**4.Discussion of your solution:**

**db.js:**

At first I put a code which emphasize a transaction for creating database in WebSql (not mysql database). After that I put addProduct: function(), this emphasize a transaction for creating table product. After that I put tx.executeSql("drop table if exists products"); it’s function is to delete table product if it does exists. In addition, I also add function(error), its function inside is to show error message while running the transaction. Not only that, I also make the program to tell user that the product is created.

On the other side, I also put function(ctg,callback), it is a function to collect data from product table in specified category. After that, I put function(tx,results), it is used if the query succeeds and the result return to the caller function with callback. After that I use callback (items), it is to return the result to the caller function. I also use console.log("error: "+error.message);, it is to be used if there is any error while running the query. I also use console.log("error: "+error.message);, it is to be used if any error happen while running the transaction. I also use console.log("Select product Completed"); it is used if the transaction succeeds.

In addition, I use crtTrans: function(), it is used as a function to create transaction table. After that I use addTrans: function(nop,nme,num,prc), it is a function to insert data into transaction table. Not only that, I also use slcTrans: function(callback), it is a function for collecting data from transaction table. After that, I use callback (items) it is used to return the result to the caller function. After that, I use crtMsg: function(), it is a function to create message table. In addition, I also use addMsg: function(frs,lst,ctr,msg), it is a function to insert message into the message table. Not only that, I also use verify: function(usr,pwd), it is a function to verify username and password when someone logging in. Inside it, it contain tx.executeSql("select \* from users where user = ? and pass = ?",[usr,pwd], it is to try to search data in user table according to username and password inputed by user. I also add if (len>0), it is for if length result is more than zero it means that the username existed. Inside it I put window.location = "admain.html"; it is to jump to admin page. I also put window.location= "main.html"; it is to display user page. I also put a coding about else if the above condition is not satisfied which means if length result equals zero, it means the username for password is invalid. After that I put console.log("error: "+error.message);, it is for if the query is error. Last, I put register: function(usr,pwd) it is a function to insert new user table.

**Index.js:**

At first, I put $(document).on("ready",function(), it is a function that is run when index.js is called. Below it I call some of the function in db.js. After that I put function verLog(), this function is to verify the value of username and password inputs in html page and send them to verify function in db.js. After that I also use function verReg(), it is a function to collect username and password for new user and send them to register function in db.js After that, I put function svMsg() it is a function to collect data for new message and send them to save message function in db.js. In addition, I also use function selPrd(callback) it is a function to call function in db.js for collecting data product and process the return items to display in html page. Below it I use prdHandler.slcProduct(ctg,function(items), it is to call select product function by sending category and receive return value items after that I prepare the return value items to be displayed in html page. Below it I use $("#prdcnt").html(tmp); it is to send prepared items to display in html page.

On the other side, I also use function buyPrd(callback) it is a function to call function in db.js for buying process. Below it I put prdHandler.slcProduct(ctg,function(items) it is to call select product function by sending category and receive return value items. After that I prepare the return value items to be displayed in html page. Not only that, I also put temp+= "<a class = 'ui-btn ui-icon-plus ui-btn-icon-left' id='btnBuy' onclick='trxBuy();'>Buy this item!!!</a> it is to add buy button for customer who wants to buy the product and calling trxBuy() function. After that, I put $("#prdcnt").html(tmp); it is to send prepared items to display in html page. In addition, I also use function trxBuy(), it is called by buyPrd() function and then collecting data from html after that send them to addTrans function in db.js to be inserted in transaction table. Below it I put function selTrx(callback) it is a function to call function in db.js for collecting data transaction and process the return transaction items to display in html page. Below it again I put var total = 0; it is a variable to store total shopping amount. After that I display total and payment methods.

**about.html:**

At first I Include meta tag to ensure proper rendering and touch zooming then I Include jQuery Mobile stylesheets, after that I Include the jQuery library. Not only that I also Include the jQuery Mobile library, below it I add a custom library which I made. In addition, I use <div data-role = "page" id = "aboutpage">, it is a Header Section, Div's position is fixed in the top of the page. Below it I put User Menu after log in. Then, I put the group member data (Which is the ceo of the store) in the member section. Lastly, I put the Footer Section, Div's position is fixed in the bottom of the page.

**admain.html:**

This is mainly same with About.html, however it is to show the main page to show the logo of the company and the name of the company. Header and footer is the same as About.html as well as the nav.

**contact.html:**

Header,footer,and nav is the same with about.html. However, inside it I put a form so that the user is able to send message to the company. After that I put <textarea id = "message" placeholder = "Enter Your Message Here" data-clear-btn = "true"> </textarea>. This is a Button to submit data from this form which will later be saved in the database with a javascript function.

**index.html:**

Header,footer,and nav is the same with about.html. It is mainly the same with admain.html, however this page is for the user when they logged in already.

**login.html:**

Header,footer,and nav is the same with about.html. However, I inside it I put a login Section, when user is already registered in company's database. I also use <input type="password" placeholder="Enter your password" id="password" data-clear-btn = "true" /><br> it is a Button to submit username and password which later will be verified by javascript function, if valid it will go to the user, if not it will show alert from javascript function. Below it, I put the register function, it is for user who want to register themselves into company's database. Below it I use <input type="password" placeholder="Enter your password" id="passreg" data-clear-btn = "true" />, it is a Button to submit username and password which later will be registered into company's database by javascript function.

**main.html:**

Header,footer,and nav is the same with about.html. However, this is the First page displayed when the user login is valid.

**payment.html:**

Header,footer,and nav is the same with about.html. However I use a Javascript function which is loaded everytime the page refreshed to display list product and payment method in the #prdcnt section. After that, I put Payment page which contain the list of product which user want to buy and the payment method. Below it I give a section the code which will display the list of product and payment method triggered by javascript function. After that, I put Button to activate javascript function and to submit payment method and finish the transaction.

**product.html:**

It is a Product page which is used by user to explore the company's products and buy them if they want. Header,footer,and nav is the same with about.html. However, I give a Drop down list which is used by user to choose a product type that they want. Every changes occurs will trigger javascript function to display the list of product according to the type that user chose, added with the input for quantity and buy's button. Below it I put a section which will display the list of product triggered by javascript function. Below it I put a button which is used for changing to payment page after the user is finish choosing product and ready to do payment.

**producti.html:**

This is a page where it is accessible by everyone (Non Registered User), to explore company's product but they cannot buy it from here. Header,footer,and nav is the same with about.html. However, I give a User menu before login. After that I give a drop down list which is used by user to choose a product type that they want. And every changes which occur will trigger javascript function to display the list of product according to the type that user chose. Lastly, I put a section which the code will display the list of product triggered by javascript function.

**success.html:**

This is the end page after all transaction is completed and the user will be automatically logged out from the company system. Header,footer,and nav is the same with about.html.

Currently all of the code above meets my expectation

**Strength:**

-The code still performing as what expected

- The code can run well with a cordova front end packaged for Android and iOS.

**Weakness:**

-Not using MySQL database, so not connected with other group member project since I have tried from many resources but still can’t find the way to use mysql database.

-If there is changes of data within the MySQL database I need to change it manually in my WebSQL database.

- The transaction succeeds only for first product in each type, it happened maybe because I use the same ID for every product name. The solution that I think about, but not executed is to display each of the product one by one by adding product number selector.

**Improvement:**

-Try more to be able to use MySQL database.

- Do more research to find a better code to give more efficiency towards the program.

**5. Self-diagnosis and evaluation:**

Requirements which are fully met:

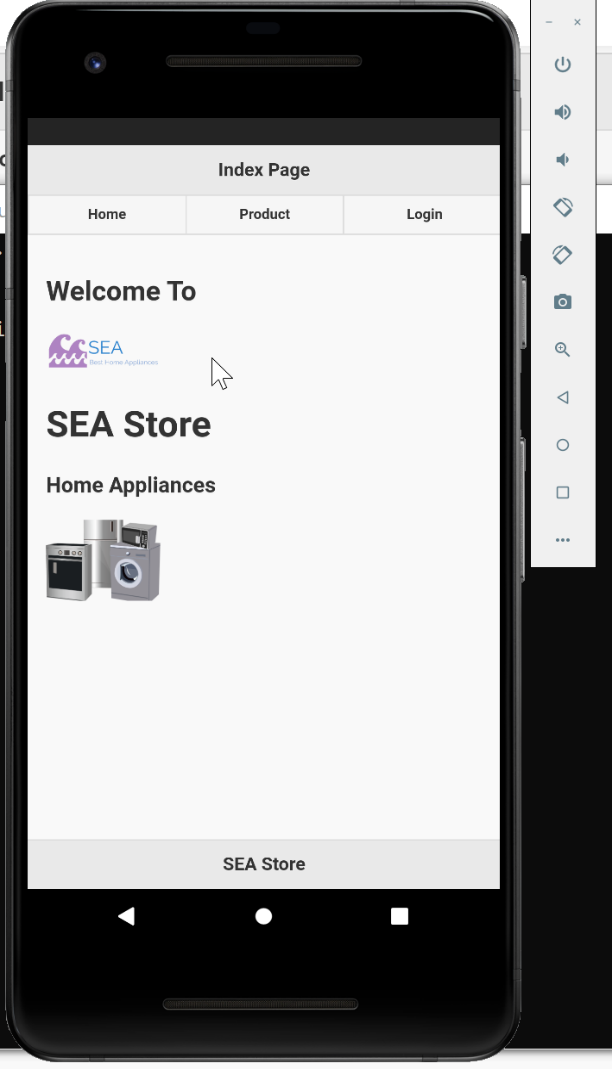
* The program can give access towards anyone without authentication process if they want to see all of the basic functionalities that the program has. In my case, unregistered user can see all of the product that my company offer.
* There is a registered customer which the user can register their username and password by themselves which later on can be use to log in into the program. There is also 3 customer and staff accounts which the assignment asks. This account can also be directly typed and login into the program to access the program.
* Since this is cordova by then the code is not hosted in ceto.

Requirements not fully met:

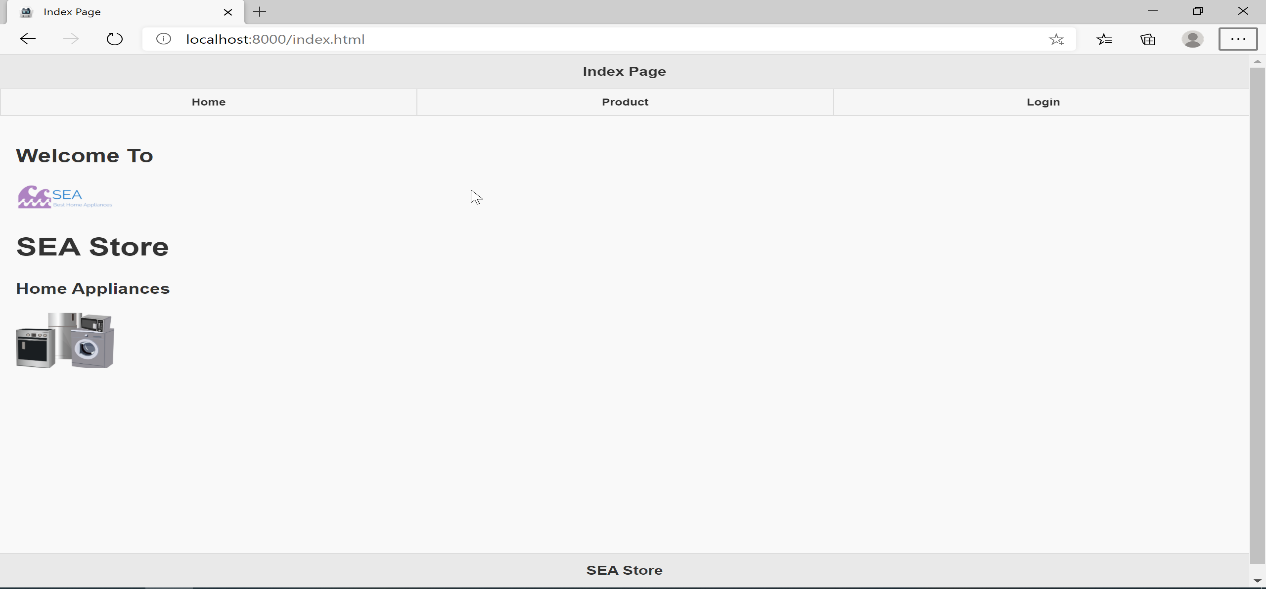
* Not using MySQL database, so not connected with other group member project since I have tried from many resources but still can’t find the way to use mysql database. However, I fix the problem by using WebSQL.
* The transaction succeeds only for first product in each type, it happened maybe because I use the same ID for every product name. The solution that I think about, but not executed is to display each of the product one by one by adding product number selector.

