

ГУАП

КАФЕДРА № 43

ОТЧЕТ ЗАЩИЩЕН С ОЦЕНКОЙ  
ПРЕПОДАВАТЕЛЬ

---

должность, уч. степень,  
звание

---

подпись, дата

---

инициалы, фамилия

ОТЧЁТ О ПРАКТИКО-ОРИЕНТИРОВАННОМ ПРОЕКТЕ

по курсу: JavaScript, его библиотеки и фреймворки в Frontend-разработке

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТЫ ГР.

4134К

---

подпись, дата

---

инициалы, фамилия

## Задание

### Требования к проекту

Разработать Single Page Application (SPA), выполненный на основе библиотеки Vue.js на согласованную с преподавателем тему, отвечающий требованиям адаптивности, кроссбраузерности, интерактивности. Допускается использование для разработки проекта библиотеки React.js в случае, если обучающийся самостоятельно в необходимом объеме освоил принципы работы с данной библиотекой.

Для проекта целесообразно выбирать тему, которая может быть достаточно полно раскрыта в объеме данной работы. Объем веб-сайта определяется требованием полноты представления информации по рассматриваемой теме. Можно выбрать тему, включенную в перечень рекомендуемых тем (см. табл. 1). Разрешается выбрать тему самостоятельно (можно использовать тематику, в соответствии с которой выполнялись практические задания по дисциплине).

Проект должен представлять собой одностраничный веб-сайт. Обязательными элементами интерфейса должны быть несколько вкладок (табов), оформленных как навигационное меню, в соответствии с которыми контент сайта должен быть разделен на несколько частей.

Навигационное меню должно быть фиксированным и не прокручиваться вместе со страницей, а нажатие на ссылки должны вести на соответствующий компонент без перезагрузки страницы (вкладки должны открываться без перезагрузки страницы).

В работе должны быть использованы современные технологии разработки web-сайтов, которые были изучены в процессе теоретического обучения по данному ИТ-модулю. Обязательным является использование HTML5, технологии CSS3, а также программирования на JavaScript, Vue.js.

При обеспечении адаптивности необходимо предусмотреть 3 контрольные точки (например, 1200px, 800px, 550px).

Следует обратить внимание на продуманность дизайна веб-страницы, ее архитектуры, на удобство навигации, привлекательность подачи информации.

На одну из вкладок следует добавить форму с необходимыми элементами (для отдельных полей должна быть предусмотрена обязательность заполнения). При нажатии на кнопку отправки должно появляться модальное окно с отображением успеха отправки (условие отправки можно не выполнять, т.е. страница не должна перезагружаться), либо с указанием ошибки при заполнении (например, пустое поле) через использование регулярных выражений JavaScript.

Еще раз обращаем внимание на то, что любая перезагрузка страницы с использованием переходов, отправок форм строго запрещена! В проекте должна существовать только одна веб-страница!

Весь функционал и реактивность страницы должны быть описаны с использованием методов и хук Vue.js.

На странице обязательно должен присутствовать компонент «слайдер изображений в галерее». Для смены картинок необходимо наличие кнопок (вправо/влево или </> и т.п.). Кнопки должны иметь стиливое оформление. Анимация перехода приветствуется, но не обязательна.

При выполнении работы разрешается использование библиотеки Bootstrap 5-ой версии.

Дизайн сайта можно придумать самому, но приветствуется также и использование бесплатных макетов Figma (верстка исключительно через position: absolute запрещена!)

В результате выполнения проекта должны быть представлены:

- веб-сайт для демонстрации,
- краткая пояснительная записка.

Для загрузки файлов проекта в LMS требуется папка проекта, созданного через Vue-CLI, без node\_modules (ее загружать запрещено). Все файлы следует объединить в единый архив и загрузить в LMS.

### **Выполнение работы:**

Сайт посвящен личному кабинету для маркетплейса, где имеется возможность управления товарами и отслеживания текущей статистики.

Работа была выполнена студентами:

1. Погодин Иван Денисович
2. Самарин Дмитрий Васильевич
3. Усов Роман Романович

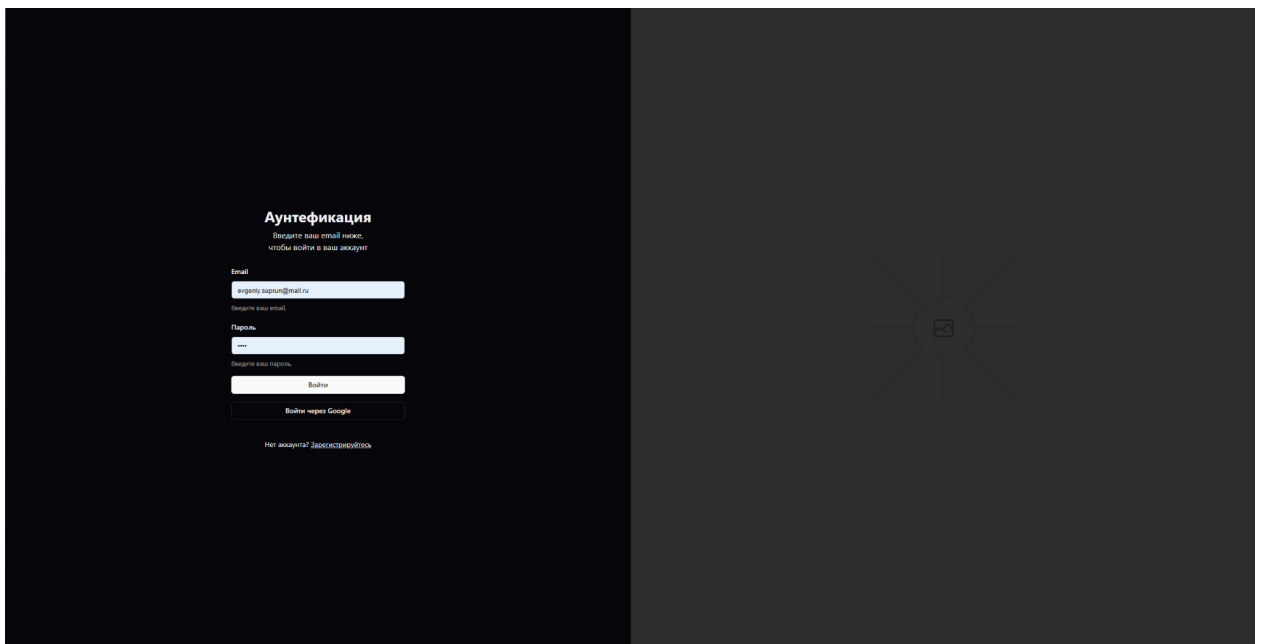
Каждый из студентов сделал свой вклад в разработку клиентских компонентов и логики приложения в целом.

Приложение имеет макетную составляющую для будущей доработки. Реализованы базовые компоненты и их логика. Также для демонстрации работы компонентов были прописаны клиентский роутинг и 3 страницы в общей сложности.

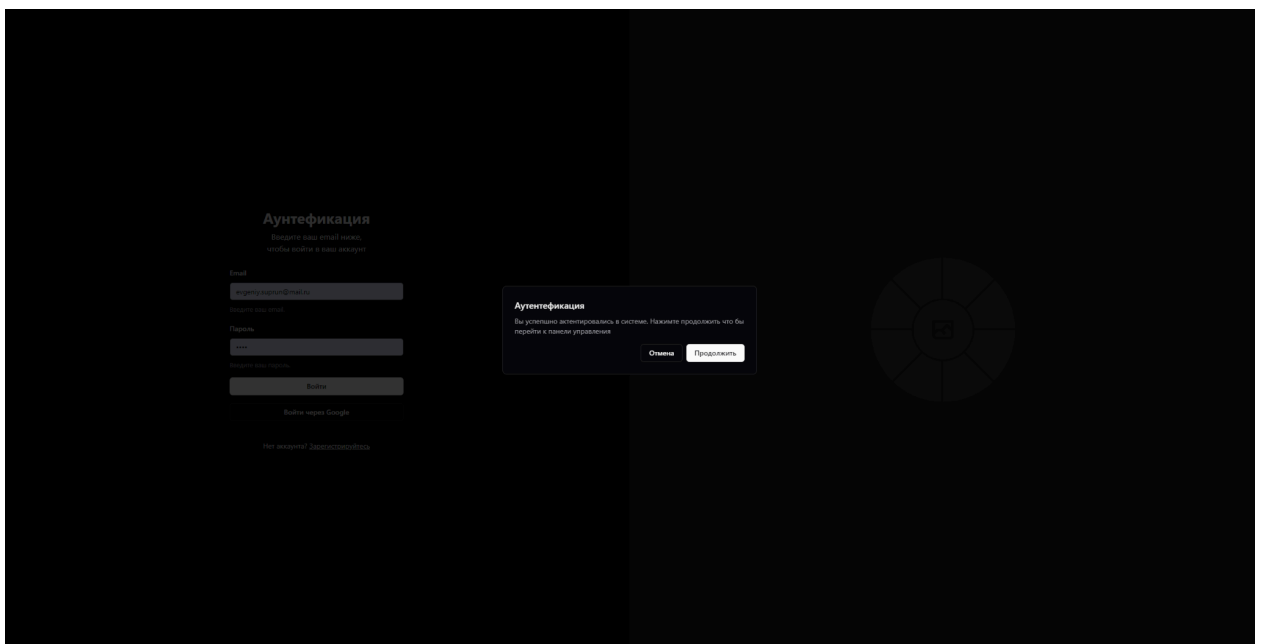
Для выполнения работы был выбран следующий стек технологий:

1. React vite - библиотека для построения клиентских WEB-интерфейсов на компонентном подходе. Как сборщик был выбран Vite как рекомендуется в основной документации библиотеки React.
2. Tailwind css - библиотека предоставляющая готовый набор css классов для стилизации приложения
3. react-router - библиотека, предоставляющая API для осуществления роутинга на стороне клиента без перезагрузки и смены экранов страницы.
4. class-variance-authority - библиотека для умного объединения стилей для избежания повторных использований css классов. Так же библиотека позволяет описывать стиливые состояния в зависимости от состояния компонента.
5. react-hook-form - библиотека для захвата действий и валидации форм на основе пользовательских хуков. Использует в себе zod для построения схем форм.
6. eslint - добавление правил для написания кода.

