Project Web Programming

By

Miss Kanyavee	**Likitwattanakij**	**6488038**
Miss Arada	**Raksapatcharawong**	**6488066**
Mr. Tanawat	**Raweepachwarathon**	**6488067**
Mr. Ittikorn	**Suksai**	**6488088**
Mr. Sarttra	**Prasongtichol**	**6488124**

A Report Submitted in Partial Fulfillment of

the Requirements for

ITCS212_Web Programming

Faculty of Information and Communication Technology

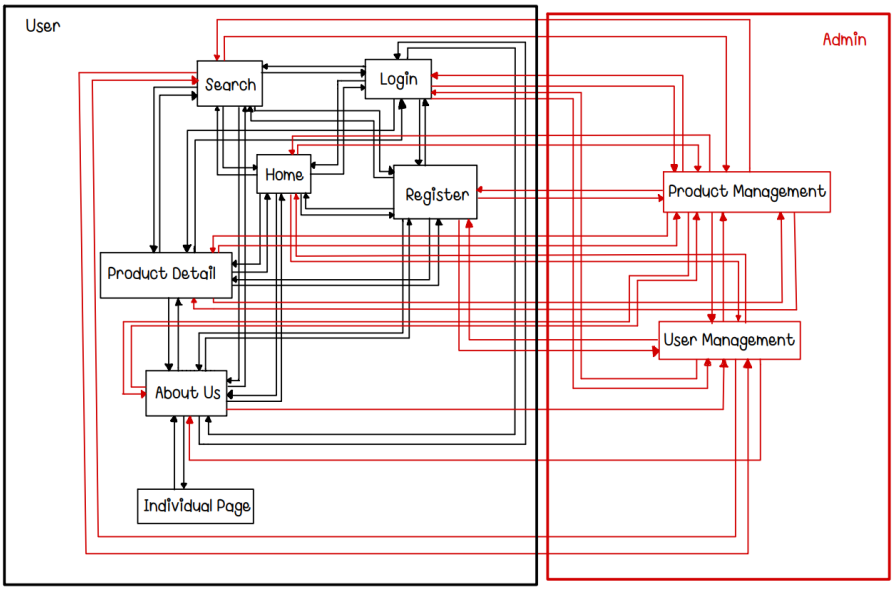
Mahidol University

2022

Project of Web Programming

1. Project Overview
    - Phase 1 (HTML,CSS)
      For this phase, it will be the design of website pages in each section using html to design various sections within the web page. Whether it's a search box navigation bar or input field. Which will be put in various pages such as home page, Admin page or Product management page. Then, CSS code will be used to decorate and design the website to be more beautiful to be ready for Phase 2.
    - Phase 2 (JavaScript [Node.js], Web Services, and DB)
      Phase 2 will be the back end which for this phase will connect data with SQL or Database and API to make web service, which will use Javascript (Node.js) to help in connecting data in different parts by uses index.js as the center for all parts of the web service.

2. Navigation Diagram of your web application.



3. Details of your web application and code.
    - All pages have the same header navigation bar and footer navigation bar template but there are changing in variations of color in individual.

Fundamental navigation bar code for all html files

style.css: this is css class source code on header and footer nav bar on IGD Website

```css
*{
  font-family:'Dangrek';
}
@keyframes fadeIn {
  0% { opacity: 0; }
  100% { opacity: 1; }
}
.fade-in-image {
  animation: fadeIn 2s;
}
.fade-in-text {
  animation: fadeIn 1.5s;
}
header, footer{
        background-color: #1FC454;
        padding: 5px;
        text-align: center;
}
header{
  background: rgb(167,255,153);
  background: linear-gradient(90deg, rgba(167,255,153,1) 0%, rgba(95,224,107,1) 0%, rgba(0,168,73,1) 100%);
  border-bottom: 3.5px solid #009B34;
  height: 200px;
  display: block;
  position: relative;
  overflow: hidden;
  z-index: 1;
}
```

```css
nav ul li {
display: inline-block
}
 .head{
    background: #ffc92e;
    border:4px solid #009B34;
    border-bottom:none;
    text-align: center;
    height: 70px;
    width: 60%;
    margin: 0 auto;
    padding-bottom: 30px;
    position: relative;
    top: 0;
    left: 0;
    z-index: 2;
}
nav ul li a.headnav{
    text-decoration: none;
    padding: 0px 30px;
    color: #009B34 ;
    border-left: 2px solid #009B34 ;
    border-right: 2px solid #009B34 ;
  }
  nav ul li a.headnav:hover{
    color: #E18508   ;
  }
  nav ul li a.footnav{
    text-decoration: none;
    padding: 30px 20px;
    color: black;
  }
```

```css
footer {
    background: rgb(167,255,153);
    background: linear-gradient(90deg, rgba(167,255,153,1) 0%, rgba(95,224,107,1) 0%, rgba(0,168,73,1) 100%);
    display: block;
    position: absolute;
    bottom: 0;
    left: 0;
    right: 0;
    margin: 0;
    min-height: 7%;
    width: 100%;
  }
```

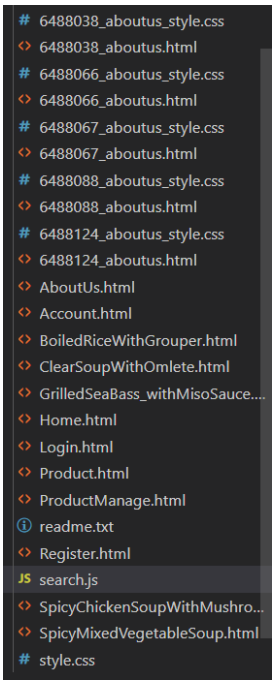
<individual about us> .css : there are numerous variations of navigation bar for this, the example below here.

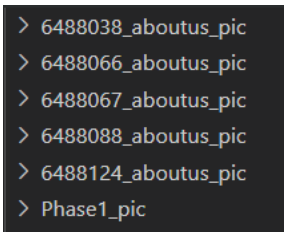
```css
*{
    font-family:'Bad Script';
  }
  header{
        background: rgb(150,230,124);
        background: linear-gradient(0deg, rgba(150,230,124,1) 0%, rgba(0,175,111,1) 100%);
        border-bottom: 3px solid white;
        padding: 15px;
        text-align: center;
        height: 180px;
        display: block;
        position: relative;
        overflow: hidden;
        z-index: 1;
  }
  body{
    background: whitesmoke;
    min-height: 100vh;
    position: relative;
    flex-direction: column;
    margin: 0;
    padding-bottom: 100px;
  }
```

```css
nav ul li {
  display: inline-block;
  margin-right: auto;
  margin-left: auto;
}
nav ul li :hover.headnav{
  color: #A06300;
}
nav ul li :hover.footnav{
  color: #A06300;
  text-decoration: underline;
}
 .head{
    background: #F7C535;
    border: 3px solid white;
    text-align: center;
    padding-bottom: 30px;
    width: 80%;
    margin: 0 auto;
    position: relative;
    top: 0;
    left: 0;
    z-index: 2;
}
```

```css
nav ul li a.headnav{
    text-decoration: none;
    color: white;
    padding:0px 40px;
    border-right: 2px solid white;
    border-left: 2px solid white;
    margin: 0px auto;
}
nav ul li a.footnav{
  text-align: center;
  text-decoration: none;
  font-weight: bold;
  color: white;
  padding:10px 30px;
  margin: 0px auto;
}
```

- This css use to determines the design for each page on the website.





- **Home**

There are three components that provide in the home page.

1. Title part which uses to show the name of the website in graphic representation.

```
<div class="foodheader">
    <div class="titlehome">
        <img class="homefood" src="Phase1_pic/IGD-removebg-preview.png" alt="headfood" width="200" height="200">
        <h1 class="foodtopic" style="color:#ffc92e;">IGD's Website</h1>
    </div>
</div>
```

```
.foodheader {
    background: #008356;
    background-image: url(Phase1_pic/pngimg.com\ -\ pasta_PNG15.png);
    background-repeat: no-repeat;
    background-position: 1200px 250px;
    background-size: contain;
    height: 420px;
    overflow: hidden;
}
```

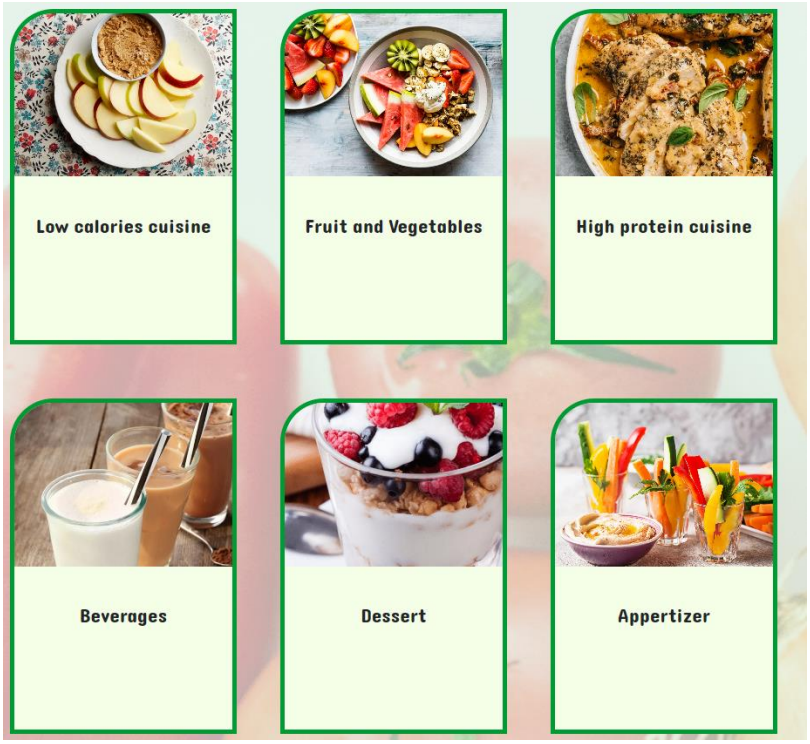
2. Recommend menus, each menu can be examined by clicking on its picture as well as more information link reference.

```
<article class="container1">
    <div class="box2">
        <a href="ClearSoupWithOmlete.html"><img src="Phase1_pic/ClearSoupWithOmlete1.jpg" width="300"
            height="170" style="object-fit: cover;"></a>
        <h3 class="recommend">Clear soup with Omlete</h3>
        <div class="description">
            <a class="more" href="ClearSoupWithOmlete.html">more information</a>
        </div>
    </div>
    <div class="box2">
        <a href="GrilledSeaBass_withMisoSauce.html"><img src="Phase1_pic/GrilledSeaBasswithMisoSauce.jpg"
            width="300" height="170" style="object-fit: cover;"></a>
        <h3 class="recommend">Grilled Sea Bass with Miso Sauce</h3>
        <div class="description">
            <a class="more" href="GrilledSeaBass_withMisoSauce.html">more information</a>
        </div>
    </div>
    <div class="box2">
        <a href="SpicyChickenSoupWithMushroom.html"><img src="Phase1_pic/SpicyChickenSoupWithMushrooms.jpg"
            width="300" height="170" style="object-fit: cover;"></a>
        <h3 class="recommend">Spicy chicken soup with mushrooms</h3>
        <div class="description">
            <a class="more" style="top:15px;" href="SpicyChickenSoupWithMushroom.html">more information</a>
        </div>
    </div>
    <div class="box2">
        <a href="SpicyMixedVegetableSoup.html"><img src="Phase1_pic/SpicyMixedVegetableSoup.jpg" width="300" height="170"
            style="object-fit: cover;"></a>
        <div class="description">
            <h3 class="recommend">Spicy mixed vegetable soup</h3>
            <a  class="more" style="padding-top: 55px;" href="SpicyMixedVegetableSoup.html">more information</a>
        </div>
    </div>
    <div class="box2">
        <a href="BoiledRiceWithGrouper.html"><img src="Phase1_pic/BoiledRiceWithGrouper.jpg"
            width="300" height="170" style="object-fit: cover;"></a>
        <div class="description">
            <h3 class="recommend">Boiled Rice with Grouper</h3>
            <a  class="more" style="padding-top: 55px;" href="BoiledRiceWithGrouper.html">more information</a>
        </div>
    </div>
</article>
```

3. Categories can be divided into 6 types.

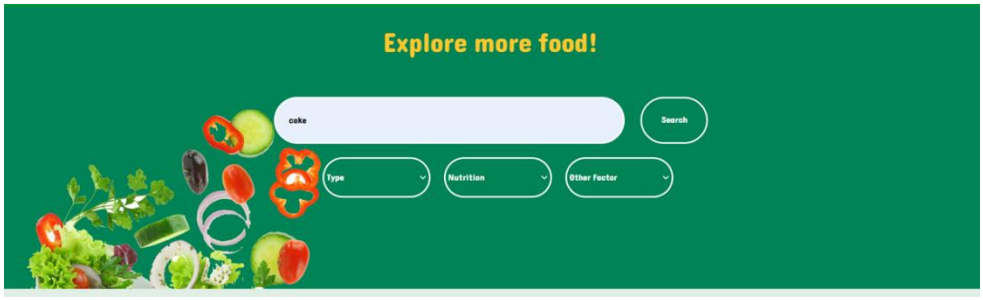
```
<article class="container1">
    <div class="box">
        <a href=""><img src="Phase1_pic//apples-peanut-butter-cashew-butter-butter-snack-1296x728-header.png"
                width="250" height="150" style="object-fit: cover;"></a>
        <div class="description">
            <h3>Low calories cuisine</h3>
        </div>
    </div>
    <div class="box">
        <a href=""><img src="Phase1_pic//low-sugar-foods-297447-1644010656361-main.700x0c.jpg" width="250"
                height="150" style="object-fit: cover;"></a>
        <div class="description">
            <h3>Fruit and Vegetables</h3>
        </div>
    </div>
    <div class="box">
        <a href=""><img src="Phase1_pic//basil-chicken-cream-sauce-featured-image.jpg" width="250" height="150"
                style="object-fit: cover;"></a>
        <div class="description">
            <h3>High protein cuisine</h3>
        </div>
    </div>
</article>
```

```
<div class="fade-in-image">
    <article class="container1">
        <div class="box">
            <a href=""><img src="Phase1_pic//protein-shake-recipes2.png" width="250" height="150"
                    style="object-fit: cover;"></a>
            <div class="description">
                <h3>Beverages</h3>
            </div>
        </div>
        <div class="box">
            <a href=""><img src="Phase1_pic//Yogurt-Parfait-with-Berries-768x1152.png" width="250" height="150"
                    style="object-fit: cover;"></a>
            <h3>Dessert</h3>
        </div>
        <div class="box">
            <a href=""><img src="Phase1_pic//low-calorie-snack.png" width="250" height="150"
                    style="object-fit: cover;"></a>
            <h3>Appertizer</h3>
        </div>
    </article>
</div>
```



- **Product**

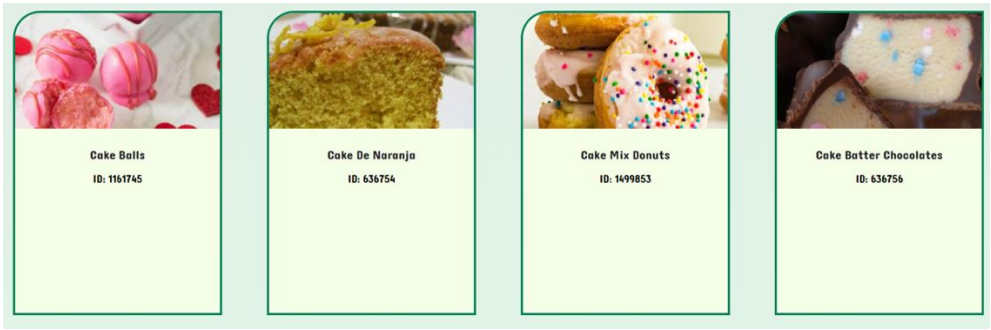
  Search which includes 3 criteria consist of Type Nutrition Other factors.




```
<form id="search-form">
    <section class="searchpart" id="search">
        <h1 class="foodselection1" style="font-size: 40px; color: #ffc92e;">
            Explore more food!
        </h1>
        <div class="searchbar">
            <input class="input1" type="search" id="search-input" placeholder="Search" />
            <button type="submit" onclick="searchFood()" >Search</button>
        </div>
        <div class="custom-select">
            <select id="type">
                <option class="dropdown" value="0">Type</option>
                <option class="dropdown" value="main dish">Main Dish</option>
                <option class="dropdown" value="appetizer">Appetizer</option>
                <option class="dropdown" value="dessert">Dessert</option>
                <option class="dropdown" value="beverage">Beverage</option>
            </select>
            <select id="nutrition">
                <option class="dropdown" value="0">Nutrition</option>
                <option class="dropdown" value="protein">Protein</option>
                <option class="dropdown" value="fat">Fat</option>
                <option class="dropdown" value="fruit and vegetables">Fruit and vegetables</option>
                <option class="dropdown" value="starch">Starch</option>
                <option class="dropdown" value="dairy">Dairy</option>
            </select>
            <select id="factor">
                <option class="dropdown" value="0">Other Factor</option>
                <option class="dropdown" value="seafood allergic">Seafood allergic</option>
                <option class="dropdown" value="low calories">Low Calories</option>
                <option class="dropdown" value="heart disease">Heart disease</option>
                <option class="dropdown" value="diabetes">Diabetes</option>
                <option class="dropdown" value="obesity">Obesity</option>
            </select>
        </div>
    </section>
</form>
```

The product will be indicated here on div id= "search-output"

```
<div id="search-output"></div>
```



- **Product detail**
  Product Details in each page. The components

```
<h1 style="text-align: center;">Boiled Rice with Grouper</h1>
<img class="food" src="Phase1_pic/BoiledRiceWithGrouper.jpg" alt="Boiled Rice with Grouper" class="food">

<h2 style="padding-left: 75px;">Description</h2>
<article class="fooddesc">
    <div class="fooddescription">
        It's a food that's easy to digest, not sticky. Beneficial to the body. Can be eaten by all genders,
        ages,
        suitable for patients.
    </div>
</article>

<h2 style="padding-left: 65px;">Recipe</h2>
<article class="fooddesc">
    <div class="fooddescription">
        <ul>
            <li>Pork bone broth 1,200 ml</li>
            <li>Young galangal, thinly sliced 3 tbsp.</li>
            <li>Shiitake mushrooms (soaked in water until soft) 100 g</li>
            <li>4 tablespoons soy sauce</li>
            <li>1 tablespoon sugar</li>
            <li>2 teaspoons ground pepper</li>
            <li>Grouper fillet (cut into pieces, boiled) 10 pieces.</li>
            <li>300 g cooked jasmine rice.</li>
            <li>Finely chopped galangal</li>
            <li>Braised leeks</li>
            <li>Chopped celery</li>
        </ul>
    </div>
</article>
```

```
<h2 style="padding-left: 75px;">Method</h2>
<article class="fooddesc">
    <div class="fooddescription">
        <ol>
            <li>Boil pork bone broth with young galangal until boiling, add shiitake mushrooms, season with
            light soy sauce,
            sugar and ground pepper, reduce heat and simmer until boiling broth. Set aside.</li>
            <li>Heat the water in another pot until it boils. Add jasmine rice to cook in boiling water until
            soft, scoop
            into a bowl, and place blanched grouper. Put the pork bone broth. sprinkle finely pounded
            galangal, fried
            garlic, and celery. Eat while hot with soybean sauce.</li>
        </ol>
    </div>
</article>
```

Boiled Rice with Grouper

Description

It's a food that's easy to digest, not sticky. Beneficial to the body. Can be eaten by all genders, ages, suitable for patients.

