WEB PROGRAMMING
FINAL REPORT

Members
Pimsuang Kanjanatawewat 6288021
Nithinee Trironnarith 6288055
Phuwanat Meeyutem 6288095
Nawadol Kuchaipoom 6288098

ITCS212 Web Programming

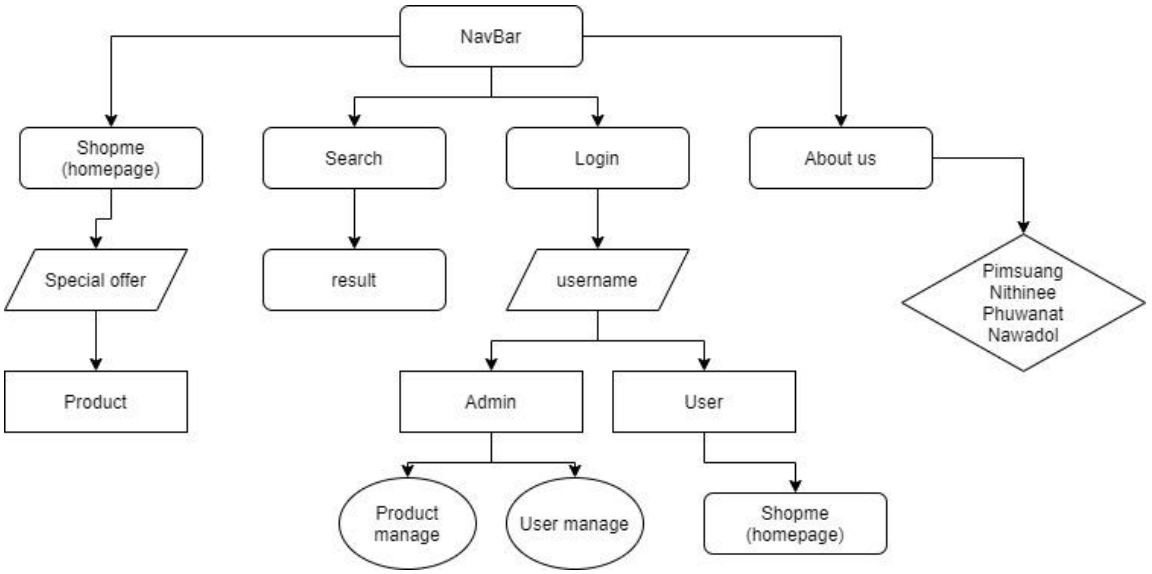
Faculty of Information and Communication Technology

Mahidol University

2020

Overview

Navigation Diagram

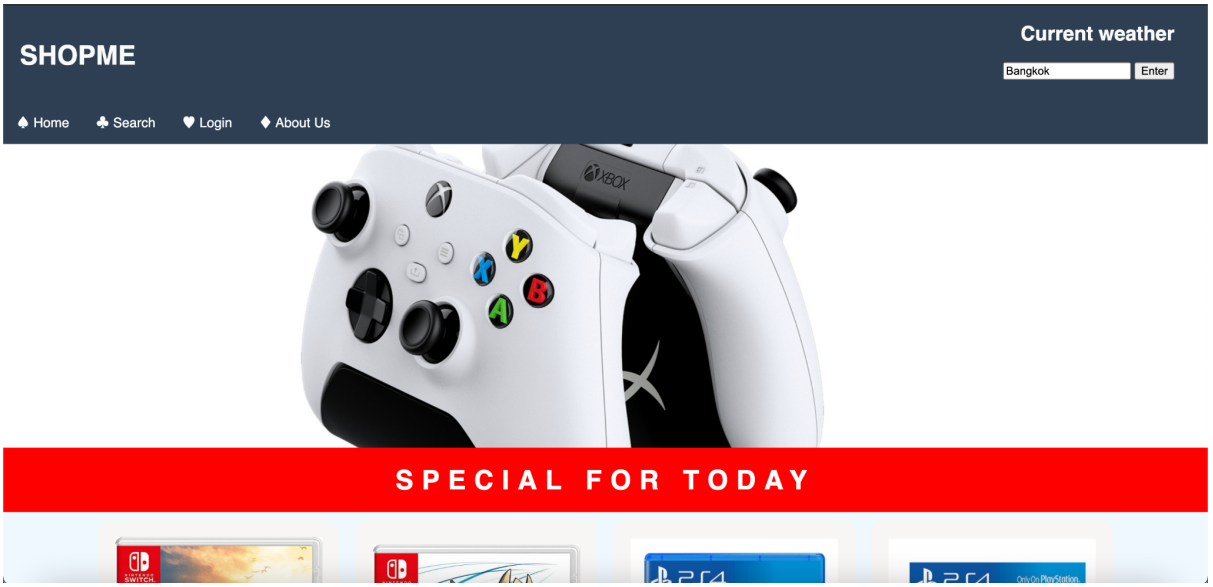


Phase1

We created 5 pages using html and CSS.

Homepage

This page shows the special offer product.

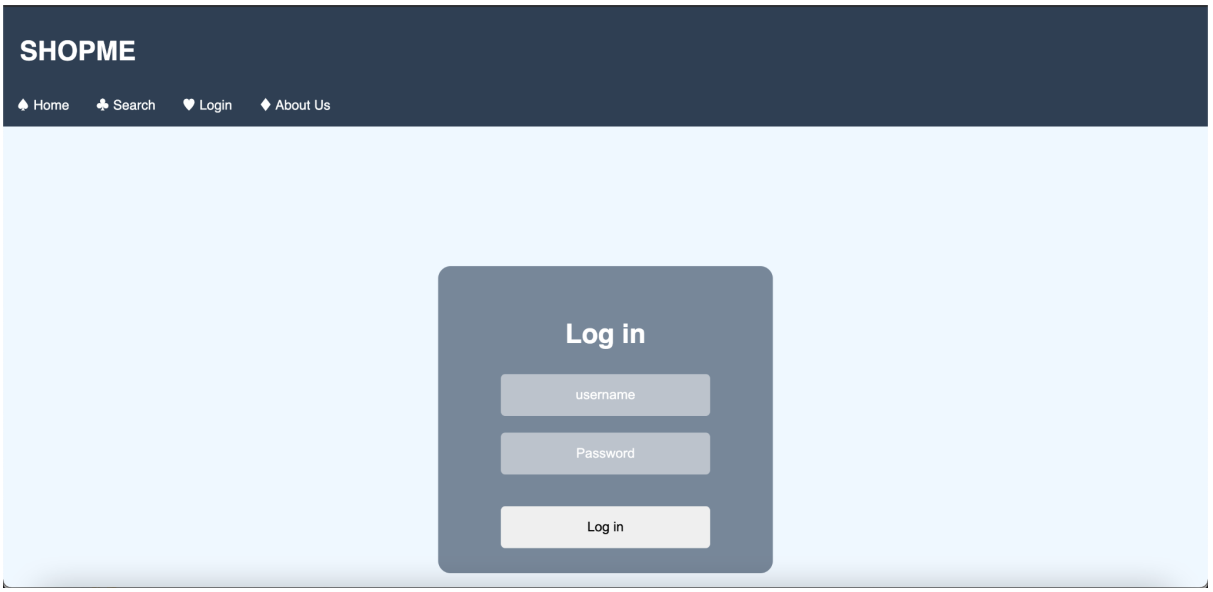


search page

This page is for searching.

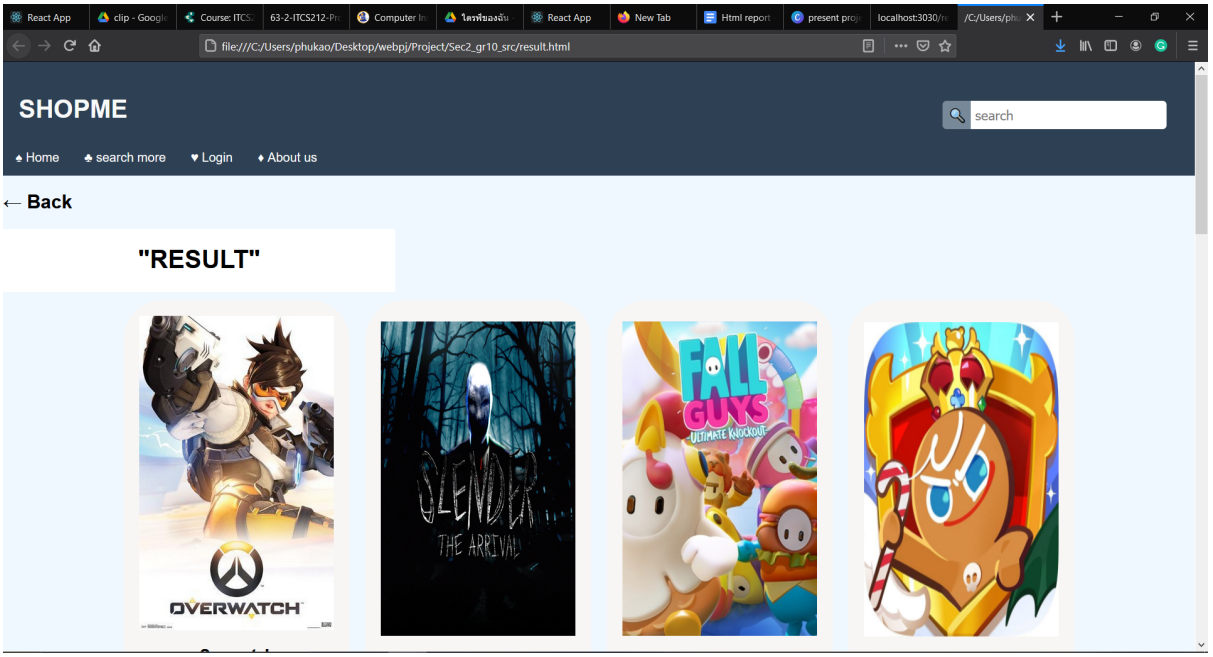
login page

This page is for users to login.