**Результат работы:**

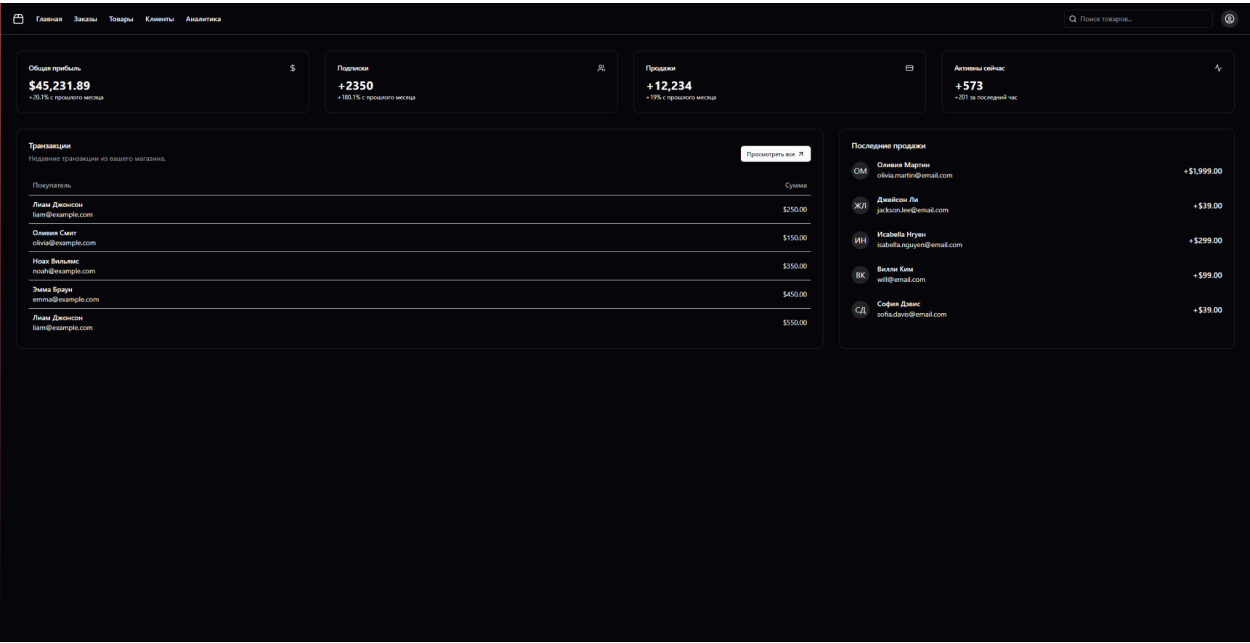


страница авторизации



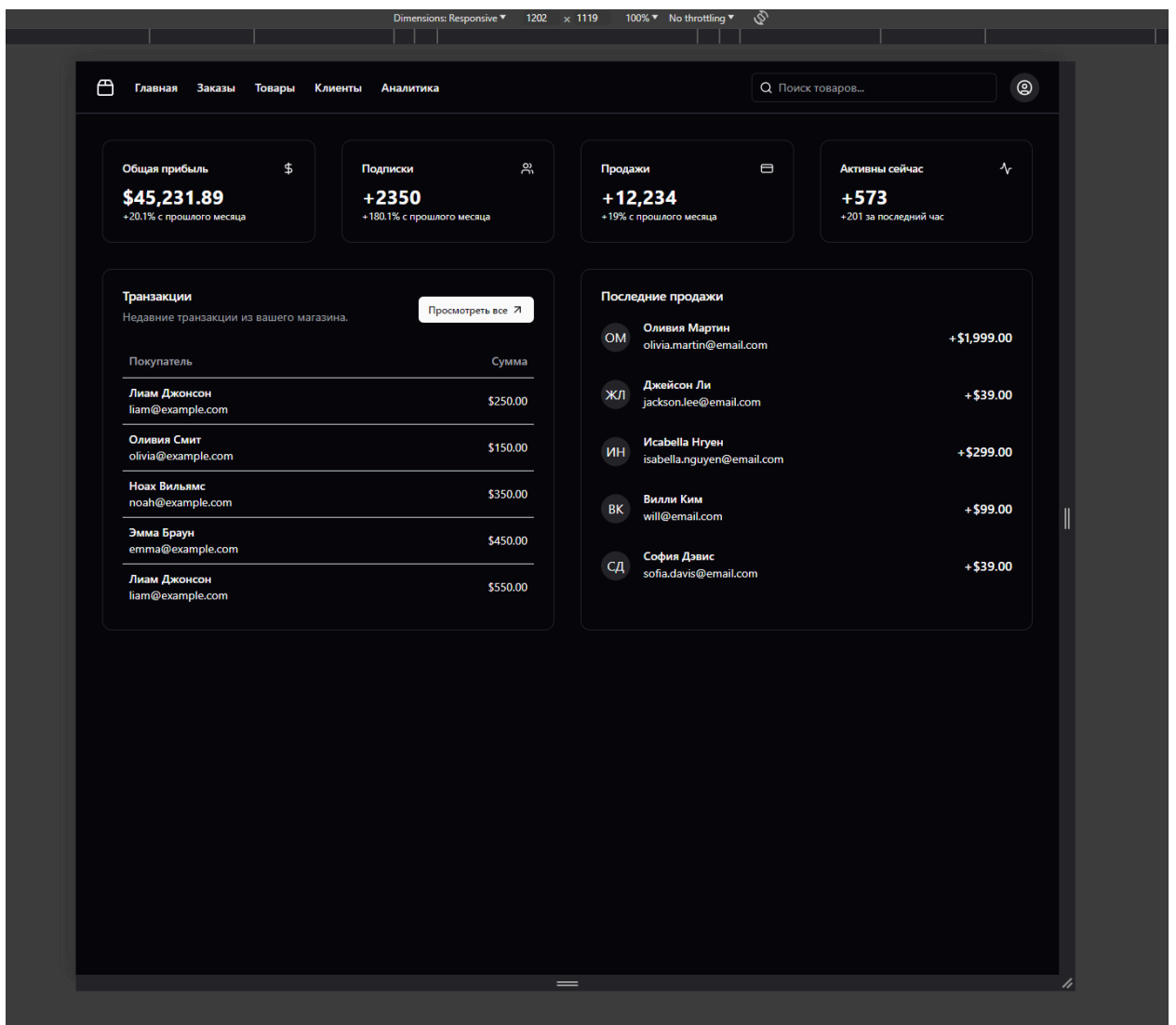
Модальное окно после авторизации

После авторизации пользователю высвечивается модальное окно как заглушка, так как серверные сценарии не были реализованы до конца. После подтверждения пользователь будет перенаправлен на главную страницу.

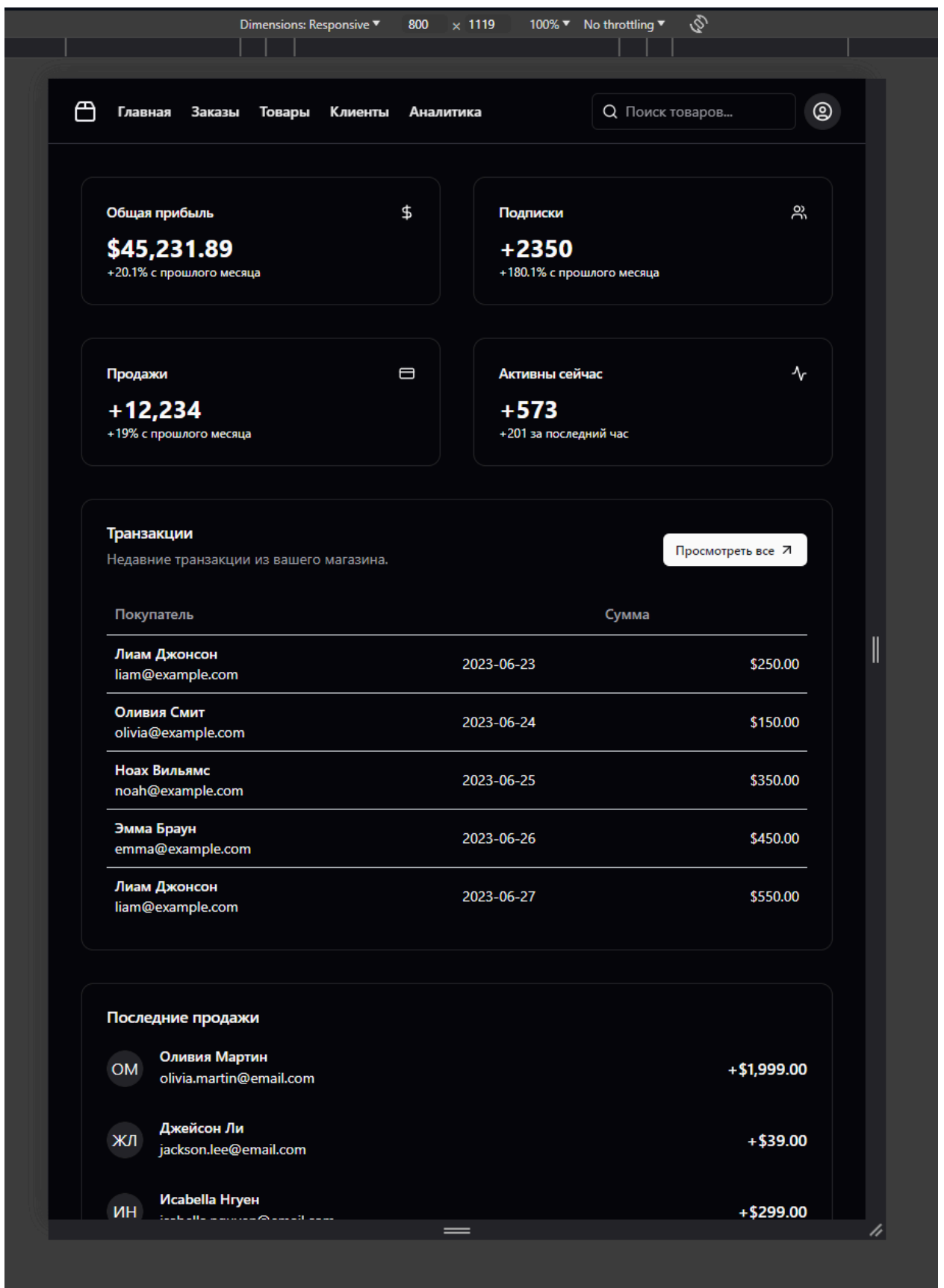


Главная страница

На главной странице представлена работа части реализованных компонентов. В них была предоставлена тестовая информация для заполнения страницы. Так же всё приложение поддерживает адаптивность на несколько экранов а так же светлую и тёмные темы.



Пример адаптивности - 1



Пример адаптивности - 2





Общая прибыль

\$

**\$45,231.89**

+20.1% с прошлого месяца

Подписки

👤

**+2350**

+180.1% с прошлого месяца

Продажи

📦

**+12,234**

+19% с прошлого месяца

Активны сейчас

📈

**+573**

+201 за последний час

## Транзакции

Недавние транзакции из вашего магазина.

[Просмотреть все ↗](#)

Покупатель	Сумма
Лиам Джонсон	\$250.00
Оливия Смит	\$150.00
Ноах Вильямс	\$350.00
Эмма Браун	\$450.00
Лиам Джонсон	\$550.00

## Последние продажи

Оливия Мартин  
olivia.martin@email.com

**+\$1,999.00**

## Пример адаптивности - 3



Главная

Заказы

Товары

Клиенты

Аналитика



\$



смотреть все ↗

Сумма

\$250.00

\$150.00

\$350.00

\$450.00

\$550.00

## Пример адаптивности - 4



# Аунтефикация

Введите ваш email ниже,  
чтобы войти в ваш аккаунт

Email

Введите ваш email.

Пароль

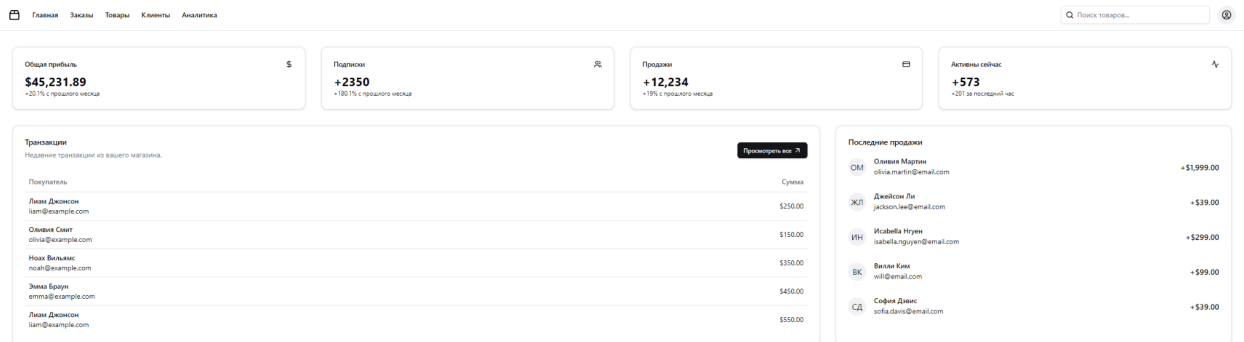
Введите ваш пароль.

