

**Московский государственный технический
университет им. Н.Э. Баумана**

Факультет “Информатика и системы управления”
Кафедра “Системы обработки информации и управления”

Курс «Парадигмы и конструкции языков программирования»

Отчет по домашнему заданию

Выполнил:
Студент группы ИУ5-31Б
Попов А.С.
Подпись и дата:

Проверил:
Преподаватель каф. ИУ5
Гапанюк Ю.Е.
Подпись и дата:

Москва, 2025 г.

ОГЛАВЛЕНИЕ

1. Задание для домашнего задания
2. Описание, листинг кода
3. Скриншоты работы

Задание для домашнего задания

1. Выберите язык программирования и реализуйте на нем небольшой проект (с детальным текстовым описанием).
2. Проект может быть посвящен отдельному аспекту (аспектам) языка или содержать решение какой-либо задачи на этом языке.
3. Необходимо установить на свой компьютер компилятор (интерпретатор) этого языка и произвольную среду разработки.
4. В случае создания проекта необходимо детально комментировать код.
5. При создании проекта необходимо изучить и корректно использовать особенности парадигмы языка и основных конструкций данного языка.

Описание проекта, листинг кода

Backend:

main.go

```
package main

import (
    "net/http"

    // "encoding/json"
    "github.com/go-chi/chi/v5"
    "github.com/go-chi/chi/v5/middleware"
    "github.com/go-chi/cors"
    "github.com/kutsushop/internal/handler"
)

func main() {
    r := chi.NewRouter()

    r.Use(middleware.Logger)
    r.Use(cors.Handler(cors.Options{
        AllowedOrigins: []string{ "*" },
        AllowedMethods: []string{ "GET", "POST", "PUT",
        "DELETE" },
        AllowedHeaders: []string{ "*" },
    }))

    DATA_API_URL := "https://cnnjnavkemrzbegsbwbw.supabase.co"
    DATA_API_KEY := "sb_secret_6biUP7aTRu5c-FylFAO-JA_jYhl2SUn"

    USERS_API_URL := "https://cnnjnavkemrzbegsbwbw.supabase.co"
    USERS_API_KEY := "sb_secret_6biUP7aTRu5c-FylFAO-JA_jYhl2SUn"

    h_data := handler.New(DATA_API_URL, DATA_API_KEY)
    h_users := handler.New(USERS_API_URL, USERS_API_KEY)

    r.Get("/data", h_data.GetAllData)
    r.Get("/sellers", h_data.GetAllSellers)
    r.Get("/shoes", h_data.GetAllShoes)

    r.Post("/shoes_sales", h_data.CreateShoesSales)
    r.Post("/create_sellers", h_data.CreateSellers)

    r.Get("/users", h_users.GetAllData)
    r.Post("/create_users", h_users.CreateUsers)
```

```

        r.Get("/excel", h_data.GetExcelFile)

        r.Post("/update_users", h_users.UpdateUsers)
        r.Post("/update_sellers", h_data.UpdateSellers)

        http.ListenAndServe(":3000", r)
    }

handler.go
package handler

import (
    "encoding/json"
    "fmt"
    "net/http"
    "strconv"
    "strings"

    "github.com/kutsushop/internal/pipes"
    "github.com/supabase-community/supabase-go"
    "github.com/xuri/excelize/v2"
)

type Handler struct {
    client *supabase.Client
}

func New(api_url, api_key string) *Handler {
    client, err := supabase.NewClient(api_url, api_key, nil)
    if err != nil {
        fmt.Println("cannot initialize client", err)
    }

    return &Handler{
        client,
    }
}

func (h *Handler) GetAllData(w http.ResponseWriter, r *http.Request) {
    tableParam := r.URL.Query().Get("table")
    fmt.Println(tableParam)

    data, _, _ := h.client.From(tableParam).Select("*", "exact",
false).Execute()

    //fmt.Println(string(data[:]))
    w.Write(data)
}

```

```

func (h *Handler) GetAllSellers (w http.ResponseWriter, r
*http.Request) {
    data, _, _ := h.client.From("sellers").Select("*", "", false).Execute()

    w.Write(data)
}

func (h *Handler) GetAllShoes (w http.ResponseWriter, r
*http.Request) {
    data, _, _ := h.client.From("shoes").Select("*", "", false).Execute()

    w.Write(data)
}

func (h *Handler) CreateShoesSales (w http.ResponseWriter, r
*http.Request) {

    type Payload struct {
        ShoesId int `json:"shoes_id"`
        Price int `json:"price"`
        Quantity int `json:"quantity"`
        SellerID int `json:"seller_id"`
    }

    var p Payload

    json.NewDecoder(r.Body).Decode(&p)

    data_to_insert := map[stringfunc (h *Handler) CreateSellers (w http.ResponseWriter, r
*http.Request) {
    type Payload struct {
        Surname string `json:"surname"`
        Name string `json:"name"`
}

```

```

        MiddleName string `json:"middle_name"`
        PhoneNumber string `json:"phone_number"`
    }
var p Payload

    json.NewDecoder(r.Body).Decode(&p)

    data_to_insert := map[stringfunc (h *Handler) CreateUsers(w http.ResponseWriter, r *http.Request) {
    type Payload struct {
        Email string `json:"email"`
        Password string `json:"password"`
    }
    var p Payload

    json.NewDecoder(r.Body).Decode(&p)

    data_to_insert := map[stringfunc (h *Handler) UpdateUsers(w http.ResponseWriter, r *http.Request) {

    type Payload struct {
        Id int `json:"id"`
        Position string `json:"position"`
    }
}

```

```

var p Payload

json.NewDecoder(r.Body).Decode(&p)

data_to_update := map[string]any{
    "id": p.Id,
    "position": p.Position,
}

h.client.From("users").Update(data_to_update, "", "")
Eq("id", strconv.Itoa(p.Id)).Execute()
}

func (h *Handler) UpdateSellers(w http.ResponseWriter, r *http.Request) {

    type Payload struct {
        Id      int      `json:"id"`
        Surname string  `json:"surname"`
        Name    string  `json:"name"`
        MiddleName string `json:"middle_name"`
        PhoneNumber string `json:"phone_number"`
    }

    var p Payload

    json.NewDecoder(r.Body).Decode(&p)

    data_to_update := map[string]any{
        "id": p.Id,
        "surname": p.Surname,
        "name": p.Name,
        "middle_name": p.MiddleName,
        "phone_number": p.PhoneNumber,
    }

    h.client.From("sellers").Update(data_to_update, "", "")
Eq("id", strconv.Itoa(p.Id)).Execute()
}

func Split(r rune) bool {
    return r == '-' || r == 'T' || r == '.'
}

func (h *Handler) GetExcelFile(w http.ResponseWriter, r *http.Request) {

    data, _, _ := h.client.From("sale_of_shoes").Select("*", "", false).Execute()
}

```

```

var excel_data []map[string]any
pipes.FromJson(data, &excel_data)

f := excelize.NewFile()
defer func() {
    if err := f.Close(); err != nil {
        fmt.Println(err)
    }
}()

index, err := f.NewSheet("заказы")
if err != nil {
    fmt.Println(err)
    return
}
f.SetCellValue("Заказы", "A1", "ID заказа")
f.SetCellValue("Заказы", "B1", "ID модели обуви")
f.SetCellValue("Заказы", "C1", "Общая сумма")
f.SetCellValue("Заказы", "D1", "ID продавца")
f.SetCellValue("Заказы", "E1", "Количество")
f.SetCellValue("Заказы", "F1", "Дата создания")
type Target struct {
    Year string
    Month string
    Day string
    Time string
}

for i, item := range excel_data {
    f.SetCellValue("Заказы", fmt.Sprintf("A%d", i+2),
item["id"])
    f.SetCellValue("Заказы", fmt.Sprintf("B%d", i+2),
item["shoes_id"])
    f.SetCellValue("Заказы", fmt.Sprintf("C%d", i+2),
item["proceed"])
    f.SetCellValue("Заказы", fmt.Sprintf("D%d", i+2),
item["seller_id"])
    f.SetCellValue("Заказы", fmt.Sprintf("E%d", i+2),
item["shoes_sale_amount"])

    date_created :=
strings.FieldsFunc(item["date_created"].(string), Split)
    t := Target{date_created[0], date_created[1],
date_created[2], date_created[3]}

    f.SetCellValue("Заказы", fmt.Sprintf("F%d", i+2),
strings.Trim(fmt.Sprintf("%v", t), "{}"))
}

```

```

        }

        f.SetActiveSheet(index)

        filePath := "./заказы.xlsx"

        if err := f.SaveAs("заказы.xlsx"); err != nil {
            fmt.Println(err)
        }

        w.Header().Set("Content-Disposition", "attachment;
filename="+strconv.Quote("заказы.xlsx"))
        w.Header().Set("Content-Type", r.Header.Get("Content-Type"))
        http.ServeFile(w, r, filePath)

        w.WriteHeader(http.StatusOK)
    }
}

```

jsonh.go

```
package pipes
```

```

import (
    "encoding/json"
    "fmt"
)

func ToJson(data interface{}) []byte {
    data_json, err := json.Marshal(data)
    if err != nil {
        println("unmarshal error: ", err)
    }
    return data_json
}

func FromJson[T any](data []byte, store_to T) {
    err := json.Unmarshal(data, &store_to)
    if err != nil {
        fmt.Println("unmarshal error: ", err)
    }
}