**8. Test documentation for my cordova application:**

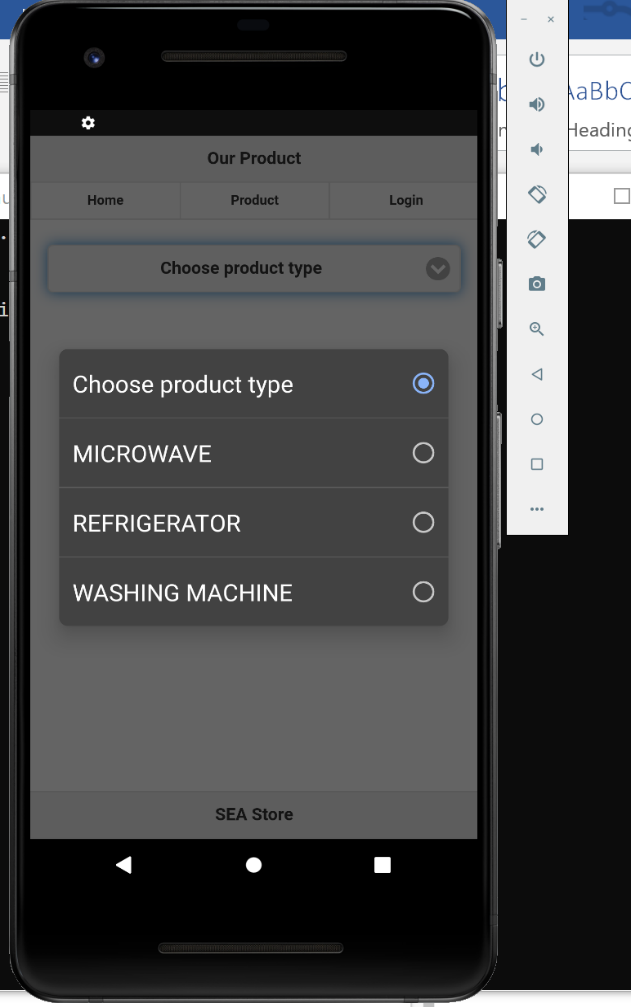
**Index page for everyone including registered and unregistered user (Android perspective):**

****

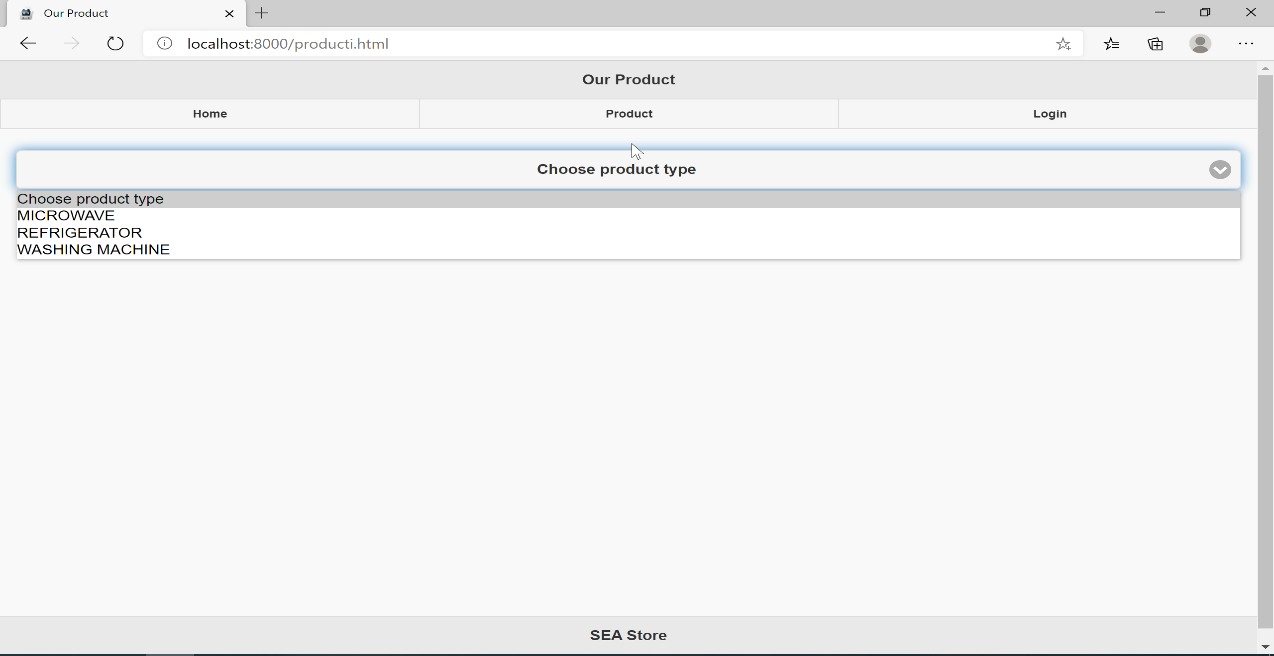
**Index page for everyone including registered and unregistered user (Browser perspective):**



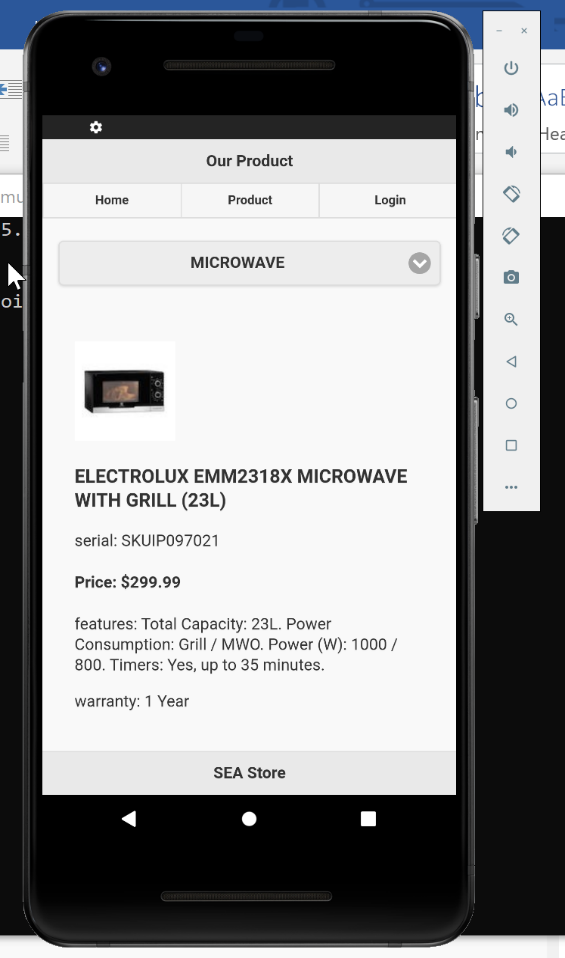
**User can choose which product that they want to see (For both registered and unregistered user) (Android perspective):**



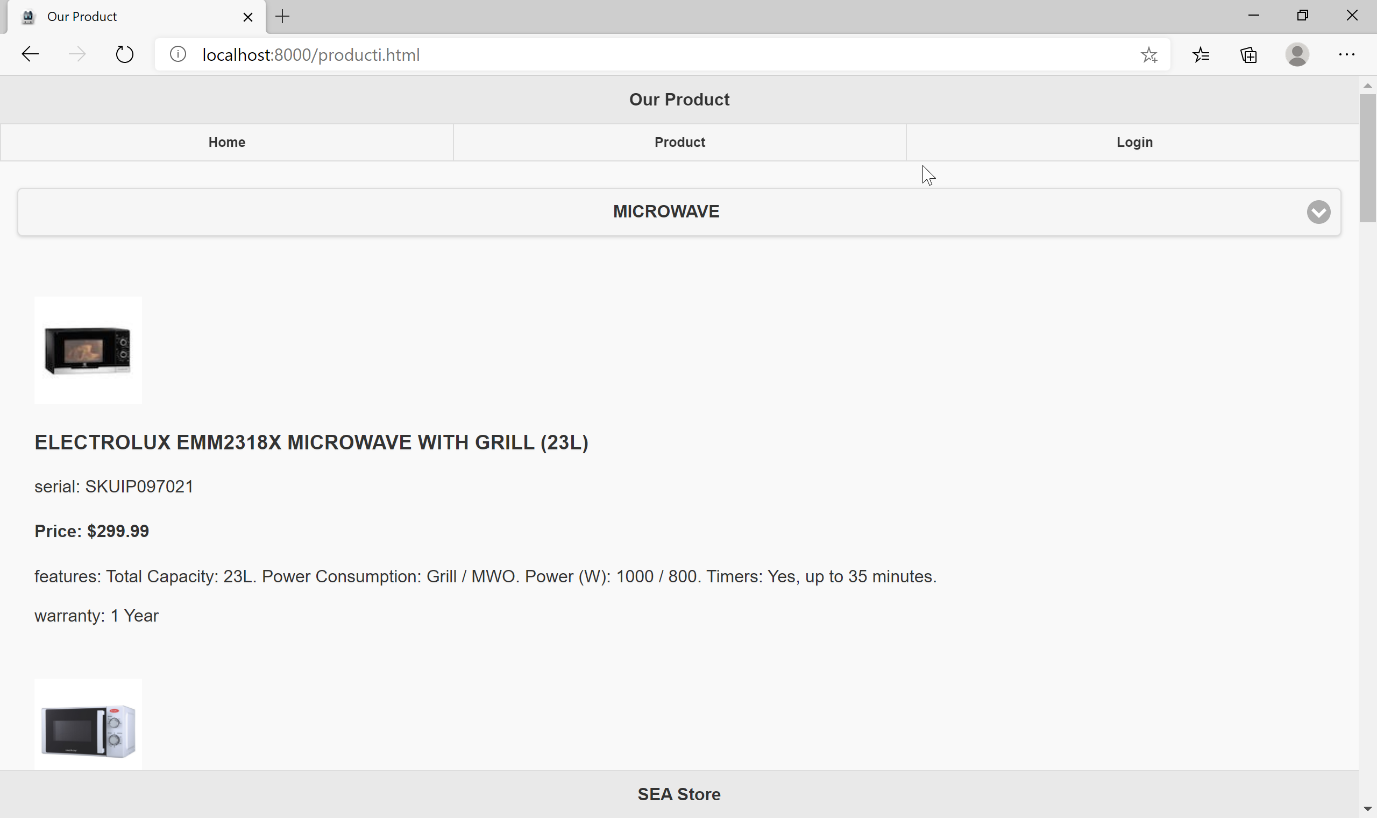
**User can choose which product that they want to see (For both registered and unregistered user) (Browser perspective):**



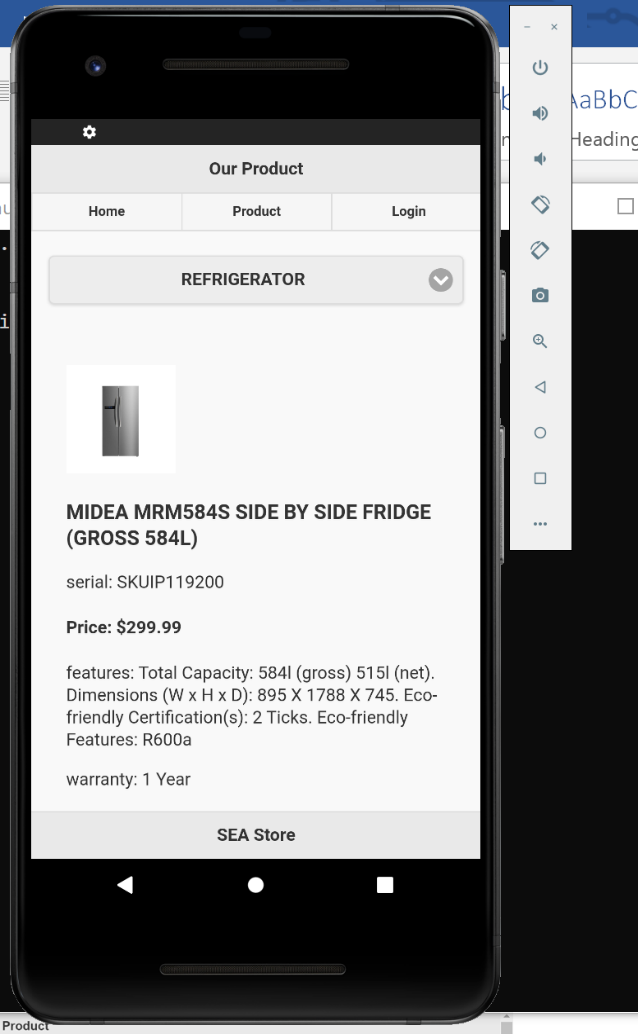
**If user choose Microwave (It consists of 10 microwave products and this page is for both registered and unregistered user)(Android perspective):**

