

**OLShop**

Project Final Report

OLShop

COMP 4115/6115 Database Systems

Spring Semester, 2015

Submitted to

Dr. Elena Strange

Department of Computer Science

University of Memphis

Prepared and Submitted by

Hari Charan Cheekati

Satya Peri

**TABLE OF CONTENTS**

[1 INTRODUCTION 4](#_Toc381800993)

[1.1 Project Summary 4](#_Toc381800994)

**2 E-R MODELING………………………………………………………………………...5**

2.0 ALGORITHIM………………………………………………………………………........5

[2.1 Entity and Relationship Sets Description 6](#_Toc381800996)

2.2 E-R Diagram ……………………………………………………………………………..9

2.3 Relational Schema (Tables)………………………………………………………….10

2.4 functional dependencies……………………………………………………………….10

**3 COMPONENT DESCRIPTION………………………………………………………11**

**4 TESTING RESULTS AND DISCUSSION………………………………………………….12**

4.1 TESTING RESULTS AND OUTPUTS………………………………………………….12

4.2 FUTURESCOPE………………………………………………………………………….17

**5** **REFERENCES…………………………………………………………………………18**

**1 INTRODUCTION**

## 1.1Project Summary

Our main objective is to design an online electronics shopping website for the customers to buy products which belong to different suppliers. Customers can search for different products sold by suppliers and buy them. This project gives a better online satisfaction of online shopping as we maintain data about every product from different suppliers. We have an administrator who has access to add delete and modifying all the items and the customers who can sign up for shopping online. Customers search for the products and at any time if a customer doesn’t find any item to buy he can request the admin to add that item. Customers can also send feedback about the suppliers and the items they have purchased. We have provided a feedback system to help the administrator to place the product from different suppliers in the olshop website. Based on the search list of the customers we provide them with some suggestions. These suggestions are based on the feedback that the other customer provided about that particular product. Moreover customers can add their products to the cart and can checkout to place the order. Whenever a customer places an order it goes to administrator and admi n can either accept or reject the order. Customers can write queries and Administrator can reply to the customers for their queries. The website will also have catalog of items according to brand, category, and items with this we can give better experience to the customers for their online shopping. Administrator will get an email when a product in the inventory goes low and then he can immediately place the items in the inventory. We will use the real data and customers can have better experience in buying. This associated with lot of database tables and we need to make the relation between tables easier and more convenient.

**Features in our system:**

* Feedback on products: Customers about suppliers and items
* Using feedback: admin adds more good products from suppliers
* Shopping cart: track your items
* Checkout
* Status of order
* Better UI
* Automatic email: Inventory
* Query

**Requirements:**

* MySQL workbench 5.6
* Eclipse
* Java, JavaScript, jsp, and html

**2 E-R MODELING**

**2.0 Algorithm:**

* Step1: Identify every functional dependency.
* Step2: Identify every candidate key in that schema.
* Step3: If there is a functional dependency that has a determinant that is not a candidate key then follow the steps to make a new relation:
* Step 4: Move the columns of that functional dependency into a new relation.
* Step 5: Make the determinant of that functional dependency the primary key of that relation.
* Step 6: Leave a copy of that determinant as a foreign key in the original relation.
* Step 7: Create a referential integrity constraint between the original relation and the new relation.
* Step 8: Repeat step3 until every determinant of every relation is a candidate key