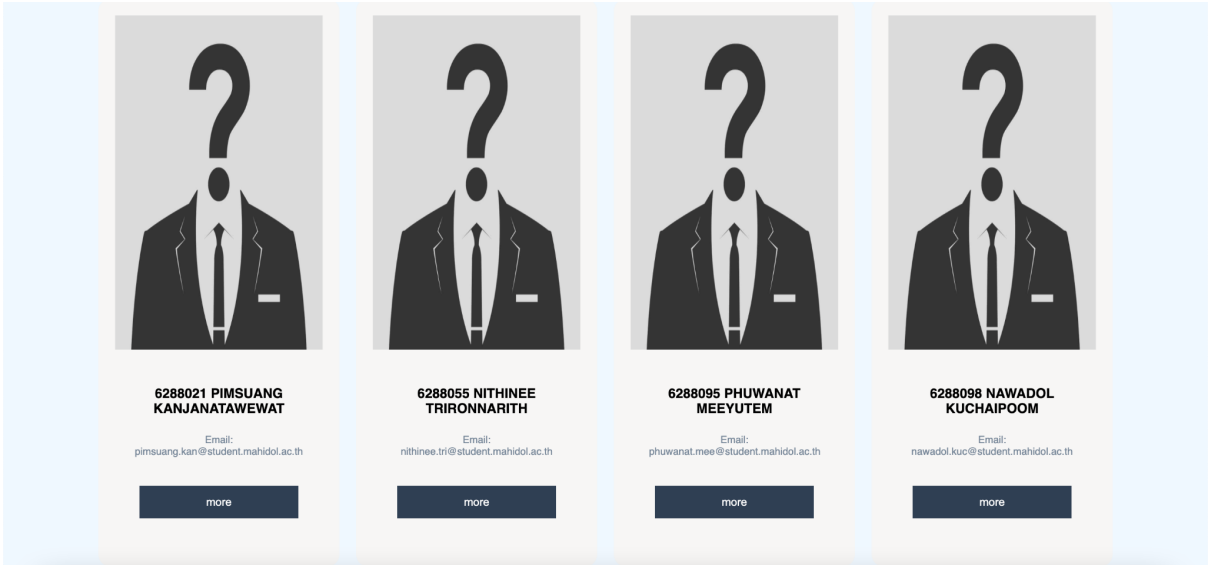
Result page

This page shows the results.



About us page

This page is about the creator's information.

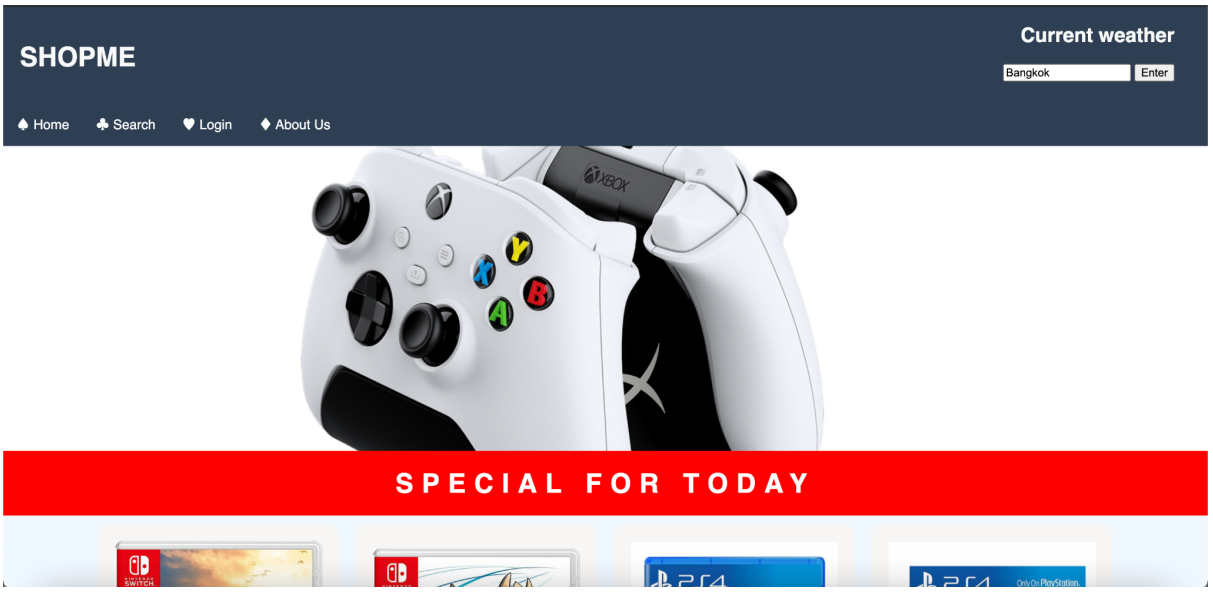


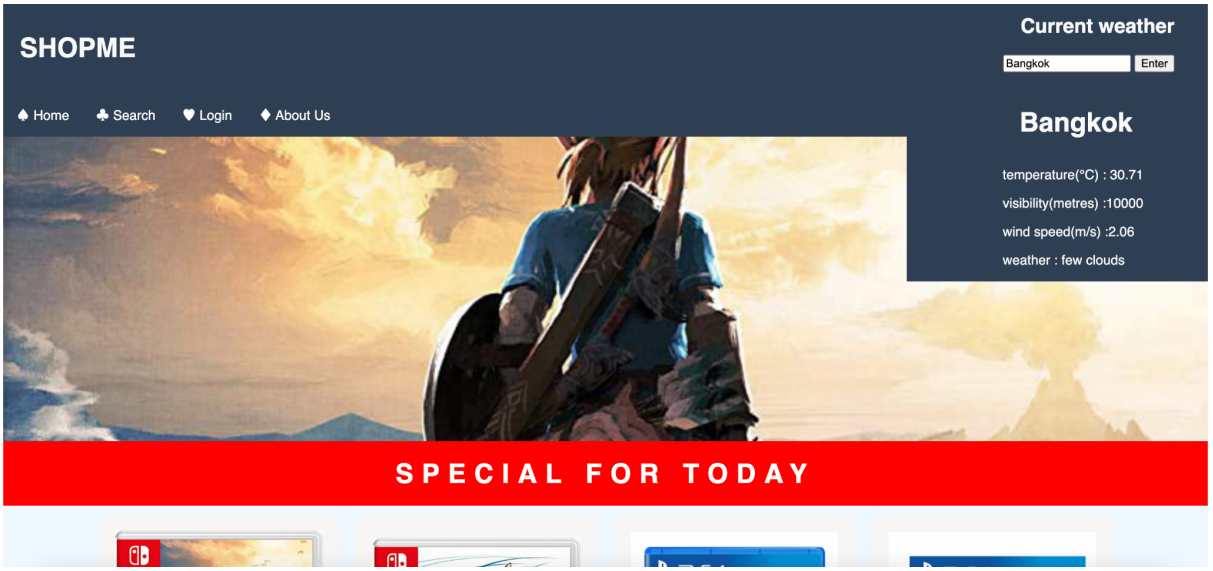
Phase2

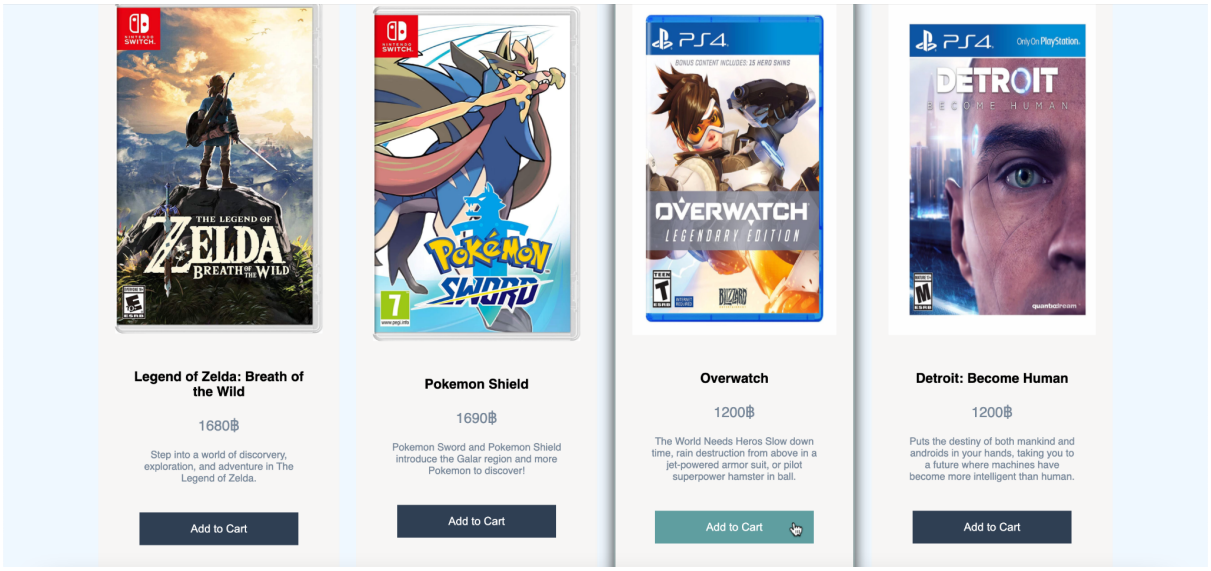
We use node.js to make a server side of the our website from phase1

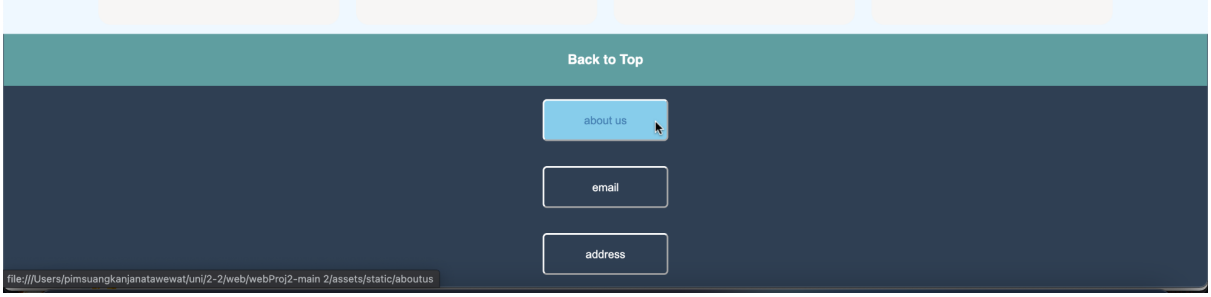
Homepage

This page shows our products that have a special offer.