Войти

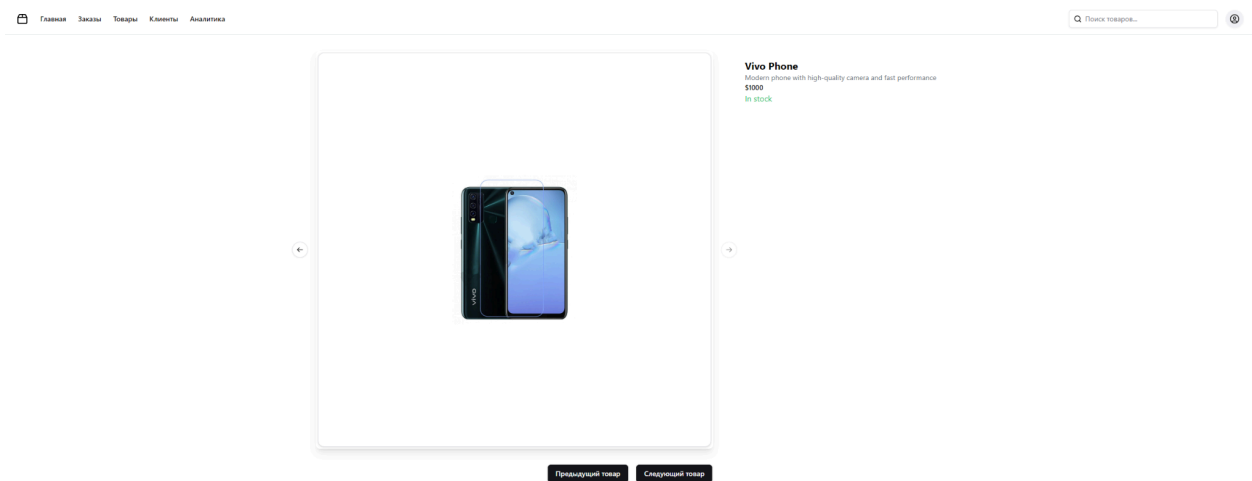
Войти через Google

Нет аккаунта? [Зарегистрируйтесь](#)

### Пример адаптивности - 5

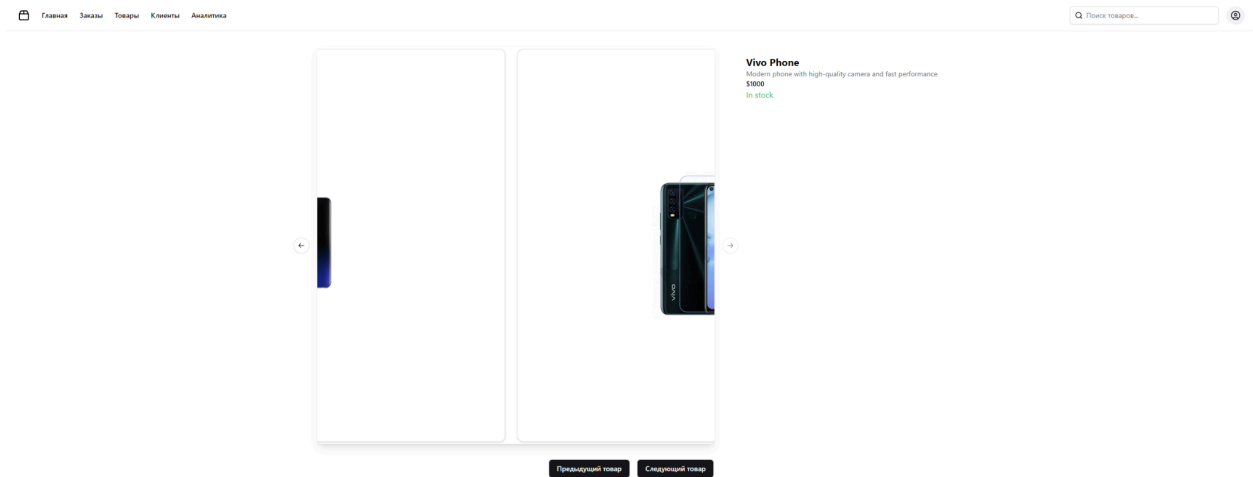


светлая тема главной страницы



страница товаров

Также пользователю доступна страница всех товаров с минимальным кол-вом описания товара а также каруселью изображений товара. Карусель поддерживает переключения по свайпу и по кнопка сбоку от карусели.



Пример карусели по свайпу

## Листинг кода:

```
"use client"

import * as React from "react"
import * as AvatarPrimitive from "@radix-ui/react-avatar"

import { cn } from "@/lib/utils"

const Avatar = React.forwardRef<
  React.ElementRef<typeof AvatarPrimitive.Root>,
  React.ComponentPropsWithoutRef<typeof AvatarPrimitive.Root>
>(({ className, ...props }, ref) => (
  <AvatarPrimitive.Root
    ref={ref}
    className={cn(
      "relative flex h-10 w-10 shrink-0 overflow-hidden rounded-full",
      className
    )}
    {...props}
```

```

    />
  ))
  Avatar.displayName = AvatarPrimitive.Root.displayName

  const AvatarImage = React.forwardRef<
    React.ElementRef<typeof AvatarPrimitive.Image>,
    React.ComponentPropsWithoutRef<typeof AvatarPrimitive.Image>
  >(({ className, ...props }, ref) => (
    <AvatarPrimitive.Image
      ref={ref}
      className={cn("aspect-square h-full w-full", className)}
      {...props}
    />
  ))
  AvatarImage.displayName = AvatarPrimitive.Image.displayName

  const AvatarFallback = React.forwardRef<
    React.ElementRef<typeof AvatarPrimitive.Fallback>,
    React.ComponentPropsWithoutRef<typeof AvatarPrimitive.Fallback>
  >(({ className, ...props }, ref) => (
    <AvatarPrimitive.Fallback
      ref={ref}
      className={cn(
        "flex h-full w-full items-center justify-center rounded-full bg-zinc-100
dark:bg-zinc-800",
        className
      )}
      {...props}
    />
  ))
  AvatarFallback.displayName = AvatarPrimitive.Fallback.displayName

  export { Avatar, AvatarImage, AvatarFallback }

```

"use client"

```

import * as React from "react"
import * as AvatarPrimitive from "@radix-ui/react-avatar"

import { cn } from "@/lib/utils"

const Avatar = React.forwardRef<

```



```

    React.ElementRef<typeof AvatarPrimitive.Root>,
    React.ComponentPropsWithoutRef<typeof AvatarPrimitive.Root>
  >(({ className, ...props }, ref) => (
    <AvatarPrimitive.Root
      ref={ref}
      className={cn(
        "relative flex h-10 w-10 shrink-0 overflow-hidden rounded-full",
        className
      )}
      {...props}
    />
  ))
Avatar.displayName = AvatarPrimitive.Root.displayName

const AvatarImage = React.forwardRef<
  React.ElementRef<typeof AvatarPrimitive.Image>,
  React.ComponentPropsWithoutRef<typeof AvatarPrimitive.Image>
>(({ className, ...props }, ref) => (
  <AvatarPrimitive.Image
    ref={ref}
    className={cn("aspect-square h-full w-full", className)}
    {...props}
  />
))
AvatarImage.displayName = AvatarPrimitive.Image.displayName

const AvatarFallback = React.forwardRef<
  React.ElementRef<typeof AvatarPrimitive.Fallback>,
  React.ComponentPropsWithoutRef<typeof AvatarPrimitive.Fallback>
>(({ className, ...props }, ref) => (
  <AvatarPrimitive.Fallback
    ref={ref}
    className={cn(
      "flex h-full w-full items-center justify-center rounded-full bg-zinc-100
dark:bg-zinc-800",
      className
    )}
    {...props}
  />
))
AvatarFallback.displayName = AvatarPrimitive.Fallback.displayName

```

```
export { Avatar, AvatarImage, AvatarFallback }
```

```
import * as React from "react"
import { cva, type VariantProps } from "class-variance-authority"

import { cn } from "@/lib/utils"

const badgeVariants = cva(
  "inline-flex items-center rounded-md border border-zinc-200 px-2.5 py-0.5 text-xs font-semibold transition-colors focus:outline-none focus:ring-2 focus:ring-zinc-950 focus:ring-offset-2 dark:border-zinc-800 dark:focus:ring-zinc-300",
  {
    variants: {
      variant: {
        default:
          "border-transparent bg-zinc-900 text-zinc-50 shadow hover:bg-zinc-900/80 dark:bg-zinc-50 dark:text-zinc-900 dark:hover:bg-zinc-50/80",
        secondary:
          "border-transparent bg-zinc-100 text-zinc-900 hover:bg-zinc-100/80 dark:bg-zinc-800 dark:text-zinc-50 dark:hover:bg-zinc-800/80",
        destructive:
          "border-transparent bg-red-500 text-zinc-50 shadow hover:bg-red-500/80 dark:bg-red-900 dark:text-zinc-50 dark:hover:bg-red-900/80",
        outline: "text-zinc-950 dark:text-zinc-50",
      },
    },
    defaultVariants: {
      variant: "default",
    },
  }
)

export interface BadgeProps
  extends React.HTMLAttributes<HTMLDivElement>,
    VariantProps<typeof badgeVariants> {}

function Badge({ className, variant, ...props }: BadgeProps) {
  return (
    <div className={cn(badgeVariants({ variant }), className)} {...props} />
  )
}
```

```
)  
}  
  
export { Badge, badgeVariants }
```

```
import * as React from "react"  
import { Slot } from "@radix-ui/react-slot"  
import { cva, type VariantProps } from "class-variance-authority"  
  
import { cn } from "@lib/utils"  
  
const buttonVariants = cva(  
  "inline-flex items-center justify-center whitespace-nowrap rounded-md text-sm  
font-medium transition-colors focus-visible:outline-none focus-visible:ring-1  
focus-visible:ring-zinc-950 disabled:pointer-events-none disabled:opacity-50  
dark:focus-visible:ring-zinc-300",  
  {  
    variants: {  
      variant: {  
        default:  
          "bg-zinc-900 text-zinc-50 shadow hover:bg-zinc-900/90 dark:bg-zinc-50  
dark:text-zinc-900 dark:hover:bg-zinc-50/90",  
        destructive:  
          "bg-red-500 text-zinc-50 shadow-sm hover:bg-red-500/90  
dark:bg-red-900 dark:text-zinc-50 dark:hover:bg-red-900/90",  
        outline:  
          "border border-zinc-200 bg-white shadow-sm hover:bg-zinc-100  
hover:text-zinc-900 dark:border-zinc-800 dark:bg-zinc-950  
dark:hover:bg-zinc-800 dark:hover:text-zinc-50",  
        secondary:  
          "bg-zinc-100 text-zinc-900 shadow-sm hover:bg-zinc-100/80  
dark:bg-zinc-800 dark:text-zinc-50 dark:hover:bg-zinc-800/80",  
        ghost: "hover:bg-zinc-100 hover:text-zinc-900 dark:hover:bg-zinc-800  
dark:hover:text-zinc-50",  
        link: "text-zinc-900 underline-offset-4 hover:underline  
dark:text-zinc-50",  
      },  
      size: {  
        default: "h-9 px-4 py-2",  
        sm: "h-8 rounded-md px-3 text-xs",  
        lg: "h-10 rounded-md px-8",  
        icon: "h-9 w-9",  
      },  
    },  
  },  
)
```

```

    },
  },
  defaultVariants: {
    variant: "default",
    size: "default",
  },
}
)

export interface ButtonProps
  extends React.ButtonHTMLAttributes<HTMLButtonElement>,
    VariantProps<typeof buttonVariants> {
  asChild?: boolean
}

const Button = React.forwardRef<HTMLButtonElement, ButtonProps>(
  ({ className, variant, size, asChild = false, ...props }, ref) => {
    const Comp = asChild ? Slot : "button"
    return (
      <Comp
        className={cn(buttonVariants({ variant, size, className }))}
        ref={ref}
        {...props}
      />
    )
  }
)

Button.displayName = "Button"

export { Button, buttonVariants }

```

```

import * as React from "react";

import { cn } from "@/lib/utils";

const Card = React.forwardRef<
  HTMLDivElement,
  React.HTMLAttributes<HTMLDivElement>
>(({ className, ...props }, ref) => (
  <div
    ref={ref}
    className={cn(

```

```

        "rounded-xl border border-zinc-200 bg-white text-zinc-950 shadow
dark:border-zinc-800 dark:bg-zinc-950 dark:text-zinc-50",
        className
    )}
    {...props}
  />
));
Card.displayName = "Card";

const CardHeader = React.forwardRef<
  HTMLDivElement,
  React.HTMLAttributes<HTMLDivElement>
>(({ className, ...props }, ref) => (
  <div
    ref={ref}
    className={cn("flex flex-col space-y-1.5 p-6", className)}
    {...props}
  />
));
CardHeader.displayName = "CardHeader";

const CardTitle = React.forwardRef<
  HTMLParagraphElement,
  React.HTMLAttributes<HTMLHeadingElement>
>(({ className, ...props }, ref) => (
  <h3
    ref={ref}
    className={cn("font-semibold leading-none tracking-tight", className)}
    {...props}
  />
));
CardTitle.displayName = "CardTitle";

const CardDescription = React.forwardRef<
  HTMLParagraphElement,
  React.HTMLAttributes<HTMLParagraphElement>
>(({ className, ...props }, ref) => (
  <p
    ref={ref}
    className={cn("text-sm text-zinc-500 dark:text-zinc-400", className)}
    {...props}
  />