## 2.1 Entity and Relationship Sets Description

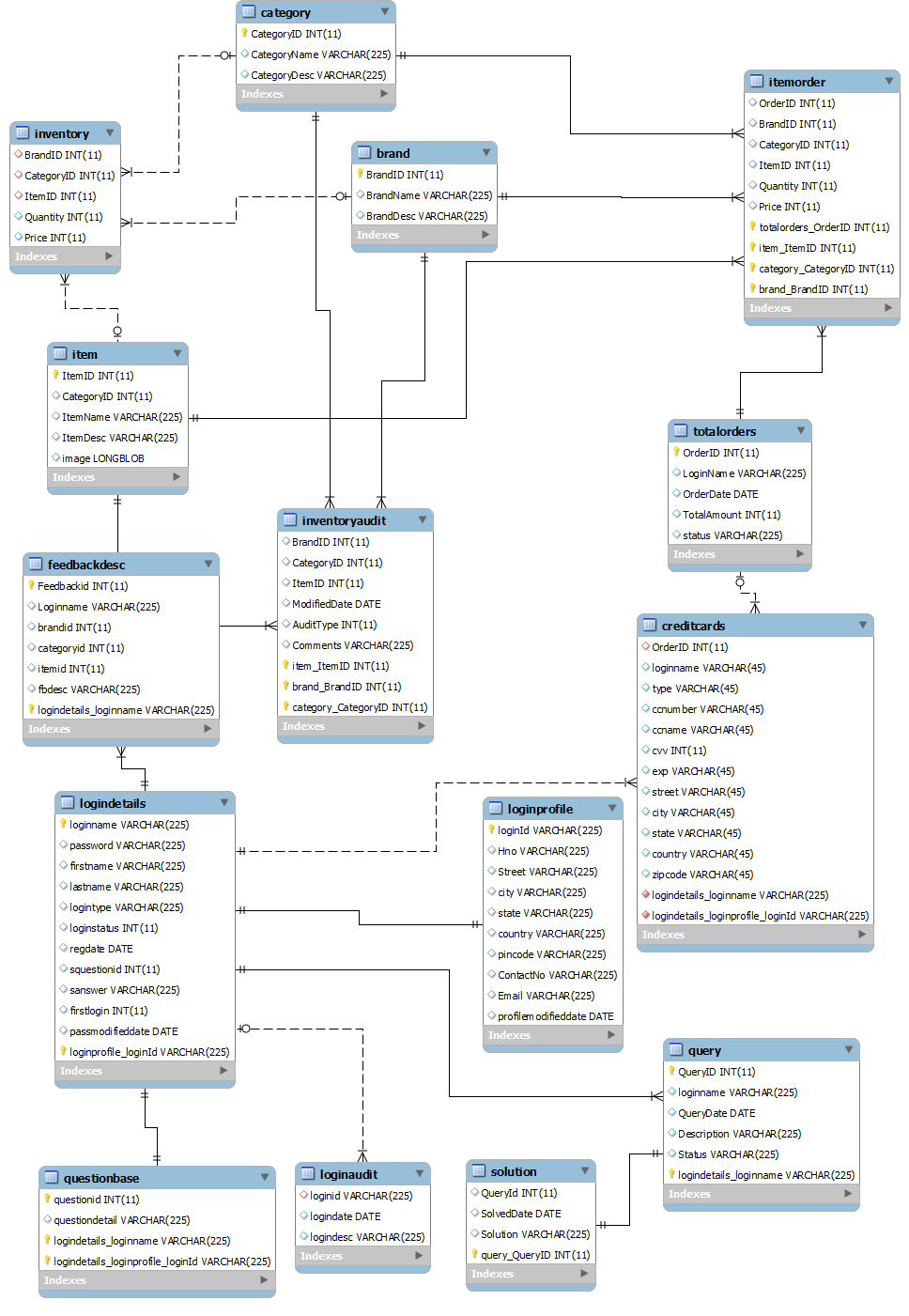
Our system has the following entities and represented with their corresponding attributes.

The entities involving in this system are

* brand
* category
* item
* inventory
* itemorder
* loginprofile
* logindetails
* loginaudit
* totalorders
* inventoryaudit
* feeydbackdesc
* questionbase
* query and
* solution

Each and every entity has their own corresponding attributes:

* **brand** 
  + brandid
  + brandname
  + branddesc
* **category**
  + categoryid,
  + categoryname
  + categorydesc
* **item**
  + itemid
  + categoryid
  + itemname
  + itemdesc
* **inventory**
  + brandid,
  + categoryid
  + itemid
  + quantity
  + price
* **itemorder**
  + orderid
  + brandid
  + categoryid
  + itemid
  + quantity
  + price
* **totalorders**
  + orderid
  + loginid
  + orderdate
  + totalamount
  + status
* **inventoryaudit**
  + brandid
  + categoryid
  + itemid
  + modifieddate
  + audittype
  + comments
* **feedbackdesc**
  + feedbackid
  + loginname
  + brandid
  + categoryid
  + itemid
  + feedbackdesc
* **logindetails**
  + loginid
  + passwd
  + firstname
  + lastname
  + logintype
  + loginstatus
  + regdate
  + squestionid
  + sanswer
  + firstlogin
  + passmodifieddate
* **loginprofile**
  + loginid
  + aptno
  + street
  + city
  + state
  + country
  + zipcode
  + mobileno
  + email
  + profilemodifieddate
* **loginaudit**
  + loginid
  + logindate
  + logindesc
* **questionbase**
  + questioned
  + questiondetail
* **query**
  + queryid
  + loginname
  + querydate
  + description
* **solution**
  + queryid
  + solvedate
  + solution
* **creditcard**



2.2 ENTITY RELATIONSHIP DIAGRAM

Based on the ER diagram some of the important details are

We have two separate tables for login details and login profile which are one to one related. Item, category and brand tables are linked to item orders. All the orders placed are stored in total orders. After proceeding to check out the card details given are stored in credit cards.

## 2.3 Relational Schema (Tables)

The ER diagram is drawn by keeping in the mind that how each and every entity interact with each other, how the features can. As we have different entities such as Brand, Category, Item, users can register with their details and log into the Store and then they can choose any specific brand like apple, Samsung etc, category like latptops, cellphones and Item like samsungS4, Apple Chromebook, etc. Finally they can place an order by adding the item to the cart. User have a flexibility to give a query or feedback about the products to the Admin and the Admin needs to keep track of the all the queries and give the necessary solution. For example, Admin gets the information regarding the lack of products then an automatic email is sent to the supplier to notify. In addition to this, Admin needs to keep track of the Inventory and InventoryAudit and LoginAudit to know the status of the products.

