Imam Mohammad Bin Saud Islamic University

College of Computer and Information Sciences

**StoresHub**

**By:**

Hashem Bajabaa 437015783

Anas Alzhrani 438011286

Rayan Alshanabah 438012286

**Supervised by:**

Dr. Qaisar Abbas

Final report | first version

December 4, 2021 |28th Rabi' al-thani, 1443

Imam Mohammad Bin Saud Islamic University

College of Computer and Information Sciences

**StoresHub**

**By:**

Hashem Bajabaa 437015783

Anas Alzhrani 438011286

Rayan Alshanabah 438012286

We hereby certify that this project satisfies the project requirements:

|  |  |
| --- | --- |
| **Approved by** | **[Supervisor Name]** |
| **Date of Approval** |  |
| **Signature** |  |

|  |  |
| --- | --- |
| **Approved by** | **[Vice Dean of Educational Affairs Name]** |
| **Date of Approval** |  |
| **Signature** |  |

##### Declaration

We are Hashem Bajabaa, Anas Alzhrani, and Rayan Alshanabah being members of final year project group.

we, declare that this report contains only work completed by members of our group except for information obtained in a legitimate way from literature, company or university sources. All information from these other sources has been duly referenced and acknowledged in accordance with the University Policy on Plagiarism.

Furthermore, we declare that in completing the project, the individual group members had the following responsibilities and contributed in the following proportions

to the final outcomes of the project:

|  |  |  |  |
| --- | --- | --- | --- |
| **Student ID** | **Responsibility1** | **% Contributed2** | **Signature** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

##### Acknowledgment

First and foremost, we would like to present our deepest gratitude to Almighty

ALLAH for his bounties and blessings and for giving us the ability to finish this project

We would like to express our deep appreciation and our sincere gratitude to our supervisor Dr. Qaisar Abbas for his valuable advice guidance throughout this project.

Finally, we would like to thank our families and friends for continued encouragements and support during this project and along the years of study.

**Table of Contents**

[Abstract 1](#_Toc89546827)

[Abstract (in Arabic) 2](#_Toc89546828)

[Keywords 3](#_Toc89546829)

[List of abbreviation 4](#_Toc89546830)

[List of Figures 5](#_Toc89546831)

[List of Tables 6](#_Toc89546832)

[1.1 Introduction: 8](#_Toc89546833)

[1.2 Aims and Objectives: 9](#_Toc89546834)

[1.2.1 Aim: 9](#_Toc89546835)

[1.2.2 The objectives: 9](#_Toc89546836)

[1.3 Methodology: 10](#_Toc89546837)

[1.4 Project Timeline 11](#_Toc89546838)

[1.5 Team Qualifications 11](#_Toc89546839)

[1.6 Conclusion: 12](#_Toc89546840)

[2.1 Introduction 14](#_Toc89546841)

[2.2 Background 14](#_Toc89546842)

[2.2.1 Web application overview 15](#_Toc89546843)

[2.2.2 Web application 16](#_Toc89546844)

[2.2.3 websites 16](#_Toc89546845)

[2.2.4 E-commerce 18](#_Toc89546846)

[2.3 Related Work 19](#_Toc89546847)

[2.3.1 E-commerce 19](#_Toc89546848)

[2.3.2 Web Application 20](#_Toc89546849)

[2.4 Conclusion 22](#_Toc89546850)

[3.1 Introduction 24](#_Toc89546851)

[3.2 Purpose 24](#_Toc89546852)

[3.3 Definitions, Acronyms, and Abbreviations 25](#_Toc89546853)

[3.4 User Characteristics 25](#_Toc89546854)

[3.5 Product Perspective 26](#_Toc89546855)

[3.6 Product Functions 26](#_Toc89546856)

[3.7 General Constraints 27](#_Toc89546857)

[3.7.1 Interface Constraints: 27](#_Toc89546858)

[3.7.2 Technology Constraints: 27](#_Toc89546859)

[3.7.3 Privacy and Security Constraint 27](#_Toc89546860)

[3.8 Assumptions and Dependencies 27](#_Toc89546861)

[3.9 Interfaces 28](#_Toc89546862)

[3.9.1 User interface: 28](#_Toc89546863)

[3.9.2 Software interface: 28](#_Toc89546864)

[3.9.3 Hardware interface: 28](#_Toc89546865)

[3.10 User Requirements 29](#_Toc89546866)

[3.10.1 Functional Requirements 29](#_Toc89546867)

[3.10.2 Non-Functional Requirements 33](#_Toc89546868)

[3.11 UML Diagrams: 35](#_Toc89546869)

[3.11.1 Use-case Diagram 35](#_Toc89546870)

[3.11.2 Activity Diagram 44](#_Toc89546871)

[3.11.3 Class Diagram 46](#_Toc89546872)

[3.12 Conclusion 54](#_Toc89546873)

[4.1 Introduction 56](#_Toc89546874)

[4.2 System architecture 56](#_Toc89546875)

[4.3 Database Design 57](#_Toc89546876)

[4.3.1 Data dictionary 59](#_Toc89546877)

[4.4 Flow Chart 62](#_Toc89546878)

[4.5 System Organization 64](#_Toc89546879)

[4.5.1 Sequence diagram 64](#_Toc89546880)

[4.6 Design pattern 69](#_Toc89546881)

[4.7 Graphical User Interface Design 70](#_Toc89546882)

[4.8 Conclusion 73](#_Toc89546883)

[5.1 Introduction 75](#_Toc89546884)

[5.2 Implementation Requirements 75](#_Toc89546885)

[5.2.1 Hardware Requirements 75](#_Toc89546886)

[5.2.2 Software Requirements 75](#_Toc89546887)

[5.2.3 Programming Language(s) 76](#_Toc89546888)

[5.2.4 Tools and Technologies 77](#_Toc89546889)

[5.3 Code Implementation 77](#_Toc89546890)

[5.4 Conclusion 78](#_Toc89546891)

[6.1 Introduction 80](#_Toc89546892)

[6.2 Testing 80](#_Toc89546893)

[6.3 Conclusion 84](#_Toc89546894)

[7.1 Introduction 86](#_Toc89546895)

[7.2 Future Work 86](#_Toc89546896)

[7.3 Project Conclusion 87](#_Toc89546897)

[7.4 Conclusion 87](#_Toc89546898)

[Appendices 88](#_Toc89546899)

[References: 159](#_Toc89546900)

### Abstract

The e-commerce industry is developing rapidly around the world year after year [1]. Now days, it’s impossible to imagine the world without e-commerce, where consumers could buy whatever, they need anytime, anywhere conveniently. With large quantity of EC web apps appearance for each store, consumers have been distracted. Although to provide the stores owners the ability to host their products, people who would like to start a business and can’t afford the cost. Our platform will solve it all where, diverse products from various shops and dealers will be available. In order, to implement it, coding will take the largest effort for frontend and backend. Frontend require JavaScript, html, CSS, and react.js and python language for server side as backend. This document consists of three chapters, each chapter deals with part of the project and discuss it in detail. For the first chapter we will give an introduction of the project and proposal. Second chapter going to provide us with background knowledge to understand the topics problem, then an important insight from the related work will be covered. Lastly, we will extend the system and user characteristics and constrains, although a web application functional and non-functional requirement, followed by high- and low-level diagrams all in third chapter.

### Abstract (in Arabic)

تتطور صناعة التجارة الإلكترونية بسرعة في جميع أنحاء العالم عامًا بعد عام [1]. في الوقت الحاضر ، من المستحيل تخيل عالم خالٍ من التجارة الإلكترونية ، حيث يمكن للمستهلكين بسهولة شراء أي شيء يحتاجونه في أي وقت وفي أي مكان. مع ظهور عدد كبير من تطبيقات التجارة الإلكترونية على الويب لكل متجر ، تم تشتيت انتباه المستهلكين. على الرغم من تزويد أصحاب المتاجر بالقدرة على استضافة منتجاتهم ، إلا أن الأشخاص الذين يرغبون في بدء عمل تجاري ولا يمكنهم تحمل تكلفة ذلك. ستعمل منصتنا على حلها جميعًا حيث ستتوفر العديد من المنتجات من مختلف المتاجر والتجار. من أجل تنفيذه ، سيستغرق الترميز أقصى جهد لكل من الواجهة الأمامية والخلفية. تتطلب الواجهة الأمامية JavaScript و html و CSS و python لجانب الخادم كخلفية. تتكون هذه الوثيقة من ستة فصول ، كل فصل يتناول جزء من المشروع ومناقشته بالتفصيل. في الفصل الأول سوف نقدم مقدمة عن المشروع والاقتراح. سيوفر لنا الفصل الثاني المعرفة الأساسية لفهم مشكلة الموضوعات ، ومن ثم سيتم تغطية رؤى مهمة من العمل ذي الصلة. بعد ذلك ، سنتوسع في خصائص وقيود النظام والمستخدم ، على الرغم من المتطلبات الوظيفية وغير الوظيفية لتطبيق الويب ، متبوعة برسوم بيانية عالية ومنخفضة المستوى كلها في الفصل الثالث. بعد ذلك ، سنناقش بالتفصيل قياسات التصميم ونقدم صورًا بيانية متعلقة بالتصميم. يليه فصل خاص بالتنفيذ سيوفر الأساليب والأدوات التي نستخدمها. أخيرًا ، سنقدم خاتمة لمشروعنا وعملنا المستقبلي.

### Keywords

Web Application, E-Commerce, Front-End, Back-End, shopping Cart, Shipping, Website, Web server, JavaScript, HTML, CSS, React, Python, Django, Database, SEO, Customer.

### List of abbreviation

SQL: Structured Query Language

API: Application Programming Interface

HTML: Hypertext Markup Language

CSS: Cascading Style Sheets

OS: Operating System

B2B: Business to Business

B2C:Business to Customer

E-commerce: Electronic Commerce

SRS: Software Requirements Specification

UML: Unified Modeling Language

SPA: Single Page Architecture

ORM: Object-Relational Mapper

### List of Figures

[**Fig. ‎1.1‑1. shares of E-Commerce sales** 8](#_Toc89544852)

[**Fig. ‎1.4‑1. Project Management Plan** 11](#_Toc89544853)

[**Fig. ‎3.12‑1. 3.12.1 Use-case Diagram** 36](#_Toc89544854)

[**Fig. ‎3.12‑2. 3.12.2 Activity Diagram of Order** 44](#_Toc89544855)

[**Fig. ‎3.12‑3. 3.12.2 Activity Diagram of Customer Activities** 45](#_Toc89544856)

[**Fig. ‎3.12‑5. Class Diagram** 46](#_Toc89544857)

[**Fig. ‎4.2‑1 SPA** 57](#_Toc89544858)

[**Fig. ‎4.3‑1 Database table schema** 58](#_Toc89544859)

[**Fig. ‎4.4‑1Create Store** 62](#_Toc89544860)

[**Fig. ‎4.4‑2Create Product** 63](#_Toc89544861)

[**Fig. ‎4.5‑1Our Users Features** 64](#_Toc89544862)

[**Fig. ‎4.5‑2Add\_to\_cart** 65](#_Toc89544863)

[**Fig. ‎4.5‑33.Registration** 66](#_Toc89544864)

[**Fig. ‎4.5‑4Login** 67](#_Toc89544865)

[**Fig. ‎4.5‑5Logout** 68](#_Toc89544866)

[**Fig. ‎4.6‑1MVT pattern** 69](#_Toc89544867)

[**Fig. ‎4.7‑1Home Page** 70](#_Toc89544868)

[***Fig. ‎4.7‑2Checkout Page*** 70](#_Toc89544869)

[**Fig. ‎4.7‑3Sign in Page** 71](#_Toc89544870)

[**Fig. ‎4.7‑4Registration page** 72](#_Toc89544871)

### List of Tables

[**Table. ‎1.2.2.1 Team Qualification** 10](#_Toc89538269)

[**Table. ‎2.2.1.1 Web Application Terms** 15](#_Toc89538270)

[**Table. ‎2.2.3.1 Web Application vs Website** 16](#_Toc89538271)

[**Table. ‎2.2.4.1 E-commerce Terms** 17](#_Toc89538272)

[**Table. ‎2.3.2.1 3.3 Definitions, Acronyms, and Abbreviations** 24](#_Toc89538273)

[Table. ‎4.3.1.1User Entry 59](#_Toc89538274)

[Table. ‎4.3.1.2Customer Entry 59](#_Toc89538275)

[Table. ‎4.3.1.3Merchant Entry 60](#_Toc89538276)

[Table. ‎4.3.1.4Store Entry 60](#_Toc89538277)

[Table. ‎4.3.1.5Product Entry 60](#_Toc89538278)

[Table. ‎4.3.1.6Order Entry 61](#_Toc89538279)

[Table. ‎4.3.1.7Order Item Entry 61](#_Toc89538280)

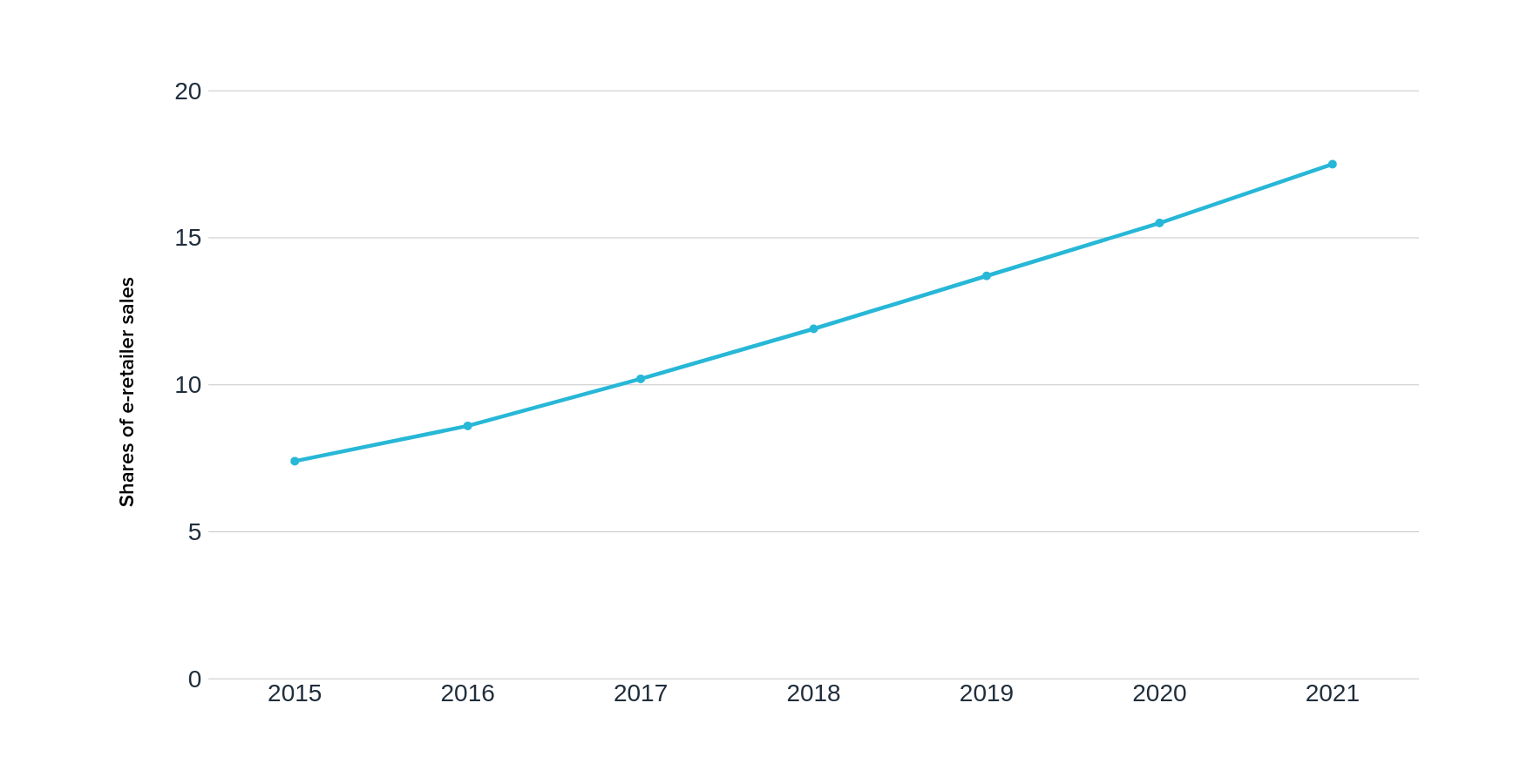
[Table. ‎5.2.2.1Software Requirements 75](#_Toc89538281)

[Table. ‎5.2.4.1Tools and Technologies 77](#_Toc89538282)

#### Introduction

# Introduction:

Trading process from long time ago until now days based on the idea of a place where merchants and consumers gather in one place to trade. People used to go out to the market to buy what they need and merchants could sell their products. Sometimes customers need to visit more than one place to buy his needs. Usually as many countries have independent market to sell a specific product for instance, gold market, Perfume market, clothing mall and tech store. Our platform will gather all those stores under one umbrella. Although we will be creating a fertile environment that will empower beginner Entrepreneurs to reach their customers easily in one place without needing to rent a store or build their own E-commerce website/app. The platform will save cost for marketing, store, development and difficulty to manage operations, inventory, payments, shipping privacy and security. “In 2020, E-Commerce sales are expected to account for 15.5 percent of retail sales worldwide.” [2]. To reach our project we will build a web application as our platform. In order, we need to learn HTML, CSS, bootstrap, JavaScript for Frontend and python, django.py and SQL for Backend. However, we will be using more technologies undecided yet to build the web application such as, AWS for hosting.



**Fig. ‎1.1‑1. shares of E-Commerce sales**

In this proposal we will provide our goals of the project through aims and objectives that are going to be achieved. Then, we will describe how our project will be implemented in methodology. After that, a project Gantt chart timeline discussed in details followed by team qualifications. Finally, a brief conclusion about the project and proposal in general.

# Aims and Objectives:

## Aim:

Our platform aims to create opportunities to new and small merchants to become entrepreneurs by digitalize their business and providing their needs. Although keeping entrepreneurs focused in business, increase profit, expansion, growth and developing products. In order to do that we need to reducing cost and distractions such as, marketing, rent a store, development, operations, inventory, payments and shipping.

## The objectives:

* Learn frontend html, CSS, JavaScript, react.js
* Learn backend python, Django, SQL, NoSQL, AWS
* Get familiar with APIs and server
* Design usable and user-friendly interface
* Developing and testing a web application
* Build up our technical writing skills and improve communication skills

# Methodology:

We will be developing a platform in web application technology to serve multiple traders and buyers. Our E-commerce project will be divided into two main sides; the first side will be the merchant’s side where they could manage their products, add, delete and edit. Second side is for customers where they could buy, review and evaluate. The most work will be in frontend and backend of the web application. Developing the project interface and features will be the frontend job. Although we need to deal with the server side, data workflow, database and APIs through the backend side. For the frontend we will be using html, CSS, bootstrap, JavaScript and react.js which is JavaScript library. As a backend python will be the backbone represented with django.py and MySQL which is python library. In order, to accomplish the project we will need to do more than coding such as, design, mange, documentation, testing, hosing and buying a domain for hosting.

# Project Timeline

****

**Fig. ‎1.4‑1. Project Management Plan**

**Note:** more tasks will be added through GP2

# Team Qualifications

**Table. ‎1.2.2.1 Team Qualification**

|  |  |
| --- | --- |
| Name | Skills |
| Hashem Bajabaa | * Team Management * Basics in web development |
| Anas Alzhrani | * Java development * Basics SQL Database |
| Rayan Alshanabah | * Java development * Basics in web development |

# Conclusion:

In conclusion, the platform used web application model to facilitate traders selling products and customers buying from multiple traders in one place. We belief our platform could add a value in the marketplace and introduce new people to the business world by making selling, buying and managing products easer. Although we picked the latest and best technologies for us to implement our project.

#### Background

# Introduction

In this chapter we will give a brief explanation about E-commerce and web application background in scientific way. Certainly, studying our topics background will make us capability to understand, improve and optimize the topics. Although reading, compare and summarize the related works is very important, helpful and will protect us from reinventing the wheel which will save valuable time. As it will give us various advantages such as, the ability to get the knowledge of where people in the same filed did achieve, reach and if they solve problems that act on our side or get an idea about what the upcoming problems they facing it. In the following sections we will discuss in detailed the background of e-commerce and web application. Followed by related work as we will examine the similar project.

# **Background**

At the first, we provide a brief description about web terminologies to clarify any dust. As going through, an overview section is important to understand the concept behind web application and e-commerce. Following, we indicate the trends-off web application and e-commerce comparing with their primitive abstractions.

## Web application overview

As the rapid development of the internet, the world wide web becomes the richest source for information. the Web has been evolved from being a huge repository of pages used primarily for accessing static information like websites. In the 1999, were an important era where the concept of web application has shown up in Java language as no need to download the whole page from server. Now days, web applications have become a powerful platform and deployment a new web technologies languages and methods to create a dynamic page in addition, desktop apps are converted to web apps. Web application could provide collaboration and cooperation among large numbers of users. A Web application consider as a client-software application approach that runs by the client in a browser. The core function of a browser is to provide an information received from a server side and send back the data to user's. web applications is a cross-platform services due to, the approach that client do not depend on the user operating system. Web application show up after plenty of problems such as, a onvenient way to make web applications working on different operating systems smoothly, reducing cost and time consuming.

**Table. ‎2.2.1.1 Web Application Terms**

|  |  |
| --- | --- |
| Terminology | Description |
| HTML | Is a language that structure the web page |
| Cascading Style Sheets (CSS) | CSS is a style sheet language used to give web pages a better design look |
| JavaScript | Is a scripting language that let web pages interactive |
| Web Server | A computer that run web application for end- users |
| Web application | is simply an application that is dealing with a Web browser over the Internet |

Web application vs websites

They both have an advantage and disadvantage, but what serve our project and will be added value for us is web application.

## Web application

Developers those days they doesn’t need to create different versions of the same web application for Linux, Mac OS, Microsoft Window, etc. The web application will only created once for any platform and it can work on any operating system. Although, content is dynamically changing. More covered on table

## websites

Website is a list of pages that are designed to show up in your browser and retrieve information to the users. Websites are static content means content does not update dynamically. It cannot be adjustable for different devices screen. Users cannot act interactive or even communicate back to the site.

**Table. ‎2.2.3.1 Web Application vs Website**

|  |  |  |
| --- | --- | --- |
| Parameter | Web application | Website |
| Content | dynamic | static |
| OS platform | Cross-platform | None |
| Screen adjustable | Yes | None |
| Deployment | changes require to recompile all the project | Small changes never require a full re-compilation |
| Compilation | must be precompiled | None |
| Task and Complexity | Complex | None |
| Authentication | Must | Unnecessary |
| User interaction | View, read and manipulate | Only view and read |

## E-commerce

e-commerce means electronic commerce that deals with goods and services throughout the internet. Since the internet has developed rapidly and information technologies have become powerful, secure and public which helped to create a good environment for enterprise sector here e-commerce phonomime appear by 2000. Companies in united states and western Europe start to present their services and goods over the internet using secure connections and online payment services. Now, e-commerce is rapidly growing affecting all industries which makes businesses changing their model. Studies shows that e-commerce create a positive impact on enterprise level which helped to increase performance and reduce cost with less effort. Although it provides huge features for companies to increase their revenue, build solid customers database. Even though, it could predict customers’ needs and answer important quotations such as, what they like and they don’t like, time they spend on shopping, when they are coming again throughout complex algorithms. E-commerce assists to make user shopping easy, usable and useful by adding a lot of functionality e.g., shopping cart, payment method, delivery and search engine optimization (SEO), related products and customer reviews.

**Table. ‎2.2.4.1 E-commerce Terms**

|  |  |
| --- | --- |
| Terms | Description |
| Gateway | Is a platform for processes payments on online purchase |
| SEO | Marketing technique for increasing quantity and quality of traffic |
| B2B | selling services or goods to another business |
| B2C | selling services or goods to customer directly |
| E-commerce | Paying or selling through the internet |
| Brick and Mortar | face-to-face selling or buying |
| trends | Is a measurement method for traffic |

# Related Work

This section will summarize papers into two subsections E-commerce and web application

## E-commerce

As this study says from the twenty-first century there has been a leap in electronic commerce and this leap throughout the global economy, especially in the improvement and development electronic commerce offers many benefits to companies in terms of geographical expansion and increased profits while reducing operational costs.

**VALUE OF LOCAL SHOWROOMS TO ONLINE COMPETITORS.**

This study talks about the traditional trade in galleries and e-commerce via the Internet. Where this study indicates that the traditional trade in the local showrooms has a specific time for sale compared to electronic commerce that is 24 hours a day, which increases the percentage of sales of the e-commerce. As this study found that there is a small percentage in the competition of traditional commerce with electronic commerce.

**Eliciting Customer Wishes Using Example-Based Heuristics in E-Commerce Applications**

In this study, it says that electronic commerce allows sellers to display their products, and with more sellers, competition increases in the global market. When the volume of sellers increases, the increase in the volume of offers available for products also increases. This study found that when the volume of sellers increases in a commodity, it becomes difficult for the customer to compare in terms of prices, but in electronic commerce, it is possible to compare in terms of prices for the required commodity.

## Web Application

**User Experience Design in Web Applications**

Web site different from and web applications in user interaction and design. Web sites deliver static content and by using content management system their content is regularly, web application have wide range of interactive features/functionalities and dynamic content. And there are different methodologies and methods for presenting the content in a more efficient and effective way to the users. This is due to the diversity of information received from web applications.

And Converting desktop applications or converting them into web applications and their main feature in displaying them via the browser can make them an effective option to simplify and facilitate the functions on different devices and also for ease of use. Companies and businesses also focus on user experience (UX) so that machine learning is embedded in applications and more and more products are automated. To ensure that a minimum of resources is spent on support, companies and organizations spend researching a user's experience before branding their product. User experience is used as a strategy for customer experience, marketing, product development, and digital product influencing. As the number of minutes of phone use has increased over the years. Web applications use analytics algorithms and relevancy metrics to measure page views, user visits, often searched items, etc. There are techniques involved in driving user behavior.

**Vulnerabilities and Security of Web Applications**

As a result of the massive use of data via the Internet, many vulnerabilities have emerged such as attacks, fraud, and the use of illegal means to harm a business on the Internet. Hackers also target web applications, by controlling networks through intrusion detection. Therefore, the security of the (web application) layer is extremely important for unauthorized users. The development of the web application depends on the communication process between the client and the server, and there are some points in avoiding penetration:

. The parameters inputted have to be checked for their validity based on their data types, length of fields.

. The legality of the parameters is checked by patterns and values for presence of null values, duplicate values etc.

. The application should have the capability to display proper error message for informing the user.

. The error message should be restricted to sensitive user credentials (i.e. userid, password), Card Details etc.  
  
**Visual methods for Web application design**

In recent years, a number of techniques have been developed for usability assessment. These depend mainly on user interactions with the systems and as such, they require a practical implementation of the system, meaning that a usable assessment can only be made late in the development process. At this stage, going back and making architectural changes to the design is extremely costly.

# Conclusion

E-commerce has been a wide topic to cover and to write about it. We tried to provide a useful and valuable information to conduct our project. We narrowed the circle of searching on gathering data from the Saudi database library (SDL) and we have been able to get interesting papers. However, in the next chapter, we will be working on analyzing the system and determines requirements for both user and system.

#### Analysis

# Introduction

This section gives a scope description and overview of everything included in this document. although, the purpose for this document is described and a list of abbreviations and definitions is provided. Followed by UML diagrams that describe the software functionality and how stakeholders could interact with the system. This chapter is very important because it will describe in details the features, functionality, of the system and how users will interact with it. At the end, a prototype shows the web application design and features as a demo.

# Purpose

The purpose of this section is to give a full explanation of the requirements for the “StoresHub” web application. It will explain the purpose and complete declaration for the development of system. Also, it will explain what the system should do and how it will do it. This meant is meant to be delivered to a customer for its agreement and a reference for the developer’s team to build the system

# Definitions, Acronyms, and Abbreviations

**Table. ‎2.3.2.1 3.3 Definitions, Acronyms, and Abbreviations**

|  |  |
| --- | --- |
| *Term* | *Definition* |
| *User* | *Someone who interacts with the web application* |
| *Stakeholder* | *Any person who has interaction with the system who is not a developer* |
| UML | Unified Modeling Language |
| Customer | A person who is using the system but created account |
| Admin | A person who has the ability to manage the system, accounts and modify products |
| SRS | Software Requirements Specification |
| Stock | The amount of any products int the inventory |
| Promotion | A price discount with a specific percentage applied by admin and merchant to customers |
| Merchant | A person who can create a store page and add/delete products |

# User Characteristics

The StoresHub users are simply anyone could access the Internet and web browser. We assumed that users familiars with the concept of using online store and we trying to make it simple as we can. Although, People who would like to have their e-commerce store online and customers who likes to shop online as it convenient.

# Product Perspective

StoresHub software is a web application that enable people to have their own store as a page on the web to presenting their products in one place along with many other stores. Although, customers have the ability to browse the system and buy what the like easily. StoresHub will support multiple devices such as, phones, tablets and computers and multiple operating system for instance, android, IOS, and windows. However, it is going to be supported on most popular browser like Internet Explorer, Mozilla Firefox and Google Chrome.

# Product Functions

StoresHub web application users divided into three possible users Customers, Merchants and Admins. Customers and merchant, they will initiate the system where merchant will create their own stores and adding the products then, customers could browse those products and added to their cart and complete the purchase process. Admin is the one who will manage the merchants and customers processes.

# General Constraints

Thy system provides access of browsing the StoresHub for all potential users, but in order to use the functionality of the system you need to log in. you could log in as a customer who would like to bought products or as a merchant to sell your products. As a privacy and security perspective we will be using encryptions protocols in order to secure our users and database would be separated on external server. Although, the database will store the username and passwords for users on encrypted way. However, users will be forced to use strong passwords as they sign up so we could prevent hackers from trying to suggest passwords.

## Interface Constraints:

Because this is a Web based application that’s why it should work on major browsers such as Google Chrome, Internet explorer, Mozilla Firefox, Opera etc.

## Technology Constraints:

The web application will be implemented in HTML, CSS, JavaScript and React for front-end and python Django and MongoDB for the back-end as a server side.

## Privacy and Security Constraint

Privacy and security will be taking seriously to protect our users and provide a trustable environment. We will be using a multi-factor authentication to login, encryption connection, encrypted database by using web cryptography APIs and the system will force the users for a strong password.

# Assumptions and Dependencies

In StoresHub we Assumed that it will be uploaded on a web server and it will be on running mode all the time which makes users capable of interacting with the system. Although we suppose customers have been dealing with other online stores and having the initial knowledge about how to use it. Each user will be separated from the other and have deferent features and functionality.

# Interfaces

## User interface:

Since, StoresHub is a web application we will implemented as a one page so, we expected to have a user-friendly interface. However, we will make sure to make the system easy to use as we can and make it useful, usable, and used. In order to reach that we will take a lot of perspectives in consideration such as choose a descriptive and meaningful icons to reflect the icon functionality, easy navigation and more.

## Software interface:

Our StoresHub web application aimed to support most of the web browsers in order to achieve convenient accessibility to the end-users. Although our web application has the advantage to run on many different operating systems and doesn’t require a specific one.

## Hardware interface:

The end-user hardware requirements are easily and simple since they only need to use any device that has the ability to browse the internet websites and could run a browser application

# User Requirements

## Functional Requirements

|  |  |
| --- | --- |
| **Functional requirement FR-1:** | Sign up |
| **Related Requirements:** | Internet connection and web browser |
| **Initiating Actor:** | Customer, Admin, Merchant |
| **Actor Goal:** | Register in the system |
| **Pre-conditions:** | The user must have name, phone number, email address  And determine user type |
| **Post-conditions:** | The user will have access to some service |
| ***Side effects:*** | None |

|  |  |
| --- | --- |
| **Functional requirement FR-2:** | Login |
| **Related Requirements:** | Internet connection and web browser |
| **Initiating Actor:** | Customer, Admin, Merchant |
| **Actor Goal:** | To log in the web application |
| **Pre-conditions:** | Must to sign up first, Correct username/email and password |
| **Post-conditions:** | The user will have access to some service |
| ***Side effects:*** | None |

|  |  |
| --- | --- |
| **Functional requirement FR-3:** | **Logout** |
| **Related Requirements:** | Internet connection and web browser |
| **Initiating Actor:** | Customer, Admin, Merchant |
| **Actor Goal:** | To logged out of system |
| **Pre-conditions:** | Must login |
| **Post-conditions:** | User will leave the system |
| ***Side effects:*** | None |

|  |  |
| --- | --- |
| **Functional requirement FR-4:** | Add Items to cart |
| **Related Requirements:** | Internet connection and web browser |
| **Initiating Actor:** | Customer |
| **Actor Goal:** | To add product to cart for purchase later |
| **Pre-conditions:** | Must login |
| **Post-conditions:** | Product will be added |
| ***Side effects:*** | None |

|  |  |
| --- | --- |
| **Functional requirement FR-5:** | Delete Items to cart |
| **Related Requirements:** | Internet connection and web browser |
| **Initiating Actor:** | Customer |
| **Actor Goal:** | To delete product from cart |
| **Pre-conditions:** | Must login |
| **Post-conditions:** | Product will be deleted |
| ***Side effects:*** | None |

|  |  |
| --- | --- |
| **Functional requirement FR-6:** | Add Promotion |
| **Related Requirements:** | Internet connection and web browser |
| **Initiating Actor:** | Admin, Merchant |
| **Actor Goal:** | To add a promotion code for customer use |
| **Pre-conditions:** | For merchant must to have products first |
| **Post-conditions:** | Promotion will be added and mail will be sent to customer |
| ***Side effects:*** | None |

|  |  |
| --- | --- |
| **Functional requirement FR-7:** | Use Promotion |
| **Related Requirements:** | Internet connection and web browser |
| **Initiating Actor:** | Customer |
| **Actor Goal:** | To use a promotion code on his purchase |
| **Pre-conditions:** | Must to complete checkout |
| **Post-conditions:** | Price discount |
| ***Side effects:*** | None |

|  |  |
| --- | --- |
| **Functional requirement FR-8:** | Checkout |
| **Related Requirements:** | Internet connection and web browser |
| **Initiating Actor:** | Customer |
| **Actor Goal:** | To process purchase |
| **Pre-conditions:** | Product on shopping cart |
| **Post-conditions:** | Complete purchase |
| ***Side effects:*** | None |

|  |  |
| --- | --- |
| **Functional requirement FR-9:** | Purchase Product |
| **Related Requirements:** | Internet connection and web browser |
| **Initiating Actor:** | Customer |
| **Actor Goal:** | To bought what on shopping cart |
| **Pre-conditions:** | Must checkout |
| **Post-conditions:** | go to paying page |
| ***Side effects:*** | None |

|  |  |
| --- | --- |
| **Functional requirement FR-10:** | Add product to store |
| **Related Requirements:** | Internet connection and web browser |
| **Initiating Actor:** | Merchant |
| **Actor Goal:** | To add product to store |
| **Pre-conditions:** | Must have store before |
| **Post-conditions:** | Product added |
| ***Side effects:*** | None |

|  |  |
| --- | --- |
| **Functional requirement FR-11:** | Delete product from store |
| **Related Requirements:** | Internet connection and web browser |
| **Initiating Actor:** | Merchant |
| **Actor Goal:** | To delete product from store |
| **Pre-conditions:** | Must have store before |
| **Post-conditions:** | Product deleted |
| ***Side effects:*** | None |

## Non-Functional Requirements

* **Privacy for user’s data**

The web application will keep user’s data without sharing and dedicated data such as emails, passwords will be encrypted stored on database.

* **Usability**

We will make sure to provide a usable interface to our users and evaluate our system with Nielsen’s 10 heuristics methodology.

* **Scalability and performance**

We will make sure that the system could be scalable to reach the expected traffic and order volume at regular and peak times

We will be able to reach a high performance and scalability, since we are using SPA pattern

* **Accessibility**

in our case to access the web application users only need a web browser which make it easy to access.

* **Portability and compatibility.**

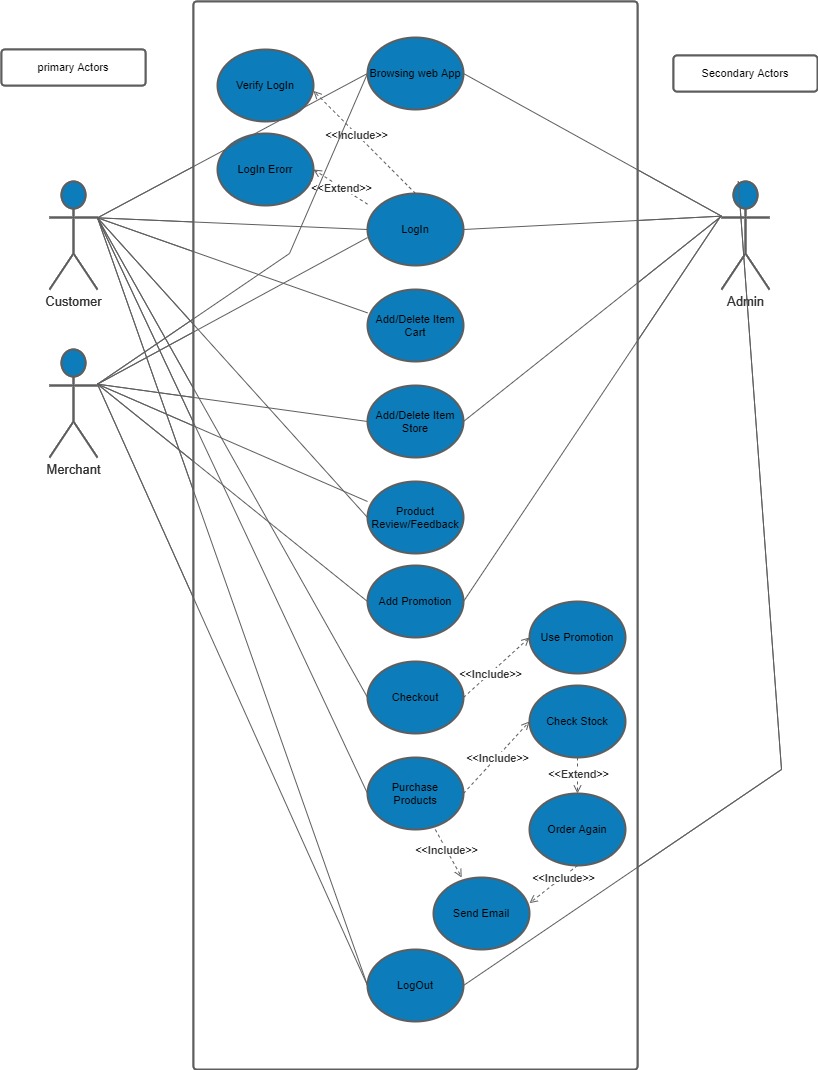
Our platform does not require a specific device only any device that could run a web browser. Although for operating system and browsers.

* **Security**

we will try our best to offer reasonable security by using encrypted connection, protocols and database by using web cryptography APIs although, the system will force the users for a strong password.

# UML Diagrams:

## Use-case Diagram

Use case diagram is a representation of our users interaction with the system in StoresHub.  


**Fig. ‎3.12‑1. 3.12.1 Use-case Diagram**

|  |  |  |
| --- | --- | --- |
| **Use Case UC-: 1** | | **Login** |
| **Actors:** | Admin, Customer, Merchant | |
| **Type:** | Primary and essential | |
| **Description:** | Start when a user attempts an action that is restricted. The user must to enter username and password in order to continue | |
| **Includes:** | Verify Login | |
| **Extends:** | Login Error | |
| **Cross Ref:** | None | |
| **Use-Cases:** | None | |

|  |  |  |
| --- | --- | --- |
| **Use Case UC-: 2** | | **Logout** |
| **Actors:** | Admin, Customer, Merchant, System | |
| **Type:** | Primary and essential | |
| **Description:** | The customer, admin or merchant will have the option to logout and if the user is idle for a specific time then the user will be logged out by the system automatically | |
| **Includes:** | None | |
| **Extends:** | None | |
| **Cross Ref:** | None | |
| **Use-Cases:** | User must Log In correctly use case | |

|  |  |  |
| --- | --- | --- |
| **Use Case UC-: 3** | | **Browse Web Application** |
| **Actors:** | Admin, Customer, Merchant | |
| **Type:** | Primary and Essential | |
| **Description:** | All Actors could browse the website | |
| **Includes:** | None | |
| **Extends:** | None | |
| **Cross Ref:** | None | |
| **Use-Cases:** | None | |

|  |  |  |
| --- | --- | --- |
| **Use Case UC-: 4** | | Add/ Delete Items to cart |
| **Actors:** | **Customer** | |
| **Type:** | Primary and Essential | |
| **Description:** | Allows the customer to add selected items in the shopping cart for later purchase | |
| **Includes:** | None | |
| **Extends:** | None | |
| **Cross Ref:** | None | |
| **Use-Cases:** | Customer must be Logged In use case | |

|  |  |  |
| --- | --- | --- |
| **Use Case UC-:5** | | Add/Delete item from store |
| **Actors:** | Merchant, Admin | |
| **Type:** | Primary and Essential | |
| **Description:** | Allows the Merchant and Admin to add products to their store | |
| **Includes:** | None | |
| **Extends:** | None | |
| **Cross Ref:** | None | |
| **Use-Cases:** | Merchant must be Logged In use case | |

|  |  |  |
| --- | --- | --- |
| **Use Case UC-:6** | | Add Promotion |
| **Actors:** | Merchant, Admin | |
| **Type:** | Primary | |
| **Description:** | This allows the Merchant and Admin to add a special promotion code. This will email all customers who are members to inform them of the new promotion | |
| **Includes:** | None | |
| **Extends:** | None | |
| **Cross Ref:** | None | |
| **Use-Cases:** | Admin and Merchant must be Logged In use case | |

|  |  |  |
| --- | --- | --- |
| **Use Case UC-:7** | | **Checkout** |
| **Actors:** | Customer | |
| **Type:** | Primary and Essential | |
| **Description:** | This takes the product from the customers shopping cart and processes them for a purchase | |
| **Includes:** | Use Promotion | |
| **Extends:** | None | |
| **Cross Ref:** | None | |
| **Use-Cases:** | Customer must be Logged In use case | |

|  |  |  |
| --- | --- | --- |
| **Use Case UC-:8** | | Use Promotion |
| **Actors:** | **Customer** | |
| **Type:** | Primary | |
| **Description:** | If the user is member then they have the option to enter a promotion code to take off a percentage from the total price | |
| **Includes:** | None | |
| **Extends:** | None | |
| **Cross Ref:** | None | |
| **Use-Cases:** | Checkout use cases, Customer must be Logged In use case and Add Promotion use case | |

|  |  |  |
| --- | --- | --- |
| **Use Case UC-:9** | | Purchase Product |
| **Actors:** | **Customer** | |
| **Type:** | Secondary | |
| **Description:** | happen when the customer clicks the order button in checkout. This decrements the inventory of all items, email the user and check stock to see if order again needs to take | |
| **Includes:** | Send Email, Check Stock | |
| **Extends:** | None | |
| **Cross Ref:** | None | |
| **Use-Cases:** | Customer must be Logged In use case and Checkout use cases | |

|  |  |  |
| --- | --- | --- |
| **Use Case UC-:10** | | Check Stock |
| **Actors:** | **System** | |
| **Type:** | Secondary | |
| **Description:** | Checks to see if stop-order is on for a particular item and if it is checks to see if the amount in stock is below the reorder amount. If it is then it will reorder | |
| **Includes:** | None | |
| **Extends:** | order again | |
| **Cross Ref:** |  | |
| **Use-Cases:** |  | |

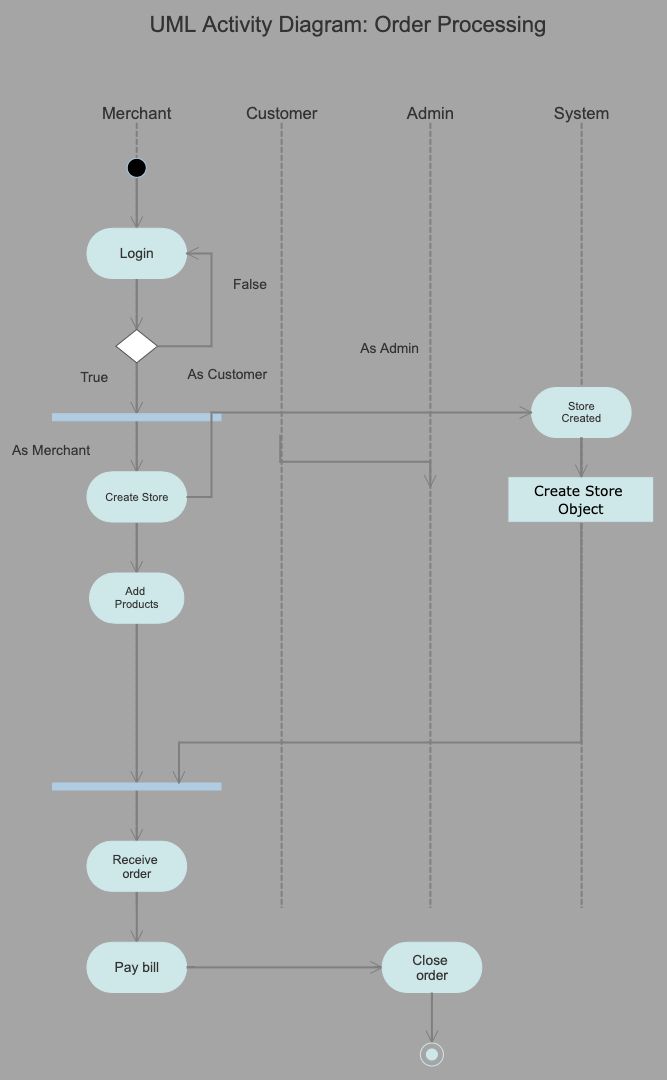
|  |  |  |
| --- | --- | --- |
| **Use Case UC-:11** | | Order Again |
| **Actors:** | **System** | |
| **Type:** | Secondary | |
| **Description:** | Reorders a particular item and emails the Admin when product is out of stock | |
| **Includes:** | Send email | |
| **Extends:** | None | |
| **Cross Ref:** | None | |
| **Use-Cases:** | check stock use case | |

|  |  |  |
| --- | --- | --- |
| **Use Case UC-:12** | | Send Email |
| **Actors:** | **System** | |
| **Type:** | Secondary | |
| **Description:** | sends email to the customer if purchase occur correctly or need to order again | |
| **Includes:** | None | |
| **Extends:** | None | |
| **Cross Ref:** | None | |
| **Use-Cases:** | Purchase Product use case | |

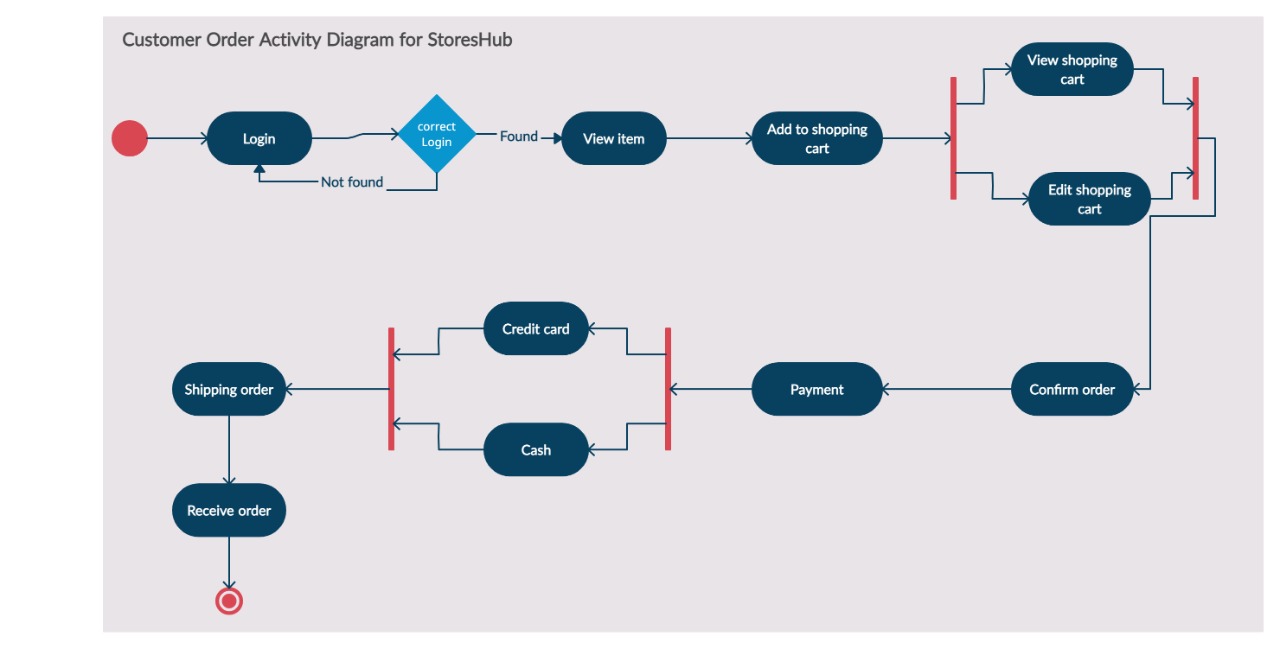
|  |  |  |
| --- | --- | --- |
| **Use Case UC-:13** | | Product Review/Feedback |
| **Actors:** | **Customer, Merchant** | |
| **Type:** | Primary | |
| **Description:** | Give a review and feedback about particular product | |
| **Includes:** | None | |
| **Extends:** | None | |
| **Cross Ref:** | None | |
| **Use-Cases:** | Must login use case | |

## Activity Diagram

This is an activity diagram that explain the activities of the system.

