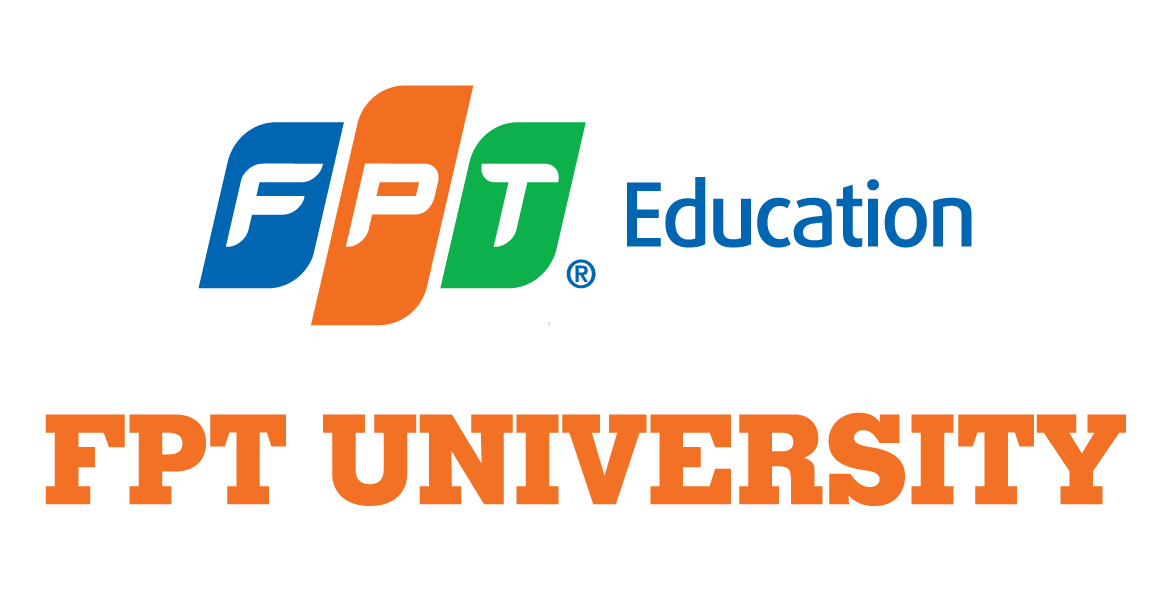
****

**Build a Exotic pet shop website**

**Java Web Appication Development(PRJ301)**

**Version 1.0**

**TABLE OF CONTENTS**

[**1. DOCUMENT HISTORY**](#_xmjifl5d3i7r) **3**

[**2. REFERENCE DOCUMENTS**](#_azx1kgokijmm) **4**

[**3. DISTRIBUTION LIST AND APPROVALS**](#_suzf588yi8g2) **4**

[**4. INTRODUCTION**](#_uya1c8oxq4xj) **5**

[4.1.Purpose](#_q8vub5i07glj) 5

[4.2.In scope](#_tp79dyu54kpj) 5

[**5. OVERVIEW**](#_veqze55hi4kr) **6**

[5.1. Actors](#_7f5szcz55ve3) 6

[5.2. System Use Case Diagram](#_c1ead98zhmv) 8

[5.2.1 Admin](#_sr9521ouzhb) 8

[5.2.1.1 Account login](#_frcjnzlu1hxk) 8

[5.2.1.2 Income Management](#_se7hpw8qymvz) 9

[5.2.1.3 Product Management](#_xrlwyi11wwg4) 9

[5.2.1.4 Article Management](#_q4j96scg285u) 10

[5.2.2 Customer](#_tzp9s56sb9nd) 11

[**6. FUNCTIONAL DESCRIPTION**](#_p8hawzffjz96) **12**

[6.1. Admin](#_izyyn2xdhhfh) 12

[6.1.1. Account Login](#_hc6wivhas2c5) 12

[6.1.2. Income Management](#_q514ihh9tmpt) 14

[6.1.3. Product Management](#_y39lf93ax9yk) 15

[6.1.4. Article management](#_xofbxv79tjl7) 16

[6.2 Customer](#_w54iw5s8yeqm) 17

[6.2.1. Registration & Account Login](#_2iwrb2hvy6m0) 17

[6.2.2. View Personal Information](#_hyyc38u1d37) 18

[6.2.3. View products](#_gw1jh0izsbcf) 18

[6.2.4.Add to cart](#_9arkxe87qpul) 19

[6.2.5 Edit shipping address](#_d0wfugermh41) 20

[6.2.6 Reading newspapers](#_cbp1gx17c174) 21

[**7. EFFECTIVE APPLICATION FUNCTIONS AND ALGORITHMS**](#_hi6au5y61rg) **22**

[7.1 Send Mail using Javax.mail and Activation](#_xwa5v8imw7xv) 22

[7.2 Use Hash Security](#_3hw27pqqstnb) 24

[7.3 Use JSON to send objects](#_81wumd34xt8n) 25

[7.4 Using AI chatbots for customer care](#_2xtwpxaft2gm) 26

[7.5 Linechart](#_i3gukihzlpbh) 26

[**8. NON-FUNCTIONAL REQUIREMENTS AND OTHERS**](#_u4e45u4cbvkc) **29**

[8.1. Performance](#_dv91yht4bpkw) 29

[8.2. Scalability](#_w77emv6ps20) 29

[8.3. Security](#_at49y4kaj622) 29

[8.4. Browser](#_vw5q4tuqagsg) 30

[8.5. Reliability](#_f9y9iqk7mxs9) 30

[8.6. Interfaces](#_dhhkb3j5o4y4) 30

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# **1.** **DOCUMENT HISTORY**

| **Date** | **Summary of Changes** | **Version** |
| --- | --- | --- |
| 5-2-2023 | - Draw a general use case diagram and assign work.  - Describe the framework of the report and outline some of the functions. | 1.0 |
| 11-2-2023 | -Draw mockup for User (main) page | 1.0 |
| 25-2-2023 | -Draw mockups for Admin pages | 1.0 |
| 12-3-2023 | - Edit and finalize mockups.  - Make reports. | 1.0 |
| 19-3-2023 | - Edit and finalize the report | 1.0 |

# **2.** **REFERENCE DOCUMENTS**

| **Document Name** | **Description** |
| --- | --- |