Recipe

- Pork bone broth 1,200 ml
- Young galangal, thinly sliced 3 tbsp.
- Shiitake mushrooms (soaked in water until soft) 100 g
- 4 tablespoons soy sauce
- 1 tablespoon sugar
- 2 teaspoons ground pepper
- Grouper fillet (cut into pieces, boiled) 10 pieces.
- 300 g cooked jasmine rice.
- Finely chopped galangal
- Braised leeks
- Chopped celery

Method

1. Boil pork bone broth with young galangal until boiling, add shiitake mushrooms, season with light soy sauce, sugar and ground pepper, reduce heat and simmer until boiling broth. Set aside.
2. Heat the water in another pot until it boils. Add jasmine rice to cook in boiling water until soft, scoop into a bowl, and place blanched grouper. Put the pork bone broth. sprinkle finely pounded galangal, fried garlic, and celery. Eat while hot with soybean sauce.

- **Search**

  Search section already include in the product page. In fact, this part call function Searchfood() from search.js

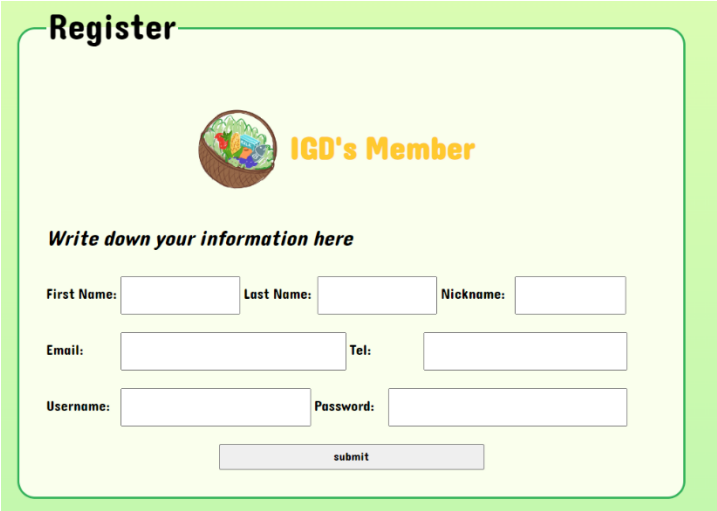
```html
<form id="search-form">
    <section class="searchpart" id="search">
        <h1 class="foodselection1" style="font-size: 40px; color: #ffc92e;">
            Explore more food!
        </h1>
        <div class="searchbar">
            <input class="input1" type="search" id="search-input" placeholder="Search" />
            <button type="submit" onclick="searchFood()" >Search</button>
        </div>
        <div class="custom-select">
            <select id="type">
                <option class="dropdown" value="0">Type</option>
                <option class="dropdown" value="main dish">Main Dish</option>
                <option class="dropdown" value="appetizer">Appetizer</option>
                <option class="dropdown" value="dessert">Dessert</option>
                <option class="dropdown" value="beverage">Beverage</option>
            </select>
            <select id="nutrition">
                <option class="dropdown" value="0">Nutrition</option>
                <option class="dropdown" value="protein">Protein</option>
                <option class="dropdown" value="fat">Fat</option>
                <option class="dropdown" value="fruit and vegetables">Fruit and vegetables</option>
                <option class="dropdown" value="starch">Starch</option>
                <option class="dropdown" value="dairy">Dairy</option>
            </select>
            <select id="factor">
                <option class="dropdown" value="0">Other Factor</option>
                <option class="dropdown" value="seafood allergic">Seafood allergic</option>
                <option class="dropdown" value="low calories">Low Calories</option>
                <option class="dropdown" value="heart disease">Heart disease</option>
                <option class="dropdown" value="diabetes">Diabetes</option>
                <option class="dropdown" value="obesity">Obesity</option>
            </select>
        </div>
    </section>
</form>
```

- **Login**

  Ther is a form for user to input username and password to access to website.

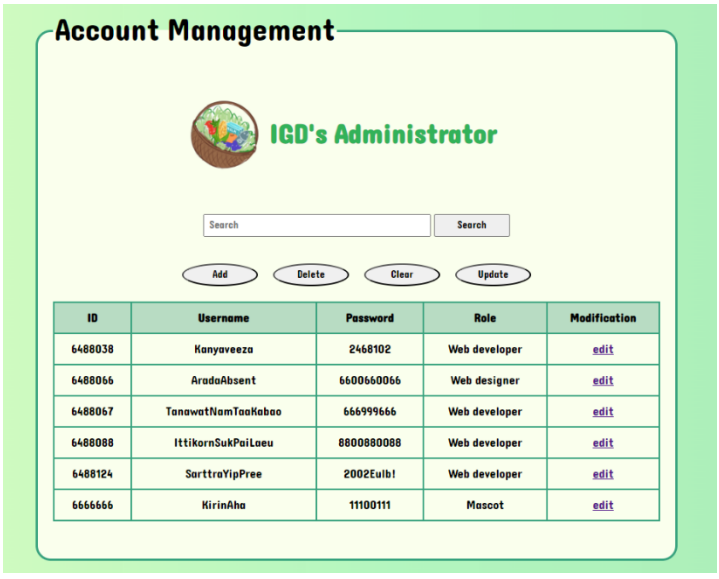

```html
<form action="/login" method="post">
<fieldset class="fieldset1">
    <legend style="color:#228160">Login</legend>
    <img class="imglogin" src="Phase1_pic/IGD-removebg-preview.png" width="100" height="100">
    <h2>IGD Website</h2>
    <input class="input1" type="text" placeholder="Enter Username" name="username" required>
    <input class="input1"type="password" placeholder="Enter Password" name="password" required>
    <button type="login">Login</button>
    Don't have account?
    <a href="Register.html">Register here</a>
</fieldset>
</form>
```

- **Register**



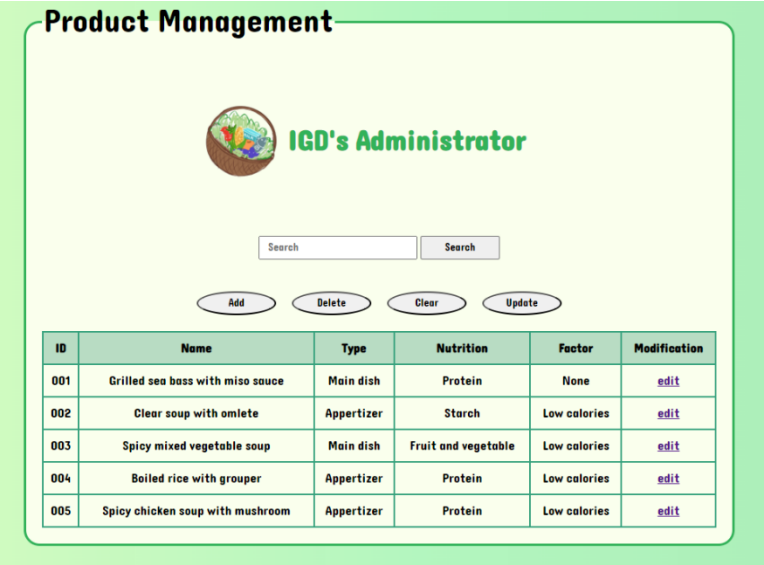

```
<form action="/register" method="post">
    <fieldset class="fieldset2">
        <legend>Register</legend>
        <div class="titlereg">
            <img class="homefood" src="Phase1_pic/IGD-removebg-preview.png" alt="headfood" width="120" height="120">
            <h1 class="foodtopic" style="color:#ffc92e;">IGD's Member</h1>
        </div>
        <h2 style="font-style: italic;"> Write down your information here</h2>
        <label>First Name:</label>
        <input class="input1" style="width: 120px;" type="text" name="firstName" required>
        <label>Last Name:</label>
        <input class="input1" style="width: 120px;" type="text" name="lastName" required>
        <label>Nickname:</label>
        <input class="input2" style="width: 110px;" type="text" name="nickname" required>
        <br>
        <label>Email:</label>
        <input class="input1" style="width: 240px;"  type="email" name="email" required>
        <label>Tel:</label>
        <input class="input1" style="width: 215px;"  type="tel" name="mobile" required>
        <br>
        <label>Username:</label>
        <input class="input1" style="width: 200px;" type="text" name="username" required>
        <label>Password:</label>
        <input class="input1" style="width: 255px;" type="text" name="password" required><br>
        <button class="reg" type="submit">submit</button>
    </fieldset>
</form>
```

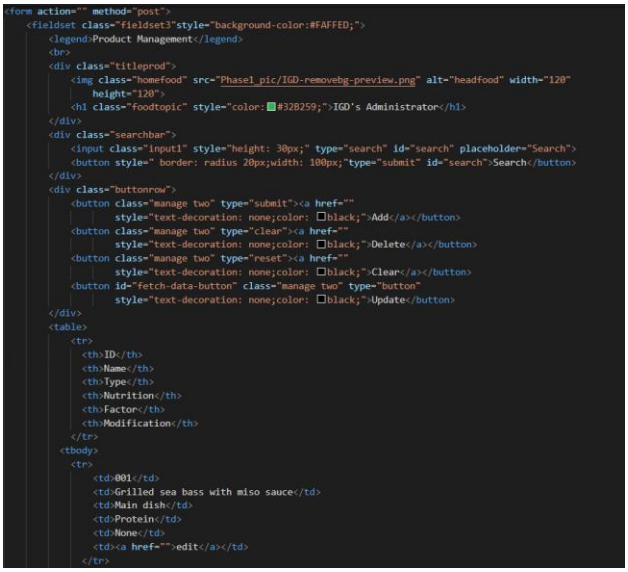
- **User Account management**




```
form>
    <div class="titleacc">
        <img class="homefood" src="Phase1_pic/IGD-removebg-preview.png" alt="headfood" width="120"
            height="120">
        <h1 class="foodtopic" style="color:#32B259;">IGD's Administrator</h1>
    </div>
    <div class="formbox">
        <div class="searchbar">
            <input class="input1" style="height: 30px;" type="search" id="search" placeholder="Search">
            <button style=" border: radius 20px;width: 100px;"type="submit" id="search">Search</button>
        </div>
        <div class="buttonrow">
            <button class="manage two" type="submit"><a href=""
                    style="text-decoration: none;color: black;">Add</a></button>
            <button class="manage two" type="clear"><a href=""
                    style="text-decoration: none;color: black;">Delete</a></button>
            <button class="manage two" type="reset"><a href=""
                    style="text-decoration: none;color: black;">Clear</a></button>
            <button id="fetch-data-button" class="manage two" type="button"
                    style="text-decoration: none;color: black;">Update</button>
        </div>
        <table>
            <tr>
              <th>ID</th>
              <th>Username</th>
              <th>Password</th>
              <th>Role</th>
              <th>Modification</th>
            </tr>
          <tbody>
            <tr>
                <td>6488038</td>
                <td>Kanyaveeza</td>
                <td>2468102</td>
                <td>Web developer</td>
```

- **Product Management**





```
<form action="" method="post">
    <fieldset class="fieldset3"style="background-color:#FAFFED;">
        <legend>Product Management</legend>
        <br>
        <div class="titleprod">
            <img class="homefood" src="Phase1_pic/IGD-removebg-preview.png" alt="headfood" width="120"
                height="120">
            <h1 class="foodtopic" style="color: #328259;">IGD's Administrator</h1>
        </div>
        <div class="searchbar">
            <input class="input1" style="height: 30px;" type="search" id="search" placeholder="Search">
            <button style=" border: radius 20px;width: 100px;"type="submit" id="search">Search</button>
        </div>
        <div class="buttonrow">
            <button class="manage two" type="submit"><a href=""
                    style="text-decoration: none;color: black;">Add</a></button>
            <button class="manage two" type="clear"><a href=""
                    style="text-decoration: none;color: black;">Delete</a></button>
            <button class="manage two" type="reset"><a href=""
                    style="text-decoration: none;color: black;">Clear</a></button>
            <button id="fetch-data-button" class="manage two" type="button"
                    style="text-decoration: none;color: black;">Update</button>
        </div>
        <table>
            <tr>
                <th>ID</th>
                <th>Name</th>
                <th>Type</th>
                <th>Nutrition</th>
                <th>Factor</th>
                <th>Modification</th>
            </tr>
            <tbody>
            <tr>
                <td>001</td>
                <td>Grilled sea bass with miso sauce</td>
                <td>Main dish</td>
                <td>Protein</td>
                <td>None</td>
                <td><a href="">edit</a></td>
            </tr>
```

- **About Us**

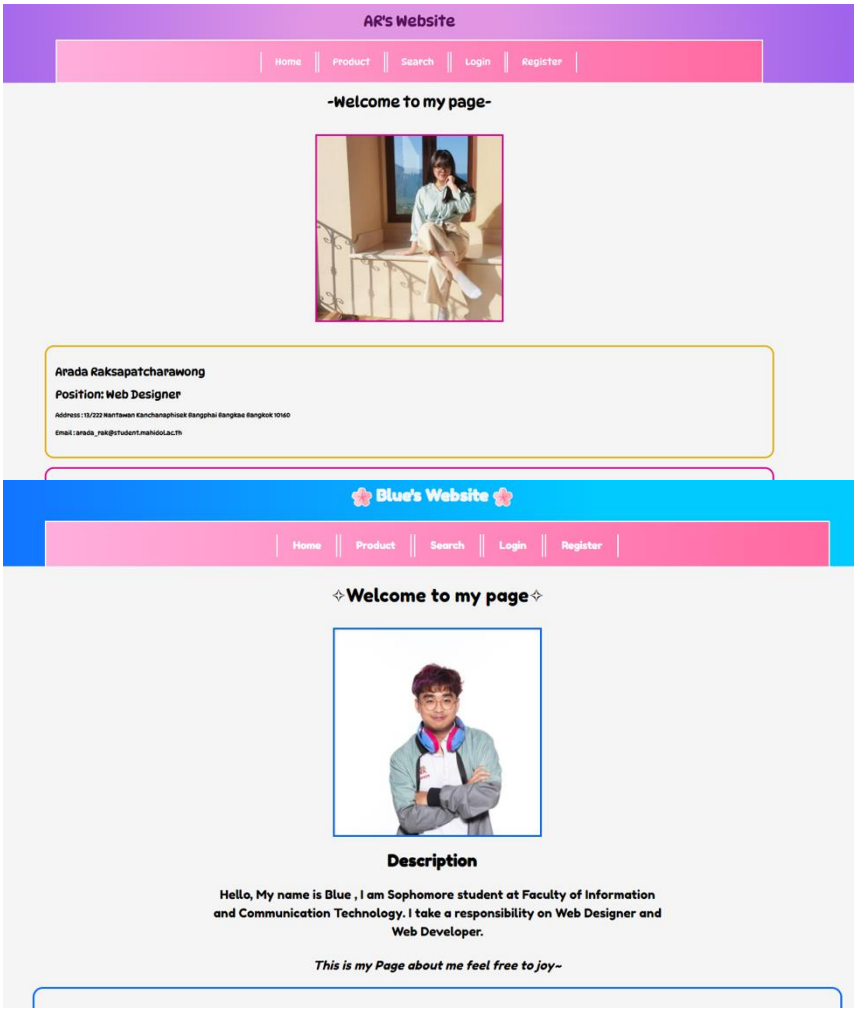
```
<article class="containerabout">
    <div class="boxabout">
        <img src="Phase1_pic/Member_pic/S__12034062.jpg" width="250" height="300">
        <div class="description">
            <h2 style="margin: 1px;padding-bottom: 0px;">Kanyavee Likitwattanakij</h2>
            <div class="memberdesc">
                <h4 style="font-style: italic;">Position: Web Developer,Web Designer</h4>
                <a class="aboutnav" href="/6488038aboutus">
                    <h4 style="font-style: italic;">About this member</h4>
                </a>
            </div>
        </div>
    </div>
    <div class="boxabout">
        <img src="Phase1_pic/Member_pic/316798456_635713815008945_4839329360067940631_n.jpg" width="300"
            height="300">
        <div class="description">
            <h2 style="margin: 1px;padding-bottom: 0px;">Arada Raksapatcharawong</h2>
            <div class="memberdesc">
                <h4 style="font-style: italic;">Position:Web Designer</h4>
                <a class="aboutnav" href="/6488066aboutus">
                    <h4 style="font-style: italic;">About this member</h4>
                </a>
            </div>
        </div>
    </div>
    <div class="boxabout">
        <img src="Phase1_pic/Member_pic/Profile.jpg" width="250" height="300">
        <div class="description">
            <h2 style="margin: 1px;padding-bottom: 0px;font-size: 22px;">Tanawat Raweepachwarathon</h2>
            <div class="memberdesc">
                <h4 style="font-style: italic;">Position:Web Developer</h4>
                <a class="aboutnav" href="/6488067aboutus">
                    <h4 style="font-style: italic;">About this member</h4>
                </a>
            </div>
        </div>
    </div>
</article>
```

```
<div class="fade-in-image">
    <article class="containerabout">
        <div class="boxabout">
            <img src="Phase1_pic/Member_pic/088myself.jpg" width="270" height="270">
            <div class="description">
                <h2 style="margin: 1px;padding-bottom: 0px;">Ittikorn Suksai</h2>
                <div class="memberdesc">
                    <h4 style="font-style: italic;">Position:Web Developer</h4>
                    <a class="aboutnav" href="/6488088aboutus">
                        <h4 style="font-style: italic;">About this member</h4>
                    </a>
                </div>
            </div>
        </div>
        <div class="boxabout">
            <img src="Phase1_pic/Member_pic/Praaek.jpg" width="270" height="270">
            <div class="description">
                <h2 style="margin: 1px;padding-bottom: 0px;">Sarttra Prasongtichol</h2>
                <div class="memberdesc">
                    <h4 style="font-style: italic;">Position:Web Developer,Web Designer</h4>
                    <a class="aboutnav" href="/6488124aboutus">
                        <h4 style="font-style: italic;">About this member</h4>
                    </a>
                </div>
            </div>
        </div>
    </article>
```