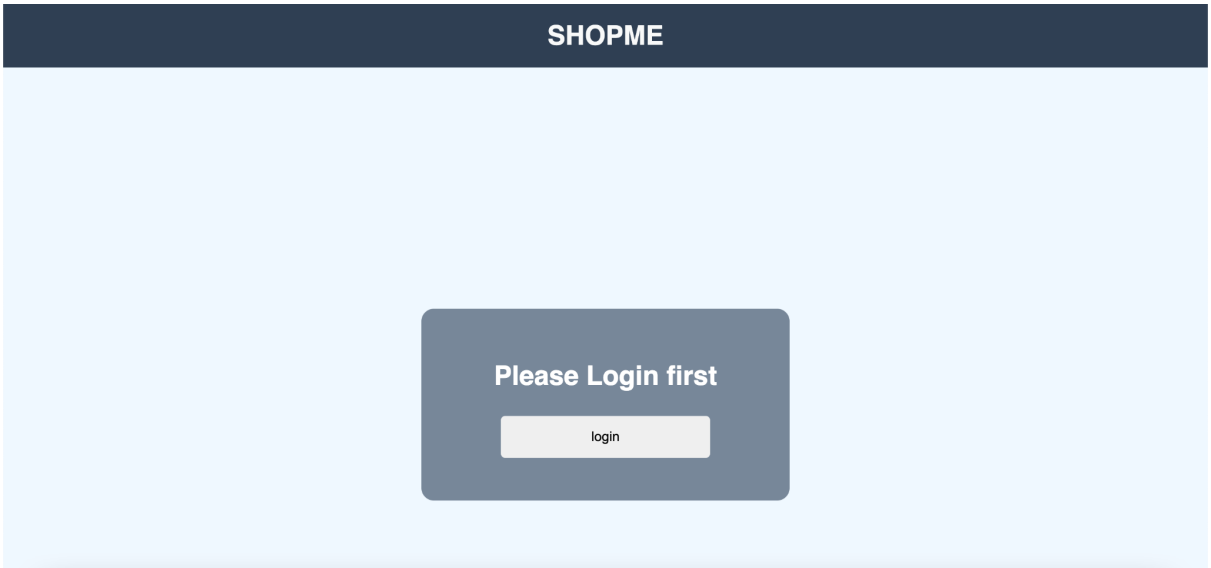


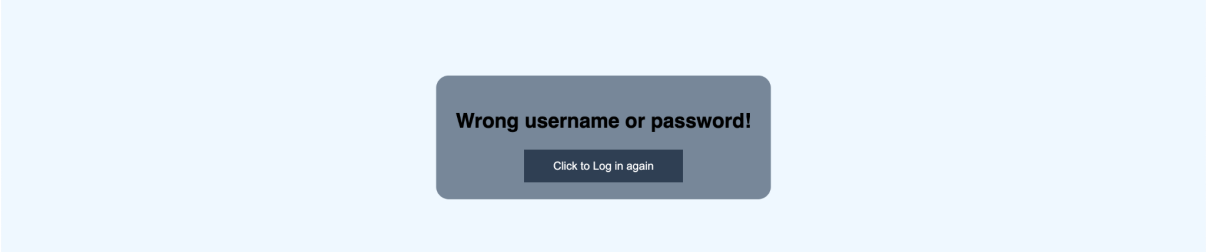




Login

This page lets the user login.



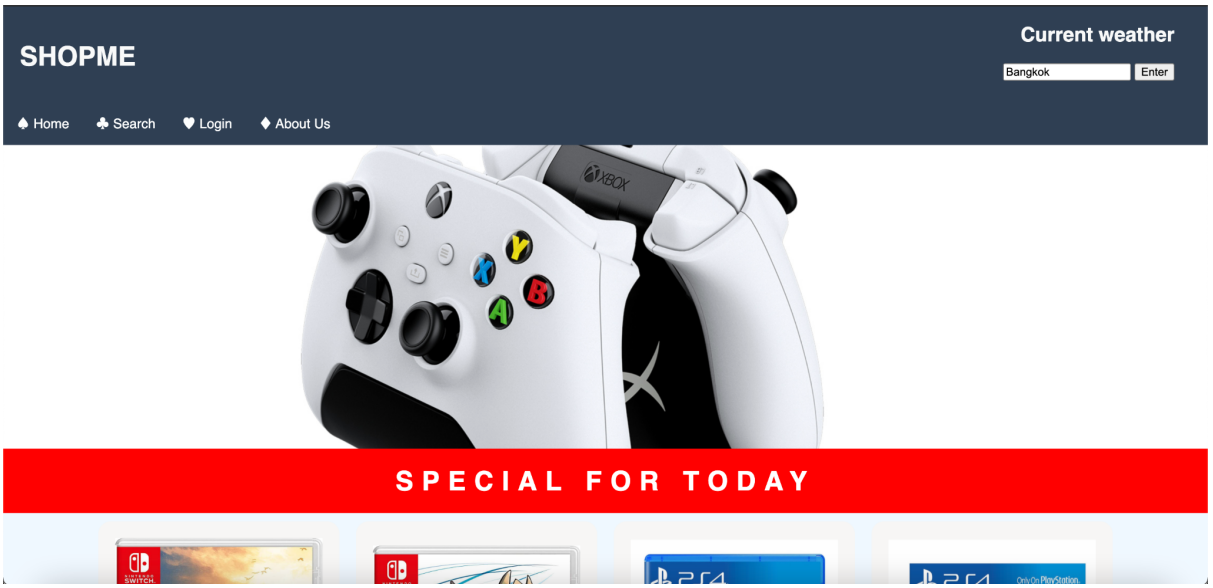


On our server side, It will check that the user role is admin or not. if the user is admin, it will redirect the user to the admin page. if the role is a user, it will redirect the user to homepage.

```
app.post('/auth',cors(), function (request, response) {
    var username = request.body.username;
    var password = request.body.password;

    //res.jsonp({success : true})
    //if("SELECT* FROM user WHERE urole ='user'")

    if (username && password) {
        connection.query('SELECT * FROM user WHERE username = ? AND pword = ?', [username, password], function (error, results, fields) {
            if (results.length > 0) {
                if (results[0].urole === 'admin') {
                    request.session.loggedin = true;
                    request.session.username = username;

                    //alert("welcome");

                    //
                    console.log(results);
                    console.log(`Welcome User: ${username}`);
                    authUser(request, response, results[0].urole)
                    response.redirect('/admin');

                } else {
                    request.session.loggedin = true;
                    request.session.username = username;

                    //alert("welcome");

                    //
                    console.log(results);
                    console.log(`Welcome User: ${username}`);
                    authUser(request, response, results[0].urole)
                    response.redirect('/homepage');

                }


            } else {
                console.log('Incorrect Username and/or Password!');
                response.redirect('/auth-alert');
            }
            response.end();

        });
```

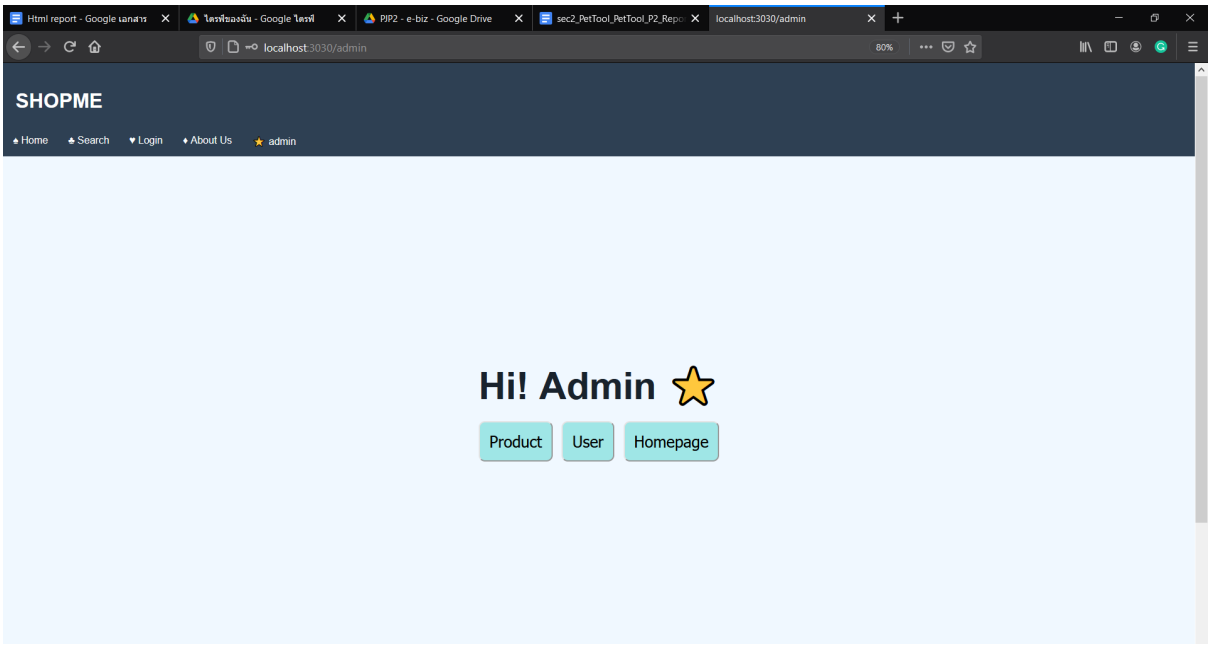
Normal User Pages with explanation for each page

Homepage for showing special products and have other buttons to go to the other page also the upper right is telling current weather as the user wants to see.

