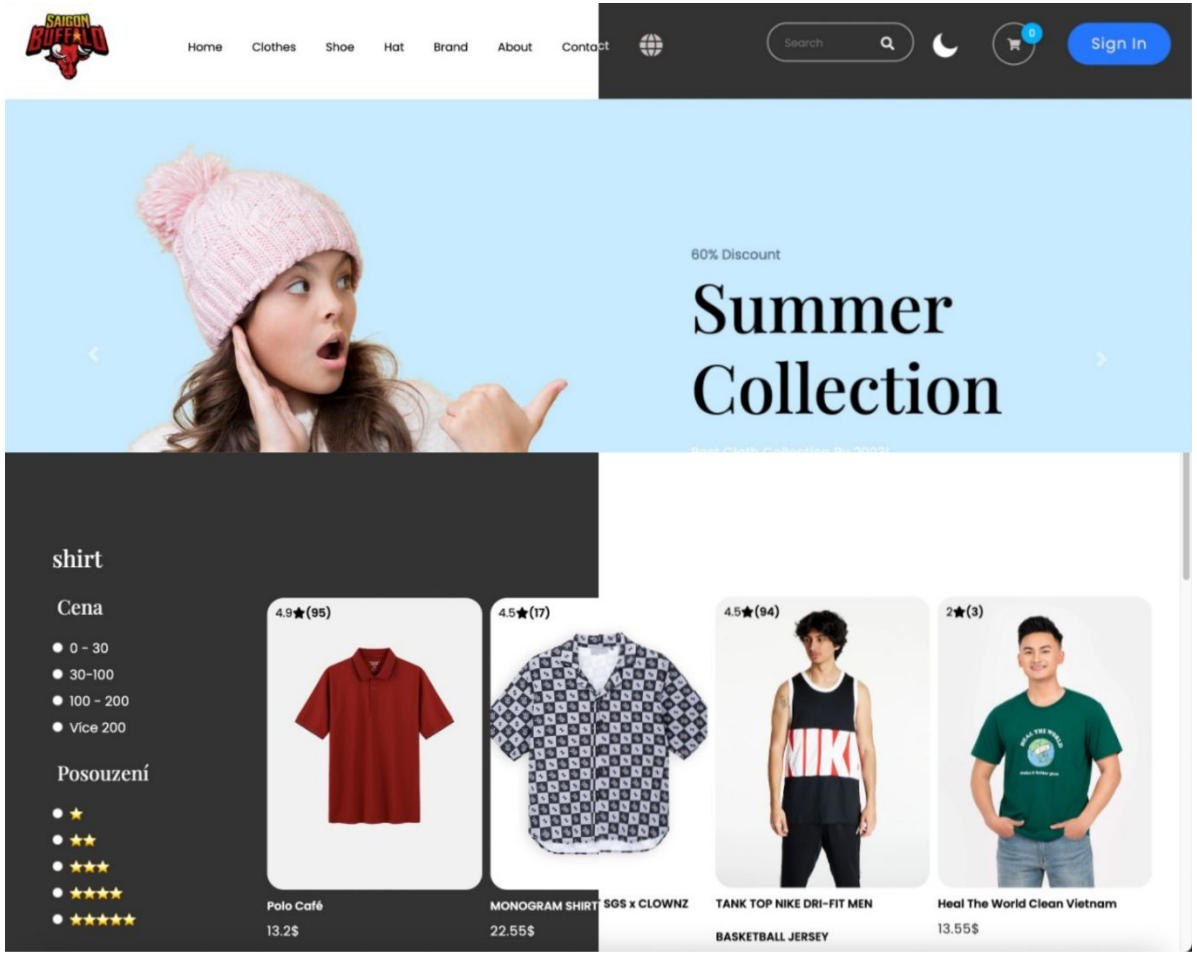






5. Evaluation

- After completing the e-commerce project, there are still many shortcomings. The speed of startup and display after activating npm start is still stable and guaranteed, users can access and use it, the product page loads data from the Back-end is still stable without lag. These functions and the website are fully functional.
- In the future, besides overcoming shortcomings, I will continuously update technology and perfect my products. So that they will be more modern and better application.



Thank you for your attention

Question of Supervisor

Question:

1/ Why did you choose the framework you did for the implementation of the eshop?

2/ What do you consider to be the biggest problem when developing an e-shop using the chosen technologies?

3/ How would you briefly summarize the main advantages of HTML 5 in terms of development and operation of modern e-shops?

Answer:

1/ Because ReactJS is one of the most popular and widely used frameworks/libraries. So there will be a lot of supporting documents during the project implementation that will minimize the difficulties that users encounter.

2/ My difficulty when doing e-shop is connecting and using technologies together, because sometimes I don't know how to connect data and I am also quite confused in learning how to use each and apply them to products, especially the shopping cart section. I'm having a lot of trouble using Redux to store data for my shopping cart.

3/HTML5 offers improved multimedia support, enhanced semantic structure, mobile-friendly and responsive design, offline capabilities, simplified form handling, better performance, and efficiency. These advantages facilitate the development and operation of modern e-shops.

Question of Opponent

Question:

1/ Why did you choose React for your implementation?

2/ Do you believe that it is appropriate to implement the frontend of the e-shop as a single-page application?

Answer:

1/ Because ReactJS is one of the most popular and widely used frameworks/libraries. So there will be a lot of supporting documents during the project implementation that will minimize the difficulties that users encounter.

2/ Yes, implementing the frontend of an e-shop as a single-page application (SPA) can be appropriate in many cases. SPAs provide a smooth and interactive user experience by loading the necessary resources upfront and dynamically updating the content without full page reloads.