## 2.4 Functional Dependency Studies

* **brand**:

brandid ->( brandname, branddesc )

* **category**:

categoryid ->( categoryname, categorydesc)

* **item**:

itemid -> ( categoryid, itemname, itemdesc)

* **itemorder**:

ordered -> ( brandid, categoryid, itemid, quantity, price)

* **feedbackdesc**:

feedbackid -> ( loginid, brandid, categoryid, itemid, feedbackdesc )

* **loginprofile**:

loginid->( birthdate, aptno, street, city, state, country, zipcode, mobileno, email, locale, profilemodifieddate)

email->(loginid, aptno, street, city, state, country, zipcode, mobileno, profilemodifieddate)

* **loginaudit**:

loginid -> ( logindate, logindesc)

* **query**:

queryid -> (loginname,querydate,description)

* **questionbase**:

questioneid ->( questiondetail)

# 

## 3 COMPONENT DESCRIPTION

1. **Admin**: The administrator is responsible for maintaining the system. The administrator can search the Items, Category and the available Brands. He can view the Inventory details by Item, Category and Brand and also add /delete any Item, Category and Brand in the Inventory. The main responsibility of administrator is to view the customer orders and accept their orders. In addition this, he can provide solutions to the queries that has been sent by the customers. Moreover, he provided an automatic email to the supplier in case of low Inventory of respective Items.
2. **Customer**: The customer can login to the website and view or search the Items by Category and Brand to buy and add them to cart after, placing the order. In addition to this, they can manage their accounts to change their personal details information.
3. **Login**: Each user either admin or customer/user should provide their login credentials to access the system to buy the products and manage their account details.
4. **Login** **Failed**: If the login credentials are provided incorrect then the login fails and then user may have to give another attempt to login to the system.
5. **View Items, Brand and Category**: Either customer or admin can view the available brands, respective categories and their associated Items. Customers/Users can place orders to the Items. Admin can add the Brands, Category and their associated Items in order to keep track of the Inventory details.
6. **Add to Cart**: After selecting the items from the system, the customer can add them to their respective carts in order to place an order.