```

Frontend:

App.tsx

```

import { useEffect, useState } from 'react'
import { Footer } from './components/Footer'
import { AdminPanel } from './components/AdminPanel'
import { LoginForm } from './components/LoginForm'

import './index.css'

```

```

function App() {
  const [isLoggedIn, setLoggedIn] = useState(false)

  const handleLogin = (flag: boolean, isAdmin: boolean, email: string, position: string) => {
    setLoggedIn(flag)
    window.sessionStorage.setItem('is_auth', String(flag))
    window.sessionStorage.setItem('is_admin', String(isAdmin))
    window.sessionStorage.setItem('email', email)
    window.sessionStorage.setItem('position', position)
  }

  useEffect(() => {
    if (window.sessionStorage.getItem('is_auth') === 'true') {
      setLoggedIn(true)
    }
  }, [])

  return (
    <>
    {isLoggedIn ? <AdminPanel handleLogin={handleLogin} /> :
    <LoginForm handleLogin={handleLogin} />}
    <Footer />
  )
}

export default App

```

Main.tsx

```

import React from 'react'
import ReactDOM from 'react-dom/client'
import App from './App.tsx'
import './index.css'
import { createBrowserRouter, RouterProvider } from 'react-router-dom'
import Orders from './components/pages/orders'
import Shoes from './components/pages/shoes'
import Suppliers from './components/pages/suppliers'
import Supplies from './components/pages/supplies'
import Sellers from './components/pages/sellers'
import ProductForm from './components/ProductForm'
import AdminForm from './components/AdminForm'
import EditForm from './components/EditForm.tsx'

const router = createBrowserRouter([
  {
    path: '/',
  }
])

```

```

element: <App />,
children: [
  {
    path: 'viewdata/shoes',
    element: <Shoes />,
  },
  {
    path: 'viewdata/supplies',
    element: <Supplies />,
  },
  {
    path: 'viewdata/orders',
    element: <Orders />,
  },
  {
    path: 'viewdata/suppliers',
    element: <Suppliers />,
  },
  {
    path: 'viewdata/sellers',
    element: <Sellers />,
  },
],
),
{
  path: '/form',
  element: <ProductForm />,
},
{
  path: '/adminform',
  element: <AdminForm />,
},
{
  path: '/editform',
  element: <EditForm />,
},
])
)

ReactDOM.createRoot(document.getElementById('root')!).render(
<React.StrictMode>
  <RouterProvider router={router} />
</React.StrictMode>
)

```

AdminForm.tsx

```

import axios from 'axios'
import { useState } from 'react'

export default function AdminForm() {

```

```
const [pressed, setPressed] = useState(false)
const onSubmit = (event: React.FormEvent) => {
    setPressed(true)
    event.preventDefault()

    const target = event.target as EventTarget & Record<'surname' | 'name' | 'middlename' | 'phone_number' | 'email' | 'password', {
        value: string
    }

    axios
        .post('http://localhost:3000/create_sellers', {
            surname: String(target.surname.value),
            name: String(target.name.value),
            middle_name: String(target.middlename.value),
            phone_number: String(target.phone_number.value),
        })
        .then(() => {
            console.log('sellers post success')
        })
}

axios
    .post('http://localhost:3000/create_users', {
        email: String(target.email.value),
        password: String(target.password.value),
    })
    .then(() => {
        console.log('users post success')
    })
}

return (
    <div className="bg-gray-100 dark:bg-gray-800 transition-colors duration-300 min-h-full">
        <div className="mx-auto">
            <div className="bg-white dark:bg-gray-900 shadow p-6 border-b h-screen flex justify-center items-center">
                <div className="w-full">
                    <h1 className="text-center text-xl font-semibold mb-4 text-gray-900 dark:text-gray-100">Добавление продавца</h1>
                    <p className="text-center text-white mb-6">Укажите информацию о продавце</p>
                    <form onSubmit={onSubmit} className="w-1/3 mx-auto">
                        <div className="flex w-full justify-between">
                            <div className="mb-4">
                                <label className="block text-sm font-bold mb-2">Фамилия</label>
                                <input name="surname" placeholder="Введите фамилию" className="shadow appearance-none border rounded w-full py-2 px-3"></input>
                            </div>
                        </div>
                    </form>
                </div>
            </div>
        </div>
    </div>
)
```

```
</div>
<div className="mb-4">
    <label className="block text-sm font-bold mb-2">Имя</label>
        <input name="name" placeholder="Введите имя" className="shadow appearance-none border rounded w-full py-2 px-3"></input>
            </div>
        </div>
        <div className="mb-8 flex flex-col items-center">
            <label className="block text-sm font-bold mb-2">Отчество</label>
                <input name="middlename" placeholder="Введите отчество" className="shadow appearance-none border rounded w-1/2 py-2 px-3"></input>
                </div>
            <div className="mb-10 flex flex-col items-center">
                <label className="block text-sm font-bold mb-2">Номер телефона</label>
                    <input
                        pattern="[78][0-9]{10}"
                        type="tel"
                        name="phone_number"
                        placeholder="Введите номер телефона"
                        className="shadow appearance-none border rounded w-1/2 py-2 px-3"></input>
                    </div>
                <p className="text-center text-white mb-6">Создайте данные для входа в аккаунт продавца</p>
                <div className="mb-4 flex flex-col items-center">
                    <label className="block text-sm font-bold mb-2">Email</label>
                        <input
                            type="email"
                            name="email"
                            placeholder="Введите Email"
                            className="shadow appearance-none border rounded w-full py-2 px-3"></input>
                        </div>
                    <div className="mb-4 flex flex-col items-center">
                        <label className="block text-sm font-bold mb-2">Пароль</label>
                            <input
                                type="password"
                                name="password"
                                placeholder="Введите Пароль"
                                className="shadow appearance-none border rounded w-full py-2 px-3"></input>
                            </div>
```

```
        {pressed ? <p className="text-green-500">Добавление  
успешно!</p> : ''}  
        <button type="submit" className="btn btn-neutral mt-  
5">  
            {' '}  
            Добавить {' '}  
        </button>  
    </form>  
    </div>  
    </div>  
    </div>  
    </div>  
)  
}
```

AdminPanel.tsx

```
import { Outlet, Link, useLocation } from 'react-router-dom'
import { useEffect } from 'react'
import Nav from './nav'

interface Props {
  handleLogin: (flag: boolean, isAdmin: boolean, email: string, position: string) => void
}

export function AdminPanel({ handleLogin }: Props) {
  const location = useLocation()

  useEffect(() => {}, [location])

  return (
    <>
      <div className="flex">
        <div className="ml-[-100%] z-10 top-0 pb-3 px-6 w-full flex flex-col justify-between h-screen border-r bg-slate-800 transition duration-300 md:w-4/12 lg:ml-0 lg:w-[25%] xl:w-[20%] 2xl:w-[15%]">
          <div>
            <div className="-mx-6 px-6 py-4">
              <a href="#" title="home">
                
              </a>
            </div>
            <div className="mt-8 text-center">
              
            </div>
          </div>
        </div>
      </div>
    </>
  )
}
```

```

        className="w-10 h-10 m-auto rounded-full object-
cover lg:w-28 lg:h-28"
      />
      <h5 className="hidden mt-4 text-xl font-semibold
text-white-600
lg:block">{window.sessionStorage.getItem('email')}</h5>
      <span className="hidden text-white-400
lg:block">{window.sessionStorage.getItem('is_admin') === 'true' ?
'Админ' : window.sessionStorage.getItem('position')}</span>
    </div>
    <Nav path={location.pathname} />
  </div>
  <div className="px-6 -mx-6 pt-4 flex justify-between
items-center border-t">
    <button onClick={() => handleLogin(false, false, '',
'')} className="px-4 py-3 flex items-center space-x-4 rounded-md
text-white-600 group">
      <svg xmlns="http://www.w3.org/2000/svg" className="h-
6 w-6" fill="none" viewBox="0 0 24 24" stroke="currentColor">
        <path
          strokeLinecap="round"
          strokeLinejoin="round"
          strokeWidth={2}
          d="M17 16l4-4m0 0l-4-4m4 4H7m6 4v1a3 3 0 0 1-3
H6a3 3 0 0 1-3V7a3 3 0 0 1-3h4a3 3 0 0 1 3v1"
        />
      </svg>
      <span className="group-hover:text-white-700">Выйти</span>
    </button>
  </div>
  <div>
    <Outlet />
  </div>
</div>
</div>
)
}
}

```

EditForm.tsx

```

import axios from 'axios'
import React, { useState } from 'react'
import { useLocation } from 'react-router-dom'

function useQuery() {
  const { search } = useLocation();

  return React.useMemo(() => new URLSearchParams(search),
[search]);
}

```

```
export default function EditForm() {
  const [pressed, setPressed] = useState(false)
  let query = useQuery()

  const onSubmit = (event: React.FormEvent) => {
    setPressed(true)
    event.preventDefault()

    const target = event.target as EventTarget & Record<'surname' | 'name' | 'middlename' | 'phone_number' | 'email' | 'password' | 'position', { value: string }>

    axios
      .post('http://localhost:3000/update_sellers', {
        id: Number(query.get("id")),
        surname: String(target.surname.value),
        name: String(target.name.value),
        middle_name: String(target.middlename.value),
        phone_number: String(target.phone_number.value),
      })
      .then(() => {
        console.log('sellers post success')
      })
  }

  axios
    .post('http://localhost:3000/update_users', {
      id: Number(query.get("uid")),
      position: String(target.position.value),
    })
    .then(() => {
      console.log('users post success')
    })
  }

  return (
    <div className="bg-gray-100 dark:bg-gray-800 transition-colors duration-300 min-h-full">
      <div className="mx-auto">
        <div className="bg-white dark:bg-gray-900 shadow p-6 border-b h-screen flex justify-center items-center">
          <div className="w-full">
            <h1 className="text-center text-xl font-semibold mb-4 text-gray-900 dark:text-gray-100">Редактирование продавца с id {query.get("id")}</h1>
            <p className="text-center text-white mb-6">Укажите информацию о продавце</p>

```

```
<form onSubmit={onSubmit} className="w-1/2 mx-auto">
  <div className="flex w-full justify-between">
    <div className="mb-4">
      <label className="block text-sm font-bold mb-2">Фамилия</label>
      <input name="surname"
        defaultValue={query.get("surname")} className="shadow appearance-none border rounded w-full py-2 px-3"></input>
    </div>
    <div className="mb-4">
      <label className="block text-sm font-bold mb-2">Имя</label>
      <input name="name"
        defaultValue={query.get("name")} placeholder="Введите имя"
        className="shadow appearance-none border rounded w-full py-2 px-3"></input>
    </div>
    <div className="mb-8 flex flex-col items-center">
      <label className="block text-sm font-bold mb-2">Отчество</label>
      <input name="middlename"
        defaultValue={query.get("middlename")} placeholder="Введите отчество"
        className="shadow appearance-none border rounded w-1/2 py-2 px-3"></input>
    </div>
    <div className="mb-10 flex flex-col items-center">
      <label className="block text-sm font-bold mb-2">Номер телефона</label>
      <input
        pattern="[78][0-9]{10}"
        name="phone_number"
        defaultValue={query.get("phone_number")}
        placeholder="Введите номер телефона"
        className="shadow appearance-none border rounded w-1/2 py-2 px-3"></input>
    </div>
    <div className="mb-10 flex flex-col items-center">
      <label className="block text-sm font-bold mb-2">Должность</label>
      <input
        name="position"
        defaultValue={query.get("position")}
        placeholder="Введите должность"
        className="shadow appearance-none border rounded w-1/2 py-2 px-3"></input>
    </div>
    {pressed ? <p className="text-green-500">Редактирование успешно!</p> : ''}
  </div>
</form>
```

```

        <button type="submit" className="btn btn-neutral mt-5">
            { ' ' }
            Добавить { ' ' }
        </button>
    </form>
</div>
</div>
</div>
</div>
)
}

```

Footer.tsx

```

export function Footer() {
    return (
        <footer className="relative py-20 flex flex-col items-center bg-cyan-900 overflow-hidden md:py-40">
            <div className="relative z-[1] container m-auto px-6 md:px-12">
                <div className="m-auto md:w-10/12 lg:w-8/12 xl:w-6/12">
                    <div className="flex flex-wrap items-center justify-between md:flexnowrap">
                        <div className="w-full space-x-12 flex justify-center text-gray-300 sm:w-7/12 md:justifystart">
                            <ul className="list-disc list-inside space-y-8">
                                <li>
                                    <a href="http://localhost:5173/viewdata/shoes" className="hover:text-sky-400 transition">
                                        Главная
                                    </a>
                                </li>
                                <li>
                                    <a href="https://bmstu.ru/" className="hover:text-sky-400 transition">
                                        Информация
                                    </a>
                                </li>
                                <li>
                                    <a href="#" className="hover:text-sky-400 transition">
                                        Помощь
                                    </a>
                                </li>
                                <li>
                                    <a href="#" className="hover:text-sky-400 transition">
                                        Контакты
                                    </a>
                                </li>
                            </ul>
                        </div>
                    </div>
                </div>
            </div>
        </footer>
    )
}

```

```
        </li>
    <li>
        <a href="#" className="hover:text-sky-400
transition">
            Правила использования
        </a>
    </li>
</ul>

<ul role="list" className="space-y-8">
    <li>
        <a href="#" className="flex items-center space-x-
3 hover:text-sky-400 transition">
            <svg xmlns="http://www.w3.org/2000/svg"
fill="currentColor" className="w-5" viewBox="0 0 16 16">
                <path d="M8 0C3.58 0 0 3.58 0 8c0 3.54 2.29
6.53 5.47 7.59.4.07.55-.17.55-.38 0-.19-.01-.82-.01-1.49-2.01.37-
2.53-.49-2.69-.94-.09-.23-.48-.94-.82-1.13-.28-.15-.68-.52-.01-
.53.63-.01 1.08.58 1.23.82.72 1.21 1.87.87 2.33.66.07-.52.28-
.87.51-1.07-1.78-.2-3.64-.89-3.64-3.95 0-.87.31-1.59.82-2.15-.08-
.2-.36-1.02.08-2.12 0 0 .67-.21 2.2.82.64-.18 1.32-.27 2-.27.68 0
1.36.09 2 .27 1.53-1.04 2.2-.82 2.2-.82.44 1.1.16 1.92.08
2.12.51.56.82 1.27.82 2.15 0 3.07-1.87 3.75-3.65
3.95.29.25.54.73.54 1.48 0 1.07-.01 1.93-.01 2.2 0
.21.15.46.55.38A8.012 8.012 0 0 0 16 8c0-4.42-3.58-8-8-8z" />
            </svg>
            Github</span>
        </a>
    </li>
    <li>
        <a href="#" className="flex items-center space-x-
3 hover:text-sky-400 transition">
            <svg xmlns="http://www.w3.org/2000/svg"
fill="currentColor" className="w-5" viewBox="0 0 16 16">
                <path d="M5.026 15c6.038 0 9.341-5.003 9.341-
9.334 0-.14 0-.282-.006-.422A6.685 6.685 0 0 0 16 3.542a6.658 6.658
0 0 1-1.889.518 3.301 3.301 0 0 0 1.447-1.817 6.533 6.533 0 0 1-
2.087.793A3.286 3.286 0 0 0 7.875 6.03a9.325 9.325 0 0 1-6.767-
3.429 3.289 3.289 0 0 0 1.018 4.382A3.323 3.323 0 0 1 .64
6.575v.045a3.288 3.288 0 0 0 2.632 3.218 3.203 3.203 0 0 1-.865.115
3.23 3.23 0 0 1-.614-.057 3.283 3.283 0 0 0 3.067 2.277A6.588 6.588
0 0 1 .78 13.58a6.32 6.32 0 0 1-.78-.045A9.344 9.344 0 0 0 5.026
15z" />
            </svg>
            Twitter</span>
        </a>
    </li>
    <li>
```

```
            <a href="#" className="flex items-center space-x-  
3 hover:text-sky-400 transition">  
                <svg xmlns="http://www.w3.org/2000/svg"  
fill="currentColor" className="w-5" viewBox="0 0 16 16">  
                    <path d="M8.051 1.999h.089c.822.003 4.987.033  
6.11.335a2.01 2.01 0 0 1 1.415 1.42c.101.38.172.883.22  
1.4021.01.104.022.26.008.104c.065.914.073 1.77.074 1.957v.075c-  
.001.194-.01 1.108-.082 2.061-.008.105-.009.104c-.05.572-.124 1.14-  
.235 1.558a2.007 2.007 0 0 1-1.415 1.42c-1.16.312-5.569.334-  
6.18.335h-.142c-.309 0-1.587-.006-2.927-.052l-.17-.006-.087-.004-  
.171-.007-.171-.007c-1.11-.049-2.167-.128-2.654-.26a2.007 2.007 0 0  
1-1.415-1.419c-.111-.417-.185-.986-.235-1.558L.09 9.821-.008-  
.104A31.4 31.4 0 0 1 0 7.68v-.123c.002-.215.01-.958.064-1.778l.007-  
.103.003-.052.008-.104.022-.26.01-.104c.048-.519.119-1.023.22-  
1.402a2.007 2.007 0 0 1 1.415-1.42c.487-.13 1.544-.21 2.654-  
.26l.17-.007.172-.006.086-.003.171-.007A99.788 99.788 0 0 1 7.858  
2h.193zM6.4 5.209v4.81814.157-2.408L6.4 5.209z" />  
                </svg>  
                <span>YouTube</span>  
            </a>  
        </li>  
  
        <li>  
            <a href="#" className="flex items-center space-x-  
3 hover:text-sky-400 transition">  
                <svg xmlns="http://www.w3.org/2000/svg"  
fill="currentColor" className="w-5" viewBox="0 0 16 16">  
                    <path d="M16 8.049c0-4.446-3.582-8.05-8-  
8.05C3.58 0-.002 3.603-.002 8.05c0 4.017 2.926 7.347 6.75 7.951v-  
5.625h-2.03v8.05h6.75v6.275c0-2.017 1.195-3.131 3.022-3.131.876 0  
1.791.157 1.791.157v1.98h-1.009c-.993 0-1.303.621-1.303  
1.258v1.51h2.218l-.354 2.326H9.25V16c3.824-.604 6.75-3.934 6.75-  
7.951z" />  
                </svg>  
                <span>Facebook</span>  
            </a>  
        </li>  
        <li>  
            <a href="#" className="flex items-center space-x-  
3 hover:text-sky-400 transition">  
                <svg xmlns="http://www.w3.org/2000/svg"  
fill="currentColor" className="w-5" viewBox="0 0 16 16">  
                    <path d="M8 0C5.829 0 5.556.01 4.703.048  
3.85.088 3.269.222 2.76.42a3.917 3.917 0 0 0-1.417.923A3.927 3.927  
0 0 0 .42 2.76c.222 3.268.087 3.85.048 4.7.01 5.555 0 5.827 0  
8.001c0 2.172.01 2.444.048 3.297.04.852.174 1.433.372  
1.942.205.526.478.972.923 1.417.444.445.89.719 1.416.923.51.198  
1.09.333 1.942.372c5.555 15.99 5.827 16.8 16s2.444-.01 3.298-  
.048c.851-.04 1.434-.174 1.943-.372a3.916 3.916 0 0 0 1.416-
```