| R02\_SRS\_v1.0-1.docx | Is the teacher's document PRJ301. |

# **3.** **DISTRIBUTION LIST AND APPROVALS**

| **Name** | **Title** |
| --- | --- |
| Nguyễn Anh Việt | * Create general interface * Create a database to store information, query data * Participate in interface design, information search, and code editing |
| Mai Văn Tâm | * Design interface and use case for User, Guest * Interface design and use case for Admin * Create AI chatbot for customer |
| Nguyễn Hồng Lĩnh | * Create interface and use case for User, Guest * Query information, retrieve data from the database * Create user interaction methods |
| Nguyễn Duy Khánh | * Create an interface, use case for Admin * Create interface and use case for User, Guest * Participate in interface design, information search |
| Nguyễn Hồ Ngọc Ấn | * Create information security * Create an interface, use case for Admin * Create interface and use case for User, Guest |

# **4.** **INTRODUCTION**

## 4.1.Purpose

This document is intended to describe in detail the functions of the "Build an Exotic pet shop" project - by illustrating the functions in detail and building a complete system, the document will clearly explain the constraints in the system, an overview of the interface and interactions with the system.

This document also explains in detail the goals and final results to be achieved of the project, helping the reader better understand the working methods and quality control process throughout the implementation process.

## 4.2.In scope

Shop IRCN V specializes in the sale of pets called exotic pets and has implemented a Web-based sales and management system. This is a solution designed to assist sellers and customers in buying and selling goods conveniently and quickly, while minimizing costs.

By using this system, IRCN V shop hopes to replace or supplement traditional sales channels, thereby creating convenience and enhancing the user experience.

The system serves two main users:

* Admin
* Customer (user)

# **5.** **OVERVIEW**

## **5.1.** **Actors**

IRCN V is a sales website with two main objects: Admin and Customer.

Admin: As the site manager IRCN V. Admin can manage the website's income, develop advertising campaigns to attract new customers, manage products, posts, update product information new, processing orders and payments. In addition, Admin can also create and modify information related to product types, prices, product images and product descriptions.

Customers: The main object of IRCN V website. Customers can log in to the website to view product information and place orders. Customers can select products to buy, adjust quantities and locations to ship, read product-related reports or other website news.