****

**If user choose Microwave (It consists of 10 microwave products and this page is for both registered and unregistered user)(Browser perspective):**



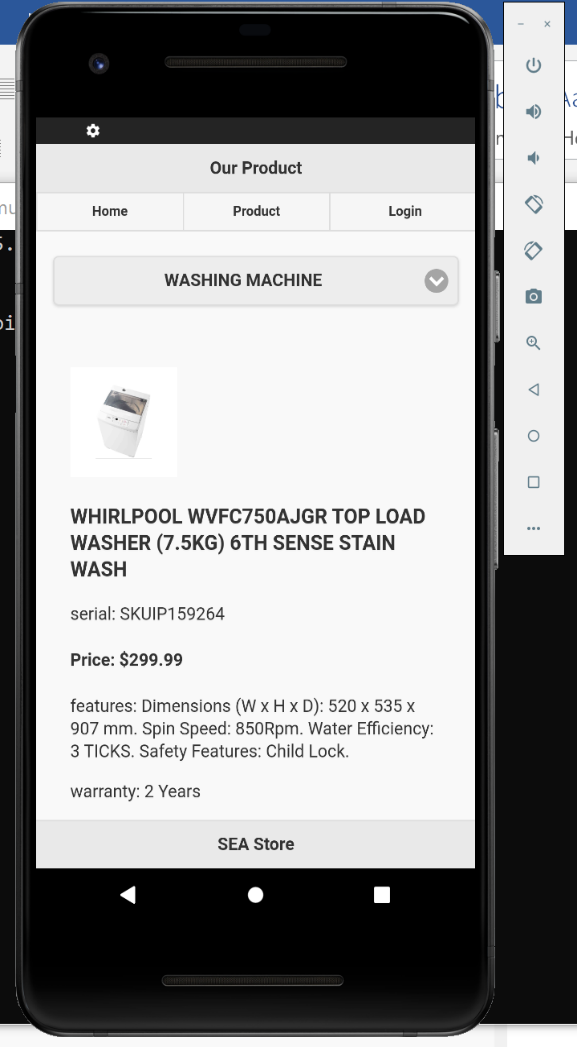
**If user choose Refrigerator ( It consists of 10 refrigerator products and this page is for both registered and unregistered user)(Android perspective):**

****

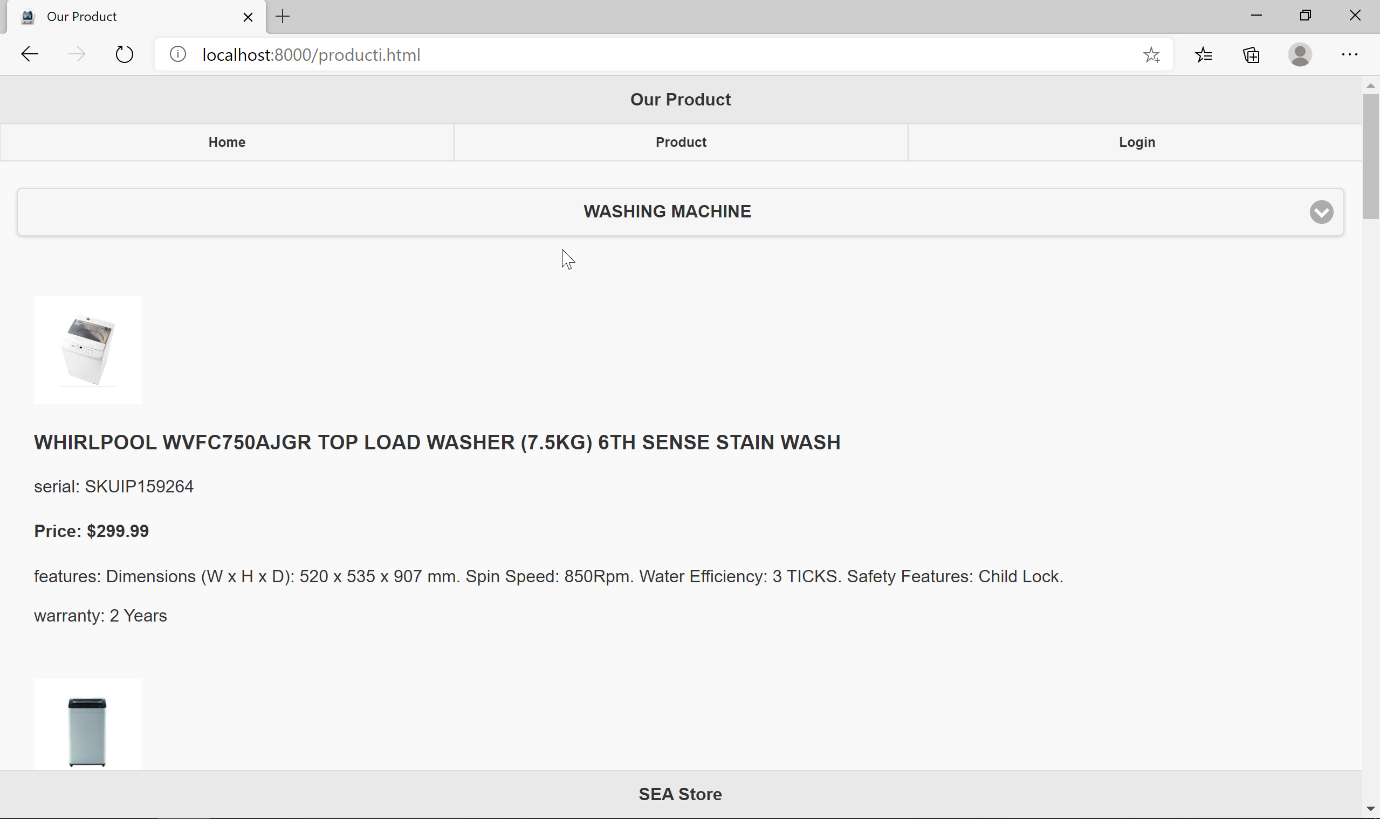
**If user choose Refrigerator ( It consists of 10 refrigerator products and this page is for both registered and unregistered user)(Browser perspective):**



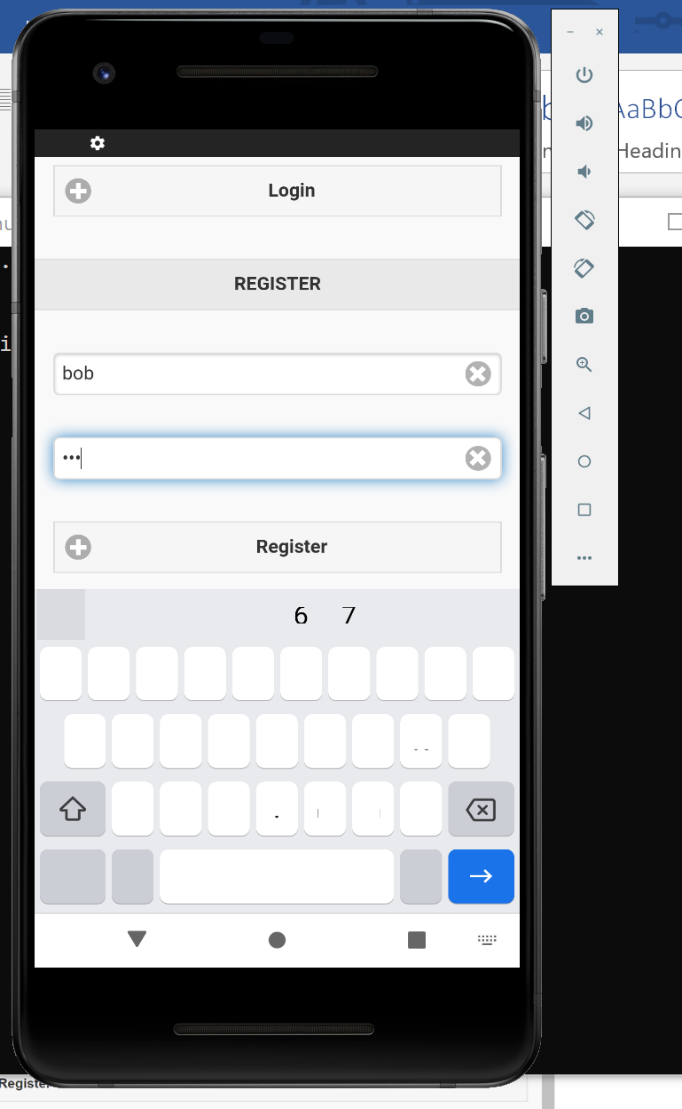
**If user choose Washing Machine ( It consists of 10 Washing Machine products and this page is for both registered and unregistered user)(Android perspective):**

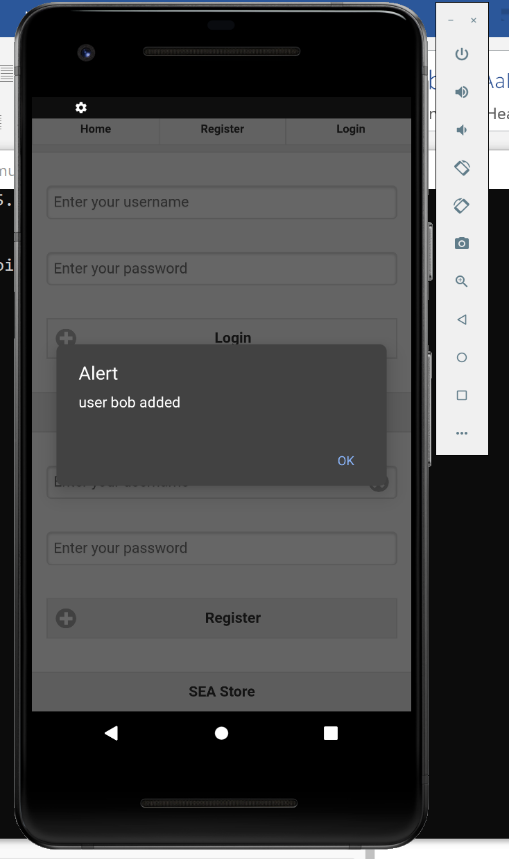


**If user choose Washing Machine ( It consists of 10 Washing Machine products and this page is for both registered and unregistered user)(Browser perspective):**