- **Individual About Us**
  Everyone has different template and css of their individual page.

```
<body>
    <header>
        <h2 style="color: aliceblue;"><img class="imghead" src="6488124_aboutus_pic/white-cherry-blossom.png"
                alt="sakura" width="40" height="40"> Blue's Website <img class="imghead"
                src="6488124_aboutus_pic/white-cherry-blossom.png" alt="sakura" width="40" height="40"> </h2>
        <nav class="head" id="navbar">
            <ul>
                <li><a href="/" class="midfont headnav">Home</a></li>
                <li><a href="/product" class="midfont headnav">Product</a></li>
                <li><a href="/search" class="midfont headnav">Search</a></li>
                <li><a href="/login" class="midfont headnav">Login</a></li>
                <li><a href="/register" class="midfont headnav">Register</a></li>
            </ul>
        </nav>
    </header>
    <p style="font-size: 36px; text-align: center;">✧Welcome to my page✧</p>
    <img class="imgportrait" src="6488124_aboutus_pic/Praaek.jpg" alt="mypic" width="400" height="400">
    <h2 style="text-align: center;">Description</h2>
    <p>Hello, My name is Blue , I am Sophomore student at Faculty of Information and Communication Technology. I take a
        responsibility on Web Designer and Web Developer.</p>
    <p style="font-style: italic;">This is my Page about me feel free to joy~</p>
    <section>
        <h2>Sarttra Prasongtichol</h2>
        <h2>Position: Web Designer,Web Developer</h2>
        <br>
        <li>Address:87/180 Burasiri Village Ratchapruek ,Pak Kret,Nonthaburi,11120</li>
        <li>sarttra.pra@student.mahidol.ac.th</li>
    </section>
        <div class="section1">
        <h2> Education Blackground</h2>
        <ul>
            <li>Faculty of Information and Communication Technology,Mahidol University, Thailand
                <br>
                Bachelor of Science(Information Technology); GPA 2.73/4.00; August 2021 - Present
            </li>
            <li>Saint Gabriel's College, Thailand(Primary 1 to Secondary 6)
                <br>
                Major: Math-Science; GPA 3.32/4.00; May 2018-Febuary 2020
            </li>
        </ul>
        </div>
```

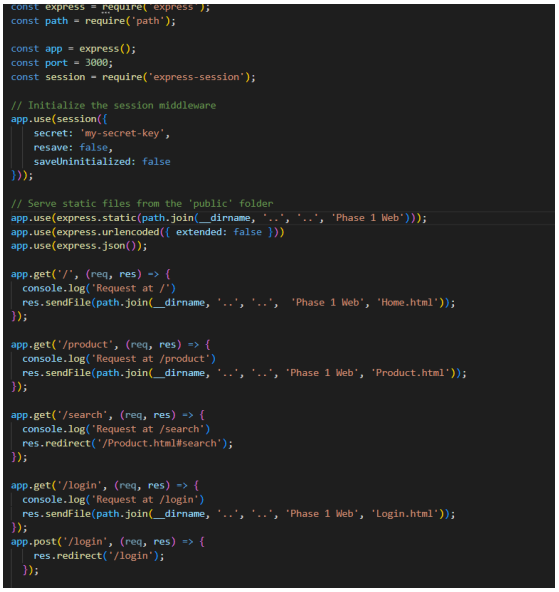
```
<body>
    <header>
        <h2 style="color: aliceblue;"><img class="imghead" src="6488124_aboutus_pic/white-cherry-blossom.png"
                alt="sakura" width="40" height="40"> Blue's Website <img class="imghead"
                src="6488124_aboutus_pic/white-cherry-blossom.png" alt="sakura" width="40" height="40"> </h2>
        <nav class="head" id="navbar">
            <ul>
                <li><a href="/" class="midfont headnav">Home</a></li>
                <li><a href="/product" class="midfont headnav">Product</a></li>
                <li><a href="/search" class="midfont headnav">Search</a></li>
                <li><a href="/login" class="midfont headnav">Login</a></li>
                <li><a href="/register" class="midfont headnav">Register</a></li>
            </ul>
        </nav>
    </header>
    <p style="font-size: 36px; text-align: center;">✧Welcome to my page✧</p>
    <img class="imgportrait" src="6488124_aboutus_pic/Praaek.jpg" alt="mypic" width="400" height="400">
    <h2 style="text-align: center;">Description</h2>
    <p>Hello, My name is Blue , I am Sophomore student at Faculty of Information and Communication Technology. I take a
        responsibility on Web Designer and Web Developer.</p>
    <p style="font-style: italic;">This is my Page about me feel free to joy~</p>
    <section>
        <h2>Sarttra Prasongtichol</h2>
        <h2>Position: Web Designer,Web Developer</h2>
        <br>
        <li>Address:87/180 Burasiri Village Ratchapruek ,Pak Kret,Nonthaburi,11120</li>
        <li>sarttra.pra@student.mahidol.ac.th</li>
    </section>
        <div class="section1">
        <h2> Education Blackground</h2>
        <ul>
            <li>Faculty of Information and Communication Technology,Mahidol University, Thailand
                <br>
                Bachelor of Science(Information Technology); GPA 2.73/4.00; August 2021 - Present
            </li>
            <li>Saint Gabriel's College, Thailand(Primary 1 to Secondary 6)
                <br>
                Major: Math-Science; GPA 3.32/4.00; May 2018-Febuary 2020
            </li>
        </ul>
        </div>
```



4. Details of your web services and code.
    - **Front-end**

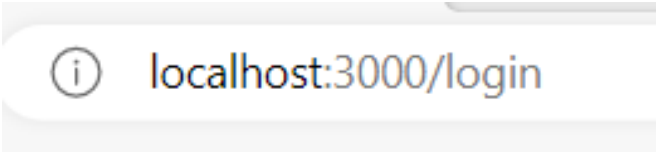
      Front-end server will be run on localhost:3000. The javascript file for front-end server consists of:
        - Index.js

```
app.get('/register', (req, res) => {
  console.log('Request at /register')
  res.sendFile(path.join(__dirname, '..', '..', 'Phase 1 Web', 'Register.html'));
});

app.get('/aboutus', (req, res) => {
  console.log('Request at /aboutus')
  res.sendFile(path.join(__dirname, '..', '..', 'Phase 1 Web', 'AboutUs.html'));
});

app.get('/accountmanage', (req, res) => {
  console.log('Request at /accountmanage')
  res.sendFile(path.join(__dirname, '..', '..', 'Phase 1 Web', 'Account.html'));
});

app.get('/productmanage', (req, res) => {
  console.log('Request at /productmanage')
  res.sendFile(path.join(__dirname, '..', '..', 'Phase 1 Web', 'ProductManage.html'));
});

app.get('/6488038aboutus', (req, res) => {
  console.log('Request at /6488038aboutus')
  res.sendFile(path.join(__dirname, '..', '..', 'Phase 1 Web', '6488038_aboutus.html'));
});

app.get('/6488066aboutus', (req, res) => {
  console.log('Request at /6488066aboutus')
  res.sendFile(path.join(__dirname, '..', '..', 'Phase 1 Web', '6488066_aboutus.html'));
});

app.get('/6488067aboutus', (req, res) => {
  console.log('Request at /6488067aboutus')
  res.sendFile(path.join(__dirname, '..', '..', 'Phase 1 Web', '6488067_aboutus.html'));
});

app.get('/6488088aboutus', (req, res) => {
  console.log('Request at /6488088aboutus')
  res.sendFile(path.join(__dirname, '..', '..', 'Phase 1 Web', '6488088_aboutus.html'));
});

app.get('/6488124aboutus', (req, res) => {
  console.log('Request at /6488124aboutus')
  res.sendFile(path.join(__dirname, '..', '..', 'Phase 1 Web', '6488124_aboutus.html'));
```



```
const express = require('express');
const path = require('path');

const app = express();
const port = 3000;
const session = require('express-session');

// Initialize the session middleware
app.use(session({
    secret: 'my-secret-key',
    resave: false,
    saveUninitialized: false
}));

// Serve static files from the 'public' folder
app.use(express.static(path.join(__dirname, '..', '..', 'Phase 1 Web')));
app.use(express.urlencoded({ extended: false }))
app.use(express.json());

app.get('/', (req, res) => {
  console.log('Request at /')
  res.sendFile(path.join(__dirname, '..', '..',  'Phase 1 Web', 'Home.html'));
});

app.get('/product', (req, res) => {
  console.log('Request at /product')
  res.sendFile(path.join(__dirname, '..', '..', 'Phase 1 Web', 'Product.html'));
});

app.get('/search', (req, res) => {
  console.log('Request at /search')
  res.redirect('/Product.html#search');
});

app.get('/login', (req, res) => {
  console.log('Request at /login')
  res.sendFile(path.join(__dirname, '..', '..', 'Phase 1 Web', 'Login.html'));
});
app.post('/login', (req, res) => {
    res.redirect('/login');
  });
```

If there are request to the page from the user, the address bar will show link like this.



Also, with the terminal since there are console.log command to indicate the status of web service.

- **Back-end**

Front-end server will be run on localhost:3030. The javascript file for front-end server consists of:

- index.js

  index.js played as a significant javascript file as it the center server that connect all the javascript file together when start node.js.

```
require('dotenv').config();
const express = require('express');
const path = require('path');
const db = require('./database');
const navbar = require('./navbar');
const auth = require('./auth');
const adminservice = require('./admin');
const productservice = require('./prod');
const app = express()
// Import the express-session module
const session = require('express-session');

// Initialize the session middleware
app.use(session({
    secret: 'my-secret-key',
    resave: false,
    saveUninitialized: false
}));

// Serve static files from the 'public' folder
app.use(express.static(path.join(__dirname, '..', '..',  'Phase 1 Web')));
app.use(express.urlencoded({ extended: false }))
app.use(express.json());

//MAKE ID FOR NAV BAR
app.use('/', navbar);

app.use(auth);
app.use(adminservice);
app.use(productservice);
app.listen(process.env.PORT, () => {
  console.log(`Server is listening on port ${process.env.PORT}`);
});

```

- database.js

  this database.js use to set the connection to sec2_gr2_database.sql to display the information from table to the system

  sec2_gr2_database.sql table:

```
create table if not exists AdminInformation(
AdminID     char(7) primary key,
AdminFname  varchar(50) not null,
AdminLname  varchar(50) not null,
AdminNname  varchar(25) not null,
AdminEmail  varchar(25) not null,
AdminPhone  varchar(12) not null
);

create table if not exists AdminLogin(
ID_Admin    char(7),
AdminUser   varchar(50) primary key,
AdminPass   varchar(50) not null,
AdminRole   varchar(25),
LoginLog    timestamp,
constraint FK_AdminInLog    foreign key(ID_Admin) references AdminInformation(AdminID));

create table if not exists ProductInfomation(
ProdID      char(3) primary key,
ProdName    varchar(50) not null,
ProdType    varchar(25),
ProdNeutri  varchar(25),
ProdFactor  varchar(25),
ProdDesc    text,
ProdRecipe  Text,
```

this is the code in database.js.

```
require('dotenv').config()
const sql = require('mysql2')
const db = sql.createPool({
    host: process.env.MYSQL_HOST,
    user: process.env.MYSQL_USERNAME,
    password: process.env.MYSQL_PASSWORD,
    database: process.env.MYSQL_DATABASE
  });

  //Connect to the DB
  db.getConnection((err, connection) => {
    if (err) {
      console.error('Error getting connection from pool:', err);
      return;
    }
    connection.release();
  });

  //DB on connect
  if (db) {
    db.on('connection', (connection) => {
      console.log(`Connection established on connection ${connection.config.database}`);
    });
  } else {
    console.log('Error');
  }
  module.exports = db;
```

- . env

  This environment variable (.env) use to determines mysql acess condition include port ,username , password, and schema.

```
const db = sql.createPool({
    host: process.env.MYSQL_HOST,
    user: process.env.MYSQL_USERNAME,
    password: process.env.MYSQL_PASSWORD,
    database: process.env.MYSQL_DATABASE
  });
```

```
# don't forget to add the schema privileges to your username in sec2_gr2_database schema
PORT=3030
MYSQL_HOST=localhost
MYSQL_USERNAME= tinycollegeUser # use your own username
MYSQL_PASSWORD= 2002Eulb! #use your own password
MYSQL_DATABASE= sec2_gr2_database
```

- navbar.js

  the same as mentioned in the index.js front-end section but it will change from app.get to route.get since this is the secondary javascript file so it need to export route by module.export = route

```
const express = require('express');
const path = require('path');
const route = express.Router();

route.get('/', (req, res) => {
  console.log('Request at /')
  res.sendFile(path.join(__dirname, '..', '..',  'Phase 1 Web', 'Home.html'));
});
```

```
module.exports = route;
```

Then we import this javascript file to index.js and call app.use function

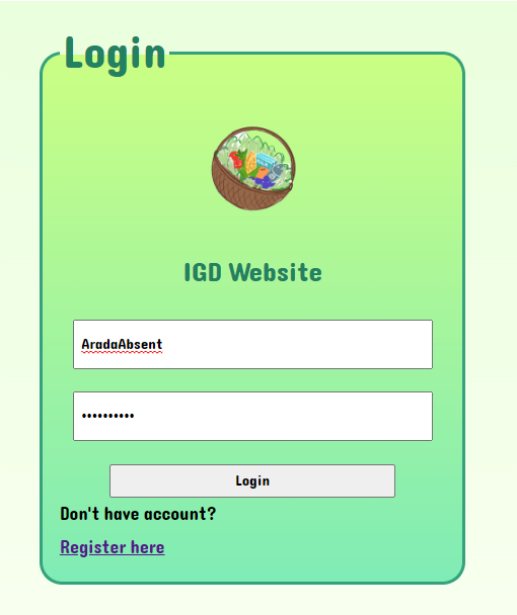
```
const navbar = require('./navbar');
```

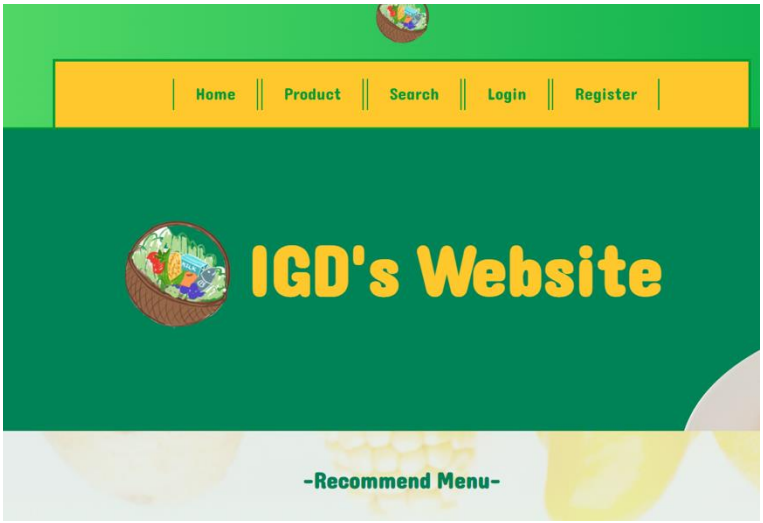
```
//MAKE ID FOR NAV BAR
app.use('/', navbar);
```

- auth,js
  this part is the authentication with getting information from user and administrators to access the web services. If the data is in the database, the login will be succeed and sent user to homepage, but if not the server will sent user back to login form again