With the analysis of the audience using the IRCN V website, the website will be designed and optimized to suit the needs and purposes of the customer, and at the same time help the Admin manage and develop the website. more effective.

| **Actor** | **Description** |
| --- | --- |
| Admin | * Have the rights to create, view, edit and delete products. * Have the right to add and delete articles. * Have the right to manage income. * Have access to order management. |
| Customer | * Sign in to the website * View product information * Customers can choose the product to buy * Adjust the number and location you want to ship * Read more news of the site |

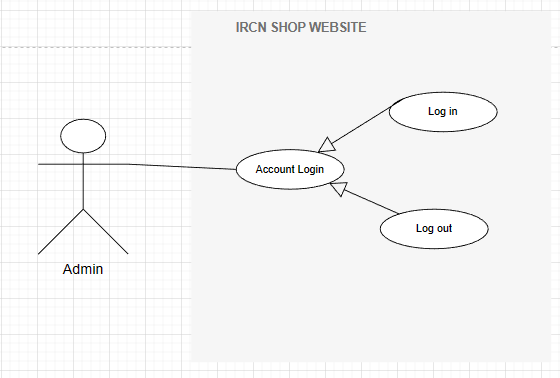
# 

## ***5.2.*** ***System Use Case Diagram***

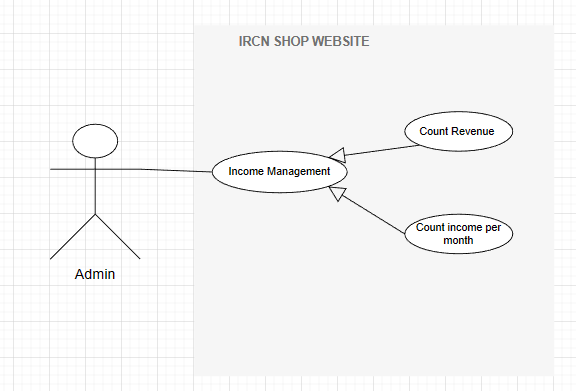
The following diagram provides an overview of the functional groups and actors involved in the web system:

