Customer Relationship Management

Seshadri Rao Gudlavalleru Engineering College

Team Members

Yarroju Rudra Prakash

Lasya Galla

Anuhya Yaralagadda

Maddhali Rudra Sai Naga Venkata Sanjay Gupta

Overview

This document provides a detailed guide on setting up a CRM application using React. The application consists of the following main components:

Sidebar: For navigation

Dashboard: Overview of key metrics and activities

Inventory: Management of products and stock levels

Orders: Tracking and managing customer orders

Customers: Managing customer information and interactions

Table of Contents

1. Project Setup

   1. Prerequisites
   2. Create New Project
   3. Install Required Packages

2. Components

   1. App Footer Component
   2. App Header Component
   3. App Routes Component
   4. Page Content Component
   5. Side Menu Component
   6. App Component

3. Styling

4. Results

5. Conclusion

1. Project Setup

Prerequisites

- Node.js and npm installed

Create a New React Project

npx create-react-app react-crm

cd react-crm

npm start

Install Required Packages

npm install react-router-dom

2. Components

The Sidebar component serves as the primary navigation menu, providing links to all major sections of the CRM.

App Footer Component

```
import { Typography } from "antd";

function AppFooter() {
  return (
    <div className="AppFooter">
      <Typography.Link href="tel:+123456789">+123456789</Typography.Link>
      <Typography.Link href="https://www.google.com" target={"_blank"}>
        Privacy Policy
      </Typography.Link>
      <Typography.Link href="https://www.google.com" target={"_blank"}>
        Terms of Use
      </Typography.Link>
    </div>
  );
}
export default AppFooter;
```

App Header Component

```
import { BellFilled, MailOutlined } from "@ant-design/icons";
import { Badge, Drawer, Image, List, Space, Typography } from "antd";
import { useEffect, useState } from "react";
import { getComments, getOrders } from "../../API";

function AppHeader() {
  const [comments, setComments] = useState([]);
  const [orders, setOrders] = useState([]);
  const [commentsOpen, setCommentsOpen] = useState(false);
  const [notificationsOpen, setNotificationsOpen] = useState(false);

  useEffect(() => {
    getComments().then((res) => {
      setComments(res.comments);
    });
    getOrders().then((res) => {
      setOrders(res.products);
    });
  }, []);

  return (
    <div className="AppHeader">
      <Image
        width={40}
        src="https://www.shutterstock.com/image-vector/crm-letter-logo-design-on-600nw-2206999995.jpg"
      ></Image>
      <Typography.Title>CRM</Typography.Title>
      <Space>
        <Badge count={comments.length} dot>
          <MailOutlined
            style={{ fontSize: 24 }}
            onClick={() => {
              setCommentsOpen(true);
            }}
          />
        </Badge>
        <Badge count={orders.length}>
          <BellFilled
            style={{ fontSize: 24 }}
            onClick={() => {
              setNotificationsOpen(true);
            }}
          />
        </Badge>
      </Space>
      <Drawer
        title="Comments"
        open={commentsOpen}
        onClose={() => {
          setCommentsOpen(false);
        }}
        maskClosable
      >
        <List
          dataSource={comments}
          renderItem={(item) => {
            return <List.Item>{item.body}</List.Item>;
          }}
        ></List>
      </Drawer>
      <Drawer
        title="Notifications"
        open={notificationsOpen}
        onClose={() => {
          setNotificationsOpen(false);
        }}
        maskClosable
      >
        <List
          dataSource={orders}
          renderItem={(item) => {
            return (
              <List.Item>
                <Typography.Text strong>{item.title}</Typography.Text> has
been
                ordered!
              </List.Item>
            );
          }}
        ></List>
      </Drawer>
    </div>
  );
}
export default AppHeader;
```

App Routes Component

```
import { BrowserRouter, Route, Routes } from "react-router-dom";
import Customers from "../../Pages/Customers";
import Dashboard from "../../Pages/Dashbaord";
import Inventory from "../../Pages/Inventory";
import Orders from "../../Pages/Orders";

function AppRoutes() {
  return (
    <Routes>
      <Route path="/" element={<Dashboard />}></Route>
      <Route path="/inventory" element={<Inventory />}></Route>
      <Route path="/orders" element={<Orders />}></Route>
      <Route path="/customers" element={<Customers />}></Route>
    </Routes>
  );
}
export default AppRoutes;
```

Page Content Component

```
import AppRoutes from "../AppRoutes";

function PageContent() {
  return (
    <div className="PageContent">
      <AppRoutes />
    </div>
  );
}
export default PageContent;
```