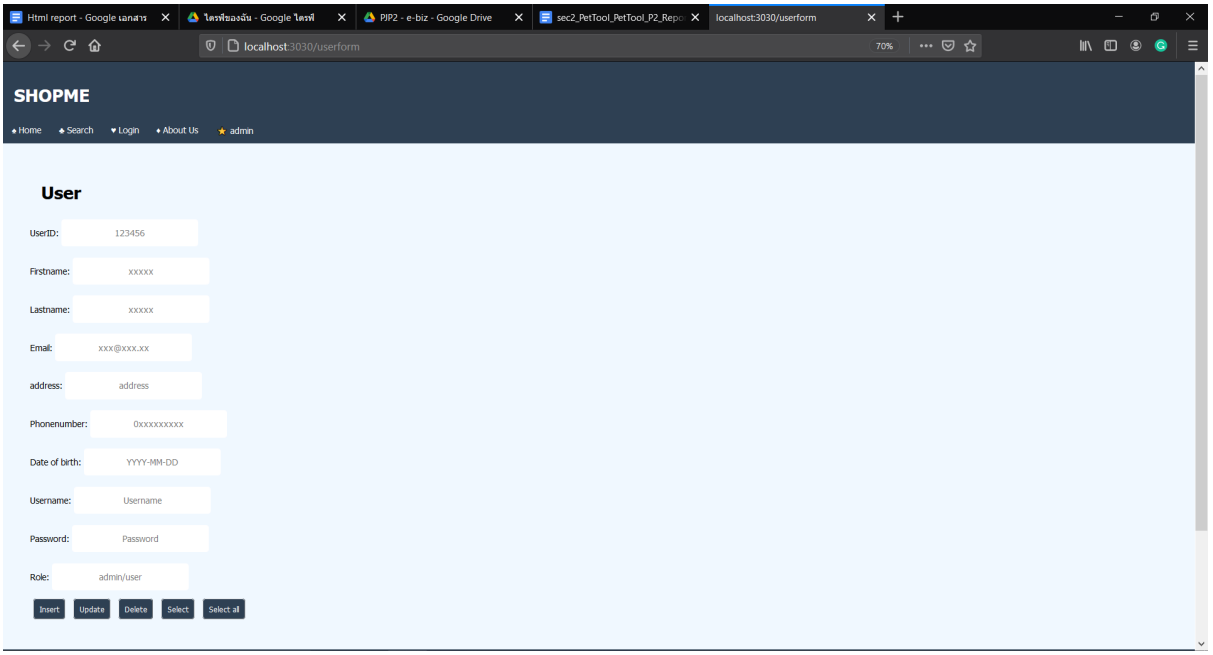


Administration Pages with explanation for each page

Admin home page when login admin role.



When clicking the User button it will go to the Userform page allowing the user to create, update, delete and etc. the user or admin then put and edit in database.



In each button, we will fetch the service from our server.

Add user

```
    });
    callUSERWS("http://localhost:3030/user", "insert", user_data).then(
      (data) => {
        console.log(data);
        if (data.data > 0) {
          alert(data.message);
          clearInput();
        }
      }
    );
  });
  updateBtnRef.addEventListener("click", () => {
```

Update user

```
  callUSERWS("http://localhost:3030/user", "update", user_data).then(
    (data) => {
      console.log(data);
      if (data.data > 0) {
        alert(data.message);
        clearInput();
      }
    }
  );
```

Delete user

```
callUSERWS("http://localhost:3030/user", "delete", user_data).then(
  (data) => {
    console.log(data);
    if (data.data > 0) {
      alert(data.message);
      clearInput();
    }
  }
);
```

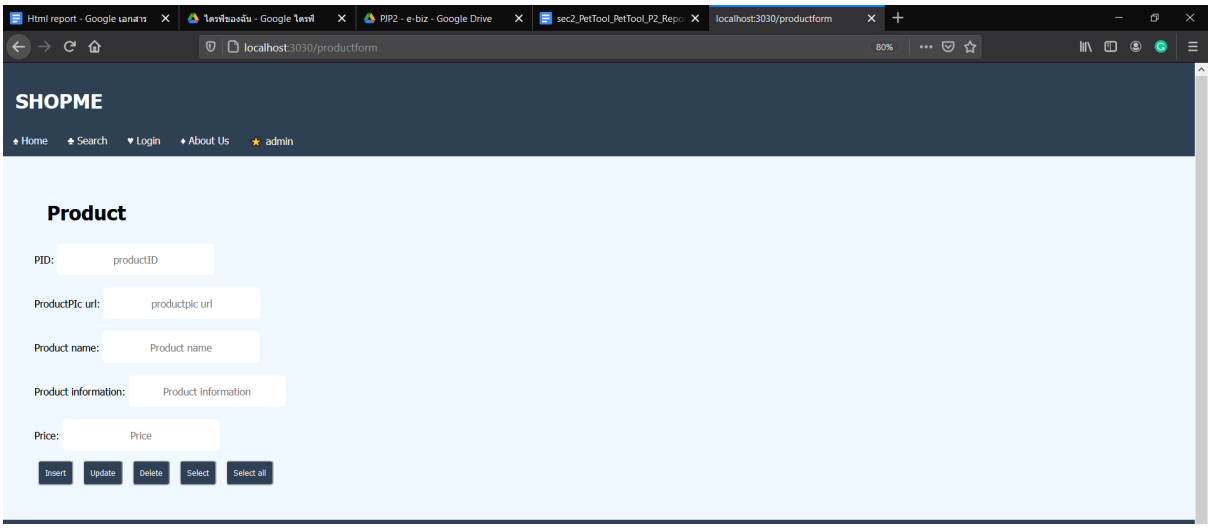
Select user by ID

```
})
selectBtnRef.addEventListener("click", () => {
  UID = U_IDTxtRef.value;
  callUSERWS("http://localhost:3030/user/" + UID, "select").then(
    (data) => {
      console.log(data);
      if (data) {
        alert(data.message);
        U_IDTxtRef.value = data.data.UID;
        U_FNAMETxtRef.value = data.data.fname;
        U_LNAMETxtRef.value = data.data.lname;
        U_EMAILTxtRef.value = data.data.Email,
          U_ADDRESSTxtRef.value = data.data.address,
          U_PHONETxtRef.value = data.data.pnumber,
          U_DOBTxtRef.value = data.data.DOB;
        U_USERTxtRef.value = data.data.username,
          U_PASSTxtRef.value = data.data.pword,
          U_ROLETxtRef.value = data.data.urole
      }
    }
  );
});
```

Select all users

```
selectallBtnRef.addEventListener("click", () => {
  callUSERWS("http://localhost:3030/users", "select").then(
    (data) => {
      console.log(data);
      if (data.data.length > 0) {
        alert(data.message);
        let output;
        output = "<h1> user List </h1>";
        output += "<table class='table'>";
        output += "<thead>";
        output += "<tr>";
        output +=
          "<th scope='col'>#</th><th scope='col'>First name</th><th scope='col'> Last name </th> <th scope='col'> Email </th> <t
        output += "</tr>";
        output += "</thread>";
        output += "<tbody>";
        data.data.forEach((element) => {
          output += "<tr>";
          output += "<td>" + element.UID + "</td>";
```

When click to Product button it will go to productform page allow to create, update, delete and etc. the products then put and edit in database.



Add product

```
    };
    callProductWS("http://localhost:3030/product", "insert", product_data).then(
      (data) => {
        console.log(data);
        if (data.data > 0) {
          alert(data.message);
          clearInput();
        }
      }
```

Update product

```
  };
  callProductWS("http://localhost:3030/product", "update", product_data).then(
    (data) => {
      console.log(data);
      if (data.data > 0) {
        alert(data.message);
        clearInput();
      }
    }
  );
})
```

Delete product

```
    };
    callProductWS("http://localhost:3030/product", "delete", product_data).then(
      (data) => {
        console.log(data);
        if (data.data > 0) {
          alert(data.message);
          clearInput();
        }
      }
    );
})
```

Select product by ID

```
selectBtnRef.addEventListener("click", () => {
    PID = P_IDTxtRef.value;
    callProductWS("http://localhost:3030/product/" + PID, "select").then(
      (data) => {
        console.log(data);
        if (data) {
          alert(data.message);
          P_IDTxtRef.value = data.data.PID;
          P_PICTxtref.value = data.data.pic;
          P_NAMETxtRef.value = data.data.Pname;
          P_INFOTxtRef.value = data.data.pinfo;
          P_PRICETxtRef.value = data.data.price;
        }
      }
    );
});
selectallBtnRef.addEventListener("click", () => {
```

Select all product

```
    });
    selectallBtnRef.addEventListener("click", () => {
      callProductWS("http://localhost:3030/products", "select").then(
        (data) => {
          console.log(data);
          if (data.data.length > 0) {
            alert(data.message);
            let output;

            output = "<h1> Product List </h1>";
```

Search

This page lets the user input the name of the product to search for the desired product.