### 5.2.1 Admin

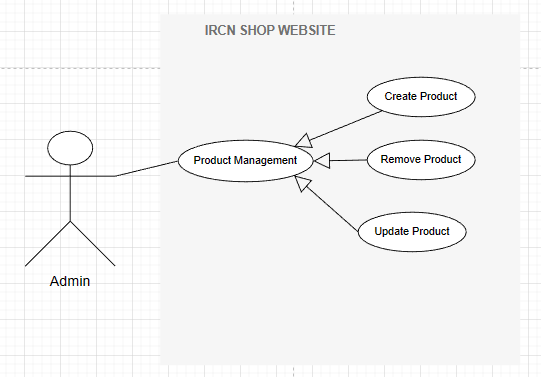
#### 5.2.1.1 Account login



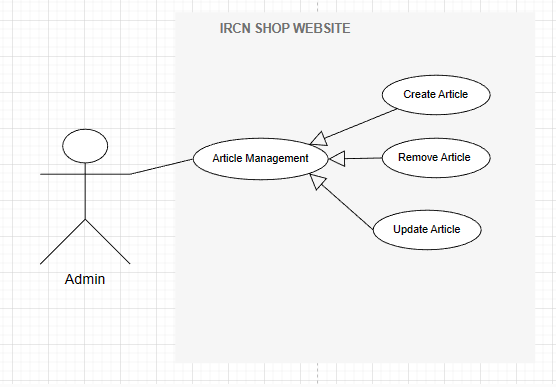
#### 5.2.1.2 Income Management



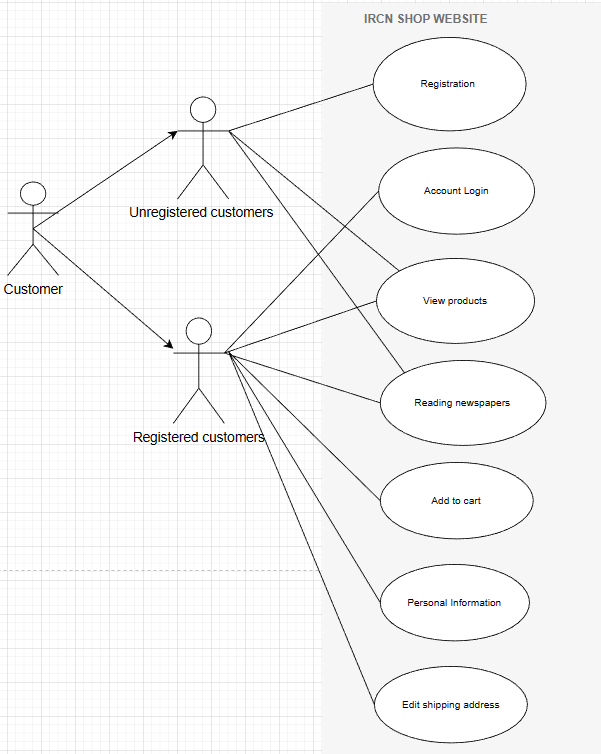
#### 5.2.1.3 Product Management



#### 5.2.1.4 Article Management



### 5.2.2 Customer



# **6.** **FUNCTIONAL DESCRIPTION**

## **6.1.** **Admin**

### 6.1.1. Account Login

| **Screen** | Log in | | | |
| --- | --- | --- | --- | --- |
| **Description** | | Allow admin to login to the system | | |
| **Screen Access** | | **Admin selects Login at the Home screen.** | | |
| **Screen Content** | | | | |
| **Item** | | **Type** | **Data** | **Description** |
| User | | Email– String(50) |  | This field is for admin to enter the account |
| Email | | Password – String(100) |  | This field is for the admin to enter the password. |
| Login | | Button |  | Log in to the system |
| **Screen Actions** | | | | |
| **Action Name** | | **Description** | **Success** | **Failure** |
| Login | | When the admin presses the login button, the system will check the validity of the data, if the data is not valid, it will display a message to re-enter, if the data is valid, continue to check the Account with the corresponding password. exists in the system or not, if so, let the admin login to the system, if not, display a message and add a Remember me button to save the password for the next session.  ”. | Go to Amin's interface to manage the page. | Displays a message of fields that are required to be entered but are left blank. If the password of ADMIN or the customer is incorrect, it will be re-entered. |

### 6.1.2. Income Management

| Use Case Name | IcomeManagement |
| --- | --- |
| Use Case ID | UC02 |
| High Level Requirement Ref | Allow admin to statistics income by month and all from start. |
| Actor | Admin |
| Description | All price information of the bill to be paid is queried from the database and displayed on the screen. |
| Trigger | NA |
| Precondition | Admin click on Dash board in the sidebar menu. |
| Post-processing |  |

## 

### 6.1.3. Product Management

| Use Case Name | Product Management |
| --- | --- |
| Use Case ID | UC03 |
| High Level Requirement Ref | Allows admin to manage product information, delete, view details, and display a product list. |
| Actor | Admin |
| Description | All Product information is queried from the database and displayed on the screen. Admin can delete, view details, and display a product list. |
| Trigger | NA |
| Precondition | Admin clicks Product in the sidebar menu. |
| Post-processing |  |

### 6.1.4. Article management

| Use Case Name | Article management |
| --- | --- |
| Use Case ID | UC04 |
| High Level Requirement Ref | Allow admin to manage Post's information, delete, view details, and display Posts list. |
| Actor | Admin |
| Description | All Post information is queried from the database and displayed on the screen. Admin can delete, view details, and display the Posts list. |
| Trigger | NA |
| Precondition | Admin click on Manage Posts in the menu. |
| Post-processing |  |

## 6.2 Customer

### 6.2.1. Registration & Account Login

| Use Case Name | Registration & Account Login |
| --- | --- |
| Use Case ID | UC05 |
| High Level Requirement Ref | Allow customers to log in to the site. |
| Actor | Customer |
| Description | For customers who have not registered an account, there will be a place to register a new account. As for registered customers, they will log in, and if correct, from the database, they will enter the user interface. |
| Trigger | NA |

### 6.2.2. View Personal Information

| Use Case Name | View Personal Information |
| --- | --- |
| Use Case ID | UC06 |
| High Level Requirement Ref | Allow guests to view personal information. |
| Actor | Customer |
| Description | Registered customers will be able to track the personal information entered when registering, this is also the information for the shop to get when closing the order. |
| Trigger | NA |

