

DOCUMENTATION

DESCRIPTION

The scope of this web application is to help old people to purchase groceries from their home. The groceries would be delivered to the veteran's home without any additional delivery charges by volunteers who wish to help the veterans.

FUNCTIONALITIES

The basic functionalities are that the veteran should be able to register himself, which could be done only if the age of the veteran is above 50. The veteran must provide a legit mail address and then set a password for logging in. So once the veteran has registered himself, he should be able to make orders from the groceries available. So the order of the veteran will only be confirmed if the veteran does not have a pending order which is yet to be confirmed by any of the volunteers. The orders made, their status and the volunteer who has signed up to deliver the order will be visible in a table which could be accessed from the main page. The volunteers could register themselves irrespective of their age, and then they could sign in using their credentials and they can view all of the orders and their respective locations marked in a map. So now the volunteer may be able to select orders which he is willing to deliver. The volunteer can take up multiple orders at the same time.

TECH STACK

C# .NET

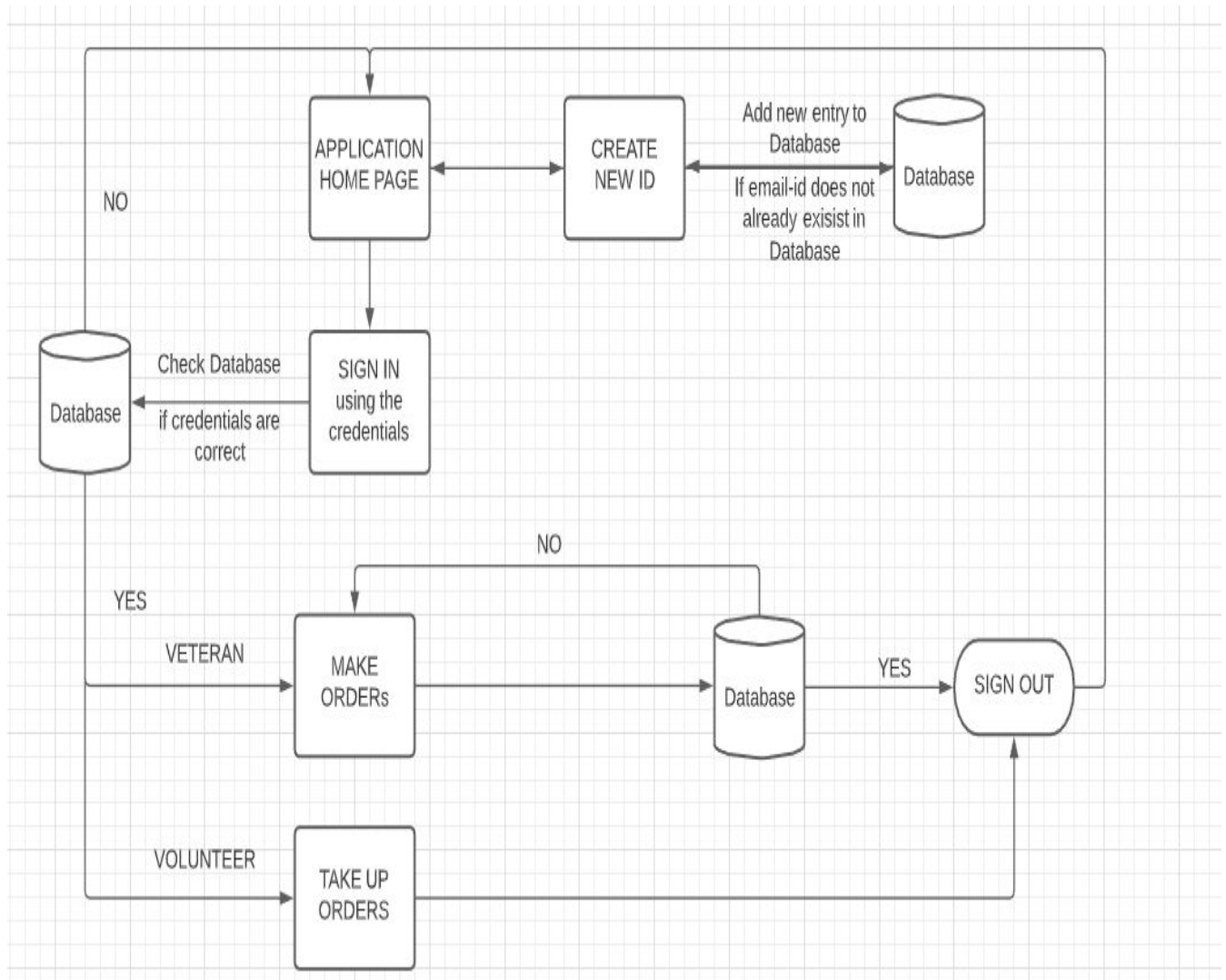
REST API

SQL SERVER DATABASE

HTML/CSS

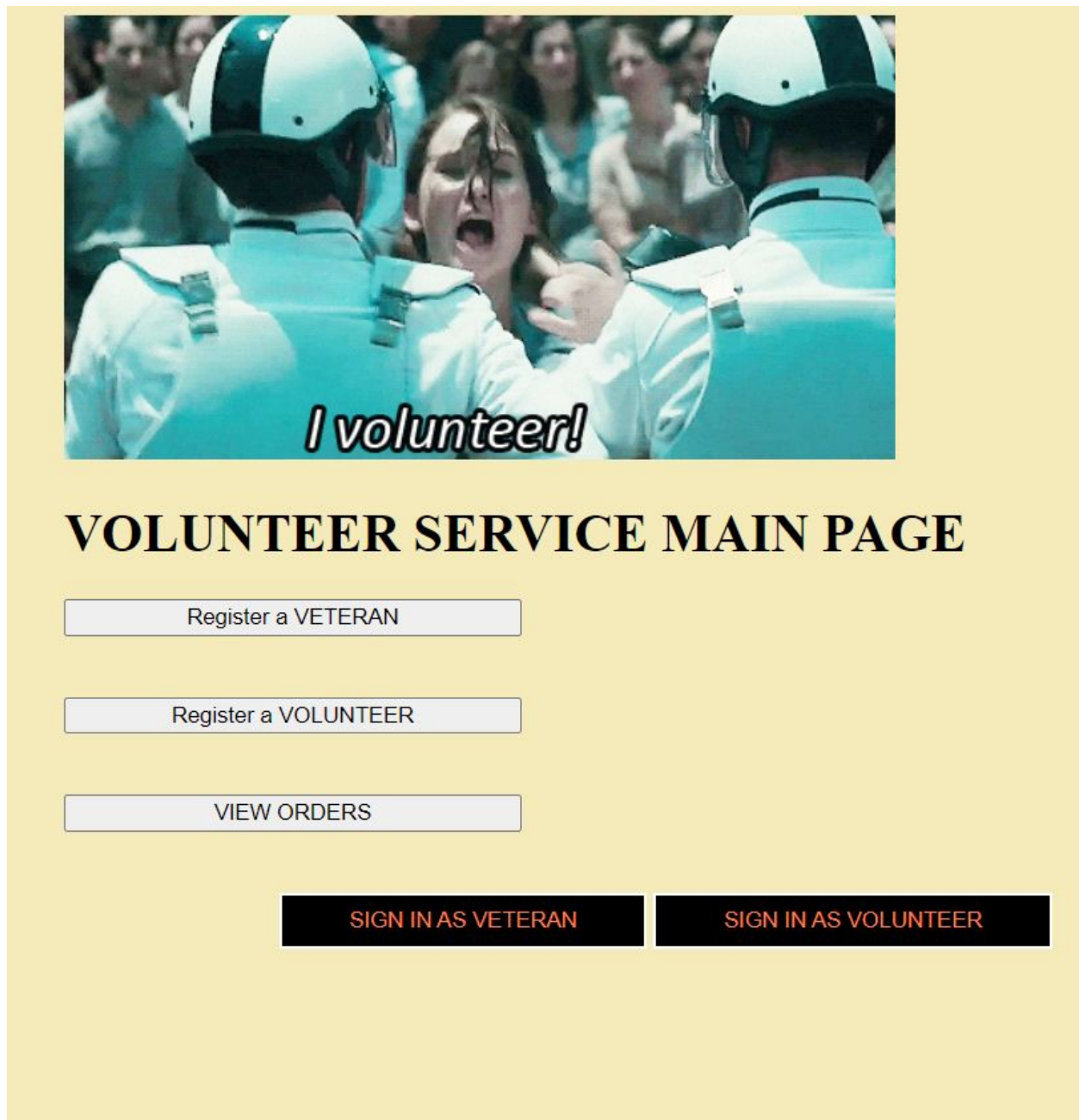
JAVASCRIPT

PROCESS MODEL



So this process flow diagram shows how the veterans and the volunteers try to create their accounts and log in using their credentials. And then they try to make and take up orders. So the REST API interacts and gets data from the Database to move further. So the flow of both the outcomes of positive and negative has been clearly illustrated in the diagram.

THE HOME PAGE - UI



So the Homepage has the options to Sign IN as a Veteran and Sign IN as a Volunteer on the Bottom right corner of the page. The options to register a veteran and a volunteer and the option to view the ORDERS which are entered in the database appears at the center of the page. A gif at the top of the page is displayed to show the dedication of our volunteers.