## 

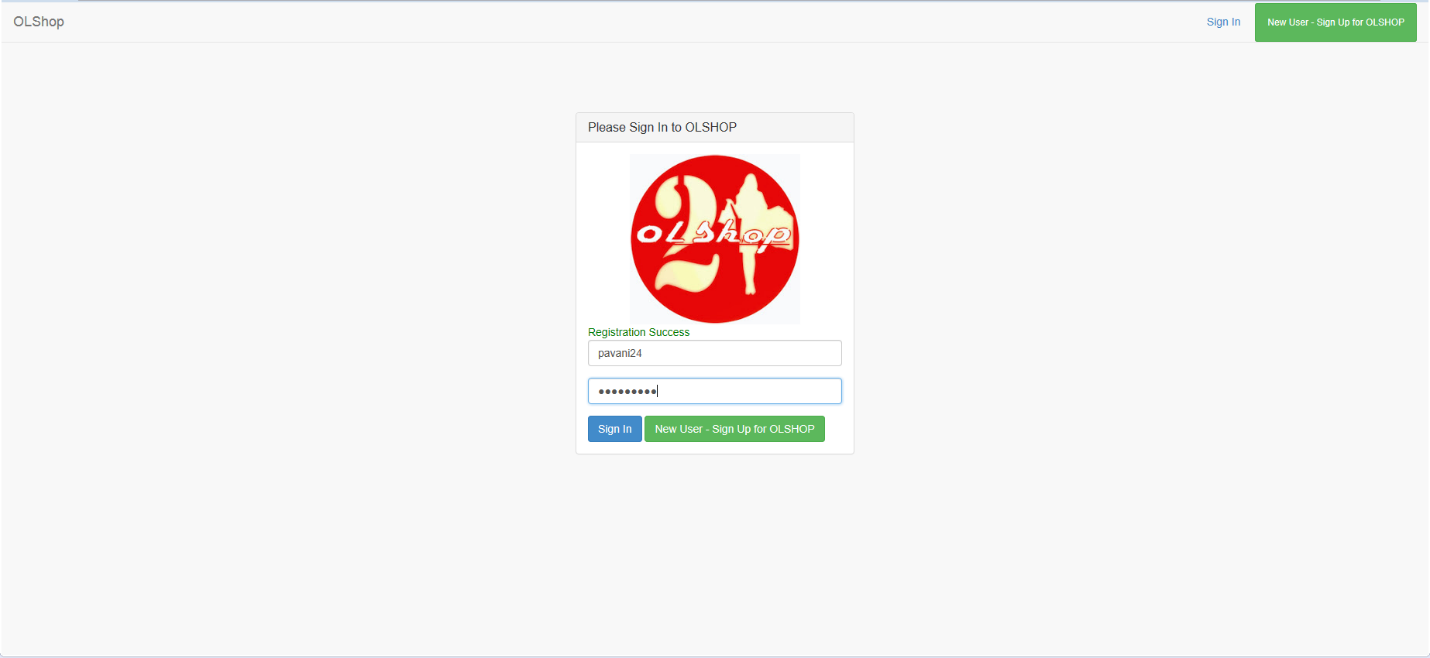
# 4 tESTING RESULTS AND OUTPUTS

## 4.1 TESTING RESULTS

**1.REGISTRATION TEST CASES:**

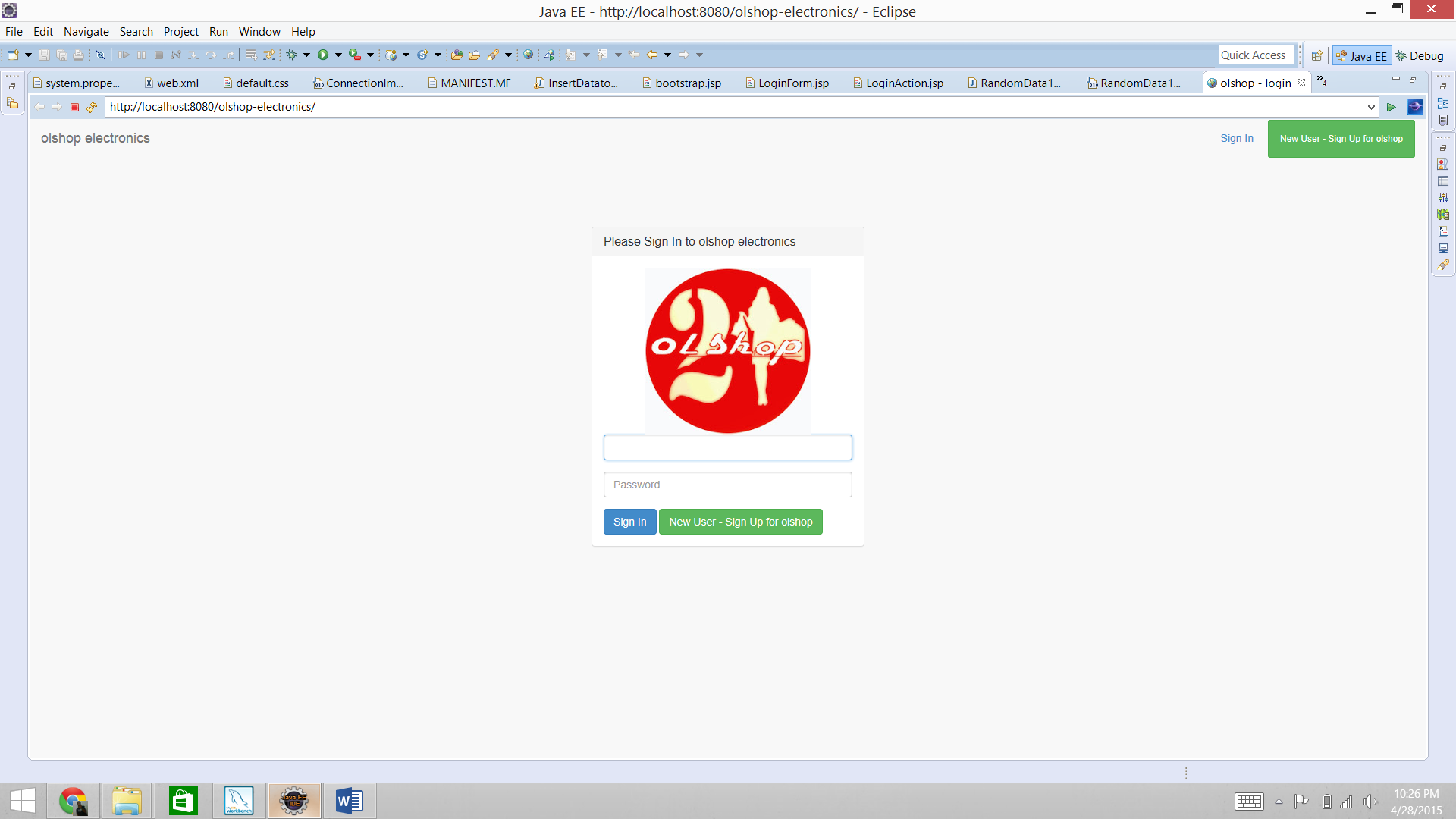
|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Expected Result** | **Pass/Fail** | **Remarks** |
| 1 | “Enter details for registration” | Pass | Working as expected |
| 2 | “Account Creation with  Username: pavani24  Password: pavanisatya | Pass | Working as expected. |