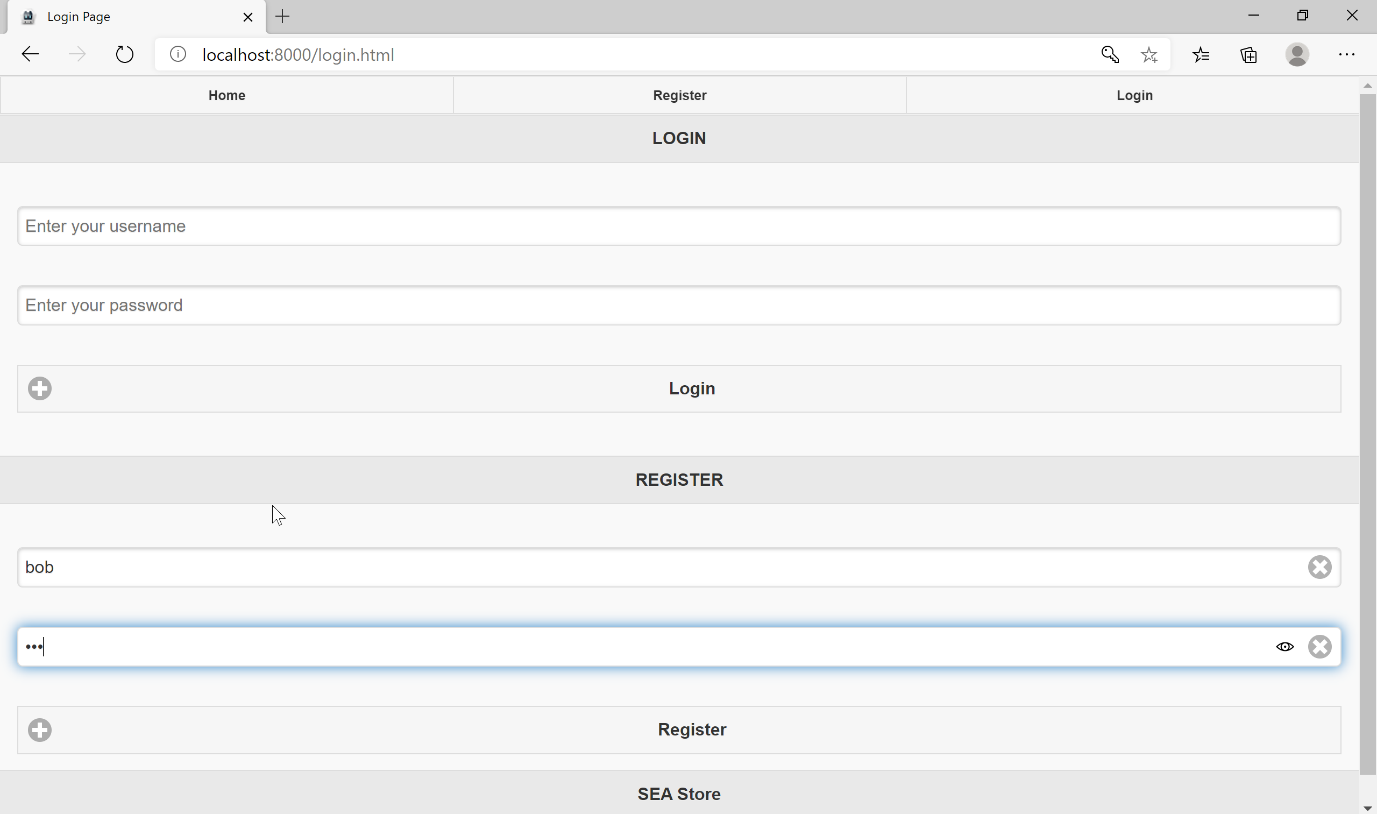


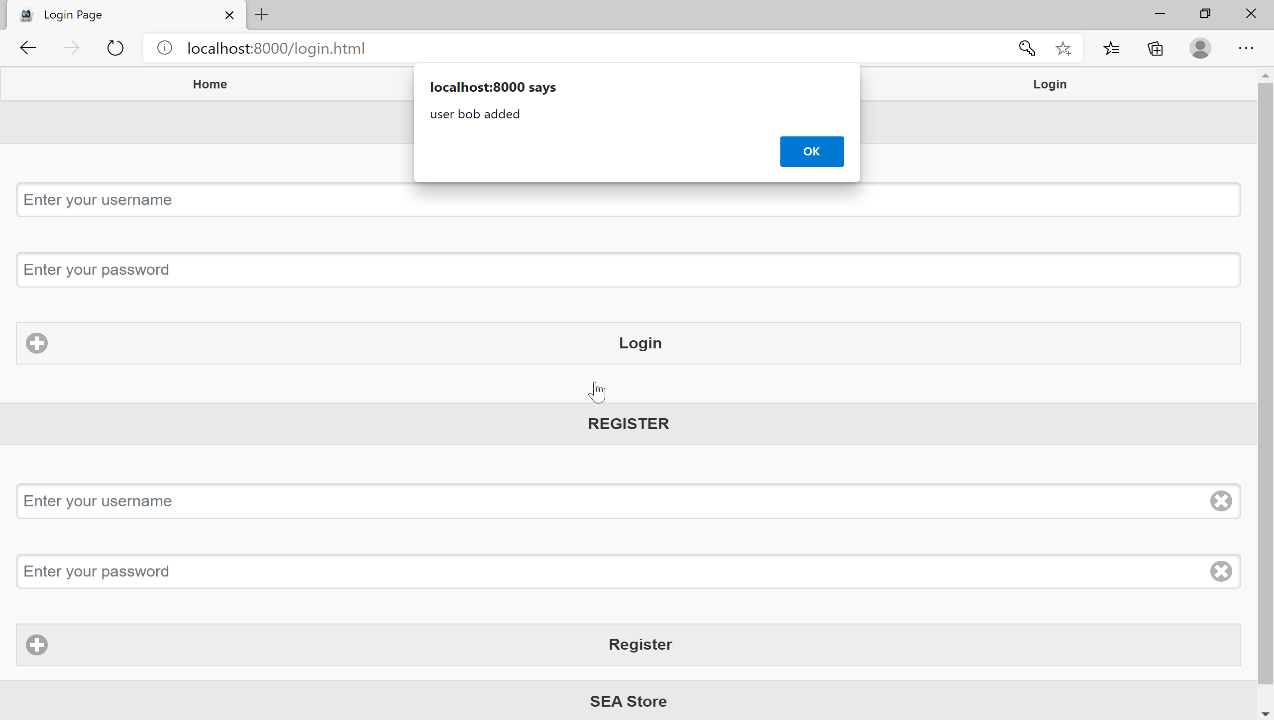
**Process if user want to register (android perspective):**



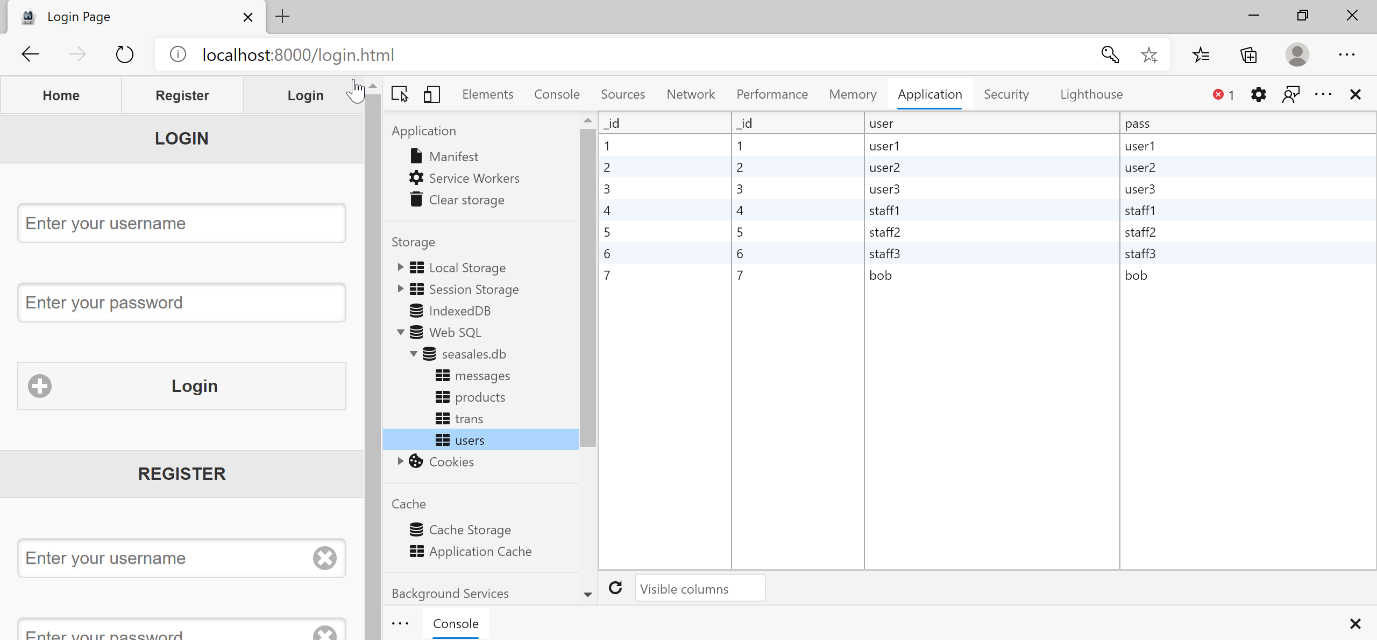


**Process if user want to register (browser perspective):**

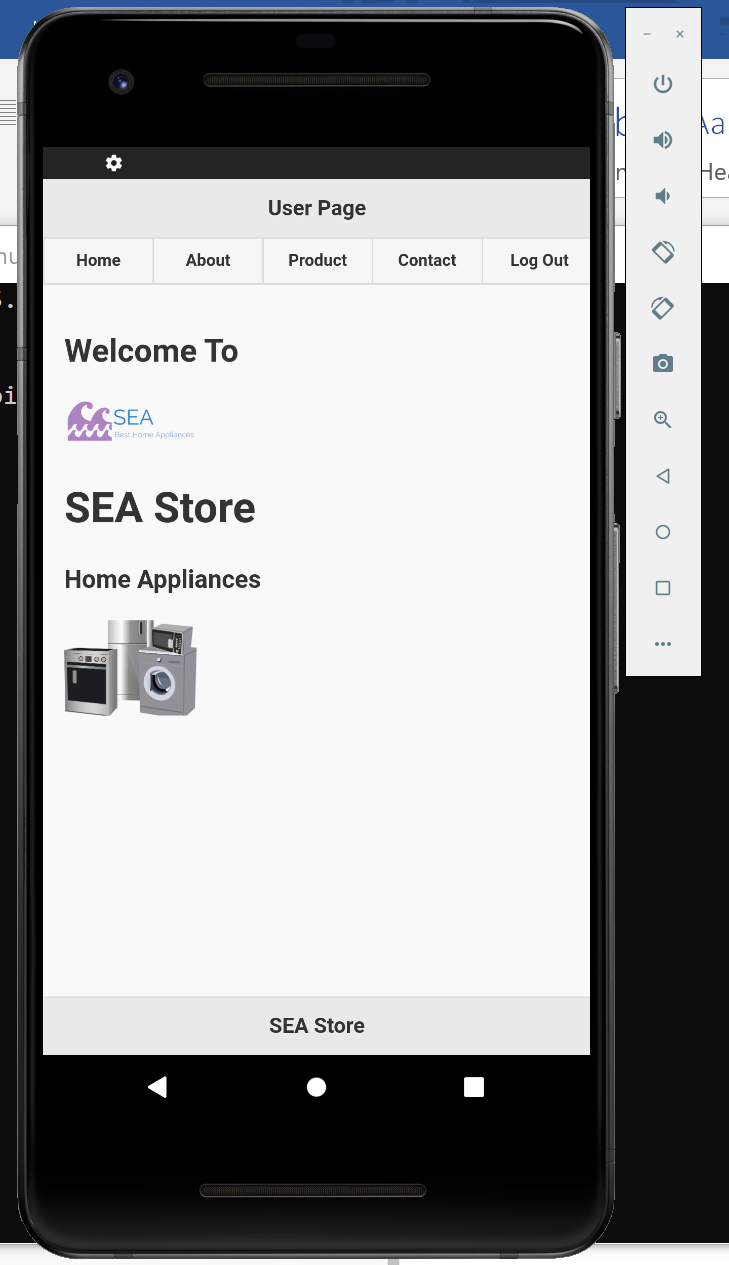




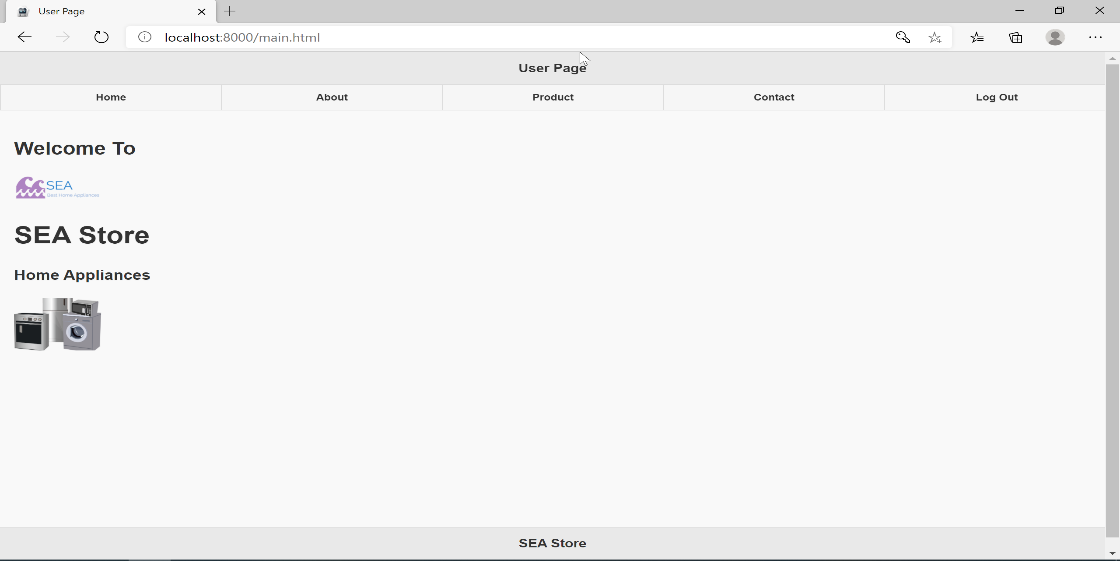
**Proof that the username and password entered is stored in database:**