```

```

));
CardDescription.displayName = "CardDescription";

const CardContent = React.forwardRef<
  HTMLDivElement,
  React.HTMLAttributes<HTMLDivElement>
>(({ className, ...props }, ref) => (
  <div ref={ref} className={cn("p-6 pt-0", className)} {...props} />
));
CardContent.displayName = "CardContent";

const CardFooter = React.forwardRef<
  HTMLDivElement,
  React.HTMLAttributes<HTMLDivElement>
>(({ className, ...props }, ref) => (
  <div
    ref={ref}
    className={cn("flex items-center p-6 pt-0", className)}
    {...props}
  />
));
CardFooter.displayName = "CardFooter";

export {
  Card,
  CardHeader,
  CardFooter,
  CardTitle,
  CardDescription,
  CardContent,
};

```

"use client"

```

import * as React from "react"
import { ArrowLeftIcon, ArrowRightIcon } from "@radix-ui/react-icons"
import useEmblaCarousel, {
  type UseEmblaCarouselType,
} from "embla-carousel-react"

import { cn } from "@/lib/utils"
import { Button } from "@/components/ui/button"

```

```

type CarouselApi = UseEmblaCarouselType[1]
type UseCarouselParameters = Parameters<typeof useEmblaCarousel>
type CarouselOptions = UseCarouselParameters[0]
type CarouselPlugin = UseCarouselParameters[1]

type CarouselProps = {
  opts?: CarouselOptions
  plugins?: CarouselPlugin
  orientation?: "horizontal" | "vertical"
  setApi?: (api: CarouselApi) => void
}

type CarouselContextProps = {
  carouselRef: ReturnType<typeof useEmblaCarousel>[0]
  api: ReturnType<typeof useEmblaCarousel>[1]
  scrollPrev: () => void
  scrollNext: () => void
  canScrollPrev: boolean
  canScrollNext: boolean
} & CarouselProps

const CarouselContext = React.createContext<CarouselContextProps | null>(null)

function useCarousel() {
  const context = React.useContext(CarouselContext)

  if (!context) {
    throw new Error("useCarousel must be used within a <Carousel />")
  }

  return context
}

const Carousel = React.forwardRef<
  HTMLDivElement,
  React.HTMLAttributes<HTMLDivElement> & CarouselProps
>(
  (
    {
      {
        orientation = "horizontal",
        opts,

```

```

    setApi,
    plugins,
    className,
    children,
    ...props
  },
  ref
) => {
  const [carouselRef, api] = useEmblaCarousel(
    {
      ...opts,
      axis: orientation === "horizontal" ? "x" : "y",
    },
    plugins
  )
  const [canScrollPrev, setCanScrollPrev] = React.useState(false)
  const [canScrollNext, setCanScrollNext] = React.useState(false)

  const onSelect = React.useCallback((api: CarouselApi) => {
    if (!api) {
      return
    }

    setCanScrollPrev(api.canScrollPrev())
    setCanScrollNext(api.canScrollNext())
  }, [])

  const scrollPrev = React.useCallback(() => {
    api?.scrollPrev()
  }, [api])

  const scrollNext = React.useCallback(() => {
    api?.scrollNext()
  }, [api])

  const handleKeyDown = React.useCallback(
    (event: React.KeyboardEvent<HTMLDivElement>) => {
      if (event.key === "ArrowLeft") {
        event.preventDefault()
        scrollPrev()
      } else if (event.key === "ArrowRight") {
        event.preventDefault()

```



```

        scrollNext()
    }
},
[scrollPrev, scrollNext]
)

React.useEffect(() => {
    if (!api || !setApi) {
        return
    }

    setApi(api)
}, [api, setApi])

React.useEffect(() => {
    if (!api) {
        return
    }

    onSelect(api)
    api.on("reInit", onSelect)
    api.on("select", onSelect)

    return () => {
        api?.off("select", onSelect)
    }
}, [api, onSelect])

return (
    <CarouselContext.Provider
        value={{
            carouselRef,
            api: api,
            opts,
            orientation:
                orientation || (opts?.axis === "y" ? "vertical" : "horizontal"),
            scrollPrev,
            scrollNext,
            canScrollPrev,
            canScrollNext,
        }}
    >

```

```

        <div
          ref={ref}
          onKeyDownCapture={handleKeyDown}
          className={cn("relative", className)}
          role="region"
          aria-roledescription="carousel"
          {...props}
        >
          {children}
        </div>
      </CarouselContext.Provider>
    )
  }
)
Carousel.displayName = "Carousel"

const CarouselContent = React.forwardRef<
  HTMLDivElement,
  React.HTMLAttributes<HTMLDivElement>
>(({ className, ...props }, ref) => {
  const { carouselRef, orientation } = useCarousel()

  return (
    <div ref={carouselRef} className="overflow-hidden">
      <div
        ref={ref}
        className={cn(
          "flex",
          orientation === "horizontal" ? "-ml-4" : "-mt-4 flex-col",
          className
        )}
        {...props}
      />
    </div>
  )
})
CarouselContent.displayName = "CarouselContent"

const CarouselItem = React.forwardRef<
  HTMLDivElement,
  React.HTMLAttributes<HTMLDivElement>
>(({ className, ...props }, ref) => {

```

```

const { orientation } = useCarousel()

return (
  <div
    ref={ref}
    role="group"
    aria-roledescription="slide"
    className={cn(
      "min-w-0 shrink-0 grow-0 basis-full",
      orientation === "horizontal" ? "pl-4" : "pt-4",
      className
    )}
    {...props}
  />
)
}))
CarouselItem.displayName = "CarouselItem"

const CarouselPrevious = React.forwardRef<
  HTMLButtonElement,
  React.ComponentProps<typeof Button>
>(({ className, variant = "outline", size = "icon", ...props }, ref) => {
  const { orientation, scrollPrev, canScrollPrev } = useCarousel()

  return (
    <Button
      ref={ref}
      variant={variant}
      size={size}
      className={cn(
        "absolute cursor-pointer h-8 w-8 rounded-full",
        orientation === "horizontal"
          ? "-left-12 top-1/2 -translate-y-1/2"
          : "-top-12 left-1/2 -translate-x-1/2 rotate-90",
        className
      )}
      disabled={!canScrollPrev}
      onClick={scrollPrev}
      {...props}
    >
    <ArrowLeftIcon className="h-4 w-4" />
    <span className="sr-only">Previous slide</span>

```

```

    </Button>
  )
})
CarouselPrevious.displayName = "CarouselPrevious"

const CarouselNext = React.forwardRef<
  HTMLButtonElement,
  React.ComponentProps<typeof Button>
>(({ className, variant = "outline", size = "icon", ...props }, ref) => {
  const { orientation, scrollNext, canScrollNext } = useCarousel()

  return (
    <Button
      ref={ref}
      variant={variant}
      size={size}
      className={cn(
        "absolute h-8 w-8 rounded-full",
        orientation === "horizontal"
          ? "-right-12 top-1/2 -translate-y-1/2"
          : "-bottom-12 left-1/2 -translate-x-1/2 rotate-90",
        className
      )}
      disabled={!canScrollNext}
      onClick={scrollNext}
      {...props}
    >
      <ArrowRightIcon className="h-4 w-4" />
      <span className="sr-only">Next slide</span>
    </Button>
  )
})
CarouselNext.displayName = "CarouselNext"

export {
  type CarouselApi,
  Carousel,
  CarouselContent,
  CarouselItem,
  CarouselPrevious,
  CarouselNext,
}

```

```

"use client"

import * as React from "react"
import * as DropdownMenuPrimitive from "@radix-ui/react-dropdown-menu"
import {
  CheckIcon,
  ChevronRightIcon,
  DotFilledIcon,
} from "@radix-ui/react-icons"

import { cn } from "@lib/utils"

const DropdownMenu = DropdownMenuPrimitive.Root

const DropdownMenuTrigger = DropdownMenuPrimitive.Trigger

const DropdownMenuGroup = DropdownMenuPrimitive.Group

const DropdownMenuPortal = DropdownMenuPrimitive.Portal

const DropdownMenuSub = DropdownMenuPrimitive.Sub

const DropdownMenuRadioGroup = DropdownMenuPrimitive.RadioGroup

const DropdownMenuSubTrigger = React.forwardRef<
  React.ElementRef<typeof DropdownMenuPrimitive.SubTrigger>,
  React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.SubTrigger> & {
    inset?: boolean
  }
>(({ className, inset, children, ...props }, ref) => (
  <DropdownMenuPrimitive.SubTrigger
    ref={ref}
    className={cn(
      "flex cursor-default select-none items-center rounded-sm px-2 py-1.5
text-sm outline-none focus:bg-zinc-100 data-[state=open]:bg-zinc-100
dark:focus:bg-zinc-800 dark:data-[state=open]:bg-zinc-800",
      inset && "pl-8",
      className
    )}
    {...props}
  />

```

```

    {children}
    <ChevronRightIcon className="ml-auto h-4 w-4" />
  </DropdownMenuPrimitive.SubTrigger>
))
DropdownMenuSubTrigger.displayName =
  DropdownMenuPrimitive.SubTrigger.displayName

const DropdownMenuSubContent = React.forwardRef<
  React.ElementRef<typeof DropdownMenuPrimitive.SubContent>,
  React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.SubContent>
>(({ className, ...props }, ref) => (
  <DropdownMenuPrimitive.SubContent
    ref={ref}
    className={cn(
      "z-50 min-w-[8rem] overflow-hidden rounded-md border border-zinc-200
      bg-white p-1 text-zinc-950 shadow-lg data-[state=open]:animate-in
      data-[state=closed]:animate-out data-[state=closed]:fade-out-0
      data-[state=open]:fade-in-0 data-[state=closed]:zoom-out-95
      data-[state=open]:zoom-in-95 data-[side=bottom]:slide-in-from-top-2
      data-[side=left]:slide-in-from-right-2 data-[side=right]:slide-in-from-left-2
      data-[side=top]:slide-in-from-bottom-2 dark:border-zinc-800 dark:bg-zinc-950
      dark:text-zinc-50",
      className
    )}
    {...props}
  />
))
DropdownMenuSubContent.displayName =
  DropdownMenuPrimitive.SubContent.displayName

const DropdownMenuContent = React.forwardRef<
  React.ElementRef<typeof DropdownMenuPrimitive.Content>,
  React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.Content>
>(({ className, sideOffset = 4, ...props }, ref) => (
  <DropdownMenuPrimitive.Portal>
    <DropdownMenuPrimitive.Content
      ref={ref}
      sideOffset={sideOffset}
      className={cn(
        "z-50 min-w-[8rem] overflow-hidden rounded-md border border-zinc-200
        bg-white p-1 text-zinc-950 shadow-md dark:border-zinc-800 dark:bg-zinc-950
        dark:text-zinc-50",