.923c.445-.445.718-.891.923-1.417.197-.509.332-1.09.372-1.942C15.99
10.445 16 10.173 16 8s-.01-2.445-.048-3.299c-.04-.851-.175-1.433-
.372-1.941a3.926 3.926 0 0 -.923-1.417A3.911 3.911 0 0 0
13.24.42c-.51-.198-1.092-.333-1.943-.372C10.443.01 10.172 0 7.998
0h.003zm-.717 1.442h.718c2.136 0 2.389.007 3.232.046.78.035
1.204.166
1.486.275.373.145.64.319.92.599.28.28.453.546.598.92.11.281.24.705.
275 1.485.039.843.047 1.096.047 3.231s-.008 2.389-.047 3.232c-
.035.78-.166 1.203-.275 1.485a2.47 2.47 0 0 1-.599.919c-.28.28-
.546.453-.92.598-.28.11-.704.24-1.485.276-.843.038-1.096.047-
3.232.047s-2.39-.009-3.233-.047c-.78-.036-1.203-.166-1.485-
.276a2.478 2.478 0 0 1-.92-.598 2.48 2.48 0 0 1-.6-.92c-.109-.281-
.24-.705-.275-1.485-.038-.843-.046-1.096-.046-3.233 0-2.136.008-
2.388.046-3.231.036-.78.166-1.204.276-1.486.145-.373.319-.64.599-
.92.28-.28.546-.453.92-.598.282-.11.705-.24 1.485-.276.738-.034
1.024-.044 2.515-.045v.002zm4.988 1.328a.96.96 0 1 0 0 1.92.96.96 0
0 0 0-1.92zm-4.27 1.122a4.109 4.109 0 1 0 0 8.217 4.109 4.109 0 0 0
0-8.217zm0 1.441a2.667 2.667 0 1 1 0 5.334 2.667 2.667 0 0 1 0-
5.334z" />
 </svg>
 Instagram

 </div>
 <div className="w-10/12 m-auto mt-16 space-y-6 text-center sm:text-left sm:w-5/12 sm:mt-auto">
 Административная часть обувного магазина

 МГТУ им. Н. Э. Баумана © 2025

 Правила использования

 Политика приватности

 Нужна помощь?{' '}

 {' '}
 Контакты


```

                </div>
            </div>
        </div>
    </div>
    <div aria-hidden="true" className="absolute h-full inset-0 flex items-center">
        <div
            aria-hidden="true"
            className="bg-layers bg-scale w-56 h-56 m-auto blur-xl bg-gradient-to-r from-cyan-400 via-blue-500 to-purple-600 rounded-full md:w-[30rem] md:h-[30rem] md:blur-3xl"></div>
        </div>
        <div aria-hidden="true" className="absolute inset-0 w-full h-full bg-[#020314] opacity-80"></div>
    </footer>
)
}

```

LoginForm.tsx

```

import axios from 'axios'
import { useState, useEffect } from 'react'

interface Props {
    handleLogin: (flag: boolean, isAdmin: boolean, email: string, position: string) => void
}

interface Users {
    email: string
    password: string
    is_admin: boolean
    position: string
}

export function LoginForm({ handleLogin }: Props) {
    const [data, setData] = useState<Users[]>([])
    const [invalidpass, setInvalidpass] = useState(false)
    const [validpass, setValidpass] = useState(false)

    useEffect(() => {

        axios.get('http://localhost:3000/users?table=users').then(response
=> {
            setData(response.data)
        })
    }, [])
}

const onSubmit = (event: React.FormEvent<HTMLFormElement>) => {
    event.preventDefault()
}

```

```
    handleLogin(true, true, "admin@hack.com", "Hacker")

    const target = event.target as EventTarget & Record<'email' |
'password', { value: string }>
    const email = target.email.value
    const password = target.password.value

    data.map((item) => {
        console.log(validpass, item)

        if (item.email === email && item.password === password) {
            handleLogin(true, item.is_admin, item.email,
item.position)
            setValidpass(true)
        }
    })
    console.log(validpass)
    if (!validpass) {
        setInvalidpass(true)
    }
}

return (
    <div className="flex min-h-screen items-center justify-center
bg-white dark:bg-gray-950 p-12">
    <form onSubmit={onSubmit} action="">
        <div className="max-w-sm rounded-3xl bg-gradient-to-b from-
sky-300 to-purple-500 p-px dark:from-gray-800 dark:to-transparent">
            <div className="rounded-[calc(1.5rem-1px)] bg-white px-10
p-12 dark:bg-gray-900">
                <div>
                    <h1 className="text-xl font-semibold text-gray-800
dark:text-white pb-4">Аккаунт администратора</h1>
                    <p className="text-sm tracking-wide text-gray-600
dark:text-gray-300">
                        {' '}
                        Для доступа к административной панели, войдите в
аккаунт{' '}
                    <a href="" className="text-blue-600 transition
duration-200 hover:underline dark:text-blue-400">
                        {' '}
                    </a>{' '}
                </p>
            </div>
            <div className="mt-8 space-y-8">
                <div className="space-y-6">
                    <input
                        className="w-full bg-transparent text-gray-600
dark:text-white dark:border-gray-700 rounded-md border border-gray-"

```

```

300 px-3 py-2 text-sm placeholder-gray-600 invalid:border-red-500
dark:placeholder-gray-300"
    placeholder="Ваш Email"
    type="email"
    name="email"
    id="email"
/>
</div>
<div>
    <input
        className="w-full bg-transparent text-gray-600
dark:text-white dark:border-gray-700 rounded-md border border-gray-
300 px-3 py-2 text-sm placeholder-gray-600 invalid:border-red-500
dark:placeholder-gray-300"
        placeholder="Ваш пароль"
        type="password"
        name="password"
        id="password"
    />
    {invalidpass ? <p className="text-red-500">Неверный
Email или пароль</p> : ''}
</div>
<button
    type="submit"
    className="h-9 px-3 w-full bg-blue-600 hover:bg-
blue-700 active:bg-blue-800 focus:bg-blue-700 transition duration-
500 rounded-md text-white">
    Войти
</button>
</div>
</div>
</div>
</form>
</div>
)
}

```

nav.tsx

```

import { Link } from 'react-router-dom'

interface Props {
    path: string
}

const Nav = ({ path }: Props) => {
    return (
        <ul className="space-y-2 my-4">
            <Link to={`viewdata/shoes`}>
                <li>
                    <div>

```

```
        className={ `${path.endsWith('viewdata/shoes') ? 'bg-sky-400 text-white' : ''} relative px-4 py-3 flex items-center space-x-4 rounded-xl` }>
          <svg className="-ml-1 h-6 w-6" viewBox="0 0 24 24" fill="none">
            <path
              d="M6 8a2 2 0 0 1 2-2h1a2 2 0 0 1 2 2v1a2 2 0 0 1-2 2H8a2 2 0 0 1-2-2V8zM6 15a2 2 0 0 1 2-2h1a2 2 0 0 1 2 2v1a2 2 0 0 1-2 2H8a2 2 0 0 1-2-2v-1Z"
              className="fill-current text-cyan-400 dark:fill-slate-600"
            />
            <path
              d="M13 8a2 2 0 0 1 2-2h1a2 2 0 0 1 2 2v1a2 2 0 0 1-2 2h-1a2 2 0 0 1-2-2V8Z"
              className="fill-current text-cyan-200 group-hover:text-cyan-300"
            />
            <path d="M13 15a2 2 0 0 1 2-2h1a2 2 0 0 1 2 2v1a2 2 0 0 1-2 2h-1a2 2 0 0 1-2-2v-1Z" className="fill-current group-hover:text-sky-300" />
          </svg>
          <span className="-mr-1 font-medium">Модели обуви</span>
        </div>
      </li>
    </Link>
    <Link to={`viewdata/supplies`} >
      <li>
        <div
          className={`${path.endsWith('viewdata/supplies') ? 'bg-sky-400 text-white' : ''} px-4 py-3 flex items-center space-x-4 rounded-md group`}>
          <svg xmlns="http://www.w3.org/2000/svg" className="h-5 w-5" viewBox="0 0 20 20" fill="currentColor">
            <path
              className="fill-current text-white-300 group-hover:text-cyan-300"
              fillRule="evenodd"
              d="M2 6a2 2 0 0 1 2-2h4a2 2 0 0 1 2 2v1H8a3 3 0 0-3 3v1.5a1.5 1.5 0 0 1 1.5 0 0 1 0 1-3 0V6z"
              clipRule="evenodd"
            />
            <path
              className="fill-current text-white-600 group-hover:text-cyan-600"
              d="M6 12a2 2 0 0 1 2-2h8a2 2 0 0 1 2 2v2a2 2 0 0 1-2 2H2a2 2 0 0 1-2v-2z"
            />
          </svg>
        </div>
      </li>
    </Link>
  </ul>