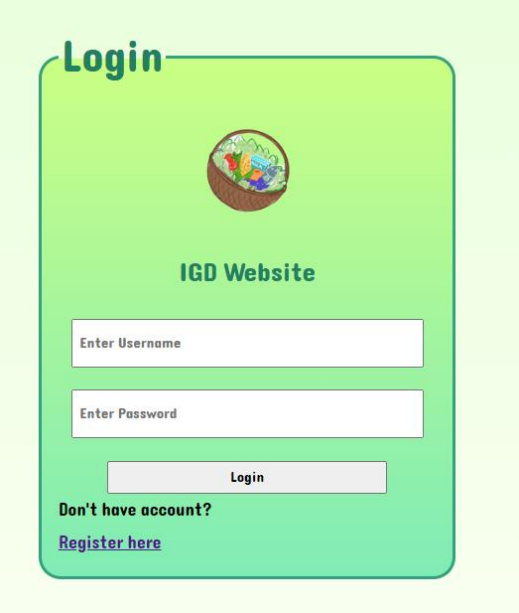
```
const express = require('express');
const db = require('./database');

const router = express.Router();

router.post('/login', (req, res) => {
  const { username, password } = req.body;
  db.query(
    'SELECT * FROM AdminLogin WHERE AdminUser = ? AND AdminPass = ?',
    [username, password],
    (error, results) => {
      if (error) {
        res.json({ success: false, error: error.message });
      } else if (results.length > 0) {
        // login successful, create session and return success response
        req.session.authenticated = true;
        res.redirect('/');
      } else {
        // login failed, return error response
        res.redirect('/login');
      }
    }
  );
});
```

- If login is true,

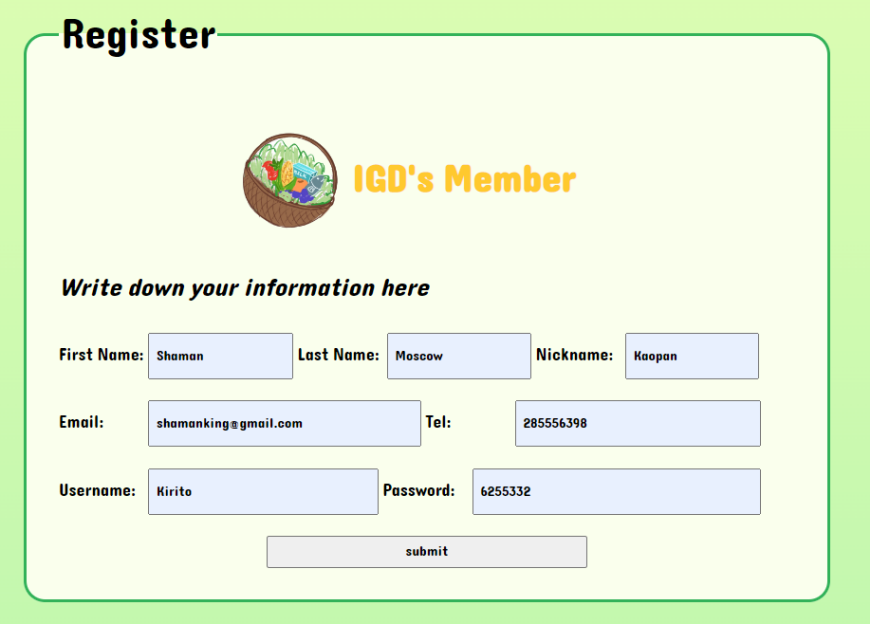


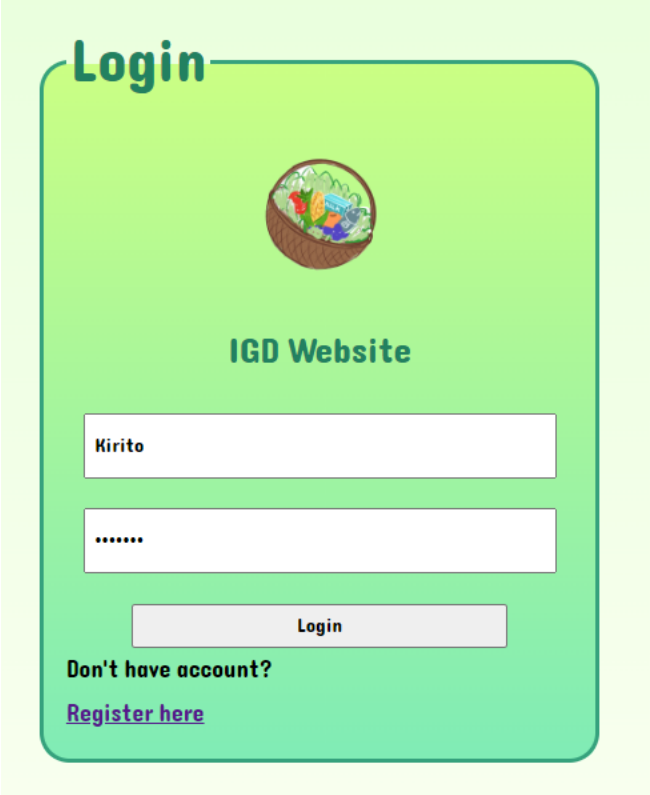


- If login is false,



For the register part, the website provides the form for user to input their data and submit it to mysql database to store the information in the system, after that they can use login function.





- **admin.js**

  admin.js is the function for administrator to modify their information in the system consist of:

  no criteria search: type localhost:3030/reg in address bar to display all admin data in the system.

```
router.get('/reg', async (req, res) => {
    try {
        const [UserData] = await db.promise().execute(
            `SELECT * FROM AdminInformation`,
        );
        const [UserData2] = await db.promise().execute(
            `SELECT * FROM AdminLogin`,
        );
        const data = {
            adminInformation: UserData,
            adminLogin: UserData2
        };
        console.log(data); // Log the merged data to the console
        return res.status(200).json({ error: false, data: data, message: "User successfully retrieved" });
    } catch (error) {
        console.error(error);
        return res.status(500).json({ error: true, message: 'Error retrieving user data' });
    }
});
```

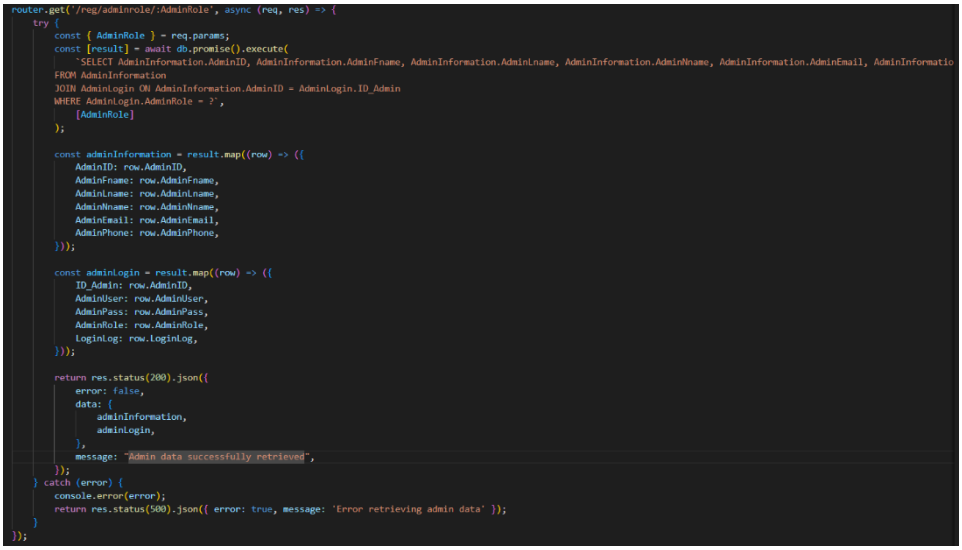

{"error":false,"data":{"adminInformation":[{"AdminID":"6488038","AdminFname":"Kanyavee","AdminLname":"Likitwattanakij","AdminNname":"Babe","AdminEmail":"Kanyavee@gmail.com","AdminPhone":"066-468-2969"},
{"AdminID":"6488066","AdminFname":"Arada","AdminLname":"Raksapatcharawong","AdminNname":"Aida","AdminEmail":"Arada@gmail.com","AdminPhone":"000-000-0000"},
{"AdminID":"6488067","AdminFname":"Tanawat","AdminLname":"Raweepachwarathon","AdminNname":"Pleum","AdminEmail":"Tanawat@gmail.com","AdminPhone":"092-625-2924"},
{"AdminID":"6488088","AdminFname":"Ittikorn","AdminLname":"Suksai","AdminNname":"Nop","AdminEmail":"Ittikorn@gmail.com","AdminPhone":"098-584-5035"},
{"AdminID":"6488124","AdminFname":"Sarttra","AdminLname":"Parsongtichol","AdminNname":"Blue","AdminEmail":"Sarttra@gmail.com","AdminPhone":"088-673-5591"},
{"AdminID":"6666666","AdminFname":"Kirin","AdminLname":"Elderdragon","AdminNname":"Kirin","AdminEmail":"Kirinza007@gmail.com","AdminPhone":"111-111-1111"},
{"AdminID":"GpXA6Yw","AdminFname":"Kaopankung","AdminLname":"Prasongtichol","AdminNname":"Sha","AdminEmail":"kaopanza@gmail.com","AdminPhone":"0618632002"}],"adminLogin":
[{"ID_Admin":"6488066","AdminUser":"AradaAbsent","AdminPass":"6600660066","AdminRole":"Web designer","LoginLog":null},{"ID_Admin":"GpXA6Yw","AdminUser":"Inosukeza","AdminPass":"420698","AdminRole":"Web
designer","LoginLog":"2023-04-25T05:43:16.000Z"},{"ID_Admin":"6488088","AdminUser":"IttikornSukPaiLaeu","AdminPass":"8800880088","AdminRole":"Web developer","LoginLog":null},
{"ID_Admin":"6488038","AdminUser":"Kanyaveeza","AdminPass":"2468102","AdminRole":"Web developer","LoginLog":null},{"ID_Admin":"6666666","AdminUser":"KirinAha","AdminPass":"11100111","AdminRole":"Mascot","LoginLog":null},
{"ID_Admin":"6488124","AdminUser":"SarttraYipPaee","AdminPass":"2002Eulb!","AdminRole":"Web developer","LoginLog":null},{"ID_Admin":"6488067","AdminUser":"TanawatNamTaaKabao","AdminPass":"666999666","AdminRole":"Web
developer","LoginLog":null}]},"message":"User successfully retrieved"}

Criteria search: Search by ID type localhost:3030/reg /:id in address bar to display the admin data that match input ID in the system.

```
router.get('/reg/:id', async (req, res) => {
    try {
        const { id } = req.params; // Get the admin ID from the request parameters
        const [UserData] = await db.promise().execute(
            `SELECT * FROM AdminInformation WHERE AdminID = ?`, [id] // Use the admin ID in the SQL query
        );
        const [UserData2] = await db.promise().execute(
            `SELECT * FROM AdminLogin WHERE ID_Admin = ?`, [id] // Use the admin ID in the SQL query
        );
        if (UserData && UserData2) {
            const data = {
                adminInformation: UserData[0], // The result of the SQL query is an array, so we need to access the first element
                adminLogin: UserData2[0]
            };
            console.log(data); // Log the merged data to the console
            return res.status(200).json({ error: false, data: data, message: "User successfully retrieved" });
        } else {
            return res.status(404).json({ error: true, message: "Admin not found" });
        }
    } catch (error) {
        console.error(error);
        return res.status(500).json({ error: true, message: 'Error retrieving admin data' });
    }
});
```


{"error":false,"data":{"adminInformation":{"AdminID":"6488067","AdminFname":"Tanawat","AdminLname":"Raweepachwarathon","AdminNname":"Pleum","AdminEmail":"Tanawat@gmail.com","AdminPhone":"092-625-2924"},"adminLogin":
{"ID_Admin":"6488067","AdminUser":"TanawatNamTaaKabao","AdminPass":"666999666","AdminRole":"Web developer","LoginLog":null}},"message":"User successfully retrieved"}

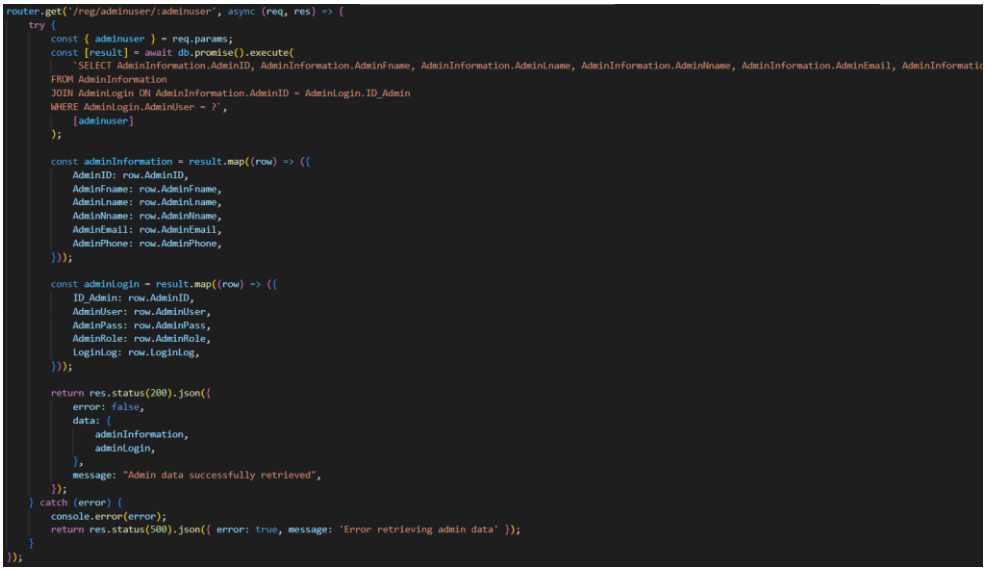
Criteria search: Search by adminrole: type localhost:3030/reg /adminrole/:adminrole in address bar to display the admin data that match input admin role in the system.



```
router.get('/reg/adminrole/:AdminRole', async (req, res) => {
    try {
        const { AdminRole } = req.params;
        const [result] = await db.promise().execute(
            `SELECT AdminInformation.AdminID, AdminInformation.AdminFname, AdminInformation.AdminLname, AdminInformation.AdminNname, AdminInformation.AdminEmail, AdminInformatio
        FROM AdminInformation
        JOIN AdminLogin ON AdminInformation.AdminID = AdminLogin.ID_Admin
        WHERE AdminLogin.AdminRole = ?`,
            [AdminRole]
        );

        const adminInformation = result.map((row) => ({
            AdminID: row.AdminID,
            AdminFname: row.AdminFname,
            AdminLname: row.AdminLname,
            AdminNname: row.AdminNname,
            AdminEmail: row.AdminEmail,
            AdminPhone: row.AdminPhone,
        }));

        const adminLogin = result.map((row) => ({
            ID_Admin: row.AdminID,
            AdminUser: row.AdminUser,
            AdminPass: row.AdminPass,
            AdminRole: row.AdminRole,
            LoginLog: row.LoginLog,
        }));

        return res.status(200).json({
            error: false,
            data: {
                adminInformation,
                adminLogin,
            },
            message: "Admin data successfully retrieved",
        });
    } catch (error) {
        console.error(error);
        return res.status(500).json({ error: true, message: 'Error retrieving admin data' });
    }
});
```


{"error":false,"data":{"adminInformation":[{"AdminID":"6488088","AdminFname":"Ittikorn","AdminLname":"Suksai","AdminNname":"Nop","AdminEmail":"Ittikorn@gmail.com","AdminPhone":"098-584-5035"},
{"AdminID":"6488038","AdminFname":"Kanyavee","AdminLname":"Likitwattanakij","AdminNname":"Babe","AdminEmail":"Kanyavee@gmail.com","AdminPhone":"066-468-2969"},
{"AdminID":"jDifDdc","AdminFname":"Shaman","AdminLname":"Moscow","AdminNname":"Kaopan","AdminEmail":"shamanking@gmail.com","AdminPhone":"285556398"},
{"AdminID":"6488124","AdminFname":"Sarttra","AdminLname":"Parsongtichol","AdminNname":"Blue","AdminEmail":"Sarttra@gmail.com","AdminPhone":"088-673-5591"},
{"AdminID":"6488067","AdminFname":"Tanawat","AdminLname":"Raweepachwarathon","AdminNname":"Pleum","AdminEmail":"Tanawat@gmail.com","AdminPhone":"092-625-2924"}],"adminLogin":
[{"ID_Admin":"6488088","AdminUser":"IttikornSukPaiLaeu","AdminPass":"8800880088","AdminRole":"Web developer","LoginLog":null},{"ID_Admin":"6488038","AdminUser":"Kanyaveeza","AdminPass":"2468102","AdminRole":"Web
developer","LoginLog":null},{"ID_Admin":"jDifDdc","AdminUser":"Kirito","AdminPass":"6255332","AdminRole":"Web developer","LoginLog":"2023-04-25T08:42:14.000Z"},
{"ID_Admin":"6488124","AdminUser":"SarttraYipPree","AdminPass":"2002Euib!","AdminRole":"Web developer","LoginLog":null},{"ID_Admin":"6488067","AdminUser":"TanawatNamTaaKabao","AdminPass":"666999666","AdminRole":"Web
developer","LoginLog":null}]},"message":"Admin data successfully retrieved"}