```

```

        "data-[state=open]:animate-in data-[state=closed]:animate-out
data-[state=closed]:fade-out-0 data-[state=open]:fade-in-0
data-[state=closed]:zoom-out-95 data-[state=open]:zoom-in-95
data-[side=bottom]:slide-in-from-top-2 data-[side=left]:slide-in-from-right-2
data-[side=right]:slide-in-from-left-2 data-[side=top]:slide-in-from-bottom-2",
        className
    )}
    {...props}
  />
</DropdownMenuPrimitive.Portal>
))
DropdownMenuContent.displayName = DropdownMenuPrimitive.Content.displayName

const DropdownMenuItem = React.forwardRef<
  React.ElementRef<typeof DropdownMenuPrimitive.Item>,
  React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.Item> & {
    inset?: boolean
  }
>(({ className, inset, ...props }, ref) => (
  <DropdownMenuPrimitive.Item
    ref={ref}
    className={cn(
      "relative flex cursor-default select-none items-center rounded-sm px-2
py-1.5 text-sm outline-none transition-colors focus:bg-zinc-100
focus:text-zinc-900 data-[disabled]:pointer-events-none
data-[disabled]:opacity-50 dark:focus:bg-zinc-800 dark:focus:text-zinc-50",
      inset && "pl-8",
      className
    )}
    {...props}
  />
))
DropdownMenuItem.displayName = DropdownMenuPrimitive.Item.displayName

const DropdownMenuCheckboxItem = React.forwardRef<
  React.ElementRef<typeof DropdownMenuPrimitive.CheckboxItem>,
  React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.CheckboxItem>
>(({ className, children, checked, ...props }, ref) => (
  <DropdownMenuPrimitive.CheckboxItem
    ref={ref}
    className={cn(
      "relative flex cursor-default select-none items-center rounded-sm py-1.5

```

```

pl-8 pr-2 text-sm outline-none transition-colors focus:bg-zinc-100
focus:text-zinc-900 data-[disabled]:pointer-events-none
data-[disabled]:opacity-50 dark:focus:bg-zinc-800 dark:focus:text-zinc-50",
  className
)}
checked={checked}
{...props}
>
  <span className="absolute left-2 flex h-3.5 w-3.5 items-center
justify-center">
    <DropdownMenuPrimitive.ItemIndicator>
      <CheckIcon className="h-4 w-4" />
    </DropdownMenuPrimitive.ItemIndicator>
  </span>
  {children}
</DropdownMenuPrimitive.CheckboxItem>
))
DropdownMenuCheckboxItem.displayName =
  DropdownMenuPrimitive.CheckboxItem.displayName

const DropdownMenuRadioItem = React.forwardRef<
  React.ElementRef<typeof DropdownMenuPrimitive.RadioItem>,
  React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.RadioItem>
>(({ className, children, ...props }, ref) => (
  <DropdownMenuPrimitive.RadioItem
    ref={ref}
    className={cn(
      "relative flex cursor-default select-none items-center rounded-sm py-1.5
pl-8 pr-2 text-sm outline-none transition-colors focus:bg-zinc-100
focus:text-zinc-900 data-[disabled]:pointer-events-none
data-[disabled]:opacity-50 dark:focus:bg-zinc-800 dark:focus:text-zinc-50",
      className
    )}
    {...props}
  >
    <span className="absolute left-2 flex h-3.5 w-3.5 items-center
justify-center">
      <DropdownMenuPrimitive.ItemIndicator>
        <DotFilledIcon className="h-4 w-4 fill-current" />
      </DropdownMenuPrimitive.ItemIndicator>
    </span>
    {children}

```



```

    </DropdownMenuPrimitive.RadioItem>
  ))
  DropdownMenuRadioItem.displayName = DropdownMenuPrimitive.RadioItem.displayName

  const DropdownMenuLabel = React.forwardRef<
    React.ElementRef<typeof DropdownMenuPrimitive.Label>,
    React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.Label> & {
      inset?: boolean
    }
  >(({ className, inset, ...props }, ref) => (
    <DropdownMenuPrimitive.Label
      ref={ref}
      className={cn(
        "px-2 py-1.5 text-sm font-semibold",
        inset && "pl-8",
        className
      )}
      {...props}
    />
  ))
  DropdownMenuLabel.displayName = DropdownMenuPrimitive.Label.displayName

  const DropdownMenuSeparator = React.forwardRef<
    React.ElementRef<typeof DropdownMenuPrimitive.Separator>,
    React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.Separator>
  >(({ className, ...props }, ref) => (
    <DropdownMenuPrimitive.Separator
      ref={ref}
      className={cn("-mx-1 my-1 h-px bg-zinc-100 dark:bg-zinc-800", className)}
      {...props}
    />
  ))
  DropdownMenuSeparator.displayName = DropdownMenuPrimitive.Separator.displayName

  const DropdownMenuShortcut = ({
    className,
    ...props
  }: React.HTMLAttributes<HTMLSpanElement>) => {
    return (
      <span
        className={cn("ml-auto text-xs tracking-widest opacity-60", className)}
        {...props}

```

```

    />
  )
}
DropdownMenuShortcut.displayName = "DropdownMenuShortcut"

export {
  DropdownMenu,
  DropdownMenuTrigger,
  DropdownMenuContent,
  DropdownMenuItem,
  DropdownMenuCheckboxItem,
  DropdownMenuRadioItem,
  DropdownMenuLabel,
  DropdownMenuSeparator,
  DropdownMenuShortcut,
  DropdownMenuGroup,
  DropdownMenuPortal,
  DropdownMenuSub,
  DropdownMenuSubContent,
  DropdownMenuSubTrigger,
  DropdownMenuRadioGroup,
}

```

```

import * as React from "react"
import * as LabelPrimitive from "@radix-ui/react-label"
import { Slot } from "@radix-ui/react-slot"
import {
  Controller,
  ControllerProps,
  FieldPath,
  FieldValues,
  FormProvider,
  useFormContext,
} from "react-hook-form"

import { cn } from "@lib/utils"
import { Label } from "@components/ui/label"

const Form = FormProvider

type FormFieldContextValue<
  TFieldValues extends FieldValues = FieldValues,

```

```

    TName extends FieldPath<TFieldValues> = FieldPath<TFieldValues>
  > = {
    name: TName
  }

const FormFieldContext = React.createContext<FormFieldContextValue>(
  {} as FormFieldContextValue
)

const FormField = <
  TFieldValues extends FieldValues = FieldValues,
  TName extends FieldPath<TFieldValues> = FieldPath<TFieldValues>
>({
  ...props
}): ControllerProps<TFieldValues, TName>) => {
  return (
    <FormFieldContext.Provider value={{ name: props.name }}>
      <Controller {...props} />
    </FormFieldContext.Provider>
  )
}

const useFormField = () => {
  const fieldContext = React.useContext(FormFieldContext)
  const itemContext = React.useContext(FormItemContext)
  const { getFieldState, formState } = useFormContext()

  const fieldState = getFieldState(fieldContext.name, formState)

  if (!fieldContext) {
    throw new Error("useFormField should be used within <FormField>")
  }

  const { id } = itemContext

  return {
    id,
    name: fieldContext.name,
    formItemId: `${id}-form-item`,
    formDescriptionId: `${id}-form-item-description`,
    formMessageId: `${id}-form-item-message`,
    ...fieldState,
  }
}

```

```

    }
  }

type FormItemContextValue = {
  id: string
}

const FormItemContext = React.createContext<FormItemContextValue>(
  {} as FormItemContextValue
)

const FormItem = React.forwardRef<
  HTMLDivElement,
  React.HTMLAttributes<HTMLDivElement>
>(({ className, ...props }, ref) => {
  const id = React.useId()

  return (
    <FormItemContext.Provider value={{ id }}>
      <div ref={ref} className={cn("space-y-2", className)} {...props} />
    </FormItemContext.Provider>
  )
})

FormItem.displayName = "FormItem"

const FormLabel = React.forwardRef<
  React.ElementRef<typeof LabelPrimitive.Root>,
  React.ComponentPropsWithoutRef<typeof LabelPrimitive.Root>
>(({ className, ...props }, ref) => {
  const { error, formItemId } = useFormField()

  return (
    <Label
      ref={ref}
      className={cn(error && "text-red-500 dark:text-red-900", className)}
      htmlFor={formItemId}
      {...props}
    />
  )
})

FormLabel.displayName = "FormLabel"

```

```

const FormControl = React.forwardRef<
  React.ElementRef<typeof Slot>,
  React.ComponentPropsWithoutRef<typeof Slot>
>(({ ...props }, ref) => {
  const { error, formItemId, formDescriptionId, formMessageId } =
    useFormField()

  return (
    <Slot
      ref={ref}
      id={formItemId}
      aria-describedby={
        !error
          ? `${formDescriptionId}`
          : `${formDescriptionId} ${formMessageId}`
      }
      aria-invalid={!!error}
      {...props}
    />
  )
})
FormControl.displayName = "FormControl"

const FormDescription = React.forwardRef<
  HTMLParagraphElement,
  React.HTMLAttributes<HTMLParagraphElement>
>(({ className, ...props }, ref) => {
  const { formDescriptionId } = useFormField()

  return (
    <p
      ref={ref}
      id={formDescriptionId}
      className={cn("text-[0.8rem] text-zinc-500 dark:text-zinc-400",
        className)}
      {...props}
    />
  )
})
FormDescription.displayName = "FormDescription"

const FormMessage = React.forwardRef<

```

```

HTMLParagraphElement,
React.HTMLAttributes<HTMLParagraphElement>
>(({ className, children, ...props }, ref) => {
  const { error, formMessageId } = useFormField()
  const body = error ? String(error?.message) : children

  if (!body) {
    return null
  }

  return (
    <p
      ref={ref}
      id={formMessageId}
      className={cn("text-[0.8rem] font-medium text-red-500 dark:text-red-900",
className)}
      {...props}
    >
      {body}
    </p>
  )
})
FormMessage.displayName = "FormMessage"

export {
  useFormField,
  Form,
  FormItem,
  FormLabel,
  FormControl,
  FormDescription,
  FormMessage,
  FormField,
}

```

```

import * as React from "react"

import { cn } from "@/lib/utils"

export interface InputProps
  extends React.InputHTMLAttributes<HTMLInputElement> {}

```

```

const Input = React.forwardRef<HTMLInputElement, InputProps>(
  ({ className, type, ...props }, ref) => {
    return (
      <input
        type={type}
        className={cn(
          "flex h-9 w-full rounded-md border border-zinc-200 bg-transparent
px-3 py-1 text-sm shadow-sm transition-colors file:border-0 file:bg-transparent
file:text-sm file:font-medium placeholder:text-zinc-500
focus-visible:outline-none focus-visible:ring-1 focus-visible:ring-zinc-950
disabled:cursor-not-allowed disabled:opacity-50 dark:border-zinc-800
dark:placeholder:text-zinc-400 dark:focus-visible:ring-zinc-300",
          className
        )}
        ref={ref}
        {...props}
      />
    )
  }
)
Input.displayName = "Input"

export { Input }

```

```

"use client"

import * as React from "react"
import * as LabelPrimitive from "@radix-ui/react-label"
import { cva, type VariantProps } from "class-variance-authority"

import { cn } from "@/lib/utils"

const labelVariants = cva(
  "text-sm font-medium leading-none peer-disabled:cursor-not-allowed
peer-disabled:opacity-70"
)

const Label = React.forwardRef<
  React.ElementRef<typeof LabelPrimitive.Root>,
  React.ComponentPropsWithoutRef<typeof LabelPrimitive.Root> &
    VariantProps<typeof labelVariants>
>(({ className, ...props }, ref) => (

```

```

<LabelPrimitive.Root
  ref={ref}
  className={cn(labelVariants(), className)}
  {...props}
/>
))
Label.displayName = LabelPrimitive.Root.displayName

export { Label }

```

```

"use client"

import * as React from "react"
import * as SheetPrimitive from "@radix-ui/react-dialog"
import { Cross2Icon } from "@radix-ui/react-icons"
import { cva, type VariantProps } from "class-variance-authority"

import { cn } from "@/lib/utils"

const Sheet = SheetPrimitive.Root

const SheetTrigger = SheetPrimitive.Trigger

const SheetClose = SheetPrimitive.Close

const SheetPortal = SheetPrimitive.Portal

const SheetOverlay = React.forwardRef<
  React.ElementRef<typeof SheetPrimitive.Overlay>,
  React.ComponentPropsWithoutRef<typeof SheetPrimitive.Overlay>
>(({ className, ...props }, ref) => (
  <SheetPrimitive.Overlay
    className={cn(
      "fixed inset-0 z-50 bg-black/80 data-[state=open]:animate-in
data-[state=closed]:animate-out data-[state=closed]:fade-out-0
data-[state=open]:fade-in-0",
      className
    )}
    {...props}
    ref={ref}
  />
))