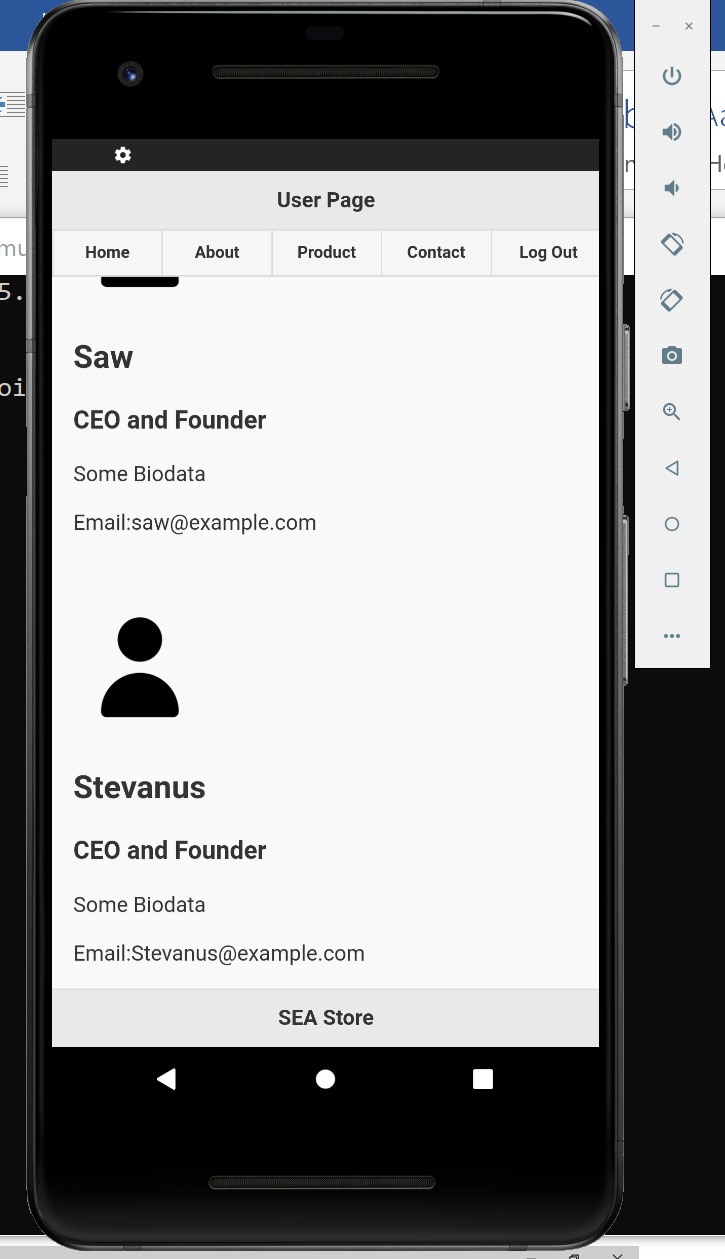
**Home page when user log in (Android Perspective):**



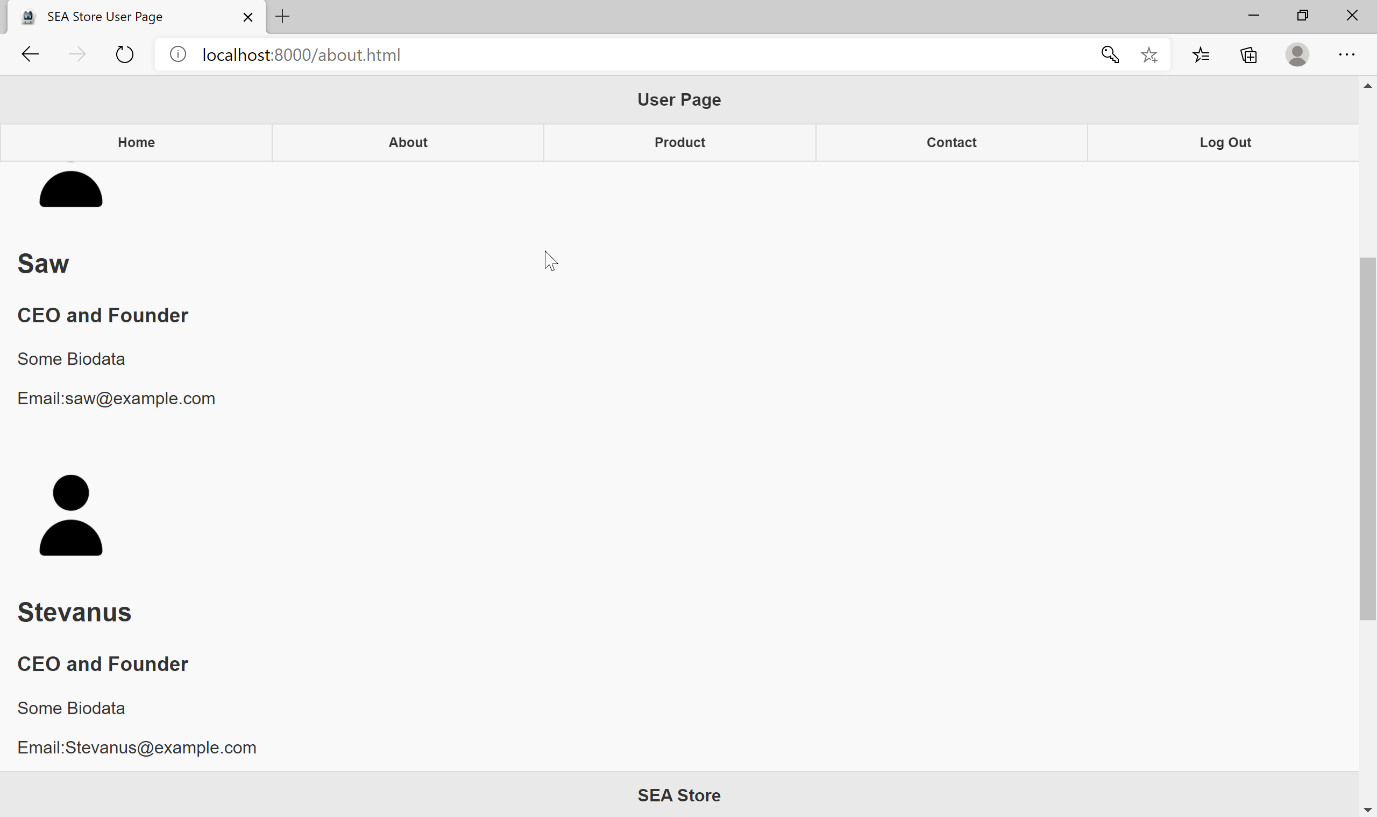
**Home page when user log in (Browser Perspective):**



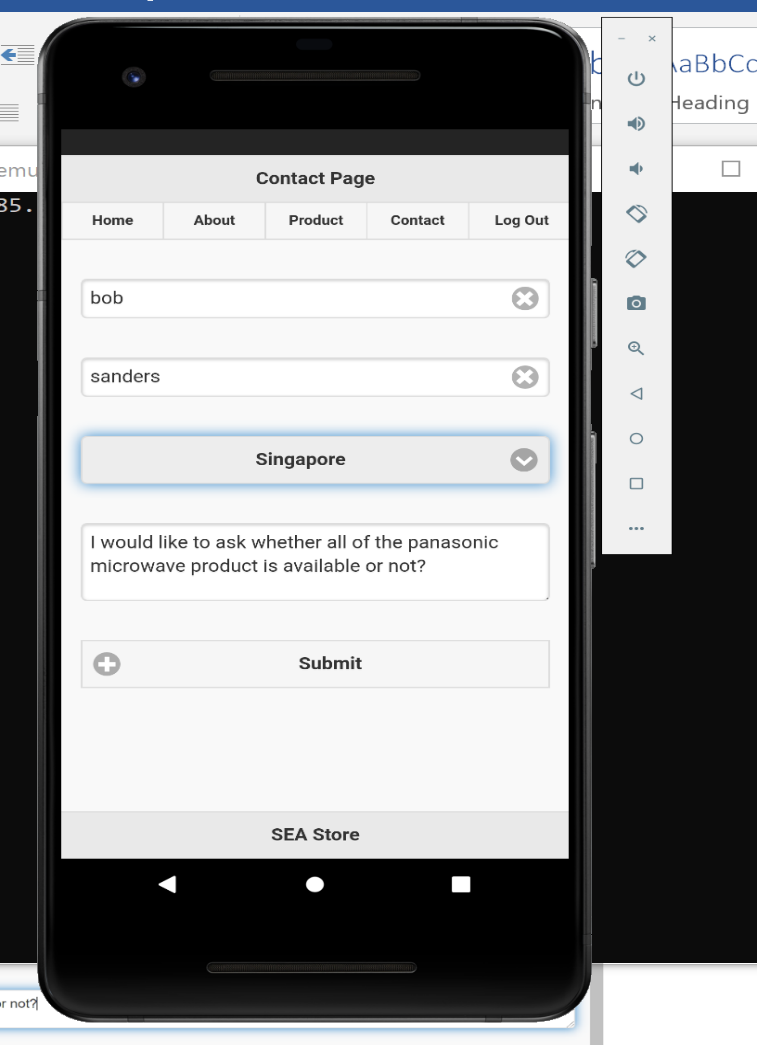
**About page (Accessible after log in only) to see member of groups who becomes ceo in this company: (Android Perspective):**



**About page (Accessible after log in only) to see member of groups who becomes ceo in this company: (Browser Perspective):**