REGISTERING A VETERAN

The details which are to be entered to register a veteran are the name , age , address , location on the map which can be loaded from the address and vice versa. So the veteran must have a legit email-id which is checked by using regular expressions. And the user must enter a password for himself. So using all these data , a JSON Object will be created and a POST request will be called from <https://localhost:44335/api/Veteran> , So next the REST API will look upon the VeteranHelp database which is currently running on the localhost. So the API will look upon the database to find if any instances of the mail id are present in the database, and only if there is no entry in the database related with this mail id , a veteran account can be created. And if a new veteran id is created , the API returns a boolean value of true and if the mail id already exists in the database the API returns the boolean false. So if the value received is true , then we display that the registration has been successful and now the veteran could log in using the credentials. If the returned value is false , we display an error message that the mail id already exists in the database.

If value returned is false

This page says

THE MAIL ID ALREADY EXISTS IN DATABASE

OK

If value returned is true

REGISTRATION DONE

Congrats now you can sign in with mail and password

RETURN TO HOMEPAGE

REGISTERING A VOLUNTEER

The details which are to be entered to register a veteran are the name , age , address , location on the map which can be loaded from the address and vice versa. So the veteran must have a legit email-id which is checked by using regular expressions. And the user must enter a password for himself. So using all these data , a JSON Object will be created and a POST request will be called from <https://localhost:44335/api/Volunteer> , So next the REST API will look upon the VeteranHelp database which is currently running on the localhost. So the API will look upon the database to find if any instances of the mail id are present in the database, and only if there is no entry in the database related with this mail id , a veteran account can be created. And if a new veteran id is created , the API returns a boolean value of true and if the mail id already exists in the database the API returns the boolean false. So if the value received is true , then we display that the registration has been successful and now the veteran could log in using the credentials. If the returned value is false , we display an error message that the mail id already exists in the database.

If value returned is false

This page says

THE MAIL ID ALREADY EXISTS IN DATABASE

OK

If value returned is true

REGISTRATION DONE

Congrats now you can sign in with mail and password

RETURN TO HOMEPAGE

VIEW ORDERS

The view orders button which is available on the main page shows the orders which have been made. They also display the status of the order and the volunteer who has agreed to take up the order. When this button is clicked a GET request from <https://localhost:44335/api/Veteran/> So now the API uses the SQLClient to connect to the Veteranhelp database. And we get all the values from the ORDER_TABLE_VETERAN, put together as a List of Orders and sent back to the function. So now we are going to create rows with the values from the API and populate the table.

ORDERS LIST

ITEMS	PRICE	VETERAN MAKING ORDER	ORDER ID	STATUS	VOLUNTEER ACCEPTING ORDER
Croissants- 2,Carrots- 2	24	sherwin@hams.com	10001	ACCEPTED	sherwin@hello.com
Banana- 4,Tissues- 1	7.2	sherwin@hamss.com	10002	ACCEPTED	sherwin@hello.com

[HOME](#)

Here we can find the items which have been purchased by the veteran , the total cost of the order , the order-id along with the Volunteer who has decided to pick up this order.

SIGN IN AS VETERAN

So when this button is clicked , it takes us to the Veteran Login Page where we would be asked for our mail address and the Password associated with it. So when the submit button is clicked a JSON structure is created with the name and the password. We then call the POST request from `https://localhost:44335/api/Credentials`.

Then the API would check up the database Veterenhelp to check if the credentials are correct. So if the mail address is correct we would be redirected to the MAKE ORDERS page. Otherwise there would be an alert message that the credentials entered are invalid.

If the entered credentials are wrong

This page says

INVALID CREDENTIALS

OK

If the credentials are correct , then they take to the Veteran Order page.

VETERAN ORDER PAGE



Carrot \$5



Banana \$1.3

MAKING ORDERS

So on reaching the Veteran Order page, we would be able to view a number of products. The products will have an image, a price and the quantity in which you would like to make the order. Then you just have to select the quantity of the products which you would like to buy. Then once we click the Submit Order button, we create two Arrays to show what are the items which have been selected and the other array shows the quantity of items which have been ordered. So we then call the POST from <https://localhost:44335/api/Items>, and then the API creates the list with the items bought, the total price is calculated there and then we check the database if any order is already made by the veteran mail which is still not accepted. So if there is no such instance, then we enter the order details into the ORDER_TABLE_VETERAN. Then we send the response with the Orderno and the price of the Order along with a flag to say if the order is legit or not. So if the flag is 1, then we display the order details to the veteran. Otherwise we just show an alert to wait till previous orders are completed.

If the flag is true

ORDER HAS BEEN ADDED

The Order has been confirmed, Order number = 10002, Price = \$16.3

[RETURN TO HOMEPAGE](#)

If the flag is false

This page says

WAIT FOR PREVIOUS ORDER TO BE COMPLETED TO MAKE NEXT ORDER

OK

SIGN IN AS VOLUNTEER

So when this button is clicked , it takes us to the Volunteer Login Page where we would be asked for our mail address and the Password associated with it. So when the submit button is clicked a JSON structure is created with the name and the password. We then call the POST request from <https://localhost:44335/api/Credentials>.