1. **REGISTRATION SUCCESS:**

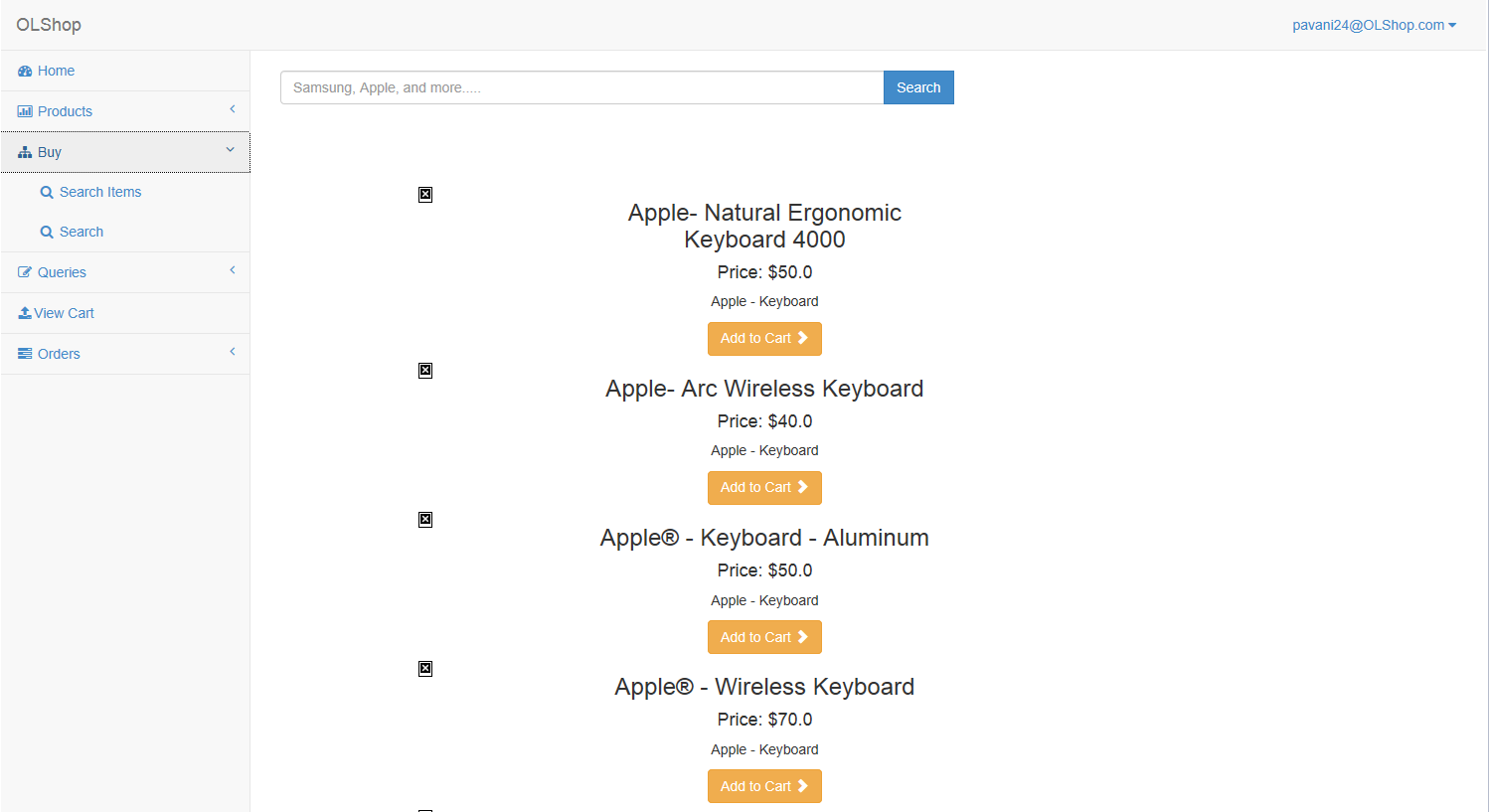


**2. LOGIN TEST CASES:**

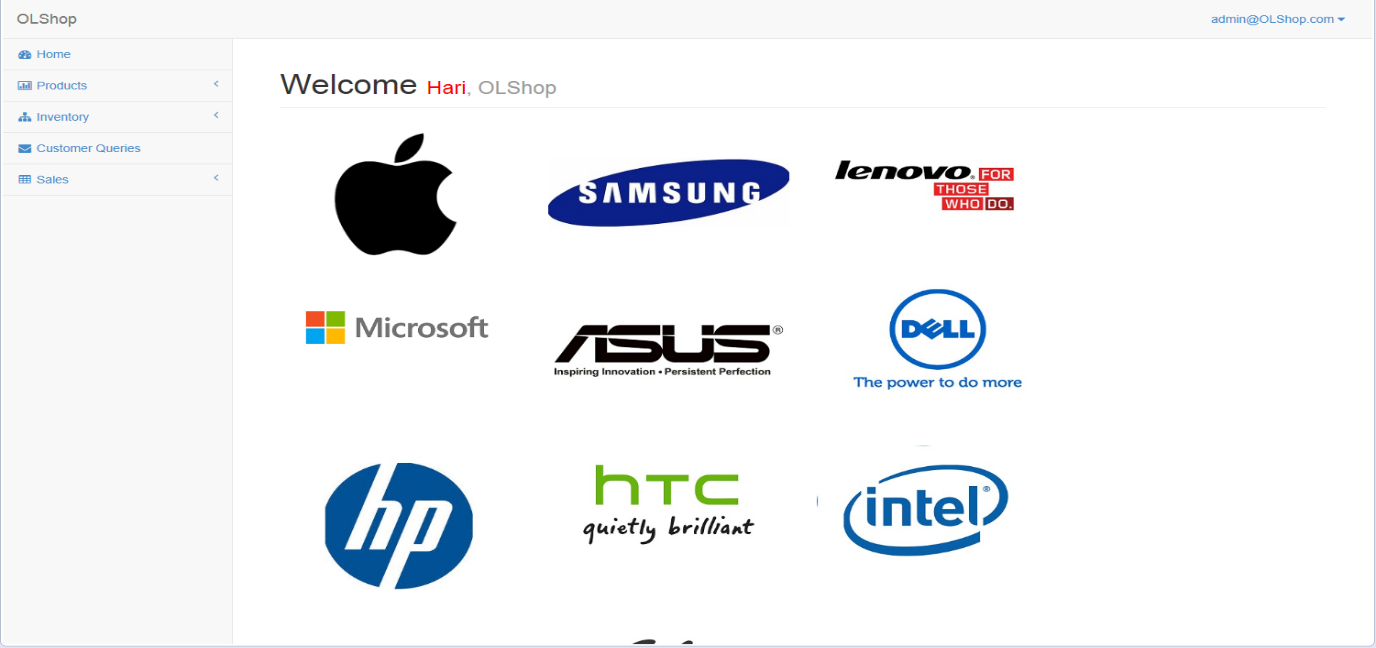
|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Expected Result** | **Pass/Fail** | **Remarks** |
| 1 | “Enter credentials to login” | Pass | Working as expected |
| 2 | “Incorrect password or username. Fail to login” | Pass | Working as expected. |