So, if we input the word, it will send a request to show the product.

```
}
callallproduct("http://localhost:3030/allproducts/" + out).then(
  (data) => {
    console.log(data);
    if (data.data.length > 0) {

      //alert(data.message);
      console.log(data.data)
      let output = "";
      output += `<h1><mark style="background-color:white"> Searching for "${out}"</mark></h1>`
      output += "<div class='container'>"
      data.data.forEach((element) => {

        output += "<div class='card' style='width: 18rem;'>"
        output += "<img src='" + element.pic + "' class='card-img-top' alt='card'>"
        output += "<div class='card-body'>"
        output += "<h4><b>" + element.Pname + "</b></h4>";
        output += "<section class='price'>$" + element.price + "</section><br>";
        output += "<section class='des'>" + element.pinfo + "</section><br>";
        output += "<p><button>Add to cart</button>"
        output += "</div>"
        output += "</div>"
      });
      output += "</div>"
```

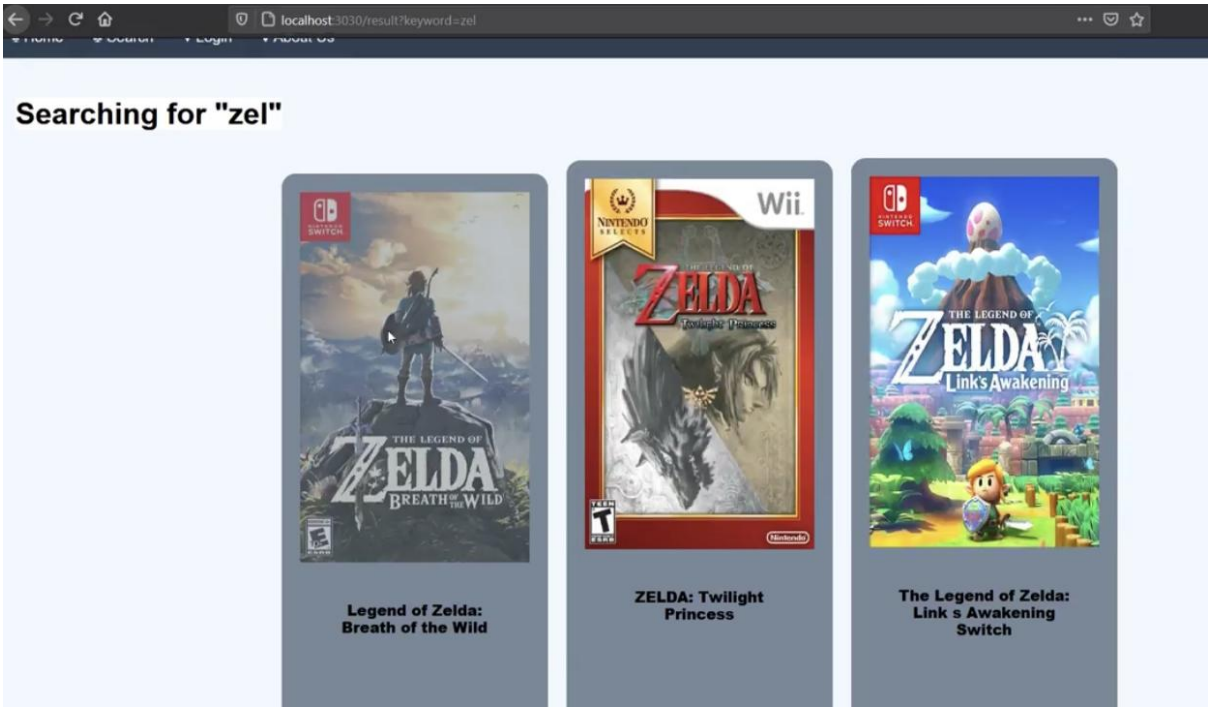
On the server side, It will check that the word that we input consists of the name of the product or not. Then it will show us the result of searching.

```
route.get('/allproducts/:keyword', cors(),function (req, res) {
    console.log("result page request");
    console.log("search for = " + req.params.keyword);
    let word = req.params.keyword;
    if (word != "") {
        connection.query(`SELECT * FROM product WHERE Pname like '%${word}%'`, function (error, results) {
            if (error) throw error;
            else {


                if (results.length > 0) {
                    return res.send({
                        error: false,
                        data: results,
                        message: 'Product list'
                    });
                } else {
                    connection.query(`SELECT * FROM product`, function (error, results) {
                        if (error) throw error;
                        else {
                            return res.send({
                                error: false,
                                data: results,
                                message: 'Product list'
                            });
                        }
                    })
                }
            }

    })
```

Example: “ZEL”

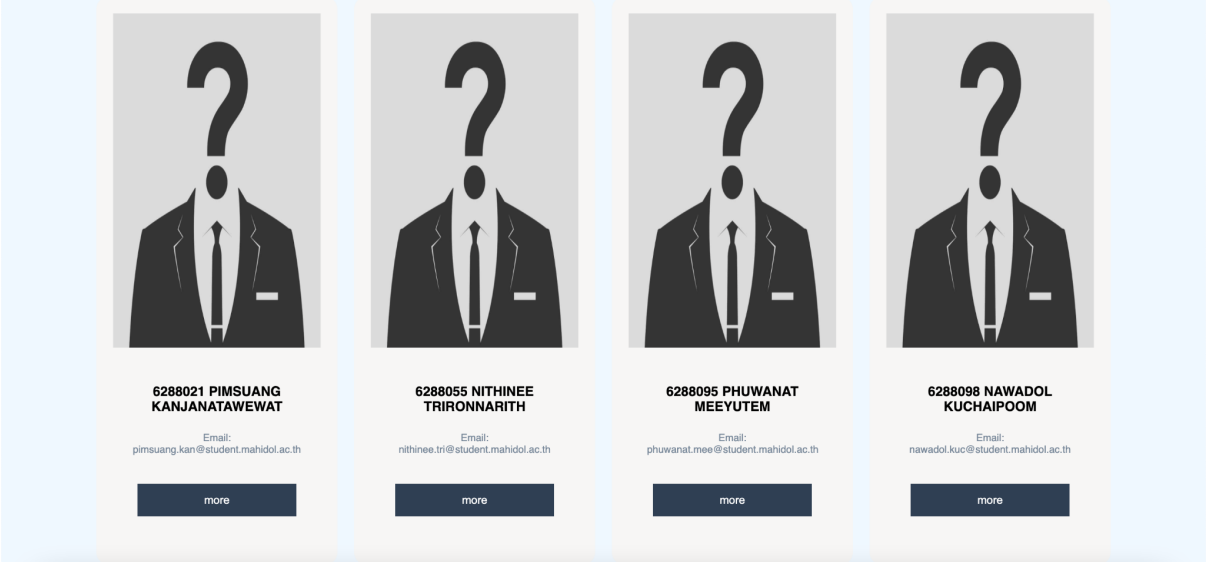
The products that contain the word ‘zel’ will be shown.

- Legend of Zelda: Breath of the Wild
- Zelda: Twilight Princess
- The Legend of Zelda: Link’s Awakening Switch



About us

This page shows the information of the developer.



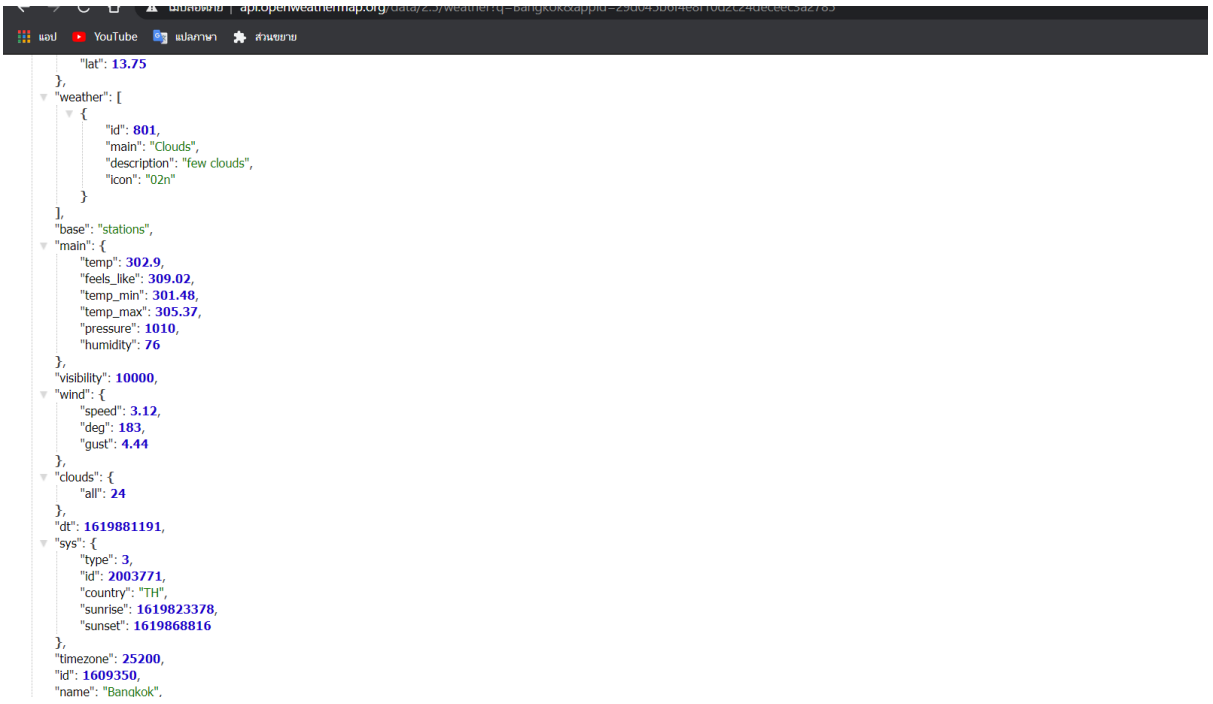
API

The api that we choose is the weather API. This API tells the current weather around the world. The reason why we choose this api is because when the user orders a product from our website we must deliver it to the customer so checking weather can know the obstacle with delivery.

Code to show in html in the homepage ask the user input country's name then show information about current weather from the input country.

```
297     </div>
298     <!-- Api weather</-->
299     <div>
300      <div class="right"><h2>Current weather</h2>
301     <div class="input">
302       <input type="text" placeholder="Type:Bangkok, New York or Japan" value="Bangkok" class="input_text">
303       <input type="submit" value="Enter" class="submit">
304       <div class="box2">
305         <h1 class="name" id="name"></h1>
306         <p class="temp"></p>
307         <p class="visibility"></p>
308         <p class="wind"></p>
309         <p class="desc"></p>
310     </div>
311   <script src="/static/weather.js"></script>
312     </div>
313   </div>
314
```

Raw data from the api we use to show on the homepage.