****

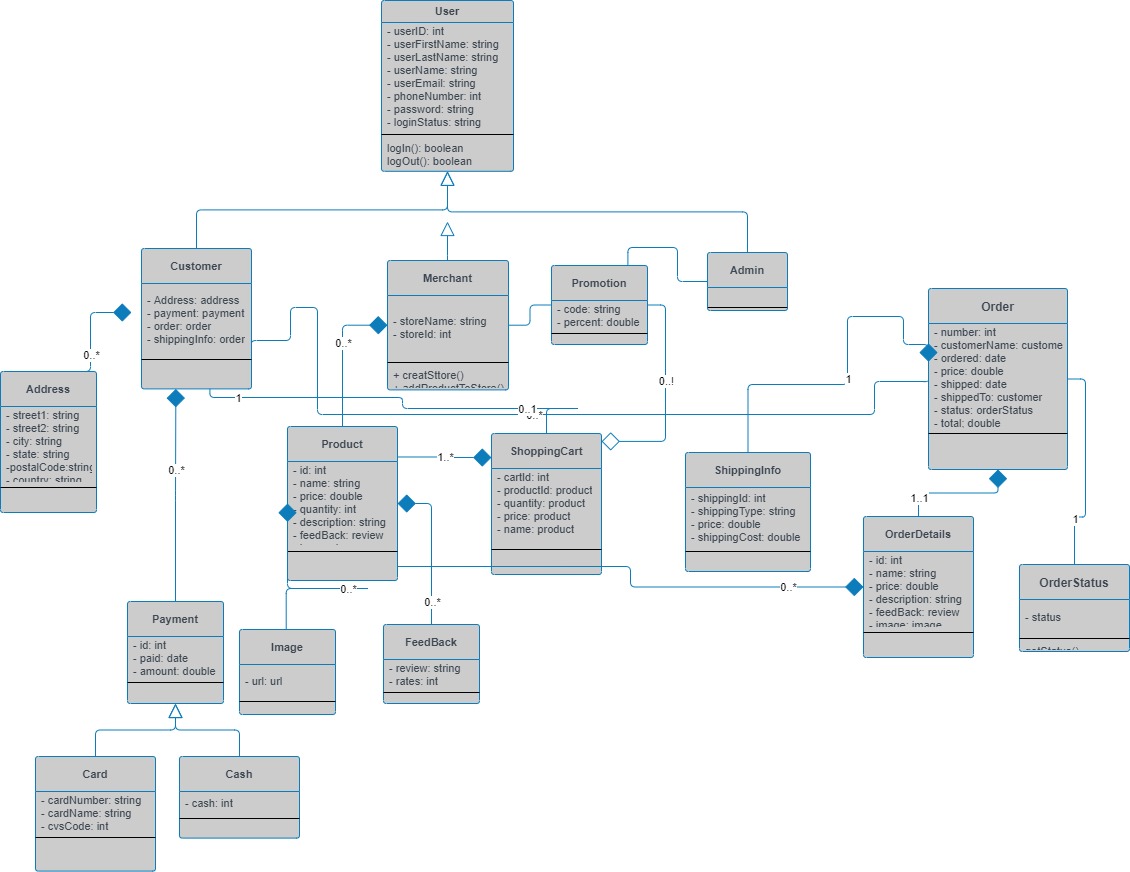
**Fig. ‎3.12‑2. 3.12.2 Activity Diagram of Order**

****

**Fig. ‎3.12‑3. 3.12.2 Activity Diagram of Customer Activities**

## Class Diagram

This is our class diagram we have planned to apply seventeen classes in order to implement our application in the best way we could although, it shows the various kinds of relations such as inheritance and composition. This diagram could be changed in any stage of implementation.



**Fig. ‎3.12‑5. Class Diagram**

|  |  |  |
| --- | --- | --- |
| **Class** | | |
| ***This is a single product in which there are many in the system. They can also be a part of a customer.*** | | ***Product*** |
| ***Public: No*** | |
| Associations: **Merchant, OrderDetails** | ***Relationships*** |
| Aggregations: **Customer** |
| Generalization: None |
| ***Variables - id: int, name: string, price: double, quantity: int, description: string, feedBack: review, image: image***  ***Functions -None*** | |

|  |  |  |
| --- | --- | --- |
| **Class** | | |
| ***This is the base class where Customer, Merchant and Admin are extended and sharing the variables.*** | | ***user*** |
| ***Public: Yes*** | |
| Associations: None | ***Relationships*** |
| Aggregations: None |
| Generalization: Customer, Admin, Merchant |
| ***Variables -*** userID: int, userFirstName: string, userLastName: string, userName: string, userEmail: string, phoneNumber: int , password: string, loginStatus: string, registerDate: date  ***Functions - None*** | |

|  |  |  |
| --- | --- | --- |
| **Class** | | |
| ***This is a shopping cart class where customer could add products.*** | | **Shopping Cart** |
| ***Public: No*** | |
| Associations: Customer | ***Relationships*** |
| Aggregations: None |
| Generalization: None |
| ***Variables -*** cartId: int - productId: product - quantity: product - price: product - name: product  ***Functions -*** | |

|  |  |  |
| --- | --- | --- |
| **Class** | | |
| ***This is a class contains shipping information.*** | | **ShippingInfo** |
| ***Public: No*** | |
| Associations: None | ***Relationships*** |
| Aggregations: Order Generalization: |
| Generalization: |
| ***Variables -*** shippingId: int - shippingType: string - price: double - shippingCost: double  ***Functions - None*** | |

|  |  |  |
| --- | --- | --- |
| **Class** | | |
| ***Customer class inherited from user and have unique features for the customers*** | | **Customer** |
| ***Public: No*** | |
| Associations: Order | ***Relationships*** |
| Aggregations: **Payment**, **Address** |
| Generalization: None |
| ***Variables -*** Address: address - payment: payment - order: order - shippingInfo: order ***Functions - None*** | |

|  |  |  |
| --- | --- | --- |
| **Class** | | |
| ***Order class contains all order informations.*** | | **Order** |
| ***Public: No*** | |
| Associations: **OrderStatus, Customer** | ***Relationships*** |
| Aggregations: |
| Generalization: None |
| ***Variables -*** number: int - customerName: customer - ordered: date - price: double - shipped: date - shippedTo: customer - status: orderStatus - total; double - payment: customer  ***Functions -*** | |

|  |  |  |
| --- | --- | --- |
| **Class** | | |
| ***This class provide a details of product to used in order class*** | | **OrderDetails** |
| ***Public: No*** | |
| Associations: **OrderDetails** | ***Relationships*** |
| Aggregations: None |
| Generalization: Item |
| ***Variables -*** d: int - name: string - price: double - description: string - feedBack: review - image: image ***Functions - None*** | |

|  |  |  |
| --- | --- | --- |
| **Class** | | |
| ***Merchant class is the one who could add product to the store and create a store*** | | **Merchant** |
| ***Public: No*** | |
| Associations: Promotion | ***Relationships*** |
| Aggregations: None |
| Generalization: None |
| ***Variables -*** toreName: string - storeId: int ***Functions -*** creatSttore() + addProductToStore() | |

|  |  |  |
| --- | --- | --- |
| **Class** | | |
| ***This is the class that will deals with customers payment.*** | | **Payment** |
| ***Public: No*** | |
| Associations: None | ***Relationships*** |
| Aggregations: **Customer** |
| Generalization: **Card, Cash** |
| ***Variables*** d: int - paid: date - amount: double - cardInfo: card  ***Functions -*** | |

|  |  |  |
| --- | --- | --- |
| **Class** | | |
| ***This class express Order class status such as, new, shipped, closed and so on,*** | | **OrderStatus** |
| ***Public: No*** | |
| Associations: Order | ***Relationships*** |
| Aggregations: None |
| Generalization: None |
| ***Variables -*** status  ***Functions - None*** | |

|  |  |  |
| --- | --- | --- |
| **Class** | | |
| This Admin class has the additional ability of modifying items and creating promotion. | | **Admin** |
| ***Public: No*** | |
| Associations: **Promotion** | ***Relationships*** |
| Aggregations: None |
| Generalization: None |
| ***Variables –***  ***Functions -*** | |

|  |  |  |
| --- | --- | --- |
| **Class** | | |
| ***Address class that provides information about customer location for later shipping*** | | **Address** |
| ***Public: No*** | |
| Associations:None | ***Relationships*** |
| Aggregations: **Customer** |
| Generalization: None |
| ***Variables -*** treet1: string - street2: string - city: string - state: string -postalCode:string - country: string  ***Functions - CheckStock()*** | |

|  |  |  |
| --- | --- | --- |
| **Class** | | |
| ***Class that has information about the customer credit card.*** | | **Card** |
| ***Public: No*** | |
| Associations:None | ***Relationships*** |
| Aggregations: None |
| Generalization: |
| ***Variables - c***ardNumber: string - cardName: string - cvsCode: int - cardDate: date ***Functions - None*** | |

# Conclusion

In this chapter we did analyze the system and discussed the user characteristics and determine the types of users who we are targeting. Followed by product perspective and functions, then the system constrains and interfaces. Although extend the software requirements specification in both functional and non-functional described in details. At the end, we tried to give an overview of how our project works in high-level and low-level description represented in UML diagrams.

#### Design

# Introduction

As we discussed in the previous chapter, we talk through system analysis, although we talked about the software requirement specifications and we provide various of use cases, also defining functional and non-functional requirements. In this chapter, we will discuss in details the web application design and the approach that we have been followed. In the first section, we will illustrate System Architecture. After that, we will describe our database in detail providing diagrams such as ERD and tables for each record. Next, we will show the class diagram for the web application and flow charts for some functionalities. Then, we will provide some activity diagrams and sequence diagram for our application use cases. Before the end, we will give a brief description of our design pattern, Finally, we will discuss the front-end part with the provided GUI design for multiple pages of our web application.

# System architecture

The system architecture that we found fit our project in convenient way is single-page applications (SPAs) pattern. It will give us the ability of downloading only a single web page once to the user side. This page contains the front-end elements such as HTML, CSS and JavaScript. JavaScript here will communicate with the server on server-side it will be the layer of communicating between two sides. This type of models has high performance on deferent devices because its lightweight and responsive. Data transfer from deferent sides shown in figure below.