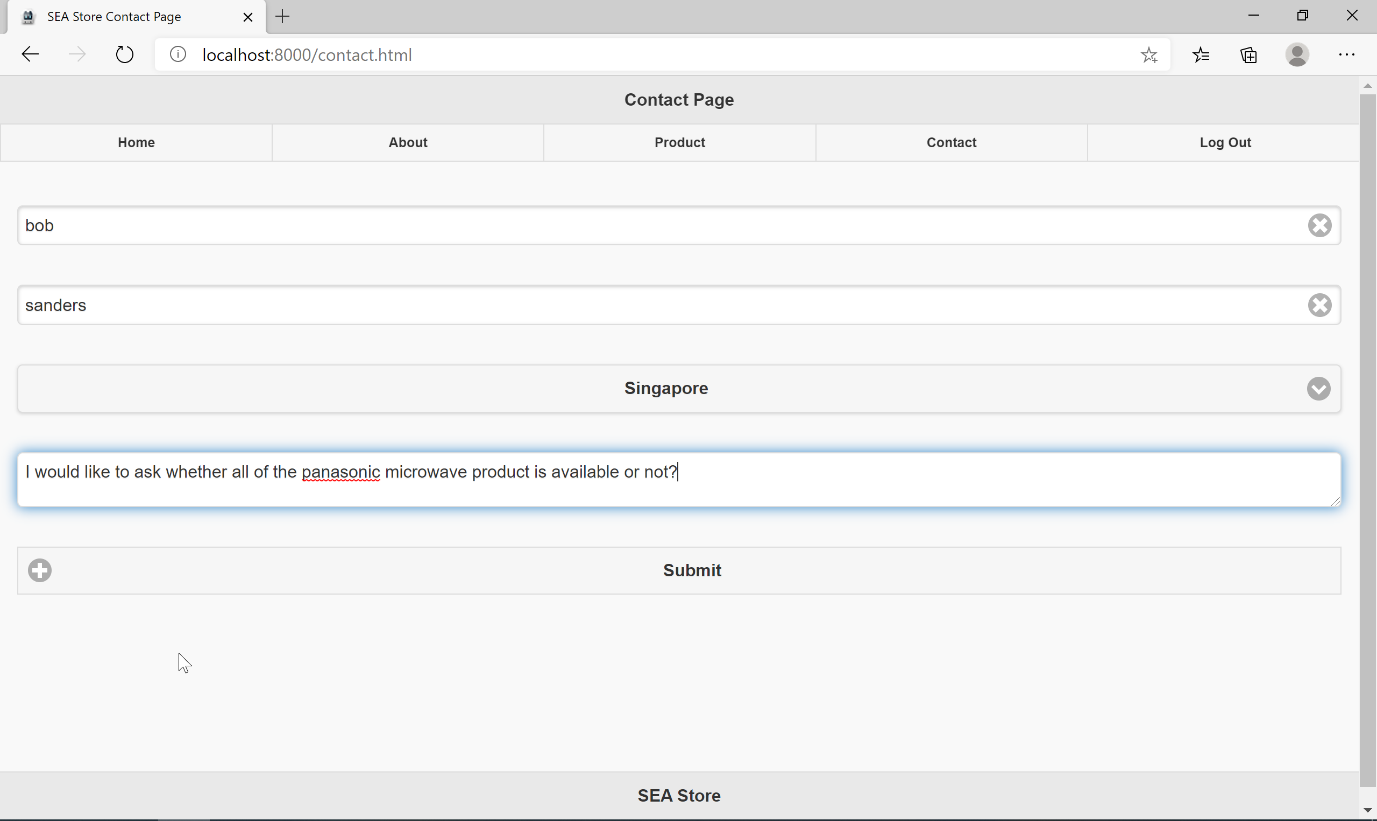


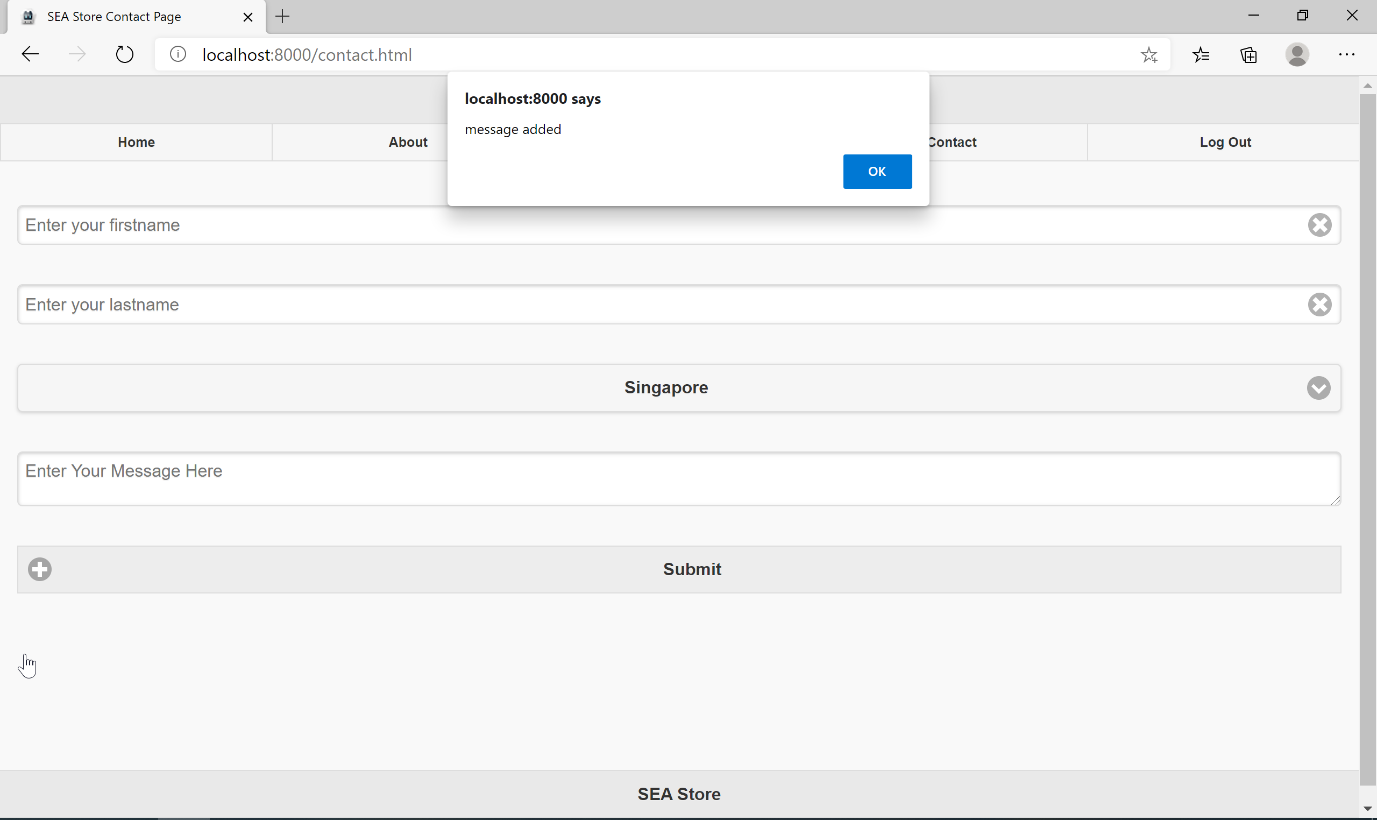
**Process if user want to contact the company to ask question or anything (Accessible after log in only) (Android Perspective):**



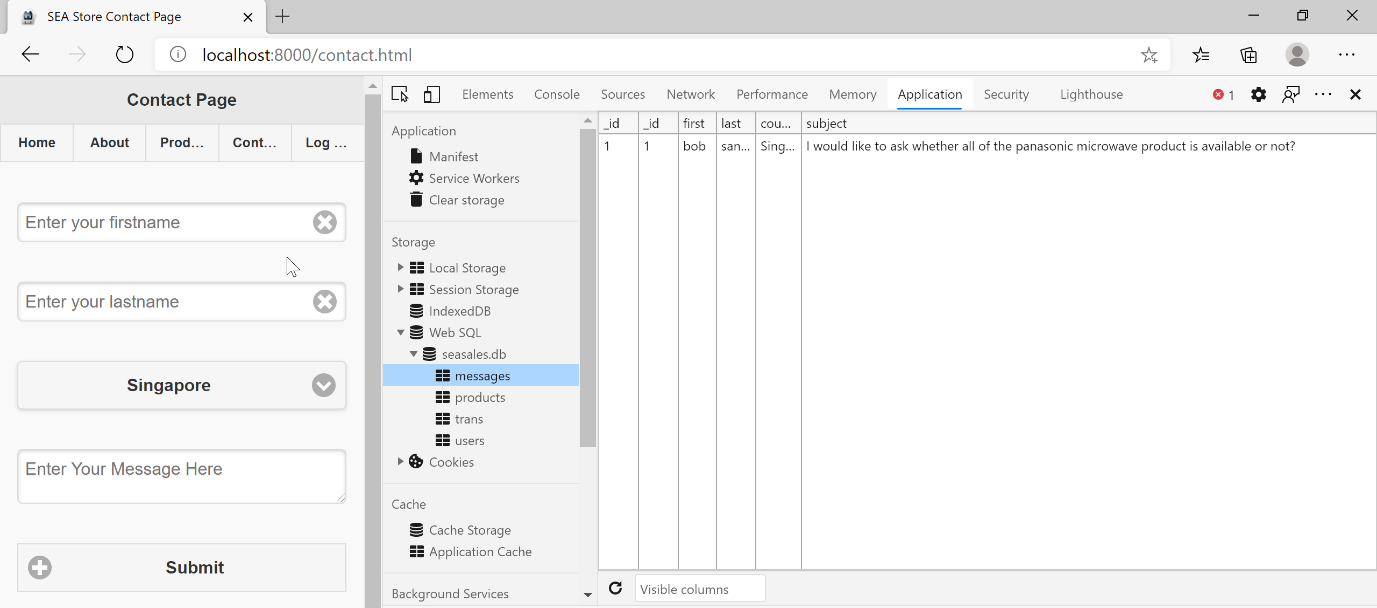


**Process if user want to contact the company to ask question or anything (Accessible after log in only) (Browser Perspective):**

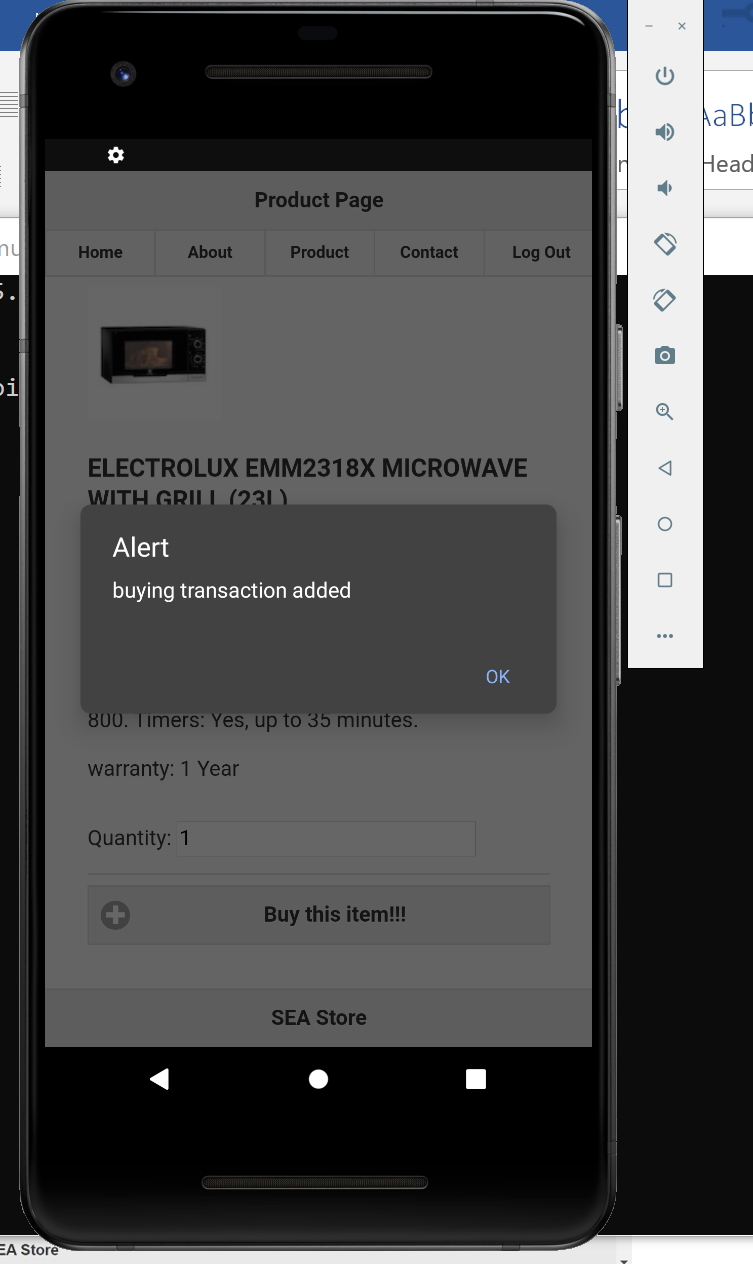
****

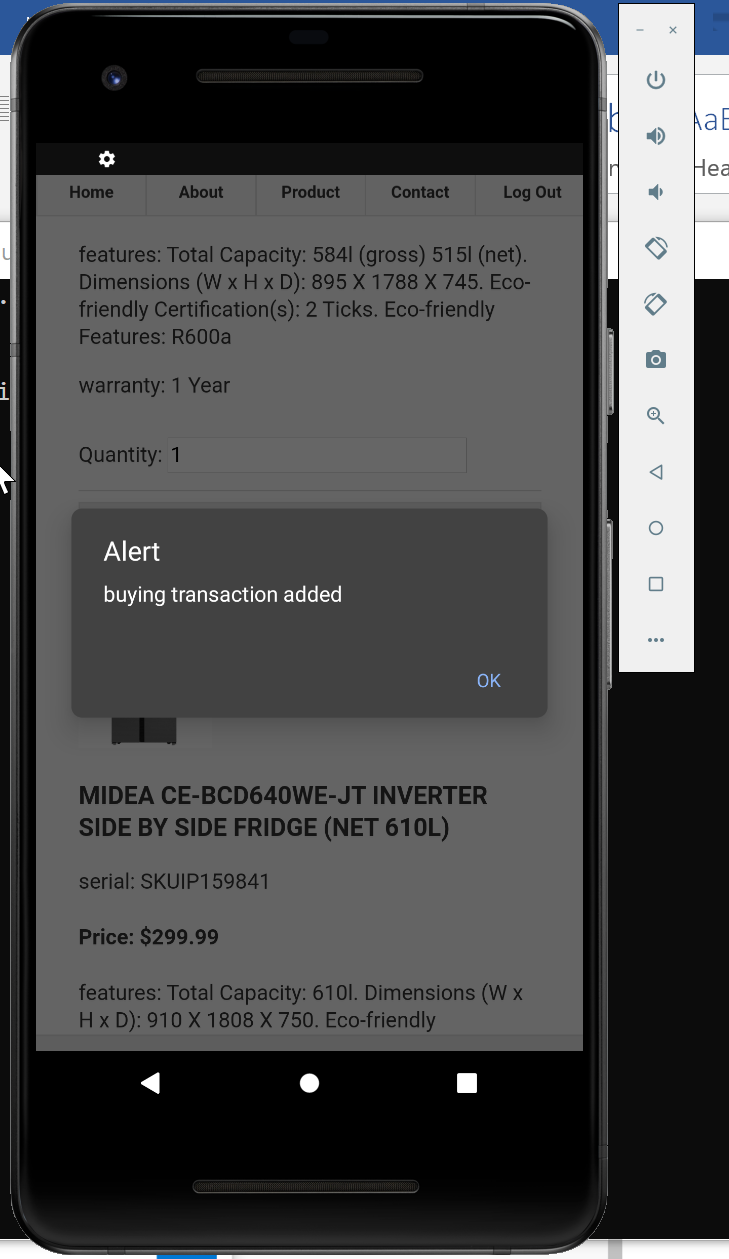
****

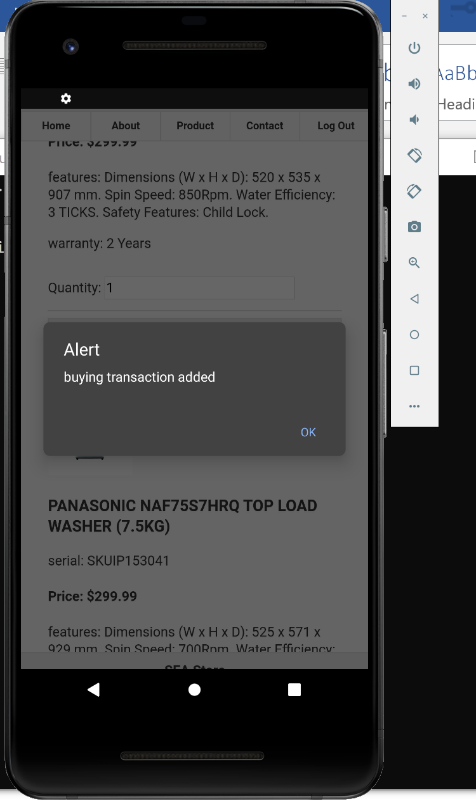
**Proof that all of the first name, last name, country, and message that the user submit is stored in database:**