Criteria search: Search by adminuser: type localhost:3030/reg /adminuser/: adminuser in address bar to display the admin data that match input admin role in the system.



```
router.get('/reg/adminuser/:adminuser', async (req, res) => {
    try {
        const { adminuser } = req.params;
        const [result] = await db.promise().execute(
            `SELECT AdminInformation.AdminID, AdminInformation.AdminFname, AdminInformation.AdminLname, AdminInformation.AdminNname, AdminInformation.AdminEmail, AdminInformatic
        FROM AdminInformation
        JOIN AdminLogin ON AdminInformation.AdminID = AdminLogin.ID_Admin
        WHERE AdminLogin.AdminUser = ?`,
            [adminuser]
        );

        const adminInformation = result.map((row) => ({
            AdminID: row.AdminID,
            AdminFname: row.AdminFname,
            AdminLname: row.AdminLname,
            AdminNname: row.AdminNname,
            AdminEmail: row.AdminEmail,
            AdminPhone: row.AdminPhone,
        }));

        const adminLogin = result.map((row) => ({
            ID_Admin: row.AdminID,
            AdminUser: row.AdminUser,
            AdminPass: row.AdminPass,
            AdminRole: row.AdminRole,
            LoginLog: row.LoginLog,
        }));

        return res.status(200).json({
            error: false,
            data: {
                adminInformation,
                adminLogin,
            },
            message: "Admin data successfully retrieved",
        });
    } catch (error) {
        console.error(error);
        return res.status(500).json({ error: true, message: 'Error retrieving admin data' });
    }
});
```


{"error":false,"data":{"adminInformation":[{"AdminID":"6488038","AdminFname":"Kanyavee","AdminLname":"Likitwattanakij","AdminNname":"Babe","AdminEmail":"Kanyavee@gmail.com","AdminPhone":"066-468-2969"}],"adminLogin":
[{"ID_Admin":"6488038","AdminUser":"Kanyaveeza","AdminPass":"2468102","AdminRole":"Web developer","LoginLog":null}]},"message":"Admin data successfully retrieved"}

Insert data: Insert data can proceed by call POST method on localhost:3030/reg in postman application to insert the admin data which required attributes that matched the table in mysql to the system.

```
router.post('/reg', async (req, res) => {
    try {
        // Destructure the required fields from the request body
        const { AdminID, AdminFname, AdminLname, AdminNname, AdminEmail, AdminPhone, AdminUser, AdminPass, AdminRole } = req.body;

        // Insert a new row into the AdminInformation table
        const [AdminInformation] = await db.promise().execute(
            `INSERT INTO AdminInformation (AdminID,AdminFname, AdminLname, AdminNname, AdminEmail, AdminPhone) VALUES (?, ?, ?, ?, ?, ?)`,
            [AdminID, AdminFname, AdminLname, AdminNname, AdminEmail, AdminPhone]
        );

        // Insert a new row into the AdminLogin table, using the AdminInformation ID as a foreign key
        const [AdminLogin] = await db.promise().execute(
            `INSERT INTO AdminLogin (ID_Admin, AdminUser, AdminPass, AdminRole) VALUES (?, ?, ?, ?)`,
            [AdminID, AdminUser, AdminPass, AdminRole]
        );

        // Return a success response to the client
        return res.status(200).json({ error: false, message: 'Admin registration successful' });
    } catch (error) {
        console.error(error);
        return res.status(500).json({ error: true, message: 'Error registering admin' });
    }
});
```

```
// router.post /reg : this will add a new administrator data to the system
//Testing:  Insert a new administrator
//method: POST
//URL: localhost:3030/reg
//body: raw JSON
//Test case here:
//1st Test case:
//{
//  "AdminID": "328569",
//  "AdminFname": "James",
//  "AdminLname": "Sunderland",
//  "AdminNname": "SH2",
//  "AdminEmail": "silenthill@hotmail.com",
//  "AdminPhone": "085-669-8545",
//  "AdminUser": "JameZAP",
// "AdminPass": "nopassword",
//  "AdminRole": "Mascot"
//}
```

update data: Insert data can proceed by call PUT method on localhost:3030/reg/:id in postman application to insert the admin data which required attributes that matched the table as well as matched the id parameter in MySQL to the system.

```
router.put('/reg/:id', async (req, res) => {
    const { id } = req.params; // Get the admin ID from the request parameters

    try {
        // Destructure the required fields from the request body
        const { AdminFname, AdminLname, AdminNname, AdminEmail, AdminPhone, AdminUser, AdminPass, AdminRole } = req.body;

        // Update the row in the AdminInformation table
        await db.promise().query(
            `UPDATE AdminInformation SET AdminFname = ?, AdminLname = ?, AdminNname = ?, AdminEmail = ?, AdminPhone = ? WHERE AdminID = ?`,
            [AdminFname, AdminLname, AdminNname, AdminEmail, AdminPhone, id]
        );

        // Update the row in the AdminLogin table
        await db.promise().query(
            `UPDATE AdminLogin SET AdminUser = ?, AdminPass = ?, AdminRole = ? WHERE ID_Admin = ?`,
            [AdminUser, AdminPass, AdminRole, id]
        );

        // Return a success response to the client
        return res.status(200).json({ error: false, message: 'Admin information updated successfully' });
    } catch (error) {
        console.error(error);
        return res.status(500).json({ error: true, message: 'Error updating admin information' });
    }
});
```

```
//2nd Test Case:
//{
//   "AdminFname": "Chayanisa",
//   "AdminLname": "Chukiatphatnakij",
//   "AdminNname": "Mind",
//   "AdminEmail": "melodymind@gmail.com",
//   "AdminPhone": "061-033-8945",
//   "AdminUser": "Conductress",
//   "AdminPass": "ilovegirl",
//   "AdminRole": "Receptionist"
//}
```

Delete data: Delete data can proceed by call DELETE method on localhost:3030/reg/:id in postman application to delete the admin data which matched the id parameter in mysql on the system.

```
// router.delete /reg/:id : this will delete an administrator data that matched required ID in the system
//Testing:  delete an administrator data in required ID
//method: DELETE
//URL: localhost:3030/reg/6488067, localhost:3030/reg/6488038 , or localhost:3030/reg/6666666
router.delete('/reg/:id', (req, res) => {
    const { id } = req.params; // Get the admin ID from the request parameters

    // Delete the admin login data from the AdminLogin table
    db.query(`DELETE FROM AdminLogin WHERE ID_Admin = ?`, [id], (err, result1) => {
        if (err) {
            console.error(err);
            return res.status(500).json({ error: true, message: 'Error deleting admin data' });
        }

        // Delete the admin information from the AdminInformation table
        db.query(`DELETE FROM AdminInformation WHERE AdminID = ?`, [id], (err, result2) => {
            if (err) {
                console.error(err);
                return res.status(500).json({ error: true, message: 'Error deleting admin data' });
            }

            // If both deletes were successful, send a success response
            if (result1.affectedRows > 0 && result2.affectedRows > 0) {
                return res.status(200).json({ error: false, message: "Admin successfully deleted" });
            } else {
                return res.status(404).json({ error: true, message: "Admin not found" });
            }
        });
    });
});
```

- prod.js

prod.js is the function for administrator to modify their product in the system consist of:
no criteria search: type localhost:3030/prod in address bar to display all products data in the system.

```
// router.get /prod this will return all search result since there is no criteria
//Testing: No criteria search
//method: GET
//URL(use this for test): localhost:3030/prod , http://localhost:3030/prod
router.get('/prod', async (req, res) => {
    try {
        const [ProductData] = await db.promise().execute(
            `SELECT * FROM ProductInfomation`,
        );

        console.log(ProductData); // Log the merged data to the console
        return res.status(200).json({ error: false, data: ProductData, message: "Product successfully retrieved" });
    } catch (error) {
        console.error(error);
        return res.status(500).json({ error: true, message: 'Error retrieving user data' });
    }
});
```

{"error":false,"data":[{"ProdID":"001","ProdName":"Grilled sea bass with miso sauce","ProdType":"Main dish","ProdNeutri":"Protein","ProdFactor":null,"ProdDesc":"It's a healthy food menu. Help nourish the bones. Can help strengthen and relieve rheumatoid arthritis. Reduces high blood pressure","ProdRecipe":"Sea bass fillet (cut into pieces) 300 g., Miso 80 g, Mirin 1 tbsp ,1 egg (yolk only), Vegetable oil, grated radish, to Japanese steamed rice","ProdStep":"1.Pour the miso and mirin into a pan over low heat, stir until the consistency, add the egg yolks and stir until homogeneous. Set aside. 2.set fire to grill. Spread vegetable oil over the sieve. Apply miso sauce (as in step 1) over the sea bass. 3.Grilled over low heat until\ncooked. Served with grated radish. and Japanese steamed rice"},{"ProdID":"002","ProdName":"Clear soup with omlete","ProdType":"Appertizer","ProdNeutri":"Starch","ProdFactor":"Low calories","ProdDesc":"It's easy to eat. Suitable for everyone, especially those who want to control their diet","ProdRecipe":"2-3 eggs, onions, spring onions, coriander, seasoning powder, fish sauce, soy sauce, sugar, water","ProdStep":"1.Slice spring onions and onions into small pieces. 2.Crack an egg into a bowl, add sliced onions,spring onions and a little fish sauce\n3.Heat the pan, add a little oil when the oil is hot, pour the eggs into the pan and fry until the eggsstart to cook on both sides.Use a spatula to cut the eggs into pieces. 4.Fill it with water. Season with sugar,\nsoy sauce, seasoning powder, and finally add the remaining springonions.Turn off the stove and sprinkle with coriander leaves."},{"ProdID":"003","ProdName":"Spicy mixed vegetable soup","ProdType":"Main dish","ProdNeutri":"Fruit and vegetable","ProdFactor":"Low calories","ProdDesc":"Mixed vegetable soup is a traditional Thai dish. It focuses more on vegetables than meat and is healthy for everyone.","ProdRecipe":"1+1/2 tsp black pepper, Fingerroot 1-2 stems ,3-4 shallots, Shrimp paste 1 teaspoon, 1/3 cup finely ground dried shrimp, 3 cups chicken broth, Chinese okra, cut into bite-sized pieces 150 g, Pumpkin, cut\ninto bite-sized pieces 200 g, Courgette, cut into bite size pieces 150 g, Baby corn, sliced 150 g, 150 g of champignon Mushrooms, 200 g of fresh shrimp, basil 80 g, fish sauce","ProdStep":"1.Prepare the curry paste first. 2.Pound the pepper with the Krachai, then add the shallots, pound thoroughly, followed by the shrimp paste, pound together, then add the ground dried shrimp, pound together again until thoroughly.\n3.Put the chicken broth in the pot. Bring to a boil and add curry paste. 4.When the soup is boiling, add vegetables that are difficult to cook first. Bring to a boil again then add the rest of the vegetables.\n5.Once the water is boiling add the chinese okra, pumpkin, courgette and mushrooms. 6.When the soup is boiling again, add fresh shrimp. Season with fish sauce as desired. 7.When the mixture boils again, \nadd the basil leaves. scoop served hot."},{"ProdID":"004","ProdName":"Boiled rice with grouper","ProdType":"Appertizer","ProdNeutri":"Protein","ProdFactor":"Low calories","ProdDesc":"It's a food that's easy to digest, not sticky. Beneficial to the body. Can be eaten by all genders, ages, suitable for patients.","ProdRecipe":"Pork bone broth 1,200 ml, Young galangal, thinly sliced 3 tbsp., Shiitake mushrooms (soaked in water until soft) 100 g, 4 tablespoons soy sauce, 1 tablespoon sugar, 2 teaspoons ground pepper, Grouper fillet (cut into pieces, boiled)\n10 pieces., 300 g cooked jasmine rice., Finely chopped galangal, Braised leeks, Chopped celery","ProdStep":"1.Boil pork bone broth with young galangal until boiling, add shiitake mushrooms, season with light soy sauce,\nsugar and ground pepper, reduce heat and simmer until boiling broth. 2.Set aside. 3.Heat the water in another pot until it boils. Add jasmine rice to cook in boiling water until soft, scoop into a bowl, and\nplace blanched grouper. 4.Put the pork bone broth. sprinkle finely pounded galangal, fried garlic, and celery. Eat while hot with soybean sauce."},{"ProdID":"005","ProdName":"Spicy chicken soup with mushroom","ProdType":"Appertizer","ProdNeutri":"Protein","ProdFactor":"Low calories","ProdDesc":"A healthy, nutritious soup is suitable for all ages. It has many benefits and does not cause obesity","ProdRecipe":"Tom Yum soup consists of galangal, lemongrass, kaffir lime leaves, shallots (peeled) 4-5 heads, Bird's Eye Chilli, Straw Mushroom, Grape Tomatoes, 1 tablespoon wet tamarind, 1+1/2 tbsp fish sauce ,lemon, thinly sliced, chicken breast, parsley","ProdStep":"1.Boil water 2.Put the Tom Yum ingredients, the galangal is pounded before putting it in. Do not add too much. 3.Smash the lemongrass and cut it into small pieces, put it in until fragrant. 4.Followed by shallots and tomatoes. 5. When everything falls into place, add the sliced chicken breast,\nwait until cooked, then add the mushrooms.6.Season with tamarind juice and fish sauce. 6.Remove the pot from the stove. 7.Then tear kaffir lime leaves and sprinkle them on top. 8.Add crushed bird's eye chili to increase the spiciness. and \nsprinkle parsley and add seasoning to your liking"}],"message":"Product successfully retrieved"}

Criteria search: Search by ID type localhost:3030/prod /:id in address bar to display the products data that match input ID in the system.

```
// router.get /prod/:id this will return all of product data that have matched input ID
//Testing: Criteria Search (Search By ID) : 1st criteria
//method: GET
//URL(use this for test): localhost:3030/prod/001 , http://localhost:3030/prod/001 , localhost:3030/prod/002 , http://localhost:3030/prod/002
router.get('/prod/:id', async (req, res) => {
    try {
        const { id } = req.params;
        const [ProductData] = await db.promise().execute(
            `SELECT * FROM ProductInfomation WHERE ProdID = ?`, [id]
        );
        console.log(ProductData); // Log the merged data to the console
        return res.status(200).json({ error: false, data: ProductData, message: "Product successfully retrieved" });
    } catch (error) {
        console.error(error);
        return res.status(500).json({ error: true, message: 'Error retrieving user data' });
    }
});
```

{"error":false,"data":[{"ProdID":"002","ProdName":"Clear soup with omlete","ProdType":"Appertizer","ProdNeutri":"Starch","ProdFactor":"Low calories","ProdDesc":"It's easy to eat. Suitable for everyone, especially those who want to control their diet","ProdRecipe":"2-3 eggs, onions, spring onions, coriander, seasoning powder, fish sauce, soy sauce, sugar, water","ProdStep":"1.Slice spring onions and onions into small pieces. 2.Crack an egg into a bowl, add sliced onions,spring onions and a little fish sauce\n3.Heat the pan, add a little oil when the oil is hot, pour the eggs into the pan and fry until the eggsstart to cook on both sides.Use a spatula to cut the eggs into pieces. 4.Fill it with water. Season with sugar,\nsoy sauce, seasoning powder, and finally add the remaining springonions.Turn off the stove and sprinkle with coriander leaves."}],"message":"Product successfully retrieved"}

Criteria search: Search by nutrition: type localhost:3030/prod/nutrition/: nutrition in address bar to display the product data that match input nutrition in the system.

```
// router.get /prod/nutrition/:nutrition this will return all of product data that have matched nutrition
//Testing: Criteria Search (Search By nutrition) : 2nd criteria
//method: GET
//URL(use this for test): localhost:3030/prod/nutrition/Protein , http://localhost:3030/prod/nutrition/Protein , localhost:3030/prod/nutrition/Starch , http://localhost:3030/prod/nutrition/Starch
router.get('/prod/nutrition/:nutrition', async (req, res) => {
    try {
        const { nutrition } = req.params;
        const [ProductData] = await db.promise().execute(
            `SELECT * FROM ProductInfomation WHERE ProdNeutri = ?`, [nutrition]
        );
        console.log(ProductData); // Log the merged data to the console
        return res.status(200).json({ error: false, data: ProductData, message: "Product successfully retrieved" });
    } catch (error) {
        console.error(error);
        return res.status(500).json({ error: true, message: 'Error retrieving user data' });
    }
});
```

{"error":false,"data":[{"ProdID":"002","ProdName":"Clear soup with omlete","ProdType":"Appertizer","ProdNeutri":"Starch","ProdFactor":"Low calories","ProdDesc":"It's easy to eat. Suitable for everyone, especially those who want to control their diet","ProdRecipe":"2-3 eggs, onions, spring onions, coriander, seasoning powder, fish sauce, soy sauce, sugar, water","ProdStep":"1.Slice spring onions and onions into small pieces. 2.Crack an egg into a bowl, add sliced onions,spring onions and a little fish sauce\n3.Heat the pan, add a little oil when the oil is hot, pour the eggs into the pan and fry until the eggsstart to cook on both sides.Use a spatula to cut the eggs into pieces. 4.Fill it with water. Season with sugar,\nsoy sauce, seasoning powder, and finally add the remaining springonions.Turn off the stove and sprinkle with coriander leaves."}],"message":"Product successfully retrieved"}

Criteria search: Search by Food Type: type localhost:3030/prod/type/: type in address bar to display the product data that match input food type in the system.

```
// router.get /prod/type/:type this will return all of product data that have matched nutrition
//Testing: Criteria Search (Search By nutrition) : 2nd criteria
//method: GET
//URL(use this for test): localhost:3030/prod/type/Appertizer, http://localhost:3030/prod/type/Appertizer, localhost:3030/prod/type/Main Dish , http://localhost:3030/prod/type/Main Dish
router.get('/prod/type/:type', async (req, res) => {
    try {
        const { type } = req.params;
        const [ProductData] = await db.promise().execute(
            `SELECT * FROM ProductInfomation WHERE ProdType = ?`, [type]
        );
        console.log(ProductData); // Log the merged data to the console
        return res.status(200).json({ error: false, data: ProductData, message: "Product successfully retrieved" });
    } catch (error) {
        console.error(error);
        return res.status(500).json({ error: true, message: 'Error retrieving user data' });
    }
});
```