Then the API would check up the database Veterenhelp to check if the credentials are correct. So if the mail address is correct we would be redirected to the VOLUNTEER ORDERS SELECT PAGE page. Otherwise there would be an alert message that the credentials entered are invalid.

If the entered credentials are wrong

This page says

INVALID CREDENTIALS

OK

If the credentials are correct , then they take to the Veteran Orders Select page.

ORDER ID: 10002

VETERAN: Sherwin

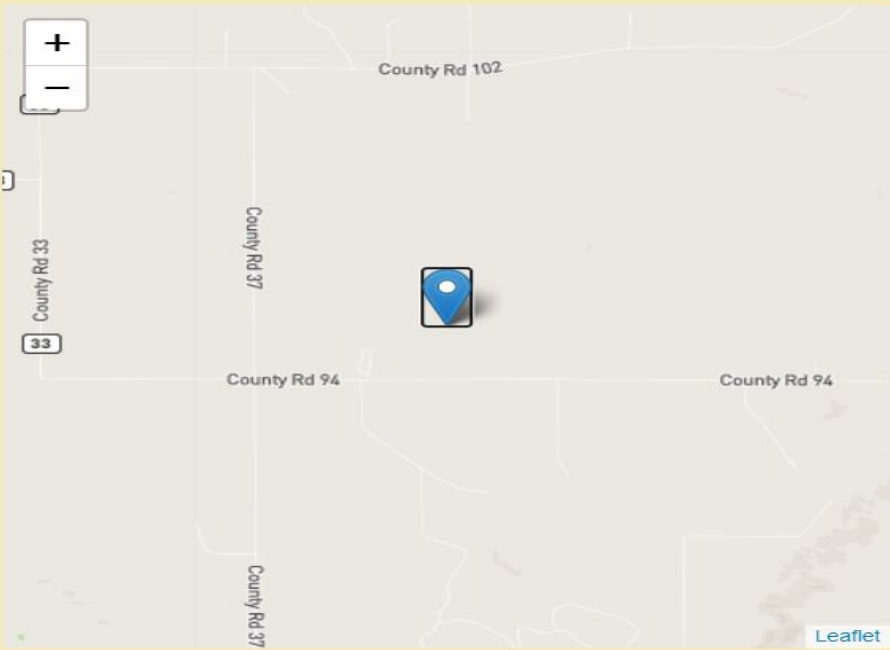
PRICE: 16.3

ITEMS: Banana-1,Carrots-3

TAKE Order

So if the credentials are correct then we make a GET request call to <https://localhost:44335/api/Items/>. So then the API tries to collect all of the orders which have been made but not yet accepted by the Volunteers. So we get the Orderno , the Veteran making the order, the price of the Order and the items bought in the order. So we also get the coordinates of the location of the veteran who has made the order. So now we run a loop for all the orders and mark the locations in a map which is available in the Veteran Orders Select Page. So when the Volunteer clicks the marker , the details of the order are displayed below. So now the Volunteer can take up Orders that he wishes.

VOLUNTEER ORDERS SELECT PAGE



ORDER ID: 10002

VETERAN: Sherwin

PRICE: 16.3

ITEMS: Banana-1,Carrots-3

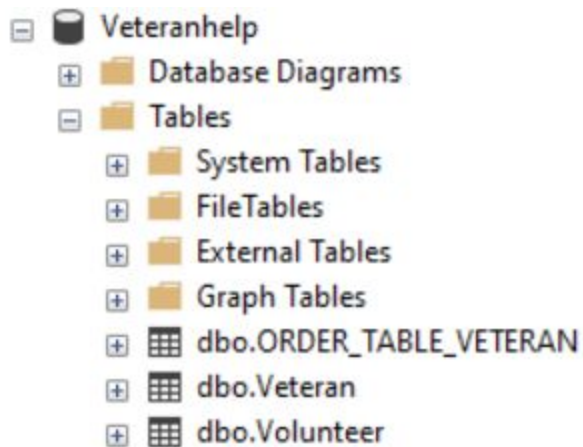
DATABASE TABLES

There are three tables in the Database VeteranHelp. They are

1. Veteran
2. Volunteer
3. ORDER_TABLE_VETERAN

So we store the details of the Veteran along with the credentials in the Veteran table. We store the details of the Volunteer along with the credentials in the Volunteer table. The orders made are stored in ORDER_TABLE_VETERAN.

Database Structure



The web application now has fully operational functionalities to register a veteran , register a volunteer , sign in as a veteran and a volunteer and make orders as a veteran. Finally being able to take up orders by the volunteers and displaying the orders in a table.