****

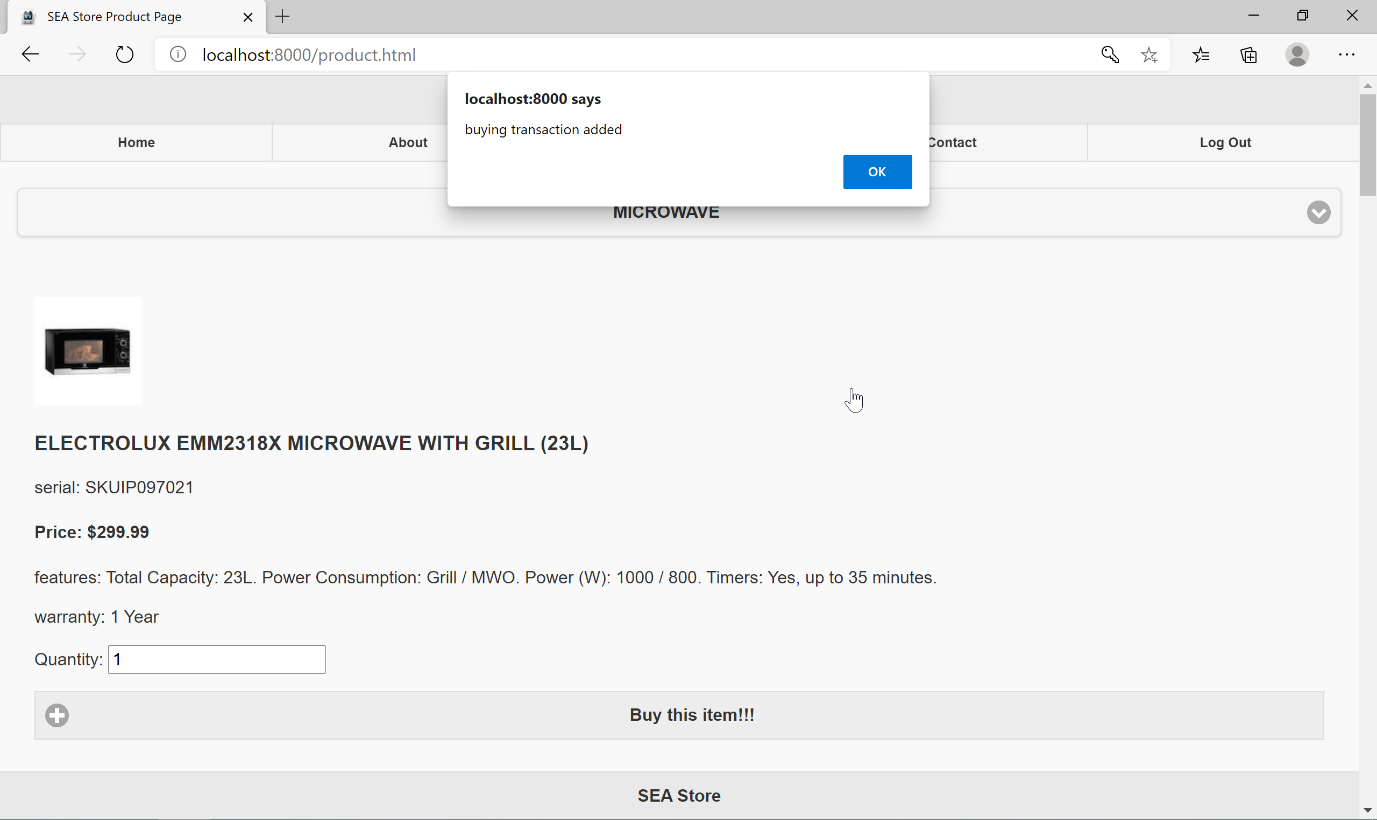
**Process if user wanted to buy the product (Accessible after log in only) (Android Perspective):**