{"error":false,"data":[{"ProdID":"001","ProdName":"Grilled sea bass with miso sauce","ProdType":"Main dish","ProdNeutri":"Protein","ProdFactor":null,"ProdDesc":"It's a healthy food menu. Help nourish the bones. Can help strengthen and relieve rheumatoid arthritis. Reduces high blood pressure","ProdRecipe":"Sea bass fillet (cut into pieces) 300 g., Miso 80 g, Hirin 1 tbsp ,1 egg (yolk only), Vegetable oil, grated radish, to Japanese steamed rice","ProdStep":"1.Pour the miso and mirin into a pan over low heat, stir until the consistency, add the egg yolks and stir until homogeneous. Set aside. 2.set fire to grill. Spread vegetable oil over the sieve. Apply miso sauce (as in step 1) over the sea bass. 3.Grilled over low heat until\ncooked. Served with grated radish. and Japanese steamed rice"},{"ProdID":"003","ProdName":"Spicy mixed vegetable soup","ProdType":"Main dish","ProdNeutri":"Fruit and vegetable","ProdFactor":"Low calories","ProdDesc":"Mixed vegetable soup is a traditional Thai dish. It focuses more on vegetables than meat and is healthy for everyone.","ProdRecipe":"1+1/2 tsp black pepper, Fingerroot 1-2 stems ,3-4 shallots, Shrimp paste 1 teaspoon, 1/3 cup finely ground dried shrimp, 3 cups chicken broth, Chinese okra, cut into bite-sized pieces 150 g, Pumpkin, cut\ninto bite-sized pieces 200 g, Courgette, cut into bite size pieces 150 g, Baby corn, sliced 150 g, 150 g of champignon Mushrooms, 200 g of fresh shrimp, basil 80 g, fish sauce","ProdStep":"1.Prepare the curry paste first. 2.Pound the pepper with the Krachai, then add the shallots, pound thoroughly, followed by the shrimp paste, pound together, then add the ground dried shrimp, pound together again until thoroughly.\n3.Put the chicken broth in the pot. Bring to a boil and add curry paste. 4.When the soup is boiling, add vegetables that are difficult to cook first. Bring to a boil again then add the rest of the vegetables.\n5.Once the water is boiling add the chinese okra, pumpkin, courgette and mushrooms. 6.When the soup is boiling again, add fresh shrimp. Season with fish sauce as desired. 7.When the mixture boils again, \nadd the basil leaves. scoop served hot."}],"message":"Product successfully retrieved"}

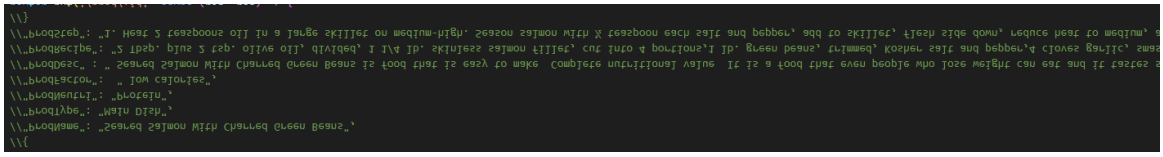
Insert data: Insert data can proceed by call POST method on localhost:3030/prod in postman application to insert the product data which required attributes that matched the table in mysql to the system.

```
router.post('/prod', async (req, res) => {
    try {
        const { ProdID, ProdName, ProdType, ProdNeutri, ProdFactor, ProdDesc, ProdRecipe, ProdStep } = req.body; // Destructure the required fields from the request body

        // Insert a new row into the ProductInfomation table
        const [ProductData] = await db.promise().execute(
            `INSERT INTO ProductInfomation (ProdID, ProdName, ProdType, ProdNeutri, ProdFactor, ProdDesc, ProdRecipe, ProdStep) VALUES (?, ?, ?, ?, ?, ?, ?, ?)`,
            [ProdID, ProdName, ProdType, ProdNeutri, ProdFactor, ProdDesc, ProdRecipe, ProdStep]
        );

        return res.status(200).json({ error: false, message: "Product successfully inserted" });
    } catch (error) {
        console.error(error);
        return res.status(500).json({ error: true, message: 'Error inserting product data' });
    }
});
```

```
// router.post /prod : this will add a new product data to the system
//Testing:  Insert a new product
//method: POST
//URL: localhost:3030/prod
//body: raw JSON
//Test case here:
//1st Test case:
//{
// "ProdID": "054",
// "ProdName": "Tom yum kung",
// "ProdType": "Main Dish",
// "ProdNeutri": "Fruit and vegetables",
// "ProdFactor": "Low calories",
// "ProdDesc" : " This hot and sour shrimp soup is an explosion of flavour and is also said to have medicinal prope
// "ProdRecipe": "15 shrimps or tiger prawns ,100g of mushroom ,600g of chicken stock ,1 tbsp chopped lemongrass ,1
// "ProdStep": "1.Clean the shrimp and remove the black thread (the intestine) from the shrimp by cutting it down t
// }
```

Update data: Insert data can proceed by call PUT method on localhost:3030/prod/:id in postman application to insert the product data which required attributes that matched the table as well as matched the id parameter in mysql to the system.

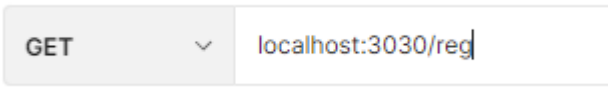
```
router.put('/prod/:id', async (req, res) => {
    try {
        const { ProdName, ProdType, ProdNeutri, ProdFactor, ProdDesc, ProdRecipe, ProdStep } = req.body; // Destructure the required fields from the request body
        const { id } = req.params; // Get the product ID from the request params

        // Update the row in the ProductInfomation table with the given ID
        const [ProductData] = await db.promise().execute(
            `UPDATE ProductInfomation SET ProdName = ?, ProdType = ?, ProdNeutri = ?, ProdFactor = ?, ProdDesc = ?, ProdRecipe = ?, ProdStep = ? WHERE ProdID = ?`,
            [ProdName, ProdType, ProdNeutri, ProdFactor, ProdDesc, ProdRecipe, ProdStep, id]
        );

        return res.status(200).json({ error: false, message: "Product successfully updated" });
    } catch (error) {
        console.error(error);
        return res.status(500).json({ error: true, message: 'Error updating product data' });
    }
});
```



Delete data: Delete data can proceed by call DELETE method on localhost:3030/prod/:id in postman application to delete the product data which matched the id parameter in mysql on the system.

```
// router.delete /prod/:id : this will delete a product data that matched required ID in the system
//Testing:  delete a product data in required ID
//method: DELETE
//URL: localhost:3030/prod/001, localhost:3030/prod/005 , or localhost:3030/prod/003
router.delete('/prod/:id', (req, res) => {
        const { id } = req.params;
        db.query(`DELETE FROM ProductInfomation WHERE ProdID = ?`, [id], (err, result1) => {
            if (err) {
                console.error(err);
                return res.status(500).json({ error: true, message: 'Error deleting product data' });
            }

            // If both deletes were successful, send a success response
            if (result1.affectedRows > 0) {
                return res.status(200).json({ error: false, message: "Product successfully deleted" });
            } else {
                return res.status(404).json({ error: true, message: "Product not found" });
            }

        });
});
```

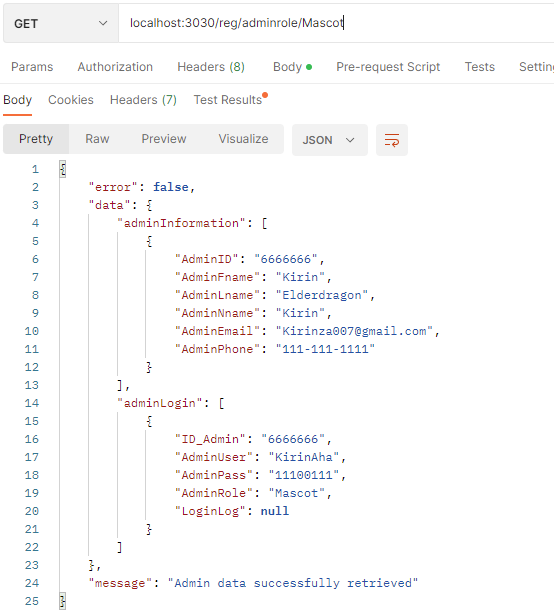
5. The testing results of web services using Postman or any program for testing web service. This is the testing result which running through postman application.
   admin.js
   no criteria search: GET localhost:3030/reg.




```
{
    "error": false,
    "data": {
        "adminInformation": [
            {
                "AdminID": "6488038",
                "AdminFname": "Kanyavee",
                "AdminLname": "Likitwattanakij",
                "AdminNname": "Babe",
                "AdminEmail": "Kanyavee@gmail.com",
                "AdminPhone": "066-468-2969"
            },
            {
                "AdminID": "6488066",
                "AdminFname": "Arada",
                "AdminLname": "Raksapatcharawong",
                "AdminNname": "Aida",
                "AdminEmail": "Arada@gmail.com",
                "AdminPhone": "000-000-0000"
            },
            {
                "AdminID": "6488067",
                "AdminFname": "Tanawat",
                "AdminLname": "Raweepachwarathon",
                "AdminNname": "Pleum",
                "AdminEmail": "Tanawat@gmail.com",
                "AdminPhone": "092-625-2924"
            },
            {
                "AdminID": "6488088",
                "AdminFname": "Ittikorn",
                "AdminLname": "Suksai",
                "AdminNname": "Nop",
                "AdminEmail": "Ittikorn@gmail.com",
                "AdminPhone": "098-584-5035"
            },
            {
                "AdminID": "6488124",
                "AdminFname": "Sarttra",
```

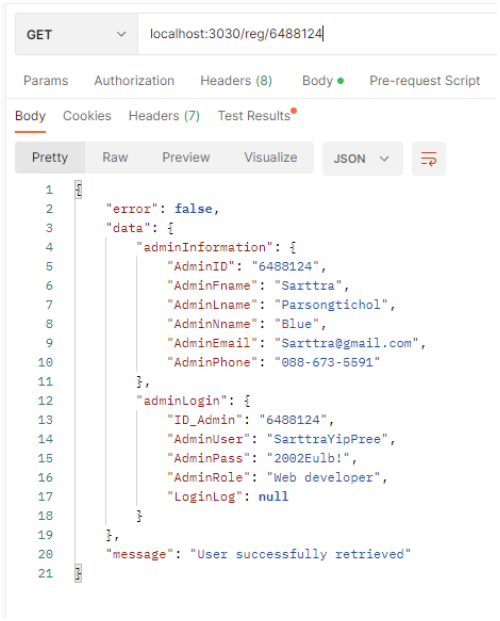
```
"adminLogin": [
    {
        "ID_Admin": "6488066",
        "AdminUser": "AradaAbsent",
        "AdminPass": "6600660066",
        "AdminRole": "Web designer",
        "LoginLog": null
    },
    {
        "ID_Admin": "6488088",
        "AdminUser": "IttikornSukPaiLaeu",
        "AdminPass": "8800880088",
        "AdminRole": "Web developer",
        "LoginLog": null
    },
    {
        "ID_Admin": "6488038",
        "AdminUser": "Kanyaveeza",
        "AdminPass": "2468102",
        "AdminRole": "Web developer",
        "LoginLog": null
    },
    {
        "ID_Admin": "6666666",
        "AdminUser": "KirinAha",
        "AdminPass": "11100111",
        "AdminRole": "Mascot",
        "LoginLog": null
    },
    {
        "ID_Admin": "6488124",
        "AdminUser": "SarttraYipPree",
        "AdminPass": "2002Eulb!",
        "AdminRole": "Web developer",
        "LoginLog": null
    },
```


Criteria search: search by Admin role: GET localhost:3030/reg/adminrole/: adminrole

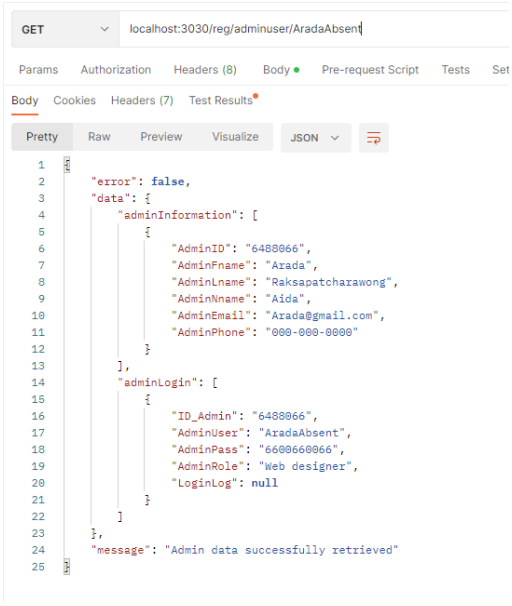



```
{
    "error": false,
    "data": {
        "adminInformation": [
            {
                "AdminID": "6666666",
                "AdminFname": "Kirin",
                "AdminLname": "Elderdragon",
                "AdminNname": "Kirin",
                "AdminEmail": "Kirinza007@gmail.com",
                "AdminPhone": "111-111-1111"
            }
        ],
        "adminLogin": [
            {
                "ID_Admin": "6666666",
                "AdminUser": "KirinAha",
                "AdminPass": "11100111",
                "AdminRole": "Mascot",
                "LoginLog": null
            }
        ]
    },
    "message": "Admin data successfully retrieved"
}
```

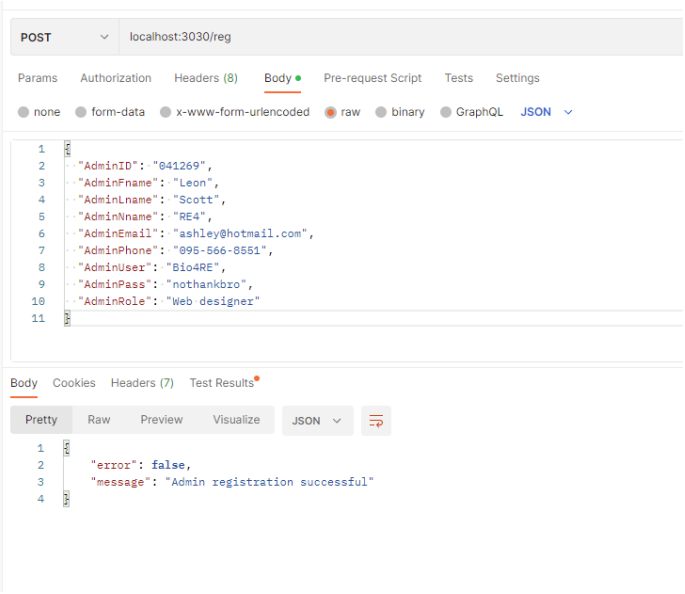
Criteria search: search by ID: GET localhost:3030/reg/:id.



Criteria search: search by Admin username: GET localhost:3030/reg/adminuser/: adminuser



Insert data: POST localhost:3030/reg.



When using GET localhost:3030/reg there will be the new admin data that was recently inserted.


```
{
    "error": false,
    "data": {
        "adminInformation": [
            {
                "AdminID": "041269",
                "AdminFname": "Leon",
                "AdminLname": "Scott",
                "AdminNname": "RE4",
                "AdminEmail": "ashley@hotmail.com",
                "AdminPhone": "095-566-8551"
            },
            {
                "AdminID": "6488038",
                "AdminFname": "Kanyavee",
                "AdminLname": "Likitwattanakij",
```

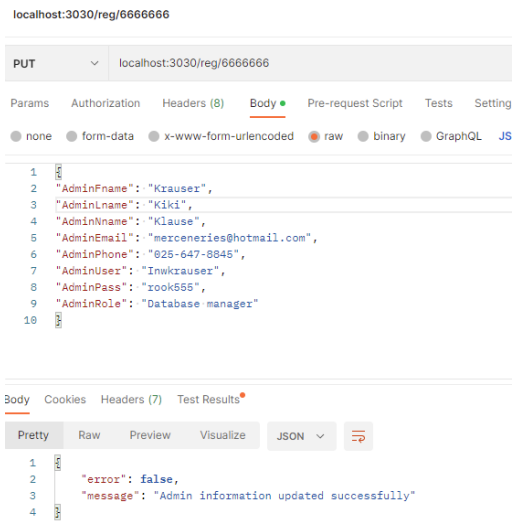
```
"adminLogin": [
    {
        "ID_Admin": "6488066",
        "AdminUser": "AradaAbsent",
        "AdminPass": "6600660066",
        "AdminRole": "Web designer",
        "LoginLog": null
    },
    {
        "ID_Admin": "041269",
        "AdminUser": "Bio4RE",
        "AdminPass": "nothankbro",
        "AdminRole": "Web designer",
        "LoginLog": null
    },
```


Update data: we use PUT localhost3030:/reg/:id with the remaining administrator. For this case, we want to update the data of Admin Kirin which has ID = '6666666'
This is the old data.

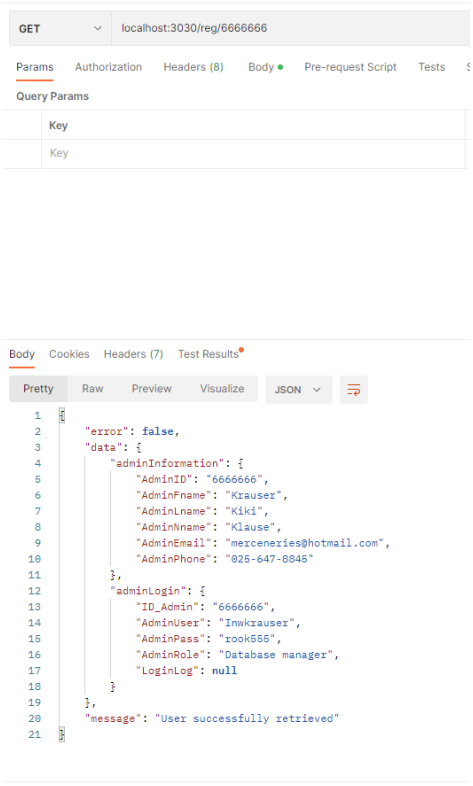

```
{
    "ID_Admin": "6666666",
    "AdminUser": "KirinAha",
    "AdminPass": "11100111",
    "AdminRole": "Mascot",
    "LoginLog": null
},
```

```
{
    "AdminID": "6666666",
    "AdminFname": "Kirin",
    "AdminLname": "Elderdragon",
    "AdminNname": "Kirin",
    "AdminEmail": "Kirinza007@gmail.com",
    "AdminPhone": "111-111-1111"
}
```

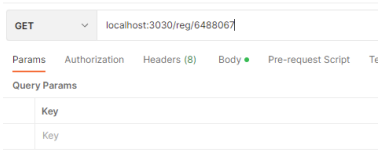
Then we use PUT localhost3030:/reg/6666666 by following data, when it done the success message will show like this.