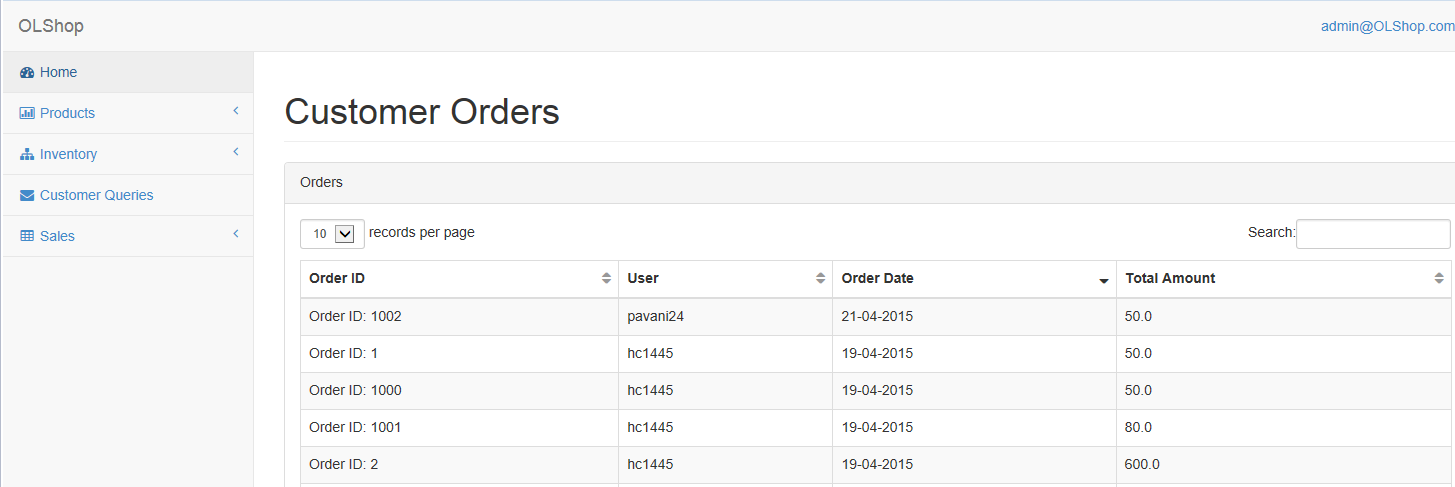
**3. ITEM SEARCH**



**4. ADMIN PAGE**

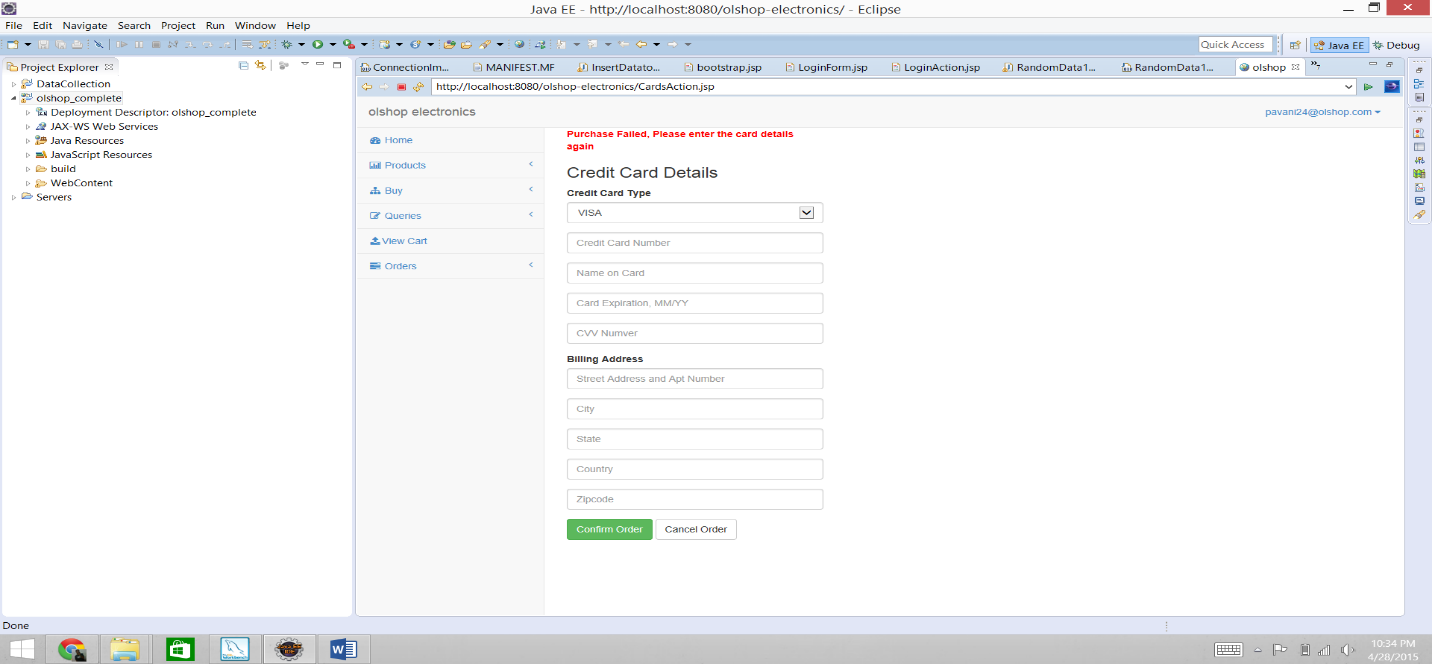


**5. CUSTOMER ORDER DETAILS**

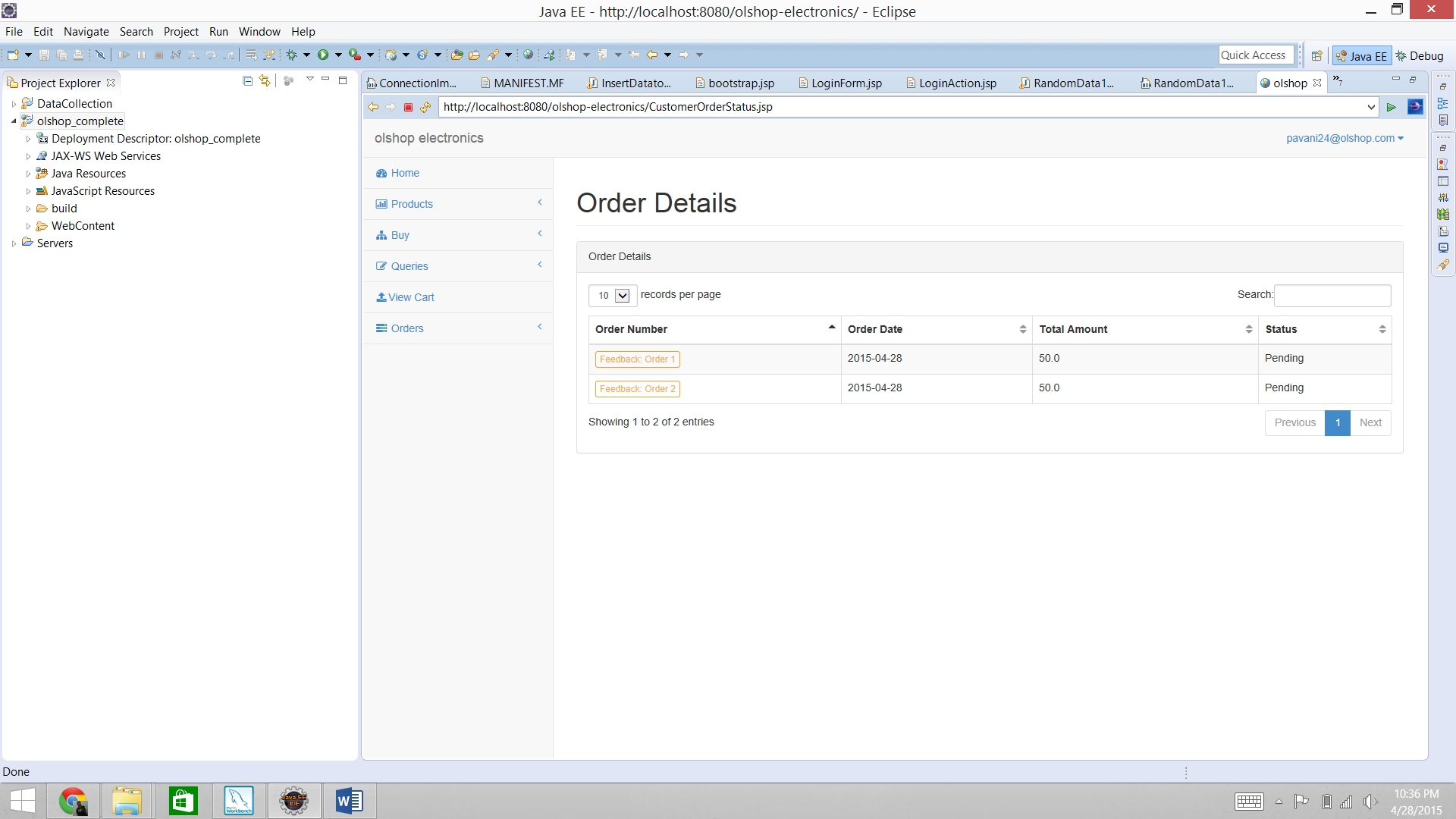


# 6. ORDER test Cases:

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Expected Result** | **Pass/Fail** | **Remarks** |
| 1 | “Purchase Failed “ | Pass | Working as expected. |