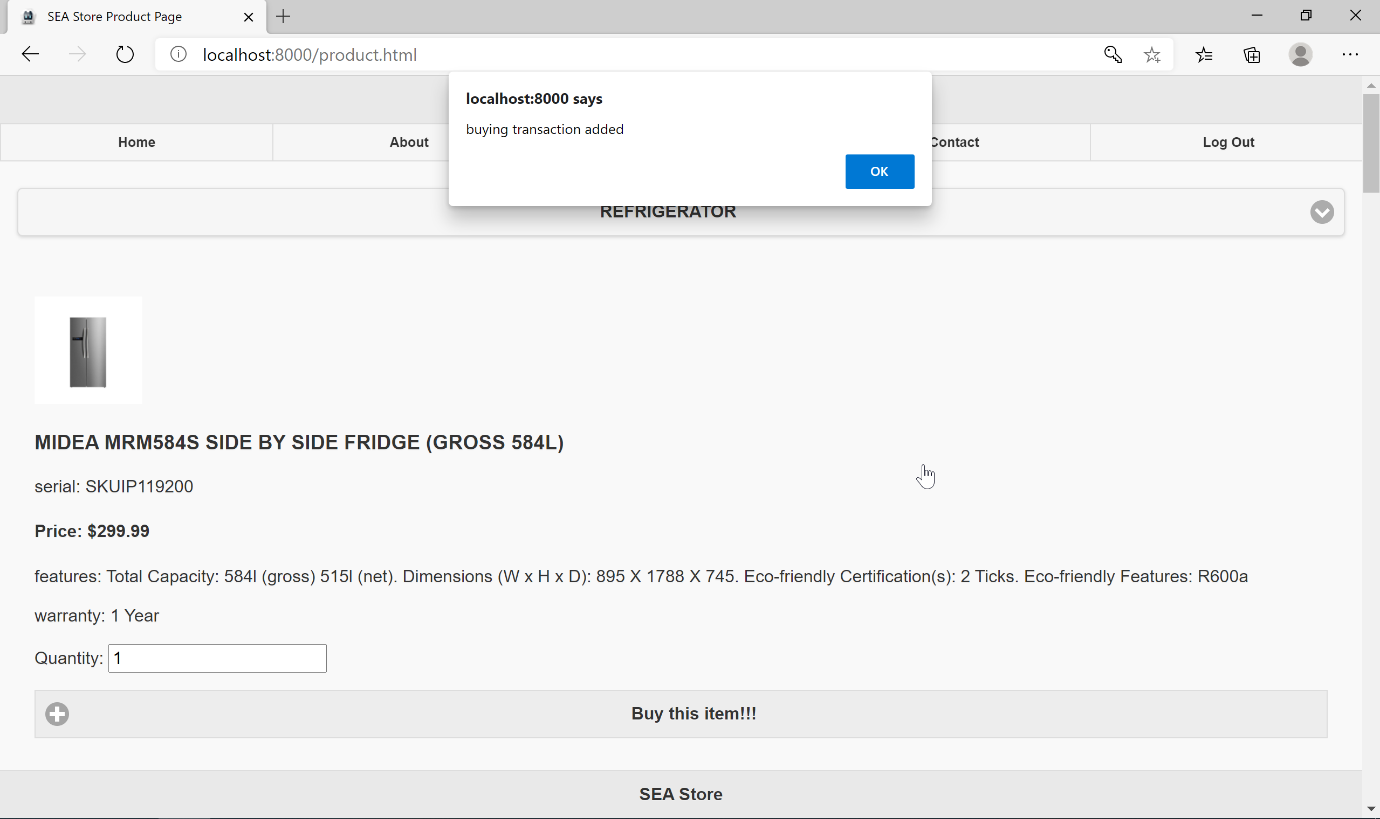


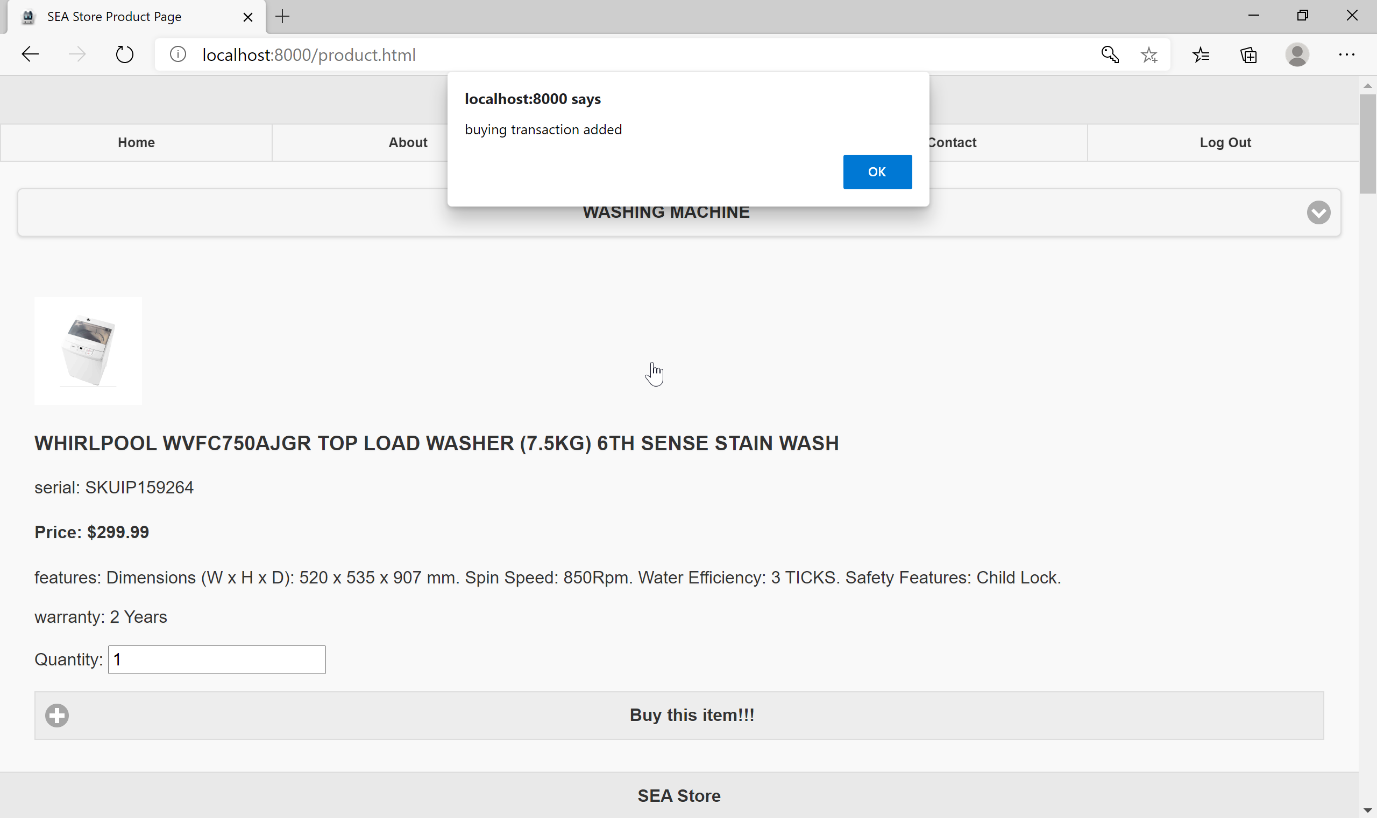




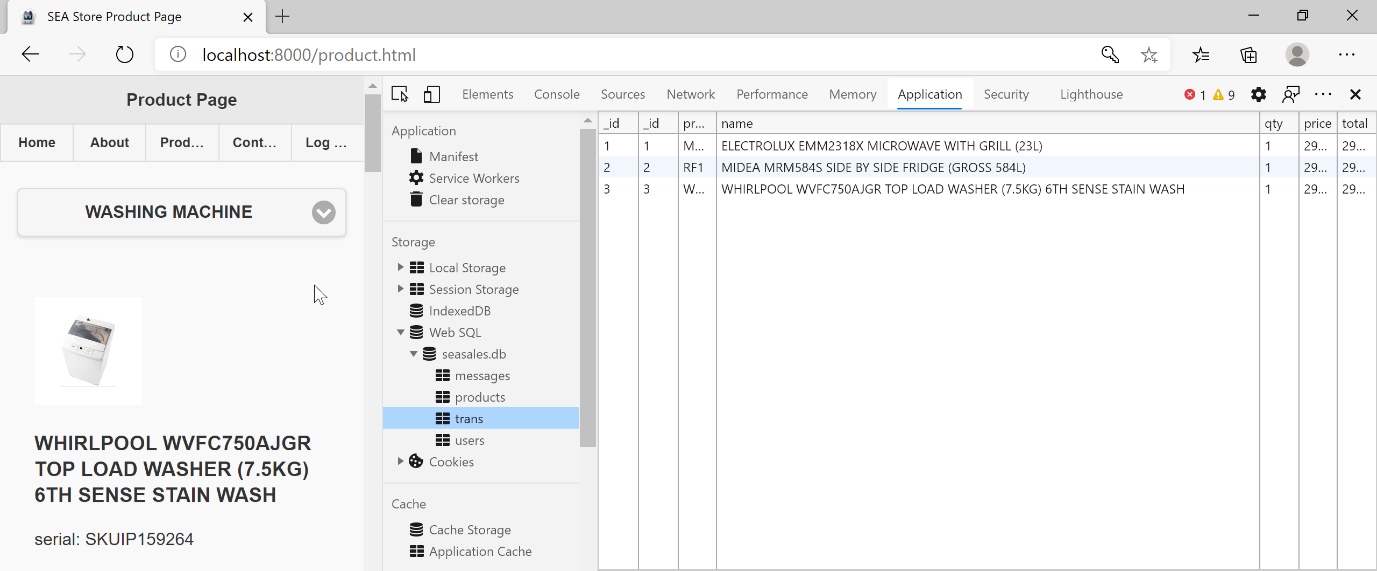
**Process if user wanted to buy the product (Accessible after log in only) (Browser Perspective):**



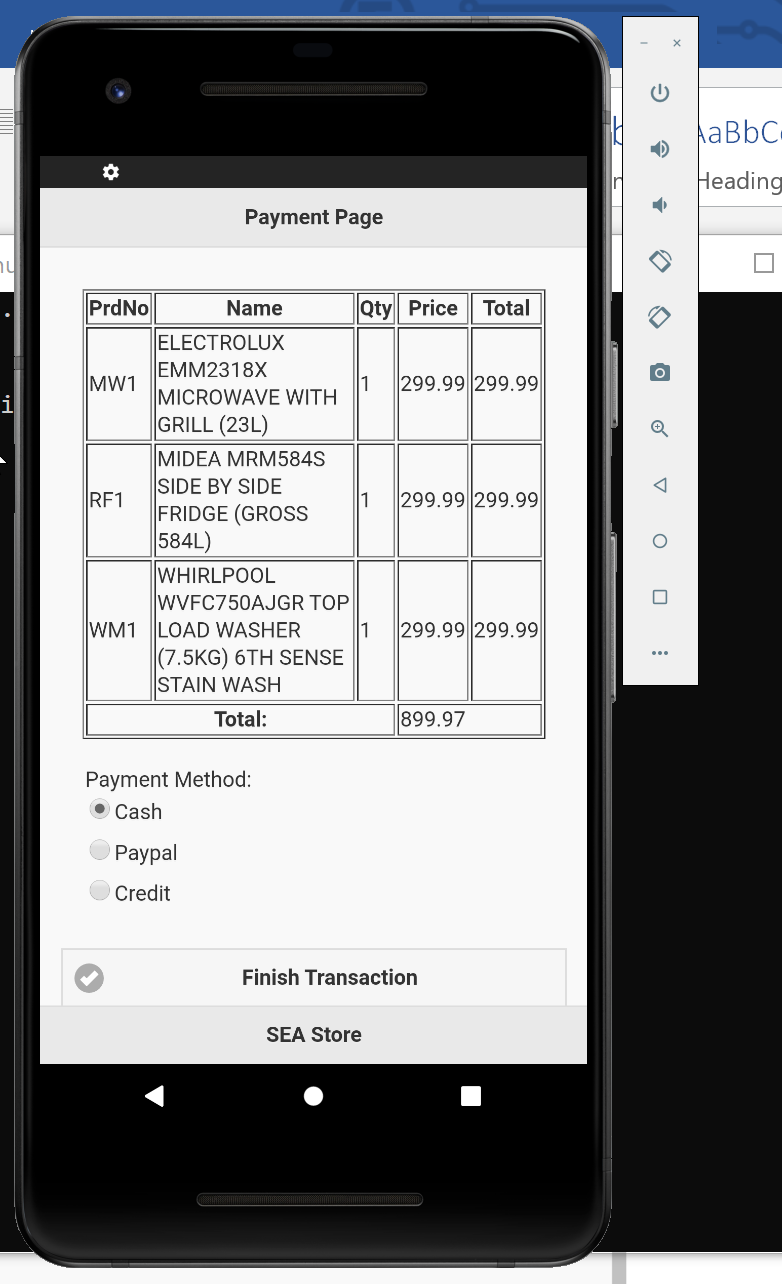




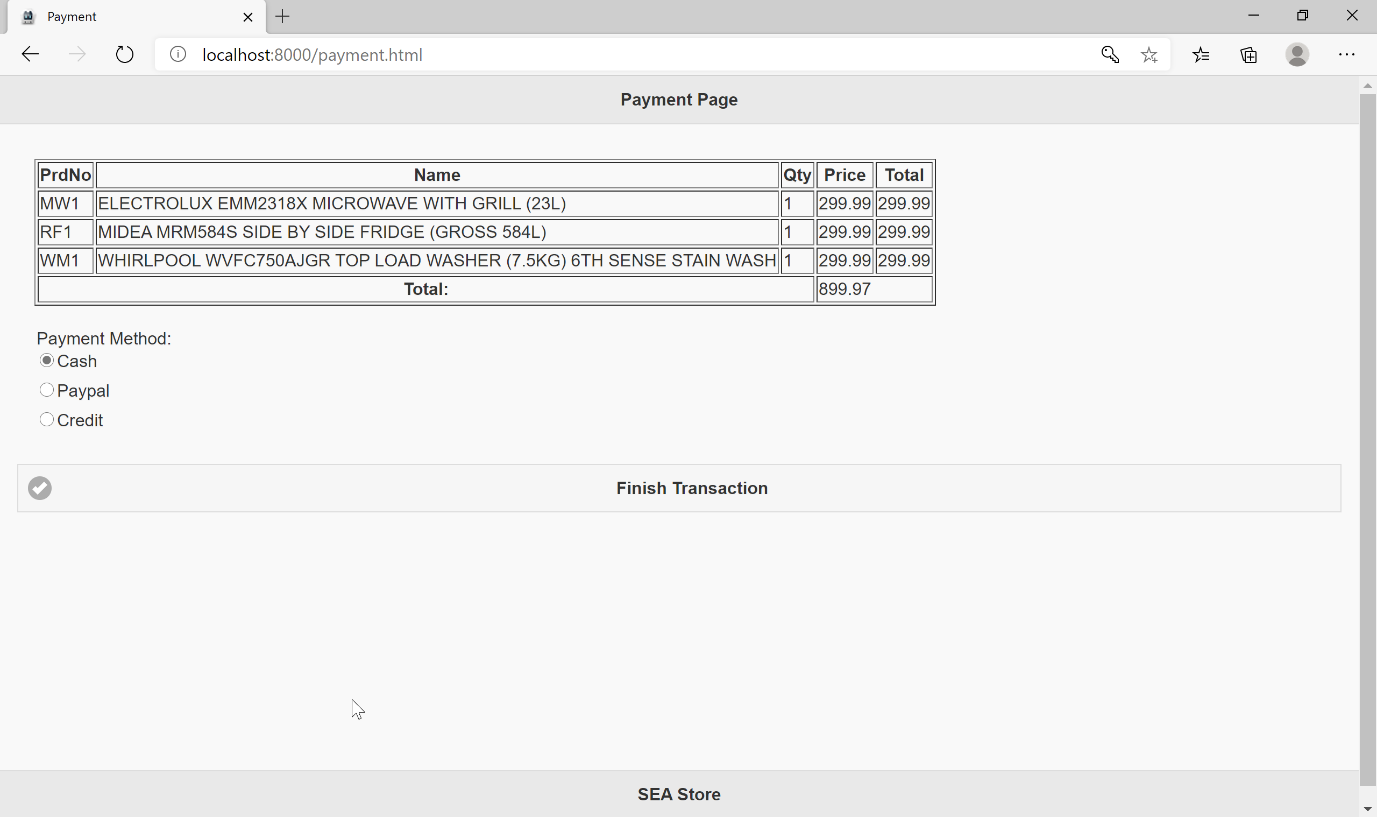
**Proof that all of the product that the user want to buy is stored in database:**



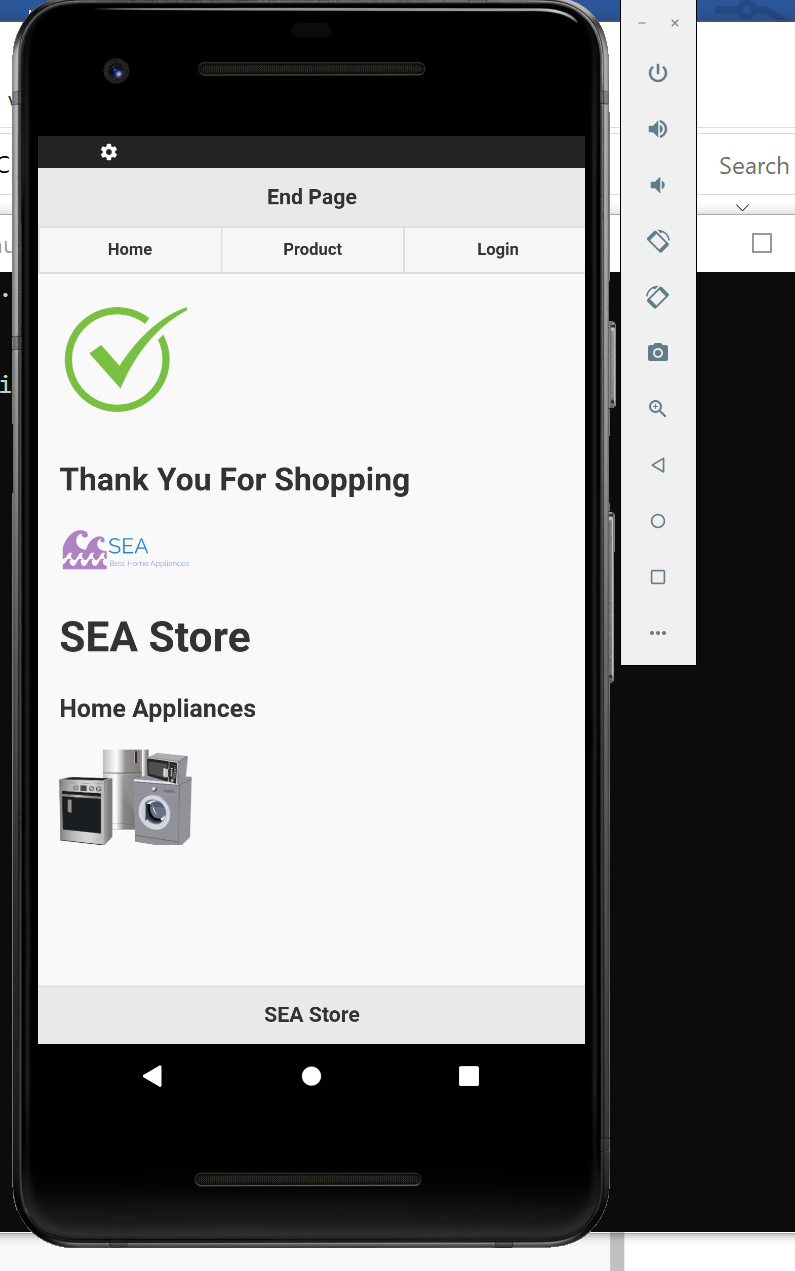
**Payment page after user submit transaction and refresh the page (Android Perspective):**

****

**Payment page after user submit transaction and refresh the page (Browser Perspective):**



**Automatically log out when user finish transaction (Android perspective):**



**Automatically log out when user finish transaction (Browser perspective):**