![A close-up of a cell phone

Description automatically generated with low confidence](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAYABgAAD/4TWuRXhpZgAATU0AKgAAAAgABgALAAIAAAAmAAAIYgESAAMAAAABAAEAAAExAAIAAAAmAAAIiAEyAAIAAAAUAAAIrodpAAQAAAABAAAIwuocAAcAAAgMAAAAVgAAEUYc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAFdpbmRvd3MgUGhvdG8gRWRpdG9yIDEwLjAuMTAwMTEuMTYzODQAV2luZG93cyBQaG90byBFZGl0b3IgMTAuMC4xMDAxMS4xNjM4NAAyMDIxOjA0OjA4IDE4OjM5OjI3AAAGkAMAAgAAABQAABEckAQAAgAAABQAABEwkpEAAgAAAAM1NQAAkpIAAgAAAAM1NQAAoAEAAwAAAAEAAQAA6hwABwAACAwAAAkQAAAAABzqAAAACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAMjAyMTowNDowOCAxODozOTowNwAyMDIxOjA0OjA4IDE4OjM5OjA3AAAAAAYBAwADAAAAAQAGAAABGgAFAAAAAQAAEZQBGwAFAAAAAQAAEZwBKAADAAAAAQACAAACAQAEAAAAAQAAEaQCAgAEAAAAAQAAJAEAAAAAAAAAYAAAAAEAAABgAAAAAf/Y/9sAQwAIBgYHBgUIBwcHCQkICgwUDQwLCwwZEhMPFB0aHx4dGhwcICQuJyAiLCMcHCg3KSwwMTQ0NB8nOT04MjwuMzQy/9sAQwEJCQkMCwwYDQ0YMiEcITIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIy/8AAEQgAmAEAAwEhAAIRAQMRAf/EAB8AAAEFAQEBAQEBAAAAAAAAAAABAgMEBQYHCAkKC//EALUQAAIBAwMCBAMFBQQEAAABfQECAwAEEQUSITFBBhNRYQcicRQygZGhCCNCscEVUtHwJDNicoIJChYXGBkaJSYnKCkqNDU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6g4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2drh4uPk5ebn6Onq8fLz9PX29/j5+v/EAB8BAAMBAQEBAQEBAQEAAAAAAAABAgMEBQYHCAkKC//EALURAAIBAgQEAwQHBQQEAAECdwABAgMRBAUhMQYSQVEHYXETIjKBCBRCkaGxwQkjM1LwFWJy0QoWJDThJfEXGBkaJicoKSo1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoKDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uLj5OXm5+jp6vLz9PX29/j5+v/aAAwDAQACEQMRAD8A81FOzmvbPJHUoNADwacKBDh1pwpiHU8UxDxTu9MQ4U4YzQIk/GlpiHCnUwHUoPagQue9LTAXNLQAd6UUCFozQAuaM0AOzRQBxwNArA6BwpwNAHYafPpY0SxVX0pccXqXURM7P5nBQ46bMd8fezWhdW/g24trmY3u24VT5UcKgL95sdMe2eDx+YwfOnobLka1KmoxeGo9M1FdP8kzH/U75GJAWZxlT6lNhxXJg1rT5re8ZT5b+6OFPH1rQzHCn/jTEPHancY4piHDpS54piHAmnZoAXNKKYDs04UCFzR70AdD4Wl06Oaf+0PsuDswbhc4XPzY+U547cE9iKty23hWKyWWO4knnEDP5bEoGfAwOuRznjjj8zzydRT93Y3jyOOpMtr4OcSE3U6HyBIoznDNuOz6r8o98nr25/WhYR6k8enYNsqqA24tuOOTz78VVN1HL3iZqCXumfn2pc1uZC5p1AjjadWB0Dh9KUUXQDhThRdCH04U7oVhwI6U8HjrTTAcPSng4piFzTwaYhwPane3HNMQoNKDQIfupc0wFz60oPvQAu9ehIH40b1x1H50roLMXev94fnRvX+8Pzougsxd69dw/Ojcv94fnRdCsxd6/wB4fnRuB6GndBZjs0uaYHIKNxwKtQW7yyLFDG0kjHCqq5J+grjnK2h1RRsjwjr7AEaXPz6gD+tL/wAIhr//AEC5v0/xrDmRpysP+EQ1/wD6Bc36f40f8Ihr/wD0C5v0/wAafOg5WH/CIa//ANAub9P8aP8AhENf/wCgXN+n+NHOu4crD/hENf8A+gXN+n+NUr7SNS0vBvLOaAHozLwfx6UKSvoJxfUgjkLcE81LnmuuErq5hJWY8GnD0qyBQafnimIXNOpgLn3pQaBC7gFJJwB1NZs93JK21CQvbHU1lVnZWRpTjd3L0PhfXbiISx6ZcFW5BZdufzqT/hEPEH/QLm/T/GuTnR08jD/hEPEH/QLm/T/Gj/hEPEH/AEC5v0/xo513DkYf8Ih4g/6Bc36f406PwZ4ilkCJpcxY9BlR/WjnQcj7E0ngPxREm5tImx7Mp/kapXXhvXLCIzXGmXUcajJcISFHqSOlCmn1E4tENpfvuEcpyDwGPatQGu2lPmics42Zy0Pc16v8ONLhh0dtSZFM87lVfHKqOMD8c/pXHWZ10lqdtXO3E7SXVwEXUm2uy5hlO1TgjgAe+efQV5mKxKoJO17nrYHAvFuS5lG3cs6NK5uJone9ZtoYi5YHHPbgY6/pW5DEZpljXqxxV4esq1PntYzxmFeGq+ybucpe6+l1eMLBdWjRHCHZaLICVJ3DO/GD9M8VraBereWkloReteWib5pLmJU3qzHHAY8gfyr0J4SdOLk5LQ8uGNhOagovU0qhu7SC+tZLa5jWSKRdrKRXMdR4Pf2p0/VLm03bvImaPd64OM0oP616FF6HDU3HZ+tLnitjMcDwKd2piFz+NOoEOFLTEQ3bEW5A7nFdH8ONMgvNUuLudA5tVXywegZs8/pXJiHqdNBHqdFcR2BRQAVl+I9RudI8P3d/ZuEuIVBRioOMsB0PsaunFSmovqyZtqLaOETxf40k0xLm3vLXa65RPLQMF37M4x03YFVbDx14tk1nS7W+u4Wt7y6WF1EKcr5mxh0+tbVHhlU9lBa6/gYR9rJc8ih4z0yHSvEk0Nuu2GRRKqj+HPUfnmm27FrdGPdRW2HZlWWpzkHQ/WvZ/AP/ACKFr/vyf+hGsKxvS3OmrmLtit5dxAQNiUOFlcp1HJB3D2rxMz+CPr+h9HkX8Wa8unqi7oyx/bLpo9uAkYO1iRnHODz3zXQ2cqw3ccjfdB5NbYFf7Oref5nNmzbxbv2X5IxfDF5Ppr6t9jsnvp11GaNokvNpjTPD+WeMZ/iq3pskQ8QeJZ1mMyv5MSMX3ZO0lufQE17VWPxy7pfmvzPm6UvehHs3+TLFFcR6J4V4i/5GbU/+vqT/ANCNVQa76OxxVNxwPNKD71uZDh0pwoAX0pwNO4rDs+tOz3ouJor3n+oH1rtPhd97VPpH/wCzVyYjc6aB3moXi6fpt1eupZLeF5WUdSFBOP0rgY/iDqc1zDbhdFjnnRXSB55d4DDcAcJjOK56dPne9jonPl6XGJ8Rr+S3luEl0IxRFVdhLN8pbOP4O+DTE+Jd3JIqJNoJZiAB5s3J/wC+K0+rv+Yn2vkzd8PeL7zUtfbSb62tVYxyOktrIzAFGCspDAdz+lavi+Ca58K38NvE8srKu1I1LMfmHQCiMHTrKMujQOXPTbXmeX3MPiU6NbWEGl6nFHApyFt3+dt5YHp2449qWG38Q6p4w0a7u9H1GKKCe3BL2zhV2ld7E7QBkgsfrW2Iw1BVVVi1dc3XuzClUqOPK1pp+Bf+I3/I0f8AbBP61m2v/HrH/u0YcVfc563+6frXtHgH/kULX/fk/wDQjWNY2pbnRySJEhdzhRgcDPXgVXntra5/11nK59Tbvn88Vi8L7eFpJW82bwxcsPPmg2n5D0MFrHhYJIUyBnyGUegycVYodD2KS6C9v7aTk3dlK78P2N7MJriwLS/31VlJ+pXrVq3sUs4NkFt5UY64TH51mscqj9jzbB9UUH7W24+iqGeFeI/+Rm1P/r6k/wDQjVIHFd1HY4qm48HvThW5mOH0p1MRJB5RmUTsyxE4YoMkD2FdTB4j0+JSki3EsSqAI9u3eQiKM4bGCV5BBx2NZzi5bFwkluJH4ngt0uMPPePKz4a4QcIxXKdTwQDn8MVk6rPDqers9gsvklESNZMbvlQLz+VT/DvOWw2+b3UZN6rJHtYEMG5BHSu0+F33tU+kf/s1Y1pKS5o6o0opp2Z3Oq2h1DR76yDhDcW8kQc9F3KRn9a8Ml1GysL0zSxwXE1iIlbyNVHlysqbQyrt54/Ksac4x+JX2/A6JRlL4XYu2v2Wyt5Ybe6tY/MCAuviBQyhc4UHbwPmPFRyagtpPb51CSXe/G3xDuUYBPzfLwOMfjitliIpK0XoZ+xk+qOo8Ak6z4ll1cT26rBHNH5AuvOkJaQc9BhPl49sV6dWNWp7SbkXCHJHlCisyzyT4jf8jR/2wT+tZdt/x7R/7td2HOKvuYFv91vrXtHgH/kULX/fk/8AQjWNY2pbm9e/6hf+usf/AKGtW7l7f+1ZluEuXAiTaIlkIHLZ+7WuHvy6ef6Gda3NqRSOH0S62eYIxOAiy53KNy8HPPXPXtihzhCRXNmEnClKS3V/yNsIuaaXoJtM91MFkwBvYEc9MmrAgNuzYdnVos7scHkdOea+Ow0Ze1VW/wBo92q1yctuhFRX1p454V4j/wCRm1P/AK+pP/QjVDPAruo7HFU3Hg07POK3Mx2a0rGYw2F5MqoXUxgFlDdc+tcWPjz0eXu4/wDpSNqDtO/k/wAho1a47pB/35WlGrT5+5B/36X/AArD+yKHd/e/8y/rdTy+5C/2rcf3IP8Av0tSw6tIsytIkZUHOFjUGoqZNRcGot39X/mOOMmmm0vuRU1jUZL9AXVFAbgAc/nXWfC772qfSP8A9mrSnho4WiqMXew1VdWfOzsvEYY+F9WCZ3mym24652GvH7OecX1gYrm1FgIodwMtuEC+V82Qfn3bq2w/Nf3bdN+w6vLb3r9fvJtI1Cay03UrjVJbSSWOAPAkksDbpAGLY2Hp93rzWxpuneFL+GF/EGq3Md66BpLW5Js1UnsqjGR75P1rzOIcZjKGGhHDJOTvfa5tl1ClOrKVTZbbm5D4c+HcB3xPYIxGN66iwb89+aq3Z8Oaec6P4mv4rsf6uCzunu9x9PLO7NfFYTMc39quZNrrdWX32R7lShhuU6fwjq9zrnhm1vrtAtw5dZAF28qxXOO3TpW5X38XdJnhtWdjyT4jf8jR/wBsE/rWTb/8e8f+7Xfhzhr7mFb/AHW+tez+A2VPB1szMFUPJkk4/iNY1jalubV7cwG2yJo22ujEKwJwGBPH0FSya7Zfa3ngvim9VUq9pI3TP09a0oK8P68iKrtIin1awksZYUuHknmlViTCyDOV9RgDA9ame4tZEZGniwwwfnFZYqPNHlfW5eHlZ3IdPXT9IsvsVlcRpBzgHDMM9eSc96ttqETKF+1Q7cHgbV9Of0r5yhhK6r2lsmetOtT9npuR/arf/n4i/wC+xT0mikOI5EYjsrA17hwHhviP/kZtT/6+pP8A0I1ng9q7qWxxz3FBpwPFbGQ4HjFX7bP9k33+/F/M1y4z+Gv8Uf8A0pGtH4vk/wAimDThXWYjgaXPrTAiuf8AVfjXcfC772qfSP8A9mrkxG50UTudWvG0/Rr69RQ7W9vJKFPQlVJx+leHS2ltqF8YbiURy3wiaR4dOiEcbNHuCqc5HH51hCEJfE7bfidEpSXwosgx6voeqSQfZ4xBamQh9JgjLKwbBUryPunmvULTxXpw0m1tfEGj31rthQEzWbTwt8o5DIGBH1xXiZ9hrxpulLXXyOvAVruSkiq+p+ADfxuv9gi32N5iNZfvC+RjHy9Ovar48V6HDA0Gg6Vd3btwEsrBo0/FmCrj3zXztTD4iUbTdlbqz0Yzgnoit8PmdvB8DOmxzPOWXOdp81uM11Ffa0rezjbsePP4meSfEb/kaP8Atgn9ayLf/j3T/dr0sOcNbcwrb7rfWu6d2Hw40xQxCtdPuHry1ZVNzSBc8SaLoWn6JaXGnXvm3DkDHmBvMBGScdqxJrW0XS0mSXMxA43dT3GK5stnKtRlKto1ceObpVIxpap2uTpYaedBNyZsXOCcbu+emKu+DdG0XWLu5j1i++yrGgMY81U3HPPLenH51d3ZmWHnKbkp9Gdj/wAIP4I/6DZ/8DYv8KP+EH8Ef9Bs/wDgbF/hUc8ux2csQ/4QfwR/0Gz/AOBsX+FYK6fpuj/EGyttGvjc27Jl23hsEhsrkcHoDVRlJ7kuKWxw/iP/AJGXU/8Ar6k/9CNJoWnLq2rR2by+WrK7FsgY2oW6kgDp1J4rrpu0Wzmkrs1LrwlcpK32aRXVUDEOQfmK7sBlyp4wQc4OQBzVSXw9dxAkTQSKFZsoW7CTI5A/55P+lUqqE6bLD+Fb6FpRJJEBFw4AckcsDwFzgbTz06c1HPaixi1i0WTzBBcLHvxjdhmGa5sVNSgrd4/+lIunBxevZ/kzKB4rW0jRW1WOWQXUMCxSIhMhxncrsMe/yYA7kiu2UuVXMEruxYPh4rOkL3XlyTzeTbq8LKXbCn5gfu/fAzz+XNLqfhu50yxN3JLG8X7oKV7l1Y49iNvI9walVNbFOm7XOfn/ANV+Nd18Lvvap9I//ZqxxBpRO9v7RNQ065spCyx3ETRMV6gMCDj864iL4eXUM8M4vtLeeFFRJn01i+AMDJ83risKdRweh0SgpblPU/BV3o3hvV5ba80yOM2rtKsOnshcBTxnzTjqe3euu0zxDrGn6PajVPDV44WFAs2mss6uMDnbkOPpg/WvDz61dU4yai1e3Y7cDHk5mtbg3jqL7YhGm60sIUh4TpExctxghugHXjB+oqabxPqV9auNL8M6kzngPfBbaMe53HcfwWvnauESjepNJetz0I1NfdRnfD7zP+EPt/NCiTzp9+3pnzWzj2rqK+2pW9nG3Y8afxM8k+I3/I0f9sE/rWPbn9wn0r0sOcNbcw7X7rfWvSdMs1v/AARpFuwBL3bKuSQASzDtWNXc0pm7/wAKquf+etn/AN9v/hR/wqu5/wCetn/32/8AhWPtDbkXcP8AhVVz/wA9bP8A77f/AAo/4VVc/wDPWz/77f8Awpe0H7MP+FVXP/PWz/77f/Cj/hVVz/z1s/8Avt/8KPaoPZB/wqq5/wCetn/32/8AhUEPhM+HPEenecYmeTcymNmOMA+v1pqpfQXJbU4LWVtG8T6t9qklQfaZNvlqDn5jUED6bbSeZDfXiPtK7ljA4IwR17gkVEp4tSapRTXm2Ry0rXk3ctJqUEVuYE1bUFiKeXsCjG3njr0+Y/masTa4LhNr6vqBzEYX+QZdcseeefvt+BNTzY9/8u4/e/8AIf7j+ZjW1dXDhtZ1Ih12MCOo59/9pvzPrUT3FrcCaIXNxNPeSoXeRRnOTk9evNHLjajUZwildPRvo0xXoxu03cikt9NhmeJri43ISpxGO341fg1kW1qltFqVysKAqE8hCCDu4Oeo+d+Dx8xr1mk9zlTsA1eMNuF9PkHKH7On7s7Qvy/3eAOmOgpkupRT20ltLqF20MjrIyGNcFlXaD+XFJQSG5FCaPSynNxcde0Y/wAa634YbfM1XaSV/d4J9PmrDEGlHc9EorkOoo6rDbXumXVhcXCxJcRNEx3AEBhjPP1rEt/EGv6PFFaPb6VqsMShEmgvRBIwAwCyNkZ+hxXnZhgVi4pXs0dGHreyexaXxNfz3kdwnh3VGdEZPKjvbcxNnHJHmckY4OPWnTax4sv8xW2hWdgjcGa8u/MI99iDn868iOQzk7VJaeR1PGxWqRb0HSV0PR4bATNMULM8jDG5mYsTjtyTxWlX08YqKSXQ85u7ueSfEb/kaP8Atgn9axYD+4T6V3Yc4q25i2v3G+tep+G/+RX0L/r+/wDZzWVU1p7nR+OXZddjAYj/AEdeh/2mrmfMf++3516eGhF0o6dD5/GVJqvNJ9T0bwduk8OY3kEytz+VbfkSf8/DV4uIsqsj6PBu9CF+weRJ/wA/DUfZ5P8An4esTpD7PJ/z8PXLeKf+Rm0b/dk/lVR3JlseI+I/+Rl1T/r6k/8AQjWVXoUtjgnuKKUGtiBw45qzZn/Trf8A66r/ADFAiS//AOQjc/8AXVv51XzTBjgfenZpiGTH93+Nd38Lvvap9I//AGauWub0TrvFGrS6H4cu9QgjWSaIKEVgSMswUcDk43ZwOtYuhvBqNtNJJ4ik1VrhDHIgdY0TPUBFwVP15rkW50ybS0JI/BfhuPpo9q/vIu//ANCzU3/CJ+Hf+gFpv/gKn+FVyoy55dyGTwV4bk5/si2Q+sQKEf8AfOKytVWDw3GJLHxNc2TL923mm+0I59AjZb/vkik4oanI63w3qza54dsdTeIRPcR7mQdAc4OPyrVpGp5J8Rv+Ro/7YJ/WsSH/AFKfSu3DnHW3MW1+431r1Tw3/wAivoX/AF/f+zmsqprT3O48T6dp13qiSXeqLayCIAIY92Rk81i/2Jon/QfT/vyf8amGaxpRUH0FPh6tXk6sb2ep1/h+1tbfRfJt7sXEO8nzANvPpV/7PB/z1P8A30K5p1Payc11OqnQdCKpPdaB9ng/56n/AL6FOjjhjfcJMn3YVJZN5if31/OuQ8U/8jPo3+7JVR3JlseI+I/+Rl1T/r6k/wDQjVKOwuZLZ7kRFYVGS7cA/T1r0KWxwz3INrf3T+VLtb0P5VsZi4YcbT+VWbJW+3W3yn/Wr29xQA/UD/xMrn/rq386gBpiFHHFKDQIbL9z8a734Xfe1T6R/wDs1c1c3o7nVeLP+QAf+vq1/wDSiOtHxbY+DoUjuvEWnQfvCVFyLZiw+roMj8TXDLc7Ec9b2fw4lAFn4oNtkcImtvH/AOOs9Wv7E8Igbj41vNv/AGGxj880XYcq7FTVtK8GWvh661ZLy81uKBliMcerySgyMQFUkPheWGc9qteFYtCnsrl7Hw9b6Ze2kzQXCjbI27aGBEnVhhhUc65+UrkfJzIPAn/Ik6Z/uN/6G1dFWy2M2eSfEb/kaP8Atgn9aw4f9Sn0rtw5x1tzGtPut9a9U8N/8ivoX/X9/wCzmsaprT3Ok8cf8hyP/r3X/wBCauar5uv/ABGffYH/AHaHoeh+ENn/AAjf7z7vmtn9K2dtn6j8zXp0f4aPl8b/ALxP1DbZ+o/M0qx2rttXBPpk1qcpJ9kh/ufrXKeKf+Rn0b/dkqo7ky2PEfEf/Iy6p/19Sf8AoRqzp15rVzZGKAGVSfKjbeqvvxkBcnLH25rvptJXZxNNvQzv7Z1Lp9sl/OlGs6j/AM/kv51tcz1FGsajn/j8l/Onf2xqP/P3L+dO4XZUaRpHZ3JLMcknvQDzTJHUooAbJ938a774Xfe1T6R/+zVzVzajudN40jabwvNEkrRO9xbKsi9UJnj5HuK10l8aaXCVmtdN1xV4V4ZTaysPdWBQn6MK4ZbnYtipJrkzH/iYfD3U2boTHHbzD899Q/2lpOcj4eanu/7BcP8A8VSGSSazNc2Utjb/AA81F7aYbXhnW3hjf6guf5VS/s3xJFpqWen2Gk+G7M7iUjY3Eoz1OAAufclqA6DfAIK+B9LBJJEbcnv8zV0laLYh7nknxG/5Gj/tgn9awYf9Uv0rtw5x1tzHtPut9a9V8N/8ivoX/X9/7OaxqmtPc9O1TTdMu7pZLy1MsgQAMHI4yfQ1R/sPQv8AoHt/39b/ABr5ivjaMKsoyjqj6GjVxKppRloa9hBZ2VgUt4THAGJ2klufxqT7Rbf88x/3yK9ahJTpqUdmedVcnUbluH2i2/55j/vkUoubdTkLg+y1qZjvtsX+1+Vct4p/5GbRv92T+VVHcmWx4j4j/wCRl1T/AK+pP/QjRpHiKTSbdIFgEkfnNJJ821mUqFwpxlSOoYd67oK6scTdnc3h8QFKhW0tihhMboLgAE/L0OzKr8nQHPoRTpPHkK3A8mxuHhSUTKz3WHLDb8udpxF8vKe/Wn7LzH7UbF4+YbxPp7So24bBcbcKViXaDtJAzHn5cH5uCMc5+p6tJ4mFpbQ2qwNArMxaQEdFBx8oKr8vQk1M+WjF1JbIFJz91dTOGky/8/Nn/wCBC0v9lS/8/Nn/AOBC1y/2rS/ll9xf1WXdDv7Kl/5+bT/wIWj+y5c/8fFp/wB/1p/2rS/ll9wvqsu6Gy6ZIsTN9otSFGcCdSa7L4Xfe1T6R/8As1NYmNdNxTVu4403TaTOm8aTLbeGJp3DFY7i2dgq5OBPGeB3NdCb6DxTZpL4d8SrC0ZyWt1jlzns6sCR+hrKW50LYpnT/HUPEWv6PcjsZ9OdD/47JSeV8Qun2zw19fs0/wD8XS1HoGq6J4l1jwjf6dealY/bpmUxtbRvChQEEox3E/NgjI7GsPTvs/gjRrmLXNQs7Tz52lgs45zIIVKqNiZALZIJwBjJqeV8/NfQfMuTltqSeAmDeB9LYZwY2PP+81dHWy2M3ueSfEb/AJGj/tgn9a5+I/u1+lduHOOtuZNp9xvrXqvhv/kV9C/6/v8A2c1jVNae56pf/wDHwP8AdqrXweNT+sT9T6Ki17NGjaNttCdpbB6Cn/aD/wA8H/Kvp8Ev9nh6Hm1v4jD7Qf8Ang/5U6OXe+0xMvuRXSZ2JcD0rj/FP/Iz6N/uyVUNyJbHiPiP/kZdU/6+pP8A0I1iDnPtXfT2OKW4tOrYgdntWjox/wBKl/695P8A0E1zYz/d5+hpR/iIo4IRWIO1iQD9OtWU0+8dlVbWYs6GRQEPKgZJHtjmujSxnZks2k6jbRNJNZXEcaAFmZCAAemfrUdzZ3VkwW6t5YWOcCRSpOPrTTT2E01uQN92vQPhd97VPpH/AOzVz1zaludV4t/5AB/6+rX/ANKI60vE+neEYCl1remR75WIFzFauXB9S8Yyv1JFcUjrRz8cXgOX/jy8Z3dr/sx63IMfg7Gr1toei3qym18d6xOsKGSTytYDbFHUnHQe9Iepn6pZeF7Pw7dauur6rrsVu6xlF1mQqXYhQGKsAoywyfSp/CyaFe2lzPZ6BaWF5aTtBMyETEsFByJSMtww681PMuflKcXycwvgT/kSdM/3G/8AQ2ro62WxkzyT4jf8jR/2wT+tc7Gf3a/Su2gcdbcyrM8MPxr0/wAJTPe+G7eC1VXurC680xFsFlJyD+fFZVTWnud3/wAJN4g/6A0P/f3/AOvR/wAJN4g/6A0P/f3/AOvXL7OJ0c7D/hJvEH/QGh/7+/8A16P+Em8Qf9AaH/v7/wDXp8iDmYf8JN4g/wCgND/39/8Ar0f8JN4g/wCgND/39/8Ar0ciDmYf8JN4g/6A0P8A39/+vWdd3Wo6hqdtqGo2sVpBZo7MwcEYx9aaikxOTaPHdSvIrzWbu9MZMU07ybM4OCScZq7dajoz6bGkNkdqtlovM2HOODnB3d6wxlDE1J03RlZJ6/5k0qlOKlzq7M77Xpf/AEDJP/Ak/wDxNL9q0s/8w2X/AMCT/wDE1p9Wx3/P5f8AgP8AwSfaUP5Px/4A77Vpf/QNl/8AAn/7GpYtRsYN7QWEiyNGyAmfIGRjpilLCY2a5ZVVZ/3f+CCq0U7qP4l/SdbsNO0pIJoDO8glDhPlK524+br/AAnoceoNT/8ACWovmRR2iiBre5gUFELqJFKph8ZAAIzg84r0eR3uY86SsGneKba0W6jnsmmhuIreN034z5UZX/0LafwqLXtftddt4XkjnS6iUjOQVYnGSe/Y01BqVxOaasc8eldz8M72GHUby0kcLJOimPJ+8VzkD35/SorDpbnoOraamraZNZSSvEJNpWSP7yMrBlYfQgGoIda8X6efLubDT9XiHAmgmNvIfcowK5+hFcTR1piXPimCXH9o+C9Tdsc/uIZx+auaht/Fmi2hk+zeDdYhMq7X8vSQu4ehx1FKzKJP+ElgnspbO18Eai9vKMPDLDDDG/1DN/SqEcfiGS2Wx07TNK8O2AzxG3nSLnqQoCrn6k/jQkI2tK02DR9KttPtyxigTaC5yT6k+5NXKsk8b8c30N/4nmaBw6RKsW4dCR1/U4/CsVOEA9q7aCOOruYcUhjcMPxFa9hqM9nOtzZXDwyj+JDg/SlJFJnQDx74iCgfbUOO5hT/AApf+E+8Q/8AP2n/AH5T/CsuRF88g/4T7xD/AM/af9+U/wAKP+E+8Q/8/af9+U/wo5EHPIP+E+8Q/wDP2n/flP8ACj/hPvEP/P2n/flP8KOSIc8g/wCE+8Q/8/af9+U/wrP1PxNq+rxeVe3jPF3jVQqn6gDn8aFFCc2zDkkB+UdO5qPNdEVZGTFpRVAOzS5piFBpc0CHA0ZpgOzTkd4ZFkjZkdTlWU4IPrWdRXVyoux0EXjnxDDGEF9vA6F4lJ/PFP8A+E+8Q/8AP2n/AH5T/Cufkibc8g/4T7xD/wA/af8AflP8KP8AhPvEP/P2n/flP8KOSIc8g/4T7xD/AM/af9+U/wAKP+E+8Q/8/af9+U/wo5IhzyD/AIT7xD/z9p/35T/Cq934z169hMMl+yowwRGioT+IGaORA5yMRFycnpU2a6qcbI55O7MGlBIORxWfMjXlY/zZP77fnR5sn99vzpXiFmL5sn99vzpfNk/vt+dF4hZh5r/3z+dL5rY++fzp3iHKxfMf++fzoLk9WJ/Gi8RWYA0oPvT5kHKxcj1pQR60+Zdxcr7Dgw9aUEZ6ijmXcOVihh7UoYdMinzx7i5WLuHrS7hjrT549w5X2FDL6il3D1H50c8e4uV9hdw9RQGX1FK8B2kLlfUUuU9RReArSFynqtGU9RTvALSDKeq0uU9VovALSF3r/eH50bx/eH51XPHuLlfY/9kA/+Ex6Gh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8APD94cGFja2V0IGJlZ2luPSfvu78nIGlkPSdXNU0wTXBDZWhpSHpyZVN6TlRjemtjOWQnPz4NCjx4OnhtcG1ldGEgeG1sbnM6eD0iYWRvYmU6bnM6bWV0YS8iPjxyZGY6UkRGIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczp4bXA9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8iPjx4bXA6Q3JlYXRvclRvb2w+V2luZG93cyBQaG90byBFZGl0b3IgMTAuMC4xMDAxMS4xNjM4NDwveG1wOkNyZWF0b3JUb29sPjx4bXA6Q3JlYXRlRGF0ZT4yMDIxLTA0LTA4VDE4OjM5OjA3LjU1MTwveG1wOkNyZWF0ZURhdGU+PC9yZGY6RGVzY3JpcHRpb24+PC9yZGY6UkRGPjwveDp4bXBtZXRhPg0KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPD94cGFja2V0IGVuZD0ndyc/Pv/bAEMAAwICAwICAwMDAwQDAwQFCAUFBAQFCgcHBggMCgwMCwoLCw0OEhANDhEOCwsQFhARExQVFRUMDxcYFhQYEhQVFP/bAEMBAwQEBQQFCQUFCRQNCw0UFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFP/AABEIAcQC+AMBIgACEQEDEQH/xAAfAAABBQEBAQEBAQAAAAAAAAAAAQIDBAUGBwgJCgv/xAC1EAACAQMDAgQDBQUEBAAAAX0BAgMABBEFEiExQQYTUWEHInEUMoGRoQgjQrHBFVLR8CQzYnKCCQoWFxgZGiUmJygpKjQ1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4eLj5OXm5+jp6vHy8/T19vf4+fr/xAAfAQADAQEBAQEBAQEBAAAAAAAAAQIDBAUGBwgJCgv/xAC1EQACAQIEBAMEBwUEBAABAncAAQIDEQQFITEGEkFRB2FxEyIygQgUQpGhscEJIzNS8BVictEKFiQ04SXxFxgZGiYnKCkqNTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqCg4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2dri4+Tl5ufo6ery8/T19vf4+fr/2gAMAwEAAhEDEQA/APjjfubvk+tPCEDJOQPzpNwXkAZPSnKwJJHXiv1A/PBUHPT35qTaVOQp560xSASCQFNHmkOR1BFGwDyhZsbcCmqAHwP4fel8wCPBPSjAXaUHXnpQBIuQMnOD05p0ZOc5wCOTTCeg6Y7HvTScE8HI7dqAFxzk8N2qSI5Xk8gfhTDhyAQen5U5SNqrjHPpQBJhcYHX270q4AP079qYxLAAYBXkmnLh85OT2HQUBqSRttjGTyetOC71+8OvykjrUYztOQPaniP5FJydp4oJJkAQkHg49KQSjJzjPuKEYN04zwaYmEdmYbiBxn/PWqAlUDGQe9PVioByCPzxUe0EHA59MYqRF2EkgfdpolimQNJxgkjPPekxsjBBBGTge9CkMxIJzj1p2Iy4Hf8AnQIk8s+TvZsDrgHIFKqZkygxkZ4pgkCgptyDkjj/ADmnBA27kgewqtReoK23I/iJz+NS/K5K5xn72aaqkR4yOOASBR94k5xnGaBkiRqhG1enOfanSnbtOGI6cUxSG5UAKDgZpsgLgKDkH8RT2RBL9oKrlugyQR3pwZ9x+U9OabCqsoQkZHA4qZEJ6nA9jTWoOw0Z8ouSVJOOTnip41Lqx3AlRnB71BtPKk5UjHHHNSIuUABwBwQeeKEJkilWUEuoz94EUsW5Yn2hduccU0lQCp24APTuTSqwRML0z9PzpiJI+RtI+VRxx3pdzJGuQeTnmmCIKAeSODhe1PBLbzuLA+p6VZOgqEs33A38RqZWUt8oz6n0qBnKqpU/dIGPUU9WKdBgPktx0pJiYi5bhMkHgKTwPep90aMFY7mAxUPH3gMjkDFPVUWL5QS5HUjP4UIGT8sh2hdoHHPFRhWUAFRgDJOfypUX5SuFHHcd+9ObAj64ULjHYmmSPihYgHJwT0Jqbabc7DydxJBPaoIwGj4zjrjFSxBGIJJJB4UjiqQmO84tE2cE8cBs96dC5QMChHp9ajKlX2DjtkDpT9qrt5X5hj0zj1oDUcQADkglenFSswfBAba20Nx6etNQktnO3bwQR2odirK5dvm/hHANAaiglTkDAPPynkj+lKribIAY5I4zjiowQ2QAVPbnjPvSRbvN+cgOehxwBQKxOyFWO5Qi9qkCBSFKcsM5xkYqvJIfLYEjDNknH4VPb5TjGFUY5OeKpCHI2Ms+GPUED7tEmXAcsPmbnIximqu75wcDP3ac7iTJL5AG4DHH4DtTAcqlQx4Hcc0pLLLnOQR1ao1JVskYJ685NKtxtyjKvY885oFYk2mN1bO4H5unNSROXGMjOSx9famwOXLAnAznmkGV2xggkdvXFBIxHZG3En72COtTD95N0JBORu7CkAOSGIYEH9aPnIYKCxB4A4oAleJCzlTgL1PQZoRfkBPXrxzUcWSCCcsecMPz5p7S/uXIGcDCk0ASOQCRnkL8p6YqMsWjGBwxpEJfDvjLKBkVJg8EZ4GCB2qtwHMkTHcfmCjB9RR5JIUgDPXGKj3L93b8zHPA61IGJzxgimAhnLSgE5fBOO4PYCnI20KSNvUHPY1HhWOQcOcjjjPpTiQGGQwb696AHj/U+ZtJYHGOvPakSR2Rzgbj+tO8z9ySWBO4Y45GKYWw2QTjHPpmgBwUSsSc7SMcDoaf5YXBzxjgHrTGVyQDKx+bpk459qfhc7T9aAA7WOM7c8kEdaJGKx4GBzj60wyGDHGDyM5pTufZu53UAKMKoB2hxkZPcelLuOCCCOOOOKiZR5hCggqcAYqUhlTJOc8kUAIVAlKYOSBipGwr7SVDAdu9RIDvLAhh0Oec+9OICy7gD9TQA5CdxOQQetDEKhDMVXkfWljYbmB59xTGILBDkAkYAFBJIF3HIOTjjNEZI4xnac4A7mkAGSMZpo3JjqQP4c80FEjMVf5AeTgY70wAI7Agk9TSZLg44/CnAHzCwA3bRn+lBOw8sMPt6AdKSNhJGq5wDwO+KbIOe/XIzTtoVQFBwc59DQUKm1kfBG4cHFPU7ACDk9s1GAMNj5c9cUo6A5BPpQKw4El8Dg9fwp/OznI+bFRqSfc9zTXY4I5APUe9ArXJSxGSQOuKRW3AAen51GnK5wMDoDUikbAcA49aB2HONqgA55wfan7TgYxkZOKiOcjB+91pW+Vsc465FBI/zA546/TrRkHk8cHODTN56dFpM44wCSeuKrUCRHyDjnHSimhDyR971FFcGI3R10dj59RS/OaGGDkfmBTQxVMA8k9qlSTK4IOPWsNztI1JK4ABzzSr7nHvS58vGB+NBIfB7e/FSA/CYwBk+lKXIwBwSOcdqj37QQF5Pp2p6sSDke1WBIqh3yTyPf2pCxXBxmmhgHyTjHTFP4ySPmGPyoAdnnIHUZzmk3nIAG7HWmggcAH0p4HOAf0oDYUZYZIwPano2FIIyAe1NUkkDHyg8inK5WTIHbn3oDUcpG05DcdPanhyDg85z1Ndt8Evg/rHx0+Imn+FNFZIZ7jdLPdTAlLeFfvSMByQMgADqSBxnI+tR+yP+zha+L4/A158WNWbxsxEDQxzQLD5542Z+zlVYnjyzLu7da46uLp0Jcsrt76K9l3OinhqlWPMrJeb6nwvv2KMHbxzT4VGV3Y5z3zmvq74d/sZ6Jqv7TmufC/xF4gmv9MstKk1KDUNDmiSVsSRqqSBlkCsAzZXGeFIODzzeh/AXwFN4K+Ld/qFx4uOseFdQ1C10r+z9OmuLN1gQmM3U8ds0aEsPmJkjAXn5RzS+u0r2V3s9u+wfValrvTdb9j54LLuGOfenIAwwT1457Vv+F/hn4w8ZWUt54d8Ka3rltE5SSbTdOmuEVuu1mRCAcHPNVdJ8FeI9e1x9E07QNS1HWI9xfT7O0kluE2/ezEFLDb344711+0jrrsc3s5aWRjxhVDLuLbSRn2pQVEobBIHB7V1jfB3x7b6fc6k/gjxIlhblhNdPpM4hiKkh9zbMLtIIOTwQc1638Ff2R9V+Knwp8X+NrhdTgj06ynfSNPtbFml1OdYyy7GKkOm4bSEDFjkAqRWc8RSpx5nJWvY0jQqTdktdz5/jBViHYDJyvfikjYFhhwecHjipL/Tr/R9Qn0/UbSexvbZjHLbXMTRyRuOCrKwBUjuCM1CfkU4+6ec8V0ppq5ztW3JmYYZDjCnI/2qQEiTYAcfzpTgqoJz3NPVgSpzzjHPeqFsKsnyFTt6546kUJjywScYHQD3puFVXLrg9ARznNOiwQoBxxxmhEiAkOVycD8alifaOhU55JOc/wCFRjO04JB7ZFPTDA5GOPTrQhsnUKsbSkhmB6DpS7kHIDfMOSOahZtq4Hyg0qn5AMnGMcd6q5JIdjHPQAAgHjJqaMllAxk4z29agLbsAD7uOoqdGCjH3frxTQmTI5U5cBQSBxkn9KQbXLFRkA9OmKiU/dPzEZ+Y9s05cZyh6dQOcU7k2HmAPgKSMAZ5p4wgG85x7ZBpsfzbgADgdc+lIJ1mIK/KpGOnQ0aBqTRS8nKL3Cj+dNjlUyBQrKQecnNLGwxjOPm7UpBVmYAYPGcVQh4YAEoeQ3PHY/WlLnO1tuAwzxUZiBXLEttJIA4xUyqpQsF4bGc96BaIHLJGEzjoCce9TRgsOWBX7oxxuxUTgzY3MFUDrjqKRgY+CQB04Hr6VWotyaSUxsM5wx4Oeg9KmADYL4Zc4AzyM1D95VJO78P1pFUknkbl6AHqKBDopQWfJKqTjoak8osyNv8AkAwB9aiG8gchATgDHU1OM+WQAqMB196EDHsBtAJA3dOMkVHLJvkCgZwcA4wMDtTEJyMt8wHTFOwBIQCSVwcn+XvTuKxOoBT0zyBg5NG4yJgDJOM4OPrTHYbmBHOMjnpUkEiJFtbrjOfXPSqEPUgx4LKo6dKHkSJRnDbug2kmmwsqrJkA5HX0p1vguQQp+Xkk8jFACxtubBDbjwD7e9SLAWQk4Jz0z0xUSMMkY5HJIPJpBJ5eBnLcEYPagCVkPYAHHJJOaCroEO8bsj8qj2uzl8kZ7D+tPwJEYkgZyenSgB7AkkhlyD0z1FKHKHcCu1v4sGoNgA3cEDAqRWEmVAAIx34FCYDrYkTElwV/vc/lQxfzWQrlR2BwDUbxkOo3ArngD/61SLgDJJyBkH3oAmVlZMAHhTjuT608uVUoARu43dzULKTGHLYJ4AUYzmnBgFzg/N+lO5NhQxToCSo5PcfhStkjI68devWmE4k3A5YLzml3iQ4Y4wR0ouIXlJGw4KjJOOSG7YzR8xfcBkA5569KeYwsmQcDOSM5oRhuIXAwcjJ6ikAkplzkAbBznvn09aVY3EbEAjv15NOIUx7iGHUDsaQsFUBSQehBFWA9XDxl8DAPTikADgsTkk4Hamq2xFBXcCecdqHY7McAA+uKAJZVQyEAk7ThuuPrRuXcAQTzwRTY13SMFYhcYA65PrSHcoAJ6dQaAHNKoJYA8NhiD0NKSC7HJI4x2zW34F8H3Hj7xlpPhyyeOKfUbpIVlk6Lk8sR3wATjvivquX9lb4MaT4qtfBuo+O9Yj8XzRjbbK0axs5GR1hZVJ6hS+SK8/EY2lhpKM7tvWyV9O52UMJUrxco2S8z45QGM5OAc8nrT8NLuyFx6g13HxW+FOofC34j3PhFt2o3G5TaSQRktOjcoQoydx6EeoOM8Vg654D8TeE7dbjW/DmraPBI+yOW+sZYEY9cKXUAn2FdMa1OSi1JWlqvMwlSnFyTWq0fkYocpg4wTzx3+tHmZRgTkk8Gug0z4X+M9Usra8svCWu31rcqWgnt9NmeOUeqMEIYYBPGaq634F8TeFbOK413w7qmjQSsUjk1CylgV2HJCllAJ74FNVqbfKpK/qJ0ppczi7ehkp8u1Qd24HrQww4GG3BfwFaeq+EfEGi2NlqGoaJqVjYXeBbXVxaSRxT5GRsYqA2Rz8pPHNXtW+HXivRNLN/qPhfWtPsAATdXWnyxxc9PnZQOc8c0/aw095a+YvZz7M53O1+vJGMChZWRjuwSMHjrit/Rfh74p8Q6al9pvhrV9SsZJPLW5s7CWaNmzjaGVSCc8YzmszWNC1Pw3qTWWradeaXeKoY299A0MgU9CVYA4NCqQlLlTVxOnNLma0KXmMwXqDnkmpGfaDjB28DPemYGQD/F0+tJhg2QDgdQa1JJQQ6nJAb+dMUMrYGNoHIzSoBnkcHkd6YF3O2RxnAoBaDixUNg805yWAAHGB/9ekBGOelJkqWPIB4PegByMCOMHB6VIAXIAJGScge1RKxDA5xnjGKfk7to6DJzQKwM3C4H8VPGdwAztwcimEgqoHpzk0rOW4UYJHJoEPaPAxkHsaVSUwR1zio1yRgnpzkUrH5s9KBEgfbknBP8qKjUHkkY/rRXFiN0d2H2Z8+/dPzEYPfFOBIP3unOD0ppOOOo680gYtyT3rn2OokYjAO0kH0NIA2ASMqPzpV6hSeGHTFK2APlbOOgFIBQSfuAGlUHqT3/AIe1NPyg8ZJ96VSVyCQKABW2++enFOUktjoB1x0pGQ7QQOnWlXJUgLjPv2poB+CoOCMZ7dqBkHgjHbPpQEO1gQeOlKqjdk53Y61QDlffkAcr6dKcq4fJYY9D0BqJgQwIPH86eDk98dQaVwPsn/gmDqNjZ/GTxBZzSql9eaOwttw5YrIhYKfXHOO4B9K8J8dfB3xtP8btc8LJoGo3niG61KYx24hbMyvK22UE8bDuDbidoBySOa8/8LeJtU8H+ILLW9EvZtN1axlE1tdwNh42HcdiOoIPBBIIINfU9v8A8FN/i1b6etvJp/heaZUCG7exmErNj7xAnCZ78Lj2ry6lOtTrOrRSfMktXax3wqUpUlTqNqzvp1Oo/YJ+Gvib4VftS6noXivSpdI1VPDU8v2eVlbKG4gAdWQkMCVIyCRwa7v4X/N+zz+1cev/ABOtd/8ARJr5M8D/ALWnjrwX8WNT+I7yWWveI9RtXspm1WJ2iEZZGwiRsm3b5agAHABPBJzUnh79q7xdoHg34g+GrfTtFksfG11d3WoyyQymWJ7hNjiEiUBQAfl3BiO5Nc9XC16k3J2u+X8HqdFPEUoRSV7K/wCKPu34hSaL8NfhZ8IdN0r4l3Pwx01RBJAmnaJcaidWcIjeW3knPzFiSGzuLdDiq+n33h7W/wBunw3rGi2uoWt5d+G7r7euo6Rdaa0jqQqOEuI0ZsrkbgCPlAzxXxr8O/21/HXgTwZpnhltO8OeJtN0h1bTj4gsHuJLTb90Rssi42/wkgkDgHHFZei/tY+O9L+NE/xOnOn6t4ikt3tVivoXNtFEwwEREdSoXHHPUknJJNc8cvrrm9H1Wt/66mssZS91ra66bH2/8Dfid4o8XfthfFjw3qmrz3fh6yhaK10xz+4hCSKoKr0BIZsnq2ec4GOL+BWvatpX7PX7RMNnq19broV7qkWlmO6dDYqsTsohIP7vDfMNuOeetfLfgf8Aau8X+APit4k+IGn6fosuteIA32q3uYZWgTcwc7FEqsOVHVmp/wAMP2sPGPwpuvFn9n2OianY+JbqS6vdO1S1ea38xixYqodTghtpDMwIA+taywFSz5YraOnmtyFjIaXb3f3PY8f1DUr3VtQnvtRvJ7+/nkaSa4uZGkkkYnlmZiSze5Oaqo54GODknPc1seMPEkvi7xZqWtz2Nhp0l7KZmtNNiMVvET/DGpLYX2yaxizIuRztPJavoY6JaWPEla/cmjcMcBTvJHUcCgMVlLAZ28YpgdFYMeuMgUvzbieAAc56GruQTJES/wAzqd344z2FKwVWIyMgdTUSvhjliWz35pyqA4BAOe3emBKWDoVVeR/Fzg4qRdhiDEKGUdc9ajUOGxtwQ2AAe1LhfmwcjuO1BA9YCzZOMMMk7uhqVQEUglgRwpPSqysEQ7hkc5+lS7g68HgDjJpoCSMosgI+ZiRk9Rk09izuvAcE8n0HtSRvsQsRkkcDtSRjcAcgEe+cVQE8ZbdsY8Yzu6UkcQTJGd2enrQ2UbJCuRTQWdyWBUZ+XvQSSEMM5Chup54+lPGFAJ3cDODTGYE4zknnOaEAIZSOnNBBYVkLKQe2BxQd7EKGIbJ56A01GCMfQjqO1JGTuUk+tWBYDY2jceepApzoY2G0/Ke+OhqB5JI9pCgg9CRT0cvKACCuOfXNArD1YSZDngqNo79aQgiXLjAA5zyR6UGXsqkMp9KFHAzkuxbHPqarUkk3BYgpJJzgHp+lSwRohZmRtykHJ/l71BESY2OMk8YAxUscxYRhsheARjqTSQNDMmTIDJhTkAt3qQTAuFd84AB2r2qMh42GACBzx1poEm4hkUblPK8HPpRewy1koWYqDk8bahUurs7bQCQSSenpTlwUQD5SoPXvTYyV4yfmxlR0/OnqJFjzCxUseJBhSo6/5xSN8gUNt+bpk0hjTC8HJ6Z7U0qY3xj5B1bOcfSgSJo8gMhAGecnvUihUYlioH+10NMSRhGASMr0OKevzSGRiM9t3cfSqQxkbbd5VgwY5J7VKrqVQMo2HPI5Oe1VtrlzgKMnketTJGN43koATkY4FJCdiVpCkQwAcnn1xTEjDYLFlDEY5zyKSYIy7gM44waXzAgKHAwMkiqEPcBmGMKvKnNNbdvYICQMKPRqYHwuFJPocZqQDzGyCMgAFveluAqxGHvkLnOOQCaemMZDY3dmpm11GCcjPX196kEZAYk4U9Ce1MBVkd/lICjkZ64FKw3/ACJzk844pq8AgHcSAc9s0KC4BySM9vWgBwh3FlAIx8pIPelQKWzjbjqfWlZnSZSRgn5st0wKRIwAyghtxz9KAH28uwh3ABbnngHFPBDSuQRxz6YppYShgBgKeSfam5JDAjaeqk96CCX7QWQlsBAQQg6/U0N8ycttPUGmBNzZIIOM4FPyVABOSx4I+lADfMysfIUZJIPf3p7qhfDuORxj+VRFcBA4H3fTnNSkFiOB0zmgBJCYUBTO89AQcfSlcPtV+nHFMmzu3E/KOBgcU/5yAgwQRw2M0Adn8IPD+seJviRodl4b1O30nXnuBJZXV47JGJEBcAlVY87SMYIPQ9a+tde+K1zN8QNP8B/Ff4Uab4j1eV1gj1XS4jKsittG+FZIw20bvmIYYweK+INO1S80i9t76wuTa31rKssM8Z2sjqcqR6YIr6Ks/wBvn4kWdjFbyWnh+8kQBGuprSXzGOPvHbKq5+ige1eJmGFq1pxnTipJK29mn3uexgsRTpQcZtp37XT+R738MvgZ4V+GP7RmtnS41WJdGS8sbeZzI1q0kjJJjPJACLgkk4YjPrk2vijw/rXw9+Iul61441f4h2tz5xLL4Wv1TTZArHZvCOqhWCkZIC7c9K+TLP4/+NbP4mt49OrCTxDIPLZpEHktCcfuSgwAnA4GDkZznmui8cftVeL/ABp4Zv8AQPsOh6DYahIZL46JZtA9yT97zGLtndgZIwTjBOOK4JZbiZTi5O+i1utLf10O1Zhh4xajG2r0tvc938ZeNta8B/sT+C7rQb2fTr2dbeH7VbMVkVcsxAI5GdoB9s1D8M45v2qf2dToWq3TXPiHRdYiZ7mYhnaNpAzPn3jeRc+q183eJPj14g8UfC3SvAN1aadHoullGgnhjkE5K7sbmLlT949FHamfBv44eIvglql9faBFZXH22FYZob6NmjIVsqwCspyOcc/xGt/7Pqexlyq1TmbT+fcx+vU/ax5neHKk0fYeqaxo/i79rjwv4QZY5NM8JabJJBb5+T7UUUjK9CVTbj0INa2leMdH/wCFkePNP1Xxzqni2ynVra58L2/hi9nTTh93G6NHGMZBIADHnNfBK/EfXbb4hP41gvjbeIWvGvROgyFkYnIAbPy4Yrg5+XivUdc/bK8b61pmp20NhoGj3WpxrDd6ppdi8d5IoBXl2dhnBIBxkZ4xXPUyur7qg7qyvrs+r/4Y2p5jStJy0d303XQ9r8B+IJ/BH7Gviy+8PXEtrLb315FaXI3LJGpnCKwyAVYKe+CD718aa94g1bxVqkupazfT6nfyKokubhy7sFUKMk+gArs9M+PGv6Z8Jb34dQ2unNol07O87xSG5yzhzhg4XGVH8PSvNyx5cDAx3PavZwWFdGVSc1q27PrY8vF4hVowjB6JarpccWUgcFgv86YzMHGA3yjoe9Gd3AHB5xTl+VuDxjBr1TzhQ4C4IOcYwOmaZHIRGQ3qTmlA25PHB69acFyqkEkDJqAFLFT0GOlOwWRSQSP5VExJA45PT3qQkMucjIXnFWAN865OV5zx1pUJ3ZGSOnPWoieBkkDripIcBC2M5xjHagCQOFYg4yelCv8AICME45ApjKGcZ/8A1UMeSAD8o7d80ASLICTgY/XFOPzuDg4HB7VGVA4GOaenABJzQJjyRgDqexFFNDAMeRgGiuLEbo66GzPnzJUEHk9c0AiQ5oUAHJ6EHOaEIJPHArmOsGAOMDnPFOOWI42+pFISCDz7cUvBIXAHbNADmJDYzxnI5pwzjkAkDPHWoyQ3txg0m7YMjgEYxQBOEDevPTNO28Y7j3qME8AdMc5pu8IoY5yT1p3HYmDFVyeR9aFOc4yO1IHIBA/iH5U0R8gZ46n2pCJMjPKZHTg9Kec4JySQcBT0NN49s/lT2BHzfKRgfWrAFUkZI5Pr2qRjgLj5scN/jTVBXORxihiBCPXuKA1HBS4wQF54xTlYqUGfUdKbGw2rkA46U9l3H5R0PagAQELgDJBORUqsPl3DGe3v2qM5LgE+3AqQZAB6dsU0RuPccqAQe5brk04sN+TnO3j2phYZByOnr3oDMpHAz0A681RI/wBPfkEetLswpAIyT3pqfLjO7Of1pytkHG7jqB1oAkXBVQTkhsn8KAuVBI5pirhckAknAyae2flypUdxQBLGoGF5Vic9c8UpO1jldxP4ZphbGXXAYADJ6ikVlXJJJYDI96dwJVJbgcsvTnrUmQI+o3EZ4HT2pqxhtrY+92p+3A24I47kcf4U0TceZDt2g/wgE9B/9emsiZyQWAPQn2oWMxkHk8DHPT0pdxGMEHPP0p+oegRyBGyAV5yOelShwFIZSWYdVqJHLMQAxXpkjOKkAGMZAwcj1prYQ4ICygFhtAwPanDhfvfxZ9hRDtJUqp3YznuaQqW28AEE7gR0piLG/e2CAQPvFuh+lPZFAQooIGc9sVWdmVDhRuY4zT0IXCnqRzg8frTuTbqTBsOCOg7UGbIIJwpbj1B/wpmMjAyDg5weOKGIV+GAwuVK85NUJIdG7R7CzZ4OeOMUqTAbnVShHAx0AohcEkFgRjIp7YbbuC/MRyDx9KSF1HhyWAAJAUHPvQrOm4OMZyQw5I/wzSLKpQjd83fFOff5MgIwfl24IPGOaYEuHIUg7geQG7VI6+WwJ+VQOR13H3Heq+8OFEXHy4PpmpGj/djc2ZAOp7GncgcAGw2zoex5Ip0cm9HAGAz/AHD1H0quvO0YbIPHzdaldWV8nKg/wn1ouBNtB3ggBm4x/WmrGd7OSDnaBxzgUxHMkgGDkmneYSpIB6njOcUwHRs21gCC3r6D0qVgHVVztYHc3PSq4k3R/OCNwwM8jr1p3ltsZyDgjgf3qEwsTSMV2IMHnI54NJLgMC+VbJyM5AP9ajRARknGB2FPQh8HB6detAak/mIMgICOD0xTSdzOARx/eHHsaZE4kbkgspxgjk0rOu1iQWYcfT0p3JJ5XAt4woLHOSe1IgwN5ww6Ad6iVy0QKjkkDIp20kAAA4zz6VQE4BRd2Mc+nSk2CNgpLH5uuajV22jcCxXgnNPY4AIIXvt/pQA93LSEE5IwCetOEn3cgEE/Nj0qAKdrEjg8j2pUYqQcbge3tSuKxMwBVkBbceQQOi9qiePb85Zs5ycGhTliyZXuQOaeEDNjOOc5o3BDjgurltsZAHA5J71IQkWcFcg5GahZGYYztxjGD/OlAJYZGSBnOetMZMtwXIJG7acDH86Q7mwTwg6AdKiCFEXB5zyueKUZJODhRznNF+4rExfI3ISCAPlJ70gz5eSW5GTz0NKs5JwCvzDGD3HtQuCcEduc0EhHMGwHTdx8vtTpMudwO0DsDTZGRY8jhh0BA5pGleTIx/Dzg0APnQqVwVG3OcnrmmDO0dMH06UrIDIEfBZhnBpoY9CAOcfSgdhzEnYFHKnkkY4qWbk4Qg7sZ9qY4J2Eg/OSpNJjYx4JCnAIP60CHbiXC7iAefoe/FKclic5A6ZPWg5GDjGc49Ka6bUBxuwegNAD9u4ttPXnJPFSZLKoxnjOD6+1Vy/ARTtORgfzqVnKLnj5uBzTuAjMOTwce3U05cqvI8snBPfimNhhgj0FIz4PfaCM5PWkA/a3OCCccnPWmrhcDAwBg81IG2OcgYx34qNRkEkjac5oAcwwxIPtikVtxPHYZFGNxJBAU9fc0pcckkA9MDrQBJ91QQOO/vSIeOgwcYz0zTEYsuOcUgAAIA74OfSncB+MggnkelDYEeFzuXHOefegNt3EHGOPwpqybtyHDM3OAO1IBSxl5B+UCpOFVf7hGPxqMsQB0BHBpTjcCegH60APAK9xihQFJyc89aFfAwRg96Ut82cfTHYVYDuFbgHPWkkXec5IPamk7m5wD0p2ccA8cD8aAFGWJ7jA/OikRSrHnp7UVw4jdHVR2Z8/BtxwOBinLlTuB46YxUbDIAC5OcfjT0yDjk4965kdxIrAEHB4HrTlkVmIXg1XbKnJHHapoYnfG0HPU5HA/GjmS1YuW44fIcnkk8e1LtMpOQCwH8Iq1HZquC53n9KsABRgAAe1cksSlpFXNY031KCW7j+A+2elK1u7MCV6c4Aq9RWX1mXZF+yXcqeW4ONhxjpShHKgFCCOMirVFL61Lsg9kisIm29Dn371IFZR0ycYzUtFV9Zl2QeyXcg2v6E0vlscjIA9MVNRR9Zl2QeyXcYc5AC/4UoULJnBOcdyAKdRR9al2QvZx7ggKk4PfqTTlba2eoOM57U2in9al2QexXcm3IjgqVPfp0/OlV0ZyxyCffAFQUU/rUuwexj3Jw4AIL5B5p6yIAMke/rVWij61LshewXctq8aqy78gtnBFPW4GCN4x2J5qjRR9al2QewXcuC4CupDAAdRgc0B4yxywHPFU6KPrUuyD2C7mkZ4uDvPIA44xilFzF1BXd05rMoo+uS7IX1ePc3Y5DK4PAGACaRX2tnG7JwD3xWKjshypKn2NWotRdcBwDj+Ida6YYuMtJKxlKg1tqahYszbSR8w+UZHHrTcHhyckHGD6023uBLgqeAMs3bintJlzkqQeSR0ruTUldHNZp2JBwyqCVbuQPXtSuWUkkknGDz2piTOWbacZ4yBxxTTJlgCwOMk4PWncRKkgjiTdtcknryfrQwDkKAMckE5pka7o96ruI46YyKfCrqu9j1PHf8AWmA8AyRqclW4JHTNKqs6ggbSMg9u9IZkfaSdxO3AHfJqXByQWyy5yo7UbkbDQAAodhkcMABUytnG1VwvByB0qJlV1BJAYDkU+F1A3A5GcEd800J7Dx90EYwTz0FPjIJyAeOCSciomYSZG3aN3THempH568EKVyDzincRKkRRiqNtYj5ScYFSKXWIb3Eh56ADNQoT5Q3ZIBI3Y5xjtUkccZjQJuwpwWJ6Z9aENkwbDKCMEHqTwTUstwWYdzjLEDqB+lVzIIpAAC6noeo/Spwo2MAfYc9PwqkZjVBI4yR1wTTwwVRtGwYzgevaohnZgKc5wcg01fmfdtICr83XFIdh5Y7QSM88L3FSK+II8FiyjJHp7U0uNxyDyMAj0/GmEhmIyY+Oc8A4781WoiYsJHBBIYfoKljDpllOU6L/AF96YgQpyRtbqc9aSMhg20kjHAXmhANCDOQWVgcc+nrU4zlhgNuyc9DUBbZIMhsA9TUpdGOQ4TJwRmgGSNIFXIBPHIA4pjOwwCcc44NIDkkAts6YPepDwFcLuJOADQGoKCv8YY9uetOMxAUMM7ewHr1qMsrNhTkkHOOcU9VGxcYOScDqTQGo+Qgpkeox605ZM4ZhkhcemKZvU8AFXPA5/OnBxG20n5dtWSOQgKSOOQeDTY2EKnAPPTLE0DGDkhAFBAJ5IxQ0iNJEQhzjHOfzqAHSA7wck7gv6U4E/KgIG3rTCxGQZARn05zVa81K3sIjNK+xc8IDlmPtQ5KKu3oNJydkXQwZiGLHHIBPamyXCQ27mZkjUnjcQOPUVxeoeLLu6ysOLdOOQMuce/b8KxpJXmbc7s7Huxya86pjoxdoK52wwresnY9EbWdNhaIfa4SVBLENuyT7imyeIdPkGBeRg565rzqiuf6/P+VG31SHc9F/tzTWZma9QnOAM8D3p39v6dHjbexkdOTXnFFL6/Psg+qQ7noz+ILB2LG8jYk55pR4g07aR9rjGeScmvOKKf1+fZB9Uj3PSX8RablQt5GQp4JY0DxBpy9LyPDHkbjXm1FH1+p2QfVI92ejt4gsC4IvoyBgdTj61KfEmn4YfbIiO3avM6KPr8/5UH1SPc9LbxJpoYMLuPOcdeaaPEOm7v8Aj7jI5I5PWvNqKPr9Tsg+qR7s9NTxLpu0H7XED05NJ/wkenFiDeR7AcgZrzOij6/U7IX1OHdnpv8AwkmmyDDXsYz35pF8Q6aGI+2xbeh5PNeZ0UfX6nZB9Th3Z6aPEmmLyLuPjoM0i+IdLkLO13GrMc8E8YrzOij6/U7IPqcO7PTP+Ei04jJvowR0wacfEumZBF3GRnnJPNeY0Uf2hP8AlQ/qke56d/wkembTi9iHtmnR67p8rgrewhumSwU/rXl9FH1+f8qF9Uj3Z67FJHKpZHV0bjIORUqsAvPzHOM/XrXkUFzLavvhkeJvVGIro9J8cT27BLxfOjOMuoAcf0P6V1U8dCTtJWMJ4SUVeLudxk7tw4IPQ9xQSVIP8hUFrfwX0KTwOJEbpg9D6H0qYSbBuwAPrmvSTTV0cTTWjJCS3z54xiiRjIoZT05570I24EEj2pRypPQDuBVkj1+bafx6UU1XDDg8njrRXDiN0d2H2Z8/BCSxJxxxTlYuuO4/So9xJJ9R0pyF2C4BBJwPeuW51E8Ft9pwMYA6mtNEEahVGAKSCEQxgdT1J9TT68urUc5abHVCKSCilAJIABJPAAr3z4Yfsf8Aijxtaw6jrc6+GdNkAZFmjL3Lqe4jyNoP+0Qe+0iuaUoxV2zaMZTdkjwKivuzTP2KPANnAFup9W1CXHzPJcqgz7BUGB9Savf8MbfDj/n21H/wNb/CsPrEDf6vM+BaK++v+GNvhx/z7aj/AOBrf4Uf8MbfDj/n21H/AMDW/wAKPrEA+rz8j4For76/4Y2+HH/PtqP/AIGt/hR/wxt8OP8An21H/wADW/wo+sQD6tPyPgWivvr/AIY2+HH/AD7aj/4Gt/hR/wAMbfDj/n21H/wNb/Cj6xAPq0/I+BaK++v+GNvhx/z7aj/4Gt/hR/wxt8OP+fbUf/A1v8KPrEA+rT8j4For76/4Y2+HH/PtqP8A4Gt/hR/wxt8OP+fbUf8AwNb/AAo+sQD6tPyPgWivvr/hjb4cf8+2o/8Aga3+FH/DG3w4/wCfbUf/AANb/Cj6xAPq0/I+BaK++v8Ahjb4cf8APtqP/ga3+FH/AAxt8OP+fbUf/A1v8KPrEA+rT8j4For76/4Y2+HH/PtqP/ga3+FH/DG3w4/59tR/8DW/wo+sQD6tPyPgWivvr/hjb4cf8+2o/wDga3+FH/DG3w4/59tR/wDA1v8ACj6xAX1afkfAtFffX/DG3w4/59tR/wDA1v8ACsvWv2I/BF9Aw0++1bTJ8fK3nLKmfdWXJ/BhQq8BvDzPhmivZfiv+y54r+GVtNqMWzX9EjG57yzUrJEvrJGSSo9wWUdyK8areLjJXTOeUZRdmiSCd4Wyp4PBHYitO3mSXGzC5+UDFZFWLO4MEvJ+VuDnt713Yes4PlexzVafMrrc1wwhCqFywyMHoRTowBuHy+w9Kj3btyk5YknJOQR9KQvhDjK47CvauecSB3QYyTtAABHAJ+lSJIdoQhiD/OoI5ldAoBDNyeM/lT0YKm1h0yOaEwJHA+TZhWboRUgDqQ+fm7jqDUOC4CFyrg7j7jtUwcc45P3Tz0NUA/zQzEMMdx6ZFNmO2VHZUQAcKDyT60kagsduenH+GKVkMShifmLdD6UX0J6kuwyHJJIbuDxTgdnybM5zziomVIxjPIYYwetSuCrrwp3Z79PrQQKSG4wNoGeDzmniTy3UD73X5u9QrlmzgBOxzTwuSXDcLjBPSrAkZwq7lA980CR+MAAsMsSO1MikU7iV+Y9Dn+lOV2RyWAZWPXOMUAPYu7A527eTz1pTJhgQmdw3BQ3WklkMuAgALcY65p37sDBOHXg46fSgXqK7OZGzjdx1HSpQw5AQonfJBye9QTOA2M7iR26U1pE2AuSvAIPf34oFa5KY/nYggA9MngVM2V2qBsODyKhjdXUAuDgYAx/OpkIzklQy/MR/Smg1EWb5W6cnHripVVOrA7QMDPqKgMiyuQM+pIHSh3ZOhyGzznpRcRKSy8ZB/i45oDPwVdRk5wRnFQ+YWA27icck/wBKkjAlhT5GI5OemMU9QFXO5uSOOnr70+MgOFPTHQHvUOW4dhj0x6VKSUUHIVicjj0pICVpPLXOOR6ik3eWuQRhmPb296iV9/zE+YzNg4HSnOVMm0A7cZ55xTDUnAExJdsKwxtyM8Um9duAcCo4gpO9jyD90dx7UXDi3jJZgqLksSOgxmi9lcW7KusahFpVuJSVd3+4oPU9z7VxF1dy3szSytuZvyA9BUupag+o3TSsTt6Kv91ewqpXz2IrutKy2R7FCkqcbvcKKUAkgAEk8ACvfPhh+x/4o8bWsOo63OvhnTZAGRZoy9y6nuI8jaD/ALRB77SK4ZSjFXbOyMZTdkjwKivuzTP2KPANnAFup9W1CXHzPJcqgz7BUGB9Savf8MbfDj/n21H/AMDW/wAKw+sQN/q8z4For76/4Y2+HH/PtqP/AIGt/hR/wxt8OP8An21H/wADW/wo+sQD6vPyPgWivvr/AIY2+HH/AD7aj/4Gt/hR/wAMbfDj/n21H/wNb/Cj6xAPq8/I+BaK++v+GNvhx/z7aj/4Gt/hR/wxt8OP+fbUf/A1v8KPrEA+rz8j4For76/4Y2+HH/PtqP8A4Gt/hR/wxt8OP+fbUf8AwNb/AAo+sQD6vPyPgWivvr/hjb4cf8+2o/8Aga3+FH/DG3w4/wCfbUf/AANb/Cj6xAPq8/I+BaK++v8Ahjb4cf8APtqP/ga3+FH/AAxt8OP+fbUf/A1v8KPrEA+rz8j4For76/4Y2+HH/PtqP/ga3+FH/DG3w4/59tR/8DW/wo+sQD6vPyPgWiv0n8A/sXfCe4+3fb9BuNT2+Xs+0ajOmz72ceW65zx1z0GMc56v/hiv4JtIUHg9S68lRq17kfh51Q8VCLtZmiwlSWx+VtFfqlJ+xP8ABXy2z4QCAjG4arecH8ZsVx2r/sD/AA3gPm20eoCJmwEe9bj6HHNKOMpy2Ilhpx3Z+b1Ffof/AMMJ/D//AJ5Xv/gc3/xNZetfsBeD76Bhp+palpk+Plbz1lTPurJk/gwrRYiBn7J9z4Eor6I+K/7FniX4Y+FdW8RrrNhrOm6eqM0cEUqXLKzqhbZtZQF3bj83CqT2r53reMoyV0zJxcdy7pWrT6RcCaFuDwyH7rD0P+NeiWGoRatZrcRNlW4Knqrdwa8urX8NawdKv1DNi3k+VwTwPRvw/lmvRwuIdOXLLZ/gcWIoqorrdHpAj2jkEccZoblSF9c9e1MyGI5zUiMCxAGec19CeKKgwGGACKKNxO7cQSPSiuSvujuw+zPAcheTnpirenJ5kpY8qoyPrVNGCrjOfXmtPTAPJZgDhm7151aVoOx3QV2W6KK2PB/h9/FnizRtFQlW1C8itdw6qHYKW/AHP4V5bdtTqSu7H1L+yL8BbcWEHjrX7ZZ5pTu0q2lXKooOPPYH+IkHbnoBu7qR9XVBYWMGl2NtZWsSw21vEsMUa9FRVCqo9gABU9eROTnK7PZpwUI2QUUUVmaBXlfxf8feJ/B3i3wdaeHdNbWU1Bb03Fgm1Gl8tYypDt93buY8EZxivVK4HxN8VbTw9rU1hJpsk8kGB5gdR1APHFcOMx1DAwVTESsm7XPWy3KsZm1V0cHDmkldryPOdQ8c+O9S/Z1tPGdprn9l6zam4kukFjC5nVbl41XDKVXaoByBzirHxD8e+Ofhj4T8ETi//wCEk1K9vWa822SIbiLZvEQVVwpAJ+ZQD8v1z1X/AAvCx8sp/Y0mw/w+YuPyxQ3xwsn250eQ7eRmVePpxXj/AOsmV/8AP5fcz6P/AFIz/wD6B396/wAzz3SP2gvEvjDXfGNxp9tNpmh2vhy4v9MS4tlLtIjoqyk4Ocln+XJX5fY16t8HPidp3xH8K2bxanHqGs29rCdSVIyhSVl5yNoXkq33eOKyh8bbBRgaLIBt24Ei9PTp0rd8FfEKz8UalJZ22mtZMIzIX3KQcHGDgD1rfD57l+JqRpUqibeiVnqcuK4SznBUZYivQahFXbutvvO2ooor3z5AKKK3PDmlQTpd6lf7hptjG0021SxYKNxAA5PAJo2A5jxkdR8NfDnxH4qgtkkj0mxlvFWZiqylFLbRjnnHWvPpfGHxR8LfCa++IGueCPDt5pSaTHqNrb6fq1xJcStKUKKyCJcfK5JwxxjvXbeKv2sPgt4y8Lal4e1LV71tM1C2a1njisZkJjZdpAITjivH9GvP2Z9BtLy1tPE/jEW91aGzaN72+ZVjJU4UEYUjaACOg4rolluYyelGX3HPHMcuita0X8zPm/au1YaBq9yvhnwzdahaJbullpt9qDyhZZlQu6yQKQi5JJUMcjpXo/wC8caz8YpvEUUx0H7RpaxN9l0m4uXkAfd94TQx44HGOtef2MP7LunrqJi17xZ51/ClvNctdXrTeWrh1VXxuUbhnAPc12nwh+Kn7PPwSvdXvPDms6211qqxrdTail1cu4TIX5mUkYBIpRyzMl/y5l9xUszy1/8AL6P3npssTwyMkisjqcMrDBBptbej+OvCvx00fUNW8JXMl3LpzLHM0lu8W7I3bfmAycc5/CsSspRnTk4TTTW6e6NFKFSKnTkmns1swoooqSgooooARlDAggEEYIPcV8S/tZ/AeDwVeL4t0C2WDRbyUJd2sQwttM2SGUdkbB46K3A4YAfbdYHj7wpB448F61oM4UpfWrwqT/C5GUb6qwVh7itac3CRlVgpqx+WdFOkjaKRkcFXUlWU9QR1FNr1jxzXsZC8CnOcArj3qUkMvViF+8D3qjppJLhRk8HHY1fXhc/KDnBHpXv0JOVNNnl1I8smkSQ4O4E7eBjmnDLFgM8+2agLDLZK5zgADrTlZtoJIHvnpW+hkTLKQwU7vmXGQe5pQQUZd5JUgcjrTTJ5TeYWXpgHHc8UpkYHJ6EdPU1QDo5g3AADYABxwKkLllyWzsPbiopmRZdwK5IIUDPIogmJXc4HocD9aF2Jt1HZJVicblPBpBLsTICiU+3H1pS2TlgNueuaTA87AYEdOe+fSgosq7KFIXa2COPSmxNuTG/aAcnPY1HCxclFOAByOeKF6soC7QeXxyf60XFYlJCcg43fxU9SkmFYBlXkjGCx9aiWNHJ+cYznBFTL12hwMZGWoJY4gLkFQB169aZGUQudobd2/rTwgIKlsnqAOTT0iGdwTCA7RnHXuaqxNyNHDs2/5T0H0p68liSSq4YD39RQ+GO4hc/TiojIORyWIIyeAB2o2C1yZXOQAMAnr1qQqFeRkHGMkepFV4WZtp3DavGcc1KsmxsMQXYfMO3NCYmrCMSArDKHGcDvUkB3MoKtgZBzz1qNVManJ69+vFSqSwwSVPqT/KmgY/d852duOegpNjOvB2A8jHGfWmlgjsGXcCBzmnoW4DpwB8pznFVqSIhIYK/uVGOtTMrGIKpAbucdqjLBYgcZIyTuJz+FODb0ZiMDbnPXHpSQMXaEVccEE5x0NKGVwANpAPSod+MZYYwPzxT4RkuGIDZ4Hp+NFwJ95d9xOCOOvYd6xfFN8Y7FYl481sfgOuP0rWwkQkRSxU9Tnoe4rlvFcmb+NMnCoD+JJ/wFcuKm403brob0Ip1EYlFFbHg/w+/izxZo2ioSraheRWu4dVDsFLfgDn8K+dbtqe0ld2PqX9kX4C24sIPHWv2yzzSndpVtKuVRQceewP8AESDtz0A3d1I+rqgsLGDS7G2srWJYba3iWGKNeioqhVUewAAqevInJzldns04KEbIKKKKzNAooooAKKKKACiiigAooooAKKKKACiiigAooooAzNa+O/g34J+T/wAJbqMmn/2nu+y+XbSTbvKxvzsU4x5i9eufauQsf2vvgdYeJL7W4/Et+bu7UJIradcFABjoPL46V4P+35/zIn/b/wD+29fIyq0jYUFmJwABkk195lvDuDx+DhXrt3d+uh87iM6xeBrTpUGknofp5ffto/BO/wBPazk8S3oiZgxK6bcA5H/AKTV/27vgxa2ES/8ACR3e1Sqg/wBmXJ6Kf+mdfnPP8NfF9rC803hTW4YkUs0kmnTKqgdSSVwBXEa9bz3UMEEELzTyTBEijUszMQQFAHJOe1ZR4byahQnUwlT4V0advU5P7SxdSt++hZy62aufpr/w318Ff+hluv8AwV3P/wARR/w318Ff+hluv/BXc/8AxFflTqWl3uj3j2uoWlxY3SY3wXMTRuuRkZVgCOKn1zw5q3he+FnrWl3ukXbIJBb39u8DlSSA21gDgkHn2NeQsHhJOMVPWSutd0t2u53+0qq7a2302P1j8M/tdfC/4pa3b+F/D2tz3ur34YQQSWE0asERpGyzIFHyqx5POMV8xftZ/AeDwVeL4t0C2WDRbyUJd2sQwttM2SGUdkbB46K3A4YAeKfsX/8AJy/g7/t8/wDSKev0n8feFIPHHgvWtBnClL61eFSf4XIyjfVWCsPcVxYimsLVUYN2a6nZR/fUnfe5+WdFOkjaKRkcFXUlWU9QR1FNroOM9E8NXhvtJhckl4/3bfUdP0xWupAY4HBGMGuR8Dzny7yLOACrD8cg/wAhXXLjnOCQOma+nw8+elFs8OtHlqNDyQAQPqMUUNtwDxnFFTX3RrQ2Z4ERlSCPxrW08g2wx61lctkE4J59q1NNJNqMnnJry6/wnfT3LVeh/s9KG+NXhAEZH25T+IBNeeV6J+zz/wAlr8If9fy/+gtXmT+FnXD4l6n6T0UUV4x7QUUUUDCvA/i7B5Pja5OOJIo3H/fOP6V75Xifxut/L8TWkwHElqo/FWb+hFfE8X0+bLebs0z9T8N6vs865f5otfqPj+C0kOj6ZqF/4p0LSk1CAXEEV5cMjlD3xt/kajHwmsuf+K88Lj/t7b/4musvPDdv8Tfhn4INtr2k2N7psE1tPDfXaxuAHwoweeig/Qiuf/4UTcZ/5Gvw3/4MFr86ng4LldKhzJpO999Ffqfr9PNKj51icbyTUpJrkWlpNLp2MjVvhpZ6Xp892njPw9fPEhYW9tcszuR2UbeTWj8DYN+uahLjhLcL+JYf4Gs3xr8J9Q8E6FbavLqOnajZXE/2dJLCcSYfBOOBjoD39PWuj+BNviPWJyPvNEgP03k/zFdeT0LZxRg6fI1ra9+m5ycQ4vm4axNVV/ap2SdkuqurKx6tRRRX7gfywFep6doyx+BpbQKN1xayF/dmU/0IH4V5ZXt+mkPptqRgqYl/9BFYzk4tM2hFSjJM/I3wZ4DvvH/jyz8L6Y8MV9eXDQwtcMVQEZPJAOOnpXu//DvX4lcf6Xon/gS//wARXjtlb69ovxrktPDMv2fX4daktrJwVXEvmsi8t8o/Gv0a+Dvwz+JmmyQ6j8QPiDc6nKuCNKsY40gHs8mwM30Xb9TX6xnWa4nARhOjVik1ezTbb+XQ/J8myzD46U6danJtPdaJLzPmjT/2DPHFvaRRzPobyhcFvMY5P12Vxvxa/Zx1n4PaPb6hrS6ZJHdSGJBa5ZtwUsScqOMCv0V8YeItQ8PaaZdM0G88QXhzstrVkQfVmdgAPpk+1fCX7VHir4j64tsfGmjLoGnfvXsLFHRwMAAlmVmJOCMk4HoBXmZLnGY5hiYxqSioa3Wib06Lcw4iyPLMrw8pUlN1Haz1cVd9Xsek/wDBPyxEPww126xjz9UZen92NP8AGvQfFNgum69dwIAse7coHZW5x+uKwP2GLD7L8BbWbGPtWoXM2fXDBP8A2Suu+IDA+JJcdkUH64r4/NZ8+Z135tfcfpWU0+TLKC8kzm6KKK4DvCiiigAooopiPyq8WqE8Va0oGAt7OAPQb2rJrX8Xf8jZrf8A1+z/APoxqyK9lbHhvcsWZKsSpIPA4rRjLHLEYJ4GR1NUdPAMjZOMD0q4WDKvYqemDivbw2lJHnVvjY9iFYELyvXHcU/aWbcDtGOMURuhP3cEcEbsg0eZ5hZBwAO57egrrOcleNFi2HBbrnHem5eNQSDkdTSYEfJLEknjPelLNtbecZx1IyaAJFdAuCpdR0JFKpPTK5xxikY7AxABx1BPGKYCsKh0JO7+HuKsCWPMiA5Bx0BqORxHIrg9DjoSCacsgXCkDk89ee9D/MwOFHGdpPAqegEwyV2HjuWzTVwmCxHzELwPypDuZwFzg9RTpGClTt4U4IP86fmHkToVVldsYAwB701QC2T/ABZAx6moiAVzncc5UDp9aUynAGAuSDuzwD7U7kWJoZA5kIYADjrgmmyEeXhG44Odx6fSmow3EAjLDk4601SrsUYYwOu0jFFx21JxKBgAHdjilBDMAUVwcE0CNI/LUnJ2k5JqNyWIHCDA2t0JNPYgmt8RszE5GcHjp7VJIu4hlODkZJ7gVFgsFYjAzgkH/GnHcoJ3BgTlTxwKoGSKvUhjnsO1LGx81QVDDB/AimiTdxjA6ZNN8wImGyGPDAHrQBMCGH3ACQSfbmkYnGdxO3Azjg/TFMYfIAhIyORyT9Kcgd2IXsN3tigVh65mfDHkLjgEcU+Nhyijap4yKbCp+YnHTkk81GSXICHaq+3OaCSVFQSNuGCOnH+c0FlUqrZXnHXGfypMkMckP249aS6YPtY/eGBwOvpQBJuCZAGwNweK5PxJk6o5P91a6rLEbmVVYdOe/wCNcr4kYNqRI/uLXBjP4XzOvC/H8jKr0P8AZ6UN8avCAIyPtyn8QCa88r0T9nn/AJLX4Q/6/l/9BavBn8LPYh8S9T9J6KKK8Y9oKKKKBhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQB8ift/f8yJ/wBv/wD7b18o6P8A8hay/wCu8f8A6EK+rv2/v+ZE/wC3/wD9t6+UdH/5C1l/13j/APQhX67lP/ImXpL9T4PMP9/fqv0Pt39pn9oH4ifDT4z2WjeF5lm0xrS3mNgbFJRNIzOGXdt3DIUDhh7V51+1B4a0vRP2kvhrfWNpFp17qtzY3d/aQgKFmMuCxA6E459SM16Z+1B+0/4j+EPxKGh6NpWjXCmwjuFu72B3lVmZxwQ4GBtGK+JvGnxQ8Q6v4ytPGmqXf9p63BfRXavcj92WTJVdq42qMAbVxxX4bkOQY3F5XDGUqMaUfZTjdP3qrkrJyVum+t2fb5hmFCnipUJTc3zp2a0ilvY6b9tOxuZv2hvETR28rqUgwyoSD+6Wui/4KFf8l2s/+wJb/wDo2avafhz+0/4j8b/Av4h+N9R0fQxq/h9o1tkjtn8p8hc7wzlj17MK+NPjF8YNY+N3ixPEOuW9nbXkdoloEsUdE2KzMDhmY5y57+lfJ8JwzXF5lhqOJoqEcBGVKTUubmlKMGna3ZHpZm8NTw9SVOd3XaklbZJs7L9i/wD5OY8H/wDb5/6RT1+o1flz+xf/AMnL+Dv+3z/0inr9Rq/WMy/jL0PCwX8N+p+VXi1QnirWlAwFvZwB6De1ZNa/i7/kbNb/AOv2f/0Y1ZFWtjie50/gXH2u6yMjYvH412mAucDJzya4vwKCby5AJHyL/OuvYFXJJ4PQZr6PB/wUeLiP4jJXwMYxgj9aKjJJUBiPlPWiqr7oqhszwfkhSSAa19NObYEn+I1jqCrEHBzWxpmPsoA6bjXl1/hPQh8Rar0T9nn/AJLX4Q/6/l/9BavO69E/Z5/5LX4Q/wCv5f8A0Fq82fws6ofEvU/SeiiivGPaCiiigYV5P8drfnR5wP8AnqhP/fBH9a9Yrzf432zS6DYyhSfLnIYgdAVP9QK+Z4kp+0yuquyT/E+64Jrewz7Dyb3bX4M4HSPBP9seAdY1+CZ3utMuYkktlXP7pxjf65DY/CuUrofB/jzXPAd1LPo14bYzALLGyB0kAOQCCCPx612P/C9PMk+2z+EPD82sg5W/NqBg+pXuffqK/FIxwtWEeabhJKz0dn5rXc/p6pUzHD1qjhSVWEnePvJNaLR3W19b3bI/GWdP+CfgayIKtd3N5dkEY+6VUH/x6ug+CNuI/C91L3kuj+QRP65rzLxp4+1vx9eQ3Gs3XnmEFYYkjCJECckAAAc4HPXgV698JbY2vgq1LKVMju/IxkFuD+Qr7HhyUcRmvtKd+WMbK67JK/zPzfjKFTA8O+yrWU6lRtpO+7crfJaHZUUUV+wH83hXqvgTVhqGhxxE/vbb92w9v4T+XH4V5VWn4f12bQL8Tx/NGw2yR9mX/GonHmWhpCXK9T4b+JEd34V/af1t7ExR3kPiVrmD7RnywzT+Yu7HO35hnHOK+q/jf+0B8Z/gJY6Tc67ZeDbyLUXeOL7Al0xUoATu3MuMg8Yz0NfMv7WKm0/aA1rVrZHW2uzbXUEjKVDMIYw2D3wytn3r6asf2yPg98XtMh0v4geHmse+zUrVby3RiMZSRRuB/wBrav1r9Gx1OVejg8R7D2sFH3kt9lbzPzTBzjRrYrD+39nNy91vRPUofCP9sfxr8QvGHhvSbrTNHii1K5SKZoY5QyoT8xXLkA4BxnNZ/wDwUQ1HF94etQceXZTSEf7zqo/9BNd5pPj79nr4QRR3/hq3tLu6ALwNYxPdTLnsryH5OvTcK+ZP2ovi/F8aNdi1Cx0+axgit0tIoZGDySHzC2SAMAnIGAT0681ll2GU8zhicPh3SpRTvfTWx5+ZYx08ungsVilVqzmrJa2V1pofXn7J9iui/s6+EhKQga3muWY+jzSOD+TCqmsagdV1S5uyCBI5Kg9l6AflioPCesNpXwm8I+HoFaJrfSbWK6JG0+YIl3Lj/ezmoa+HxD9piatV9W397P1HDe5haVJdEl+AUUUVibBRRRQAUUUUxH5V+Lv+Rs1v/r9n/wDRjVkVr+Lv+Rs1v/r9n/8ARjVkV7K2PDe5Zss72wT07VdVwSV3jnnHeqVizJIxU445q2GJ5BHHPHWvbw/8NHn1viY7cEc4AGQCSakSQqMkYX1qIYcHHDFfzp8JIXByy44AHT8a6luYj2dHIBLEty1OkIXAVTtxgnrTCwC5KBQTjB/nQJ2ZcAYGduRTILK52KXX5cY560wKGXHIOTnJxgdqbkrDhgCpIwAO3oaHlVWyRxt2hemKrQCQIjqwJIA5pdqFFKMCuR7mmGQBtgAGTkH1pWQx9CAT1J6VIEikEA5yR0p+SzN1wex9ajRcjIcgMMbQO1KI0bgMWweQTzmrAkVQrhcgj1zx9Kc0ezHXH90dKjLM52cL/d5o8jgZOAG6HkGgCRRuJYDIxyfSjaWAOc8En19qaXOdoZhgYGfWnFmUKWyxbPI7UEj3fcqgkqeufWkMh2AYyoXcPlORSEDCjBwPzqVcNHhgfl4DZp7ivYUbigx824dCKGQbBkYYkAY71reD/B+p/EDxRY6DoNob/WbxylvbeYke8qhYgs7KB8qk8kdK3PDvwh8aeLvGWoeF9G0Ke+1yxkkgureBlKwOjlG3SFtigMCNxbB7E1EqkI35pJW136FRpykk4pu/5nHmQIwGAxPBHpQRuUuoG0ED8/evTfHH7M3xO+G+hzav4h8K3Frp6EeZdRTw3Kx+7eU7bV7ZbArmvh/8M/FXxU1Y6Z4Y0WbWLyIK0whAVIhnhmdiFUHBxuIzioValKLmpJpdblOlUUlBp37HMRyEhlVTuX05qRGwBnPU59/avSfGX7M/xJ+Huky6zrXhe4ttJjJMl1bXEN0seO7eU7bV92wK5/wr8H/GXjTwnq3ibRdFk1LRNHP+mTxTRBosLvPyMwdsKc/KpojXpW5lJNbXvpcHRqX5eV33tY5kMQ4JIHcAdqQSbZG2gZ6YB5rZ8CeBPEHxO8SJoXhuzOpalMjSCAzRx/Koyx3Oyjge9Zd3HJp97PbSx7ZopGidAQcMpwRkcdR2rVTi5ct9UZOMlHmtoxiKY8/KBwWxnv70g5yS6+xzxSlA7ckbSMDHb1FRoxO0DByeuMcVYiZ4iu0YLYPXsM1yniIY1Nsf3RXVSsWlAGcngdz7ZrlPEOV1IgjBCqMYrgxn8P5nVhvj+RmV6J+zz/yWvwh/1/L/AOgtXndeifs8/wDJa/CH/X8v/oLV4M/hZ60PiXqfpPRRRXjHtBRRRQM/Mrxt+0J8Yv2mvilrmkfDbXLnw94d0x28hbG6NoWiViqySSjDFn67c4HTHBJqf8Kv/ak/6KRq4/7ma4/xrP8A2Bv+Ry8bf9e8X/o1qyD8Tvjh8RPip4p0DwZq0txJp9xcOtnH9miWKBJvLGGkAzgso6knOfWvahRy7D4CGMxrleTezSSt6nGpY/FY2WDwUU+VLdNt3XkdP/wq/wDak/6KPq//AIU1x/jSf8Ku/ak/6KPq/wD4U1x/jWh+zF8VviHrfxm1rwl4z1WS8axtrhJrWRIj5U8UiocMgAOCWHBIP5Vy/wAUPjB8V9U+PWueEPB+syQ+TOY7WzjWBF2pGGJLOOuATyfYdhXb9Xyd4WOLiptSdkk1e5w1MRmNGvOhVcI8iu200klv6WNj/hV37Un/AEUjV/8Awprj/Gj/AIVf+1H/ANFH1f8A8Ka4/wAaqfBX4tfFKL9oLS/BfjPWJJw4kW6s5FhYD/R2lQhkGM8KeD3INd/8btN+PV18QbuTwDLInhoxRCHbPaIN+wb+JDu+9nr+FbUsvy2th3iKdOo7PlaVrpmMsdjI1Y03OnaS5k+jT2szi/8AhV/7Un/RSNX/APCmuP8AGj/hV/7Un/RSNX/8Ka4/xqr/AGH+1h/z8Tf+Ben/APxVH9iftYf8/E3/AIF6f/8AFVn9Rwf/AEDVvuNvreJ/5/0vvLf/AAq79qP/AKKPrH/hTXH+NQ3nw6/af0+2eaT4j6z5a4zt8S3BPJx61Gmi/tXeYN0823PP+l6f/wDFV7B+05afEy6sNDPw8Z1hUzf2ntlgTIJi8rPmkZ53/d/HtXbQynA1YSqOjVXLbTq7voc1bMsVTnGmqtN3vrfRWXX16Hh1n8c/jv8As3+INM1XxF4lvvEmkXEoWWz1O+a8imUcsgZ8tG2OQVx07jIr9SfDOv2vivw3pOt2RZrLUrSK9gLDBMciK65/BhX5dfteCYfD/wANi4/14uh5nT73lc9OOua/Rr4Bf8kJ+HH/AGLem/8ApLHXh5zgaeW4+WHot8qSavvqj0stxc8dg4Yioldtp220djvaKKK8U9E+RP2/v+ZE/wC3/wD9t6+SbeZre4jmTG+Ng4z6g5Ffop8fvgD/AMLy/sH/AIn39if2X9o/5c/tHm+b5f8Atrtx5fvnd2xz5D/wwF/1Pn/lH/8At9fpeUZtgcPgYUK07PW6s3uz4/H4DE1cTKpTjdO1mfOXxM+J2t/FrxINd8QSQyagIFtw0EQjXYpYjgd/mNeeeJuNPT/roP5GvtH/AIYC/wCp7/8AKP8A/b6qan/wTz/tG3WL/hP/AC8MGz/YuexH/Px710yzTKqGE+q4WSUUtFZ/5GMcDjZ1va1YtvufI3hn4ua/4S8A+IvB9g1sNH14q14JId0hIxja2eOg7VxYx3r7Z/4dr/8AVRv/ACh//dNH/Dtj/qov/lD/APumvjqE8Bhp1KlFJObvJrq9j3J0sTUUVLVJWXkeJfsX/wDJy/g7/t8/9Ip6/Uavlz4L/sQ/8Kg+Jej+Lf8AhNP7W/s7zv8AQ/7K8nzPMheL7/nNtxv3fdOcY4zmvqOvOx1WFaqnB3Vjuw1OVODUlZ3Pyr8Xf8jZrf8A1+z/APoxqyK1/F3/ACNmt/8AX7P/AOjGrIroWx5z3Om8DnF3c/7g/nXYcK245PoK47wO226uT6oB+tdlABvIIzzX0eC/hL5ni4n42G0nBJyCcUUM24sM9OmB0oq6+6KobM8EjJBBHU8Vt6d/x6/8CNYYHYc55zW5p3/HsOO5ry63wHow+Is16J+zz/yWvwh/1/L/AOgtXndeifs8/wDJa/CH/X8v/oLV5k/hZ0w+Jep+k9FFFeMe0FFFFAwpGUMpBAIIwQehFLRVckpLYXOovezOb1T4d+H9WLNLp0cUjdXg+Q/+O8fpXMyfA7Tjdh01G4W2/iiKqW/Bu35GvSqK8evkWCxMuapQTfkrfefTYTirNsFHko4lpbWbvb0uc5o/w90HRCrQ2CSyryJZ/nYH1GeB+Aro6KK9ChgqeFjyUadvRHi4rMcRjp+0xVVzfmwooorezWjOPfYKKKKQzJ8S+FdI8YaebLWbCG/t87lWVclW/vKeqn3FeO67+yL4evpGfTNVvdM3fwSKs6D6Z2t+te8UV6OGzLE4P+DU5V26fiedicuw2M1rQTffr+B4Npv7LK2scUM/iqZ4EGMW9ksbEf7zOw/SvQ/B/wAHvDXgu4S6tbRru/T7t5et5sin1XPC/wDAQK7aita+cYzFLlqVLp9Fb9Dlw+SYDBy9pSopPvu/xCiiivJPbCiiigAooooAKKKKYj8q/F3/ACNmt/8AX7P/AOjGrIrX8Xf8jZrf/X7P/wCjGrIr2VseG9ya2zlsY6d6soCqkM23PvVe1XLnOCAOnrUm0Kx35OOQc4Ar2cN/DRwVfiZZVipAAOWXApykowLDJUfh9KYzbVDEHsF/DpxR5zHIJ5HP1rrMCVmfaVB35IIB+tOydqYXnJ3VHAxQMR26c84oClCckksc8mncVh5k/eDJAVjkYqYhHfD53jnkdagPzhATjuD0H408AyTk7ySMckUxku3bwSRkYziiNFELAEnrnuRTVLbjkE/KckUApsGBgAcgcZ9fxoIJI0EpUB2BAzkfyp6IU3EHHUkjv/8AXqLdsyQQRkcYxipyxjxkkE/MAOgpoBseAuDkMRlaBNhMHAy2OT6etKjEEnHzHlW9KXygGkkbqw5OAeaoBwQM+DlW+veglgcFgWHT/PrTMFmwCBxnI6mlHK4Dfd+6SM4/OgB75EwUEsW6EmpinlYV3AkZsEVTxmYbWJZjhsCrtrMtnfQztELkRyKzRyYZHwclWHoen40rhboew/sc5P7S3gjpgXU3Pf8A495K9Q8EfE7wn4f8cfHTwV4o1W68KDxNq11FD4itIWkMBW5kG1gvzAHdnPTG7JXrWjoXjr4CaX8UNH+K+neIJvDdxZWe9vBFhoToDc+UyHEqL5QB3Z46nksORXm/gT4mfDPxe/jvSPiNpP8AZJ8R6lLqln4ls7NZ7vT3Zy5jLBS5TOPlXIOWyBnNeBUvXlKbi0rJbap36d7Hr07UYxgpJu7e+lrHo3hL4TeOPhP4b8Wah8MPFvhX4qeG72we31HS4pWdthHMn2dXwHC5x85JBI2nisfxvdT/AAt/Y18C2Xh+d7NvFt7Lc6rdW/ytOoBxEzLg4wFGO4Qg96Z8M/E3wl/ZvudV8VaJ8QL74geIWspLOy0220ebT4cvj5pWkyCAQDweOeCcY5nwD8XPCXjT4VyfDb4mT3ekWtvdvf6P4gsYTObKRyxZXiHzFCWbhcnDY4wDUxjUcudxbimm9LN79OtvQrmglyppSaaWt/x8yL9jXxpf6H8ctD0WCZpNG1wyWWoWBJaKZCjYLL0ypA5I6ZHevfvhZ4gtv2efA/xkurOBL7StG8Yw2zW4xzau8SMgznJVHIHqV7V4/wCAPEPwh/Z3v5vF2leLLn4j+Koonj0u0h0iawt4JGUr5srS8ngkfLnGTweo5/Rfi5pFz+zl8T9E1fVceL/EWtwajFbmCQ+cPNieRt4UouMMcFgeOKqvSeIqOUYvlbinpa+u9hUaioQSlJXV3vfp3PePhf8ACG0+H37V+g694bK3PgXxJp11faVcRDMceYstDnoNucgf3SB/Ca+KfFLFfFesFsH/AE2fHP8AttX01+x/+05ofw5sJfC3ju7aPRrNmvNJ1FoZJjaysrK8e1FZgrBmI4wCWz1FfLviK6S+8QalcwvvhlupJIm5G5S5IPPIyD0rrwkKsK01UWyST76v+mc2JlTnSi4Pdttdih5h2kjpkgEdqcA0iYA5HJHrQijd5g6k4LAZA9Keigbjvw2DzmvVVmeXsIzF5VChhtUEknOK5vxH/wAhRun3V6V0o+VACBngcDnFcx4gIOpMRn7q1x4z+H8zpw3x/Iza9E/Z5/5LX4Q/6/l/9BavO69E/Z5/5LX4Q/6/l/8AQWrwZ/Cz14fEvU/SeiiivGPaCiiigZ+V/wCwL/yOfjf/AK94v/RrVl/8Ks+OPw7+KfinX/BelPA+oXNwq3iG1lWWB5vMGFkJxnah6AjGPWuv8cfs7/GP9mf4oa5rHw10W58QeHdTdvIaxtjdlY2YssckQywZOgbGD1zyQOJ0X9qH45eI9cvNG0rRxqWr2Yc3Nja6O0s0O1grb0GSNrEKcjgkCvZhXy3EYCng8apXi29ErO/qcihj8NjZYvBNJyS3burLpY7v9mP4TfEPQfjLrXi3xppT2Zv7a4aa6kkizNPLIrkhUJxkhjwABXM/FD4N/FjTvj1rni/wdpDS+dO0lrexyQMCrRhWBWRuuCRyPcdjWf4i/aY+O3hG1NzrehtpNsJBEZb3R2hTec4XLYGeDx7GtfQ/jR+0p4m0m21TSPBOoanpt0m+C8tPD8ssUq5xuVgCCMg8iu1YrJ/qkcJFzSi7pq17nDUwuZVcROvWUJc6s07tNdfvJfgr8I/im/7QWl+NPGmjvAIxI11eO8Cg/wCjtEgCoev3RwOxJrvvjbqHx7t/H90ngGORvDYii8nbDZsN+wb+ZBu+9nrx6VxX/Cyv2pv+idav/wCE1N/hS/8ACyv2pv8Aoner5/7Fqb/CuilmGW0MO8PTqVFd8zatdswlgcZKrGo4U7RXKl0SXZdCr/a37WP/ADxl/wDAbTv8KP7W/ax/54y/+A2nf4VZ/wCFlftT/wDROtX/APCbm/wqO6+Kn7UFlay3Fx8P9Vt4IlMkksnhyZVVVGSxJHAABJNZf2hgv+git95t9UxX/Pil9wxdW/av8wboZtuef9H07/CvX/2nLn4m2+n6J/wrxWMLGb+09scD4GY/K/1oyOd/3fx7V4B4V/ac+OnjpLl/DmiHXVtdomOnaO04iLZ27tucZ2nGeuDVHxd+1B8Z/D9tFD4h0tNIjusiP7ZpZh8zaVLbd2M4yuceorsoZxgKUJU3Wqvmtr1VuxzVssxlScaipU1y30to7rr6dDf/AGvTMfh/4bNx/rzdDzOn3vK56cdc1+jXwC/5IT8OP+xb03/0ljr839M+Bvx0/aY1nSLDX/D93oGhxOsj3+oWZtIURhy6q2Glbb0C569gSa/UvwzoFr4U8N6VoliCtlptpFZwBjkiONFRQfwUV4mc46nmWPniaKfK0kr76JI9PLcJUwOEjQqNXTbdttWaVFFFeKegFFFFABRRRQAUUUUAFFFFMR+Vfi7/AJGzW/8Ar9n/APRjVkVr+Lv+Rs1v/r9n/wDRjVkV7K2PDe50ngnP2y4A/uj+ddcGzKxyRz19BXIeCzi7uP8AdH866tSScc5brX0WD/hI8bEL94ybIYsQOR3xjNFMRfLUjphuBRV1t0OhszwpQFGcjOa2dLObUH/aNYgA28jPFbelcWajnqeteXW+E9GHxFuvRP2ef+S1+EP+v5f/AEFq87r0T9nn/ktfhD/r+X/0Fq82fws6YfEvU/SeiiivGPaCiiigZ5tH4N074mfGzUtH8QCa60yx0lJ4LZZWRFkZ1Bb5SMnBNdx/wy58O/8AoDy/+BUn/wAVWJ8Pf+Th/EX/AGBIv/Ri17Lr8z2+hahIjFHW3kZWB5BCkg191WxmJoOlRo1HFcsdE9NUj4+jhaFb2tWtBSfNL7l0PNP+GXfh1/0CJP8AwKk/+Ko/4Zd+Hf8A0CJP/AqT/wCKrkPhb8GovHHgDR9dv/FniiO7vIjJIsGohUB3MPlBUkdPU11f/DN+n/8AQ3+Lv/BmP/iK6quJq0akqUsZK6bT0e6+Zy0qEKsFUjhFZ6rVf5Ev/DLvw7/6BEn/AIFSf/FUD9lz4d/9AiX/AMCpP/iqx3tte+BnibRS+vXniDwfqt2lhImpMHmtJn+4wYAZUkEHgD2zg17hnNceIxmOo8soYiUoy2ab+5rujsw+Fwda8ZUFGS0aaX3o+cvCeg23gn4t+K/DWlNLFosFtb3EVrJKzhHZVLEFuecmvRq4s/8AJw/jL/rwtP8A0Fa7SvHzpuVeM5O7cIN923Fanp5QuWhKC2UpJL5hTokMsiIDgswXP1ptV5bgreW8SnGXUtj618XmWYU8tw7r1NtEl3bPpsLh5Yqoqcf6R0uo+F7ewna3l12xt51xuSQNlcjI4+hFZsnheyk+94osT/31UXxH/wCRwvv92P8A9FrXM1+GZ1xBUxGKqUq0LqLaS5mlb5M+9wWXwpUoyg7Nrtc6lfCdgrZHiawB9t1WYdBsoiA3ibT3X0IbNcbRXk4TPpYKanh6fK1/elb56nXWwCrrlqO/yR6DaeF7O/lWK31yzmkY7VVQSSaw7qA2t1NCSGMTshI74OKg8DSFPFemgHhplBHrV3WP+Qte/wDXZ/8A0I1+48O5086wzqyVmnZrz7nwmZYJYKoop3T1RUooor6w8cKKKKACiiimI/Kvxd/yNmt/9fs//oxqyK1/F3/I2a3/ANfs/wD6MasivZWx4b3JIX2kj+8MYxnNW3xkFgSCOnpVSHJbggdDmrplEZBwuR69xXs4b+GjgrfEIzGQZQYBJpseJQ5C4I45PpQZGTBB5wVI6AipFjC9DgHr9a6jHYkXG7c+Ap6YGeaegDEsQcZ65xUXptPHYD0pwGzqMnPc8jNXqIVCNzZDE7cdOMVOyKiAoMHjjnOaiCF2JyG7DmgqqHAdj0zk8e9LYVh6yEH5SCTxnH+NSbdpxwSAAMkY5qB5NkmAMrt5x15pybfmG3dtwOTn6U/IZYjUMrByPmxjr2pWLs4+YHd8vHYDtVcNIBkMNpOMf/rp6q3LB1K4weO9O5NiaJHkIJycZAxj9adKA5Bb5RgYdDnP4VHFJzhFIGcGlkkGCE4BI59KmUuWLl2Go3dh7uEiwAG3c9MHrSiQ5KAAY+bnpX0l4q8VWnhHT4ru7inljkmEKiAKWDFWbJ3MBjCnv6Vyg+OOglSfsmocHH+rj5/8iV+K4PjzNsxp+2weVucLtXU9Lr/t0+yrZDhMPJQrYpJ9mv8AgniokERLgkLj6nP86fKyBY22FmJz8pI/PFe0r8b9CYZFpqP/AH7j/wDi6ePjVoTdLa/JHby0z/6HXeuLOIf+hPL/AMD/APtTn/snL/8AoMX/AID/AME8VQnI5AJyQMY4pPNCq4wOwyMdTXtX/C7ND/59NQx6+UmP/Q6afjdoQ62moDPT90nP/j9P/W7iH/oTy/8AA/8A7UP7Iy//AKDF/wCA/wDBPGQp2ADO4KTkfXrRISQGOdzAc54Fe0H42aGBn7LqB7f6tP8A4ul/4XTovANpqAJ9Y04/8fo/1u4h/wChPL/wNf8AyIf2Tl//AEGL/wAB/wCCeMbw6AkjngYPNKE4RBy3YEc17OfjTogBP2W/AHfy0/8Ai6Q/GrQ+ptdQweM+WmP/AEOj/W7iHrk8v/A//tQ/sjL+mMX/AID/AME8aO9dxBGOBjBzmnBGAwpGSwOWP6V7L/wufRBz9lv/APv2n/xdB+M+iDP+i35A7iNP/i6f+tvEP/Qnl/4H/wDah/ZOX/8AQYv/AAH/AIJ5r4U1q30XW45Z1EtnIfKuInXcrIfvZB64OD+FejeIPhLY6gpuNJlFpI3zCJiWib6Hqv4ZHtU0fxi0eYxqlnqDGRgqARISSTgD79d2rfKCw25GSDjIr8t4o4qzzC46jjIYeWGqNWtzcyml/dt0Pp8ryrA1aE6EqiqxWt7Ws/U+dNY8O6j4fuAl/avAmeJB8yMfZhwfpXEa7j+0DgYG0V9I+KviJommQy2pC6rKykNboAyfRmOR+HJ9q+dfFlzHea1NNFbR2kbgEQxElVHtkk/57V+xcO55medYL2mYYV0trS2Uvk9f0PkcdgcLgq/Lh6ql3XYx69E/Z5/5LX4Q/wCv5f8A0Fq87r0T9nn/AJLX4Q/6/l/9Bavo5/Czlh8S9T9J6KKK8Y9oKKKKBnwt8Yv+ChHiW1+I2oeE/hV4TtfETabK8Et3dQzXRuHQ4cxxQspCAggMWOeuBXzD8O/FHxh+GPxK8QeN9L+Ht5Pq2trcLcQ3mi3bQL5syzNsCkMMMgAyx4znJ5rtv2B1Enjbxq7fM4t48MeTzI2ea0fF37bnibTPGWtaRpHhWyu4NPupbdS5leRlRipY7SAMkZ6cZA56169LA4T6nDF4uq48zaSSvsccsViPrMsNhqXNZJt3tucD8WvHHxm+MmhtpetfDy6t7droXhaw0W8R94DADLMwx857Z6c16F8Nf2ov2gvhZ4D0bwppPwyiuNP0qHyIJLzQL95SuScsVlUE8noBXafs7/tR6v8AGDxte+HdX0K10ySK0e4WS2ZwVZGVSrK2f73tjHvxzvxZ/bI17wT8StW8MaP4cs72OwkEPmTtI0kjBQxICkYHPTnpXY8sy2OHWK+sPlbstP0Od4/GyrPDyoK6V3rpY0/+G4/2k8Z/4Vbp/wD4Tuo//HqP+G4v2k/+iW6f/wCE9qP/AMeqt8Ff2vdd+I3xO03wnq/h6zsFvlkAlgaRXjZYmkBIYnIIUjt1Br6lr0MFkGEx9N1KFd2Ts9LanBis5r4OahVopNq61vofMX/Dcn7Sff4WWH/hO6j/APHqqa1+2d+0Zrmj32mz/C6zSC8gkt3aPw9qAYK6lSQTMRnB4yDXT/F79sLw38Pb6fSdGgbxJrURMciwvtt4nHG0vzuIPUKPbINeeabrX7Tnx0US+HNDutC0mdfknigW0i2+qyy/Mfqpr5zNv7FydN4nGWtvsl97aR7OBnmGOSdPD2T7t/lY88+B/jL4x/s/W+sx+Hvh9d3aao0TTnU9Fu32mMOF27GTH32znPQVy/xr8cfEL4oR6X/wlPhZtJTTfNMRg0+4hz5mzdu8xmz9xcYx3619G2//AAT4+MXjhEfxh4/toh18ua7nvHX8OB+RrotN/wCCVVtgHUviHPKe62+ngfqzn+VfkuM4+4VwrcVir+l3+Sa/E+ypZRmVRXcLf15nM/Cv/gpr4i0vWdM0vxv4b0v+wVEdvJcaVHNFcQIAF3lXdg+AMlRtz2Pavu+w+NHw/wBUCfZPHHhy53AMoi1aBjg+wfNfKcP/AASv8Gqv73xlrTN6rDCB+oNQ3X/BK7wsy/6N431eNuxlt4mH6YrzY+J3DidnXf8A4DL/ACNJcPYyWqh+KPtiy1iw1JVa0vra6VuVMMyuD9MGrlfnvqX/AASx1CDL6T8Rl3DkLc2DJz/vK5/lVGb9mD9qP4WqknhTx3PrFtb8x21rq8gXA7GGbCn6civawvH3DuMaVPFRv53j+aRxVMjxtNXcG/Q/Raivzm079t744fBXUIrP4o+DjqdqWIaa4tTZzEDg7ZEHltj/AHT9a+w/gT+0r4L/AGgtLkm8O3rRajbqGutKuwFuIB/e25wy5ONy5Hrg8V9xQxVHEwU6Mk09mmmn80eNUozpO01Y9WooorqMQooooAKKKKYj8q/F3/I2a3/1+z/+jGrIrX8Xf8jZrf8A1+z/APoxqyK9lbHhvc6TwQcXlxnug5/GuqfAAIO4lhjmuS8HEi6uBgcoB+tdSDyMnkHivocH/BR42I+Nk+4M7c9qKjJ4OB3yaKqvui8Pszw/a0eT1x6VsaZ/x6DHqaxi4xgHmtnSyTZjP9415tf4Tvp7luvRP2ef+S1+EP8Ar+X/ANBavO69E/Z5/wCS1+EP+v5f/QWrzZ/Czph8S9T9J6KKK8Y9oKKKKBnJ/D3/AJOI8Q/9gSL/ANGLXsXib/kXdT/69Zf/AEE1478Pf+TiPEP/AGBIv/Ri17F4m/5F3U/+vWX/ANBNfYYn+LR/ww/JHy2H/hVv8UjhP2ftQtYfg/4ZR7mJHW3bKs6gj527Zr0P+1LL/n7h/wC/q/414D8HfgH4M8VfDXQtV1LT5Zry5hLSutzIgY7mHQNgcAV2X/DMfw/PTS5//AyX/wCKrtxtPAfWqvNVknzO/urv/iOfB1MasPTUacWuVW9636Gb8ZvEFl4u1jwr4O0mdL/VJNVhvLgW7Blt4YiWZmI4BORgfX2z7OOMCvANQ8Dwfs/+LtG13w9I40PVbuPTdRs7g72XeTskVjyADnIz6de3v4zxxXJj1SjTpRw7vCz1e9+t106d+5vgJVJVarrq072aWyXTU8EP/Jw/jL/rwtP/AEFa7SuLP/Jw/jL/AK8LT/0Fa7SvMzn+NT/wQ/8ASUejlX8Kf+KX5hWVE5fVIye8q/zFatcf4o8XaX4DsZdY1m5+zWNvIu5sbmY7uFVRyWPpX4hxuqs44aEE2nLZfI/QsicYyqyk7WR23xG/5HC+/wB2P/0WtYFhEs19bxuuUeRVYZ6gkZrntL+OPhb40+INSu/D09wDGIy1veRCOUKFVd2ASCMjqD6ZxXSaYCNSs8g8zJj/AL6Ffk+YUZxzSpGcbXl17Nn2GHqRlhoyg+h2XjbwjY2PiDSLDTofsq3ZCsd7PyWAzyT0zWxqmk+DNDuI9KvYHjmaLcbgluOuCWB6nHQDFU/izcSWeuaRPCdssal1PowYEfrVuz8TaJ41aGx1ux8jUM7FbBGWPYMORz2PFfb+zwdPG4jDwhBTbjy8y916LRdmeNzVZUac5NuOt7PU4/wv9nXxxYi1Z3thcjy2kADFe2cVJfuX1O+z2uZB/wCPGtRfDQ8L/ETTLeNzJBJIJIy33gOeD9DWTegrqV/nvdSH/wAeNexwhSrYavUozVrOV0tl8Ox5+cyhUpxmnfRWf3kVFFFfrJ8eFFFFABRRRTEflX4u/wCRs1v/AK/Z/wD0Y1ZFa/i7/kbNb/6/Z/8A0Y1ZFeytjw3uSRPsDEYzx1NTFhuHHQc5qvuCAkjPIHWrQk7ACvZw/wDDRw1fiEbzZlUEjKls/TuTSxzEN1JAGAabJz8452nnJpYXG3gDFdPUy6EynflCCOnfBqQhd2AegxzyTUURIfcflJIxj1z1qRpPnyR8/PPTr7VRBJGAw6nrnHcCk2ESNll25+Vu2DUaElsu4A7DHNPlYOrKo59cUX0BoTzU3LvBbdxkCphGYJ94O9G6knoKr+YBkEDjvipkZtiZ5OenrihAwLs+0AsQec47VJ0jJ7rnBx0zTSSys4GGzyB2FNYgkOhyqk5A9arUCRQWUENtbrwepok3RZJdpOe3Qe1MZskMAACKljClSWBPp6Gsan8OXoVH4ke5/G1gvhS1J/5/U/8AQHrw8A+T5i4ba2do7/Svbfjln/hErTHa+TP/AH7krxFVK8fKd3OR1+lflvhp/wAiNf45fmfTcSf798kCNxhiVdvmP+FO8xQVBQgkYU+pPvTCpD5G0Y6Zpdn8YwvXHcV+sanyuhO0oUBMZY8HJ6GkUkgnO1gu0gj+VRIucDAbcc+mfamxr86gghVJGB1I7U7iLEDAnAbJzz7GiaUDCKN+cZLcUzzQowqMSp7inuxeEADaRjJHfFHQVtRWykewAg5zgCnMSrFsYwMbSPyP50qk+XuIyQMfWm7QUfKlsAA57fSqEP8AMCxZYruz1x0pY8uuWUsv+yMCmphRuA+UD+IcmpFABUADBB4/ChA9DX8KataaLqcd7dQNcC3BaGFSBuk/hLE9AOT0JyBVrxJ4+1fxG7RSy/ZrRsqYICVU/wC8erfice1c8HVdxYdgBgU0NkjaOQcc814lXJcDiMYsfXpqVRKyvrZeS/pnbDG16dF0IStFu7S0v8x6KEUEEkg4we1c9rfF8RzworfKsMsSNwP3vasDXABfnacjatdmL0pWXcyw/wAZn16J+zz/AMlr8If9fy/+gtXndeifs8/8lr8If9fy/wDoLV4k/hZ6sPiXqfpPRRRXjHtBRRRQM/K/9gX/AJHLxt/17xf+jWrgfCvxhuP2f/j1451eTQW1WS5mvLT7NJMbcruuFkD52tnhBxjkNnNbuk61rv7Evxg8V6brXhybUrS6LRW8rP5AniDlo5kfawIKnleoJwcEEV2p/wCChNkTz4FmP/cTH/xqvZ5cBj8tp4TFVuRxburN3v6HLSrY/LsxljMLT5rpWd0rNLzuc/8Ask65N42/aS8VeJhYtaR6hb3l28YJdYTLOjhC2BnqRnAzjpXJ/ErxhcfCr9q7X/EUumPeeVcSMsDMYvMWSHaGDYPHOc4OcV6aP+ChVkOngWYfTUx/8aob/goVZZ/5EWY/9xMf/Gq9CMcshgoYSGKs4Sunyv8AI8zERxuLxNWtiMPeNSLi1zW0e+qOC+Cfi+b4oftdaT4mTTnsUm81pIVcyCJVs3iBLYHUgdhyQK+x/jZql1ovwj8XXtnM1vdQ6bM0cqHDKdpwQexHrXzqv/BQqy/6EWYfTUx/8arn/iH+2/beOfAuueH18HyWTalaSW32htQDiMsMbtvljOPTIr08Hj8BgMHVoxr80pXd+VrVnnVcvxVavQ5aHJCmoxtdPRP/ACOm/wCCeOrfB+w1K/HjA2UPjtrkHTbjWNv2fytq4ETN8qy7weWwSCoU/er9M1YNGpTBUjKlehHtX5q/si/sR+Fv2m/2e9S1u51O80DxTaa7cWcGoW+JYjGsFu6rJESAQGdyCrKfm5JAAron/ZQ/ao+AJK+APFLa/pUbERW9jfDaFHQm2n+QE+ilvrX8TcX8OYTP8yq1KeO5Kq0cal+X/t19F9+p+8ZbjKmEoRTpXj3W5+hPNLX54N+1j+1B8OlFv4o+G7X5iOJLi60SdSwHU7oWVPxAxVu3/wCCo2r2KiLVfhgy3C8P5d+8Qz/utExH51+a1vDbPY60FCqv7s4/rY9yOdYV/FdeqZ+geBRgV8Cj/gq5Zrw/w0uc+2sAf+0KH/4KsQSDEPw0uCffVwf5QVy/8Q54m/6Bv/Jof/JF/wBs4H+f8GffdJzX57XX/BSrx34i/deFPhdG1weB5rT3nP8Auxqh/Wo2+K37Y3xeAj0DwjdeHbdhhmt9MW0BB777kkj6qRXdR8Ns4tzYqdOkv701+lzOWdYb/l2nL0R9s/E288MWWgXMniqXTYtH2Fbo6oUEJHYNv4J9utfmT4J8WeF9B/be8L3fwreaz8N3etWtjtbcI5EmdY5winnyyGOA3IPIxgY9k03/AIJ2/FT4jajban8WvHflIx3eQtw9/cgEglAzEJHx0K7gPTFec+NvhH4f+CP7cHww8L+G4plsItR0eUvcSl5JZGuBucnpk4HAAHHAFfr/AALg8tybHyy+jjXWrON2o3UEk1r5vz7HzOb1K2Koe2lTUY30b3Z+ptFFFf0IfABRRRQAUUUUxH5V+Lv+Rs1v/r9n/wDRjVkVr+Lv+Rs1v/r9n/8ARjVkV7K2PDe50Xg7H2i5J/uD+ddXhUGCSQa5Lwif9IuD/sj+ddVnLgkcg5r6LB/wkeNif4jFwAx4wewooB3NkjnNFFfdF4fZnh/GeDx0+lbWl/8AHoB/tGsMZPJrc0o5tAR/eNebW+E76e5br0T9nn/ktfhD/r+X/wBBavO69E/Z5/5LX4Q/6/l/9BavOn8LOmHxL1P0nooorxj2gooooGcn8Pf+TiPEP/YEi/8ARi17F4k/5F7VP+vWX/0A14x4FuobX9ofXzNKkIbRYtvmMFziRemete2SalYTRsj3VuyMCrKZFIIPUda+wxakp0ZJXtCH5I+Wwsly1Yt/al+Z5P8AAr4g+F9J+E3h20vvEWlWd1FbsJIZ76JHU7ycMpYEfjXef8LS8G8/8VZon/gwh/8Aiqx2+Ffw3Y/8gLRsn/ZX/Gk/4VT8Nx/zAdGP/AV/xrWv9Ur1Z1Wprmbey6v1MqP1uhSjTTg0l3Zx3xE8Yad8V/Efhzwj4YuE1cw6jDqOoXlqQ8MEMRyRvHBZiQBg47d69yX7orB0LTfDnha1a30mHT9OhY5K2+1QT746/jWp/atl/wA/kH/f1f8AGuPETU4wp0oNRj33be7OzDRdNyqVZpylvbZJHhx/5OH8Zf8AXhaf+grXaVw8M8dz+0J4zkikWVPsFoNyMGGdo4yK7iuXOV+/pp/yQ/8ASUb5U70ZtfzS/MK83+N3wuPxT8KnSYrlbS6MyTW8rglBIoK4YDnaVZhx0PPOMV6RSjAZSQDtIYA+or4nM8D9eoqMXacWnF+aPp8LiFQndrRqzXqeB/Bz9nPU/gj4n1O6129tbrUmhW3SK0LNGEYo5LFgMk7V4xxg+tfUtt4/8Ozw2kt5pGy7tVAQRIpVSP7vI4zyAelZurS6Drl0+o6hJf29zIFDpbqrICFC8E89qzvsvhH/AJ/dT/79rX4/mDzLBY+tUnKC5npzW2WzR9phlhq2HhGKenbzH6t48n1DxNb6qlvGI7b5YoZQGG3PJPuc9R04rov+E98MyXS6jJpMg1EYbcFUnd65zgn3xmub+y+EP+f3U/8Av0tPj0/wnM2FvNUJ/wCua152FrZnKq+SpCcpu+tnr0sdNWOGUVzRaSVr7Ct4vbWPGlpqk8flwwsNkSnO1Rk4z6mq9xMbi4llbAaRmYgdMk5NRNaWttcSG18wxZwpmILY/AYpa/W+HctxOBozqYxp1Ju78vI+OzLFU8RNRoq0UrLzCiiivrTxwooooAKKKKYj8q/F3/I2a3/1+z/+jGrIrX8Xf8jZrf8A1+z/APoxqyK9lbHhvccGUZBGdwwOKljIVRgbTnHIz+NRxruJ4DYHQ0qSFXIwfbJ7V7GH/ho4qnxE6guuFyMn86RsoxAUPjnnmmL8zckDPT2pfMIJOQQPWukxFWVoxtKL83U9MVLGRKFK9VODx1qISEEEj1PsaVG5JUbW9AelCFYnYBXG8DaTng+nvSqwabI5DfjzTJHBOAMkE5Of6UgbbIoJzjoBzVXF0JiQFAYAjrjHWmjdIobAAUn8PwpJCGwOMEZ4PSnKAFALBu/59RQIlDkrhD1PfuKluCAm1DyMZA5/CqrcEKTkDHQU8TD5jhlIYD6+lNMVhQDJku55+UAD7tOSdc7GTiM7Qfb1oUngAlT1I9aZtEjspBG7nPHNZVf4cvRlx+JHu3x3Yr4RsyP+f5O+P+WcleItKWiweWAJwP5Zr2747p5nhKyGM/6cnH/bOSvD1XGATk5G30xX5Z4aX/sP/t6X5n1PEn+/fJEisWVygXGANpPSpUWPABYhieSOg+lVA7FdpGDnJOOKmZgI1U8/KefQ1+tJnyQu4BgCcZJzg5zUqqHcOSAFGFBGM1XDFuoG4j73epDuOC5+UjBAOTQA9PmXAXkZJz3FELElVJK5645C0JlCRj5ccUwSLzkEMpHX609hEkg+f5CeeMCpDkIyljjgben51HuBk3KwGB3pu4jLDLb+3Wi4rEm4xhgBlT0yec1YUFh8n3sfdzUODxzwvTsRSiQEEDnuQKa0E9R6LygfOc0jKRKSGITooHXPemlzvHGSfXtUhAIBH3geMcgUweghyZFBYYxkrnt61ga0u2+I/wBkV0GeHYZ+Uc1z2s/8fhx/dFcWM/h/M6cP8RRr0T9nn/ktfhD/AK/l/wDQWrzuvRP2ef8AktfhD/r+X/0Fq8Kfws9SHxL1P0nooorxj2gooooGfMfx6/bU+E/w18Ty+Gda0yfxfqVoxW5hsrSGdLZ/7rNKyru9QucHIOCMV8FfBv44eD/APxs8XeLdb8Oz6noWrLdC0sI7WF2g8y5SVMqzBV2qpX5ScZwOK6/9j3w3pXxJ+JPjTUvFGnW2v3Sp5+dRiWdTJJKS7kMCCxx1I9a7zxP8c/gH4X8QahpE3w/trqexmaCWW10CzMZZThgCzKxwQRnA6ccV6dHK1Ww0cVWrKCbaSfkc8sf7GtLD06Tk0k3bzPIf2hPj54J+KPhd9P8ADvhm40a6bUFuRNLawxgRgOCuUYnPzLx04r1/4J/tqfCT4e/Cvw34c1z4f3uqavp1r5NzeR6faSLK25juDO4J4I6jNdJ8JvGvwU+MmuXOkaL4D022voYTceXf6FaqHQMAxUruHBZeDjrxnmsn4gfFr4E/DnxXeeHr/wAAWV3fWjBZmsdBtGjViAduWKkkAjoMe9df9h04UliPrMeVu1+lzn/taU6jo+wlzJXtpsdR/wAPCPgfnH/Cr9Q/8Fdj/wDHK5H4vftufCHx18MfE3h7R/h5fadqupWMttbXcmnWcaxSMuAxZXLDB7gZq58Nvih8Dfil4ttvDumeArG11G5VzD9u0G0RHKqWKgqWwdqseQBx1ziuv+OPwt8GaX8IPF95Z+EdCs7uHTpXiuLfTYUeNgOGVlUEEeorWOQzrYeWIo14yik728tbGTzaFKvGjUpSjKVt/PQ9E/4JP/8AJtutf9jPc/8ApNaV9oV+Nv7If7ZPiL9mDR5rK88OvrvgDUr95GZVMTxXISMSGKUgqxCeUSjf7JBXJJ/Q74b/ALenwW+JEUQi8WxeH7113NZ68ptGT2MhzGT9HNfyRxRkmPePq4unTcoSe61++2x+tYHFUvZRpuVmkfQhAIwRkVSu9D03UBi60+1uR6TQq/8AMUzR/EWleI7OO60nU7PVLWQZSeznWVGHqGUkGtCvgH7am7O6Z7HuyR5/40j+FvgK1gvPFcXhLw/b3Ehjin1aO2t1kfGcKzgAnAJwOcCtrRvCng++sLTUNL0XQ57K5iSe3urO0haOWNlDK6sq4IZSCCDggg1xn7RH7Nfhr9pfw9pWjeJ7/VrC1066N3E+kzRxuzlCuGMkbgrgnoAc969B8G+F7XwT4Q0Pw5YyTTWOkWEGnwSXBDSNHFGqKzEAAsVUEkADOcAdK9OpUprCQlCrJ1W3zLol0sznjF87vFW6M0YLO3tVCwW8cKjoI0CgflU9H1rl/FfxS8G+A0DeI/FWjaHuBKrf30ULNj+6rMCfwFebGnWrvlinJ+Wpu5RirvQv+Ifu2+fU/wBK/M79pb/lIx8Of+v3RP8A0oFe8/Gn/gpV8LvDMBh8KtdeNdTTeqi2ia3tlbgfNJIASPdVYHHXvXw/ofxK8UfFz9sT4eeKfFdobG9vtd0treBYGiRLcXCCMIG5K9fmJOTnmvv+BuHcxwed1MyxNPkpuNlfRt6dN+nU8fNsZRqYRUIO7vrbb7z9f6KKK/pM/OwooooAKKKKYj8q/F3/ACNmt/8AX7P/AOjGrIrX8Xf8jZrf/X7P/wCjGrIr2VseG9zf8Iruupx/sj+ddSrjeAx4A4965bwgM3M/b5Rz+NdOvzEnPPGMivocH/CXzPHxHxsc2QcjI4opGJJBIwelFOvuisPszxIH5QOtbel4+xjGep61hhyM8cVt6S2bMc5+Y15tZ+6d9PcuV6J+zz/yWvwh/wBfy/8AoLV53Xon7PP/ACWvwh/1/L/6C1edP4WdMPiXqfpJNNHbwvLK6xxIpZnc4VVHJJJ6CvFNd/a28H6VqEttawX2qJGxU3ECKqMR3UsQSPfFa37T+pz6b8IdSEDtGbiaKBypwSrN8w/HGPpXw5XnU6akrs9OpUcXZH15/wANkeGv+gNqX/jn/wAVR/w2R4a/6A2pf+Of/FV5H8M/2YPFvxJ0qLVY2tdI0uUboZ71m3TD+8qqCSPc4B7ZrJ+KnwA8U/CaJLrUo4L3THYIt9ZMzIrHorAgFSfcY9Ca8uOaZbLE/VI1U57Wv1Oh4bFql7ZwfL3seoeJv2hfhz4wnin1fwpqF1PGu1ZVk8twvoWR1JHsaxf+FqfCP/oTtV/8DZf/AI7Xz/Wx4f8AC1/4kkYWiKI0OGlkOEX/ABP0r7DD4rHRtRoVZeSTZ87iaeDinWxEI+baR7R/wtT4R/8AQnar/wCBsv8A8do/4Wp8I/8AoTtV/wDA2X/47XlmsfDXVNLtWnRo7tFGWEWd4Hrgjn8K5Kuiti8zoO1StNP/ABM5sPTy3Fx5qFODS7JH0B/wtT4R/wDQnar/AOBsv/x2j/hanwj/AOhO1X/wNl/+O14douh3viHUI7Kwgae4foo4AHckngCu8n+AuuxWXmJdWcs4GTArMCfYMVwT9cVyPNsZF2eIn/4EyKyyzDyUasIJvo0j1bwx+0n4A8HQyxaP4Yv7NZWDSMGV3Yjplmcsfpmtv/hsjw1/0BtS/wDHP/iq+Sr6yn067ltrmJoJ42KvG4wVNQVw1W68/aVZNt9Xrc9ujKNOCjSSS6JbH15/w2R4a/6A2pf+Of8AxVH/AA2R4a/6A2pf+Of/ABVfM/gT4deI/iXrB0zw1pc2qXarvdY8BUXONzMcBRn1Nemf8MX/ABb/AOhbj/8AA6D/AOLrmcaSdmzojKrJXSPTP+GyPDX/AEBtS/8AHP8A4qon/a98KSctoeok/wDAB/7NXnH/AAxf8W/+hbj/APA6D/4uj/hi/wCLf/Qtx/8AgbB/8XXLWwuDxK5a0VL1sbQq4mm7wbT8j0Vf2ufCanI0PUfxKn/2apl/bF8MIMDRtSA9tn+Neaf8MX/Fv/oW4/8AwNg/+Lo/4Yv+Lf8A0Lcf/gdB/wDF1nQwOAwutCnGPokVPEYqr8cm/U9M/wCGyPDX/QG1L/xz/wCKo/4bI8Nf9AbUv/HP/iq8z/4Yv+Lf/Qtx/wDgbB/8XR/wxf8AFv8A6FuP/wADYP8A4uu3913Rj+97M9M/4bI8Nf8AQG1L/wAc/wDiqP8Ahsjw1/0BtS/8c/8Aiq8z/wCGL/i3/wBC3H/4Gwf/ABdH/DF/xb/6FuP/AMDYP/i6P3XdB+97M9M/4bI8Nf8AQG1L/wAc/wDiqP8Ahsjw1/0BtS/8c/8Aiq8z/wCGL/i3/wBC3H/4Gwf/ABdcT8RPgj40+FMdvN4m0SXT7a4bZHcBlkjZsE7dykgNgE4PPB9KajSk7J/iJyqpXaPrD4fftFeFfiDqiaZC1xp2oSf6qG8UKJT/AHVYEjd7HGe2a9Rr8ybW5ks7mKeF2jliZXR1OCGByCD2Oa/SjQ7p77RNPuZTmSa3jkY/7TKCf51nUgo7GlObluflz4u/5GzW/wDr9n/9GNWRWv4u/wCRs1v/AK/Z/wD0Y1ZFeotjyHuKrbeQMmnbmBJwCQOvTFNxnPPSlQsASWx6eleth/4aOKp8THyDYhO5XPbrkU9y8q7hxkDggUyMllYAArjPuKcsgfqRjH611mTGyb2jwOpPPGSBUkajGQeCOOD2piSCQAhsEH86k2lnKjgY6k8/pQheQYyAQBnu3TNShsRjaihuxzUO4xjY5yzNgcHijcOhIByR+VCYrE6gELkDnuaRWCz5IBGPypgUzHAJYjsDSsdqPuJGBjkd6Bk+0qwK9+uaeSF+eTGByP6VH5hgUFRv4zxzx60kMpmYZOD157+lVcnfUl3nqPlbH8XelDAzK2BtIOeMY/8ArUwxjbzwc5HNO6ISVHTHHQ1nV/hy9GOPxI93+OY/4pSyz0F8hP8A37krw+R1mkyigAevHFe3fHg48H2gxn/TkH/jkleFMvOSQevIOMCvy7w0/wCRF/29L8z6jiRf7d/26h7ZRsYI3cDFSsxKohHC5O7uc1GCZTxnbnAGfbrmnNgsqgkFTwRzX6xqfKkjy5HGOnHqKjUuFZgASo+UDrSbAco5y3XOcDH4VIR5ZCsQAehBBBFLVk7DlbcoDEjjgjsacFSVidygDkH1IqPJAUnozY4HWpB8pBG0oF5GM81SEKoBbLDg8cngU7zGR0IIGeMMBx6VFuG7+IDGeRwaXlmBRd/P3SaQNE3WTBJZwfwp6ggFkIUA4+mOahRzuBAJwfmBHSlDqWMeRktn/wCtVXDUVQWJk35xkj3z7U+EfKwYMe9MOWkYDIRh6c8dfwpVcCUgnqDj64pi3EEpyVwSrEA+w9KxNXJa8JIwdo4rbiYTFtvAC9h3rF1gk3pJGMqK4MX/AAzpw/x/Io16J+zz/wAlr8If9fy/+gtXndeifs8/8lr8If8AX8v/AKC1eJP4WenD4l6n6T0UUV4x7QUUUUDPyv8A2Bf+Ry8b/wDXvF/6NauE+H/iLwD4b+PXjm5+IdlHfaO817HDHLbGcCc3SkNtAODtD8+/vW5+zl450n9nv4peNdI8dTSaPMC1m0nkvKqSRynKkIrHB7EAivQ9e8Q/sr+JtYutU1GS3mvbqQyzSRwalEGY9W2oqqCTycDk5PWvclhKWZ5TSw/t4wknK95Wer6HJQxlXLM0niVRc00rWV1oupxv7KF5pWoftP8Aiq50GIQaJNDfSWMapsCwNOhjUL2AUjjtXK+O9S8O6R+1t4huvFVulxoaXcvnRtEZASYcL8o6/MQa908B/FD9m74ZXs954a1CDTbuaPynm+y38rFc52gyI2ASBnGM4Gegqj4u8afsw+Otbm1fXLiC81GYDzZ1t9QiL4GAWCKoJx3xnivSjgaUMvp4VYim5RlfWWjPJxGIqYrF1q0qE4xnHl0Vmr9V28jyn4P6l4e1b9sTR7rwtAtvoUhk+zxpF5YGLFw+FPT5gx/HNfXn7QP/ACRPxrn/AKBc3/oNeOeDPHX7Mnw+1tNY0C6gsdSRWRLg22oSsgIw2A6sFJGRkDOCR3NaXxh/ae+Gfif4X+J9I0zxKLrUbywkhgh+w3K73YYAyYwB9SQK9fAyw+By+tSqV4OUuZ2T01WyPLrUq1bE4f2dKfLBRV5LXR7t/met/wDBMXwro3jH9lnxBpeu6Xaaxp03ia5ElrewrLGw+zWnVWBFdh8Qv+CZvwe8aSy3Gl22o+ErqQlv+JZcbosn/pnIGAHsu2ub/wCCTuo20n7P3iCxS5ia9h8RzzSW4YF0Rra3CsV6hWKMAe5U+lfbNfw7nuZY3L84xDw9Rx97o9Ht02P37C0aVbDw54p6H5va1/wSf8QaTOZvCXxOVNp3It7bSQOMdPmjY8++BVZv2M/2rvDEYj0f4qNcQoMKkGv3KAD/AHWUAV+lPPrRXNHi7MrWq8s/WKf6FvL6P2br5n5lt8Af22rdtsXjXUZFHdfEq/1anxfs5/tpagwW58eX1sh6l/Ep4/75Jr73+LXxx8E/AzSLHU/HGtjQ7K9nNtBKbWa43yBSxXESMRwCckAV1Ph3XbHxToOma3pc/wBq0vUraO9tZ9jL5kMih0bawBGVYHDAEZ5ANdcuJMdGnGs8LTUXopcmja7MzWDpOTipu66XPzluf2Af2ifG0Yj8UfFeGSFvvR3Gq3dzgf7pUD9a6Pwv/wAEk9PaRZvFfxDvr85y0WnWqx5/4HIzH8dtfoRxRXHPi3NGrU5KHpFI0WX0N3d+rPm7wb+xF8IfhI9rc6Z4XTUtQU5+2aw5un3DBDBW+VTnnKqK+RP2kkWP/gor8OFVVVRe6KAFGAB9oFfpn4hztg/4F/SvzF/aI1S01L/gox4BFrcx3Jt9S0aCbymDbJBOCVbHQgEZHvRwPjcZjuKqk8TUlNKD3bdr8v3BmtOnSy5KEUtf8z9IaKKK/qc/MwooooAKKKKYj8q/F3/I2a3/ANfs/wD6Masitfxd/wAjZrf/AF+z/wDoxqyK9lbHhvc3vCWftFxj+6P511COVwetcx4SIFxcDHVR/OulLFOByeBX0OD/AIS+Z4+I+Nj2bfj0znmimN8vXJzzRTr7orD7M8Q3HOQTjHXHWt7SBizHBHzGsINnBHA7Vv6WMWgB55NeZW+E9GG5br0T9nn/AJLX4Q/6/l/9BavO69E/Z5/5LX4Q/wCv5f8A0Fq8+fws3h8S9T7F/au/5JFP/wBfkH/oRr4v01YH1G1W5O22aVRKfRdw3fpmvtD9q7/kkU//AF+Qf+hGviauCK56bR6E3yzufqzYQwW9jbRWqJHaxxqsSRgBFQABQuOMYxiuU+MkFncfCvxSl+FNr9gkZt/QMFyp+u4Lj3xXyZ8L/wBrjXfAWg2+jajpsevWVqojgd5jFKiDopbawIHQZGQMCsb4wftMa78VtNGkpbR6Lo+4PJbQyF3mI5Ad8DKg87QAMgE9BX4Th+DMzp5lGUrKClfmvur39T76rnmFlhWo7tWt8jx2vavh7HCnhOyMW3LbmcjqW3EHP5D8MV4rXQeF/GV54YLpGqz2znc0TnHPqD2Nf0plmJhhKvNU2ta/Y/Gc8wNTMMLyUXqne3fyPbfrXgOvJFHrV8sOPKEzhcdMbjXVat8VLu9tXhtLZbMupUyb97AH04GK4c5JyTkmu3NcbSxKjGlrbW55nD+V18A5zr6N6JHsX7PUduZNakO03aiNRnqEO7OPxAz9BXstfJvhnxNe+E9US+sXCyAbWRhlXU9VIr0ab9oO6a1KxaRGlxjHmNMWQH124B/WvjqtGUpXRx5vk+KxOLdakrp267GT8do4E8ZxmIL5jWqNLj+9uYDP/AQtec1c1bVrrXNRnvr2UzXMzbmY8fgB2A7CqddkI8sVFn2WDoyw9CFKbu0kffP/AATst7Jfhz4nnRY/7QfVgkrDG/yxChQHvjc0mPfNfWVfkZ8KfjJ4n+DOtTaj4bu1iNwojuLadd8M6g5G5eORzgggjJ55Nexf8PAPiL/z46J/4DP/APF1wVcPOU211Pep14xilLofojRX53f8PAPiL/z46J/4DP8A/F0f8PAPiL/z46J/4DP/APF1l9WqGv1imfojRX53f8PAPiL/AM+Oif8AgM//AMXR/wAPAPiL/wA+Oif+Az//ABdH1aoH1imfojRX53f8PAPiL/z46J/4DP8A/F0f8PAPiL/z46J/4DP/APF0fVqgfWaZ+iNFfnd/w8A+Iv8Az46J/wCAz/8AxdH/AA8A+Iv/AD46J/4DP/8AF0fVqgfWaZ+iNeQ/taQWs/7PfjD7WqMqW6PGXx8sglTaR75x+dfJn/DwD4i/8+Oif+Az/wDxded/F39pfxr8Z9Pg07W7m3ttLicS/YrGMxo7jozZJLY7AnA64zzVww84yTZE8RBxaR5TX6T+Fv8AkWNI/wCvOH/0Ba/Niv0n8Lf8ixpH/XnD/wCgLXZV2Ry0ep+X/i7/AJGzW/8Ar9n/APRjVkVr+Lv+Rs1v/r9n/wDRjVkV3rY8x7iMegBwSeKVCCCuOG6jHSgkDqcen1pYy2TkMVPOT6V6+H/ho5Km5f0aASa5piyIGVrmNHQjKsCwyCDX2b+1d8SNH+B/xUi8MeHvhX8NriwOnW12W1DwxDJIXcNu5UqMfKO2eetfGvhghdf0sklc3UXXn+IV91/tcftT/ET4K/HzTdN8O60F0KDTrS6l0mW3ieKcndvDOULqGAA+VhjqK4sVFyrQilfR6P5HXh3GNKTk7arVHmn7RHg/RPEH7OPgT4hv4L0/4feMtS1B7GTStIsjaxXsQ37ZFh/hztUgnOQ2MnK1y+m/sR+N2/s61vvEPhDRvEmpQfabPwxqGr+XqcoKkhRGEI3EA8buxzjBr1P4/Wtr4r8SfCn482Gq3+peBdQ1Oy/tGzvbp7hdIkWZGkjAJYIh2uCBgBl4+8K7v4z2MUn7Qg1S1/Z+13xrqk9xbXmm+KLHxJdR2s+1UaNxtiaKIKVAKswAxk8Nk8cK9SnCMYPe79LdNX0OqVGE23Jdl6+Z8efD39nPxt8RPFGv6PBb22hNoAY6xe65N9ntrDaSD5j4Yj7p+6DwM9Oa6Dxz+yf4m8E/Dy/8cv4g8L674YtXhjF3ot+9wtw0j7CIz5YGVb7wJBGeM19Ia1qmtfGDwZ+0R4csvD6eG/iHc3dje3Ph21vkvpZI41iWRUkjAWQlYyCqjgtg8mvO7DwH4n8D/sEePx4l0y80hL3XbOa0tNQjaKTYJIVZtjYKglSBkDO3PTBO/wBaqOzbS1St69TH6vTWiTe7v6dDw/4i/AXxB8L/AIlaZ4H1O602TVtSW2aG4s5ZGgXz22puZkU8HrhT7Zratf2VfGt18YNX+G1r9hvdZ0pFn1DUFndbC3jMaP5jyMisBh1GNuSc4BAzX0T+0V8LfFfxB/ak+H2v6BoV7qWg3Nrpc66pBCz2yJHJvdnlHyqAvJ3HkEYzmvRLXUtPu/jp+0t4d/sc6/rOoWFnLBo8V4bObUYVtFEkMcoBZT8w5A/jFTLGz5E1Zu1363Q1hIOTTulfTzR8Y+Pf2ZvEfgvwZL4qs9a8N+MvD1tMLW7vvCuom7Szc4wsoKqRnI5AI5GcZFdLpv7Fniy5uNLs77xT4L0TxBqFuk9t4d1LV2i1FwwJQGMRsNzYOBnsfQ16bYaleeFvg38R4PD/AOz5qngbQtUsDZ6nqWveI5VjgyGVJVhuIVaQoWztjBJ4HXFdzqei6r408UeGPC3xa+Ddr47uLy3t44vH3hCS4RfJYYR3kCIDtBLMGZVHUKcgFPFVrb7X7XtZedv1GsPSvovztc82+DH7N9hcfBv4qy+JrnwvYeKrNhYRzaxdEHRmSUqzTEKwjD7flddxZe4Br5NuI/sd5PEJoZ0iYoHibcj7Tjcp7qeoPpX2l4b+G9v4X+H/AO054J8JSXHiEWctrFaxQDzp3GdxXaoyxU7lJA5KGvid0eB5IZUaKRWwyOCCrDggg9DmuqhNzjVblft9xzYiMYezSjb/AIc93+O2P+ESsiTjbfoc/wDbOSvCmdJJXzt2/d49BXu3x158JWfXH25Og/6ZyV4LLIgIBG5jnPGOa+A8NP8AkRL/ABy/M9jiT/fv+3UToxRVCkBc+neg4V1AJyW54pilUi5yO/AzUiFTGACW7mv1nU+UAszZGFZs/eNAh+bO75OmM8j8PejPlyYKfe6Y9K6fWvhv4l8N+EdC8S6lp7Wmha4HOn3hmjcTbfvDarFlI/2gM9qzlKMbJvcqMZS2Wx7t4y/ZG0+P9ofwp8OPDWqX0Vlqujw6rc32pMkrwg+aZCoVUBACABcZyeTjpJovwx/Z88S+PW8E6Z4l8cWWuyztY22q6hHaHT3uM7QAoUSYZuBnH1HFeh/tDWfj6b9qjwne/DaL7R4l0nwna3vlebEgeJZJRIrCRlDKQ4BUHJByORkcZo/x0+Gnxi8S2ekfET4P6faaxqNytrNrPhyVrWb7Q7hdzQjBPzHJ3Ox9jXgKpWlGMuZtW1s1debT3PXcKUZONkteq0fkjgfBP7L+ra/8W/FPhPU9Ug0jTPC2+XWtbk+aOGAcqyL3ZlGQCRgZyeMHY03wj+zp4k8RLoVj4i8d6XeXTi3tdb1CG1axMhOFZo1USBSSOpBA67a9l8OfC+H4d+Ef2i/hhoV0+o6tDawXdqcA3E9qY92wgAZZQSpx/eBwM4r4dstNudYvLWwso2nu7mdYYYVGWkdmCqqj1JIH411UpSxDk3NpK1radL3MKkVQ5Uop33vr12Po74Z/snxXX7RWo/DPxreXVrDbWMt8t9pcqKZ0G0xyK0iMApDHIIyCCM8V578Vvgrd/Cf4unwjftLLp1xcRmzv12/6RayMAsinGN2CVPH3lPGMV9vJeW3/AA2BoOmkiXVtN8ByW+oTKQT5mQwVj6gEn/gfvXiHwo1C3/ak+F1j4T1OVW+IPgiRL3R7lyGkvrJWUtCSeSQAF+oQnvXNDF1ub2knpZX8t9TonhqXLyR3vp57aHi37THwq0n4N/Fi78L6Jc3lzZQW0Mqy3zq8pZ1DMCVRVIyePl/OvKVUZ9C2ST16jFfQ/wC3ort+0hqqYO37Faj8fKFfPJB2gkNgjnJr2cLJyoQlJ3bSPKxEVCtKKVkmOKshwSeFG0D6Viavn7Yc/wB0VsRkx4P97jB6isfV/wDj8P8AujvWeM/h/MMP8RSr0T9nn/ktfhD/AK/l/wDQWrzuvRP2ef8AktfhD/r+X/0Fq8Wfws9OHxL1P0nooorxj2gooooGeU/FD9lv4Y/GLWF1bxV4XhvtTChGu4ZpbaR1AwNzROpbAAA3ZwOBXFf8O+vgV/0J83/g2vP/AI7X0XRU8sXq0WpyWiZ86f8ADvr4Ff8AQnzf+Da8/wDjtH/Dvv4F/wDQnzf+Da8/+O19F0UckOwe0l3PnT/h318Cv+hPm/8ABtef/Ha4H4/fsTfB7wP8FfGmv6N4YltdW03S5rm2nOpXThJFXIba0hVuexBFfZNeU/tWf8m3fEf/ALAlx/6CaiUI2ehUZybWp+dP7NP7Pfxf8QeAZvil8I9ba21TT9Sl06XT7a4MFxIqRwyZBPySK3mYKNgfL/FnA9w0j/gol8Z/g7cR6X8Vfh19veFdjXElu+n3Eh/vFgrRn/gKgGvXv+CUP/Jtutf9jPc/+k1rX2NqOl2esWr219aW97buMNDcRK6MPQqwINfgOeZ5QjmFXC47DxqRi7J7SXzPusLhZexjOlNpvfsfGvhf/gqz8LNUhj/tnRvEGhztjeFhS4jU+zBwxH/ARXo+lf8ABQr4D6oqn/hNRZs3VLqxnQj6nYR+tdj4o/ZJ+DvjEltT+HmiM55L2tv9lYn1JiKk/jXn2rf8E4PgVqefK8M3enE97TUZh+jMw/SvA9pw1W1cKkH5NNfidnLjY7NMy/i58af2WPj7o1hpnjbxjpuqWVjObmCIXN1blZCpXdmPaTwSMHiup0X9sf8AZ98FeHdM0TTfH2nRaXplrFZWsKLPKUijQIi7ihLYVQMkknHJrhZv+CWPwXlYss/iaEf3U1CPH6xGrFl/wS5+CdqwaRPEF2B2n1FcH/vmNa6pSyCVGNGVeq4LVR0sn5LYzSxak5KMbvqbevf8FIvgXo0ZMPiO81Zv7ljp0pJ/Fwo/WvK/F3/BWrwhZ/u/C3gvV9ZlOQHv5ktVz2wF3k/pXs2l/wDBPf4DaXtP/CEC8Zf4rq/uHz9RvA/SvUPCvwD+G/gcxnQ/A+g6dJHyk0dhGZAfXewLZ981zLEcN0NYUZ1H5tJfgacmNlvJL0R+eusfH/8Aai/alC2Xhbw7N4T0ObKG50+3a2UqSOWupTnIH/PMgnPQ5FeOaZ8G9W+B/wC1/wDDPw9rmoxanq0msaVe3E0JJUPJcqSu5uWxj7xAznpX7H+IUCx26qNoGcADAHSvzQ/aW/5SMfDj/r90T/0oFe3whxH9ez2eXYfDwpUlG+i1b03l13OTMsH7LBqtOTlK9tdl8j9GKKKK/oY/PwooooAKKKKYj8q/F3/I2a3/ANfs/wD6Masitfxd/wAjZrf/AF+z/wDoxqyK9lbHhvc3fCa5uJx/sj+ddNk5+nUmuY8KcXMx4+6P510iEuvJK5NfQ4P+EvmePiPjZI2CAQ27+VFMI8pgB29e9FOvuisPszxIH5sg8mt/RyTZDP8AeNYIJBBPToK3tHz9jGR/Ea8yt8J6MNy7Xon7PP8AyWvwh/1/L/6C1ed16J+zz/yWvwh/1/L/AOgtXnz+Fm8PiXqfav7SOkjWfhlPbGUwg3ULbgu7oT2yK+TP+Fdx/wDQQb/vwP8A4qvsL47f8k/m/wCviP8Ama8p+C/w1i+Kvi99Fnvn09FtXuPNjQOcqyDGCR/e/SvNjLljdnpzjzSseJ/8K7j/AOgg3/fgf/FUf8K7j/6CDf8Afgf/ABVfb/8AwxRp/wD0NFz/AOAq/wDxVH/DFGn/APQ0XP8A4Cr/APFUvrEe4/q8ux8Qf8K7j/6CDf8Afgf/ABVH/Cu4/wDoIN/34H/xVfb/APwxRp//AENFz/4Cr/8AFUf8MUaf/wBDRc/+Aq//ABVH1iPcPq8ux8Qf8K7j/wCgg3/fgf8AxVH/AAruP/oIN/34H/xVfb//AAxRp/8A0NFz/wCAq/8AxVH/AAxRp/8A0NFz/wCAq/8AxVH1iPcPq8ux8Qf8K7j/AOgg3/fgf/FUf8K7j/6CDf8Afgf/ABVfb/8AwxRp/wD0NFz/AOAq/wDxVH/DFGn/APQ0XP8A4Cr/APFUfWI9w+ry7HxB/wAK7j/6CDf9+B/8VR/wruP/AKCDf9+B/wDFV9v/APDFGn/9DRc/+Aq//FUf8MUaf/0NFz/4Cr/8VR9Yj3D6vLsfEH/Cu4/+gg3/AH4H/wAVR/wruP8A6CDf9+B/8VX2/wD8MUaf/wBDRc/+Aq//ABVH/DFGn/8AQ0XP/gKv/wAVR9Yj3D6vLsfEH/Cu4/8AoIN/34H/AMVR/wAK7j/6CDf9+B/8VX2//wAMUaf/ANDRc/8AgKv/AMVR/wAMUaf/ANDRc/8AgKv/AMVR9Yj3D6vLsfEH/Cu4/wDoIN/34H/xVH/Cu4/+gg3/AH4H/wAVX2//AMMUaf8A9DRc/wDgKv8A8VR/wxRp/wD0NFz/AOAq/wDxVH1iPcPq8ux8Qf8ACu4/+gg3/fgf/FUf8K7j/wCgg3/fgf8AxVfb/wDwxRp//Q0XP/gKv/xVH/DFGn/9DRc/+Aq//FUfWI9w+ry7HxB/wruP/oIN/wB+B/8AFUf8K7j/AOgg3/fgf/FV9v8A/DFGn/8AQ0XP/gKv/wAVR/wxRp//AENFz/4Cr/8AFUfWI9w+ry7HxB/wruP/AKCDf9+B/wDFUf8ACu4/+gg3/fgf/FV9v/8ADFGn/wDQ0XP/AICr/wDFUf8ADFGn/wDQ0XP/AICr/wDFUfWI9w+ry7HxB/wruP8A6CDf9+B/8VX3r4dj8nw/pkec7bWNc464UV8leI9KGh+IdU01ZDKtndS24kIwWCMVzjtnFfXGh/8AIF0//r3j/wDQRTnJySCnHluflx4u/wCRs1v/AK/Z/wD0Y1ZFa/i7/kbNb/6/Z/8A0Y1eifDX4uaR4O8MJp17bXss6yu5aBEZMMeOWcH9K5s2xuKwGGVXCUHWle3KnbTuc2Go069Vxqz5F3ep5E7BcEgEe/al35xuCjjjFfQv/DQvhwEA2eqDPH+pj/8Ai6bJ+0P4cjXP2LVCB6RRH/2pXzVLivP4xssok1/j/wDtT0J5XgHLXFr7v+CfP9rK8EolQmKRCCrBiDuHQgjpg1q+IfF2s+MdQ+3a9qt9rWoNGIhdajcvcSbB0UM5LbRk4GcV7aP2hPDjEAWWqdM/6qP/AOLpD+0J4dAB+xaocnH+qi4/8iVr/rVn+/8AY8v/AAP/AO1I/svA7fW193/BPGrPxt4i0/w7c6Bb+INVg0K4YtLpcN7ItrKePmaENtJyByR2HpWrpPxk8eaJpKaXp/jbxHp2nxLsjs7XVbiKFR6BFcAD2xXqH/DQ3hzeF+xapk/9MY//AI5Sn9oTw8rYNhqvJwD5MWD/AORKh8UZ698ml/4Ev/kS1lmCW2MX3f8ABPEdJ8UavoOrR6ppmqXum6mjMReWVw8U4LdWDqQ2Tnk5r2Vf2jG1n9nPxh4H8TX+v674q1jVbe7ttSvpvtMaQRmI7GkkkLg/I5CgEfN1GTVj/hobw5jP2LVP+/MfH/j9SJ8f/Dz/APLnqa/WKP8A+LqZ8TZ7UabyeWjX2+3/AG6OOX4KCaWLWt1t/wAE8o0/4seNNL0MaHZ+MNettICmNdOh1OdLcL/dEYfbj2xVFvGXiG68UHxDPrmqS+It6uNVe8ka63KoUN5xbdkKAoOegA6V7Mvx+8PMwAs9T5/6ZR//ABdNT9oHw677fsmpZzj/AFUfX/vurXFGerX+xpf+Bf8A2pP9m4K1vri+7/gnlHib4meMvGlott4i8V61r9ojB0t9U1Ga4QN2IV2IB96taT8XvHOj6Wmk6b418Safp0a7BZWuqTxwKp6gIrgAe2K9QX4++Hm/5dNSH/bKPj/x+k/4X94e5/0TUsjt5Uf/AMXR/rPntrf2NK3+Jf8AyIf2dgt/rmvp/wAE8h8PeNvEXg++uLvw/r+qaFcXC7ZZ9NvZLd5RnOGZGBYZJPJrHvLh7q8knnkeaeZmeWWRizOxOWLMeSSSST3r3hfj34fYkCz1Pjv5Uf8A8XSn486ABxaak/piKPn/AMfpy4oz5p2yeV3/AHv/ALUj+zcB/wBBi+7/AIIvx4yfCFpj/n+TnHT93JXg4ZJHc5LNgAjH516T8TPibpfjLQbaysYLqKVLpZmNwiqu0K64+Vic5Ydq7b4b36aZ8LbS9mVmjt4riZ1QAsVWR2OMkDOB617HAOBxWXZOqGMpuE+aTs/NnNnlWlisZz0ZJqy1R4DIgwNwJX6dKegXkjqOn+e1e3D456CQD9k1HBOP9XHx/wCP0D456CWA+yajk/8ATKP/AOLr9IufPci/mPENxOw45J7D+teyfDn9qLxN8OvCNr4Yk0Xw34r0ezuTd2Nv4k043X2OQnOYiGXacknnOCTjFXR8cNCOcWmocf8ATOP/AOLpW+N2hqufsmoEe0cZ/wDZ6yqU4VY8s1dGkJezd4ysctq37RHjnVPix/wsUau9r4lRgsc0CBYo4wNohVDkGPGRtOc5ycnmvQW/ba8Spqz6zF4F8AW3iA8jXItBIvQ5GN/mGQjPvj8KyD8bdCzgWuoHjJ/dpx/4/Tf+F4aEFybXURg4x5cec/8AfdZSw1KVlKK0LVWau1PfU4bRfjB4y8PfESTx1aa7cr4ollaWa9Zg5mLfeV0I2spwBtxtAAwBgY9M/wCGxPEFvqEurab4I8BaT4ikDZ12y0Lbeq7D5pAxcjcc8nHNZjfHDQlQn7JqBx2ESZ/9DoX446ExA+yagDjODHHx/wCP1UsPSm1eIo1JRWk/+HOd+Hnx48VfDz4iX/jeOW31nXb6GaK5l1dHlWTzMbidrqcjAxzgDjGMVzPgXx1qvw38W6b4i0Gf7NqenyiWJiMqcfeRhkZVgSpHoe1elj41aGf+XXUMYzny0/8Ai6RfjZobEhbTUDj/AKZp/wDF1p7OGvu7qz9Cb7e9s7r1OM+Lfxa1j4zeNbjxPrttZ2uo3EUcbR2EbpDhVwpCuzHOOvNcgZclSDkAYy1exj42aGR/x6ahwMn90nH/AI/Sr8atEfG201A5OP8AVp/8XVwiqcVCCskZySnJylLVnjOQy4Dkrn5jWLquPtZxjG0V9An40aIrYNtfg/8AXNP/AIuqtx8ePD9tKUa01Ikc5WKPH/odY4hKUbN2LpRUZXTufPleifs8/wDJa/CH/X8v/oLVJ8U/iRpnjfT7GCwguonglZ2NwqqCCuONrGo/2ef+S1+EP+v5f/QWrxKsVFNJ3PQp/GvU/SeiiivEPaCiiigYUUUUAFFFFABXBfHrwrf+Ofgv410DS0Euo6hpU8FvGSBukKHauTwMnjn1rvaKHqrAtHc+AP8Agnf+1F4P+Cvh/Wfhn48lm8MajJq8l3Fe30bLCJGSKJoZeMxMpizlhjk5K4Gf0j0fWtO8Q6fDfaVf22pWMy7o7m0mWWN19VZSQRXzb8cf2P8A4efHaSS+1XT20vXmXA1fTSI5mOMDeMbZMcfeGcDAIr5evP2B/jD8J76W++GHxE3ooykcd1LYTt/skAlD+LAGvyvPOCYZlXliqFTlm901dP8AU+pweceygqc1oj9QaOK/MNfiZ+2x8Lo40vdKvPEUC8DdYwaiSB6tDlvxJq5/w8E/aI8MIP8AhIPhVEVUcvLpN3bk/iWI/Svz+twJmtN+44y9Hb80j245thpbto+t/wBrj4WfE74qeEtDsfhd4sbwhqdrfNNd3K6ncWPmwmMqF3QqzN8xBweOM16x8PNJ1TQfh/4Z0zXbw6jrllpltbX94ZWl8+4SJVlk3sAzbmDNuYAnOTya/O3/AIeveObX5Lr4ZWAk6HM8yc/QqaF/4KmfErVG2aZ8L7F5DwApuJefoAK2nwxnlTCwwbpR5YttO6vr3dxLH4RTdTmd2fpngUfhX5m3P7a/7VHiaPGjfDFrRX+7Jb+H7qU/XLMQfyqjdaP+2r8Xgi32pXvh2wn4b/SYNPVAf7yx4k/QmlR4CzOo/wB5KMfvf5IU84w8drs+9fi58YPBnwzsftXibxFY6RHGpbZPKPMfpwkYyzH2UHrX5y6Z4kk/az/bs8O+JvCmnXMWhaNc2lzLc3C9IbVt5d8ZClyNqrk9RnvjvPBH/BMq71TUF1H4k+OJtSkZt0lrpm52f/enl5+uF/Gvsj4Z/Cfwp8IdBXR/Cej2+lWnBkMYzJMR/E7n5mPuTX3/AAzwPQyLFTx85udWSt2SXZL/ADZ4mYZw8VTVCKtFanX0UUV+oHzAUUUUAFFFFMR+Vfi7/kbNb/6/Z/8A0Y1ZFa/i7/kbNb/6/Z//AEY1ZFeytjw3ubfhbH2ifP8AdH866XdswM9eM1zXhZd1zN7KD+tdATuZR0B68V9DhP4SPIxH8RkvzDaQc/0oppJSMnHIHGKKdbdF0dmeK7jt461vaNzZA5z8xrn8jvnkVv6KMWIz13GvLrbHfDcv16J+zz/yWvwh/wBfy/8AoLV53Xon7PP/ACWvwh/1/L/6C1cE/hZvD4l6n3J8dv8Akn83/XxH/M1zv7H3/JWJv+wZN/6HHXRfHb/kn83/AF8R/wAzXO/sff8AJWJv+wZN/wChx15Mv4bPYj/ER9pySLFGzucKoLMfQCvPP+GhPh//ANDCv/gLP/8AEV3uo/8AIPuv+uTf+gmvzqr1clyujmKm6smrWtY+Z4izqvlHslQinzXvc+2P+GhPh/8A9DAv/gLP/wDEUf8ADQnw/wD+hgX/AMBZ/wD4ivieivpv9V8J/PL8P8j43/XTH/yR/H/M+2P+GhPh/wD9DAv/AICz/wDxFH/DQnw//wChgX/wFn/+Ir4noo/1Ywn88vw/yF/rpj/5I/j/AJn3HpXxw8Ea1qVrYWWuLPd3MiwxR/Zpl3MxwBkoAOfU13VfBnwp/wCSmeF/+whB/wChivvOvkc5y+nl9WMKbbTV9T73h7Nq2bUZ1K0UmnbQKK4zVLqZNRuAJZAA7AAMcDmqv2yf/nvL/wB9GvBsfXcp3tFcF9sn/wCe8v8A30aPtk//AD3l/wC+jRYfKd7RXBfbJ/8AnvL/AN9Gj7ZP/wA95f8Avo0WDlO9orgvtk//AD3l/wC+jR9sn/57y/8AfRosHKd7RXBfbJ/+e8v/AH0aPtk//PeX/vo0WDlO9orgvtk//PeX/vo0fbJ/+e8v/fRosHKd7RXH6LczPqkCtK7KScgsSOhrsKRLVj82fiD/AMj94k/7CVz/AOjXr6p0P/kC6f8A9e8f/oIr5W+IP/I/eJP+wlc/+jXr6p0P/kC6f/17x/8AoIr1ZfCjzI9T8uPF3/I2a3/1+z/+jGrIrX8Xf8jZrf8A1+z/APoxqyK9RbHjvcjmfbt9zTR/e6VJICQPrzUBjcg5IUZ6g161D+GjlqbslEikZB2hu/NKmchWKjHamMiqwOQRgcCpCABvIJGMY/xroMCQOVkGFDH+tNyFLAljgFuc8GojlFRwTnvUyOS+5dpXuG600OwBiqgAAkdie1OjYYGDjnrnOaibLHAIB9cjmpYyGVQwBYHl8jAH0oQmKjfvAMkDGSPenybRiQDHqV9fWo/NAZ0JG4HqOMilUDy8g8Z5G7gVQaj9wU5fIXPPNPRhJK2FwTjnPaoThicMWXOMcVLCyxlSU43Y7nP5UkJkgck5xk4wcdqUsZECoeVHWo1kQOCvXOcHpT/LBcsCQQctg4FMQ6RCgBIyBwTnFe7+FiD8EZiOf9AvOv8AvSV4M3KMMM2G5zXu/hXA+B8+OR9hvcY9N0tBdPqeHIzEAn7vT8KCyB9oUbc/Lg8++aiUHzPmLKO1TOhjKOMElSM47/hxVGI9uWwoxxxjoaRWJJIbaeRgcZpCdpViFzu4HOMYpWnImIUYBOANtMdgVtxbKkrjt609XyML1/2sfpUbZkYkcHOSBnkUkoLJhSyqG6igRNkZBJ4+venKpJLLy/Ix6ioAxVFBIOOMAc/U1MWc7TGAuTzx2oE9B+c5OCvAyB7daFcoVwDlh2HakaQKAhOPm5HenEEBQRjtknpVahYVV8pcjKnPJzTkyxyikspyfeozhpFAOV6c96VA6SDBAULzk4z/APXpkj5CTLkscj26Z7Vk6p/x9nAwMCtUSiUYIAPYk8msnVP+Pv8A4CK4cX/D+Z00PiKleifs8/8AJa/CH/X8v/oLV53Xon7PP/Ja/CH/AF/L/wCgtXiz+FnpQ+Jep+k9FFFeMe0FFFVdU1Sz0TTbm/1C6hsbG2jaWa5uHCJGijLMzHgADuaBlqivhv4tf8FJIm1d/D/wn8Oy+JtRdjFHqN1E5jds/wDLKBfnkGM4JK/QiuK03wX+1l8atQh1HWPEt14Lsd6yoklz9jCjOcCCEF8gdnAz3NZ893aKuW4qKvJpH6KXd5BYWs1zczR29vCrSSSysFRFUZLMTwABySa8O8Q/txfBPw3O8E/je3u5lyCthbTXCn6MsZU/99V6HqmnjxF4budF1dzfWl5ata3S/wCr85WXY2duMZBPT1rwDR/2BPg9pMnmS6Pfamc5C3l9IVHthdv61rKE+ljnjXp/auM1j/gpt8JNOZltbTxHqhHRreyjRD+Lyqf0rlbz/gqn4Ojz9k8F63P/ANdp4Y/5Fq9l039l34T6Tt8jwFop29DNbCU/juzmuosPhT4L0sKLPwlolqF6CLT4lx+S0vZ1Hu0DxFJbRZ8tTf8ABVyx3HyPhpdSr2L6wqn9IDVZv+Crh/h+FchX1Ov4/wDbWvsePw3pMQATS7JB2C26j+lSLounjgWFqP8Ativ+FHsp/wAwvrUP5fxPjiP/AIKuQZ/ffDC4jXuV1sN/O3Fatj/wVV8MSEfa/AmrQDuYbyOT+arX1g2gaY4w2nWjD3gU/wBKzdQ+HnhbVVK3vhvSbpT1E1lG4P5rR7Kfcf1qn1j+J4Npv/BUH4W3TKt3onieyJ6t9lgdR+Imz+ldxo//AAUA+COrtGjeKptOd8ALeafOuD6FlRlH1zitrUP2bvhZqQbz/AOgnd1MdiiH/wAdAriNe/YV+DuuBynhuTTJG/jsbuVMfRSxX9KXs6i6oaxFJ7po9w8NfHD4e+MZIotF8a6FqM8gykEOoRea3/AC278MV2ysHUEEEHoQeDXwP4o/4Jp+GZsv4f8AE2qae/UC4RJx+AGw/qa4X/hnX9oP4LtJL4E8cTahaxMGWzt714mYerQy/u/w3Gl+8juvuLVSlLRSt6n6Z0V+efhH/goL8QvhbrEOi/GDwdLMmQGvIIDa3W0dWCt+7lHT7pUe5r7g+GXxS8MfGDwvB4g8K6pFqenyHaxT5XhcDlJFPzKwyOCOhBGQQaUZp6dTRxcVfodZRRRVkBRRRQAUUUUxH5V+Lv8AkbNb/wCv2f8A9GNWRWv4u/5GzW/+v2f/ANGNWRXsrY8N7m14YYrcT4/uD+ddAcA9eevSuf8ADIJnmA7qP510YAZwo64NfQYT+EjyMR/EYxSW5xgDnmilUEMQTjA6Zoqq26Lo7M8XABGSM88fSug0cYswOeprAiABx2P6Vv6Rj7GMeprzK3wnfDcu16J+zz/yWvwh/wBfy/8AoLV53Xon7PP/ACWvwh/1/L/6C1efP4Wbw+Jep9yfHb/kn83/AF8R/wAzXO/sff8AJWJv+wZN/wChx10Xx2/5J/N/18R/zNc7+x9/yVib/sGTf+hx15UvgPYj/ER9m6j/AMg+6/65N/6Ca/Oqv0V1H/kH3X/XJv8A0E1+dVfW8LbVfl+p+c8cb0P+3v0CiiivvT8tCiiigDqvhT/yUvwx/wBhCD/0MV9518GfCn/kpfhj/sIQf+hivvOvzbif/eKfp+p+xcE/7tU9SNraFmJMSEnqSoyaT7LB/wA8Y/8AvkVVm1yzglaN5SHU7SNp60z/AISCx/56n/vk18UfpGpd+ywf88Y/++RR9lg/54x/98iqX/CQWP8Az1P/AHyaP+Egsf8Anqf++TTDUu/ZYP8AnjH/AN8ij7LB/wA8Y/8AvkVS/wCEgsf+ep/75NH/AAkFj/z1P/fJoDUu/ZYP+eMf/fIo+ywf88Y/++RVL/hILH/nqf8Avk0f8JBY/wDPU/8AfJoDUu/ZYP8AnjH/AN8ij7LB/wA8Y/8AvkVVi12zmkSNJSWZgqjaepq/SFqRfZYP+eMf/fIo+ywf88Y/++RUtFAiNbeJGBWJFYdCFANSUUUwPzZ+IP8AyP3iT/sJXP8A6NevqnQ/+QLp/wD17x/+givlb4g/8j94k/7CVz/6NevqnQ/+QLp//XvH/wCgivVl8KPNj1Py48Xf8jZrf/X7P/6Masitfxd/yNmt/wDX7P8A+jGrIr1FseO9yOb+E9geabvJAAAA9xzT5WKgEHjPI9aYsgJO4rluMYr1aHwI5KnxCghFIPzYOOB2pSSvOM5FMyBtIIxnr3pSyhuRjHbrXSYkiIZMgfMRx0pQoRsFfT86iA80gg5LZIH096cMR53DP485oAeCVUBOGY/MO4ApS27JC7s9+9ITuZiAVz6cnFIyqrDBPCjv3oAexABwNpB64606OQKmMfM3TPSoy6BwuMnvz1p6r5GMgGNiSTnpTuBKvDbiwHtilP7tMAkndnpwBUcZ3ZYHPYDuBT5CrTAF88bTx09qoXUVWGMkc+wp8kvnSYADDgcdRUJUhx7fjmpBhgGBxnsBihMTHFCWwN3XBANe8eFPk+BsxyGxY3p4/wB6WvBA5Utu+8DjdmvfPCTBvgfKQCubG84/4FLQjSG79DweMmVlGMHlsA0oleRyg4wB0HekZQWUgkYOMAVK0mFBG0BSOG6igzAONpJPQYz2FPWaLBOG3n3xnNRliHwSx3DPBxmlZkVXY7vmxj1B7Zp3IJEDhck8L3P8qcE8zA3bVPHXgkVAWP3OGIxxmlLlSBvO49OOlO47EjYBUlhuzjHpUpyzBAB06jtUG1sbm56Zp87BnJHDdMe1MTHKv8RAxyAfpUgcbeSVXGCR3NQISEBJ3EevH6VKXC4BLH0UcUIBQBwo5I9PenFiY85wOnSk3CFuEOevXihGDoSAFOMmq1AkUBYxg7++aydRYtdEnrgCtLf82ELHIHJ45/8ArVl3x3TfgK4sU/3fzNqC94r16J+zz/yWvwh/1/L/AOgtXndeifs8/wDJa/CH/X8v/oLV4s/hZ6UPiXqfpPRRRXjntBXxd/wVE8Valo/wp8NaPaXDQWWraiwu1Q481YkDKp9V3EHHqor7Rr4W/wCCqw3eCvAQ9dRuB/5DWs6vws1p/Ejvv2MfDHw30/4U6TqPgwWt3qs9sg1a9YKbsXBUF43z8yqG6KOMAEZzk/QdfFepf8E4fiT4I0XRPHXwS8XSzXF9p1vdyaXPcfZLtTIiOyJIMRyruOdr7eAAdx68+37XHx3+Bs5sfij8O5ruKJzGbu5tXs2kYdds6K0TD0Kr+Jp06yikmrHNWwspScou596UV8n+Ff8AgpB8N9Ykji1jTtZ0CRh80jwrcQqfTcjbj/3zXqfh39rT4ReKG22njnTYX7rfFrX9ZVUH8DXTGpCWzOKVGpHeLPXaK5vS/iX4Q1sgaf4p0W+Y/wANvqELn8lat+O7gmAKTxuOxVgapNPYzcWtyWiiiqJswoopjzxx/fkVfqwFAWY+isXVPGvh7RFLajr2mWCr1N1eRx4/NhXFeIP2nPhX4ZhaS98daO23qlpci5f/AL5j3GlzR6spQlLZHp9ZOvJHsjbgS5x7la+cvFf/AAUO+GGjR7dMk1LXJOg+yWhQD6mUr+ma8f1f9u7xr4+vpbH4e+AJru6b7skiS3suP73lxKoX8SwrN1qceposNVmtI/efV/xC0Pwvr3hW+g8YW9jPoaxl521DaEjABy4Y/dIHRgQR2NfIf7A/iODw/wDtV674e8L301x4Q1KG7Eazf8tY4iWhcj+8BkZ4OGPAzitbRf2Rvj/+0leQXnxK1xvDGgs6yi3u2VnAIOClrEQoIHGXKkZ781R/ZZ8B2Xwv/b61bwpp009xZaTFfWsU1yQZHCxj5jtAGSeeBXJUqe0kmlY9LD0fYxacru23Q/TCiiitQCiiigAooopiPyr8Xf8AI2a3/wBfs/8A6Masitfxd/yNmt/9fs//AKMasivZWx4b3NrwyxW4mI/ujP51vs4DqQetc94cOJZ+Odo/nW6XTOAOD+lfQ4T+EjyK/wAbHqgAJJOKKEJKgZJ9Dmiitui6OzPGVYk9cjHrW/ouPsIwP4jXPg9MHg+9dDoxLWIz/eNeZW+E74bl6vRP2ef+S1+EP+v5f/QWrzuvRP2ef+S1+EP+v5f/AEFq4J/CzeHxL1PuT47f8k/m/wCviP8Ama539j7/AJKxN/2DJv8A0OOui+O3/JP5v+viP+Zrnf2Pv+SsTf8AYMm/9Djrypfw2exH+Ij7N1H/AJB91/1yb/0E1+dnlP8A3G/Kv0bIDDBqt/Ztl/z6wf8Afpf8K7cqzZZappxve3Wx4meZDLOnTcZ8vL5X3Pzt8l/+ebflR5L/APPNvyr9Ev7Nsv8An1g/79L/AIUf2bZf8+sH/fpf8K97/WqP/Pr8T5f/AFGn/wA/v/Jf+Cfnb5L/APPNvyo8l/8Anm35V+iX9m2X/PrB/wB+l/wo/s2y/wCfWD/v0v8AhR/rVH/n1+If6iz/AOf3/kv/AAT4Y+Fcbj4leGCVYD+0Ye3+2K+8Krx6faRyBktoVZTkERqCD+VWK+bzTMlmVRTUbWVt7n1+SZM8mpypufNzO+1jmL/Qbu4vZpUVSrMWGWHSoP8AhG77+4n/AH0K66ivFufTczOR/wCEbvv7if8AfQo/4Ru+/uJ/30K66ii4czOR/wCEbvv7if8AfQo/4Ru+/uJ/30K66ii4czOR/wCEbvv7if8AfQo/4Ru+/uJ/30K66ii4czOXstAu4LyCR1Xajqxww6A11FFFITYUUUUCCiiimB+bPxB/5H7xJ/2Ern/0a9fVOh/8gXT/APr3j/8AQRXyt8Qf+R+8Sf8AYSuf/Rr19U6H/wAgXT/+veP/ANBFerL4UebHqflx4u/5GzW/+v2f/wBGNWRWv4u/5GzW/wDr9n/9GNWRXqLY8d7kcwJ24IAzz9KjKAfXPUinXWdqkdQenqKYWLYJGOK9Wh8COWpuAkw4QkZJ6etN3EOSO3BFdJ4V+H+u+L5R9gsttsDg3U3yRr/wL+LHouTWdr2mwaHrF1YxXK3iW7eU06LtVnH3sDngNkA98Z4ziunUz6FPzdrsSTtPX2owCQAdyt0YikV1Zgo+buc0jEiQDAK+3YUEDpME4AyCMgD1pecLuHzcDmm8Fg7HaAOMcGpJGztUfN3B6GgBdmASWxuP4UgIbJ8zA9+9JIjs55GR2HANKAJVwQE6ZwKAHjapByQw55oLiT58YDc01ivII5xjnGDSJuCAbjhe2eKdx26kzqdy5T5sc49KcxMU2QTggH2x6VCWbYFUkuepz0FWAqrGuP8AWMRu3HOAaaJehGAYixJGPQda9+8Jtn4GzNk/8eN6ef8AelrwDcVmLA9ORx0r33wmxb4Fzknk2N6ckf7UtCNI7v0PCi0Q2gtyeRxxn1o8zqATyeSe30+tM8oscltxUcc8UNJk8KSvPJHNMysiUMUYHBPPHPSnuxKpnLDvnufemYURBgCCfXmkRgY/mf8Ah+UDPegXmPX/AFhycZ7UScKg2j72M96as3lqvy7iOvNPILYZgeDnIxxQMXktkgYzx7VKxEahiRnAGDVdV+bqwOc59qeXHGW3FuBlenpVE2J8r5e0thjzgVGsgORuxzwT3pmW8wBju44xTwP3jEDIxgcDigLWJowVI3fKxyAaRiC3IOBgZHfmmKncknHABHSlXKqctjdwSRmmIeSBIcBmIIxWbfnNyTnqBWgoYngjGOKzr0YnP0FcWK/h/M3o/EV69E/Z5/5LX4Q/6/l/9BavO69E/Z5/5LX4Q/6/l/8AQWrx5/Cz0YfEvU/SeiiivHPaCvhf/gqp/wAiV4A/7CVx/wCi1r7or4X/AOCqn/IleAP+wlcf+i1rOr8LNafxH6MfCf8A5JZ4N/7Atn/6ISulurO3v4GhuYI7mFhho5UDqw9CDwa5r4T/APJLPBv/AGBbP/0QldBq2rWWg6XealqV3DY6fZwvcXF1cOEjhjVSzMzHhVCgkk9AK5xnlPjT9kH4M+PpHl1n4daHLO+S09tb/ZpGPqWiKkn614z4k/4JS/A7XJnksotf0Ld0jsdRDoPoJUY/rX0J4N/aH+GXxCuktvDnj3w/q90/S2ttQjaU/wDAN279K9AjmjlGUdXHqrA0rJlc0kfnjr3/AARw8MTTFtD+JGradHnhLywjuT/30rp/Kudn/wCCP2uWSk6Z8XdrD7ofTXjH5rMcflX6Z0UcsR877n5b3H/BKr4w25/0D4s6e4HTzLm8j/8AQVaoP+HYf7QCfKnxX0jZ3/4mt+P08qv1Poo5UHMz8tI/+CWXxsuW/wBM+K2mKp6lL29kP6otalr/AMEivF18v/E3+L0YJ6iOzmmH5tIv8q/TeilyhzM/N/Tf+CNumiZW1T4pXtzEPvR2+kKhP0Zpmx/3ya9F0H/gkj8G9NaNr/U/E+rMuCyyXsUasfosYIH4/jX23Xxx/wAFBP2xNa/Z9sdE8JeB0jbxtr6NIlw8QlNpDu2KyIchndshcgj5W4JxSdoq7Li5SfKj0rwj+wf8CPBrRvZ/DvTbuaPGJNTZ7sk+pErMv6V6hqPhnSPDOgxW2kaVZaXbLIoWKzt1iQDB7KAK/KrUvGf7XfwTsf8AhYWq+LtS1GzBWa80+7vheRQqzDIeBsqi9iUxtz1Ffod8CfjxaftIfAfQ/GcFutldzTNbX9mrZWC5jBDqD12nKsuedrrnmscNiqGKTlQkmk7XTvqa4rC18K7Vk02r66HU1+cvwk/5SdeMv+uuof8AoAr9Gq/OX4Sf8pOvGX/XXUP/AEAV2S3j6nHS2fofoVRRRXQZhRRRQAUUUUxH5V+Lv+Rs1v8A6/Z//RjVkVr+Lv8AkbNb/wCv2f8A9GNWRXsrY8N7mz4bz5s/+6P51utwcYBIHNYHh8nzpgO6j+db6oC3P3sHv1r6DCfwkeTX/iMF+RQg+XcDRSMASvIJAx70U625VHZnjOApyB7iui0T/jwHOfmNc7gscg4Aro9F/wCPEf7xry6vwnfDcvV6J+zz/wAlr8If9fy/+gtXndeifs8/8lr8If8AX8v/AKC1cM/hZvD4l6n3J8dv+Sfzf9fEf8zXO/sff8lYm/7Bk3/ocddF8dv+Sfzf9fEf8zXO/sff8lYm/wCwZN/6HHXlS+A9iP8AER9m6h/x4XX/AFyb/wBBNfnx50v/AD0b/vo1+g+o/wDIPuf+ubf+gmvz0r4DPG707H7ZwHGMo1+ZdY/qSedL/wA9G/76NHnS/wDPRv8Avo1HRXy3NLufrHs4fyknnS/89G/76NHnS/8APRv++jUdFHNLuHs4fynWfC6aT/hY3hrLsQdQh4LH+8K+56+Ffhb/AMlI8N/9hCH/ANDFfdVfY5J/Cl6n4vxzFRxVLl/l/U5m+8QXdveTRJs2oxUZXnFQf8JNe/8ATP8A75/+vW3PoFpcTPK4fcxLHDd6Z/wjVj6P/wB9V9Lofm10Y/8Awk17/wBM/wDvn/69H/CTXv8A0z/75/8Ar1sf8I1Y+j/99Uf8I1Y+j/8AfVF0F0Y//CTXv/TP/vn/AOvR/wAJNe/9M/8Avn/69bH/AAjVj6P/AN9Uf8I1Y+j/APfVF0F0Y/8Awk17/wBM/wDvn/69H/CTXv8A0z/75/8Ar1sf8I1Y+j/99Uf8I1Y+j/8AfVF0F0Y//CTXv/TP/vn/AOvR/wAJNe/9M/8Avn/69bH/AAjVj6P/AN9Uf8I1Y+j/APfVF0F0Y/8Awk17/wBM/wDvn/69H/CTXv8A0z/75/8Ar1sf8I1Y+j/99Uf8I1Y+j/8AfVF0F0UdM166u76KJ9mxiQcLg9DXR1n22hWtpOksYYOpyMtmtCgnTofmz8Qf+R+8Sf8AYSuf/Rr19U6H/wAgXT/+veP/ANBFfK3xB/5H7xJ/2Ern/wBGvX1Tof8AyBdP/wCveP8A9BFepL4UeZHqflx4u/5GzW/+v2f/ANGNWRWv4u/5GzW/+v2f/wBGNWRXqLY8d7mhoXh258UanFYWktvFPJ91riYIv4Z5Y+ygn2r3Hwh8DdH0IJPqR/ti8XnEq4hU+y/xf8CJHsK+dr1WKKV+8pyD6V2/g34ya74YWOG4l/tWyH/LG5bLqPRX6j6HIHpXq4drkVznlbm1PpK+tJJNLntrOZbGRomjilVNwiJGAwUEdOwzXj//AAzcMnHiI8nJzZZz/wCRK7HQ/iRoXxA0+axt7+TS7+4jaNY3by5kYjAZGBwxB5GDnjkV4PqniLxXomqXlhd67qiXNtIYmAvZcZB6jLdD1Hsa6m0xStuejH9mwMQf+EhII/6cv/tlSD9nPCFR4hx7/Yv/ALZXlZ8ceI32qPEGpg92+2y8/wDj1Snxz4hVGH9vamTjAxeS5/8AQqSsRddj00/s4ZUg+Ievf7D/APbKVP2cdgIHiE9sf6F0x/20ry1vG/iMISNf1Qbh3vZePf71H/CbeJAQTr+qEbf+f2T8/vUaBp2PUz+zqSAP+Eh6f9OP/wBspR+zrgH/AIqHOf8Apy6f+RK8xHjfxG4cf29qYJ5XF3Jn/wBCpY/G2v4BfX9UABwT9sk5/wDHqrQV49j0z/hnMFSP7fGTxn7F/wDbKP8AhnLkEeIen/Tl/wDbK8v/AOE28REkDxBqefT7ZJ/8VUyeNfEMZCtrupsWI5N7J/8AFVN0wvHselL+zoQOfEJJ9fsWP/Z6l/4Z4ByTr3XH/Ln/APbK8ybxt4gPA13U/TBvH/8AiqU+MvELOVGu6ocYAxeS9fT71VoF49j0s/s8ZJP9v9f+nL/7Ouym8PDwr8LdS0o3H2oW9hdfvdmzduV26ZOMbsde1eBf8Jp4gdlB1vVBtzk/bJOf/HqbL4s124t2im1nUJYpFKukl27IVYYKsCeQRnrTBSj0RlBmOQAxye9OaTcpBO0ZBGfftimhiWABwwOPWnkMVwQepA4oMWERGeCW56Z71NwAWALHgbR2qJCNgVfocU75Y92G74oAdkFAAAST261Ir/KwK8E8g9qhjcgKQoUg8YqRMOQCAZME5zimhbCFggAw3zcg5pU3iMtjPr3P4U0klSSencdKl4dMgMvfGKEDGxOGyQclevFSRkeYpX5t3H51CGK544bn3FSRZDE45AxQhMkdjjZnlSRkGkZgF2g5yPQ00q/lEnC85/8Ar0u8leMAsOo7VQhytgDjacAAetUL05nOR2FXYVK/Ox4GQCao3v8Ax8Nx1xXHif4fzOij8RBXon7PP/Ja/CH/AF/L/wCgtXndeifs8/8AJa/CH/X8v/oLV40/hZ6EPiXqfpPRRRXjntBXwv8A8FVP+RK8Af8AYSuP/Ra190V8Lf8ABVfP/CE+A8df7RuMf9+lrOr8LNafxI/Rn4T/APJLPBv/AGBbP/0Qlani/wAK6X478Lat4d1u2+2aRqlrJZ3cBYrvjdSrDcpBBwTyCCOor8xf2dv+CjXiX4BaZpXgL4y+FtTn0+ztoksdUjh2Xkdtt/db432iZMFcOGB2j+I8197fDH9rL4SfF6KL/hGvHWk3F1IARYXUwtrrPceTLtY49VBHvXMpI1cXHU+XfGv/AAR8+H+qytL4Y8Y694fZiT5V4kV7GvptACNj6sfrXAR/8EzPjr8Prpv+EB+M8dtCPuut7e6azfVYi4/U1+nYIYAggg8gjvS0uVBzs/L2b4N/t5eAp9ul+MLjxCqn/WRavBcrx7Xaqf0q1/wmP/BQbQUzc6PJqSr1Js9Lcn8IiD+lfpxRRy+Yc/dI/MB/2kP249L4uPh3Jc49NAL/APot6Z/w1x+2mrbD8Kptx7/8Ivdfz34r9QqKOV9w5l2Py/X9pr9t+9/1Xw2ki/7l51/9Cep0+If/AAUA8QLutPDr6ev/AF4afGR+ErE1+nNFHL5hzrsjwT9j2T43y+A9Ub45+T/bv24/YdqW6y/Z9q53i3Ajxuzt/i657V8q/wDBUz4X+I9E8f8Agf4zaJYSalYaRHFbX21C6WrxTGWF3A5CMWZS3QFQCQWGf0lqK5tob23kguIkngkBV4pFDIykYIIPBB9KJQUouL1TCFRwmpx3R+P3xe/bx0j4kfC2/wDDGgeGtRj1zWoBZzLcbWiiDjDbCpLO3YZVeufavtH9hv4M6x8FP2WdMsPEFtJYaxrGpSazcWMww9v5iIiIw6q2yJGKnkFiDyDXuul/BX4ZeDNZOt6d4J8N6Tqm7cLy202FJQ3qrKuQfpXPfFX47+CvBNvnxH4m0vQreM7gl7dIkshx/CmdzcdgCa87Lssw+WQcMOrJu71v+Z6GYZlXzGSlWd2lZaGtX5y/CT/lJ14y/wCuuof+gCu3+Mn/AAU00KzWXR/hdpN14k1ibMUWo3cDR2wYkAFIv9ZIevBC845PSvB/2I5fEVz+2lLP4sS6TxJPb301+t4nlzCZk3NvXA2nJ+7gY6YFeo5JtJdzzYQcU2+x+pdFFFdRgFFFFABRRRTEflX4u/5GzW/+v2f/ANGNWRWv4u/5GzW/+v2f/wBGNWRXsrY8N7mt4fYedKDnkD+db5x1JOO31rn9B/1sx64A/nW4rFhj8K+gwn8JHlV177H52MMD8faimM2T8vbg4opVt0VR2Z43knvx9K6TQyTYKT3Y1zWSOQeT3rpNC/5B4/3jXm1fhO+O5oV6J+zz/wAlr8If9fy/+gtXndeifs8/8lr8If8AX8v/AKC1cM/hZtD4l6n3J8dv+Sfzf9fEf8zXO/sff8lYm/7Bk3/ocddF8dv+Sfzf9fEf8zXO/sff8lYm/wCwZN/6HHXlS+A9iP8AER9m6j/yD7n/AK5t/wCgmvz0r9C9R/5B9z/1zb/0E1+elfn+eb0/mft/APw1/WP6hRRRXyx+shRRRQB1Pwt/5KR4b/7CEP8A6GK+6q+Ffhb/AMlI8N/9hCH/ANDFfdVfY5J/Cl6n4px1/vVL/CNMqA4LqCPek85P76/99CuO1SCVtRuSI3I3tghT61V+zzf88pP++TX01j815Tu/OT++v/fQo85P76/99CuE+zzf88pP++TR9nm/55Sf98miwcp3fnJ/fX/voUecn99f++hXCfZ5v+eUn/fJo+zzf88pP++TRYOU7vzk/vr/AN9Cjzk/vr/30K4T7PN/zyk/75NH2eb/AJ5Sf98miwcp3glQnAZST706uJ06CVdQtiY3AEi5JU+ortqRLVgooopCCiiimB+bPxB/5H7xJ/2Ern/0a9fVOh/8gXT/APr3j/8AQRXyt8Qf+R+8Sf8AYSuf/Rr19U6H/wAgXT/+veP/ANBFerL4UebHqflx4u/5GzW/+v2f/wBGNWRWv4u/5GzW/wDr9n/9GNWRXqLY8d7kVyQFBYZAzx79qrBgF6jLcgd6sXLbVXkDccZNb/wsUn4o+DyRnGsWf/o9K9KlLlpXOdrmnY5r/VMANuSDg57VLc3897N5s9w9zKFVd0jbm2gYAJPPAwB7AV+iP7UnxD/aS0H42avYfDu18US+F1SD7M2neH/tNvuKLuxKYWB+bOfmwOelcF+1z4X1Pxv4V+DVrr+n2a/HHVj9lv7W1VFmkjY4Rp1j+VTuKnOMDL4wAcctPGc3LzJJPs7teqOuWFUbqLd13Wj9D4obgYOOecmjblA4JI6AkdDX1JN+xb4c/wCEsk8Cw/GLS5PiasJYaA2lTLbNOF3eR9r37d2O23P+zXJeB/2X1n8E674t+Ifi5fh5oel6m2jMzabJqE8l0pw6iKN12qp43ZOcHtgnpWKotXv+D/Duc/1eqna34nhBPyYPIz6frmlyAVZDuyQM173Y/sz6B4g+Lvgfwj4b+J2k+JtH8VRyPFqtjbf6RZssbOVntGk3Rk4GAzAn5uBtweoj/Yv8P6vrmt+EfDfxZ03XfiFpMEkkvh86VLDE7Jy0aXJdlZl6EKpwcg4wSB4qkuvns/xD6vUa0XW258wArG6knJIz6YqUIJFLEArk9Byc+le0+A/2bLXWfh3P49+IHjGD4e+GWu20+zlmsJL65urhWKuqwoythSrAnOco2QAM1tah+yO9t4o+H8Wi+MLXxB4K8bXS2tj4ntbNkMcnO5ZLdnDKwwflLdmBwRir+tUrtN/g7EfV6lk0u3U+etqkqMEn1Axj60rIeAfmbpxzivquH9ijw3e+ONS8BWPxfsrn4g2qSumjf2LMsTlAWVWuN5VWK7SygNtyR82M1wngf9miyvfAOoeN/HvjOPwFoNvqTaVEw0yTUZprlSd67I3XaqkEbiecHjGCZ+tUbXv26d+3cf1epe1vxPE5NoAHBbGeDQiske8kEqfXvXtV9+zZHrXxI8K+FvAHjbR/HkHiCIzw39oDE9oqjc5uoNzNCVXnDHJ5GAeK6yX9knw/4gm8Q6L4G+K9h4x8aaFE80+gDSZLRJ/L/wBYsM7OyyMp44GM9SOtU8VRja738np69hLD1JXsvxPmYoeWQE45znpUrOGUKAAOc59qYd8buh3IVbBXGCD6UKPlI6NnOT611o5mPjlY4ZVJxzn17U/zC8jEknGSV7ECoQD5hwQvuOgqZPu434HXOOuetNCYockHICrjnPFACM64PLdm7+4NMLAkD3zwKVwOcYU8Z+noKYrE6HgkA7hnk9KFUliucMMnA4x9ajjyOCTleg9alJkaFmOCWPY4PvTQmI0xVlUgEsPuml2qzBySpXng4pUkxtznI6ZpHBHmYwSzckHmkMUn94GJ3Z5AxT1YgbuCvPHocUwsFUqQcYHOKBkkHOCMnj0qhWHNJjZtHDAhvrU6qAuQME5471W3ByuTkjqPapGJyu0/N0GDQhMXzAq4PVeck/0qnf4NwSOQQKuBm8wFgVxwQfWqF0cykgY46Vx4r+H8zej8RDXon7PP/Ja/CH/X8v8A6C1ed16J+zz/AMlr8If9fy/+gtXjy+FnoQ+Jep+k9FFFeOe0FfC//BVT/kSvAH/YSuP/AEWtfdFfC/8AwVU/5ErwB/2Erj/0WtZ1fhZrT+JH3Vp3wj8G/F34KeDtM8Y+G9P8QWf9i2gQXsKs8RaBAWRvvI3+0pB96+afiR/wSM+HHiGWa58H+INW8ITtylu5F5bKfYMQ/wD4+a+xPhP/AMks8G/9gWz/APRCV1dc3KupfM47M/Lxv2Cv2n/hBCq/Dv4pm8soGzFZ2esT2WfrE/7v8CcVNd/ED9vf4ZGNtQ0K58SW8eBth020v94Hqbcb/wAeDX6e0UuXsVzt7o/Meb/goh+0n4QiDeJvgsiRr96SXSL62Bx15LEUWP8AwWJ1i1OzU/hTG0inDeRqjx4P0aJv51+nFQXFhbXfE9tFN/10QN/OjlfcOZdUfnTb/wDBZDSTj7R8MNQj9fL1NW/nGKtj/gsd4axz8NtYz7X0X/xNffM/gvw9dDE2haZMD2ks42/mtVD8NfCDHJ8K6IT6nTof/iaevcLx7HwXP/wWP0Mf6j4Zao5/29QRf5Iax7//AILH3DAiw+FLZ7NPq5P6CH+tfonB4D8NWpzD4d0mE+sdjEv8lrRt9GsLPHkWNtBjp5cKr/IUrPuF49j80LX/AIKc/HDxnx4T+DUF2GHyGG0vLs/+ObQagb48ftzfE26MWj+CLzw3G393Q1tU/wC/lzn/ANCr9Qq8/wDjZ8dPB/7P/gyXxL4x1H7FZK3lwwxLvnuZMEiONf4mOD1IA6kgc0a9WCkukT89h+yX+2R8W7x5PFvjuTw/byjEiTa2VQg9f3VtlfzxXRaB/wAEjdK0m3S+8bePb3WJ2kG+30uAQqSck5kcsxzjrtB5rvfDf/BXb4Xat4iSy1Lw74g0XTZZFRdSkSKVVBON0iK25VHU7dx9jX2Bf69p3ijwjp+r6TeQ6hpl8I7i2urdg6SxspKspHUEGiKUn3HNyiux4j8K/wBmr4b/AAZCv4W8L2dpejP/ABMJgZ7k5GCBK+WUEdgQPavjL4R/8pOPGX/XXUP/AEAV+jVfnL8JP+UnXjL/AK66h/6AK0kkmrdzKm3Lmb7H6FUUUV0mQUUUUAFFFFMR+Vfi7/kbNb/6/Z//AEY1ZFa/i7/kbNb/AOv2f/0Y1ZFeytjw3uamgsBLNnuo/nW7HgYBPHtWFoOPOlyM/KP51sY5ycivfwn8JHlYj42TAFGbOAO3uKKaAETJ59qKqtuiqOzPGyQDnNdJoRzYKefvN1rmiCVJOK6XQSDp6kf3jXlVfhPQiaFeifs8/wDJa/CH/X8v/oLV53Xon7PP/Ja/CH/X8v8A6C1cU/hZrD4l6n3J8dv+Sfzf9fEf8zXO/sff8lYm/wCwZN/6HHXRfHb/AJJ/N/18R/zNc7+x9/yVib/sGTf+hx15Uv4bPYj/ABEfackayxsjjKspVh6g151/wzz4D/6A7f8AgTL/APFV6M5IRiOoBrif+El1H/nuP++F/wAK+EznNsFljgsVByve1lex9blv11qX1So4bXs7XM//AIZ58B/9Adv/AAJl/wDiqP8AhnnwH/0B2/8AAmX/AOKrQ/4STUP+fgf98L/hR/wkmof8/A/74X/Cvmv9asn/AOfL/wDAV/me1y5z/wBBD/8AAmZ//DPPgP8A6A7f+BMv/wAVR/wzz4D/AOgO3/gTL/8AFVof8JJqH/PwP++F/wAKP+Ek1D/n4H/fC/4Uf61ZP/z5f/gK/wAw5c5/6CH/AOBMh0n4HeDdF1O11Cz0poru2kWaJ/tEjbWByDgtg813tchp2v31xf28bzAozqrDaoyCfpXX19Zk+ZYXMqcp4WLSTs79z5/Mli1OP1qo5u2jbvZBRWPc+JIrW4kiMLsUYqSCOai/4SuH/njJ+Yr6Gx5Opu0Vhf8ACVw/88ZPzFH/AAlcP/PGT8xRYfKzdorC/wCErh/54yfmKP8AhK4f+eMn5iiwcrN2isL/AISuH/njJ+Yo/wCErh/54yfmKLBys3aKwv8AhK4f+eMn5ij/AISuH/njJ+YosHKzdorC/wCErh/54yfmKP8AhK4f+eMn5iiwcrN2ismz8RRXlykKxOpc4BJGBxWtQS1Y/Nn4g/8AI/eJP+wlc/8Ao16+qdD/AOQLp/8A17x/+givlb4g/wDI/eJP+wlc/wDo16+qdD/5Aun/APXvH/6CK9WXwo82PU/Ljxd/yNmt/wDX7P8A+jGrIrX8Xf8AI2a3/wBfs/8A6MasivUWx473IbogKMhcHOcj2rd+FDD/AIWh4PwWx/bFmOR/02SufviQqdcEkHBqrwuVLZHUEcV6VJc1O3qc7dp3PtD9rz48eOvhX+1xqVx4f8S6pa2lgLSddNN5IbOQeUpdWh3bSrDIPHfIwcGur8Waf4V0r4j/AAt/aX8N24t/BmrX8Y8Q28XzDTrtgY2cqOAu4sGx/EoIHz18CFSGAHTI49vWpC7dCBtHQ5rm+px5YpOzSs9N0dP1p3bavd3Wux+mGv6F8UW+LV34h8K/C/4RXehSXLajYeP72AKojOWWaWZbgybwMBmVOTyBjpxHwv8AEfxU8T2vj3xF8P8AxL4S8ba5da2/9s+AUswdNlUjb9qt5J5UdkYKP7ob5uWIGfghGODsOAPT09KRsKv+yTnA7VmsDpa67bfnd/kX9b1vZn6Qw+E/CuhftG/Ae7/4R3SPBPxJvXupNc8PaFIjW9un2eXYzKhKoxOcY6/Ny23Ncr8M/wBn/wARfBr9pPWPil4turLTvh/pE9/qEWufb4XS+WVZAiKoYsWO/lSB8y4GeM/IvwO+Jn/CnPipoHjIad/bA0mR5fsXn+R5u6N0xv2Nt+9noemK5vxXrieJ/E2sayIvs/2+8mu/J379nmOzld2BuxuxnAzjpTWEqJuKlo1ZvfrsgeJhZStqndI+xvEOmXf7XX7OGg6f4DS3ufFHh7W725vPDjXUcNwbeeV3WVQ7KpwHXJzj7wBJGD1nhfT4/g/pHwA+FOq3trL4zj8WDWb/AE+CVZmsY2EgVHZSQrEuvQ84bGRzX589VAB+QDOTzilVxJwPungAdfxrR4O65FLS7a01uyPrVnzcutrPXS3ofYfwzYt/wUslZmyTr2pY3df+Pef+ldT8Gj8Sbbw742ufhlrGj+M5z4ku01T4ea3ZxlIkaRwLhXeVCwYBcqNo+91K8/CwYghi2Npwo9qVs4ZxjIYE+uDVzwfN1WyWq7ExxPL06t6eZ+kFtrHw8+DPx5+GWq6ro2ieAPGWtaXdWWv6RpcqNZWEkir5LSBTtjJZSvynoxySBuqrpek/GP4eeKtW1yH4ZfBrwZp+nxyyr4xuIGt4J4jnkNFcNLl1OcFfY9q/OkDkknbnpxTkcOen3TyQOcetYfUP719Nd/w1NPrn923Vf8E0dY1JtW1jULyRYA9zcSTsbdWWMFmLHarfMF54B5xjNUVChlJf7xxkjr6Vt+CfB17421WaysJbeOVIWmLXLMq7QyrgbVJzlh29a7E/s8+JCMf2hpf/AH9k6f8AfFcWO4kynK6v1fF4hQmknZ72FRy/FYqPtKUG0+qPM+VZlI7/ACkHqKVAXBUbUI4+or01P2evEQCE3ulkrn/lrIR/6BTl/Z88QrjF3pYx6Syf/EV5/wDrpw//ANBcTf8AsfH/APPpnmhIC4i445HejcpGXHbnsCa9L/4Z88Qh8i80vHf97Jk/+Q6d/wAM+a+Ol3pZIORmWT/4in/rpkH/AEFxF/Y2P/59M8zQMeSAQRxg4oRT8xySoyMHvXprfs/+IcYF3pfXP+tk/wDiKd/woHxBxm70s7TwPOkx/wCgUf66cP8A/QXH7w/sbH/8+WeaqF3KSOVHTpQCVVfkzu/uivST8AfEBbJvNMz/ANdZP/iKevwD8QBsm903GCOJZM/+gU/9deH/APoLj94v7Gx3/PpnmhBzycE8ZznildwQDnknjI6V6R/woDxAet7pvv8AvZP/AIill+AOvuABd6Zj1M0mf/QKP9deH/8AoLj94v7Fx/8Az6Z5suSSehPPXqPXFOB+YYU7s5wTXow+APiDC4vdNyP+msn/AMRUg+A3iDcGN5pnHbzZOf8Axyhca8P/APQXH7w/sXMP+fTPNJHbDAhiWOQCelVbjmTOMcCvVn+A/iCQ5N3pgOeMSydP++KrS/s/eInbIvdM6d5ZP/iK58RxlkM42jiov5mtPJ8dF3dJnlteifs8/wDJa/CH/X8v/oLVwF1btaXU1u5UvG5RivQkHBx+Vd/+zz/yWvwh/wBfy/8AoLV7PMp0+aOzRzxTU0n3P0noooryj2Qr4X/4Kqf8iV4A/wCwlcf+i1r7or4X/wCCqzbfBPgI/wDURnP/AJCWs6vws2p/Ej9GPhP/AMks8G/9gWz/APRCVZ+Itv4ivPAPiKDwjc29n4pksJ10y4uhmOO5KMI2bgjAbB5BHqCOK80/ZL+O3g740fB/w23hzVoJ9Q07Tba11DTWcC5tJUjVGDIedpKna2NrDoete21zrVFNWep+YX/C1v27PgzeSf234Vn8aW3OFOmw36Y9QbMq4+jflUmnf8FaPHXhF2tfHfwhWO6Q7XMNxNYMp945Y3Of+BCv05qGazguEZJYI5FYYYOoII981NmtmVzLqj4E0H/gsV8PrgD+2vA/iSwbv9ha3uR/488f8q7XT/8AgrD8D7wDzh4ksM9Rcaapx/3w7V9F61+z18L/ABHcNPqnw88MX87feluNJgd2+rFM1yeqfsT/AAM1jPnfDDw9Hnqba0EH/oG2iz7jvDsee23/AAVA/Z+nxv8AE9/b/wDXTSbk/wDoKGrQ/wCCmX7PO3P/AAmtxn0/se9z/wCiq0br/gnP+z3dNlvh/FGf+mWoXaD8hJiq3/Dtj9nrr/wg7fT+1Lr/AOOUai9zzM+b/gp5+z7Hnb4qvZf93Sbr+qCsW/8A+CrXwLs8+Vca/ff9cNMIz/32612EX/BOH9nqIj/igUY/7WpXZ/8AatdDpX7DPwH0fHk/DLQ5tv8Az9xNcf8AoZbNGoe55m9+zt+0x4P/AGnPDOoa34QF/HBp9z9luINRgEUqsVDKcKzKVIPGG7HIFfBX/BVS4lvv2jPhjpniKWW38FiyjYyLkKN9yVuWB/vBFj+gx61+lfgj4d+F/hppUmmeFPD+m+HNPklMz22mWqwI0hABYqoALYUDJ5wBXn/7TP7L/hP9qHwbFo3iJZLS+s3aXT9WtgpntHIw2M8FWwNyng4B4IBCnHmg43s31LpzUJqVrpHxx+058Nvhdpf7O+q3CaVo2mpb2itpF3ZxorvNgeWqOvL7u/JyCSfWvSv+Cat/rF7+yBbpqm/7Lba/dQ6aX7221GOPbzmnH4GvMPDn/BHiYa9EPEPxJN34eifIgsbJknkXP3cs5VCfUBvpX3hYeDNA+GPgLRfCPhyyj03SdNRYra1j/hUZyzHqzMzFmY8sxJOSTXi5PllTLKcqc6rndt3fQ9bN8yp5hKMoU1GytoUq/Ob4Sf8AKTrxl/111D/0AV+gHizxhongPQrrWfEGqWukaXbKWlubuQIoHpz1J6ADkngc1+bv7Mfjiw+JP/BQbXvE2lCT+zdS/tGa2Mg2sYymFYjtkAHHbNe+3rH1PCpp2k/I/SuiiiukyCiiigAooopiPyr8Xf8AI2a3/wBfs/8A6Masitfxd/yNmt/9fs//AKMasivZWx4b3NHRv9ZLgEnaOn1rc3kqCSAaxNFJE0mP7v8AWtZc4GOtfQYP+Gjy63xskDnZz70UiEbxx93oMUUq+6HR2Z5AQQQT34rpNC4sAPRmrm85wAa6TQv+PEf7zV5lX4TvjuaFeifs8/8AJa/CH/X8v/oLV53Xon7PP/Ja/CH/AF/L/wCgtXFP4WbQ+Jep9yfHb/kn83/XxH/M1zv7H3/JWJv+wZN/6HHXRfHb/kn83/XxH/M1zv7H3/JWJv8AsGTf+hx15UvgPYj/ABEfacn+rb/dNebV6TJ/q2/3TXm1fiHHXxYf/t79D7vI9qnyCiiivyo+qCiiigC5pH/IUtf+uq/zrv64DSP+Qpa/9dV/nXf1+zcD/wC61f8AEj47O/4sfQwLzw1Jc3UsomVQ7FsFTxUP/CJy/wDPdf8Avk10tFfpdz53mZzX/CJy/wDPdf8Avk0f8InL/wA91/75NdLRRcOZnNf8InL/AM91/wC+TR/wicv/AD3X/vk10tFFw5mc1/wicv8Az3X/AL5NH/CJy/8APdf++TXS0UXDmZzX/CJy/wDPdf8Avk0f8InL/wA91/75NdLRRcOZnNf8InL/AM91/wC+TR/wicv/AD3X/vk10tFFw5mYWn+HZLO8jmMysEJOADzxW7RRQJu5+bPxB/5H7xJ/2Ern/wBGvX1Tof8AyBdP/wCveP8A9BFfK3xB/wCR+8Sf9hK5/wDRr19U6H/yBdP/AOveP/0EV6svhR5kep+XHi7/AJGzW/8Ar9n/APRjVkVr+Lv+Rs1v/r9n/wDRjVkV6i2PHe5XveUUccnr6VS4UYAB5796t35+VABySf5VUBxwdox1r0qPwI5p/EOYvuwCNvHPTANLt25BHA45qMg4b+761IsmUzjjHSujckc+1DwODgZ7daWJk+ZSrDBxz0+tRKrlwAzDjJGKkJDybDneeKaIJCyyS/ICABtIohU/cGABncTioshVCtgA9yeaduQbhhhtAyT39KLldB2THhMYJOOOlSI5WbcAoIA6Co924AEkdqdEwc5ySenTg00SyZlG4OTkj7o7fjTS5OOevOB04pdxDHIA7mlMh25QDkEdMjiq1AaLgMQCATnABp+0IpILfMMsAcYqOJQWORyTmpF4AAOedp4pLzA9R/Z4IPjS9xnH9nvzn/ppHV74nfE7xL4f8c6lp2n6l9ntIFjKReRE33okY8shJ5J796ofs6n/AIrO955GnyZ4/wCmkdYnxnx/ws7ViQTjyT14x5KZr8Y+pYXH8bVqWLpxqRVJNKSulqu59f7arQyaEqUmnzbrfqPX4zeMGXH9sjLDg/ZYeP8AxykX4z+MTn/ib5A4z9lh6/8AfFcSo53KTk1YBAVgVJwOuO/vX6L/AKtZL/0B0/8AwCP+R8//AGjjf+f0v/Amdi3xl8Y+YANXwe/+iw//ABFC/GTxiTg6wDkjbi2h5/8AHK4rJQnnrycUgBOXAUIOFyf1o/1byX/oDp/+AR/yF/aON/5/S+9nbD4x+MA4B1ngnGDaw9f++KfJ8ZfFqLg6xh88kW0PT/viuKyz7QeSRTEgMrk4yVBPJ6Uf6tZL0wdP/wAAj/kH9o4z/n9L72dyvxi8YsT/AMTXrjA+zQ9PX7lO/wCFxeLQADq5yfW2h/8AiK4vhSoB6jP1prPjqMsxo/1ayT/oDp/+AR/yJ/tLGv8A5ey/8CZ2p+Mni4OV/tb5vT7LCR+iU9fjD4t6HVgAwyD9mh4/8crisCMKcHaAfm9DS/Lnb+GKr/VrJP8AoDp/+AR/yD+0sb/z+l97OxX4weLzknWOAOgtYeff7lOHxh8WgZGrk/W1h4/8crjdp6gcg4wO4oeRVyM/MenHWj/VnJP+gOn/AOAR/wAg/tLG/wDP6X/gTO0j+L3i7JP9rhgeQDbQ/wBEqGf4xeMI5CBrHHX/AI9Yf/iK5EAjByMEcgdQagmOWz6iuXEcOZNGF44Omtf5I/5G1PMcY3rVl97CeZ7maSWRt0kjFmOMZYnJPFegfs8/8lr8If8AX8v/AKC1ed16J+zz/wAlr8If9fy/+gtXU0owaW1hQd5JvufpPRRRXkntBXwv/wAFVRnwV4BHb+0bgf8AkNa+6K+F/wDgqp/yJXgD/sJXH/otazq/CzSn8SJ9Y/4Jj6/aaH4b8d/BbxrcaJrdxp9teNYXly8DxyPErMYbmP5gCSMIwPU5btVGH9qT9r/9mzZaeP8AwZL4r0qFgrXd9YGX5B2W5tiFz/tPuNfo/wDCf/klng3/ALAtn/6ISuqI3AggEHqDXLy9jXnf2tT8+vBf/BYXwbebovF/gXWtCmXgNp08d4p9chhGV/8AHq9o8Mf8FJPgD4khjZ/GT6PK3WHU7GaNl+rKrL+TGvZPF3wJ+HXj5pX8Q+BvD+ryyAhprrTomlP/AAPbuB985rxjxB/wTW/Z/wBeSXy/B02kySEnzdP1K4Qqf9lWdlH024o95BeD6Hp2h/tT/B/xIqnT/iV4ZmLdEfUoo3P/AAFiD+ldlZfETwrqSq1p4l0i5U9DDfRPn8mr411T/gkD8J7h2ax8TeLLPcThZLi3lA+n7lTj8a4/Uv8AgjZokjsdM+KOoWQP3ftGkJMR9SsyZou+w7Q7n6Hw6xYXAzFe20o/2Jlb+Rqb7Zb/APPeL/vsV+aE3/BHPVYj/o3xfDem/RXT+Vwaj/4dA+KF4X4voF/7B0o/9rU9ewuWPc/TCTU7OIZe7gQerSKP61nXXjXw9Ygm517TLfHXzbyNMfm1fnEn/BHfW5W/f/F1APVdId/5zitOx/4I1Wm4HUfizdXI7rDoYT9WuG/lS97sFo9z7g1n9on4XeHlY6j8Q/DNpt6q+qw7v++Q2TXm/iT/AIKFfAPw2r7/AB7a38i5+TT7eack+gKpj9a8U0z/AII8/DSEL/aHjHxPdn+I25t4QfpmNsV6D4Z/4Jc/AXw/EFu9E1TxA4x+81PU5A35ReWv6Uah7nc878bf8Fdvh9Y286eGvD2uavcLkR+ckdvG3vuLMyj/AICa8X1L9vD48/Ga4e3+G3w6aySb5ReJaSX8qMe/msFiUf7y4r9BfBX7LvwX8E7YNC8AeGo5oTkPLapdTKR6vLvb9a73xLbQ2ejRRQRRwRLIoVI1Cqo2ngAcCmk31JbildK/qfl3oX7DPxk+Pes2+tfGbxncWFoDu+xtOLm5UZGURF/dRZGeV3YPVTWB+zN4J034a/8ABQTWvC2jiQaZpaX1tB5z7nKrGOWPcnqeg9hX6cV+cvwk/wCUnXjL/rrqH/oAp8qi1buOM3JO/Y/Qqiiius5wooooAKKKKYj8q/F3/I2a3/1+z/8AoxqyK1/F3/I2a3/1+z/+jGrIr2VseG9zR0f/AFkg/wBkVsIwB6nJ6cfnWNpBAkfPcCtVnBGBwM9Qa9/Cfw0eXX+NkiEBiTnAzz60VHuwcDkYJ47UUq26HR2Z5LwSG6Y610uh7fsAx03NXNAEng9OldLoX/HiP95q86r8J3x3NCvRP2ef+S1+EP8Ar+X/ANBavO69E/Z5/wCS1+EP+v5f/QWrhn8LNofEvU+5Pjt/yT+b/r4j/ma539j7/krE3/YMm/8AQ466L47f8k/m/wCviP8Ama539j7/AJKxN/2DJv8A0OOvJl/DZ7Ef4iPtOT/Vt/umvNq9KYblIz1GK5v/AIQ0f8/X/kP/AOvX5TxZlWLzJ0nhYXte+u2x9jlOKpYZT9rK17WOZorpv+ENH/P1/wCQ/wD69H/CGj/n6/8AIf8A9evz/wD1Vzb/AJ9fiv8AM97+1MJ/OczRXTf8IaP+fr/yH/8AXo/4Q0f8/X/kP/69H+qubf8APr8V/mH9qYT+cxdH/wCQpbf9dV/nXf1gWnhQWt1FN9pLeWwbGzGcfjW/X6ZwrluKy2hUhio2bd1r0Pm80xNPEVIum7pI5LUdWu4b+dEnZVVyAPQVX/tu9/5+G/Surk0u0ldneBGZjkkjqab/AGPZf8+0f5V9zc8fmRy39t3v/Pw36Uf23e/8/DfpXU/2PZf8+0f5Uf2PZf8APtH+VF0PmRy39t3v/Pw36Uf23e/8/DfpXU/2PZf8+0f5Uf2PZf8APtH+VF0HMjlv7bvf+fhv0o/tu9/5+G/Sup/sey/59o/yo/sey/59o/youg5kc7Yaxdy3tujTsVaRVI45Ga66qqaVaRurpAispBBA6GrVJkNhRRRSEFFFFMD82fiD/wAj94k/7CVz/wCjXr6p0P8A5Aun/wDXvH/6CK+VviD/AMj94k/7CVz/AOjXr6p0P/kC6f8A9e8f/oIr1ZfCjzY9T8uPF3/I2a3/ANfs/wD6Masitfxd/wAjZrf/AF+z/wDoxqyK9RbHjvcrXwBjB7jP8qoNKAFGACOmavagAY1B6Z7Vc8N+DdW8YTi30qwkuSp+aQLtRP8AfY8D1x19K6ZYmjhKHtq81GK6t2M1TnVnyQi230WplDaoOzgsOuOaMBQCzAsOcV23jz4fQ/DnS7KK9vlvddu23iCAYjijHUkn5mJOADwOG4NcQQxI4wxOSM0YDMKGZ0VicM7wbsn0fmvIVehPDz5Kis1uu3qPWXyyWPBOCMdfpTmlJUMQNx5xjGaaAY97Yy+PummKzAqcAEHkDnFeldo57DlYynOcheoI70qh2BOVKj7xx2pVBeMMByc9D1qQEKBxjGOKEguMDCPCZzn9acrhGCIME9/501SWfIOM/wA6EJLbsZYe+aYiRyducszHk5pyEIpJHTDfU0zcTyARuODkcYqUKVRTkk5OAaEDGsxUY53hs57YpyO6o+NpXOcL2+tMkySAOCaeyBBlCNzHLAdOPWjqB6j+zox/4TS+BA4098Edx5kdYvxnJHxO1j5MqRD35J8lK2f2cznxpekgbv7Pk/EeZFWJ8bVc/ErViO5h/EeSnFfk2F/5Lmv/ANeV+cT6ur/yJIf4/wDM45ZDxxnHXPpSFy7Y568AdK9i/Z38K634mt/Hx0fSPCWqrY6DLcXJ8VWzzNBEOslqVB2zDszYFWvB/wCyj4g8QeFdC17U/FXhDwXa63k6bD4m1U2s94gIG+NAjZUkjkkdR6iv1GWIhGTjN2sfNKjKSUoq9zxhWLY4A/DtSJ87srELu6YPAr23wP8As667pP7TXh74b+KLWESi+jmujEwlhltVHmswburKrDkDrgjPFen+OtP8IeJP2cfjF4s07wtomnSw+OhZ6bcWlhDG9vaqsCKkbKgKKwyxVSAS7HnJJmeKjFxS1vbX1ZUcPKSb2sfI30Ygr7cGkWUbCMZdiNy5/nX0j+wX4V0Pxj8aNSsNd0iw1yzj0C6mW31K2S4jWRWjw4VwRuGTg4yMmm/s3fGWzv8AxZ4J8Cah8NPh7qtjdXkVnNqeoaAs9/KrPyzSs+C2DgEqeg4oqYhxclGN7JPfuKFBNRcna7sfOhwewAHGBzimREtOWYAqpOPSvqn9qT4taf4S+IfjnwBpXww+Hen6bA5sodTh8PpHfRBo1O9ZVcBXBY4YLxgcV6t+2N8GPCSfAvT9X8LeHdJ0XWNBSxu9RbTbCOB57e4jKBn8tQTh1zls4w3rmsljHeHPG3Ntr9xp9V0lyyvy7nwLECFYEDJJPB4xSFT8z/K3bBr7m/YL+DXhnVvA+r+I/Fvh/Tdem1i4lstKi1S0iuESOCJ3klRXVsEsQpIwflH4+UHwlof/AAwaPEB0XTzr/wDwk/kHVPsqfavLx9zzcbtv+znFV9cjzumls0r+pP1WXIpt7pv7j5wJOFLfdz06cUqj5t2BuB/iPamqRuwGzgcjPFAfdxt+YnjIzXpHGWWQSEgcEHr71SkBVsE5qxI+1fmwPUCoZiWfPtXLivg+ZpR+Ijr0T9nn/ktfhD/r+X/0Fq87r0T9nn/ktfhD/r+X/wBBavHn8LO+HxL1P0nooorxz2gr4X/4Kqf8iV4A/wCwlcf+i1r7or4X/wCCqn/IleAP+wlcf+i1rOr8LNafxH6MfCf/AJJZ4N/7Atn/AOiErU8W+KtL8DeF9V8Q63dLZaRpdrJeXVwVZtkaKWZtqgljgHgAk9AM1l/CZgfhX4NIOQdFsv8A0Qlb+r6RZa/pV5pmpWkN/p95C0Fxa3CB45o2Uqysp4ZSCQQexrnK6nz54P8A+Ch3wE8ZSxww+O7fS55DjZq9vLaKv1eRAg/76r2jw18UPBvjMgaB4r0XWyRkDT9Qinz/AN8sa+dfG3/BMP4EeLmL2mhX/hmZiSZNHv5FB/4DLvUfgorxfxT/AMEb9AnuS/hn4jalp0WciPVLGO5Yf8CRo/8A0Gp95F2i+p+i+9T0YH8adX5hXH/BMf43+EWx4O+Mq7F6f6XeWB/KNn/nTof2T/219A4sfi1Pdqv3Q/im6kH4CVCKOZ9h8sejP07or8x2+Ff7fGmjbD4ue6A9NTtpM/8Afa1ueAfCP7eFt430J9W1yL+yVvImvDfXFlJAYQ43hlRd5BXI+X5vQg80c3kLk80fo5RXM/Eiy8Saj8PfEVr4RvYNO8VTafMmm3dwN0cVyUIjZgQRgNjqCB6Hofztm/Zl/bg8QEm7+KU1juPPl+JJocfhEgxTb8iVFdz9OK/P/wD4KZftFeLvDeteF/hF4FvptM1PxDEs17dWrmOZ0kkMUMKOD8oZg24jBwFGcEg/Qv7Hfwp+KHwl8A6lp3xT8ZyeMdXuL4zW0kl5NeG3i2gFfOlAZssCdvQduprw3/gpJ+y74t+Il54b+J/w+tZr/wASeHkWG5srVd07xK/mRSxL/EyMWyoBJDcdMGZ8zg+TfoaUuRVFz7HyR46/ZD8ZfAXwovjrQ/Hsj+ItKUXc6WfmQSRgfeaOTdltvP3gMjP0r9Cv2Sfj5f8A7RX7Nuk+ItYKPr9levpepSRoEWWeJAwcAcAsjxsQMDJbAA4r87/Efxs+NPx40p/AWn+AbpdZv/8AR76S1spfMcZAYYb5Yl7MScAZ5Ffo5+y/8AJv2b/2ddH8K38sc2tz3TalqbwnKC4lUAqp7hFVFz3KE8ZxXiZLHMI05LMGnK7tbt8j285+oc0fqKdra+p6HX5y/CT/AJSdeMv+uuof+gCv0ar85PhHz/wU48ZY5/e6h/6AK+hlvH1Pnae0vQ/QuiiiugzCiiigAooopiPyr8Xf8jZrf/X7P/6Masitfxd/yNmt/wDX7P8A+jGrIr2VseG9zQ0jHmSZP8IrQ2Fhj88Vm6Wdsjk+gFaiMSwI4z1Ne7hf4SPNrfGx2dxBAHAwaKazZ3DPOODRTrbouhszygHBHJUV0mhEGw4bPzNzXNEkgnPOOlbnhqbMc0R4IIYe+Rj+grzqmsTrjubVd58CL6PT/jJ4PllIVDqUUeT2LNsH6sK4OprS7lsLuG5t3aKeF1kjdeqspyCPoQK4pK6aNovlaZ+kfx2BPgCbA6XEf8zXDfsx+K9I8HfEiW/1q/j06zawkiE02dpYshC8D0B/Ku+8M61pnxy+FdpehwseoQATLGfmguF+8uP9lhxnquD0NeQ6p8FvFGn3bxRWS3sQJ2zQuuGHrgkEfSvKsmnFnrXd1KJ9m/8AC/Ph7/0NVj+bf/E0f8L8+Hv/AENVj+bf/E18T/8ACpfFn/QIk/76T/Gj/hUviz/oESf99J/jWP1eHc19tPsfbH/C/Ph7/wBDVY/m3/xNH/C/Ph7/ANDVY/m3/wATXxP/AMKl8Wf9AiT/AL6T/Gj/AIVL4s/6BEn/AH0n+NHsIdw9tPsfbH/C/Ph7/wBDVY/m3/xNH/C/Ph7/ANDVY/m3/wATXxP/AMKl8Wf9AiT/AL6T/Gj/AIVL4s/6BEn/AH0n+NHsIdw9tPsfbH/C/Ph7/wBDVY/m3/xNH/C/Ph7/ANDVY/m3/wATXxP/AMKl8Wf9AiT/AL6T/Gj/AIVL4s/6BEn/AH0n+NH1eHcPbT7H2x/wvz4e/wDQ1WP5t/8AE0f8L8+Hv/Q1WP5t/wDE18T/APCpfFn/AECJP++k/wAaP+FS+LP+gRJ/30n+NH1eHcPbS7H2x/wvz4e/9DVY/m3/AMTR/wAL8+Hv/Q1WP5t/8TXxP/wqXxZ/0CJP++k/xo/4VL4s/wCgRJ/30n+NH1eHcPbS7H2x/wAL8+Hv/Q1WP5t/8TR/wvz4e/8AQ1WP5t/8TXxP/wAKl8Wf9AiT/vpP8aP+FS+LP+gRJ/30n+NH1eHcPbS7H2x/wvz4e/8AQ1WP5t/8TR/wvz4e/wDQ1WP5t/8AE18T/wDCpfFn/QIk/wC+k/xo/wCFS+LP+gRJ/wB9J/jR9Xh3D20ux9sf8L8+Hv8A0NVj+bf/ABNH/C/Ph7/0NVj+bf8AxNfE/wDwqXxZ/wBAiT/vpP8AGj/hUviz/oESf99J/jR9Xh3D20+x9sf8L8+Hv/Q1WP5t/wDE0f8AC/Ph7/0NVj+bf/E18T/8Kl8Wf9AiT/vpP8aP+FS+LP8AoESf99J/jR9Xh3D20+x9sf8AC/Ph7/0NVj+bf/E0f8L8+Hv/AENVj+bf/E18T/8ACpfFn/QIk/76T/Gj/hUviz/oESf99J/jR7CHcPbT7GT40vIdR8Y67dW0izW89/cSxSL0ZWkYqR9QRX1dopCaJYliFC28eSeMfKK8J8G/A/V73VIZdZiFjYRsGdCwLyY/hAHTPcmuz/aM+IcHw3+FepvHIseo38bWFjGpG7e6kMwHoq7mz0yFHet7czUUY35U5M/PTX7xNQ17UrqPmOe5klX6MxI/nVCiivUPI6ns3w7+Dvh3WrWHUb3UotcKkMba2YrGjejZwxPsQv0NezWVjb6bbR21pBFbW8YwsUShEUewHFfGEer3ujX8VzYXctlcxjKyQsVY+3HUeo6V6z4L/aMuIdtt4jtftCLgfbbVQrj3ZOh+q4+hr8K4v4Pz/HN4mlWdeG/Js16LZ/mfb5Tm2Bor2coKEu/c77xb8GNI8Z61JqmoX+oid1VVWKSMKiqMBVBQkDqevUmsRv2afDLMD9u1UEHORLF/8brgPjXaaZq9zbeKdCuoryxvisVz5J5jmUfKWXqpZR0IH3Ce9eYb2iyFOSecbjXr5Dw7nWKy6lOjmUqaStyOFnFrpv0OTHY/B068lPDKTbvfm3R9If8ADN3hsZxf6oM9f3kXP/jlB/Zu8OkAf2hqnBznzI//AIivm4OcMSzBuCvPU1Ipb5SXbJ689K9//VbiH/obS/8AAP8A7Y4P7TwH/QIvv/4B9GH9m3w8QB/aWq4XoPMj/wDiKD+zf4ebk6jqh4x/rI//AIivnVZXVuXYBuAdx60fapQ43uxA4zuPaj/VbiH/AKG0v/AP+CH9p4D/AKBV9/8AwD6LP7OHh8nP9o6l/wB9R/8AxFNX9m7w8u7Goaj83X5o/wD4ivnmSWTqJWAbnBY/pT4Z5SCRK5xjOWP40/8AVbiL/obv/wAA/wCCL+08B/0CL7z6FH7OOgAYGo6j/wB9x/8AxFH/AAznoZxnU9SPploz/wCyV89vcTkkiVwMcfMaFubhwo8+Tg45Yij/AFX4j6Zu/wDwD/gh/aeX/wDQKvvPoX/hnXQ+n9pah/30n/xNNb9nPQ2TYdS1DH+8n/xNfPpvJ41UmeTB/wBo/wCNSLdzKS5uJD0x854p/wCrHEn/AENn/wCAf8EP7Sy9f8wq+8+oPA/wo07wJqkl9Z3d1O8kLQFJtu0KWVs8Ac5UfnXiXxqkP/CzNYB3Mi+Tx6ZhTmuON7cqFbzpMHtvOahlLSfO0jSbsZOeT9a9DIeFsblmaTzTG4r20pR5fhs7XXn5HPjczo4nDRw1GlyJO+59Q/sQqP7O+M5Awf8AhELjvXovh9vEQ+DPw8sPHPwl0/41+E7qz2aRqnhr7R/aGmwkjMMjrGCrLwAFKg7MFiRmvFvgn8Z/A3wh+FXjKOKy16+8feI9Mm0s+YkI0yGNmIVgwfzC2MEgjBIwMda8p8N/FTxr4Jsn03QPGOu6BbSN5httM1Oa3iLHqSsbgZ98V9nOhOrUm7WV9L6dLbnnxrRpwir306H2lqXwn8PfsyeKvix4t8MyTQWOk+FIorC3vZRI1nf3zMghJBySqqp5OcSdT1rjPgP8TPEfws/Yd8deIvC2pf2VrMHiuKJbhYY5cK0VuGG2RWU5HtXyvceN/EN5p+oWVzrupT2V/cC5u7aW8keK5nHSWRS21mH95sn3qKPxVrNj4Zn0KHV7+PQ7m4E82lxXLrbSSDaA7RA7SwCrhiM8D0rRYSTjapK+q37IzeJSleMbaO1u7Pub9jD9p74l/F34ranoXi3xM2r6Wuh3NysBsbaECRTGA26ONW4DNxnHNfKn7NMoHx88AbActrNuG+m4YrhfDfizXPBuoNfaBq+oaHeSRGI3GnXL28hjOMqWQglSQMjODgVT07Wb3Qr2G/srm4sr6B1lhurWVo5ImB+VlZSCGB6EGto4ZQc+SyTSWiMniHNR5rtp31PYv2y8j9pzx/gAg369/wDpklfYvibWLXUP2iLP4dam+NL8cfDm300q2f8Aj4X7Q8TA9iBv/Eivze1fXNQ8S6pPqOr391qeo3B3S3d5K0s0pwBuZ2JJOABkmtO48feJ9S1yy1i58R6vca1YIqWuozX0rT26JkqsbltyhcnAUgDJxUTwjnCMb/CrfPTUuOIUJSlbd3+R+hfwvkh8F/H/AMH/AApspE+z+D/Bl19t8t+GvpxHJMWUcZA2nP8Atn8fF/D/AIV1jxp/wT9Gn6BpF9rl7/wlJk+y6favcSbVHLbUBOBnrivlyz+Iniuz8TXPiGHxPrEGt3AKz6pFfzJdTKQAQ8obcwIABBJzgVf8N/Ffxt4P00af4e8Y6/oViXL/AGTTdUngjDN1bajhcnHJ68Vl9TnHWLV9H6tb/mW8VGWkk7ar5Mj8Q/C/xj4JsRf+IPCOvaFZFxGtzqOmTW8W45woZ0AJIB468VzfmbXAJ4HUkV1XiX4p+M/GGntYeIfGGva9Zq4kFtqWqTXMW4A4YK7suQCecZ5rkw6qMsMAda9SnzqP7y1/I8+XJf3L28yRhGzbV4LYLGopMbuPSnoQEDABju457UyQ7jnp7VliFencqk7SGV3nwIvo9P8AjJ4PllIVDqUUeT2LNsH6sK4OprS7lsLuG5t3aKeF1kjdeqspyCPoQK8iSumjvi7NH6y0Vy/wy8d2nxK8D6Vr9oy/6VEPOiU/6qVeHQ+mGBxnqMHvXUV4zTTsz2k01dBXjH7Vv7PUX7Rfw1bRYrpLDW7GX7Xp1zKD5ayBSCr45CsDgkAkcHBxg+z0VLSkrMtNxd0fn58H/wBuH4o/sdmy+Hnxd8I3er6BYjyLK5yEuoYVwAIpDmO4jUcAZBAON2AAPtv4Z/t6/BD4owL9j8bWeiXhCl7LXz9hkUn+Hc+I2P8Aus1aXiXwnovjLS5dN13SrPWLCQYa3vYVlRvwYH86+bPH3/BOH4U+LZJp9Ij1Dwpdvkj+z5vMhB9fLfPHsrLXO6co/C7m3PGW6sfftlf22pW6XFpcRXUEihllhcOjKeQQRwQanr8oJv8Agnf8T/h/fLd/Dr4q/ZnUH5nmuLCRfYGIuD+OKsQWn7c3w5uDHY69deIraPgSNe2l4jD6TYf9Aan3lug5YvZn6r0V+Wn/AA2F+2T4PbZqvw6bVAvWT/hHJpV/FoW21ah/4KefG/RsDW/g9CzL94raXlt/MNilfuPkfQ/UGivzEb/gsD4ntOLz4RQxEdc6lKn84aT/AIfLaiOD8K4A3p/bDf8AxmlzIORn6eUV+Yg/4LFa9c8W3wmgkb/sKu38oabL/wAFUfixrH/IG+DsIz0z9quP/QUWjmDkZ+ntFflo37dH7W3idtmkfCwWgf7skXhy7dR9Wdiv51Tv/FX7dXxCATN1odvJ1FubKx2g98lg4/PNO/ZMOS27R+pl5qVppyM88yRcZOTz+VeE/Fr9rH4YeBVmj13xnpVmbVsPZQzi4ud3YGKLcwP1HHevhmT9h34+fEyaFvHfxQQWrHdJHLqNxdun0TCoT9GAr0HwP/wTB8A6O6z+Jte1bxNMGBMce21iPsQNzH/voVaUuiJfL1l9xznxU/4KVXnia6Phv4N+F77UNWu2MMOpXkBkkJIxmG3XcWPcFzxjlTXX/sV/sr+Ivh/rmpfEr4hzM/jHV1fZayPvkgWVt0kkzDgyOewztGcnJIX6O+Hvwe8F/CuzNv4V8N2GiqwAaS3hHmvjpvc5ZvxJrsa0jB3vJkOSS5YqwUUUVqZhRRRQAUjMFBJIAAySewpa8v8A2jviLF8OvhbqkyyhNS1BGsbJAfm3upDMB/srubPTIUd6qMXJqxMpJJtn556/eJqGvaldR8xz3Mkq/RmJH86oUUV7J4he0vh5D3AFaKyAEDGdwqlpa7UZiOp6/T/9dXIyGw2DwDXvYdONNHmVnebHMgYdc445opuflyeD7UVGI3RpQ2Z5UDjj+dW9KvPsl6jH7jfKx9jVRgM9aQAqoORz2rk3Vjq0R3FFZWh6gLmIQOf3iD5efvL/AIitWuGUXF2ZoerfAP47Xvwa1yRZY3vvD96y/a7NW+ZWHAljzwGA4IPDDg4wCPvjwd440Px9o8ep6DqMOoWrgbvLb542P8LqfmVvYgV+WNaWg+I9V8L363uj6ldaZdr0mtJmjbHoSpGR7HiuWpRU9Vozpp13T0eqP1Zor8+NM/a4+JemwCN9ZgvgowGurOItj6qqk/jmr3/DZHxH/wCfrTv/AACX/Guf6vLyOv6zDzPvmivgb/hsj4j/APP1p3/gEv8AjR/w2R8R/wDn607/AMAl/wAaX1eYfWYeZ980V8Df8NkfEf8A5+tO/wDAJf8AGj/hsj4j/wDP1p3/AIBL/jT+rzD6xDzPvmivgb/hsj4j/wDP1p3/AIBL/jR/w2R8R/8An607/wAAl/xpfV5h9Yh5n3zRXwN/w2R8R/8An607/wAAl/xo/wCGyPiP/wA/Wnf+AS/40/q8w+sQ8z75or4G/wCGyPiP/wA/Wnf+AS/40f8ADZHxH/5+tO/8Al/xpfV5h9Yh5n3zRXwN/wANkfEf/n607/wCX/Gj/hsj4j/8/Wnf+AS/40/q8w+sQ8z75or4G/4bI+I//P1p3/gEv+NH/DZHxH/5+tO/8Al/xo+rzD6xDzPvmivgb/hsj4j/APP1p3/gEv8AjR/w2R8R/wDn607/AMAl/wAaX1eYfWIeZ980V8Df8NkfEf8A5+tO/wDAJf8AGj/hsj4j/wDP1p3/AIBL/jR9XmH1iHmffNFfA3/DZHxH/wCfrTv/AACX/GsvWv2rviXrMDRf26thGww32K1jjb8G2lh+BFCw8vITxMPM+3/iN8VPDnwt0hr7Xb5InKkw2cZDXE59FXOTz1Y4UdyK/Pz4w/FrU/i/4rfVb4fZ7SIGKzslbckEec4z3Y9WbufQAAcfqWqXus3kl3qF3cX13IcvPcytI7H3ZiSaq11U6Shr1OSpWdTTZBQxCgk8AcmiqOoXRUiNP+BH+ldUIuUrHNJ2VyrcTEys/JJbgemKQMd2QOaQsPlBGB3pS4VSQVGTjmvTStojmFWTA2knJ44OARTQdvJGTn05pC2xRkA89RT9mATndkcc07AOUsFGcYHfilc8YzuPXOeKjAJGXJIHXFPbAT5OgGD607gKy5IOAe/Hak6oAQVbPc0Bic5BUeq0BiUyM56ZPegCZVC4JGSB+VIsh2MMNljwcVCoZUyZO+OoqVSDj5yQp5IpoCTP0LA8c0oUNu3E5ByMU3hyWB78Gk2uEHBXcSMgZqiBVJfCgHC+tO8o7sbhtyOPWkTBABz6g+tKGKgAld4OcDtQBJvBTcAQR6/lSFHZUQEYByT71CCdwUbioPOfWpWYkEKASCec5pXuOxImVwuc4PUCnLGGkbBXcDyDiol3BVJOSRnjoKcjEliQQDkLx3FUiWOJKA5PGOc9KVAV38ADPHP9KTJwOhHUj3qNiRyctg5yv9aeoFjJOCw4XgYoVFcnuqj179qarmQDBHPUUnI3IOg5470EjkccfOdwPQinM4BIGSQTniowONwYKCe55zThKY2bIyxBxRfuOwbREA4HUjNPSQ7nPQ45z/OkVjsG/cCeRnBpUwjZZz83H4UIQnmZTA6ZzkVMv3VDDd9ajWMKxJAIzxtNOUttwoBGO56015huO8sbtx52jtTf9YGfGcdx6UnIJXDHK5wD3p6A8gEZPIA44pSipKzBNxdyKinug6g7h7dqZXjTi4SszvjJSV0erfAP47Xvwa1yRZY3vvD96y/a7NW+ZWHAljzwGA4IPDDg4wCPvjwd440Px9o8ep6DqMOoWrgbvLb542P8LqfmVvYgV+WNaWg+I9V8L363uj6ldaZdr0mtJmjbHoSpGR7HiuOpRU9Vozqp13T0eqP1Zor8+NM/a4+JemwCN9ZgvgowGurOItj6qqk/jmr3/DZHxH/5+tO/8Al/xrn+ry8jr+sw8z75or4G/wCGyPiP/wA/Wnf+AS/40f8ADZHxH/5+tO/8Al/xpfV5h9Zh5n3zRXwN/wANkfEf/n607/wCX/Gj/hsj4j/8/Wnf+AS/40fV5h9Zh5n3zRXwN/w2R8R/+frTv/AJf8aP+GyPiP8A8/Wnf+AS/wCNH1eY/rMPM++aK+Bv+GyPiP8A8/Wnf+AS/wCNH/DZHxH/AOfrTv8AwCX/ABo+rzD6zDzPvmivgb/hsj4j/wDP1p3/AIBL/jR/w2R8R/8An607/wAAl/xo+rzD6zDzPvmivgb/AIbI+I//AD9ad/4BL/jR/wANkfEf/n607/wCX/Gj6vMX1mHmffNFfA3/AA2R8R/+frTv/AJf8aP+GyPiP/z9ad/4BL/jT+rzD6zDzPvmivgb/hsj4j/8/Wnf+AS/40f8NkfEf/n607/wCX/Gl9XmH1mHmffNFfA3/DZHxH/5+tO/8Al/xo/4bI+I/wDz9ad/4BL/AI0fV5h9Zh5n3zRXwN/w2R8R/wDn607/AMAl/wAay9a/au+JeswNF/bq2EbDDfYrWONvwbaWH4EULDy8hPEw8z7f+I3xU8OfC3SGvtdvkicqTDZxkNcTn0Vc5PPVjhR3Ir8/PjD8WtT+L/it9Vvh9ntIgYrOyVtyQR5zjPdj1Zu59AABx+pape6zeSXeoXdxfXchy89zK0jsfdmJJqrXVTpKGvU5KlZ1NNkFKASQAMk0lXrC3/5auMAD5a7KdN1JJI5JyUFdlqNBFCiHjb396fkc4PJpGbcvseR7URnOVOBgda+hSUUkjy276sOcgZopB9cnPH0orkrbnZQ2Z5b25POetPzketNPI2nJxQPQdK5DoY+J2Rw6MQy8giuj07WEuQqSkJLjqeA3/wBeuaIIxjtQSegB49amUVJagnZnb0Vy9nrU9qApPmqP4W7fQ1qW/iG3k++GjP5iuaVNlXNSiqi6taMMidfxBFOOp2oGTOvNRZ9hlmiq39p2vXz1/OkGqWhGROh/GlZ9gLVFVv7TtScCdaP7Stv+ey/nTs+waFmiq39pW3/PZaP7Ttc489Pzo5X2C5Zoqv8A2lbdPOWk/tK1/wCe60WfYNCzRVb+0bbp5y8e9H9pWv8Az2XNHK+wXLNFVv7Sts489cj3pwv7c8iVaLPsGhPRUBv7cDPmrg0h1G2BwZlzRZ9g0LFFQG+gHBlWj7dBjPmrzT5X2C6J6KgN/bjgyrQt9Af+Wq0rPsGhPRUIvYDyJRTWv7deso59ATRyvsGhYoqlLqkaKCoZsnHpVKe/efAJ2KeorSNKUt9CXJF+5vQgwmCf72ePwrN3FSQSSzHg9qQkMBkE4GMk/epoyyjbng/hXbGKgrIxbb3JTIdoHGR1GO1GC3bIHao3BUd8bsnvzTlIMgK7iO4xjmqDUC+07SCRnFSqp2EgkDsO9RMpMnI3cY56VIpTaee+CKaJG5KqSxAA7U9XV1JAHPbNIz5UEn5jwD601AEycnI5xnOaewDwoQkNklvT1pUiO3HTrwPWkDFgBkN3wDSK5iIGCBknGaNAHgbFyADzzUrqVw2BjjoOtREBeflIzyh4p2QVO1jnH6UwH7i8JAOwgcgjnrSw4MYJIIyQMHmo2wQDyCaUEOw5GF9sY9aAHOzb+MkdacyFsg446EelM8sAkqxYAevvTkYRpnaWOeRnigXoSMRgKQT2/Gm7irLyd2MA9AaN5L5KkdsUsigAEnC8lSODiq9CRQTJ0HzAYOe9OUHC59Bxnoe9QpI2GAGBnjmpomClSBksMcjr+HamncB2ApAyMY64pEcbyGVlVc4IokB4IBYk4x6UvEikEMvPf1+tVqAoc44ft0p6na7euOSPSmbMS7TkqBxjnFAQMysDjt+FCIFGAQTg9xzTyxOSFBGc+4FRkjccLxjqT3p6lwARjAPcZFACux6pyDxn60MBIMAjPXryaQgNz1Oe3FG0gg5J3UAL8jDIO3HBFPVgAcNnimHKlQMnB5BprERrv+9n09Kew9yYM3HJ+b0p+4dSDx3qKJmZOCACcAMKEySST36etFxWJM7iCTncOB0prKQehp7TFSAoLEDJGOlJ5x2lTuP0rOpTVRWe5UZOOwyinqFLMCcYxk/WkKHbkdBXnSw847K51RqJjaKXax4waMEn3rHkl2ZfNHuJRTthzjBpMc4o5JdmHMu4lFO8s+lHltz8vSjkl2Ycy7jaKcY2HUHmjy29KOSXZi5l3G0U4RuegPrRsPp0o5JdmPmXcbRTvLY9jS+U45waOSXZhzLuMop4hdsYUnNBifuDRyS7MOZdxlFPMTjAKnJo8l+u00ckuzDmXcZRT/JcdVNHkvuxtOaOSXZi5l3GUU8Quf4TS+TJ/dNHs5dmHNHuR0U/yX/uml+zyMuQvH1p+zn/ACv7g5l3I6KnW0ckZ4FWIoERgQCT71vDDTk9dEZyrRjtqQwWufmccf3e9XQQoOT0GKQZLEHPFNBGWByRz2r1KdONNWRxSk5u7HqQ2AD0p6sCc+nBqPeuMgdelNjYh+Gwfp2rYixIpLcc+lFISFYknkHOaK5a251UdmeYDjHPWnFjuxgHFJ94kEDHrSAYOCOOhNch1DuTSs25eR0NNzjgDkUuSBg4GeaCAJHQYFGB2GOO1AUN1pWyD0HHGaCw5OeetOB4wOlNAG484p3KoOOtNEC55wR14pgyeAOhPFKrBj9P1oGASR0PWkA9STyQOKCDkHPtSZ5znAHFC46kjgZ4oAeGI5Hp2o5444wKZ6gfWnqwwBkigAGDjAPFBI7DApB8mCc80uOeO9ACjJ6en50N0xyOetC4RhknmjnIAPy56mgBdvqc/rUis3zEEZ7c0wgEnHAPQUHhSQOPzqgFI3fe+tOXBPqSfypoBIXHTHNOABLE5BA7UwHyghgOcY6mhCSOOgHrTXkyQc5IAHSmhiGBAx6igXQcSW57jinb2GeOD1puXGTxgdB2perA/wAPcZoGOV2C5IIxyAGpUbBIA5PWkAMYY7RjqPpQPm59OTnvT1IHFvLc44GOnuaXAySTjNRMpZ22jA654/LNSkDcAST6YouA5c54PI6f/WpwY7suMDHHFQgbsFiRgZH0qTO5GJ/OmmAuCMkHHsOn4UJgnJPHcgUgY44zgDvS7lMYwec8+1CAkVCQDuwTj6U5fuEM+TnrUaKu3I5wc9cZo424yCfQjpQKwpRD90kgDkVIW+QgfKwxjnqKYWCuwJI7DHpTnZdgJG79KaHqJnbjByQMc0RDPLdsj8KZG20jI4zxnmpFJkO4HgZyOxoQthwUuCBnB+8e4pUkCJgA8enGabuO5WJIHXCsRUrMSGbJY4zn+lV5iYuATnj5v508tgbTkHHHGMioQSdp5AHQA0ocMcs2COAO1CYrD/cgAd6Xb0I9sUEiR1JzhfTvTJATISFJU4OAaBk0bFY8kAZ4Y45NI04ZFLoMLx060hcZwTkDoCe/vQzYIAAZVHfGeetPoKw7PAIO0A9qfhCwyTkdBVcMwLjbgkjGeakT55cNwwbG4e3tTTFYcFUtx9SScYp6kbcYXI6GowArcjI5FOJMag4ONuT36U1oJj87SemaVQ/yknnkYpm5shgA2cHntUjSkBj0A44HOaoBowoBIGcnt1p6ktwDgHggjNMJKxrkA/XrRGxGOAAPfrSuAowSSgX5TipCQGB+8TycDoahXLAjcoycYAqSPCMSWADcAChCZKeDnLBuoHrTEGw8YweuRQZmbJJLHoPcU3dgYIO3rkdqYkiRCMscdM9advyo4yVPWoo5y3Ygk9T/AIU4SH5VIwOc80XuBIXIOOmQelNXg5IwT0NBYjJJB45FA+ZhkHjkAGgQrKVYuSM4yQOOKXeQwJ6dqfgTKFZQX74PaogB8wBPtmqegEisx4J+XOce9OZssxJwOnAyaZIzBcHBPGMUDKZJP4YpisKdwYEMR2FOwdpJAz71GWGcAY4znmnKzMMHkKOCOtJD1Hs3PPLDP14pd5bA5yR09aaWxjOW5JPtR1JI6455ouGpIGAcB/Q0ZJOCcZqPGF3uDlT9fwp6uCuTxkcUyRxJ3YBOMfnQjCPtznNJ5gK45z2//XTgQGBO05/HFAAMg5ByfShWDEjBX15pjn5s8jHPFKuP4SQMcg+tADyx2jA46U9lPAzimkEscEDoKU5VSxIB7j1qgGkA4zyKcRlht4TGD7UzdghieMdKczhskDAPGAKWhOom44wecmnryvA5UYGaYMDgk5xwKCx3Z4GOM4qiiTAUYJ6UwSdOORTpPvDOeaQhQT0APIPpQQO5Ax0Jp4zkHIBHY0wjcQcjFAwuBnJxzzQAFi3OBkH8KeGJUdM9j7U1SApAB4PegOQCAQQOgxVagOJI4BOe/tSM/J4x2oU8YJ696GwuSe4x9KNQFOOCBkUhcbulAQbcA800443fpRqCHZwOCeTRTCQq8dveiuGvujsobM882j0p3lrRRXw3t6v8z+8+w9lT/lX3CbR6UmwHtRRS9vV/nf3sn2VP+VfcLtFGBRRR7er/ADv72Hsqf8q+4No9KNoooo9vV/nf3sPZU/5V9wgjUN0pdooorT2tT+Z/eV7Kn/KvuFoX5elFFZ+3q/zP7yfZU/5V9wbjtoooo9vV/nf3sPZU/wCVfcGaVmNFFHt6v87+9h7Kn/KvuEzS7j60UUe3q/zv72Hsqf8AKvuEaRqNx9aKKPb1f5397D2VP+VfcL5jbl5oEzLnBooo9vV/nf3sPZU/5V9wvmN60nnNlT3oop+3q/zv72Hsqf8AKvuF85/m5pA5XoaKKXt6v87+9h7Kn/KvuF+0SetL57+v6Ciin7er/O/vYeyp/wAq+4aJ3GVzxSi4deQeaKKXt6v87+9h7Kn/ACr7gWVvWlW4kx979BRRT9vV/nf3sPZU/wCVfcO+1StwW447CmC4cN1/QUUVXt6v87+9h7Kn/KvuHfaZP736Cj7TJ/e/QUUUe3q/zv72Hsqf8q+4T7VI3Vv0FL9pk4G7gdOBRRU+3q/zv72Hsqf8q+4X7VL/AHv0FCX0y9H/AEFFFV7er/O/vYeyp/yr7hPt02c7v0FBvJmABfj6Ciij29X+d/ew9lT/AJV9wfbpv7/6CpPts20Dfx/uj/Ciin7er/O/vZPsqf8AKvuGDUJ16P8AoKlN/cH/AJaf+Oj/AAooo9vV/nf3sPZU/wCVfcNN9OWHz/8Ajo/wpy6hcbf9YP8Avhf8KKKPb1f5397D2VP+VfcH9oXH/PT/AMdH+FJ9vuMsfM5/3R/hRRR7er/O/vYeyp/yr7hx1C4P/LT/AMdH+FRtqVwzcyfoKKKPb1f5397D2NP+VfcSDULhekn/AI6P8KX+0rnn971/2V/woope3q/zv72X7Gl/KvuEOpXLf8tf/HV/wo/tK5wB5vH+6P8ACiij29X+d/ew9jS/lX3A2p3P/PT/AMcX/Cj+1br/AJ6D/vhf8KKKr29X+d/ew9hS/kX3IX+1bvb/AK7/AMdX/Cj+07rGDLke6r/hRRU+3rfzv72HsKX8i+5AuqXW3/W/+Or/AIUo1a7UqfO5/wB1f8KKKPb1v5397D2FL+RfcgGrXalsTf8Ajq/4Ug1i8Bz53P8Aur/hRRT+sVv5397D2FL+Rfch7atef89z/wB8r/hTF1i8VQBN/wCOr/hRRVe3rfzv72HsKX8i+5CnV7st/rv/AB1f8KX+2Lzd/rv/AB1f8KKKPb1v5397D2FL+Rfcg/ta7/57f+Or/hQutXitxN/46v8AhRRR7et/O/vYewpfyL7kJ/bV5tx53H+4v+FKNZvCv+ux/wABX/Ciij29b+d/ew9hS/kX3IDrl6V2+fx/ur/hS/21e/8APbp0+Rf8KKKPb1v5397D2FL+RfchP7cveP3/AE/2F/wo/ty9/wCe/wD44v8AhRRR7et/O/vYewpfyL7kJ/bV7/z3/wDHV/wp39vX23/X/wDji/4UUUe3rfzv72L2FL+RfchRrl8Mfv8A/wAcX/Cm/wDCQX//AD3/APHF/wAKKKy+s1/5397I9hS/kX3IX+3b7b/r/wDxxf8AChddvl6T/wDji/4UUVr7et/O/vZfsKX8i+5B/wAJBf8AXz+f9xf8KP7evsH9/wD+OL/hRRWX1mv/ADv72R7Cl/IvuQHXr89Z/wDxxf8ACk/4SC//AOe//ji/4UUUfWa/87+9h7Cl/IvuQ5dfv16T/wDji/4Un9v3+7/X/wDji/4UUUfWa/8AO/vYewpfyL7kKfEF/g/v/wDxxf8ACkGu3wx+/wD/ABxf8KKKPrNf+d/ew9hS/kX3IP8AhItQ/wCe/wD44v8AhR/wkF+y8z/+OL/hRRV/Wa/87+9h7Cl/IvuQf8JHqH/Pf/xxf8KX/hIL/GPP/wDHF/wooo+s1/5397D2FL+Rfcg/t+/248/j/cX/AAoooqZVqsrNyf3mkaVNN2ivuP/Z)