```



```

SheetOverlay.displayName = SheetPrimitive.Overlay.displayName

const sheetVariants = cva(
  "fixed z-50 gap-4 bg-white p-6 shadow-lg transition ease-in-out
  data-[state=open]:animate-in data-[state=closed]:animate-out
  data-[state=closed]:duration-300 data-[state=open]:duration-500
  dark:bg-zinc-950",
  {
    variants: {
      side: {
        top: "inset-x-0 top-0 border-b data-[state=closed]:slide-out-to-top
        data-[state=open]:slide-in-from-top",
        bottom:
          "inset-x-0 bottom-0 border-t data-[state=closed]:slide-out-to-bottom
        data-[state=open]:slide-in-from-bottom",
        left: "inset-y-0 left-0 h-full w-3/4 border-r
        data-[state=closed]:slide-out-to-left data-[state=open]:slide-in-from-left
        sm:max-w-sm",
        right:
          "inset-y-0 right-0 h-full w-3/4 border-l
        data-[state=closed]:slide-out-to-right data-[state=open]:slide-in-from-right
        sm:max-w-sm",
      },
    },
    defaultVariants: {
      side: "right",
    },
  }
)

interface SheetContentProps
  extends React.ComponentPropsWithoutRef<typeof SheetPrimitive.Content>,
    VariantProps<typeof sheetVariants> {}

const SheetContent = React.forwardRef<
  React.ElementRef<typeof SheetPrimitive.Content>,
  SheetContentProps
>(({ side = "right", className, children, ...props }, ref) => (
  <SheetPortal>
    <SheetOverlay />
    <SheetPrimitive.Content
      ref={ref}

```

```

        className={cn(sheetVariants({ side })), className)}
        {...props}
      >
        {children}
        <SheetPrimitive.Close className="absolute right-4 top-4 rounded-sm
opacity-70 ring-offset-white transition-opacity hover:opacity-100
focus:outline-none focus:ring-2 focus:ring-zinc-950 focus:ring-offset-2
disabled:pointer-events-none data-[state=open]:bg-zinc-100
dark:ring-offset-zinc-950 dark:focus:ring-zinc-300
dark:data-[state=open]:bg-zinc-800">
          <Cross2Icon className="h-4 w-4" />
          <span className="sr-only">Close</span>
        </SheetPrimitive.Close>
      </SheetPrimitive.Content>
    </SheetPortal>
  ))
  SheetContent.displayName = SheetPrimitive.Content.displayName

const SheetHeader = ({
  className,
  ...props
}: React.HTMLAttributes<HTMLDivElement>) => (
  <div
    className={cn(
      "flex flex-col space-y-2 text-center sm:text-left",
      className
    )}
    {...props}
  />
)
SheetHeader.displayName = "SheetHeader"

const SheetFooter = ({
  className,
  ...props
}: React.HTMLAttributes<HTMLDivElement>) => (
  <div
    className={cn(
      "flex flex-col-reverse sm:flex-row sm:justify-end sm:space-x-2",
      className
    )}
    {...props}
  >

```

```

    />
  )
  SheetFooter.displayName = "SheetFooter"

  const SheetTitle = React.forwardRef<
    React.ElementRef<typeof SheetPrimitive.Title>,
    React.ComponentPropsWithoutRef<typeof SheetPrimitive.Title>
  >(({ className, ...props }, ref) => (
    <SheetPrimitive.Title
      ref={ref}
      className={cn("text-lg font-semibold text-zinc-950 dark:text-zinc-50",
className)}
      {...props}
    />
  ))
  SheetTitle.displayName = SheetPrimitive.Title.displayName

  const SheetDescription = React.forwardRef<
    React.ElementRef<typeof SheetPrimitive.Description>,
    React.ComponentPropsWithoutRef<typeof SheetPrimitive.Description>
  >(({ className, ...props }, ref) => (
    <SheetPrimitive.Description
      ref={ref}
      className={cn("text-sm text-zinc-500 dark:text-zinc-400", className)}
      {...props}
    />
  ))
  SheetDescription.displayName = SheetPrimitive.Description.displayName

  export {
    Sheet,
    SheetPortal,
    SheetOverlay,
    SheetTrigger,
    SheetClose,
    SheetContent,
    SheetHeader,
    SheetFooter,
    SheetTitle,
    SheetDescription,
  }

```

```

import * as React from "react"

import { cn } from "@lib/utils"

const Table = React.forwardRef<
  HTMLTableElement,
  React.HTMLAttributes<HTMLTableElement>
>(({ className, ...props }, ref) => (
  <div className="relative w-full overflow-auto">
    <table
      ref={ref}
      className={cn("w-full caption-bottom text-sm", className)}
      {...props}
    />
  </div>
))
Table.displayName = "Table"

const TableHeader = React.forwardRef<
  HTMLTableSectionElement,
  React.HTMLAttributes<HTMLTableSectionElement>
>(({ className, ...props }, ref) => (
  <thead ref={ref} className={cn("[&_tr]:border-b", className)} {...props} />
))
TableHeader.displayName = "TableHeader"

const TableBody = React.forwardRef<
  HTMLTableSectionElement,
  React.HTMLAttributes<HTMLTableSectionElement>
>(({ className, ...props }, ref) => (
  <tbody
    ref={ref}
    className={cn("[&_tr:last-child]:border-0", className)}
    {...props}
  />
))
TableBody.displayName = "TableBody"

const TableFooter = React.forwardRef<
  HTMLTableSectionElement,
  React.HTMLAttributes<HTMLTableSectionElement>
>(({ className, ...props }, ref) => (

```

```

<tfoot
  ref={ref}
  className={cn(
    "border-t bg-zinc-100/50 font-medium [>tr]:last:border-b-0
dark:bg-zinc-800/50",
    className
  )}
  {...props}
/>
))
TableFooter.displayName = "TableFooter"

const TableRow = React.forwardRef<
  HTMLTableRowElement,
  React.HTMLAttributes<HTMLTableRowElement>
>(({ className, ...props }, ref) => (
  <tr
    ref={ref}
    className={cn(
      "border-b transition-colors hover:bg-zinc-100/50
data-[state=selected]:bg-zinc-100 dark:hover:bg-zinc-800/50
dark:data-[state=selected]:bg-zinc-800",
      className
    )}
    {...props}
  />
))
TableRow.displayName = "TableRow"

const TableHead = React.forwardRef<
  HTMLTableCellElement,
  React.ThHTMLAttributes<HTMLTableCellElement>
>(({ className, ...props }, ref) => (
  <th
    ref={ref}
    className={cn(
      "h-10 px-2 text-left align-middle font-medium text-zinc-500
[&:has([role=checkbox]):pr-0 [>[role=checkbox]]:translate-y-[2px]
dark:text-zinc-400",
      className
    )}
    {...props}

```

```

    />
  ))
  TableHead.displayName = "TableHead"

  const TableCell = React.forwardRef<
    HTMLTableCellElement,
    React.TdHTMLAttributes<HTMLTableCellElement>
  >(({ className, ...props }, ref) => (
    <td
      ref={ref}
      className={cn(
        "p-2 align-middle [&:has([role=checkbox])]:pr-0
        [&[role=checkbox]]:translate-y-[2px]",
        className
      )}
      {...props}
    />
  ))
  TableCell.displayName = "TableCell"

  const TableCaption = React.forwardRef<
    HTMLTableCaptionElement,
    React.HTMLAttributes<HTMLTableCaptionElement>
  >(({ className, ...props }, ref) => (
    <caption
      ref={ref}
      className={cn("mt-4 text-sm text-zinc-500 dark:text-zinc-400", className)}
      {...props}
    />
  ))
  TableCaption.displayName = "TableCaption"

  export {
    Table,
    TableHeader,
    TableBody,
    TableFooter,
    TableHead,
    TableRow,
    TableCell,
    TableCaption,
  }

```

```

import { Moon, Sun } from "lucide-react";

import { Button } from "@/components/ui/button";
import {
  DropdownMenu,
  DropdownMenuContent,
  DropdownMenuItem,
  DropdownMenuTrigger,
} from "@/components/ui/dropdown-menu";
import { useTheme } from "@/components/theme-provider";

export function ModeToggle() {
  const { setTheme } = useTheme();

  return (
    <DropdownMenu>
      <DropdownMenuTrigger asChild>
        <Button variant="outline" size="icon">
          <Sun className="h-[1.2rem] w-[1.2rem] rotate-0 scale-100
transition-all dark:-rotate-90 dark:scale-0" />
          <Moon className="absolute h-[1.2rem] w-[1.2rem] rotate-90 scale-0
transition-all dark:rotate-0 dark:scale-100" />
          <span className="sr-only">Toggle theme</span>
        </Button>
      </DropdownMenuTrigger>
      <DropdownMenuContent align="end">
        <DropdownMenuItem onClick={() => setTheme("light")}>
          Light
        </DropdownMenuItem>
        <DropdownMenuItem onClick={() => setTheme("dark")}>
          Dark
        </DropdownMenuItem>
        <DropdownMenuItem onClick={() => setTheme("system")}>
          System
        </DropdownMenuItem>
      </DropdownMenuContent>
    </DropdownMenu>
  );
}

```

```

import { Link, Outlet } from "react-router-dom";
import { Sheet, SheetContent, SheetTrigger } from "../ui/sheet";
import { CircleUser, Menu, Package2, Search } from "lucide-react";
import { Button } from "../ui/button";

import { Input } from "../ui/input";
import {
  DropdownMenu,
  DropdownMenuContent,
  DropdownMenuItem,
  DropdownMenuLabel,
  DropdownMenuSeparator,
  DropdownMenuTrigger,
} from "../ui/dropdown-menu";

const NavBar = () => {
  return (
    <div className="flex min-h-screen w-full flex-col">
      <header className="sticky top-0 flex h-16 items-center gap-4 border-b
bg-background px-4 md:px-6 dark:border-b-zinc-700 z-50">
        <nav className="hidden flex-col gap-6 text-lg font-medium md:flex
md:flex-row md:items-center md:gap-5 md:text-sm lg:gap-6">
          <Link
            to="#"
            className="flex items-center gap-2 text-lg font-semibold
md:text-base"
          >
            <Package2 className="h-6 w-6" />
            <span className="sr-only">Politrium Limited</span>
          </Link>
          <Link
            to="/dashboard"
            className="text-foreground transition-colors hover:text-foreground"
          >
            Главная
          </Link>
          <Link
            to="#"
            className="text-muted-foreground transition-colors
hover:text-foreground"
          >
            Заказы

```



```

        </Link>
        <Link
            to="/dashboard/products"
            className="text-muted-foreground transition-colors
hover:text-foreground"
        >
            Товары
        </Link>
        <Link
            to="#"
            className="text-muted-foreground transition-colors
hover:text-foreground"
        >
            Клиенты
        </Link>
        <Link
            to="#"
            className="text-muted-foreground transition-colors
hover:text-foreground"
        >
            Аналитика
        </Link>
    </nav>
    <Sheet>
        <SheetTrigger asChild>
            <Button
                variant="outline"
                size="icon"
                className="shrink-0 md:hidden"
            >
                <Menu className="h-5 w-5" />
                <span className="sr-only">Открыть меню навигации</span>
            </Button>
        </SheetTrigger>
        <SheetContent side="left">
            <nav className="grid gap-6 text-lg font-medium">
                <Link
                    to="#"
                    className="flex items-center gap-2 text-lg font-semibold"
                >
                    <Package2 className="h-6 w-6" />
                    <span className="sr-only">Politrium Limited</span>

```

```

        </Link>
        <Link to="/dashboard" className="hover:text-foreground">
            Главная
        </Link>
        <Link
            to="#"
            className="text-muted-foreground hover:text-foreground"
        >
            Заказы
        </Link>
        <Link
            to="/dashboard/products"
            className="text-muted-foreground hover:text-foreground"
        >
            Товары
        </Link>
        <Link
            to="#"
            className="text-muted-foreground hover:text-foreground"
        >
            Клиенты
        </Link>
        <Link
            to="#"
            className="text-muted-foreground hover:text-foreground"
        >
            Аналитика
        </Link>
    </nav>
</SheetContent>
</Sheet>
<div className="flex w-full items-center gap-4 md:ml-auto md:gap-2 lg:gap-4">
    <form className="ml-auto flex-1 sm:flex-initial">
        <div className="relative">
            <Search className="absolute left-2.5 top-2.5 h-4 w-4 text-muted-foreground" />
            <Input
                type="search"
                placeholder="Поиск товаров..."
                className="p1-8 sm:w-[300px] md:w-[200px] lg:w-[300px]"
            />