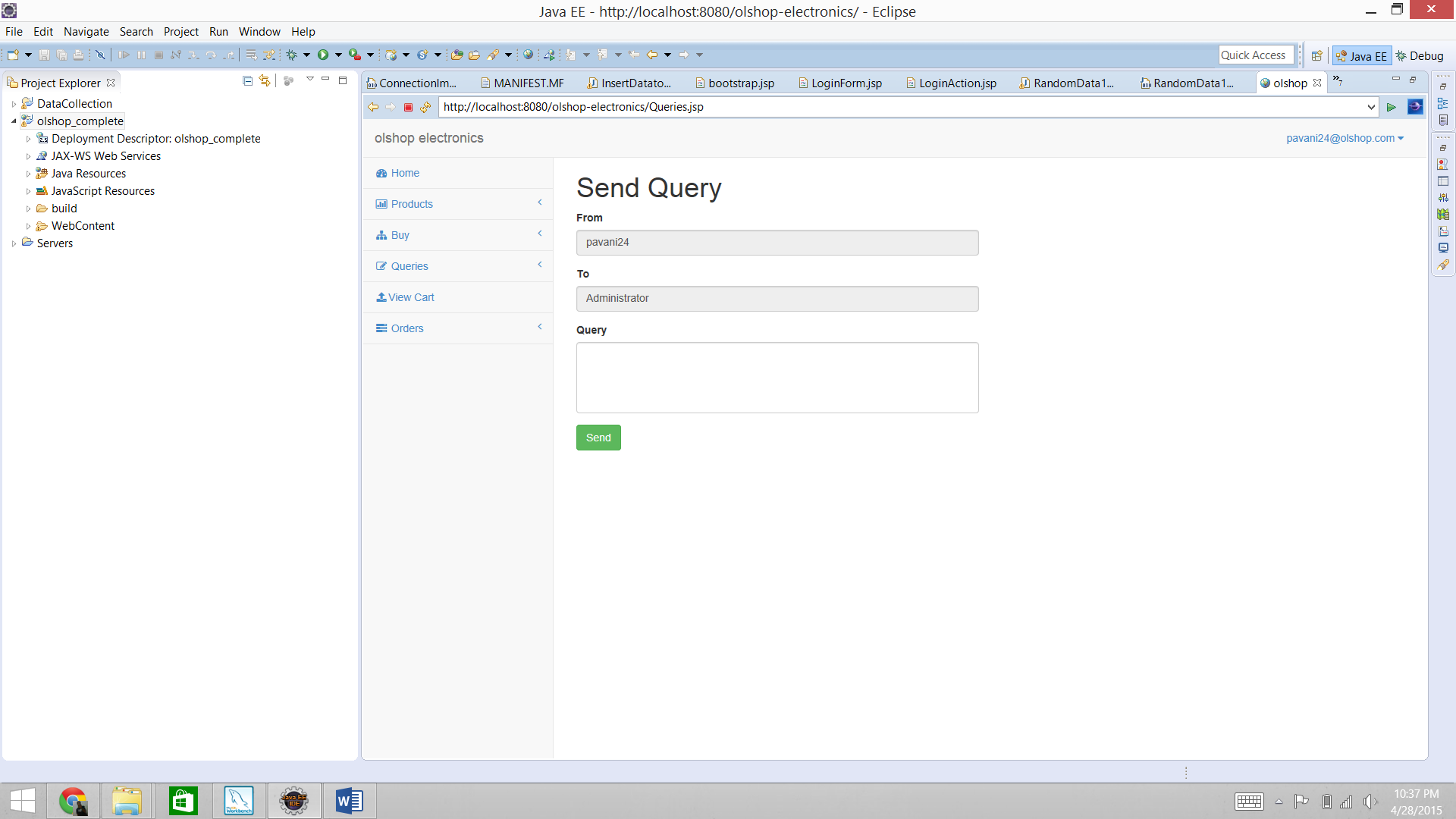


**7. ORDER DETAILS**



**8 QUERY TEST CASES:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Expected Result** | **Pass/Fail** | **Remarks** |
| 2 | “query” Submitted Successfully “ | Pass | Working as expected. |



**4.2 Future work:**

This project brings all the electronic goods under one roof which helps gadget lovers to shop more. We implemented many new functionalities to make the online shopping easier. We can add some more features in future for making the online shopping easier. Functionalities and features for this project definitely encounter the customer satisfaction. We have implemented “ the automatic email” feature to send an automatic email to the supplier in case of lack of item in the inventory, giving feedback about the product such as Good, Average and Best helps the admin to add more of that product. There is a lot of scope for this project In future.

**5 REFERENCES**

1. Query optimization: <http://beginner-sql-tutorial.com/sql-query-tuning.htm>

2. Database optimization: http://dev.mysql.com/doc/refman/5.0/en/optimizing-database-structure.html