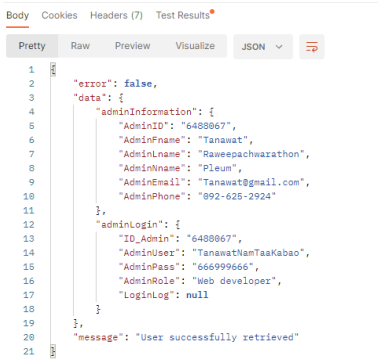


When we check by use GET localhost:3030/reg/6666666 the data inside has been replaced with the new ones.

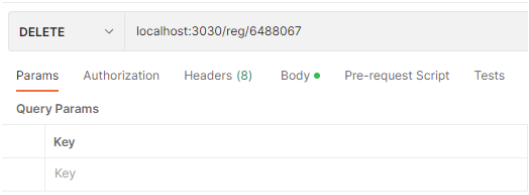


Delete data: we use DELETE localhost3030:/reg/:id with the remaining administrator. For this case, we want to dalete the data of Admin Tanawat which has ID = '6488067'





After that we use DELETE on this admin.

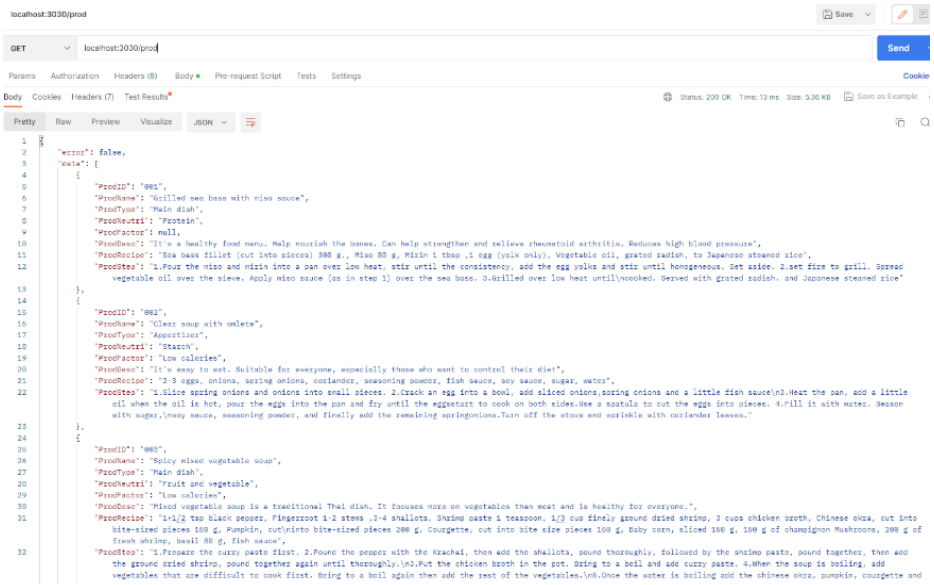


And when we check by GET localhost:3030/reg there is no admin Tanawat in the admin data result.


```
,
{
    "AdminID": "6488066",
    "AdminFname": "Arada",
    "AdminLname": "Raksapatcharawong",
    "AdminNname": "Aida",
    "AdminEmail": "Arada@gmail.com",
    "AdminPhone": "000-000-0000"
},
{
    "AdminID": "6488088",
    "AdminFname": "Ittikorn",
    "AdminLname": "Suksai",
    "AdminNname": "Nop",
    "AdminEmail": "Ittikorn@gmail.com",
    "AdminPhone": "098-584-5035"
},
```

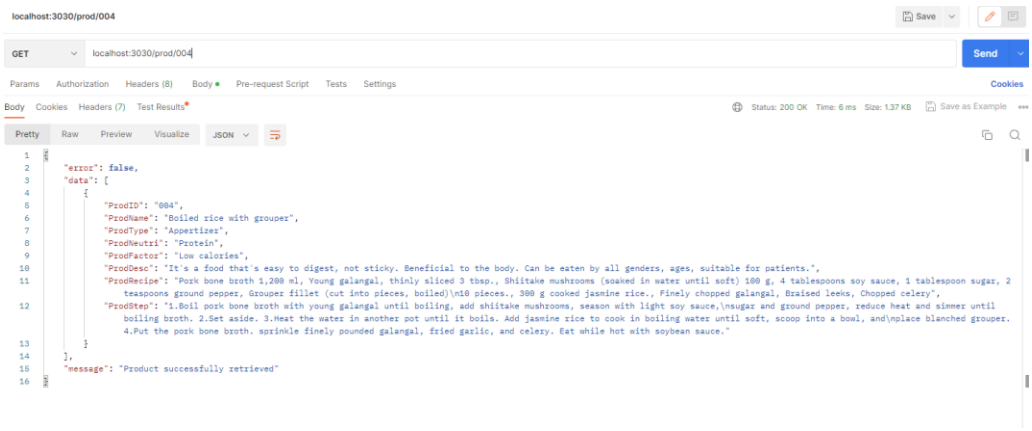
```
"adminLogin": [
    {
        "ID_Admin": "6488066",
        "AdminUser": "AradaAbsent",
        "AdminPass": "6600660066",
        "AdminRole": "Web designer",
        "LoginLog": null
    },
    {
        "ID_Admin": "041269",
        "AdminUser": "Bio4RE",
        "AdminPass": "nothankbro",
        "AdminRole": "Web designer",
        "LoginLog": null
    },
    {
        "ID_Admin": "6666666",
        "AdminUser": "Inwkrauser",
        "AdminPass": "rook555",
        "AdminRole": "Database manager",
        "LoginLog": null
    },
    {
        "ID_Admin": "6488088",
        "AdminUser": "IttikornSukPaiLaeu",
        "AdminPass": "8800880088",
        "AdminRole": "Web developer",
        "LoginLog": null
    },
    {
        "ID_Admin": "6488038",
        "AdminUser": "Kanyaveeza",
        "AdminPass": "2468102",
        "AdminRole": "Web developer",
        "LoginLog": null
    },
    {
        "ID_Admin": "6488124",
```


prod.js
no criteria search: GET localhost:3030/prod.



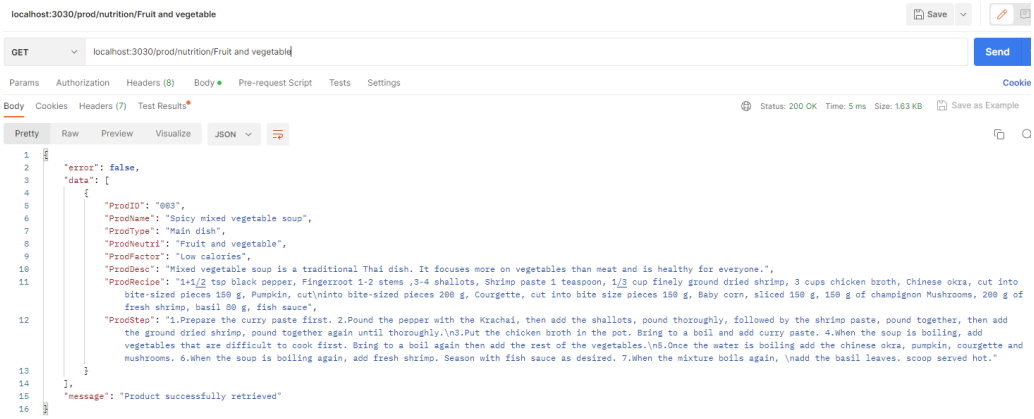
```
                the ground dried shrimp, pound together again until thoroughly.\n3.Put the chicken broth in the pot. Bring to a boil and add curry paste. 4.When the soup is boiling, add
                vegetables that are difficult to cook first. Bring to a boil again then add the rest of the vegetables.\n5.Once the water is boiling add the chinese okra, pumpkin, courgette and
                mushrooms. 6.When the soup is boiling again, add fresh shrimp. Season with fish sauce as desired. 7.When the mixture boils again, \nadd the basil leaves. scoop served hot."
        },
        {
            "ProdID": "004",
            "ProdName": "Boiled rice with grouper",
            "ProdType": "Appertizer",
            "ProdNeutri": "Protein",
            "ProdFactor": "Low calories",
            "ProdDesc": "It's a food that's easy to digest, not sticky. Beneficial to the body. Can be eaten by all genders, ages, suitable for patients.",
            "ProdRecipe": "Pork bone broth 1,200 ml, Young galangal, thinly sliced 3 tbsp., Shiitake mushrooms (soaked in water until soft) 100 g, 4 tablespoons soy sauce, 1 tablespoon sugar, 2
                teaspoons ground pepper, Grouper fillet (cut into pieces, boiled)\n10 pieces., 300 g cooked jasmine rice., Finely chopped galangal, Braised leeks, Chopped celery",
            "ProdStep": "1.Boil pork bone broth with young galangal until boiling, add shiitake mushrooms, season with light soy sauce,\nsugar and ground pepper, reduce heat and simmer until
                boiling broth. 2.Set aside. 3.Heat the water in another pot until it boils. Add jasmine rice to cook in boiling water until soft, scoop into a bowl, and\nplace blanched grouper.
                4.Put the pork bone broth. sprinkle finely pounded galangal, fried garlic, and celery. Eat while hot with soybean sauce."
        },
        {
            "ProdID": "005",
            "ProdName": "Spicy chicken soup with mushroom",
            "ProdType": "Appertizer",
            "ProdNeutri": "Protein",
            "ProdFactor": "Low calories",
            "ProdDesc": "A healthy, nutritious soup is suitable for all ages. It has many benefits and does not cause obesity",
            "ProdRecipe": "Tom Yum soup consists of galangal, lemongrass, kaffir lime leaves, shallots (peeled) 4-5 heads, Bird's Eye Chilli, Straw Mushroom, Grape Tomatoes, 1 tablespoon wet
                tamarind, 1+1/2 tbsp fish sauce ,lemon, thinly sliced, chicken breast, parsley",
            "ProdStep": "1.Boil water 2.Put the Tom Yum ingredients, the galangal is pounded before putting it in. Do not add too much. 3.Smash the lemongrass and cut it into small pieces, put
                it in until fragrant. 4.Followed by shallots and tomatoes. 5. When everything falls into place, add the sliced chicken breast,\nwait until cooked, then add the mushrooms.6.
                Season with tamarind juice and fish sauce. 6.Remove the pot from the stove. 7.Then tear kaffir lime leaves and sprinkle them on top. 8.Add crushed bird's eye chili to increase
                the spiciness. and \nsprinkle parsley and add seasoning to your liking"
        }
    ],
    "message": "Product successfully retrieved"
}
```


Criteria search: search by ID: GET localhost:3030/prod/:id.



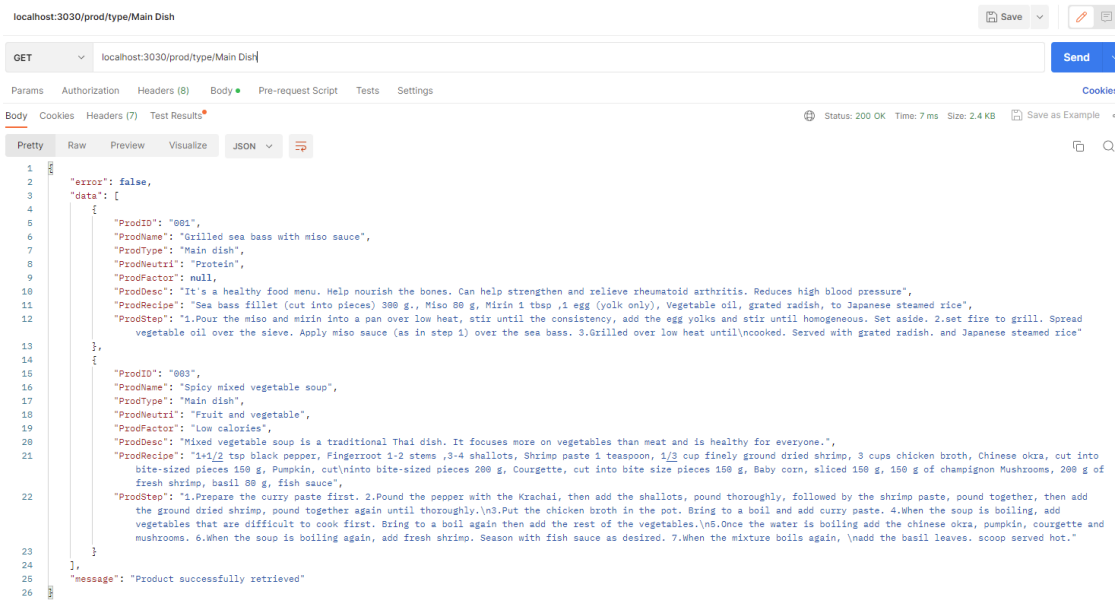

```
{
    "error": false,
    "data": [
        {
            "ProdID": "004",
            "ProdName": "Boiled rice with grouper",
            "ProdType": "Appertizer",
            "ProdNeutri": "Protein",
            "ProdFactor": "Low calories",
            "ProdDesc": "It's a food that's easy to digest, not sticky. Beneficial to the body. Can be eaten by all genders, ages, suitable for patients.",
            "ProdRecipe": "Pork bone broth 1,200 ml, Young galangal, thinly sliced 3 tbsp., Shiitake mushrooms (soaked in water until soft) 100 g, 4 tablespoons soy sauce, 1 tablespoon sugar, 2
                teaspoons ground pepper, Grouper fillet (cut into pieces, boiled)\n10 pieces., 300 g cooked jasmine rice., Finely chopped galangal, Braised leeks, Chopped celery",
            "ProdStep": "1.Boil pork bone broth with young galangal until boiling, add shiitake mushrooms, season with light soy sauce,\nsugar and ground pepper, reduce heat and simmer until
                boiling broth. 2.Set aside. 3.Heat the water in another pot until it boils. Add jasmine rice to cook in boiling water until soft, scoop into a bowl, and\nplace blanched grouper.
                4.Put the pork bone broth. sprinkle finely pounded galangal, fried garlic, and celery. Eat while hot with soybean sauce."
        }
    ],
    "message": "Product successfully retrieved"
}
```


Criteria search: search by Product nutritious: GET localhost:3030/prod/nutrition/: nutrition.

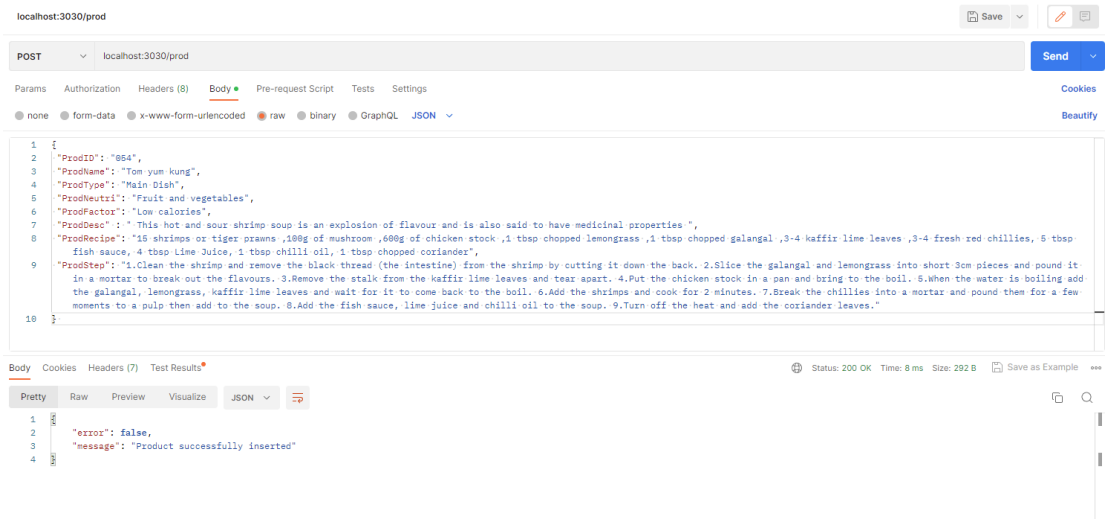



```
{
    "error": false,
    "data": [
        {
            "ProdID": "003",
            "ProdName": "Spicy mixed vegetable soup",
            "ProdType": "Main dish",
            "ProdNeutri": "Fruit and vegetable",
            "ProdFactor": "Low calories",
            "ProdDesc": "Mixed vegetable soup is a traditional Thai dish. It focuses more on vegetables than meat and is healthy for everyone.",
            "ProdRecipe": "1+1/2 tsp black pepper, Fingerroot 1-2 stems ,3-4 shallots, Shrimp paste 1 teaspoon, 1/3 cup finely ground dried shrimp, 3 cups chicken broth, Chinese okra, cut into
                bite-sized pieces 150 g, Pumpkin, cut\ninto bite-sized pieces 200 g, Courgette, cut into bite size pieces 150 g, Baby corn, sliced 150 g, 150 g of champignon Mushrooms, 200 g of
                fresh shrimp, basil 80 g, fish sauce",
            "ProdStep": "1.Prepare the curry paste first. 2.Pound the pepper with the Krachai, then add the shallots, pound thoroughly, followed by the shrimp paste, pound together, then add
                the ground dried shrimp, pound together again until thoroughly.\n3.Put the chicken broth in the pot. Bring to a boil and add curry paste. 4.When the soup is boiling, add
                vegetables that are difficult to cook first. Bring to a boil again then add the rest of the vegetables.\n5.Once the water is boiling add the chinese okra, pumpkin, courgette and
                mushrooms. 6.When the soup is boiling again, add fresh shrimp. Season with fish sauce as desired. 7.When the mixture boils again, \nadd the basil leaves. scoop served hot."
        }
    ],
    "message": "Product successfully retrieved"
}
```

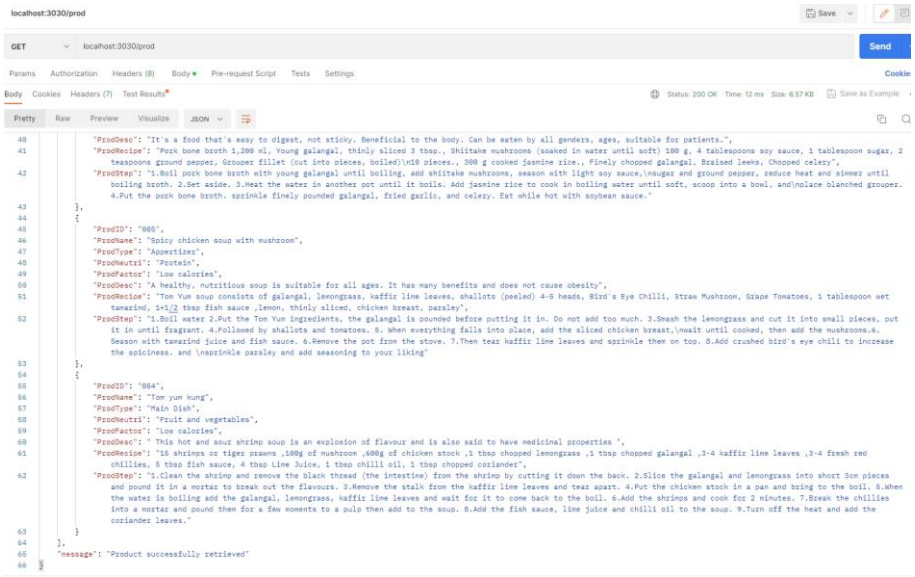
Criteria search: search by Product Type: GET localhost:3030/prod/type/: type.