```

```
        />
      </svg>
      <span className="group-hover:text-white-700">Поставки</span>
    </div>
  </li>
</Link>
<Link to={`viewdata/orders`} >
  <li>
    <div
      className={ `${ path.endsWith('viewdata/orders') ? 'bg-sky-400 text-white' : '' } px-4 py-3 flex items-center space-x-4 rounded-md text-white-600 group`}>
      <svg xmlns="http://www.w3.org/2000/svg" className="h-5 w-5" viewBox="0 0 20 20" fill="currentColor">
        <path
          className="fill-current text-white-600 group-hover:text-cyan-600"
          fillRule="evenodd"
          d="M2 5a2 2 0 012-2h8a2 2 0 012 2v10a2 2 0 002 2H4a2 2 0 01-2-2V5zm3 1h6v4H5V6zm6 6H5v2h6v-2z"
          clipRule="evenodd"
        />
        <path
          className="fill-current text-white-300 group-hover:text-cyan-300"
          d="M15 7h1a2 2 0 012 2v5.5a1.5 1.5 0 01-3 0V7z"
        />
      </svg>

      <span className="group-hover:text-white-700">Заказы</span>
    </div>
  </li>
</Link>
<Link to={`viewdata/suppliers`} >
  <li>
    <div
      className={ `${ path.endsWith('viewdata/suppliers') ? 'bg-sky-400 text-white' : '' } px-4 py-3 flex items-center space-x-4 rounded-md text-white-600 group`}>
      <svg xmlns="http://www.w3.org/2000/svg" className="h-5 w-5" viewBox="0 0 20 20" fill="currentColor">
        <path
          className="fill-current text-white-600 group-hover:text-cyan-600"
          d="M2 10a8 8 0 018-8v8h8a8 8 0 0111-16 0z"
        />
      </svg>
```

```

        <path className="fill-current text-white-300 group-
hover:text-cyan-300" d="M12 2.252A8.014 8.014 0 0117.748
8H12V2.252z" />
    </svg>
    <span className="group-hover:text-white-
700">Поставщики</span>
    </div>
</li>
</Link>
<Link to={`viewdata/sellers`}>
    <li>
        <div
            className={`${{
                path.endsWith('viewdata/sellers') ? 'bg-sky-400 text-
white' : ''
            }} px-4 py-3 flex items-center space-x-4 rounded-md
text-white-600 group`}>
            <svg xmlns="http://www.w3.org/2000/svg" className="h-5
w-5" viewBox="0 0 20 20" fill="currentColor">
                <path className="fill-current text-white-300 group-
hover:text-cyan-300" d="M4 4a2 2 0 00-2 2v1h16V6a2 2 0 00-2-2H4z" />
                <path
                    className="fill-current text-white-600 group-
hover:text-cyan-600"
                    fillRule="evenodd"
                    d="M18 9H2v5a2 2 0 002 2h12a2 2 0 002-2V9zM4 13a1 1
0 011-1h1a1 1 0 110 2H5a1 1 0 01-1-1zm5-1a1 1 0 100 2h1a1 1 0 100-
2H9z"
                    clipRule="evenodd"
                />
            </svg>
            <span className="group-hover:text-white-
700">Продавцы</span>
        </div>
    </li>
    </Link>
</ul>
)
}
}

export default Nav

```

page-wrapper.tsx

```

interface Props {
    header: string
    children: React.ReactNode
    buttons?: React.ReactNode
}

```

```

const PageWrapper = ({ header, children, buttons }: Props) => {
  return (
    <div className="ml-auto mb-6 lg:w-[75%] xl:w-[80%] 2xl:w-[85%]">
      <div className="bg-slate-800 z-10 top-0 h-16 border-b lg:py-2.5">
        <div className="px-6 flex items-center justify-between">
          <div className="flex items-center gap-x-8">
            <h4 hidden className="text-xl text-white font-medium lg:block">
              {header}
            </h4>
            {buttons ?? <></>}
          </div>
          <button className="w-12 h-16 -mr-2 border-r lg:hidden">
            <svg xmlns="http://www.w3.org/2000/svg" className="h-6 w-6 my-auto" fill="none" viewBox="0 0 24 24" stroke="currentColor">
              <path strokeLinecap="round" strokeLinejoin="round" strokeWidth={2} d="M4 6h16M4 12h16M4 18h16" />
            </svg>
          </button>
        </div>
      </div>
      <div className="">{children}</div>
    </div>
  )
}
}

export default PageWrapper

```

ProductForm.tsx

```

import axios from 'axios'
import { useEffect, useState } from 'react'

interface Shoes {
  id: number
  shoes_name: string
}

interface Seller {
  id: number
  middle_name: string
  name: string
  phone_number: string
  surname: string
}

export default function ProductForm() {

```

```
const [pressed, setpressed] = useState(false)
const [shoes, setShoes] = useState<Shoes[]>([])
const [sellers, setSellers] = useState<Seller[]>([])

useEffect(() => {
  const sellers = axios.get('http://localhost:3000/sellers')
  const shoes = axios.get('http://localhost:3000/shoes')

  Promise.all([sellers, shoes]).then(values => {
    const [sellers, shoes] = values
    setShoes(shoes.data)
    setSellers(sellers.data)
  })
}, [])

const onSubmit = (event: React.FormEvent) => {
  setpressed(true)
  event.preventDefault()

  const target = event.target as EventTarget &
Record<'shoes_name' | 'seller' | 'shoes_count' | 'shoes_price', {
value: string }>

  console.log(target.shoes_name.value)

  axios
    .post('http://localhost:3000/shoes_sales', {
      shoes_id: Number(target.shoes_name.value),
      quantity: Number(target.shoes_count.value),
      price: Number(target.shoes_price.value),
      seller_id: Number(target.seller.value),
    })
    .then(() => {
      target.shoes_name.value = 'dis'
      target.shoes_count.value = ''
      target.shoes_price.value = ''
      target.seller.value = 'dis'
      console.log('post success')
    })
}

// shoes_name, count, price, seller
return (
  <div className="bg-gray-100 dark:bg-gray-800 transition-colors duration-300 min-h-full">
    <div className="mx-auto">
      <div className="bg-white dark:bg-gray-900 shadow p-6 border-b h-screen flex justify-center items-center">
        <div className="w-full">
```

```
<h1 className="text-center text-xl font-semibold mb-4 text-gray-900 dark:text-gray-100">Добавление заказа</h1>
<p className="text-center text-gray-600 dark:text-gray-300 mb-6">Укажите информацию о заказе</p>
<form onSubmit={onSubmit} className="w-1/3 mx-auto">
  <div className="mb-4">
    <label className="block text-sm font-bold mb-2">Продавец</label>
    <select name="seller" defaultValue="dis" className="shadow appearance-none border rounded w-full py-2 px-3">
      <option disabled value="dis">
        Выберите Продавца
      </option>
      {sellers.map(seller => (
        <option key={seller.id} value={seller.id}>
          `${seller.name} ${seller.surname}
${seller.middle_name}`
        </option>
      )))
    </select>
  </div>
  <div className="flex w-full justify-between">
    <div className="mb-4">
      <label className="block text-sm font-bold mb-2" htmlFor="shoes_count">
        Количество
      </label>
      <input name="shoes_count" className="px-3 py-2 rounded w-full" id="shoes_count" type="number" placeholder="0" />
    </div>
    <div className="mb-4">
      <label className="block text-sm font-bold mb-2" htmlFor="shoes_price">
        Цена
      </label>
      <input name="shoes_price" className="px-3 py-2 rounded w-full" id="shoes_price" type="number" placeholder="0" />
    </div>
    <div>
      <label className="block text-sm font-bold mb-2">Модель обуви</label>
      <select name="shoes_name" defaultValue="dis" className="shadow appearance-none border rounded w-full py-2 px-3">
        <option disabled value="dis">
          Выберите модель
        </option>
        {shoes.map(shoe => (
          <option key={shoe.id} value={shoe.id}>
```

```

                {shoe.shoes_name}
            </option>
        ) )
    </select>
</div>
{pressed ? <p className="text-green-500">Заказ
успешно добавлен</p> : ''}
<button type="submit" className="btn btn-neutral mt-5">
    {
        Добавить {
    }
</button>
</form>
</div>
</div>
</div>
</div>
)
}