**Fig. ‎4.2‑1 SPA**

# Database Design

In this section, we will discuss in details the database schema providing an entity-relationship diagram for the system and data dictionary. We try to build our schema in the best way that fit with our application. We have provided seven tables with deferent kind of attribute, type, if it’s required or not, the type of each field and a description for each field. Also, we provide a schema diagram that shows the deferments tables we have and demonstrate the relations between each table.

Diagram

Description automatically generated

**Fig. ‎4.3‑1 Database table schema**

## Data dictionary

1. User

Table. ‎4.3.1.1User Entry

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Description** | **Type** | **PK** | **Required** |
| id | The user id | Integer | Yes | Yes |
| first\_name | First name for the user | Varchar | No | Yes |
| last\_name | Last name for the user | Varchar | No | Yes |
| email | The user email | Varchar | No | Yes |
| role | The user role customer or merchant | Varchar | No | Yes |
| username | The username | Varchar | No | Yes |
| password | The user password field | Varchar | No | Yes |
| Is\_admin | If the user is admin | Boolean | No | No |
| Is\_staff | If the user for staff | Boolean | No | No |

1. customer

Table. ‎4.3.1.2Customer Entry

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute** | **Description** | **Type** | **FK** | **PK** | **Required** |
| Id | The customer id | Integer | No | Yes | Yes |
| user | The customer foreign key | User | Yes | No | Yes |

1. merchant

Table. ‎4.3.1.3Merchant Entry

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute** | **Description** | **Type** | **FK** | **PK** | **Required** |
| Id | The merchant id | Integer | No | Yes | Yes |
| user | The merchant foreign key | User | Yes | No | Yes |

1. Store

Table. ‎4.3.1.4Store Entry

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute** | **Description** | **Type** | **FK** | **PK** | **Required** |
| id | The store id | Integer | No | Yes | Yes |
| merchant | The merchant foreign key | merchant | Yes | No | Yes |
| name | The store name | Varchar | No | No | Yes |
| image | The store image | image | No | No | Yes |
| description | The store description | Text | No | No | Yes |
| Slug | The store slug for URL use | Slug | No | No | Yes |

1. Product

Table. ‎4.3.1.5Product Entry

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute** | **Description** | **Type** | **FK** | **PK** | **Required** |
| Id | The product id | Integer | No | Yes | Yes |
| Store | The store foreign key | Store | Yes | No | Yes |
| Customer | The customer foreign key | customer | Yes | No | Yes |
| Name | The product name | Varchar | No | No | Yes |
| Image | The product image | image | No | No | Yes |
| Price | The product price | Float | No | No | Yes |
| Digital | If product is digital | boolean | No | No | Yes |

1. Order

Table. ‎4.3.1.6Order Entry

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute** | **Description** | **Type** | **FK** | **PK** | **Required** |
| Id | The order id | Integer | No | Yes | Yes |
| Customer | The customer foreign key | Customer | Yes | No | Yes |
| Date | The order date | Date | No | No | Yes |
| Complete | If order is complete | Boolean | No | No | Yes |
| Transaction | The order transaction | Varchar | No | No | Yes |

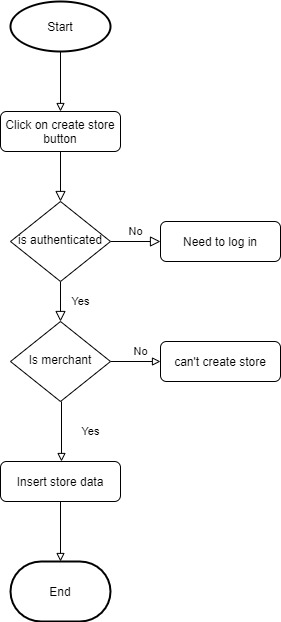
1. Order Item

Table. ‎4.3.1.7Order Item Entry

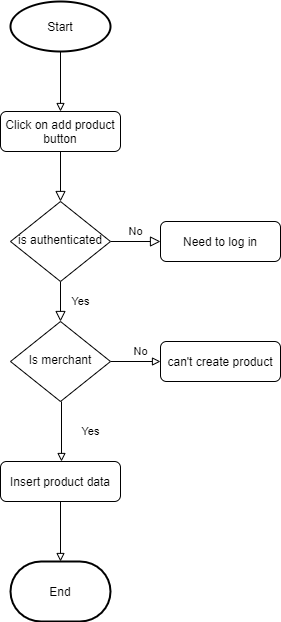
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute** | **Description** | **Type** | **FK** | **PK** | **Required** |
| Id | The order item id | Integer | No | Yes | Yes |
| Product | The order item foreign key | Product | Yes | No | Yes |
| Date | The order item date | Date | No | No | Yes |
| Order | If order item foreign key | Order | Yes | No | Yes |
| Quantity | The order item quantity | Integer | No | No | Yes |

# Flow Chart

In this section we will be adding some useful flow charts to describe how our web application works and make it simpler to be understood. First chart, will describe the create store function which is a feature for our merchant user. Second, will be the create product function which is also another feature for out merchant user.



**Fig. ‎4.4‑1Create Store**

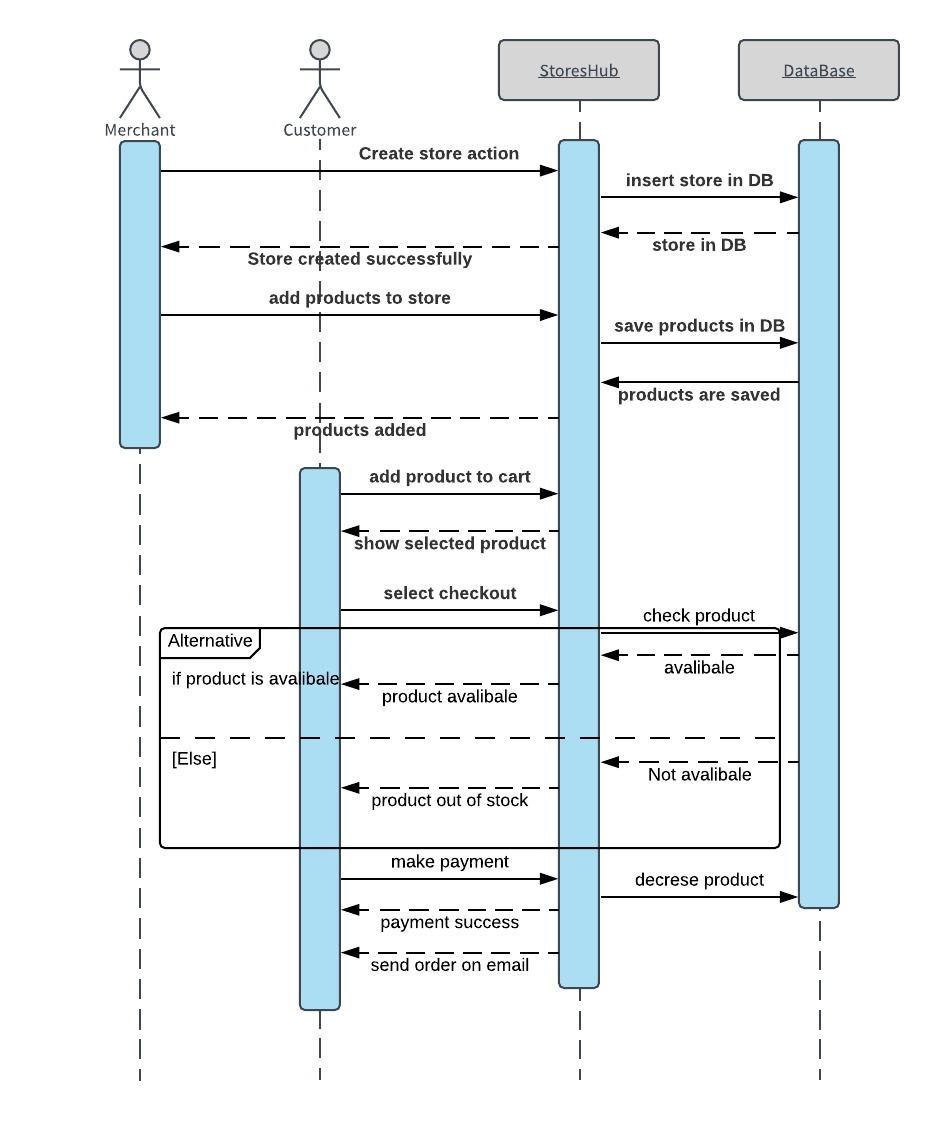
****

**Fig. ‎4.4‑2Create Product**

# System Organization

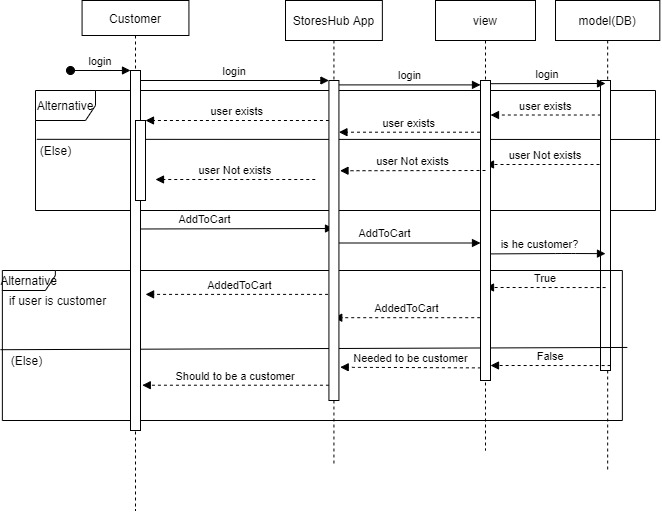
## Sequence diagram

1. **Merchant and Customer**

****

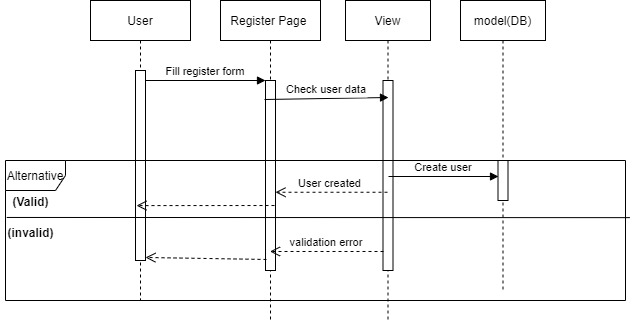
**Fig. ‎4.5‑1Our Users Features**

1. **Add to cart**



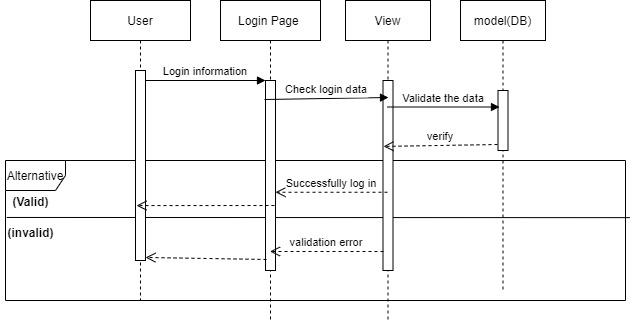
**Fig. ‎4.5‑2Add\_to\_cart**

1. **Registration**

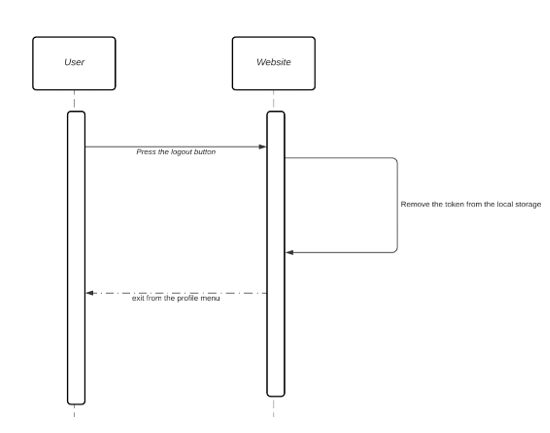
****

**Fig. ‎4.5‑33.Registration**

1. **Login**



**Fig. ‎4.5‑4Login**

1. **Logout**  
   

**Fig. ‎4.5‑5Logout**

# Design pattern

Since we are using Django framework, we choose to model, view, template (MVT) pattern because it’s strongly support by Django and give the ability to increase the performance of our application. MVT works as follow, M for model which will handle the database using ORM engine that allow us to work with database easily in class based, V for views which will hold our application logic, T stands for template that will contain the front-end part and all the interface design will be there. This figure below will illustrate our design pattern.

Diagram

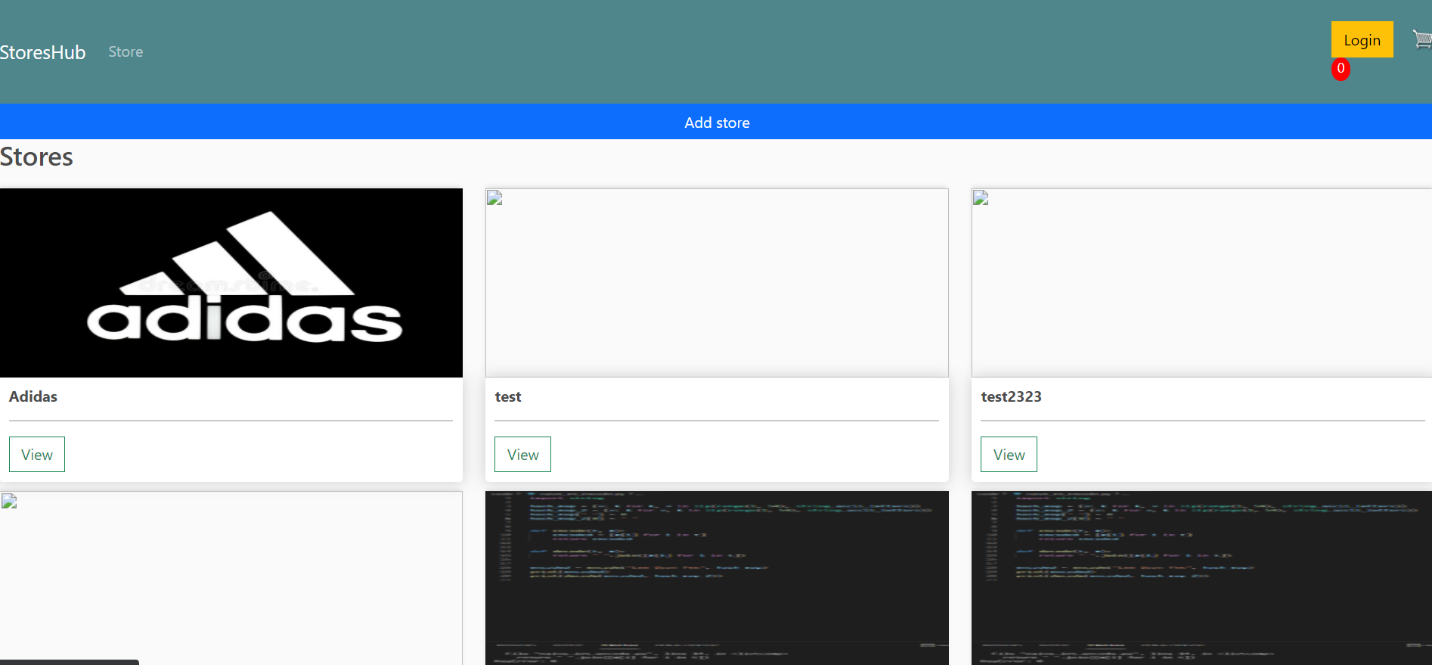
Description automatically generated

**Fig. ‎4.6‑1MVT pattern**

# Graphical User Interface Design

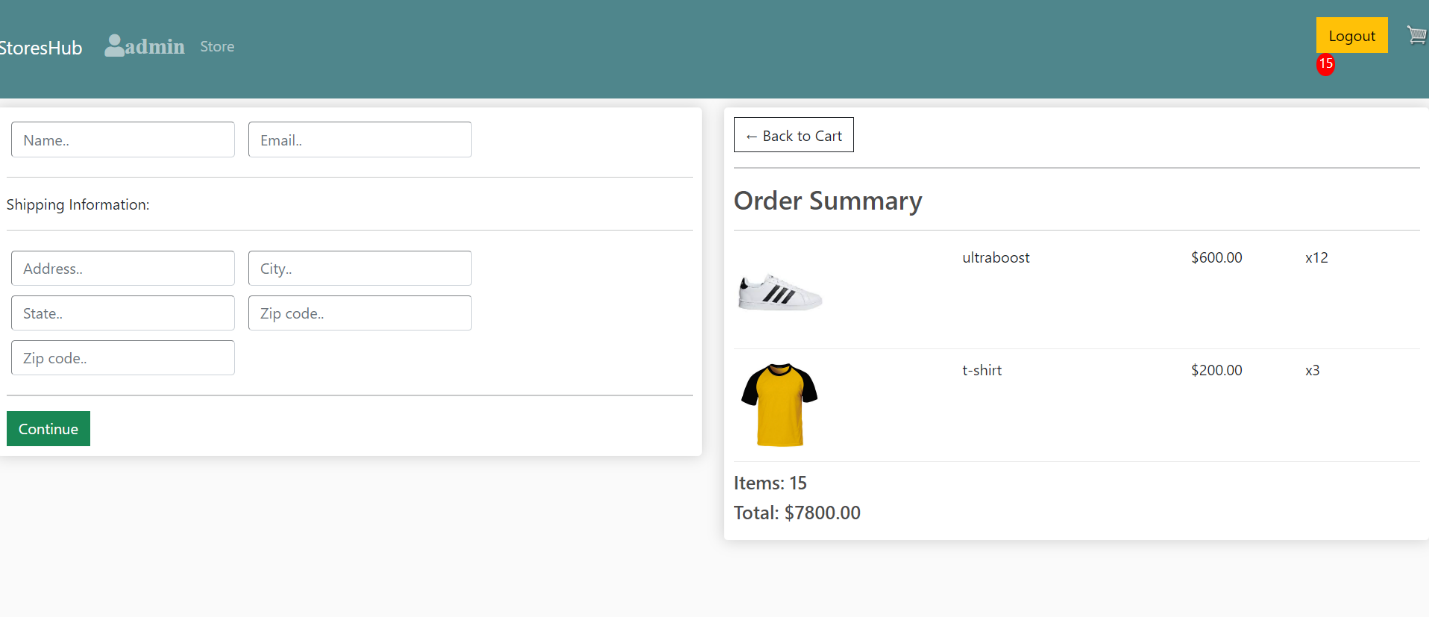
At this point we will afford some images as a prototype to the user interface GUI. Although it **could be changed during the implementation as our project needs.**

* **Home Page**

****

**Fig. ‎4.7‑1Home Page**

* **Checkout Page**

****

***Fig. ‎4.7‑2Checkout Page***

* **Sign in Page**

Graphical user interface, website

Description automatically generated

**Fig. ‎4.7‑3Sign in Page**

* **Registration page**

Graphical user interface, table

Description automatically generated

**Fig. ‎4.7‑4Registration page**

# Conclusion

In conclusion, we did provide design for our application in deferent levels. We have discussed the System Architecture with figures to describe it better. Then, we did provide a database schema with schedules for each table in database with explanation for each entry attribute, types description, and keys. Next, we showed System Organization, GUI design with a prototype, and design pattern that we use and implement. In the next chapter 5, we will talk more about implementation phase.

#### Implementation

# Introduction

For this chapter, we will talk more about the implementation phase. First, we will discuss the implementation requirements and listing the hardware and software requirements with what programming languages and tools we have been used in order to implement our project.

# Implementation Requirements

## Hardware Requirements

Since we are devolving a web application the hardware requirements will simple and available for most of the time. In implementation we used a computer with MacOS and windows operating system but also using different operating system will be fine as well. For devices we can use any device with a web browser software such as smart phone, laptop and desktop

## Software Requirements

Table. ‎5.2.2.1Software Requirements

|  |  |  |
| --- | --- | --- |
| **Software** | **Discerption** | **Example** |
| Code editor | Used to write code in different Programming languages | -Visual Studio code  -Atom  -PyCharm  -Sublime |
| Browser | For running software on it.  with browsers developer tools for debugging and development | -Google Chrome  -Firefox  -Safari |
| Database management  Tool | to control and manage the databases | -python prompt |
| GUI Design | Tools for design and prototyping GUI | -Adobe XD  -Figma  -Photoshop |
| Diagram tool | Tools for drawing diagrams and use cases | -Lucid Chart  -Draw.io  -Microsoft Visio |

## Programming Language(s)

* **Python**

We did choose python because it’s simple. very powerful and popular language with good documentation and community. As well with Django which is a python framework used to develop our web application.

* **JavaScript**

For the clint side logic and computing we choose java scrip that will run on browser. It will give us the ability to make our application more dynamic.

* **HTML/CSS/ Bootstrap**

We used HTML/CSS to crate our front-end pages with using bootstrap which is CSS framework to implement the design.

* **SQL**

We used SQL language to create, update, and query our database

* **Installing libraries**

We been used pip tool which is the package manager installer for python. To install any package on python you can type this command.

pip install [package name]

for our case with Django framework, we need to use command

pip install django=3.2

the best way for our case is to type all our requirements on one file and then install those libraries at once like this:

pip install requirement.txt

## Tools and Technologies

Here will mention all the tools that we have user that nether considered as programming language or requirements in table below.

Table. ‎5.2.4.1Tools and Technologies

|  |  |
| --- | --- |
| **Tool** | **Discerption** |
| Git | is software for tracking and managing the code changes |
| GitHub | Is a platform for uploading and hosting project repositories and managing the team |
| Slack | It’s a team management tool |
| PayPal API | It’s an API for payment gateway |
| Virtual Environment | It’s a tool to create virtual environment for project |
| Digital Ocean | It’s a hosting platform |

# Code Implementation

At the code level, since we using MVT architecture that allow us to divide our code into three main classes model, view and template and others.

* **Model**

In model class we will create our database and add our needed tables with fields and functions. We will have eight classes each class represent a table and have attributes as fields. The way that we use classes as tables this feature comes from ORM engine which gave us the ability to deal with our database as python objects.

* **View**

In view class we will write all the logic to deal with the template and model and handle the requests that comes from the template then check if the client request need to query the database or create an object then we direct him back to web page as a response.

* **Template**

in template we will create and design our web pages as a presentation step and then the view class will return the template as response.

* **URLs**

This class will be responsible about our web application URLs. For example each time we create a template page in order to add it to our URL path we need to register this template as a URL.

* **Admin**

This class is responsible to show our database tables in admin panel. For example if we need to see any table from mode class we need to register it here.

* **Sittings**

This class have all the sittings of our web application.

# Conclusion

In this chapter, we have given a good explanation for the implementation phase and provided the system and software requirement, the programming languages, libraries, frameworks and technologies that we used. Although the code implementation section. At the next chapter we will discuss the testing phase which is an important phase that will show us if our project met the specifications.

#### Testing

# Introduction

In this phase, we will test our web application functionality with different uses cases to ensure that our system met the requirement specifications in all the situations. Testing will be applied on the following functionalities Add to cart, Checkout, Payment, create store, Create product, Registration, Login and Logout

# Testing

Here we will write our test cases and test it manually. Unfortunately, we couldn’t have a lot of test cases and functionality to be tested because of the time matter.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function name** | Registration | | | |
| **Description** | Show user details and Properties had been listed | | | |
| **Pre-conditions** | User must be logged in to show his profile and Properties | | | |
| **Test Number** | **Test Case** | **Expected**  **Outcome** | **Actual Outcome** | **Pass/Fail** |
| 1 | Show user details | User details should be shown | User details will be shown | Pass |
| 2 | List user properties | All user properties should be listed | All user properties will be listed | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function name** | Login | | | |
| **Description** | Show user details and Properties had been listed | | | |
| **Pre-conditions** | User must be logged in to show his profile and Properties | | | |
| **Test Number** | **Test Case** | **Expected**  **Outcome** | **Actual Outcome** | **Pass/Fail** |
| 1 | Show user details | User details should be shown | User details will be shown | Pass |
| 2 | List user properties | All user properties should be listed | All user properties will be listed | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function name** | Add\_to\_cart | | | |
| **Description** | Show user details and Properties had been listed | | | |
| **Pre-conditions** | User must be logged in to show his profile and Properties | | | |
| **Test Number** | **Test Case** | **Expected**  **Outcome** | **Actual Outcome** | **Pass/Fail** |
| 1 | Show user details | User details should be shown | User details will be shown | Pass |
| 2 | List user properties | All user properties should be listed | All user properties will be listed | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function name** | CreateStore | | | |
| **Description** | Show user details and Properties had been listed | | | |
| **Pre-conditions** | User must be logged in to show his profile and Properties | | | |
| **Test Number** | **Test Case** | **Expected**  **Outcome** | **Actual Outcome** | **Pass/Fail** |
| 1 | Show user details | User details should be shown | User details will be shown | Pass |
| 2 | List user properties | All user properties should be listed | All user properties will be listed | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function name** | AddProduct | | | |
| **Description** | Show user details and Properties had been listed | | | |
| **Pre-conditions** | User must be logged in to show his profile and Properties | | | |
| **Test Number** | **Test Case** | **Expected**  **Outcome** | **Actual Outcome** | **Pass/Fail** |
| 1 | Show user details | User details should be shown | User details will be shown | Pass |
| 2 | List user properties | All user properties should be listed | All user properties will be listed | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function name** | Payment | | | |
| **Description** | Show user details and Properties had been listed | | | |
| **Pre-conditions** | User must be logged in to show his profile and Properties | | | |
| **Test Number** | **Test Case** | **Expected**  **Outcome** | **Actual Outcome** | **Pass/Fail** |
| 1 | Show user details | User details should be shown | User details will be shown | Pass |
| 2 | List user properties | All user properties should be listed | All user properties will be listed | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function name** | Checkout | | | |
| **Description** | Show user details and Properties had been listed | | | |
| **Pre-conditions** | User must be logged in to show his profile and Properties | | | |
| **Test Number** | **Test Case** | **Expected**  **Outcome** | **Actual Outcome** | **Pass/Fail** |
| 1 | Show user details | User details should be shown | User details will be shown | Pass |
| 2 | List user properties | All user properties should be listed | All user properties will be listed | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function name** | Profile | | | |
| **Description** | Show user details and Properties had been listed | | | |
| **Pre-conditions** | User must be logged in to show his profile and Properties | | | |
| **Test Number** | **Test Case** | **Expected**  **Outcome** | **Actual Outcome** | **Pass/Fail** |
| 1 | Show user details | User details should be shown | User details will be shown | Pass |
| 2 | List user properties | All user properties should be listed | All user properties will be listed | Pass |

# Conclusion

At the end of this chapter, we have provided various of test cases to our functionalities from authentication cases, until dealing with the store and product. We have used the manual test method to test our test cases. In the next chapter we will provide Conclusion and Future Work of our project.

#### Conclusion and Future Work

# Introduction

In this chapter, we will give a conclusion of our project work. Although, for future work we will provide what we could add to improve our project in general.

# Future Work

Here we will list the features that we need to include in future work.

* **Stocks service**

Here we want to add a microservice to our system to handle the stocking service. We want to give our merchants a service to their store product in our stocks.