Javascript code for connect with api, we create various variables as we want from the api have

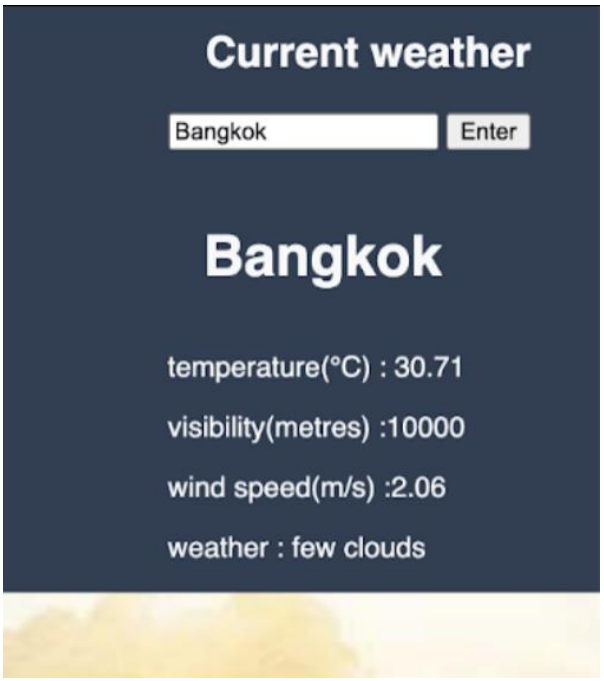
We take the country's name from the homepage's input, insert it into the api link and take the current data and show it to the homepage.

lines (23 sloc) 1.08 KB

```
var input = document.querySelector('.input_text');
var main = document.querySelector('#name');
var temp = document.querySelector('.temp');
var desc = document.querySelector('.desc');
var wind = document.querySelector('.wind');
var visibility = document.querySelector('.visibility');
var button = document.querySelector('.submit');


button.addEventListener('click', function () {
  fetch('https://api.openweathermap.org/data/2.5/weather?q=' + input.value + '&appid=29d045b6f4e8f10d2c24deceec3a2785&units=metric')
    .then(response => response.json())
    .then(data => {
      var tempValue = data['main']['temp'];
      var nameValue = data['name'];
      var descValue = data['weather'][0]['description'];
      var visibilityValue = data['visibility'];
      var windValue = data['wind']['speed'];

      main.innerHTML = nameValue;
      desc.innerHTML = "weather : " + descValue;
      temp.innerHTML = "temperature(°C) : " + tempValue;
      visibility.innerHTML = "visibility(metres) :" + visibilityValue;
      wind.innerHTML = "wind speed(m/s) :" + windValue;
    })
})
```

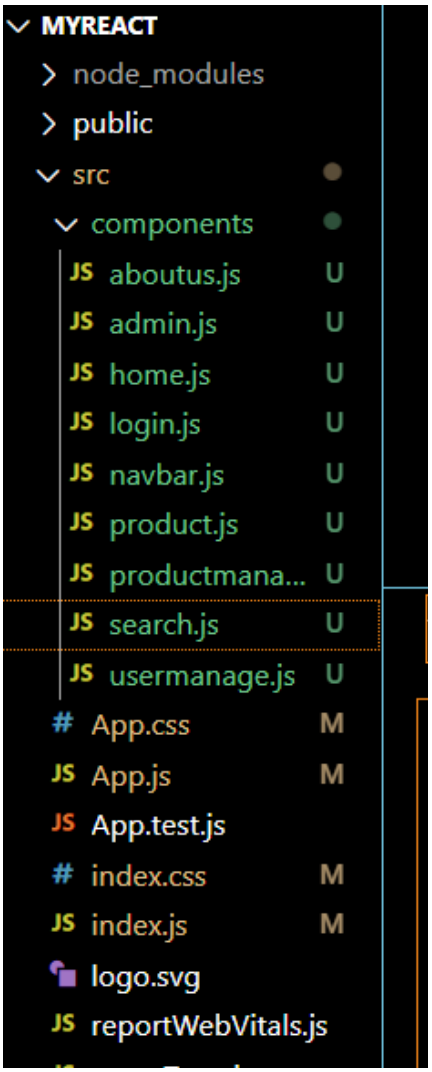
All web services testing results

1.API current weather



Phase 3

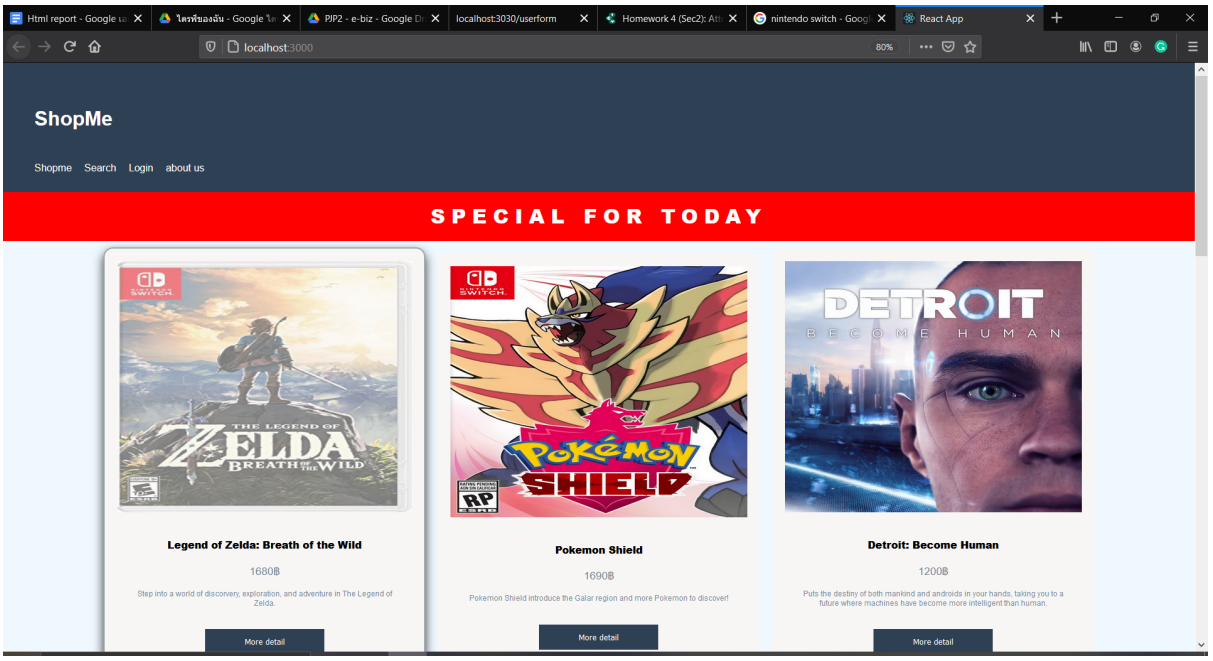
We use React to create the website and we also use projects from Phase1 and Phase 2 to implement our website.



As you can see from our files. We have navbar.js. We use navbar.js to link between each page.

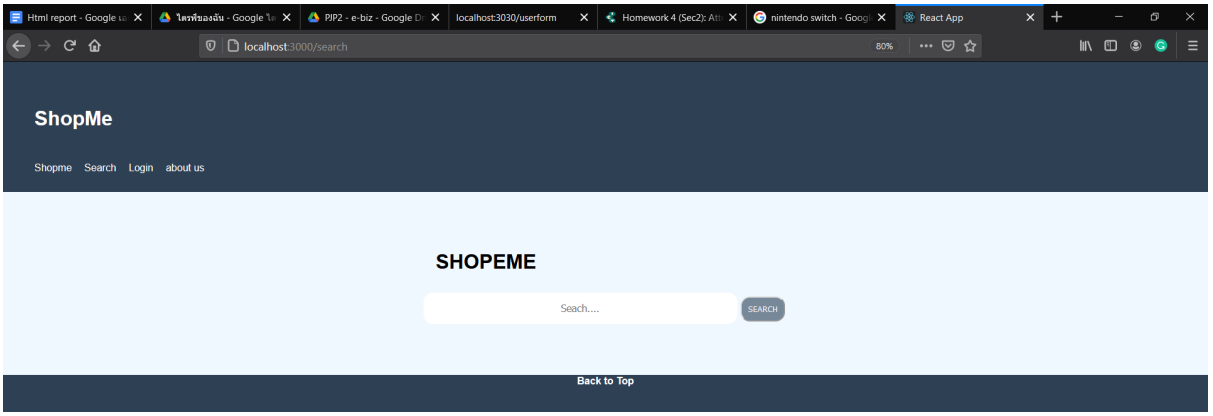
Homepage

This page shows the special offer product but we implement it from react.



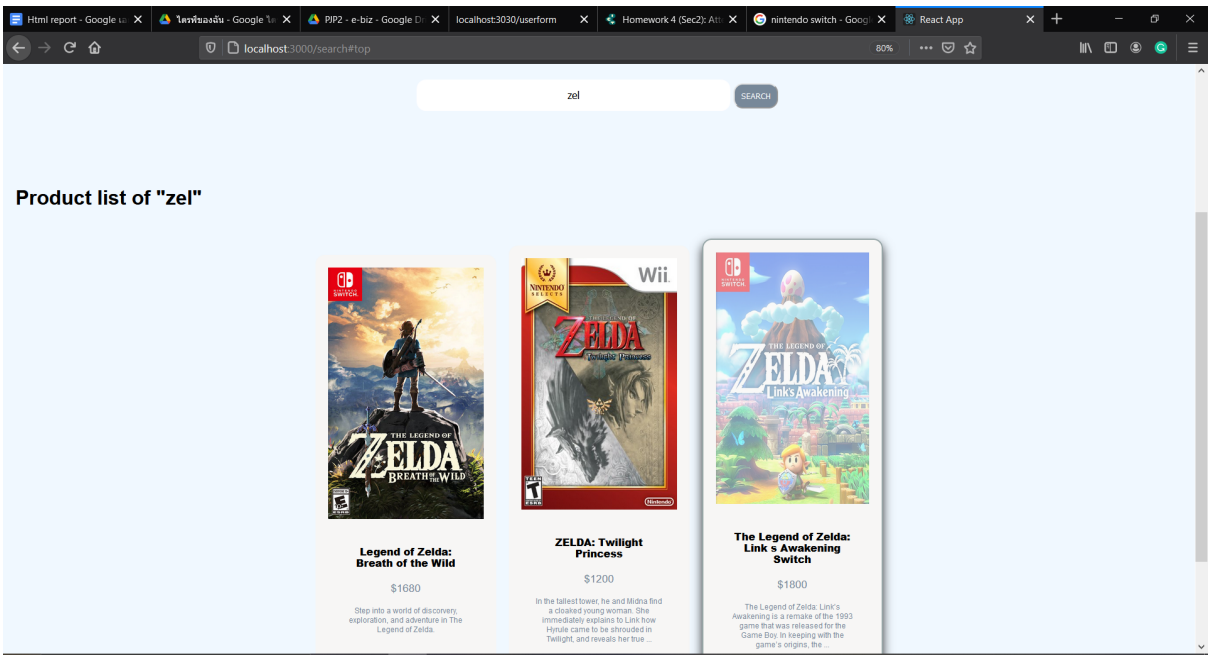
Search page

This page is for user to search the desired product.