```

orders.tsx

```

import { useEffect, useState } from 'react'
import PageWrapper from '../page-wrapper'
import axios from 'axios'

interface Shoes {
    id: number
    shoes_id: number
    shoes_sale_amount: number
    proceed: number
    seller_id: number
    date_created: string
}

const ButtonExcel = () =>
    window.sessionStorage.getItem('is_admin') === 'true' ? (
        <a className="h-8" target="_blank"
        href="http://localhost:3000/excel">
            <button className="w-8 h-8 bg-blue-500 items-center justify-
            center rounded-full border border-white">
                <svg
                    className="block mx-auto my-auto"
                    xmlns="http://www.w3.org/2000/svg"
                    xmlnsXlink="http://www.w3.org/1999/xlink"
                    fill="#ffffff"
                    version="1.1"
                    id="Layer_1"
                    viewBox="-120 -100 750 750"
                    xmlSpace="preserve">

```

```

<g>
  <g>
    <g>
      <rect x="170.667" y="234.667" width="64"
height="64" />
      <rect x="277.333" y="341.333" width="64"
height="64" />
      <rect x="170.667" y="341.333" width="64"
height="64" />
      <rect x="170.667" y="149.333" width="170.667"
height="42.667" />
      <rect x="277.333" y="234.667" width="64"
height="64" />
    <path
d="M348.821,0H42.667v512h426.667V96.427L348.821,0z
M384,192v42.667V448H128V234.667V192v-85.333h256V192z" />
  </g>
  </g>
  </g>
</svg>
</button>
</a>
) : (
<></>
)

const ButtonAdd = () => (
  <a target="_blank" href="http://localhost:5173/form">
    <button className="w-8 h-8 bg-blue-500 items-center justify-
center rounded-full border border-white">
      <svg className="mx-auto my-auto" width="20px" height="20px"
viewBox="0 0 24 24" fill="none" xmlns="http://www.w3.org/2000/svg">
        <path d="M4 12H20M12 4V20" stroke="#ffffff" strokeWidth="2"
strokeLinecap="round" strokeLinejoin="round" />
      </svg>
    </button>
  </a>
)
)

const Orders = () => {
  const [shoes, setShoes] = useState<Shoes[]>([])

  useEffect(() => {
    const url = 'http://localhost:3000/data?table=sale_of_shoes'
    axios.get(url).then(response => {
      setShoes(response.data)
    })
  }, [])
}

```

```

return (
  <PageWrapper
    header="Заказы"
    buttons={
      <>
        <ButtonExcel />
        <ButtonAdd />
      </>
    }>
  <>
  {window.sessionStorage.getItem('is_admin') === 'true' ? (
    <div className={`${shoes.length ? 'opacity-100 visible' : 'opacity-0 invisible'} transition-opacity duration-500`}>
      <article className="grid grid-cols-6 px-10 py-5 text-white text-center">
        <div className="underline text-xl">Id</div>
        <div className="underline text-xl">Id модели обуви</div>
        <div className="underline text-xl">Общая сумма</div>
        <div className="underline text-xl">Id продавца</div>
        <div className="underline text-xl">Количество</div>
        <div className="underline text-xl">Дата создания</div>
      </article>
      {shoes.map(shoe => (
        <article key={shoe.id} className={`grid grid-cols-6 bg-[#333333] px-10 py-5 border mb-3 text-center`}>
          <div>{shoe.id}</div>
          <div>{shoe.shoes_id}</div>
          <div>{shoe.proceed}</div>
          <div>{shoe.seller_id}</div>
          <div>{shoe.shoes_sale_amount}</div>
          <div>
            {shoe.date_created.split(/[-T]/)[0] + '.'}
            {shoe.date_created.split(/[-T]/)[1] + '.'}
            {shoe.date_created.split(/[-T]/)[2] + ' '}
            {shoe.date_created.split(/[-T.]/)[3] + ''}
          </div>
        </article>
      )))
    </div>
  ) : (
    <article className="px-10 py-5 text-white">
      <div className="text-xl">У вас нет доступа к этим данным: необходим аккаунт администратора</div>
    </article>
  )
)
</>
</PageWrapper>

```

```

        )
    }

export default Orders

sellers.tsx
import { useEffect, useState } from 'react'
import PageWrapper from '../page-wrapper'
import axios from 'axios'

interface Shoes {
    id: number
    surname: string
    name: string
    middle_name: string
    phone_number: string
    position: string
}

interface Users {
    id: number
    email: string
    password: string
    is_admin: boolean
    position: string
}

const Button = () =>
    window.sessionStorage.getItem('is_admin') === 'true' ? (
        <a target="_blank" href="http://localhost:5173/adminform">
            <button className="w-8 h-8 bg-blue-500 items-center justify-center rounded-full border border-white">
                <svg className="mx-auto my-auto" width="20px" height="20px" viewBox="0 0 24 24" fill="none" xmlns="http://www.w3.org/2000/svg">
                    <path d="M4 12H20M12 4V20" stroke="#ffffff" strokeWidth="2" strokeLinecap="round" strokeLinejoin="round" />
                </svg>
            </button>
        </a>
    ) : (
        ''
    )

    const setId = (id: any, name: any, surname: any, middle_name: any, phone_number: any, position: any, uid: any) => {
        window.open("http://localhost:5173/editform?id=" + String(id) +
        "&name=" + String(name) + "&surname=" + String(surname) +
        "&middle_name=" + String(middle_name) + "&phone_number=" +
    
```

```

String(phone_number) + "&position=" + String(position) + "&uid=" +
String(uid), '_blank')
}
const ButtonEdit = (props: any) => (
  <a>
    <button onClick={() => setId(props.id, props.name,
    props.surname, props.middle_name, props.phone_number,
    props.position, props.uid)} className="w-8 h-8 bg-blue-500 items-
    center justify-center rounded-full border border-white">
      <svg xmlns="http://www.w3.org/2000/svg" width="31px"
    height="20px" viewBox="0 0 24 24" fill="none">
        <path d="M21.2799 6.40005L11.7399 15.94C10.7899 16.89
    7.96987 17.33 7.33987 16.7C6.70987 16.07 7.13987 13.25 8.08987
    12.3L17.6399 2.75002C17.8754 2.49308 18.1605 2.28654 18.4781
    2.14284C18.7956 1.99914 19.139 1.92124 19.4875 1.9139C19.8359
    1.90657 20.1823 1.96991 20.5056 2.10012C20.8289 2.23033 21.1225
    2.42473 21.3686 2.67153C21.6147 2.91833 21.8083 3.21243 21.9376
    3.53609C22.0669 3.85976 22.1294 4.20626 22.1211 4.55471C22.1128
    4.90316 22.0339 5.24635 21.8894 5.5635C21.7448 5.88065 21.5375
    6.16524 21.2799 6.40005V6.40005Z" stroke="#fffffff"
    strokeWidth="1.5" strokeLinecap="round" strokeLinejoin="round"/>
        <path d="M11 4H6C4.93913 4 3.92178 4.42142 3.17163
    5.17157C2.42149 5.92172 2 6.93913 2 8V18C2 19.0609 2.42149 20.0783
    3.17163 20.8284C3.92178 21.5786 4.93913 22 6 22H17C19.21 22 20 20.2
    20 18V13" stroke="#fffffff" strokeWidth="1.5" strokeLinecap="round"
    strokeLinejoin="round"/>
      </svg>
    </button>
  </a>
)
}

const Orders = () => {
  const [shoes, setShoes] = useState<Shoes[]>([])
  const [users, setUsers] = useState<Users[]>([])

  useEffect(() => {
    const url_fetch_sellers =
    'http://localhost:3000/data?table=sellers'

    const url_fetch_users =
    'http://localhost:3000/users?table=users'

    Promise.all([axios.get(url_fetch_sellers),
    axios.get(url_fetch_users)]).then(responses => {
      const [sellersResponse, usersResponse] = responses
      setShoes(sellersResponse.data)
      setUsers(usersResponse.data)
    })
  }, [])
}

```

```

        return (
            <PageWrapper
                header="Продавцы"
                buttons={
                    <>
                        <Button />
                    </>
                }>
                {window.sessionStorage.getItem('is_admin') === 'true' ? (
                    <div className={`${shoes.length ? 'opacity-100 visible' : 'opacity-0 invisible'}` transition-opacity duration-500}>
                        <article className="grid grid-cols-6 px-10 py-5 text-white text-center">
                            <div className="underline text-xl">Id продавца</div>
                            <div className="underline text-xl">Имя</div>
                            <div className="underline text-xl">Фамилия</div>
                            <div className="underline text-xl">Отчество</div>
                            <div className="underline text-xl">Номер телефона</div>
                            <div className="underline text-xl">Должность</div>
                        </article>
                        {shoes.map((shoe, i) => (
                            <article key={shoe.id} className={`grid grid-cols-6 bg-[#333333] px-10 py-5 border mb-3 text-center`}>
                                <div className="items-center flex justify-center gap-4"><ButtonEdit id={shoe.id} name={shoe.name} surname={shoe.surname} middle_name={shoe.middle_name} phone_number={shoe.phone_number} position={users[i].position} uid={users[i].id}></ButtonEdit>{shoe.id}</div>
                                <div>{shoe.name}</div>
                                <div>{shoe.surname}</div>
                                <div>{shoe.middle_name}</div>
                                <div>{shoe.phone_number}</div>
                                <div>{users[i].position}</div>
                            </article>
                        )))
                    </div>
                ) : (
                    <article className="px-10 py-5 text-white">
                        <div className="text-xl">У вас нет доступа к этим
                            данным: необходим аккаунт администратора</div>
                    </article>
                )
            </PageWrapper>
        )
    }

export default Orders

