This is the Cart.tsx file, which is to handle and keep track of orders (not fully implemented).

```
"use client";

import { Box, Button, Icon } from "@chakra-ui/react";
import { useState } from "react";
import { CiShoppingCart } from "react-icons/ci";
import CartDialog from "@/components/ui/cart/cartDialog";

const Cart = () => {
  const [DialogVisible, setDialogVisible] = useState(false);
  const handleToggleDialog = () => setDialogVisible(!DialogVisible);

  return (
    <Box>
      <Button
        mx={2}
        aria-label="view-cart"
        onClick={handleToggleDialog}
        bg="orange.500"
        _hover={{ bg: "orange.600" }}
      >
        <Icon>
          <CiShoppingCart size="1.5em" />
        </Icon>
      </Button>
      <CartDialog
        cartItems={[]}
        visible={DialogVisible}
        onClose={handleToggleDialog}
      />
    </Box>
  );
};
export default Cart;
```

This is the nested CartDialog component, which does the actual handling and what to display. If the cart is empty a dialog box will be displayed instead of the list of orders.

```
import { Box, Text, Dialog, Portal, Button, Link } from "@chakra-ui/react";

type CartDialogProps = {
  cartItems: any[];
  visible: boolean;
  onClose: () => void;
};

const CartDialog = ({ cartItems, visible, onClose }: CartDialogProps) => {
  const dialogStyles = {
    maxW: "600px",
    width: "100%",
    position: "relative",
    p: 4,
    margin: "0 auto",
  };

  const exitButtonStyles = {
    position: "absolute",
    top: 4,
    right: 4,
  };

  if (!visible) return null;

  return (
    <Box>
      <Dialog.Root
        size={{ mdDown: "full", md: "lg" }}
        open={visible}
        placement="center"
      >
        <Portal>
          <Dialog.Backdrop>
            <Dialog.Positioner css={dialogStyles}>
              <Dialog.Content>
                <Dialog.Header>
                  <Dialog.Title textStyle="lg">Cart</Dialog.Title>
                  <Button onClick={onClose} css={exitButtonStyles}>
                    X
                  </Button>
            </Dialog.Header>
            <Dialog.Body>
              {cartItems.length == 0 ? (
                <Box textAlign="center" py={4} px={4}>
                  <Text textStyle="lg" mb={4}>
                    Looks like your cart is empty.
                  </Text>
                  <Link
                    href="/menu"
                    color="black"
                    style={{ textDecoration: "none" }}
                  >
                    <Button>Start Your Order</Button>
                  </Link>
                </Box>
              ) : (
                <Box>{/* Cart items would go here */}</Box>
              )}
            </Dialog.Body>
          </Dialog.Content>
        </Dialog.Positioner>
      </Dialog.Backdrop>
    </Portal>
  </Dialog.Root>
</Box>
 );
};

export default CartDialog;
```

This is the login page to handle sign up and sign ins. (Not fully implemented yet).

```
import {
  Container,
  Flex,
  Box,
  Stack,
  Heading,
  Text,
  Field,
  Checkbox,
  Input,
  Link,
  Button,
} from "@chakra-ui/react";
import type { Metadata } from "next";

export const metadata: Metadata = {
  title: "Login",
};

const Login = () => {
  return (
    <Container maxW="lg" py={12}>
      <Flex align="center" justify="center">
        <Box
          w="full"
          p={{ base: 6, md: 8 }}
          borderWidth="1px"
          borderRadius="lg"
          boxShadow="sm"
          bg="white"
          borderColor="gray.200"
        >
          <Stack gap={6}>
            <Stack gap={1} textAlign="center">
              <Heading size="lg" color="gray.600">
                Welcome back
              </Heading>
              <Text color="gray.600">Sign in to your account</Text>
            </Stack>

            <Stack as="form" gap={4}>
              <Field.Root>
  <Field.Label htmlFor="email" color="gray.600">
    <Field.RequiredIndicator />
    Email address
  </Field.Label>
  <Input
    id="email"
    type="email"
    placeholder="you@example.com"
    bg="gray.900"
    color="white"
    borderColor="white"
    _placeholder={{ color: "whiteAlpha.700" }}
    _hover={{ borderColor: "white" }}
    _focusVisible={{ borderColor: "white", boxShadow: "none" }}
  />
</Field.Root>

<Field.Root>
  <Field.Label htmlFor="password" color="gray.600">
    <Field.RequiredIndicator />
    Password
  </Field.Label>
  <Input
    id="password"
    type="password"
    placeholder="••••••••"
    bg="gray.900"
    color="white"
    borderColor="white"
    _placeholder={{ color: "whiteAlpha.700" }}
    _hover={{ borderColor: "white" }}
    _focusVisible={{ borderColor: "white", boxShadow: "none" }}
  />
</Field.Root>

<Flex justify="space-between" align="center">
  <Checkbox.Root color="gray.600">
    <Checkbox.HiddenInput id="remember" />
    <Checkbox.Control />
    <Checkbox.Label>Remember me</Checkbox.Label>
  </Checkbox.Root>
  <Link href="#" color="blue.500" fontWeight="semibold">
    Forgot password?
  </Link>
```

```
              </Flex>

              <Button
                size="md"
                mt={4}
                type="submit"
                css={{ bg: "orange.500", _hover: { bg: "orange.600" } }}
              >
                Sign in
              </Button>
            </Stack>

            <Text fontSize="sm" color="gray.600" textAlign="center">
              Don&apos;t have an account?{" "}
              <Link href="#" color="blue.500" fontWeight="semibold">
                Sign up
              </Link>
            </Text>
          </Stack>
        </Box>
      </Flex>
    </Container>
  );
};

export default Login;
```