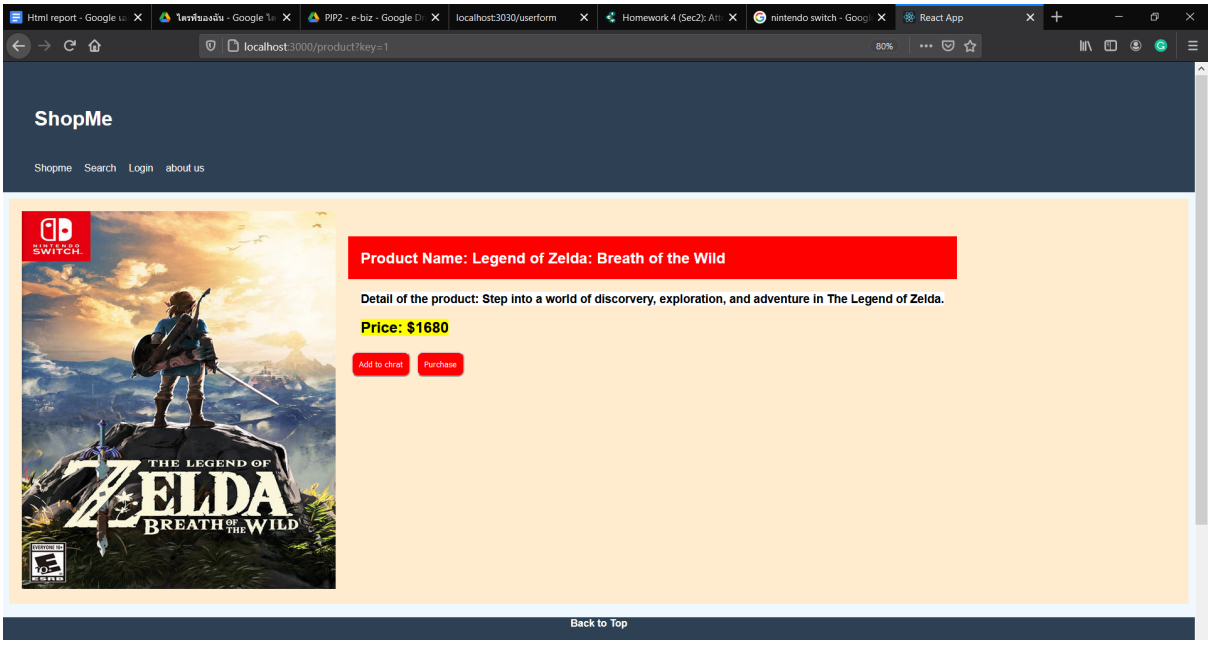
We do similar things with phase2 but this time we fetch the web service from our project2 to show the data from our DB instead of calling the DB directly.

```
let rooturl = "http://localhost:3030/allproducts/"+this.state.menu;
fetch(rooturl,
    {
        method: "GET",
        headers: {
            "content-type": "application/json",
            "accept": "application/json"
        },
    })
    .then((response) => response.json())
```



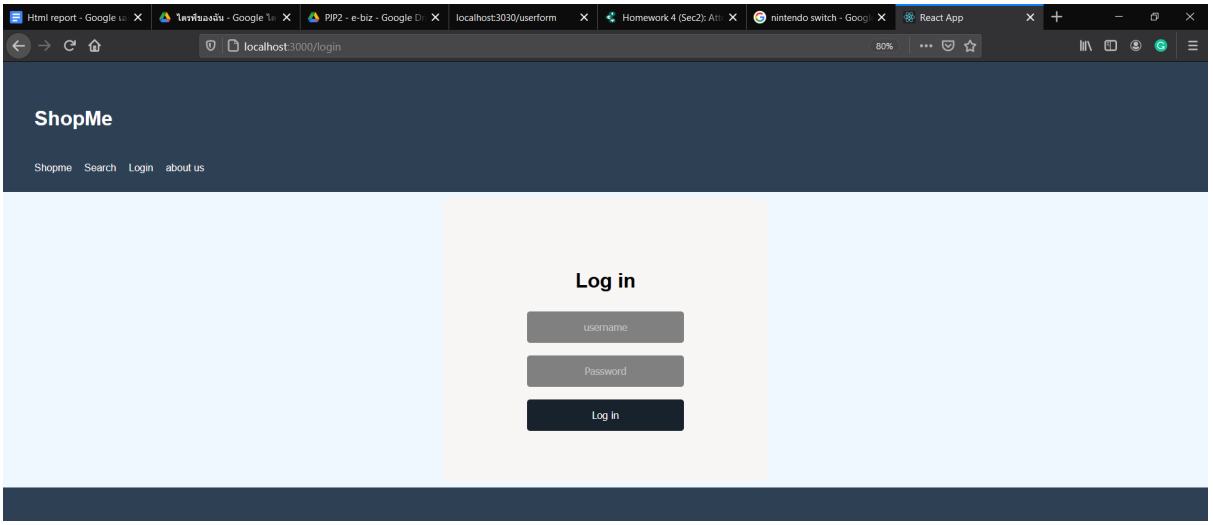
If we click More detail, it will bring us to the detail of the product page.
In the product page we also fetch the data from our web service from project2 but in this case we add the specific “key” that is product ID to classify the product page.

```
fetch("http://localhost:3030/allproductsdetail/" + key, {
    method: "GET",
    headers: {
        "content-type": "application/json",
        accept: "application/json",
    },
})
    .then((response) => response.json())
```



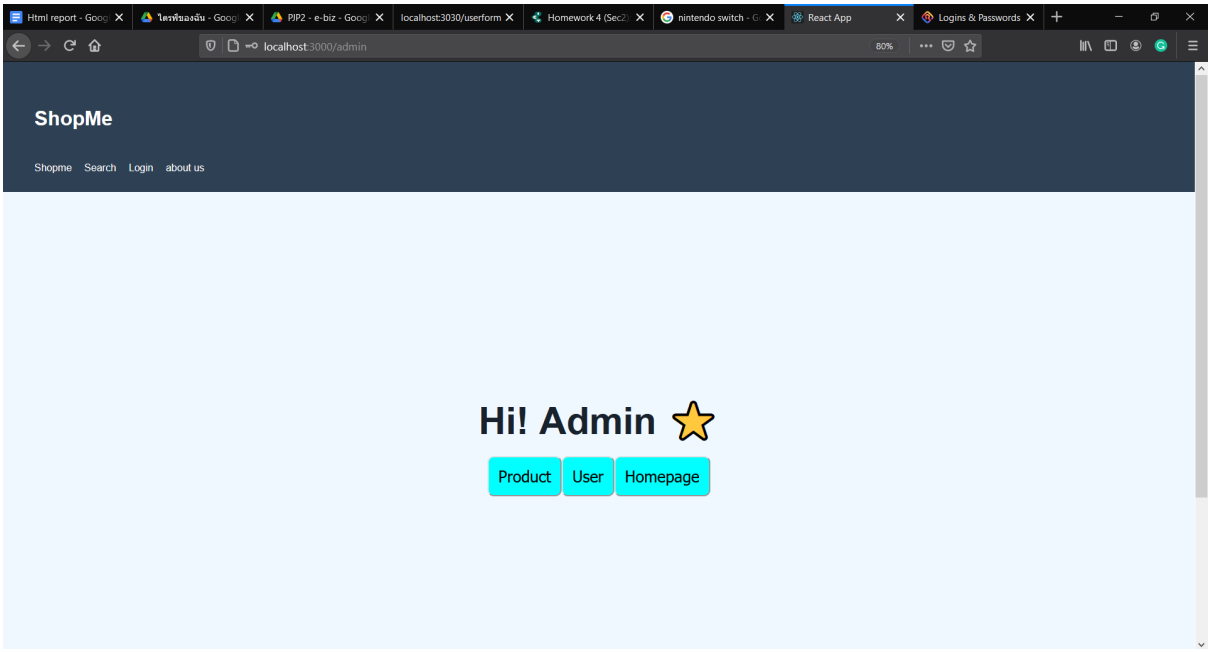
Login Page

This page asks users to login to their account.