Side Menu Component

```
import AppRoutes from "../AppRoutes";

function PageContent() {
  return (
    <div className="PageContent">
      <AppRoutes />
    </div>
  );
}
export default PageContent;
```

App.js

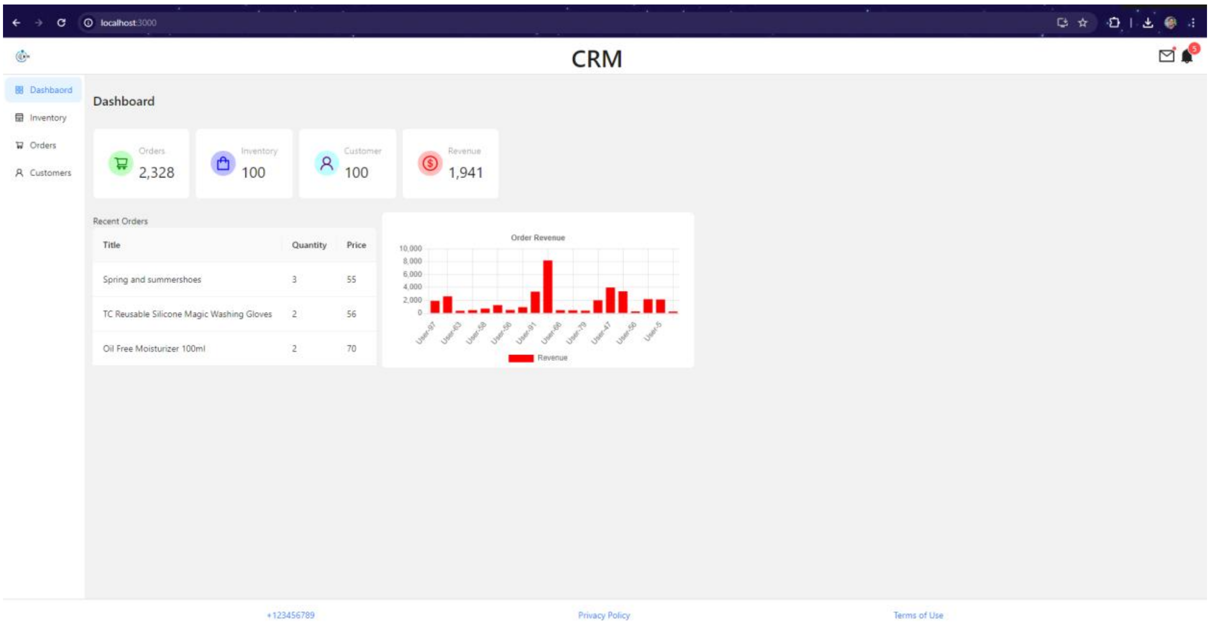
```
import { Space } from "antd";
import "./App.css";
import AppFooter from "./Components/AppFooter";
import AppHeader from "./Components/AppHeader";
import PageContent from "./Components/PageContent";
import SideMenu from "./Components/SideMenu";

function App() {
  return (
    <div className="App">
      <AppHeader />
      <div className="SideMenuAndPageContent">
        <SideMenu></SideMenu>
        <PageContent></PageContent>
      </div>
      <AppFooter />
    </div>
  );
}
export default App;
```

3. Styling

```
.App{
    display: flex;
    flex-direction: column;
    width: 100vw;
    height: 100vh;
}
.AppHeader{
    height: 50px;
    display: flex;
    justify-content: space-between;
    align-items: center;
    padding: 4px 20px 4px 12px;
    border-bottom: 1px solid rgba(0, 0, 0, 0.15);
}
.AppFooter{
height: 50px;
display: flex;
justify-content: space-evenly;
align-items: center;
border-top: 1px solid rgba(0, 0, 0, 0.15);
```

```css
}
.SideMenuAndPageContent{
    display: flex;
    flex: 1;
    justify-content: flex-start;
    align-items: flex-start;
    background-color: rgba(0, 0, 0, 0.05) ;
}
.SideMenu{
    height: 100%;
}
.SideMenuVertical{
    height: 100%;
}
.PageContent{
    padding-left: 12px;
}
```