### 6.2.3. View products

| Use Case Name | **View products** |
| --- | --- |
| Use Case ID | UC07 |
| High Level Requirement Ref | Allow customers to log in to the site. |
| Actor | Customer |
| Description | When clicking on the species, a list of products will appear that you want to see the price and detailed information, then click to go to the product page. |
| Trigger | NA |

### 6.2.4.**Add to cart**

| Use Case Name | Registration & Account Login |
| --- | --- |
| Use Case ID | UC08 |
| High Level Requirement Ref | Allow customers to log in to the website to add the products they like to their shopping cart. |
| Actor | Customer |
| Description | When customers click on the image of a product, they will have a choice of the quantity and add to cart button. |
| Trigger | NA |

### 6.2.5 **Edit shipping address**

| Use Case Name | Edit shipping address |
| --- | --- |
| Use Case ID | UC09 |
| High Level Requirement Ref | Allow customers to edit ship locations. |
| Actor | Customer |
| Description | Customers can edit the delivery location if they want in the cart section. |
| Trigger | NA |

### 6.2.6 **Reading newspapers**

| Use Case Name | Reading newspapers |
| --- | --- |
| Use Case ID | UC10 |
| High Level Requirement Ref | Allow customers to read news. |
| Actor | Customer |
| Description | Even if you are not logged in, clicking the NEWS section will switch to the newspaper section. Selecting the article will display the full article you want to read. |
| Trigger | NA |

# **7**. **EFFECTIVE APPLICATION FUNCTIONS AND ALGORITHMS**

## 7.1 Send Mail using Javax.mail and Activation

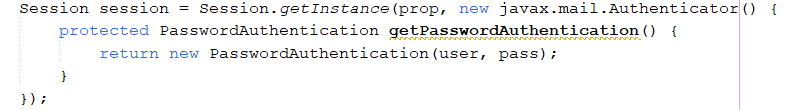
Using the libraries: activation.jar and javax.mail-1.6.2.jar javax.mail: is a Java library used to send and receive email in Java applications.

Javax.mail is a library that provides classes and methods for creating and managing email messages, connecting to email servers, user authentication, and email information encryption.

activation.jar: is one of the libraries of Java Mail API, which is used to support authentication features and manage attachments. This library provides necessary classes and interfaces for creating and managing MIME (Multipurpose Internet Mail Extensions) objects and attachments in Java.

To be able to send mail using the javax.mail library, we need to set some properties for sending email using the Simple Mail Transfer Protocol (SMTP) protocol in JavaMail API. prop.put("mail.smtp.host", "smtp.gmail.com");

* The first line sets the SMTP server to "smtp.gmail.com". prop.put("mail.smtp.port", "465");
* The second line sets the port number to "465". prop.put("mail.smtp.auth", "true");
* The third line specifies that authentication is required for the SMTP server. prop.put("mail.smtp.starttls.enable", "true");
* The fourth line activates Transport Layer Security (TLS) encryption when connecting to the SMTP server. prop.put("mail.smtp.starttls.required", "true");
* The fifth line specifies that a TLS connection is required for the SMTP server. prop.put("mail.smtp.ssl.protocols", "TLSv1.2");
* The sixth line sets the SSL protocol to TLSv1.2. prop.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");
* The seventh line specifies the socket factory class to use, which is javax.net.ssl.SSLSocketFactory. prop.put("mail.smtp.starttls.required", "true");
* The eighth line repeats that a TLS connection is required. prop.put("mail.smtp.ssl.protocols", "TLSv1.2");
* The ninth line repeats that the SSL protocol used is TLSv1.2.



* The parameter passed to props is a Properties object that allows saving mail server configuration information, ports, transport protocol.
* PasswordAuthentication is an object with 2 properties userName and password to store login information.
* Authenticator is an abstract class in this case we implement getPasswordAuthentication method to return login information.



* Create a MimeMessage object from session.
* Set the sender's email address using the setFrom() method.
* Set the email addresses of multiple recipients using the setRecipient() method.
* Set up the subject using the setSubject() method.
* Set email content using setText() method.
* Call Transport.send() to send the email.
* If there is an error, print the stack trace using the printStackTrace() method.

## 7.2 Use Hash Security

Using the MessageDigest class in java.security to perform SHA-256 hash encoding. The getSHA() method takes a string input and returns a byte array representing the SHA-256 hash value of that string. First, we create a MessageDigest object using the static getInstance() method of the MessageDigest class, passing in the hashing algorithm as "SHA-256".