```

```

        </div>
      </form>
      <DropdownMenu>
        <DropdownMenuTrigger asChild>
          <Button variant="secondary" size="icon" className="rounded-full">
            <CircleUser className="h-5 w-5" />
            <span className="sr-only">Открыть меню пользователя</span>
          </Button>
        </DropdownMenuTrigger>
        <DropdownMenuContent align="end">
          <DropdownMenuLabel>Мой аккаунт</DropdownMenuLabel>
          <DropdownMenuSeparator />
          <DropdownMenuItem>Настройки</DropdownMenuItem>
          <DropdownMenuItem>Поддержка</DropdownMenuItem>
          <DropdownMenuSeparator />
          <DropdownMenuItem>Выход</DropdownMenuItem>
        </DropdownMenuContent>
      </DropdownMenu>
    </div>
  </header>
  <main className="flex flex-1 flex-col gap-4 p-4 md:gap-8 md:p-8">
    <Outlet />
  </main>
</div>
);
};

export default NavBar;

```

```

import { createContext, useContext, useEffect, useState } from "react";

type Theme = "dark" | "light" | "system";

type ThemeProviderProps = {
  children: React.ReactNode;
  defaultTheme?: Theme;
  storageKey?: string;
};

type ThemeProviderState = {
  theme: Theme;
  setTheme: (theme: Theme) => void;

```

```

});

const initialState: ThemeProviderState = {
  theme: "system",
  setTheme: () => null,
};

const ThemeProviderContext = createContext<ThemeProviderState>(initialState);

export function ThemeProvider({
  children,
  defaultTheme = "system",
  storageKey = "vite-ui-theme",
  ...props
}: ThemeProviderProps) {
  const [theme, setTheme] = useState<Theme>(
    () => (localStorage.getItem(storageKey) as Theme) || defaultTheme
  );

  useEffect(() => {
    const root = window.document.documentElement;

    root.classList.remove("light", "dark");

    if (theme === "system") {
      const systemTheme = window.matchMedia("(prefers-color-scheme: dark)")
        .matches
        ? "dark"
        : "light";

      root.classList.add(systemTheme);
      return;
    }

    root.classList.add(theme);
  }, [theme]);

  const value = {
    theme,
    setTheme: (theme: Theme) => {
      localStorage.setItem(storageKey, theme);
      setTheme(theme);
    }
  };

```

```

    },
  };

  return (
    <ThemeProviderContext.Provider {...props} value={value}>
      {children}
    </ThemeProviderContext.Provider>
  );
}

export const useTheme = () => {
  const context = useContext(ThemeProviderContext);

  if (context === undefined)
    throw new Error("useTheme must be used within a ThemeProvider");

  return context;
};

```

```

@tailwind base;
@tailwind components;
@tailwind utilities;

html.dark {
  @apply text-zinc-100 bg-[hsl(240_10%_3.9%)];
}

* {
  @apply select-none;
}

```

```

import { type ClassValue, clsx } from "clsx"
import { twMerge } from "tailwind-merge"

export function cn(...inputs: ClassValue[]) {
  return twMerge(clsx(inputs))
}

```

```

import { Button } from "@/components/ui/button";
import { Input } from "@/components/ui/input";

```

```
import { useForm } from "react-hook-form";
import { Link } from "react-router-dom";
import { z } from "zod";
import { zodResolver } from "@hookform/resolvers/zod";
import {
  Form,
  FormControl,
  FormDescription,
  FormField,
  FormItem,
  FormLabel,
  FormMessage,
} from "@components/ui/form";
import React from "react";
import {
  AlertDialog,
  AlertDialogAction,
  AlertDialogCancel,
  AlertDialogContent,
  AlertDialogDescription,
  AlertDialogFooter,
  AlertDialogHeader,
  AlertDialogTitle,
} from "@components/ui/alert-dialog";

const formSchema = z.object({
  email: z.string(),
  password: z.string(),
});

export function Authentication() {
  const [opened, setOpened] = React.useState(false);

  const form = useForm<z.infer<typeof formSchema>>>({
    resolver: zodResolver(formSchema),
    defaultValues: {
      email: "",
      password: "",
    },
  });

  function onSubmit(values: z.infer<typeof formSchema>) {
```

```

    console.log(values);

    setOpened(true);
  }

  return (
    <>
      <AlertDialog defaultOpen={false} open={opened} onOpenChange={setOpened}>
        <AlertDialogContent>
          <AlertDialogHeader>
            <AlertDialogTitle>Аутентификация</AlertDialogTitle>
            <AlertDialogDescription>
              Вы успешно акцентировались в системе. Нажимте продолжить что бы
              перейти к панели управления
            </AlertDialogDescription>
          </AlertDialogHeader>
          <AlertDialogFooter>
            <AlertDialogCancel>Отмена</AlertDialogCancel>
            <Link to="/dashboard">
              <AlertDialogAction>Продолжить</AlertDialogAction>
            </Link>
          </AlertDialogFooter>
        </AlertDialogContent>
      </AlertDialog>
      <div className="w-full lg:grid lg:min-h-[600px] lg:grid-cols-2
xl:min-h-[800px] h-screen">
        <div className="flex items-center justify-center py-12">
          <div className="mx-auto grid w-[350px] gap-6">
            <div className="grid gap-2 text-center">
              <h1 className="text-3xl font-bold">Аунтефикация</h1>
              <p className="text-balance text-muted-foreground">
                Введите ваш email ниже, чтобы войти в ваш аккаунт
              </p>
            </div>
            <Form {...form}>
              <form
                onSubmit={form.handleSubmit(onSubmit)}
                className="grid gap-4"
              >
                <FormField
                  control={form.control}
                  name="email"

```

```

        render={({ field }) => (
          <FormItem>
            <FormLabel>Email</FormLabel>
            <FormControl>
              <Input
                id="email"
                type="email"
                placeholder="m@example.com"
                required
                {...field}
              />
            </FormControl>
            <FormDescription>Введите ваш email.</FormDescription>
            <FormMessage />
          </FormItem>
        )}
      />
    <FormField
      control={form.control}
      name="password"
      render={({ field }) => (
        <FormItem>
          <FormLabel>Пароль</FormLabel>
          <FormControl>
            <Input
              id="password"
              type="password"
              required
              {...field}
            />
          </FormControl>
          <FormDescription>Введите ваш пароль.</FormDescription>
          <FormMessage />
        </FormItem>
      )}
    />
    <Button type="submit" className="w-full">
      Войти
    </Button>
    <Button variant="outline" className="w-full">
      Войти через Google
    </Button>

```



```

        </form>
      </Form>
      <div className="mt-4 text-center text-sm">
        Нет аккаунта?{" " }
        <Link to="#" className="underline">
          Зарегистрируйтесь
        </Link>
      </div>
    </div>
  </div>
  <div className="hidden bg-muted lg:block">
    
  </div>
</div>
</>
);
}

```

```

import {
  Activity,
  ArrowUpRight,
  CreditCard,
  DollarSign,
  Users,
} from "lucide-react";

import { Avatar, AvatarFallback, AvatarImage } from "@/components/ui/avatar";
import { Badge } from "@/components/ui/badge";
import { Button } from "@/components/ui/button";
import {
  Card,
  CardContent,
  CardDescription,
  CardHeader,
  CardTitle,

```

```

} from "@components/ui/card";

import {
  Table,
  TableBody,
  TableCell,
  TableHead,
  TableHeader,
  TableRow,
} from "@components/ui/table";
import { Link } from "react-router-dom";

export function Dashboard() {
  return (
    <>
      <div className="grid gap-4 md:grid-cols-2 md:gap-8 lg:grid-cols-4">
        <Card x-chunk="dashboard-01-chunk-0">
          <CardHeader className="flex flex-row items-center justify-between space-y-0 pb-2">
            <CardTitle className="text-sm font-medium">Общая
прибыль</CardTitle>
            <DollarSign className="h-4 w-4 text-muted-foreground" />
          </CardHeader>
          <CardContent>
            <div className="text-2xl font-bold">$45,231.89</div>
            <p className="text-xs text-muted-foreground">
              +20.1% с прошлого месяца
            </p>
          </CardContent>
        </Card>
        <Card x-chunk="dashboard-01-chunk-1">
          <CardHeader className="flex flex-row items-center justify-between space-y-0 pb-2">
            <CardTitle className="text-sm font-medium">Подписки</CardTitle>
            <Users className="h-4 w-4 text-muted-foreground" />
          </CardHeader>
          <CardContent>
            <div className="text-2xl font-bold">+2350</div>
            <p className="text-xs text-muted-foreground">
              +180.1% с прошлого месяца
            </p>
          </CardContent>
        </Card>
      </div>
    </>
  );
}

```

```

    </Card>
    <Card x-chunk="dashboard-01-chunk-2">
      <CardHeader className="flex flex-row items-center justify-between space-y-0 pb-2">
        <CardTitle className="text-sm font-medium">Продажи</CardTitle>
        <CreditCard className="h-4 w-4 text-muted-foreground" />
      </CardHeader>
      <CardContent>
        <div className="text-2xl font-bold">+12,234</div>
        <p className="text-xs text-muted-foreground">
          +19% с прошлого месяца
        </p>
      </CardContent>
    </Card>
    <Card x-chunk="dashboard-01-chunk-3">
      <CardHeader className="flex flex-row items-center justify-between space-y-0 pb-2">
        <CardTitle className="text-sm font-medium">
          Активны сейчас
        </CardTitle>
        <Activity className="h-4 w-4 text-muted-foreground" />
      </CardHeader>
      <CardContent>
        <div className="text-2xl font-bold">+573</div>
        <p className="text-xs text-muted-foreground">
          +201 за последний час
        </p>
      </CardContent>
    </Card>
  </div>
  <div className="grid gap-4 md:gap-8 lg:grid-cols-2 xl:grid-cols-3">
    <Card className="xl:col-span-2" x-chunk="dashboard-01-chunk-4">
      <CardHeader className="flex flex-row items-center">
        <div className="grid gap-2">
          <CardTitle>Транзакции</CardTitle>
          <CardDescription>
            Недавние транзакции из вашего магазина.
          </CardDescription>
        </div>
        <Button asChild size="sm" className="ml-auto gap-1">
          <Link to="#">
            Просмотреть все
          </Link>
        </Button>
      </CardHeader>
    </Card>
  </div>

```

```

        <ArrowUpRight className="h-4 w-4" />
      </Link>
    </Button>
  </CardHeader>
  <CardContent>
    <Table>
      <TableHeader>
        <TableRow>
          <TableHead>Покупатель</TableHead>
          <TableHead className="hidden xl:table-column">Тип</TableHead>
          <TableHead className="hidden xl:table-column">
            Статус
          </TableHead>
          <TableHead className="hidden
xl:table-column">Дата</TableHead>
          <TableHead className="text-right">Сумма</TableHead>
        </TableRow>
      </TableHeader>
      <TableBody>
        <TableRow>
          <TableCell>
            <div className="font-medium">Лиам Джонсон</div>
            <div className="hidden text-sm text-muted-foreground
md:inline">
              liam@example.com
            </div>
          </TableCell>
          <TableCell className="hidden xl:table-column">
            Продажа
          </TableCell>
          <TableCell className="hidden xl:table-column">
            <Badge className="text-xs" variant="outline">
              Одобрено
            </Badge>
          </TableCell>
          <TableCell className="hidden md:table-cell lg:hidden
xl:table-column">
            2023-06-23
          </TableCell>
          <TableCell className="text-right">$250.00</TableCell>
        </TableRow>
        <TableRow>