4. Results



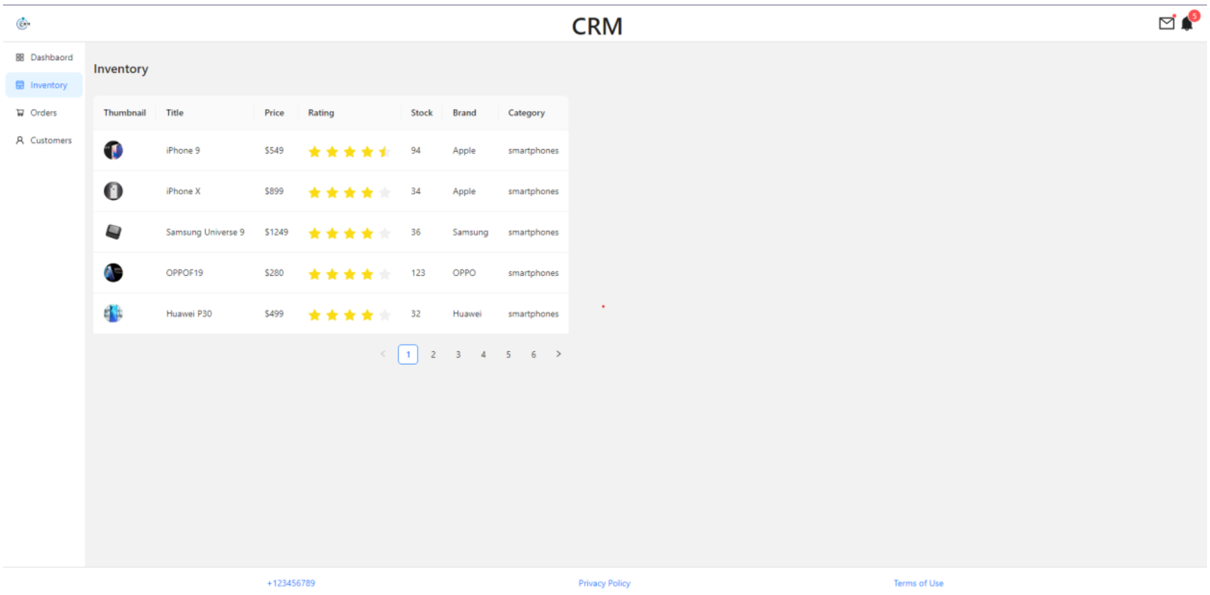
Dashboard

Orders	Inventory	Customer	Revenue
2,328	100	100	1,941

Recent Orders

Title	Quantity	Price
Spring and summershoes	3	55
TC Reusable Silicone Magic Washing Gloves	2	56
Oil Free Moisturizer 100ml	2	70