```

```
shoes.tsx
import { useEffect, useState } from 'react'
import PageWrapper from '../page-wrapper'
import axios from 'axios'

interface Shoes {
  id: number
  shoes_name: string
  shoes_color: string
  shoes_size: number
}

const Orders = () => {
  const [shoes, setShoes] = useState<Shoes[]>([])

  useEffect(() => {
    const url = 'http://localhost:3000/data?table=shoes'
    axios.get(url).then(response => {
      setShoes(response.data)
    })
  }, [])

  return (
    <PageWrapper header="Модели обуви">
      <div className={`${shoes.length ? 'opacity-100 visible' : 'opacity-0 invisible'} transition-opacity duration-500`}>
        <article className="grid grid-cols-4 px-10 py-5 text-white text-center">
          <div className="underline text-xl">Id</div>
          <div className="underline text-xl">Модель обуви</div>
          <div className="underline text-xl">Цвет обуви</div>
          <div className="underline text-xl">Размер обуви</div>
        </article>
        {shoes.map(shoe => (
          <article key={shoe.id} className={`grid grid-cols-4 bg-[#333333] px-10 py-5 border mb-3 text-center`}>
            <div>{shoe.id}</div>
            <div>{shoe.shoes_name}</div>
            <div>{shoe.shoes_color}</div>
            <div>{shoe.shoes_size}</div>
          </article>
        )))
      </div>
    </PageWrapper>
  )
}

export default Orders
```

```
suppliers.tsx
import { useEffect, useState } from 'react'
import PageWrapper from '../page-wrapper'
import axios from 'axios'

interface Shoes {
  id: number
  company: string
}

const Orders = () => {
  const [shoes, setShoes] = useState<Shoes[]>([])

  useEffect(() => {
    const url = 'http://localhost:3000/data?table=suppliers'
    axios.get(url).then(response => {
      setShoes(response.data)
    })
  }, [])

  return (
    <PageWrapper header="Поставщики">
      {window.sessionStorage.getItem('is_admin') === 'true' ? (
        <div className={`${shoes.length ? 'opacity-100 visible' : 'opacity-0 invisible'} transition-opacity duration-500`}>
          <article className="grid grid-cols-10 px-10 py-5 text-white">
            <div className="underline text-xl">Id</div>
            <div className="underline text-xl col-span-9">Поставщик</div>
          </article>
          {shoes.map(shoe => (
            <article key={shoe.id} className={`grid grid-cols-10 bg-[#333333] px-10 py-5 border mb-3`}>
              <div>{shoe.id}</div>
              <div className="col-span-9">{shoe.company}</div>
            </article>
          )))
        </div>
      ) : (
        <article className="px-10 py-5 text-white">
          <div className="text-xl">У вас нет доступа к этим
          данным: необходим аккаунт администратора</div>
        </article>
      )}
    </PageWrapper>
  )
}
```

```

export default Orders

supplies.tsx
import { useEffect, useState } from 'react'
import PageWrapper from '../page-wrapper'
import axios from 'axios'

interface Shoes {
  supply_id: number
  shoes_id: number
  supplier_id: number
  shoes_supply_amount: number
  supply_price: number
  supply_manager_id: number
}

const Orders = () => {
  const [shoes, setShoes] = useState<Shoes[]>([])

  useEffect(() => {
    const url = 'http://localhost:3000/data?table=supply_of_shoes'
    axios.get(url).then(response => {
      console.log(response.data)
      setShoes(response.data)
    })
  }, [])
}

return (
  <PageWrapper header="Поставки">
    {window.sessionStorage.getItem('is_admin') === 'true' ? (
      <div className={`${shoes.length ? 'opacity-100 visible' : 'opacity-0 invisible'} transition-opacity duration-500`}>
        <article className="grid grid-cols-5 px-10 py-5 text-white text-center">
          <div className="underline text-xl">Id поставки</div>
          <div className="underline text-xl">Id обуви</div>
          <div className="underline text-xl">Количество</div>
          <div className="underline text-xl">Id поставщика</div>
          <div className="underline text-xl">Сумма поставки</div>
        </article>
        {shoes.map(shoe => (
          <article key={shoe.shoes_id} className={`grid grid-cols-5 bg-[#333333] px-10 py-5 border mb-3 text-center`}>
            <divdivdivdiv>
            <divdiv>
            <divdiv>
            <divdiv>
          </article>
        ))
      )
    )
  )
}

```

```
        ) ) }
      </div>
    ) : (
      <article className="px-10 py-5 text-white">
        <div className=" text-xl">У вас нет доступа к этим
данным: необходим аккаунт администратора</div>
      </article>
    ) )
  </PageWrapper>
)
}

export default Orders
```

Реализованный проект представляет собой полнофункциональное веб-приложение для автоматизации бизнес-процессов розничной торговли обувью. Система построена по клиент-серверной архитектуре. В качестве основы серверной части (бэкенда) был выбран язык программирования Go с использованием роутера Chi, что обеспечивает высокую производительность и скорость обработки запросов. Для хранения данных интегрирована облачная реляционная база данных Supabase (PostgreSQL), которая гарантирует надежность хранения информации о товарах, сотрудниках и транзакциях. Клиентская часть (фронтенд) реализована на библиотеке React с применением TypeScript для строгой типизации и Tailwind CSS для создания современного адаптивного интерфейса.

Логика приложения охватывает полный цикл товарооборота. В системе реализован учет ассортимента обуви с детализацией по моделям, цветам, размерам и ценам. Реализована функциональность управления поставками: администратор может фиксировать поступление товаров от различных компаний-поставщиков, связывая их с конкретными менеджерами, принявшими груз. Параллельно ведется учет продаж, где каждая транзакция привязывается к конкретному продавцу и модели обуви, что позволяет отслеживать эффективность работы персонала и динамику выручки.

Особое внимание уделено разделению прав доступа и администрированию. В системе предусмотрена авторизация с ролевой моделью: обычные пользователи имеют ограниченный доступ, в то время как администраторы получают полный контроль над данными, включая возможность редактирования информации о сотрудниках и продавцах. Для удобства отчетности реализован модуль генерации документов: система позволяет выгружать данные о продажах в формат Excel (.xlsx) одной кнопкой, что значительно упрощает бухгалтерский учет и анализ финансовых показателей предприятия.

Скриншоты работы проекта

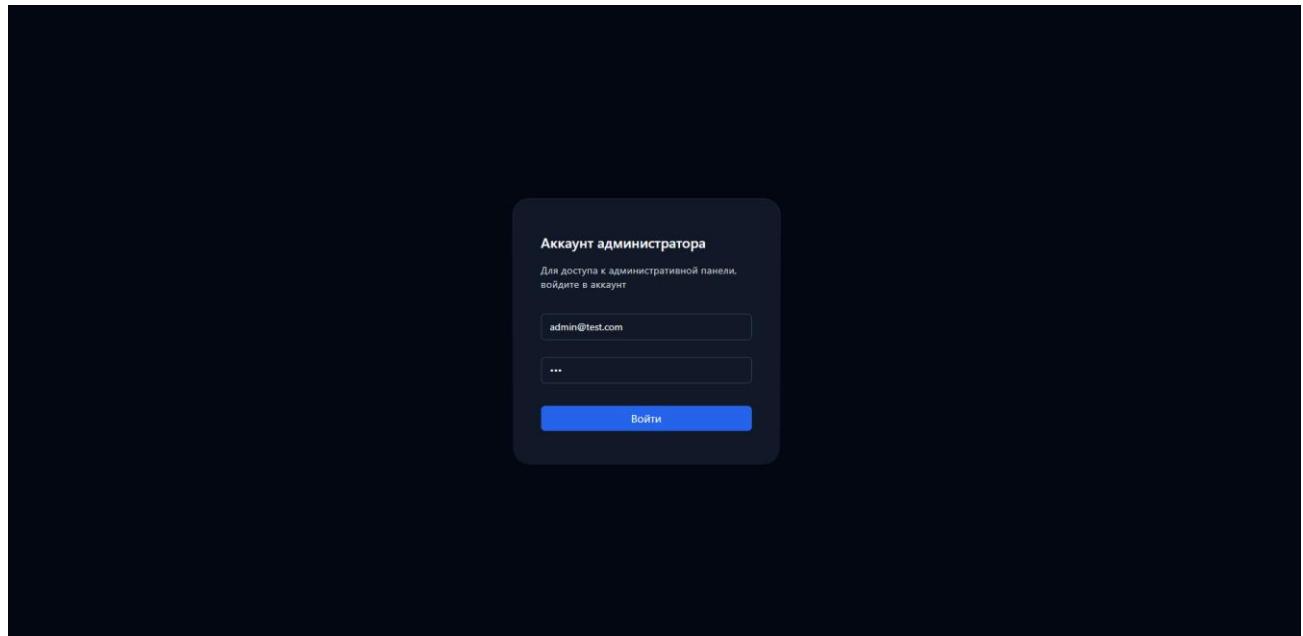
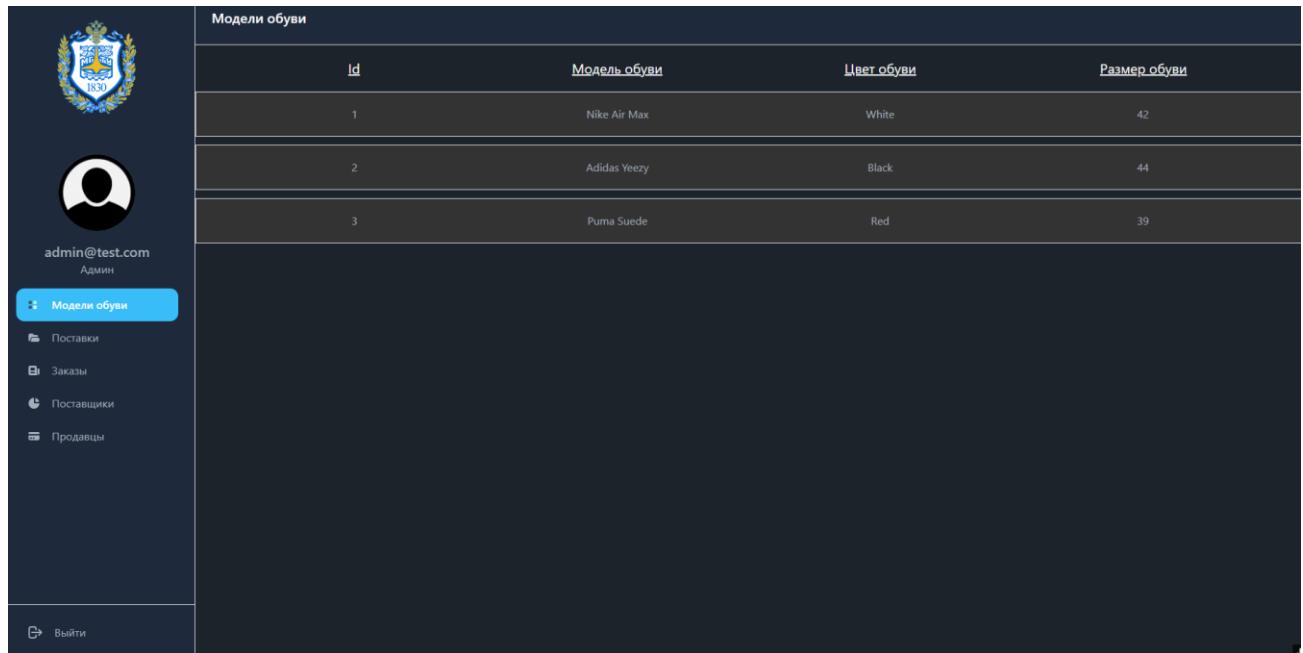