```

```

        <TableCell>
          <div className="font-medium">Оливия Смит</div>
          <div className="hidden text-sm text-muted-foreground
md:inline">
            olivia@example.com
          </div>
        </TableCell>
        <TableCell className="hidden xl:table-column">
          Возврат
        </TableCell>
        <TableCell className="hidden xl:table-column">
          <Badge className="text-xs" variant="outline">
            Отклонено
          </Badge>
        </TableCell>
        <TableCell className="hidden md:table-cell lg:hidden
xl:table-column">
          2023-06-24
        </TableCell>
        <TableCell className="text-right">$150.00</TableCell>
      </TableRow>
      <TableRow>
        <TableCell>
          <div className="font-medium">Ноах Вильямс</div>
          <div className="hidden text-sm text-muted-foreground
md:inline">
            noah@example.com
          </div>
        </TableCell>
        <TableCell className="hidden xl:table-column">
          Подписка
        </TableCell>
        <TableCell className="hidden xl:table-column">
          <Badge className="text-xs" variant="outline">
            Одобрено
          </Badge>
        </TableCell>
        <TableCell className="hidden md:table-cell lg:hidden
xl:table-column">
          2023-06-25
        </TableCell>
        <TableCell className="text-right">$350.00</TableCell>

```

```

</TableRow>
<TableRow>
  <TableCell>
    <div className="font-medium">Эмма Браун</div>
    <div className="hidden text-sm text-muted-foreground
md:inline">
      emma@example.com
    </div>
  </TableCell>
  <TableCell className="hidden xl:table-column">
    Продажа
  </TableCell>
  <TableCell className="hidden xl:table-column">
    <Badge className="text-xs" variant="outline">
      Одобрено
    </Badge>
  </TableCell>
  <TableCell className="hidden md:table-cell lg:hidden
xl:table-column">
    2023-06-26
  </TableCell>
  <TableCell className="text-right">$450.00</TableCell>
</TableRow>
<TableRow>
  <TableCell>
    <div className="font-medium">Лиам Джонсон</div>
    <div className="hidden text-sm text-muted-foreground
md:inline">
      liam@example.com
    </div>
  </TableCell>
  <TableCell className="hidden xl:table-column">
    Продажа
  </TableCell>
  <TableCell className="hidden xl:table-column">
    <Badge className="text-xs" variant="outline">
      Одобрено
    </Badge>
  </TableCell>
  <TableCell className="hidden md:table-cell lg:hidden
xl:table-column">
    2023-06-27

```

```

        </TableCell>
        <TableCell className="text-right">$550.00</TableCell>
    </TableRow>
</TableBody>
</Table>
</CardContent>
</Card>
<Card x-chunk="dashboard-01-chunk-5">
    <CardHeader>
        <CardTitle>Последние продажи</CardTitle>
    </CardHeader>
    <CardContent className="grid gap-8">
        <div className="flex items-center gap-4">
            <Avatar className="hidden h-9 w-9 sm:flex">
                <AvatarImage src="/avatars/01.png" alt="Аватар" />
                <AvatarFallback>ОМ</AvatarFallback>
            </Avatar>
            <div className="grid gap-1">
                <p className="text-sm font-medium leading-none">
                    Оливия Мартин
                </p>
                <p className="text-sm text-muted-foreground">
                    olivia.martin@email.com
                </p>
            </div>
            <div className="ml-auto font-medium">+$1,999.00</div>
        </div>
        <div className="flex items-center gap-4">
            <Avatar className="hidden h-9 w-9 sm:flex">
                <AvatarImage src="/avatars/02.png" alt="Аватар" />
                <AvatarFallback>ЖЛ</AvatarFallback>
            </Avatar>
            <div className="grid gap-1">
                <p className="text-sm font-medium leading-none">Джейсон Ли</p>
                <p className="text-sm text-muted-foreground">
                    jackson.lee@email.com
                </p>
            </div>
            <div className="ml-auto font-medium">+$39.00</div>
        </div>
        <div className="flex items-center gap-4">
            <Avatar className="hidden h-9 w-9 sm:flex">

```

```

        <AvatarImage src="/avatars/03.png" alt="Аватар" />
        <AvatarFallback>ИИ</AvatarFallback>
    </Avatar>
    <div className="grid gap-1">
        <p className="text-sm font-medium leading-none">
            Иcabella Hryen
        </p>
        <p className="text-sm text-muted-foreground">
            isabella.nguyen@email.com
        </p>
    </div>
    <div className="ml-auto font-medium">+$299.00</div>
</div>
<div className="flex items-center gap-4">
    <Avatar className="hidden h-9 w-9 sm:flex">
        <AvatarImage src="/avatars/04.png" alt="Аватар" />
        <AvatarFallback>БК</AvatarFallback>
    </Avatar>
    <div className="grid gap-1">
        <p className="text-sm font-medium leading-none">Вилли Ким</p>
        <p className="text-sm text-muted-foreground">will@email.com</p>
    </div>
    <div className="ml-auto font-medium">+$99.00</div>
</div>
<div className="flex items-center gap-4">
    <Avatar className="hidden h-9 w-9 sm:flex">
        <AvatarImage src="/avatars/05.png" alt="Аватар" />
        <AvatarFallback>СД</AvatarFallback>
    </Avatar>
    <div className="grid gap-1">
        <p className="text-sm font-medium leading-none">София
Дэвис</p>
        <p className="text-sm text-muted-foreground">
            sofia.davis@email.com
        </p>
    </div>
    <div className="ml-auto font-medium">+$39.00</div>
</div>
</CardContent>
</Card>
</div>
</>

```



```
);  
}
```

```
import { Button } from "@components/ui/button";  
import { Card, CardContent } from "@components/ui/card";  
import {  
  Carousel,  
  CarouselContent,  
  CarouselItem,  
  CarouselNext,  
  CarouselPrevious,  
} from "@components/ui/carousel";  
import React from "react";  
  
const Product = ({  
  title,  
  description,  
  price,  
  inStock,  
  images,  
}): {  
  title: string;  
  description: string;  
  price: number;  
  inStock: boolean;  
  images: string[];  
}) => {  
  return (  
    <div className="flex flex-row gap-[3rem]">  
      <div className="flex flex-col rounded-lg shadow-lg max-h-[800px]  
aspect-square">  
        <div className="flex flex-col">  
          <Carousel>  
            <CarouselContent>  
              {images.map((src, idx) => (  
                <CarouselItem key={idx}>  
                  <div className="p-1">  
                    <Card>  
                      <CardContent className="flex aspect-square items-center  
justify-center p-6">  
                        <img  
                          key={idx}
```

```

                src={src}
                alt={` ${title} image ${idx}`}
                height={250}
                width={250}
            />
        </CardContent>
    </Card>
</div>
</CarouselItem>
    )))
</CarouselContent>
<CarouselPrevious />
<CarouselNext />
</Carousel>
</div>
</div>
<div className="flex flex-col p-4">
    <div className="text-xl font-bold">{title}</div>
    <div className="text-sm text-gray-500">{description}</div>
    <div className="text-sm font-semibold">${price}</div>
    {inStock && <div className="text-green-500">In stock</div>}
    {!inStock && <div className="text-red-500">Out of stock</div>}
</div>
</div>
);
};

const ProductsPage = () => {
    const products = [
        {
            title: "Vivo Phone",
            description: "Modern phone with high-quality camera and fast
performance",
            price: 1000,
            inStock: true,
            images: ["/vivo-1.jpg", "/vivo-2.jpg", "/vivo-3.jpg"],
        },
        {
            title: "Product 2",
            description: "Modern phone with high-quality camera and fast
performance",
            price: 2000,

```

```
    inStock: false,
    images: [
      "https://picsum.photos/200/300",
      "https://picsum.photos/200/300",
      "https://picsum.photos/200/300",
      "https://picsum.photos/200/300",
    ],
  },
  {
    title: "Product 3",
    description: "Modern phone with high-quality camera and fast
performance",
    price: 1500,
    inStock: true,
    images: [
      "https://picsum.photos/200/300",
      "https://picsum.photos/200/300",
      "https://picsum.photos/200/300",
    ],
  },
  {
    title: "Product 4",
    description: "Modern phone with high-quality camera and fast
performance",
    price: 800,
    inStock: true,
    images: [
      "https://picsum.photos/200/300",
      "https://picsum.photos/200/300",
      "https://picsum.photos/200/300",
    ],
  },
  {
    title: "Product 5",
    description: "Modern phone with high-quality camera and fast
performance",
    price: 10,
    inStock: true,
    images: [
      "https://picsum.photos/200/300",
      "https://picsum.photos/200/300",
      "https://picsum.photos/200/300",
    ],
  },
}
```

```

    ],
  },
];

const [currentProductIndex, setCurrentProductIndex] = React.useState(0);

const nextProduct = () => {
  setCurrentProductIndex((prevIndex) => (prevIndex + 1) % products.length);
};

const prevProduct = () => {
  setCurrentProductIndex((prevIndex) =>
    prevIndex === 0 ? products.length - 1 : prevIndex - 1
  );
};

const currentProduct = products[currentProductIndex];

return (
  <div className="flex flex-col items-center h-full gap-4">
    <div className="max-h-screen">
      <Product {...currentProduct} />
    </div>
    <div className="flex gap-4 mt-4">
      <Button onClick={prevProduct}>Предыдущий товар</Button>
      <Button onClick={nextProduct}>Следующий товар</Button>
    </div>
  </div>
);
};

export default ProductsPage;

```

```

import { Dashboard } from "../routes/dashboard";
import {
  Navigate,
  RouterProvider,
  createBrowserRouter,
} from "react-router-dom";
import { ThemeProvider } from "../components/theme-provider";
import { Authentication } from "../routes/authentication";

```

```

import "@core/styles/globals.css";
import ProductsPage from "../routes/products";
import NavBar from "../components/nav-bar";

const router = createBrowserRouter([
  {
    path: "/dashboard",
    element: <NavBar />,
    children: [
      {
        index: true,
        element: <Dashboard />,
      },
      {
        path: "/dashboard/products",
        element: <ProductsPage />,
      },
    ],
  },
  {
    path: "/",
    element: <Authentication />,
  },
  {
    path: "*",
    element: <Navigate to="/dashboard" />,
  },
]);

function App() {
  return (
    <ThemeProvider defaultTheme="dark" storageKey="vite-ui-theme">
      <RouterProvider router={router} />
    </ThemeProvider>
  );
}

export default App;

```

```

import ReactDOM from "react-dom/client";
import App from "../App.tsx";

```

```
ReactDOM.createRoot(document.getElementById("root")!).render(<App />);
```

```
module.exports = {
  root: true,
  env: { browser: true, es2020: true },
  extends: [
    'eslint:recommended',
    'plugin:@typescript-eslint/recommended',
    'plugin:react-hooks/recommended',
  ],
  ignorePatterns: ['dist', '.eslintrc.cjs'],
  parser: '@typescript-eslint/parser',
  plugins: ['react-refresh'],
  rules: {
    'react-refresh/only-export-components': [
      'warn',
      { allowConstantExport: true },
    ],
  },
}
```

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" type="image/svg+xml" href="/vite.svg" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>LMS</title>
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/main.tsx"></script>
  </body>
</html>
```

```
/** @type {import('tailwindcss').Config} */
module.exports = {
  darkMode: ["class"],
  content: [
    './pages/**/*.{ts,tsx}',
  ],
}
```

```

'./components/**/*.{ts,tsx}',
'./app/**/*.{ts,tsx}',
'./src/**/*.{ts,tsx}',
],
prefix: "",
theme: {
  container: {
    center: true,
    padding: "2rem",
    screens: {
      "2xl": "1400px",
    },
  },
},
extend: {
  keyframes: {
    "accordion-down": {
      from: { height: "0" },
      to: { height: "var(--radix-accordion-content-height)" },
    },
    "accordion-up": {
      from: { height: "var(--radix-accordion-content-height)" },
      to: { height: "0" },
    },
  },
  animation: {
    "accordion-down": "accordion-down 0.2s ease-out",
    "accordion-up": "accordion-up 0.2s ease-out",
  },
},
plugins: [require("tailwindcss-animate")],
}

```

## Вывод:

Было спроектировано и разработано SPA приложение с использованием библиотеки React и react-router-dom.

На данный момент реализованы базовые компоненты. Подключен роутинг и сделаны несколько основных страниц без серверной логики приложения. В дальнейшем планируется добавить серверные сценарии для обработки пользовательских запросов и пересобрать приложения из SPA в SSR fullstack приложение с применением META-фреймворка Remix.