* **Delivery service**

Since the merchants’ products in our stocks we want to give them the delivery service so, when a customer buy a product it will be deliver directly from our stock and no need to our merchants to be involve in that process.

* **Subscriptions**

Here we want to have subscriptions method for our merchants. Since our merchant have many great services so when he want to create store in our system he will need to subscribe first and pay for that store. Until now there’s no clear subscriptions method to use but we will consult financial company to do that.

* **Sign in APIs**

In order to improve our clints experience we need to provide more than one method for sign in and registration. We thinking to use google, apple and twitter APIs for signing in.

* **bot sign in API**

This feature gives us security aspect to block bots such that, bot API that google provide

* **Deployment and hosting**

In deployment and hosting we need to use cloud services that AWS provide or Oracle.

# Project Conclusion

When got started this project, we been focusing only to solve the problem of merchants and customers by developing a web application platform for e-commerce, but once we got in the middle of work, we notice that we we're missing a lot of knowledge about techniques, tools programming, and APIs. Through this project we have learned that any problem has a solution and you just need to think in another way. This project taught us how to solve problems, working with team to reach a sharable goal, how to manage our selves and time to get work done, and the important thing is how to examine and gather information from credible and authoritative sources.

# Conclusion

At the end of this chapter, we did provide what our project will need in future, we provided different features and services. Although, we did talk about conclusion in details. This is the last chapter in this project, we have been working on this project almost a year and it was a long journey until now that we learned a lot from it.

### Appendices

**Code**

* **Front-end**

**Login**

{% extends 'store/main.html' %}

<!DOCTYPE html>

<html lang='en'>

<head>

<title>Login</title>

</head>

<body>

{% block content %}

<div class="d-flex justify-content-center">

<div class="login\_card" >

<div class="d-flex justify-content-center m-3">

<h3 id="form-title">Sign In into StoresHub</h3>

</div>

<div class="d-flex form-control justify-content-center form\_container">

<form method="POST" action ="">{% csrf\_token %}

<div class="input-group mb-3">

<input type="text" name="username" placeholder="Username..." class="form-control">

</div>

<div class="input-group mb-2">

<input type="password" name="password" placeholder="Password..." class="form-control" >

</div>

<div class="d-flex justify-content-center mt-3 login\_container">

<input class="btn login\_btn" type="submit" value="Sign In">

</div>

</form>

</div>

{{form.errors}}

</div>

</div>

{% endblock content %}

</body>

</html>

**Register**

{% extends 'store/main.html' %}

<!DOCTYPE html>

<html lang='en'>

<head>

<title>Register</title>

</head>

<body>

{% block content %}

<div class="d-flex justify-content-center">

<div class="regisret-card" >

<div class="d-flex justify-content-center m-3">

<h3 id="form-title">Sign In into StoreHub</h3>

</div>

<div class="d-flex justify-content-center form\_container">

<form method="POST" action ="">

{% csrf\_token %}

{{ form }}

<div class="d-flex justify-content-center mt-3 login\_container">

<input class="btn login\_btn" type="submit" value="Register">

</div>

</form>

</div>

</div>

</div>

{% endblock content %}

</body>

</html>

**Main**

<!DOCTYPE html>

{% load static %}

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<!--This view port to make sure it looks good on phone-->

<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, minimum-scale=1" />

<!--Adding boostrap-->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.8.1/css/all.css">

<link rel="stylesheet" type="text/css" href="{% static 'css/main.css' %}">

<title>StoresHub </title>

<script type="text/javascript">

var user = '{{request.user}}'

</script>

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

<a class="navbar-brand" href="{% url 'store' %}">StoresHub</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav mr-auto">

<li class='nav-item'>

<a class="nav-link" href="/profile"><i class="fa fa-user" style="font-size:24px">{{user.username }}</i></a>

</li>

<li class="nav-item active">

<a class="nav-link" href="{% url 'store' %}">Store <!--<span class="sr-only">(current)</span></a> -->

</li>

</ul>

-->

<div class="form-inline my-2 my-lg-0" style="margin-left:auto">

{% if user.is\_authenticated %}

<a href="/logout"class="btn btn-warning">Logout</a>

{% else %}

<a href="/login"class="btn btn-warning">Login</a>

<a href="/register"class="btn btn-warning">Register</a>

{% endif %}

{% if user.role == "CU" %}

<a href="{% url 'cart' %}"><img id="cart-icon" src="{% static 'images/cart.png' %}"></a>

<p id="cart-total">{{cartItem}}</p>

{% endif %}

</div>

</div>

</nav>

{% block content %}

<!-- this is a hlock to extend -->

{% endblock content %}

</div>

<!--Addong boostrap javascript-->

<script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js" integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6" crossorigin="anonymous"></script>

<!-- Adding our javascript -->

<script type="text/javascript" src="{% static 'js/cart.js' %}"></script>

</body>

</html>

**Store**

{% extends 'store/main.html' %}

{% load static %}

{% block content %}

<div class="row">

<!--this loop to render out products

if a product not having pic we will get error so we need to handel this

-->

{% if user.role == 'ME' %}

<a href="/add\_store" class="btn btn-primary">

Add store

</a>

{% endif %}

<h3>Stores</h3>

{% for store in stores %}

<div class="col-lg-4">

<a href="stores/{{ store.slug }}"><img class="thumbnail" src="{{ store.imageURL }}"> </a>

<div class="box-element product">

<h6> <strong> {{ store.name }} </strong></h6>

<hr>

<a class="btn btn-outline-success" href="#">View</a>

</div>

</div>

{% endfor %}

<hr>

<h3>Products</h3>

{% for product in products %}

<div class="col-lg-4">

<img class="thumbnail" src="{{ product.imageURL }}">

<div class="box-element product">

<h6> <strong> {{ product.name }} </strong></h6>

<hr>

{% if user.role == 'CU' %}

<form method="POST" action ="/add\_to\_cart/{{ product.id }}"> {% csrf\_token %}

<input type="submit" data-product="{{product.id}}" data-action="add" class="btn btn-outline-secondary add-btn update-cart">Add to Cart</input>

</form>

{% endif %}

<a class="btn btn-outline-success" href="#">View</a>

<h4 style="display: inline-block; float: right"><strong> ${{ product.price|floatformat:2 }} </strong></h4>

</div>

</div>

{% endfor %}

</div>

{% endblock %}

**Stores**

{% extends 'store/main.html' %}

<!doctype html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- Bootstrap CSS -->

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

<title>Hello, world!</title>

</head>

{% block content %}

<body>

<div id="container-float">

<div class="row justify-content-center m-0">

<div class="col-sm-4 col-md-5 m-0 text-center">

<img style="width: 300px; height: 300px; border-radius: 50%;" src='{{ store.imageURL }}'/>

</div>

<div class="col text-left">

<h1 class="header">{{ store.name}}</h1>

<p>{{ store.description }}</p>

{% if user.role == 'ME' %}

<a href="/add-product" class="btn btn-primary">

Add product

</a>

{% endif %}

</div>

</div>

<div class="row highlight">

<h3>Products</h3>

{% for product in products %}

<div class="col-lg-4">

<img class="thumbnail" src="{{ product.imageURL }}">

<div class="box-element product">

<h6> <strong> {{ product.name }} </strong></h6>

<hr>

<form method="POST" action ="http://127.0.0.1:8000/add\_to\_cart/{{ product.id }}">

{% csrf\_token %}

<input type="submit" class="btn btn-outline-secondary add-btn update-cart">Add to Cart</input>

</form>

<a class="btn btn-outline-success" href="#">View</a>

<h4 style="display: inline-block; float: right"><strong> ${{ product.price }} </strong></h4>

</div>

</div>

{% endfor %}

</div>

</div>

<h1>Hello, world!</h1>

<!-- Optional JavaScript -->

<!-- jQuery first, then Popper.js, then Bootstrap JS -->

<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl" crossorigin="anonymous"></script>

</body>

{% endblock content %}

</html>

**Checkout.html**

{% extends 'store/main.html' %}

{% load static %}

{% block content %}

<div class="container">

<div class="row">

<div class="col-lg-6">

<div class="box-element">

<a class="btn btn-outline-dark" href="{% url 'cart' %}">&#x2190; Back to Cart</a>

<hr>

<h3>Order Summary</h3>

<hr>

{% for item in items %}

<div class="cart-row">

<div style="flex:2"><img class="row-image" src="{{item.product.imageURL}}"></div>

<div style="flex:2"><p> {{item.product.name}}</p></div>

<div style="flex:1"><p>${{item.product.price| floatformat:2}}</p></div>

<div style="flex:1"><p>x{{item.qunatity}}</p></div>

</div>

{% endfor %}

<h5>Items: {{order.get\_cart\_items}}</h5>

<h5>Total: ${{order.get\_cart\_total|floatformat:2}}</h5>

</div>

</div>

<div class=" col box-element" id="form-wrapper">

<form id="form" method="POST">

{% csrf\_token %}

{{ form.as\_p }}

<hr>

<input id="form-button" onclick="toggleDisplayPaypal();" class="btn btn-success btn-block" type="submit" value="Continue">

</form>

<div class="box-element hidden" id="payment-info">

<small>Paypal Options</small>

</div>

<div id="collapsePaypal" class="wrapper">

<div id="paypal-button-container"></div>

<!--This sript is will give access for paypal API Buttons-->

<script src="https://www.paypal.com/sdk/js?client-id=AfwHwz-Sz8PtrZgvCrvQhrXHxJyt7MRUEodU1zvL3MRyGej8oQ\_2AolosviosQuGAkGl9ra1ilHGbzCn&currency=USD&disable-funding=credit" data-sdk-integration-source="button-factory"></script>

<script>

function getCookie(name) {

let cookieValue = null;

if (document.cookie && document.cookie !== '') {

const cookies = document.cookie.split(';');

for (let i = 0; i < cookies.length; i++) {

const cookie = cookies[i].trim();

// Does this cookie string begin with the name we want?

if (cookie.substring(0, name.length + 1) === (name + '=')) {

cookieValue = decodeURIComponent(cookie.substring(name.length + 1));

break;

}

}

}

return cookieValue;

}

const csrftoken = getCookie('csrftoken');

var price = "{{ order.get\_cart\_total|floatformat:2 }}"

function completeOrder(userData, userActions) {

var url = "/complete\_paypal\_payment"

fetch(url, {

method: 'POST',

headers:{

'Content-type': 'application/json',

'X-CSRFToken': csrftoken

},

body:JSON.stringify({

"Data": userData,

"Actions": userActions})

})

}

paypal.Buttons({

style: {

shape: 'rect',

color: 'black',

layout: 'vertical',

label: 'checkout',

},

createOrder: function(data, actions) {

return actions.order.create({

purchase\_units: [{

amount: {

value: price

}

}]

});

},

onApprove: function(data, actions) {

return actions.order.capture().then(function(details) {

completeOrder(data, actions)

alert('Transaction completed by ' + details.payer.name.given\_name + '!');

});

}

}).render('#paypal-button-container');

</script>

<script type="text/javascript">

function toggleDisplayPaypal() {

var x = document.getElementById("collapsePaypal");

console.log(x)

if (x.style.display === "none") {

x.style.display = "block";

} else {

x.style.display = "none";

}

};

</script>

</div>

</div>

</div>

</div>

{% endblock content %}

**Cart.html**

{% extends 'store/main.html' %}

{% load static %}

{% block content %}

<div class="row">

<div class="col-lg-12">

<div class="box-element">

<a class="btn btn-outline-dark" href="{% url 'store' %}">&#x2190; Continue Shopping</a>

<br>

<br>

<table class="table">

<tr>

<th><h5>Items: <strong>{{order.get\_cart\_items}}</strong></h5></th>

<th><h5>Total:<strong> ${{order.get\_cart\_total|floatformat:2 }}</strong></h5></th>

<th>

<a style="float:right; margin:5px;" class="btn btn-success" href="{% url 'checkout' %}">Checkout</a>

</th>

</tr>

</table>

</div>

<br>

<div class="box-element">

<div class="cart-row">

<div style="flex:2"></div>

<div style="flex:2"><strong>Item</strong></div>

<div style="flex:1"><strong>Price</strong></div>

<div style="flex:1"><strong>Quantity</strong></div>

<div style="flex:1"><strong>Total</strong></div>

</div>

{% for item in items %}

<div class="cart-row">

<div style="flex:2"><img class="row-image" src="{{item.product.imageURL}}"></div>

<div style="flex:2"><p> {{item.product.name}}</p></div>

<div style="flex:1"><p>${{item.product.price|floatformat:2}}</p></div>

<div style="flex:1">

<p class="quantity">x{{item.qunatity}}</p>

<div class="quantity">

<form method="POST" action ="http://127.0.0.1:8000/update\_item/add/{{ item.product.id }}">

{% csrf\_token %}

<button type="submit" class="btn btn-outline-secondary add-btn update-cart"><img data-product={{item.product.id}} data-action="add" class="chg-quantity update-cart" src="{% static 'images/arrow-up.png' %}"></button>

</form>

<form method="POST" action ="http://127.0.0.1:8000/update\_item/remove/{{ item.product.id }}">

{% csrf\_token %}

<button type="submit" class="btn btn-outline-secondary add-btn update-cart"><img data-product={{item.product.id}} data-action="remove"class="chg-quantity update-cart" src="{% static 'images/arrow-down.png' %}"></button>

</form>

<!-- <img data-product={{item.product.id}} data-action="add" class="chg-quantity update-cart" src="{% static 'images/arrow-up.png' %}">

<img data-product={{item.product.id}} data-action="remove"class="chg-quantity update-cart" src="{% static 'images/arrow-down.png' %}"> -->

</div>

</div>

<div style="flex:1"><p>${{item.get\_total}}</p></div>

</div>

{% endfor %}

</div>

</div>

</div>

{% endblock content %}

**Add\_store.html**

{% extends 'store/main.html' %}

<!doctype html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- Bootstrap CSS -->

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

<title>Hello, world!</title>

</head>

{% block content %}

<body>

{% if messages %}

{% for message in messages %}

<div class="alert alert-warning alert-dismissible fade show" role="alert">

<li {% if message.tags %} class="{{ message.tags }}" {% endif %}>{{ message }}</li>

<button type="button" class="close" data-dismiss="alert" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

{% endfor %}

{% endif %}

<div id="container-float">

<div class="row justify-content-center m-0">

<form method="POST" enctype="multipart/form-data">{% csrf\_token %}

{{ form.as\_p}}

<input class="btn" type="submit" value="create store">

</form>

</div>

</div>

<!-- Optional JavaScript -->

<!-- jQuery first, then Popper.js, then Bootstrap JS -->

<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl" crossorigin="anonymous"></script>

</body>

{% endblock content %}

</html>

**Addproduct.html**

{% extends 'store/main.html' %}

<!doctype html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- Bootstrap CSS -->

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

<title>Hello, world!</title>

</head>

{% block content %}

<body>

{% if messages %}

{% for message in messages %}

<div class="alert alert-warning alert-dismissible fade show" role="alert">

<li {% if message.tags %} class="{{ message.tags }}" {% endif %}>{{ message }}</li>

<button type="button" class="close" data-dismiss="alert" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

{% endfor %}

{% endif %}

<div id="container-float">

<div class="row justify-content-center m-0">

<form method="POST" enctype="multipart/form-data">{% csrf\_token %}

{{ form.as\_p}}

<input class="btn" type="submit" value="add product">

</form>

</div>

</div>

<!-- Optional JavaScript -->

<!-- jQuery first, then Popper.js, then Bootstrap JS -->

<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl" crossorigin="anonymous"></script>

</body>

{% endblock content %}

</html>

**Main.css**

body{

background-color: hsl(0, 0%, 98%);

}

h1,h2,h3,h4,h5,h6{

color:hsl(0, 0%, 30%);

}

.box-element{

box-shadow:hsl(0, 0%, 80%) 0 0 16px;

background-color: #fff;

border-radius: 4px;

padding: 10px;

}

.thumbnail{

width: 100%;

height: 200px;

-webkit-box-shadow: -1px -3px 5px -2px rgba(214,214,214,1);

-moz-box-shadow: -1px -3px 5px -2px rgba(214,214,214,1);

box-shadow: -1px -3px 5px -2px rgba(214,214,214,1);

}

.form-control {

border-style:inset;

}

.checkbox {

margin-left: 21px;

border-width: 20px;

}

/\* On mouse-over, add a grey background color \*/

input ~ .checkmark {

background-color: #ccc;

}

/\* When the checkbox is checked, add a blue background \*/

input:checked ~ .checkmark {

background-color: #2196F3;

}

.product{

border-radius: 0 0 4px 4px;

}

.bg-dark{

background-color: #4f868c!important;

}

#cart-icon{

width: 25px;

display: inline-block;

margin-left: 15px;

}

#cart-total{

display: block;

text-align: center;

color:#fff;

background-color: red;

width: 20px;

height: 25px;

border-radius: 50%;

font-size: 14px;

}

.col-lg-4, .col-lg-6, .col-lg-8, .col-lg-12{

margin-top: 10px;

}

.btn{

border-radius: 0;

}

.row-image{

width: 100px;

}

.form-field{

width:250px;

display: inline-block;

padding: 5px;

}

.cart-row{

display: flex;

align-items: flex-stretch;

padding-bottom: 10px;

margin-bottom: 10px;

border-bottom: 1px solid #ececec;

}

.quantity{

display: inline-block;

font-weight: 700;

padding-right:10px;

}

.chg-quantity{

width: 12px;

cursor: pointer;

display: block;

margin-top: 5px;

transition:.1s;

}

.chg-quantity:hover{

opacity: .6;

}

.hidden{

display: none!important;

}

.highlight {

border-style:solid;

color: red;

}

.header {

font-family: 'Courier New', Courier, monospace;

font-size: 5rem;

}

.text-left {

text-align: left;

}

* **Back-end**

**Model.py**

from django.db import models

from django.db.models.fields.related import OneToOneField

# importing to create custom user

from django.contrib.auth.models import AbstractUser

from django.utils.translation import ugettext\_lazy as \_

# import to create custom user model

from django.contrib.auth.base\_user import BaseUserManager

from django.utils.translation import ugettext\_lazy as \_

from django.urls import reverse

from django.utils.text import slugify

ROLES\_CHOICES = (

('ME', 'Merchant'),

('CU', 'Customer')

)

# Create your models here.

class CustomUserManager(BaseUserManager):

""" Custom user manger"""

def create\_user(self, username, password, \*\*extra\_fields):

"""

Create and save a User with the given email and password.

"""

if not username:

raise ValueError(\_('The username must be set'))

user = self.model(username=username, \*\*extra\_fields)

user.set\_password(password)

user.save()

return user

def create\_superuser(self, username, password, \*\*extra\_fields):

"""

Create and save a SuperUser with the given email and password.

"""

extra\_fields.setdefault('is\_staff', True)

extra\_fields.setdefault('is\_superuser', True)

extra\_fields.setdefault('is\_active', True)

if extra\_fields.get('is\_staff') is not True:

raise ValueError(\_('Superuser must have is\_staff=True.'))

if extra\_fields.get('is\_superuser') is not True:

raise ValueError(\_('Superuser must have is\_superuser=True.'))

return self.create\_user(username, password, \*\*extra\_fields)

def create\_staffuser(self, username, first\_name, last\_name, password=None):

if not username:

raise ValueError("User must have an email")

if not password:

raise ValueError("User must have a password")

if not first\_name:

raise ValueError("User must have a first name")

if not last\_name:

raise ValueError("User must have a last name")

user = self.model(

email=self.normalize\_email(username)

)

user.first\_name = first\_name

user.last\_name = last\_name

user.set\_password(password) # change password to hash

user.is\_admin = False

user.is\_staff = True

user.save(using=self.\_db)

return user

class User(AbstractUser):

""" Custom django user model """

username = models.CharField(unique=True, max\_length=50)

role = models.CharField(

max\_length=50, choices=ROLES\_CHOICES,

blank=False, null=False, default="OF",

error\_messages={'invalid': "role must be ME or CU"}

)

email = models.CharField(unique=True, max\_length=200, null=True)

objects = CustomUserManager()

def \_\_str\_\_(self):

return self.username

class Merchant(models.Model):

user = models.ForeignKey(

User, on\_delete=models.CASCADE, null=True, blank=True)

def \_\_str\_\_(self):

return str(self.user)

class Store(models.Model):

merchant = models.ForeignKey(

Merchant, on\_delete=models.SET\_NULL, blank=True, null=True)

name = models.CharField(unique=True, max\_length=200, null=True)

image = models.ImageField(null=True, blank=True)

description = models.TextField()

slug = models.SlugField(unique=True, max\_length=50,

auto\_created=True, editable=False)

# no need to use just in case

def \_\_str\_\_(self):

return self.name

@property

def imageURL(self):

try:

url = self.image.url

except:

url = ''

return url

def save(self, \*args, \*\*kwargs):

self.slug = self.name

super(Store, self).save(\*args, \*\*kwargs)

# def get\_absolute\_url(self):

# kwargs = {

# 'slug': self.slug

# }

# return reverse('stores', kwargs=kwargs)

class Customer(models.Model):

user = models.ForeignKey(

User, on\_delete=models.CASCADE, null=True, blank=True)

def \_\_str\_\_(self):

return str(self.user)

class Product(models.Model):

# when store is removed the product will be deleted too

store = models.ForeignKey(

Store, null=True, on\_delete=models.SET\_NULL, blank=True)

customer = models.ForeignKey(

Customer, on\_delete=models.SET\_NULL, blank=True, null=True)

name = models.CharField(max\_length=200, null=True)

price = models.DecimalField(max\_digits=9, decimal\_places=2)

digital = models.BooleanField(default=False, blank=False, null=True)

image = models.ImageField(null=True, blank=True, upload\_to="images/")

# rating

# desc(optinal)

def \_\_str\_\_(self):

return self.name

# here we will handel pics url so we dont get error if product not having pic

# this func will be called in store template

@property

def imageURL(self):

try:

url = self.image.url

except:

url = ''

return url

class Order(models.Model):

customer = models.ForeignKey(

Customer, on\_delete=models.SET\_NULL, blank=True, null=True)

date\_ordered = models.DateTimeField(auto\_now\_add=True)

complete = models.BooleanField(default=False, null=True, blank=False)

transaction\_id = models.CharField(max\_length=200, null=True)

def \_\_str\_\_(self):

return str(self.id)

# this func to add the cart total in out cart template with respect to quantity thanks to get\_total

@property

def get\_cart\_total(self):

# this will return all our items

orderitems = self.orderitem\_set.all()

# here we will sum all the items in our order by using get\_total to have

total = sum([item.get\_total for item in orderitems])

return total

# this func to know how many items we have in our cart with respect to quantity

@property

def get\_cart\_items(self):

orderitems = self.orderitem\_set.all()

total = sum([item.qunatity for item in orderitems])

return total

# this func will loop through the items in order and check if it's digital or no so we can ship

@property

def shipping(self):

shipping = False

orderitems = self.orderitem\_set.all()

for i in orderitems:

# this means check if it phisical item

if i.product.digital == False:

shipping = True

return shipping

class OrderItem(models.Model):

product = models.ForeignKey(

Product, on\_delete=models.SET\_NULL, blank=True, null=True)

order = models.ForeignKey(

Order, on\_delete=models.SET\_NULL, blank=True, null=True)

qunatity = models.IntegerField(default=0, null=True, blank=True)

date\_added = models.DateTimeField(auto\_now\_add=True)

# here we will calc the total price

@property

def get\_total(self):

total = self.product.price \* self.qunatity

return total

class ShippingAddress(models.Model):

customer = models.ForeignKey(

Customer, on\_delete=models.SET\_NULL, blank=True, null=True)

order = models.ForeignKey(

Order, on\_delete=models.SET\_NULL, blank=True, null=True)

address = models.CharField(max\_length=200, null=True)

city = models.CharField(max\_length=200, null=True)

state = models.CharField(max\_length=200, null=True)

zip\_code = models.CharField(max\_length=200, null=True)

date\_added = models.DateTimeField(auto\_now\_add=True)

def \_\_str\_\_(self):

return self.address

**views.py**

import datetime

import json

from django.contrib import messages

from django.contrib.auth import authenticate, login, logout

from django.http import HttpResponseRedirect, request, JsonResponse

from django.shortcuts import redirect, render

from .models import Merchant, Customer

from .forms import StoreForm, UserChangeForm, CreateUserForm, ProductForm, AddressForm

from .models import \*

from .utils import cartData, gustOrder

from .decorators import(

redirect\_authenticated, redirect\_unauthenticated, redirect\_merchant, redirect\_customer, store\_owner

)

# import to handel csrf token

# from django.views.decorators.csrf import csrf\_exempt

# Create your views here.

def store\_view(request, \*args, \*\*kwargs):

data = cartData(request)

cartItem = data['cartItem']

products = Product.objects.all()

stores = Store.objects.all()

context = {'stores': stores, 'products': products,

'cartItem': cartItem, 'shipping': False}

return render(request, 'store/store.html', context)

# @csrf\_exempt

@redirect\_merchant

def checkout\_view(request):

data = cartData(request)

cartItem = data['cartItem']

order = data['order']

items = data['items']

if request.method == "GET":

form = AddressForm()

context = {'form': form, 'items': items,

'order': order, 'cartItem': cartItem}

return render(request, 'store/checkout.html', context)

elif request.method == 'POST':

form = AddressForm(data=request.POST)

if form.is\_valid():

shipInfo = form.save()

shipInfo.order = order

shipInfo.customer = order.customer

shipInfo.save()

context = {'form': form, 'items': items,

'order': order, 'cartItem': cartItem}

return render(request, 'store/checkout.html', context)

return render(request, 'store/checkout.html')

@redirect\_merchant

def cart\_view(request):

data = cartData(request)

cartItem = data['cartItem']

order = data['order']

items = data['items']

context = {'items': items, 'order': order, 'cartItem': cartItem}

return render(request, 'store/cart.html', context)

@redirect\_merchant

@redirect\_unauthenticated

def add\_to\_cart(request, \*args, \*\*kwargs):

product\_id = kwargs['pk']

customer = Customer.objects.get\_or\_create(user=request.user)

# get\_or\_create func retrive object form order if exisit else it will create it by ur attributes

order, created = Order.objects.get\_or\_create(

customer=customer[0], complete=False)

if isinstance(product\_id, int):

product = Product.objects.get(id=product\_id)

if OrderItem.objects.all().filter(product=product).exists():

order\_item, created = OrderItem.objects.get\_or\_create(

order=order, product=product)

order\_item.qunatity += 1

order\_item.save()

else:

order\_item = OrderItem.objects.create(order=order, product=product)

order\_item.save()

return HttpResponseRedirect(request.META.get('HTTP\_REFERER'))

# @csrf\_exempt

def update\_item(request, \*args, \*\*kwargs):

white\_list = ['add', 'remove']

if not isinstance(kwargs['pk'], int):

return redirect('cart')

if kwargs['action'] not in white\_list:

return redirect('cart')

productID = kwargs['pk']

action = kwargs['action']

print('Action:', action)

print('productId:', productID)

# here we will get the logged in customer

customer = Customer.objects.get\_or\_create(user=request.user)

print(customer)

# here we will get the product that has been selected

product = Product.objects.get(id=productID)

# here we will create the order that customer choosed if exisit it will change the quantity

order, created = Order.objects.get\_or\_create(

customer=customer[0], complete=False)

orderItem, created = OrderItem.objects.get\_or\_create(

order=order, product=product)

if action == 'add':

orderItem.qunatity = (orderItem.qunatity + 1)

elif action == 'remove':

orderItem.qunatity = (orderItem.qunatity - 1)

orderItem.save()

if orderItem.qunatity <= 0:

orderItem.delete()

return HttpResponseRedirect(request.META.get('HTTP\_REFERER'))

template\_name = 'accounts/login.html'

@redirect\_authenticated

def signIn(request):

"""Handling all GET requests for Login Page, If user loged in; can't access """

if request.method == 'GET':

form = CreateUserForm()

context = {'form': form}

return render(request, template\_name, context)

if request.method == 'POST':

"""Handling all POST requests for Login Page"""

username = request.POST.get('username')

password = request.POST.get('password')

# built in func to give access for unauth users and log them in until they logout

user = authenticate(request, username=username, password=password)

print(user)

if user is not None:

# login func will check the DB and data form

login(request, user)

return redirect('store')

else:

messages.add\_message(request, messages.INFO,

"Password or username is incorrect")

return redirect('login')

else:

context = {}

return render(request, template\_name, context)

@redirect\_unauthenticated

def signOut(request):

""" Logout user, terminate csrf session then redirect to home page"""

try:

logout(request)

# appName:URLName

redirect('store')

except Exception as EX:

print(EX)

messages.add\_message(request, messages.INFO, 'Please login')

return redirect('login')

def profile(request):

return render(request, "store/profile.html", {})

@redirect\_authenticated

def register(request):

if request.method == "GET":

form = CreateUserForm()

context = {'form': form}

return render(request, "accounts/register.html", context)

if request.method == 'POST':

# post request business logic

form = CreateUserForm(data=request.POST)

if form.is\_valid():

# this return user object or none

check = User.objects.all().filter(

username=form.cleaned\_data['username'])

if check.exists():

messages.add\_message(

request, level=messages.INFO, message='form error: username exists')

return redirect('regsiter')

user = form.save()

user.is\_active = True

user.save()

if form.cleaned\_data['role'] == "ME":

Merchant.objects.create(user=user)

else:

Customer.objects.create(user=user)

return redirect('login')

else:

context = {'form': form}

messages.add\_message(

request, level=messages.INFO, message='form error')

return render(request, "accounts/register.html", context)

else:

pass

return redirect('register')

def stores(request, \*args, \*\*kwargs):

slug = kwargs['slug']

store = Store.objects.get(slug=slug)

products = Product.objects.all().filter(store=store)

data = cartData(request)

cartItem = data['cartItem']

context = {'products': products, 'store': store, 'cartItem': cartItem}

return render(request, 'store/stores.html', context)

@redirect\_unauthenticated

@redirect\_customer

@store\_owner

def add\_products(request):

if request.method == 'GET':

# give empty form

form = ProductForm()

context = {'form': form}

return render(request, 'store/addproduct.html', context)

elif request.method == 'POST':

print(request.FILES)

form = ProductForm(request.POST, request.FILES)

if form.is\_valid():

merchant = Merchant.objects.get(user=request.user)

store = Store.objects.get(merchant=merchant)

product = form.save()

product.store = store

product.save()

messages.add\_message(request, messages.INFO,

'product added')

else:

messages.add\_message(request, messages.INFO,

form.errors)

else:

# other http methods

pass

context = {'form': form}

return render(request, 'store/addproduct.html', context)

@redirect\_unauthenticated

@redirect\_customer

def add\_store(request):

if request.method == 'GET':

# give empty form

form = StoreForm()

context = {'form': form}

return render(request, 'store/add\_store.html', context)

elif request.method == 'POST':

form = StoreForm(request.POST, request.FILES)

if form.is\_valid():

merchant, \_ = Merchant.objects.get\_or\_create(user=request.user)

store = form.save()

store.merchant = merchant

store.save()

messages.add\_message(request, messages.INFO,

'store created')

else:

messages.add\_message(request, messages.INFO,

form.errors)

else:

# other http methods

pass

context = {'form': form}

return render(request, 'store/add\_store.html', context)

# paypal

def complete\_paypal\_payment(request):

print('data:', request.body)

# this will give us long char field

transaction\_id = datetime.datetime.now().timestamp()

# data for bring body data from frontEnd

data = json.load(request.body)

if request.user.is\_authenticated:

customer = request.user.customer

order, created = Order.objects.get\_or\_create(

customer=customer, complete=False)

total = float(data['form']['total'])

order.transaction\_id = transaction\_id

# here we wll check if total is same as the cart total for secuirty

if total == float(order.get\_cart\_total):

order.complete = True

order.save()

if order.shipping == True:

ShippingAddress.objects.create(

customer=customer,

order=order,

address=data['shipping']['address'],

city=data['shipping']['city'],

state=data['shipping']['state'],

zipcode=data['shipping']['zipcode'],

)

return JsonResponse('payment submitted', safe=False)

**utils.py**

import json

from django.shortcuts import redirect

#from .models import \*

from .models import Customer, Order

def cartData(request):

# here we will check and handel logged in user and not logged in

if request.user.is\_authenticated:

role = request.user.role

print(role)

user = request.user

if role == 'CU':

customer = Customer.objects.get\_or\_create(user=user)

# get\_or\_create func retrive object form order if exisit else it will create it by ur attributes

order = Order.objects.get(customer=customer[0], complete=False)

items = order.orderitem\_set.all()

# this for cart icon updata and we need to set it for all our views

cartItem = order.get\_cart\_items

else:

return {'cartItem': None, 'role': role}

role = None

return {'cartItem': cartItem, 'order': order, 'items': items, 'role': role, 'order': order}

def gustOrder(request, data):

print('user is not logged in')

print('cookies:', request.COOKIES)

name = data['form']['name']

email = data['form']['email']

cookieData = cookieCart(request)

items = cookieData['items']

# here we want to create custome for the not log in user (gust) so we can trace him by his email

customer, created = Customer.objects.get\_or\_create(email=email)

# here after we find him we set his name value

customer.name = name

customer.save()

order = Order.objects.create(customer=customer[0], complete=False)

# now we need to add items (orderItems model) to DB to be created then link it with order

for item in items:

# we need to create product and order(done above) to connect item with thim cuz our model bulid like that

product = Product.objects.get(id=item['product']['id'])

orderItem = OrderItem.objects.create(

product=product, order=order, qunatity=item['qunatity'])

return customer, order

**urls.py**

from django.urls import path

from django.urls.resolvers import URLPattern

from . import views

urlpatterns = [

path('', views.store\_view, name='store'),

path('cart/', views.cart\_view, name='cart'),

path('checkout/', views.checkout\_view, name='checkout'),

path('update\_item/<str:action>/<int:pk>',

views.update\_item, name='update\_item'),

path('login/', views.signIn, name='login'),

path('logout/', views.signOut, name='logout'),

path('profile/', views.profile, name='profile'),

path('register/', views.register, name='register'),

path('add\_to\_cart/<int:pk>', views.add\_to\_cart, name='add\_to\_cart'),

path('stores/<slug:slug>/', views.stores, name='stores'),

path('add-product/', views.add\_products, name='add-product'),

path('add\_store/', views.add\_store, name='add\_store'),

path('complete-paypal-payment', views.complete\_paypal\_payment,

name='complete\_paypal\_payment')

]

**Forms.py**

from django.db.models import fields

from django.db.models.fields import CharField

from django.forms.widgets import Widget

from .models import Merchant, ShippingAddress, User, Store, Product

from django import forms

from django.contrib.auth.forms import UserChangeForm, UserCreationForm

ROLES\_CHOICES = (

('ME', 'Merchant'),

('CU', 'Customer')

)

class AddressForm(forms.ModelForm):

address = forms.CharField(widget=forms.TextInput(

attrs={'class': 'form-control', 'placeholder': 'Enter street address'}))

city = forms.CharField(widget=forms.TextInput(

attrs={'class': 'form-control', 'placeholder': 'Enter city name'}))

state = forms.CharField(widget=forms.TextInput(

attrs={'class': 'form-control', 'placeholder': 'Enter state name'}))

zip\_code = forms.CharField(widget=forms.TextInput(

attrs={'class': 'form-control', 'placeholder': 'Enter zip code'}))

class Meta:

model = ShippingAddress

fields = ['address', 'city', 'state', 'zip\_code']

class ProductForm(forms.ModelForm):

# flieds

name = forms.CharField(widget=forms.TextInput(

attrs={'class': 'form-control', 'placeholder': 'Enter a price for your product'}))

price = forms.CharField(widget=forms.TextInput(

attrs={'class': 'form-control', 'placeholder': 'Enter a price for your product'}))

digital = forms.BooleanField(widget=forms.CheckboxInput(

attrs={'class': 'checkbox'}))

image = forms.ImageField()

class Meta:

model = Product

fields = ['name', 'price', 'digital', 'image']

class StoreForm(forms.ModelForm):

name = forms.CharField(widget=forms.TextInput(

attrs={'class': 'form-control', 'placeholder': 'Enter store name'}))

image = forms.ImageField()

description = forms.CharField(widget=forms.TextInput(

attrs={'class': 'form-control', 'placeholder': 'Enter store description'}))

class Meta:

model = Store

fields = ['name', 'image', 'description']

# UserCHangeForm --> to change user details in backend from view bulit in from django

class CustomUserChangeForm(UserChangeForm):

class Meta:

model = User

fields = ['username', 'email', 'password']

# UserCreatForm -> to create users bulit in from django

class CreateUserForm(UserCreationForm):

username = forms.CharField(widget=forms.TextInput(

attrs={'class': 'form-control', 'placeholder': 'Username'}))

first\_name = forms.CharField(widget=forms.TextInput(

attrs={'class': 'form-control', 'placeholder': 'First Name'}), max\_length=32)

last\_name = forms.CharField(widget=forms.TextInput(

attrs={'class': 'form-control', 'id': 'text-inpt', 'placeholder': 'Last Name'}), max\_length=32, )

email = forms.EmailField(widget=forms.EmailInput(

attrs={'class': 'form-control', 'placeholder': 'Email'}), max\_length=64)

password1 = forms.CharField(widget=forms.PasswordInput(

attrs={'class': 'form-control', 'placeholder': 'Password'}))

password2 = forms.CharField(widget=forms.PasswordInput(

attrs={'class': 'form-control', 'placeholder': 'Password Again'}))

role = forms.ChoiceField(choices=ROLES\_CHOICES, widget=forms.Select(

attrs={'class': 'form-control'}))

# this class is for user model, here the user will save the data in backEnd as meta class

class Meta:

model = User

fields = ['username', 'first\_name', 'last\_name',

'email', 'password1', 'password2', 'role']

**Decorators.py**

from django.http import HttpResponse

from django.shortcuts import redirect

from django.contrib.auth.models import AnonymousUser

from .models import Merchant, Store

from django.shortcuts import get\_object\_or\_404

from django.http import HttpResponseRedirect

def redirect\_authenticated(view\_func):

"""redirect unathenticated users to home view"""

def wrap(request, \*args, \*\*kwargs):

if request.user.is\_authenticated:

return redirect('store')

else:

return view\_func(request, \*args, \*\*kwargs)

return wrap

def redirect\_unauthenticated(view\_func, \*args, \*\*kwargs):

"""redirect uanthenticated users to login"""

def wrap(request, \*args, \*\*kwargs):

if not request.user.is\_authenticated:

return redirect('login')

else:

return view\_func(request, \*args, \*\*kwargs)

return wrap

def redirect\_customer(view\_func):

""" redirect clients when they try to acess officer portal"""

def wrap(request, \*args, \*\*kwargs):

if isinstance(request.user, AnonymousUser):

return view\_func(request, \*args, \*\*kwargs)

if request.user.role == 'CU':

return redirect('profile')

return view\_func(request, \*args, \*\*kwargs)

return wrap

def redirect\_merchant(view\_func):

""" redirect clients when they try to acess officer portal"""

def wrap(request, \*args, \*\*kwargs):

print(type)

if isinstance(request.user, AnonymousUser):

return view\_func(request, \*args, \*\*kwargs)

if request.user.role == 'ME':

return redirect('profile')

return view\_func(request, \*args, \*\*kwargs)

return wrap

def store\_owner(view\_func):

def wrap(request, \*args, \*\*kwargs):

merchant = Merchant.objects.get(user=request.user)

if request.META.get('HTTP\_REFERER') == None:

return redirect('store')

slug = request.META.get('HTTP\_REFERER').split(

'/')[4] # get store slug user came from

store = Store.objects.get(slug=slug)

print(store.merchant, merchant)

if str(merchant) == str(store.merchant):

return view\_func(request, \*args, \*\*kwargs)

else:

return view\_func(request, \*args, \*\*kwargs)

return wrap

**admin.py**

from django.contrib import admin

from .models import \*

# Register your models here.

admin.site.register(Customer)

admin.site.register(Product)

admin.site.register(Order)

admin.site.register(OrderItem)

admin.site.register(ShippingAddress)

admin.site.register(Merchant)

admin.site.register(Store)

admin.site.register(User)

**apps.py**

from django.apps import AppConfig

class StoreConfig(AppConfig):

default\_auto\_field = 'django.db.models.BigAutoField'

name = 'store'

**manage.py**

#!/usr/bin/env python

"""Django's command-line utility for administrative tasks."""

import os

import sys

def main():

"""Run administrative tasks."""

os.environ.setdefault('DJANGO\_SETTINGS\_MODULE', 'StoresHub.settings')

try:

from django.core.management import execute\_from\_command\_line

except ImportError as exc:

raise ImportError(

"Couldn't import Django. Are you sure it's installed and "

"available on your PYTHONPATH environment variable? Did you "

"forget to activate a virtual environment?"

) from exc

execute\_from\_command\_line(sys.argv)

if \_\_name\_\_ == '\_\_main\_\_':

main()

**sittings.py**

"""

Django settings for StoresHub project.

Generated by 'django-admin startproject' using Django 3.2.7.

For more information on this file, see

https://docs.djangoproject.com/en/3.2/topics/settings/

For the full list of settings and their values, see

https://docs.djangoproject.com/en/3.2/ref/settings/

"""

from pathlib import Path

import os

# Build paths inside the project like this: BASE\_DIR / 'subdir'.

BASE\_DIR = Path(\_\_file\_\_).resolve().parent.parent

# Quick-start development settings - unsuitable for production

# See https://docs.djangoproject.com/en/3.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!

SECRET\_KEY = 'django-insecure-a\_=nygxf&omxs^mlt!qqbw1bn%a2piuki%-h&bizn&k8effpsk'

# SECURITY WARNING: don't run with debug turned on in production!

DEBUG = True

ALLOWED\_HOSTS = []

# Application definition

INSTALLED\_APPS = [

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

# installed apps

'store',

]

MIDDLEWARE = [

'django.middleware.security.SecurityMiddleware',

'django.contrib.sessions.middleware.SessionMiddleware',

'django.middleware.common.CommonMiddleware',

'django.middleware.csrf.CsrfViewMiddleware',

'django.contrib.auth.middleware.AuthenticationMiddleware',

'django.contrib.messages.middleware.MessageMiddleware',

'django.middleware.clickjacking.XFrameOptionsMiddleware',

]

ROOT\_URLCONF = 'StoresHub.urls'

TEMPLATES = [

{

'BACKEND': 'django.template.backends.django.DjangoTemplates',

'DIRS': [os.path.join(BASE\_DIR, 'templates')],

'APP\_DIRS': True,

'OPTIONS': {

'context\_processors': [

'django.template.context\_processors.debug',

'django.template.context\_processors.request',

'django.contrib.auth.context\_processors.auth',

'django.contrib.messages.context\_processors.messages',

],

},

},

]

WSGI\_APPLICATION = 'StoresHub.wsgi.application'

# Database

# https://docs.djangoproject.com/en/3.2/ref/settings/#databases

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.sqlite3',

'NAME': BASE\_DIR / 'db.sqlite3',

}

}

# this is needed cuz we changed the default user model for django and create a custome one

AUTH\_USER\_MODEL = 'store.User'

# Password validation

# https://docs.djangoproject.com/en/3.2/ref/settings/#auth-password-validators

AUTH\_PASSWORD\_VALIDATORS = [

{

'NAME': 'django.contrib.auth.password\_validation.UserAttributeSimilarityValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.MinimumLengthValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.CommonPasswordValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.NumericPasswordValidator',

},

]

# Internationalization

# https://docs.djangoproject.com/en/3.2/topics/i18n/

LANGUAGE\_CODE = 'en-us'

TIME\_ZONE = 'UTC'

USE\_I18N = True

USE\_L10N = True

USE\_TZ = True

# Static files (CSS, JavaScript, Images)

# https://docs.djangoproject.com/en/3.2/howto/static-files/

STATIC\_URL = '/static/'

STATICFILES\_DIRS = [

os.path.join(BASE\_DIR, 'static')

]

MEDIA\_URL = '/images/'

MEDIA\_ROOT = os.path.join(BASE\_DIR,'static/images')

ALLOWED\_HOSTS = ['127.0.0.1']

# Default primary key field type

# https://docs.djangoproject.com/en/3.2/ref/settings/#default-auto-field

DEFAULT\_AUTO\_FIELD = 'django.db.models.BigAutoField'

# secure ssl redirect ( DEPLOYMENT ONLY SET TRUE)

SECURE\_SSL\_REDIRECT = False

# secure cookies to make it diffrecult to traffic sinffers to hijack users sessions

SESSION\_COOKIE\_SECURE = True

# CSRF cookie makes it more difficult for network traffic sniffers to steal the CSRF token

CSRF\_COOKIE\_SECURE = True

**StoresHub/urls.py**

**"""StoresHub URL Configuration**

**The `urlpatterns` list routes URLs to views. For more information please see:**

**https://docs.djangoproject.com/en/3.2/topics/http/urls/**

**Examples:**

**Function views**

**1. Add an import: from my\_app import views**

**2. Add a URL to urlpatterns: path('', views.home, name='home')**

**Class-based views**

**1. Add an import: from other\_app.views import Home**

**2. Add a URL to urlpatterns: path('', Home.as\_view(), name='home')**

**Including another URLconf**

**1. Import the include() function: from django.urls import include, path**

**2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))**

**"""**

**from django.contrib import admin**

**from django.urls import path, include**

**from django.conf.urls.static import static**

**from django.conf import settings**

**urlpatterns = [**

**path('admin/', admin.site.urls),**

**path('', include('store.urls')),**

**]**

**urlpatterns += static(settings.MEDIA\_URL, document\_root=settings.MEDIA\_ROOT)**

### References:

[1] Kaura, A. (2020, October 15). *E-commerce Web Application: Why Your Business Needs One?* Seasia Infotech Blog. https://www.seasiainfotech.com/blog/e-commerce-web-application-business-needs-one/

[2] Editorial Team. (2021, January 14). *What Is An E-Commerce Website & How To Build One*. CyberChimps. <https://cyberchimps.com/e-commerce-websites/>

[3] "From History of Web Application Development", *Devsaran*, 2021. [Online]. Available: https://www.devsaran.com/blog/history-web-application-development. [Accessed: 11- Mar- 2021].

[4]"Web Application vs. Website", *Vibethink*, 2021. [Online]. Available: https://vibethink.com/web-application-vs-website/. [Accessed: 14- Mar- 2021].

[5] "Web App development for Data Scientists", *Medium*, 2021. [Online]. Available: https://towardsdatascience.com/web-app-development-for-data-scientists-48e445e54d7b. [Accessed: 15- Mar- 2021].

[6] "Medium", *Medium*, 2021. [Online]. Available: https://medium.com/@essentialdesign/website-vs-web-app-whats-the-difference-e499b18b60b4. [Accessed: 10- Mar- 2021].

[7] "Difference between Website and Web Application", *Guru99.com*, 2021. [Online]. Available: https://www.guru99.com/difference-web-application-website.html#:~:text=Summary%3A,you%20in%20branding%20your%20business. [Accessed: 11- Mar- 2021].

[8] IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.

[9] J. Samuel, Z. Zheng and Y. Xie, "Value of Local Showrooms to Online Competitors", *MIS Quarterly*, vol. 44, no. 3, pp. 1073-1106, 2020. Available: 10.25300/misq/2020/15291.

[10] C. Lofi, W. Balke and U. Guntzer, "Eliciting Customer Wishes Using Example-Based Heuristics in E-Commerce Applications," 2011 IEEE 13th Conference on Commerce and Enterprise Computing, Luxembourg-Kirchberg, Luxembourg, 2011, pp. 74-81, doi: 10.1109/CEC.2011.26.

[11] Samuel, J., Zhiqiang (Eric) Zheng and Ying Xie (2020) ‘Value of Local Showrooms to Online Competitors’, MIS Quarterly, 44(3), pp. 1073–1106. Available at: http://search.ebscohost.com.sdl.idm.oclc.org/login.aspx?direct=true&db=bsu&AN=145446039&site=eds-live (Accessed: 11 April 2021)

[12] Wienclaw, R. A. (2020) ‘E-Commerce’, Research Starters: Business. Available at: http://search.ebscohost.com.sdl.idm.oclc.org/login.aspx?direct=true&db=ers&AN=89163657&site=eds-live (Accessed: 11 April 2021).

[13] Yadav, D. et al. (2018) ‘Vulnerabilities and Security of Web Applications’, 2018 4th International Conference on Computing Communication and Automation (ICCCA), Computing Communication and Automation (ICCCA), 2018 4th International Conference on, pp. 1–5. doi: 10.1109/CCAA.2018.8777558.

[14] Chatley, R. et al. (2003) ‘Visual methods for Web application design’, IEEE Symposium on Human Centric Computing Languages and Environments, 2003. Proceedings. 2003, Human Centric Computing Languages and Environments, 2003. Proceedings. 2003 IEEE Symposium on, Human centric computing languages and environments, pp. 242–244. doi: 10.1109/HCC.2003.1260236.

[15] Kiruthika, J. et al. (2016) ‘User Experience Design in Web Applications’, 2016 IEEE Intl Conference on Computational Science and Engineering (CSE) and IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC) and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering (DCABES), Computational Science and Engineering (CSE) and IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC) and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering (DCABES), 2016 IEEE Intl Conference on, CSE-EUC-DCABES, pp. 642–646. doi: 10.1109/CSE-EUC-DCABES.2016.253.