Next, we use the digest() method of the MessageDigest object to compute the hash value for the input string passed into the method. Finally, we return the hash value as a byte array. The toHexString() method takes a byte array input and returns a hexadecimal string representation of the byte array's value.

First, we create a BigInteger object from the input byte array using the constructor of the BigInteger class with parameters of 1 (to indicate that this is a positive number) and the input byte array.

Next, we use the toString(16) method of the BigInteger object to convert that value to a hexadecimal string.

Then, we use a while loop to ensure that the resulting hexadecimal string has a length of 64 characters (corresponding to the 256 bits of the SHA-256 hash value). If the length of the hexadecimal string is less than 64, we prepend '0' characters to the beginning of the string so that it has a length of 64.

Finally, we return the processed hexadecimal string.

## 7.3 Use JSON to send objects

JSON is used to send data to the server through an HTTP POST request. JavaScript creates an object (data) with properties containing the data that needs to be sent to the server. This data is retrieved from HTML tags and collected using a for loop if there is multiple data in a group. Then, the JSON.stringify() method is used to convert the object (data) into a JSON string.

This is necessary because the HTTP request can only pass text data, so the data needs to be standardized into a string format before being passed. Finally, the line xhr.setRequestHeader('Content-Type', 'application/json') sets the Content-Type of the HTTP request to application/json, indicating that the data is being sent as JSON.

When the xhr.send() method is called, the standardized JSON data will be sent from the HTTP POST request to the server. Then, the server can re-standardize the JSON data and process it as needed. xhr stands for XMLHttpRequest. It is a JavaScript object that allows the client-side to make HTTP requests to the server without having to reload the page.

## 7.4 Using AI chatbots for customer care

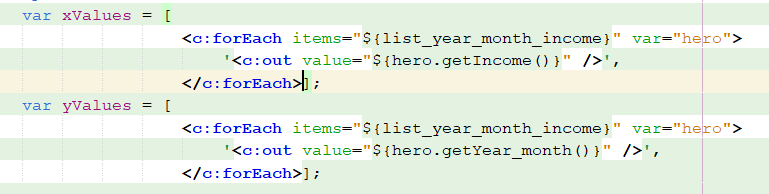
IRCN V can use Typebot.io to create an intelligent chatbot that can address common questions about their products and services. With the chatbot, IRCN V can generate automated keywords and responses for frequently asked customer queries such as pricing, return policies, product information, and more. Customers can interact directly with the chatbot through online chat channels like Messenger and email to receive quick and efficient support. By doing so, IRCN V can optimize the customer experience and minimize its response time to customers. This helps IRCN V to focus on building better relationships with its customers and enhance their interaction on the website.

## 7.5 Linechart

To draw the graph we use the javascript library "https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.5.0/Chart.min.js"

The charts that Chart.js supports include: line, bar, pie, pie, pie chart, radar chart, and innovation chart. This library also gives users many options to customize and configure their charts, from colors to the position of the axes.

First, we use <c:foreach> to put the values into the xValues and yValues arrays corresponding to the horizontal and vertical axes.