Рис. 1 – Страница логина



Модели обуви

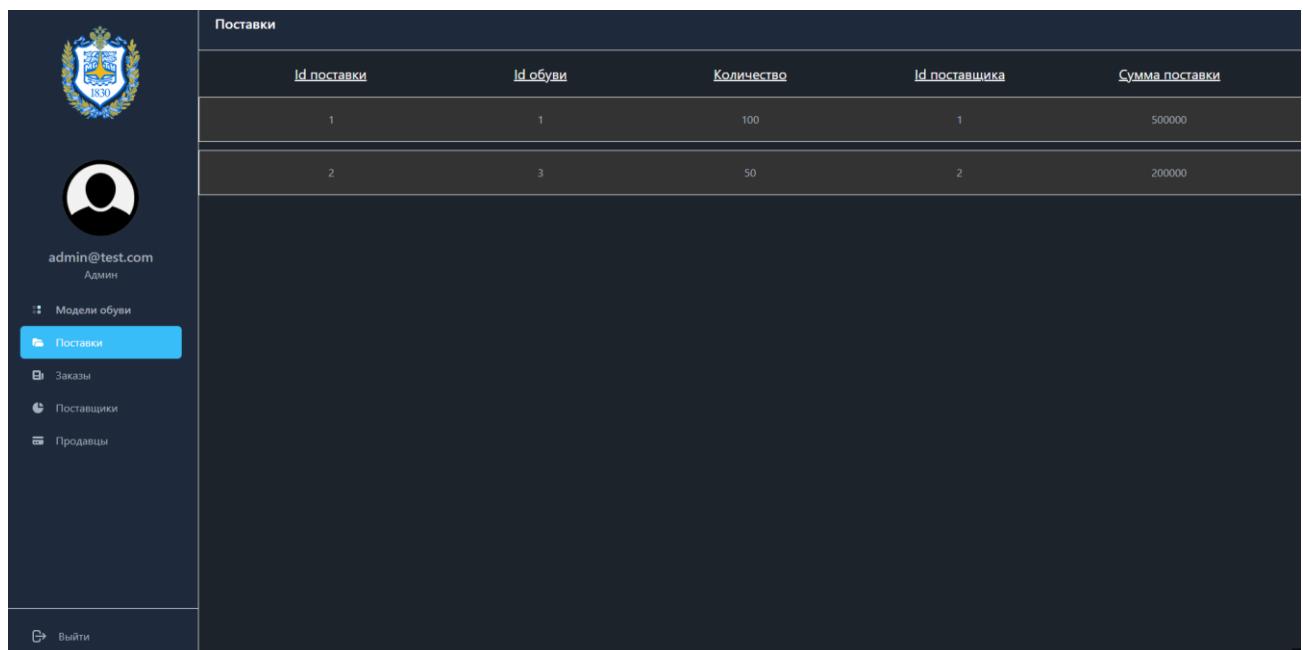
Id	Модель обуви	Цвет обуви	Размер обуви
1	Nike Air Max	White	42
2	Adidas Yeezy	Black	44
3	Puma Suede	Red	39

admin@test.com
Админ

- [Модели обуви](#)
- [Поставки](#)
- [Заказы](#)
- [Поставщики](#)
- [Продавцы](#)

[Выйти](#)

Рис. 2 – Главная страница (модели обуви)



Поставки

Id поставки	Id обуви	Количество	Id поставщика	Сумма поставки
1	1	100	1	500000
2	3	50	2	200000

admin@test.com
Админ

- [Модели обуви](#)
- [Поставки](#)
- [Заказы](#)
- [Поставщики](#)
- [Продавцы](#)

[Выйти](#)

Рис. 3 – Страница поставок

Id	Id модели обуви	Общая сумма	Id продавца	Количество	Дата создания
1	1	30000	1	2	2025.12.26 22:52:01
2	2	25000	2	1	2025.12.26 22:52:01
3	1	0	1	1	2025.12.26 22:54:38

Left sidebar navigation:

- Модели обуви
- Поставки
- Заказы** (highlighted)
- Поставщики
- Продавцы

Bottom left corner:

Рис. 4 – Страница заказов

Добавление заказа
Укажите информацию о заказе

Продавец
Выберите Продавца

Количество
0 Цена
0

Модель обуви
Выберите модель

Добавить

Рис. 5 – Страница добавления заказа

The screenshot shows a dark-themed application interface. On the left is a sidebar with user information (admin@test.com, Admin) and navigation links (Модели обуви, Поставки, Заказы, Поставщики, Продавцы, Выйти). The main content area is titled 'Поставщики' and contains a table with two rows:

Id	Поставщик
1	ООО "Обувной Мир"
2	Nike Official Rus

Рис. 6 – Страница поставщиков

The screenshot shows a dark-themed application interface. On the left is a sidebar with user information (admin@test.com, Admin) and navigation links (Модели обуви, Поставки, Заказы, Поставщики, Продавцы, Выйти). The main content area is titled 'Продавцы' and contains a table with three rows:

Id продавца	Имя	Фамилия	Отчество	Номер телефона	Должность
1	Иван	Петров	Васильевич	89005554433	Директор
2	Анна	Смирнова	Игоревна	89001112233	Старший продавец
3	Василий	Петров	Павлович	79851727731	Младший продавец

Рис. 7 – Страница продавцов

Добавление продавца

Укажите информацию о продавце

Фамилия
Введите фамилию

Имя
Введите имя

Отчество
Введите отчество

Номер телефона
Введите номер телефона

Создайте данные для входа в аккаунт продавца

Email
Введите Email

Пароль
Введите Пароль

Добавить

This screenshot shows the 'Добавление продавца' (Adding a seller) form. It consists of several input fields: 'Фамилия' (Family) with placeholder 'Введите фамилию', 'Имя' (Name) with placeholder 'Введите имя', 'Отчество' (Middle name) with placeholder 'Введите отчество', 'Номер телефона' (Phone number) with placeholder 'Введите номер телефона', 'Email' with placeholder 'Введите Email', and 'Пароль' (Password) with placeholder 'Введите Пароль'. Below these fields is a large 'Добавить' (Add) button.

Рис. 8 – Добавление продавцов

Редактирование продавца с id 3

Укажите информацию о продавце

Фамилия
Петров

Имя
Василий

Отчество
Павлович

Номер телефона
79851727731

Должность
Младший продавец

Добавить

This screenshot shows the 'Редактирование продавца с id 3' (Editing seller with id 3) form. It displays the following information: 'Фамилия' (Family) set to 'Петров', 'Имя' (Name) set to 'Василий', 'Отчество' (Middle name) set to 'Павлович', 'Номер телефона' (Phone number) set to '79851727731', and 'Должность' (Position) set to 'Младший продавец'. At the bottom is a large 'Добавить' (Add) button.

Рис. 9 – Изменение продавцов