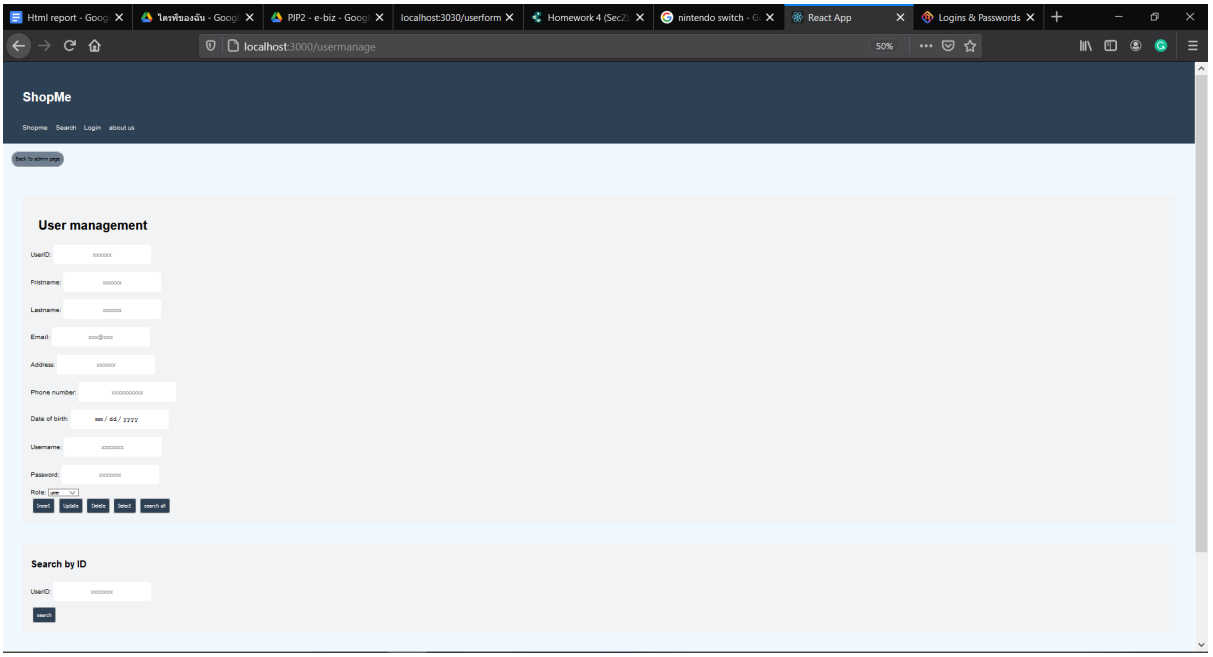
In the log in page, we also fetch the service from our service in project 2 to check whether the user and password is correct or not. and also check the role of the user. If the role of the user is admin, it will bring the user to the admin page.

```
let rooturl = "http://localhost:3030/logincheck/"+this.state.username+'&'+this.state.password;
fetch(rooturl,
    {
      method: "GET",
     /*body: JSON.stringify({

        username: this.state.username,
        password:  this.state.password
    }),*/
      headers: {
        "content-type": "application/json",
        "accept": "application/json"
      },
    })
    .then((response) => response.json())
    .then((data) => {
        console.log(data.data[0])
        if (data.data[0].urole==='admin')
        {
          window.location.replace('http://localhost:3000/admin');
        }
        else
        {
          window.location.replace('http://localhost:3000/');
        }
```



Admin page consists of 2 features, product management and User management.

User management page



We fetch the web service from our last project. This page allows the admin to manage all users in the database and also can search users by their ID.

Add user

```
let rooturl = "http://localhost:3030/user";
fetch(rooturl,
    {
        method: "POST",
        headers: {
            "content-type": "application/json",
            "accept": "application/json"
        },
        body: JSON.stringify(user_data)
    })
    .then((response) => response.json())
    .then((data) => {
        alert("Insert successfully")

    })
```

Update user

```
let rooturl = "http://localhost:3030/user";
fetch(rooturl,
    {
        method: "PUT",
        headers: {
            "content-type": "application/json",
            "accept": "application/json"
        },
        body: JSON.stringify(user_data)
    })
    .then((response) => response.json())
    .then((data) => {
        alert("Update successfully")

    })
```

Delete user

```
let rooturl = "http://localhost:3030/user";
fetch(rooturl,
    {
        method: "DELETE",
        headers: {
            "content-type": "application/json",
            "accept": "application/json"
        },
        body: JSON.stringify(user_data)
    })
    .then((response) => response.json())
    .then((data) => {
        alert("user deleted");
    })
```

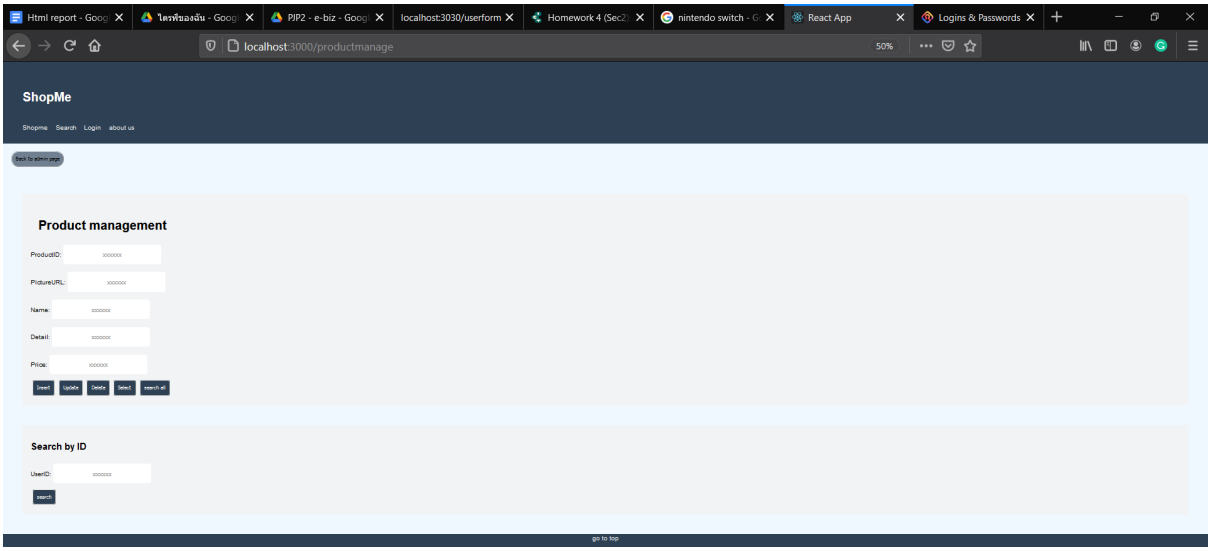
Search user by ID

```
let rooturl = "http://localhost:3030/user/" + this.state.UID;
fetch(rooturl,
    {
        method: "GET",
        headers: {
            "content-type": "application/json",
            "accept": "application/json"
        },
    })
    .then((response) => response.json())
    .then((data) => {
```

Select all users

```
let rooturl = "http://localhost:3030/users";
fetch(rooturl,
    {
        method: "GET",
        headers: {
            "content-type": "application/json",
            "accept": "application/json"
        },
    })
    .then((response) => response.json())
    .then((data) => {
```

Product management Page



In the product management page we also do the same things as the user management page.

Add product

```
let rooturl = "http://localhost:3030/product";
fetch(rooturl,
    {
        method: "POST",
        headers: {
            "content-type": "application/json",
            "accept": "application/json"
        },
        body: JSON.stringify( product_data )
    })
    .then((response) => response.json())
    .then((data) => {
        alert("Insert successfully")
```

Update product

```
let rooturl = "http://localhost:3030/product";
fetch(rooturl,
    {
        method: "PUT",
        headers: {
            "content-type": "application/json",
            "accept": "application/json"
        },
        body: JSON.stringify( product_data )
    })
    .then((response) => response.json())
    .then((data) => {
        alert("Update successfully")
```

Delete products

```
let rooturl = "http://localhost:3030/product";
fetch(rooturl,
    {
        method: "DELETE",
        headers: {
            "content-type": "application/json",
            "accept": "application/json"
        },
        body: JSON.stringify( product_data )
    })
    .then((response) => response.json())
    .then((data) => {
```

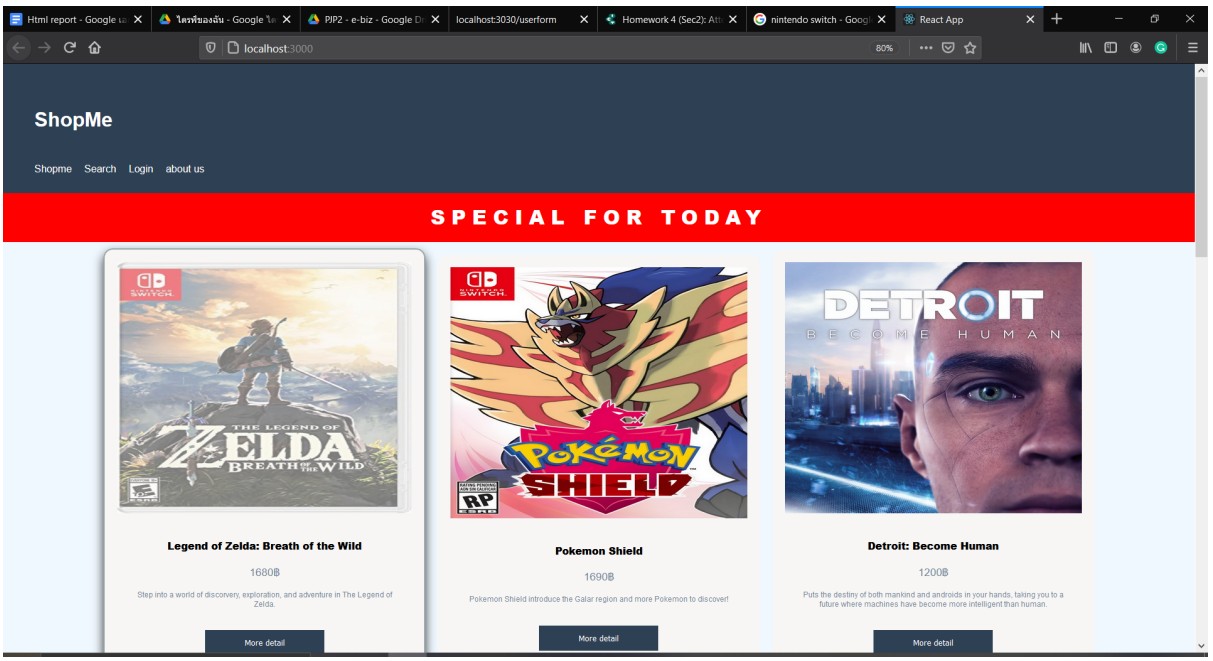
Search product by ID

```
let rooturl = "http://localhost:3030/product/"+this.state.PID;
fetch(rooturl,
    {
        method: "GET",
        headers: {
            "content-type": "application/json",
            "accept": "application/json"
        },
    })
    .then((response) => response.json())
    .then((data) => {
        if(data.data!=null){
```

Select all products

```
let rooturl = "http://localhost:3030/products";
fetch(rooturl,
    {
        method: "GET",
        headers: {
            "content-type": "application/json",
            "accept": "application/json"
        },
    })
    .then((response) => response.json())
    .then((data) => {
        console.log(data.data);
```

In case that user role is a normal user, it will bring the user to the home page.



About us page

This page shows the information of the developer.