Then initialize the chart and put the parameters in

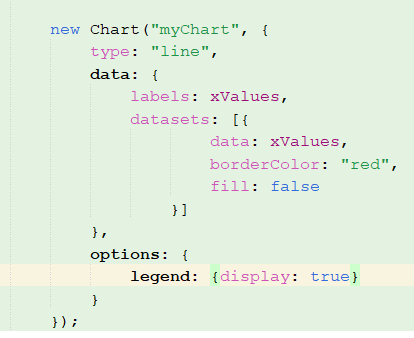
* first parameter: ID of the chart



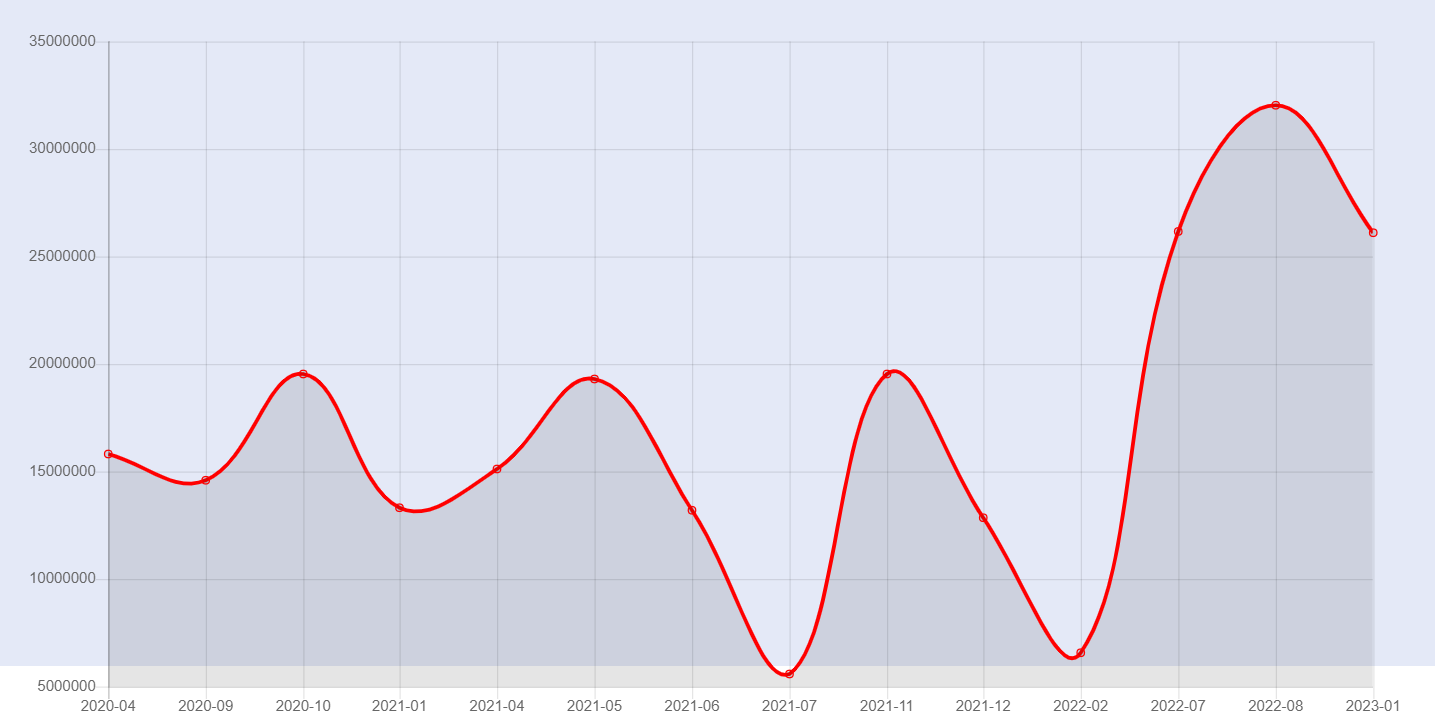
+ The second is chart type such as line, bar, …

+ 3rd is the data including yValues vertical axis data, color, fill will fill the color below the chart, fill the space below

+ 4th is an option to help you choose whether to add a toggle button or not



and this is the result:



# **8.** **NON-FUNCTIONAL REQUIREMENTS AND OTHERS**

## 8.1. Performance

| **No.** | **Requirement** |
| --- | --- |
| 1. | Fast response time |
| 2. | Intuitive, easy-to-use, user-friendly interactions |
| 3. | Suitable for a variety of screen sizes |

## 8.2. Scalability

| **No.** | **Requirement** |
| --- | --- |
| 1. | Large data storage capacity |
| 2. | Allows multiple users to perform the same operations in the same amount of time |

## 8.3. Security

| **No.** | **Requirement** |
| --- | --- |
| 1. | Use Hash Code for Password |
| 2. | Send an email to register for an account to avoid spoofing |
| 3. | Use Session & Cookies to save passwords |

# 

## 8.4. Browser

| **No.** | **Requirement** |
| --- | --- |
| 1. | Microsoft Edge and IE the latest. |
| 2. | Chromium, Firefox and Opera |

# 

## 8.5. Reliability

| **No.** | **Requirement** |
| --- | --- |
| 1. | Please contact the information on the website if you encounter any issues. |

## 8.6. Interfaces

| **No.** | **Requirement** |
| --- | --- |
| 1. | Using Bootstrap 5 library and Jquery to create the interface |