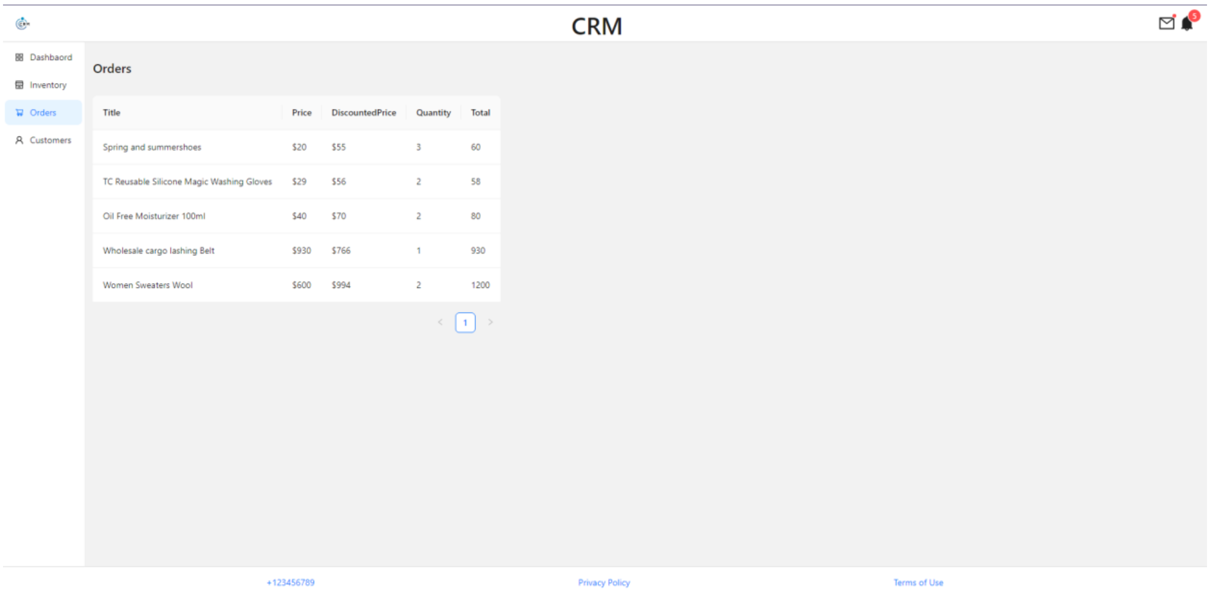
+123456789 Privacy Policy Terms of Use



Inventory

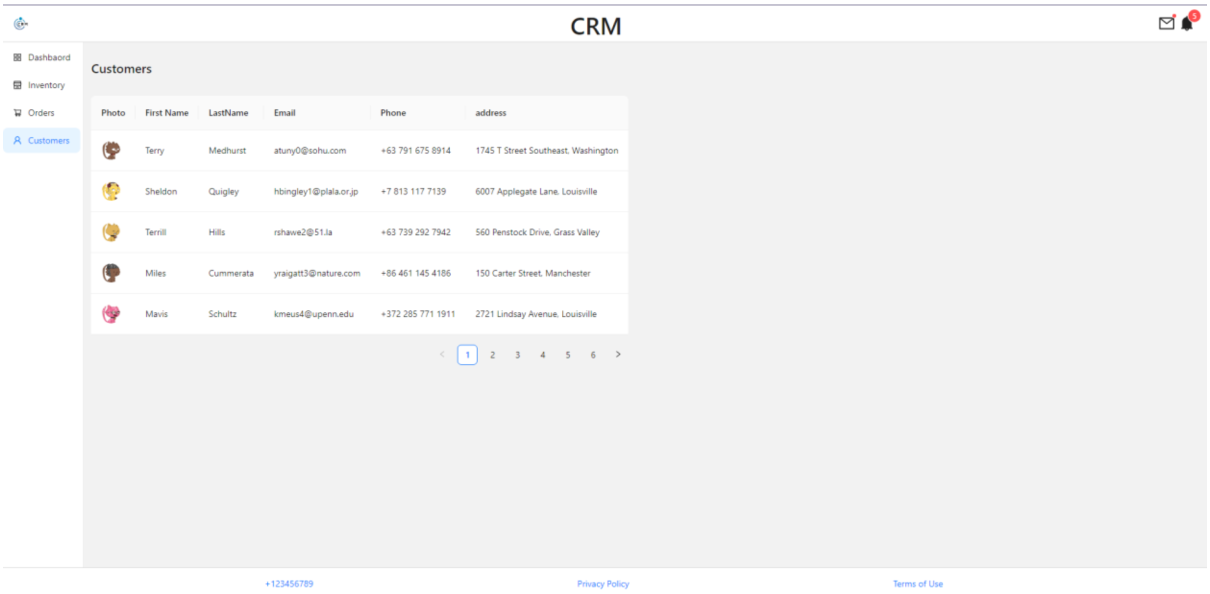
Thumbnail	Title	Price	Rating	Stock	Brand	Category
	iPhone 9	$549	★★★★⯨	94	Apple	smartphones
	iPhone X	$899	★★★★☆	34	Apple	smartphones
	Samsung Universe 9	$1249	★★★★☆	36	Samsung	smartphones
	OPPOF19	$280	★★★★☆	123	OPPO	smartphones
	Huawei P30	$499	★★★★☆	32	Huawei	smartphones

+123456789 Privacy Policy Terms of Use



Orders

Title	Price	DiscountedPrice	Quantity	Total
Spring and summershoes	$20	$55	3	60
TC Reusable Silicone Magic Washing Gloves	$29	$56	2	58
Oil Free Moisturizer 100ml	$40	$70	2	80
Wholesale cargo lashing Belt	$930	$766	1	930
Women Sweaters Wool	$600	$994	2	1200



Customers

Photo	First Name	LastName	Email	Phone	address
	Terry	Medhurst	atuny0@sohu.com	+63 791 675 8914	1745 T Street Southeast, Washington
	Sheldon	Quigley	hbingley1@plala.or.jp	+7 813 117 7139	6007 Applegate Lane, Louisville
	Terrill	Hills	rshawe2@51.la	+63 739 292 7942	560 Penstock Drive, Grass Valley
	Miles	Cummerata	yraigatt3@nature.com	+86 461 145 4186	150 Carter Street, Manchester
	Mavis	Schultz	kmeus4@upenn.edu	+372 285 771 1911	2721 Lindsay Avenue, Louisville

5. Conclusion

This projects provides a comprehensive guide to building a basic CRM system using React, covering project setup, component creation, routing configuration, and basic styling. You have established a modular architecture with key components: Sidebar, Dashboard, Inventory, Orders, and Customers, each tailored to specific functionalities. The Sidebar enables smooth navigation, the Dashboard offers an overview of key metrics, and the Inventory, Orders, and Customers components manage essential CRM data. Moving forward, you can enhance the CRM system by adding forms for data management, integrating state management solutions like Redux or Context API, and connecting to APIs for dynamic data fetching. This foundational setup equips you to build a scalable and feature-rich CRM application.