This is the contact page for any feedback or inquiry in general.

```
import {
  Box,
  Text,
  Input,
  Button,
  Field,
  Fieldset,
  Stack,
  NativeSelect,
  For,
  Textarea,
} from "@chakra-ui/react";

const Contact = () => {
  const buttonStyle = {
    size: "md",
    bg: "orange.500",
    m: 2,
    _hover: { bg: "orange.600" },
    color: "white",
  };

  const mainStyle = {
    position: "relative",
    maxW: "800px",
    mx: "auto",
    py: 10,
    px: 4,
  };

  const formStyle = {
    mt: 6,
    mb: 4,
    gap: 4,
  };

  return (
    <Box as="main" role="contentinfo" {...mainStyle}>
      <form>
        <Fieldset.Root>
          <Stack>
            <Fieldset.Legend>Contact Us</Fieldset.Legend>
            <Text mb={4}>
              We would love to hear from you! Please fill out the form below and
              we will get back to you as soon as possible.
            </Text>
            <Fieldset.HelperText>Fill in the fields below</Fieldset.HelperText>
          </Stack>

          <Fieldset.Content css={formStyle}>
            <Field.Root>
              <Field.Label htmlFor="name">Name</Field.Label>
              <Input name="name" type="text" placeholder="Your Name" required />
            </Field.Root>

            <Field.Root>
              <Field.Label htmlFor="email">Email</Field.Label>
              <Input
                name="email"
                type="email"
                placeholder="Your Email"
                required
              />
            </Field.Root>

            <Field.Root>
              <Field.Label htmlFor="subject">Subject</Field.Label>
              <NativeSelect.Root>
                <NativeSelect.Field name="subject">
                  <For
                    each={[
                      "General Inquiry",
                      "Complaint",
                      "Compliment",
                      "Suggestion",
                      "Other",
                    ]}
                  >
                    {(item) => (
                      <option key={item} value={item}>
                        {item}
                      </option>
                    )}
                  </For>
                </NativeSelect.Field>
              </NativeSelect.Root>
            </Field.Root>
```

```
          <Field.Root>
            <Field.Label htmlFor="message">Message</Field.Label>
            <Textarea placeholder="Your message" required />
          </Field.Root>
        </Fieldset.Content>
        <Button type="submit" css={buttonStyle}>
          Submit
        </Button>
      </Fieldset.Root>
    </form>
  </Box>
 );
};

export default Contact;
```

This is the backend code for performing requests for Cart.

```
import {
  getCartController,
  putCartController,
  clearCartController,
} from "../../../lib/domains/cart/controller.js";

function cid(req) {
  return req.headers.get("x-customer-id");
}

export async function GET(req) {
  const customerId = cid(req);
  if (!customerId)
    return new Response(
      JSON.stringify({ ok: false, msg: "Missing x-customer-id" }),
      { status: 401 }
    );
  const out = await getCartController(customerId);
  return new Response(JSON.stringify(out), {
    status: 200,
    headers: { "Content-Type": "application/json" },
  });
}

export async function PUT(req) {
  const customerId = cid(req);
  if (!customerId)
    return new Response(
      JSON.stringify({ ok: false, msg: "Missing x-customer-id" }),
      { status: 401 }
    );
  let body;
  try {
    body = await req.json();
  } catch {
    return new Response(JSON.stringify({ ok: false, msg: "Invalid JSON" }), {
      status: 400,
    });
  }
  const { items = [], updatedAt: clientUpdatedAt } = body || {};
  try {
    const out = await putCartController(customerId, items, clientUpdatedAt);
    return new Response(JSON.stringify(out), {
      status: out.status || 200,
      headers: { "Content-Type": "application/json" },
    });
  } catch (e) {
    return new Response(JSON.stringify({ ok: false, msg: e.message }), {
      status: 400,
    });
  }
}

export async function DELETE(req) {
  const customerId = cid(req);
  if (!customerId)
    return new Response(
      JSON.stringify({ ok: false, msg: "Missing x-customer-id" }),
      { status: 401 }
    );
  const out = await clearCartController(customerId);
  return new Response(JSON.stringify(out), {
    status: 200,
    headers: { "Content-Type": "application/json" },
  });
}
```