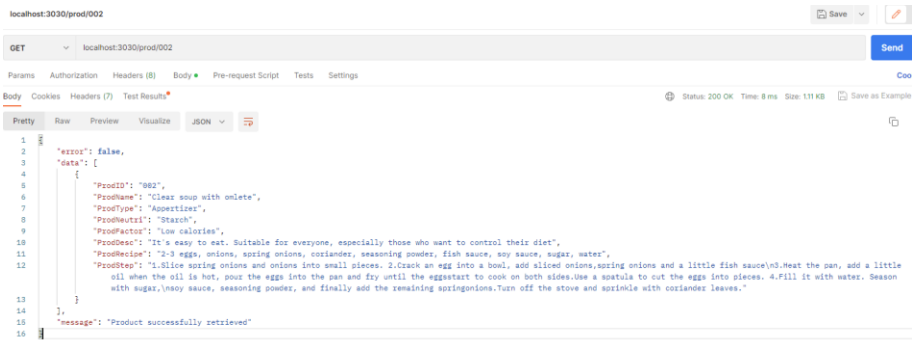
Insert data: POST localhost:3030/prod



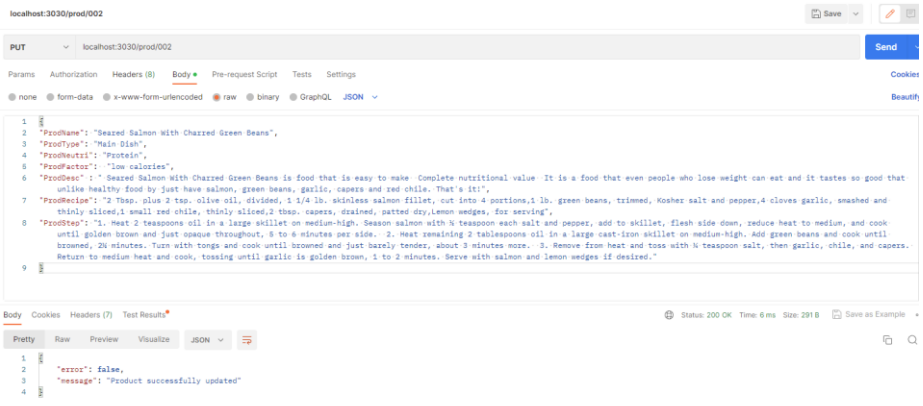
When using GET localhost:3030/prod there will be the new product data that recently insert.



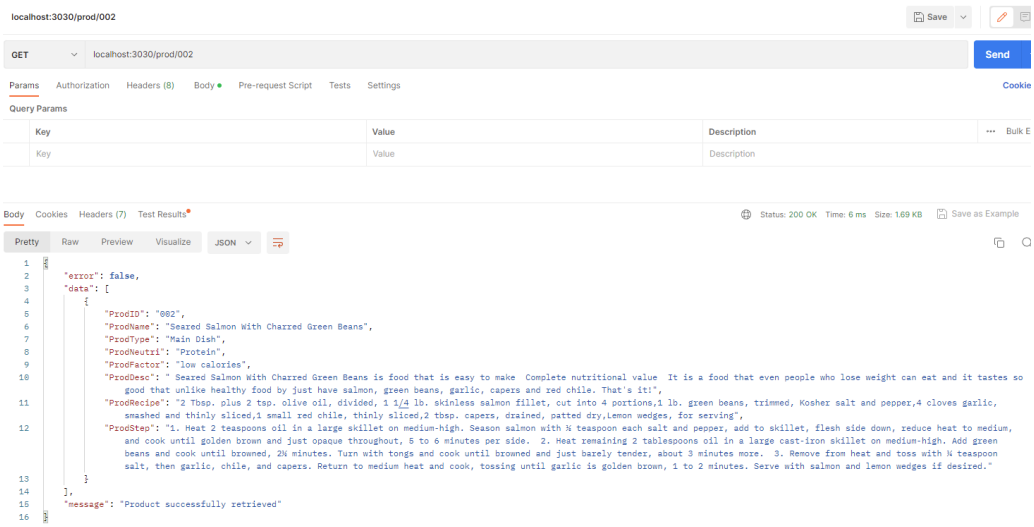
Update data: we use PUT localhost3030:/prod/:id with the remaining product. For this case, we want to update the data of Clear soup with omelet which has ID = '002'



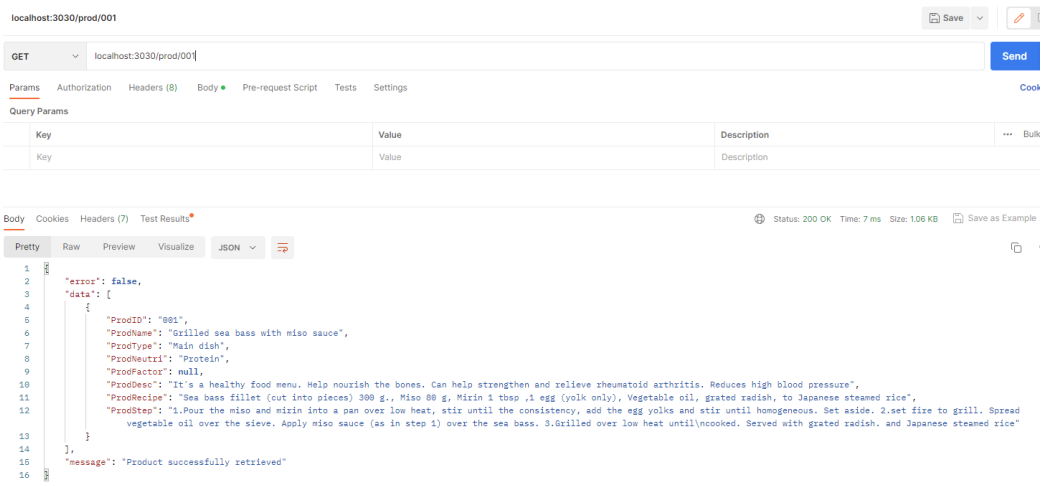
Then we use PUT localhost3030:/reg/002 by following data, when it done the success message will show like this.



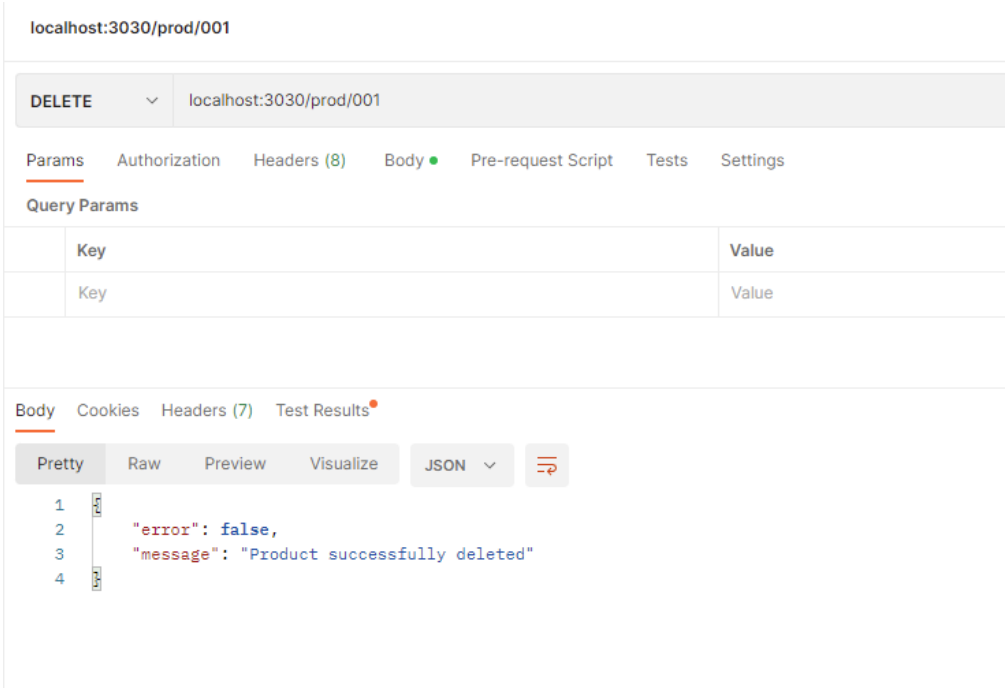
When we check by using GET localhost:3030/prod/002 the data inside has been replaced with the new ones.



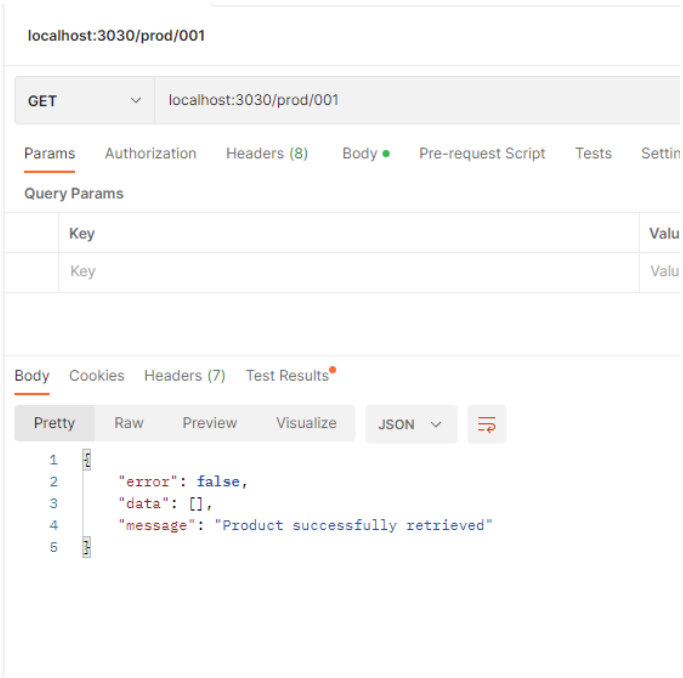
Delete data: we use DELETE localhost3030:/prod/:id with the remaining product. For this case, we want to delete the data of Grilled Sea bass with miso sauce which has ID = '001'



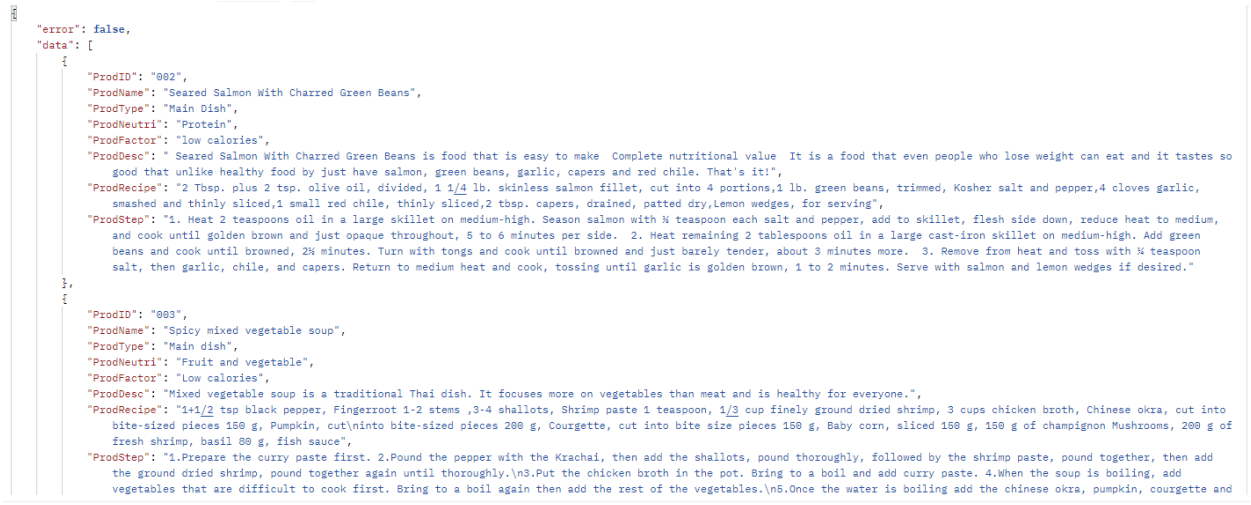
After that we use DELETE on this product.



When we use GET localhost:3030/prod/002 to call admin Grilled Sea bass with miso sauce data it provides an empty array.



And when we check by GET localhost:3030/prod there is noGrilled Sea bass with miso sauce in the admin data result.




```
"error": false,
"data": [
    {
        "ProdID": "002",
        "ProdName": "Seared Salmon With Charred Green Beans",
        "ProdType": "Main Dish",
        "ProdNeutri": "Protein",
        "ProdFactor": "low calories",
        "ProdDesc": " Seared Salmon With Charred Green Beans is food that is easy to make  Complete nutritional value  It is a food that even people who lose weight can eat and it tastes so good that unlike healthy food by just have salmon, green beans, garlic, capers and red chile. That's it!",
        "ProdRecipe": "2 Tbsp. plus 2 tsp. olive oil, divided, 1 1/4 lb. skinless salmon fillet, cut into 4 portions,1 lb. green beans, trimmed, Kosher salt and pepper,4 cloves garlic, smashed and thinly sliced,1 small red chile, thinly sliced,2 tbsp. capers, drained, patted dry,Lemon wedges, for serving",
        "ProdStep": "1. Heat 2 teaspoons oil in a large skillet on medium-high. Season salmon with ¾ teaspoon each salt and pepper, add to skillet, flesh side down, reduce heat to medium, and cook until golden brown and just opaque throughout, 5 to 6 minutes per side.  2. Heat remaining 2 tablespoons oil in a large cast-iron skillet on medium-high. Add green beans and cook until browned, 2½ minutes. Turn with tongs and cook until browned and just barely tender, about 3 minutes more.  3. Remove from heat and toss with ¼ teaspoon salt, then garlic, chile, and capers. Return to medium heat and cook, tossing until garlic is golden brown, 1 to 2 minutes. Serve with salmon and lemon wedges if desired."
    },
    {
        "ProdID": "003",
        "ProdName": "Spicy mixed vegetable soup",
        "ProdType": "Main dish",
        "ProdNeutri": "Fruit and vegetable",
        "ProdFactor": "Low calories",
        "ProdDesc": "Mixed vegetable soup is a traditional Thai dish. It focuses more on vegetables than meat and is healthy for everyone.",
        "ProdRecipe": "1+1/2 tsp black pepper, Fingerroot 1-2 stems ,3-4 shallots, Shrimp paste 1 teaspoon, 1/3 cup finely ground dried shrimp, 3 cups chicken broth, Chinese okra, cut into bite-sized pieces 150 g, Pumpkin, cut\ninto bite-sized pieces 200 g, Courgette, cut into bite size pieces 150 g, Baby corn, sliced 150 g, 150 g of champignon Mushrooms, 200 g of fresh shrimp, basil 80 g, fish sauce",
        "ProdStep": "1.Prepare the curry paste first. 2.Pound the pepper with the Krachai, then add the shallots, pound thoroughly, followed by the shrimp paste, pound together, then add the ground dried shrimp, pound together again until thoroughly.\n3.Put the chicken broth in the pot. Bring to a boil and add curry paste. 4.When the soup is boiling, add vegetables that are difficult to cook first. Bring to a boil again then add the rest of the vegetables.\n5.Once the water is boiling add the chinese okra, pumpkin, courgette and
```