GITHUB.COM/VISHALVERMA31

VE(Vex Electronics)

E-commerce Portal

Vishal Verma
3/13/2017

This is the documentation of the electronics E-commerce Portal. Electronic commerce, also known as ecommerce is a type of industry where buying and selling of a product is conducted over electronic systems such as the internet

Table of Contents

Table of Figures

CHAPTER-2

CHAPTER-4

CHAPTER-1

INTRODUCTION

E-commerce is fast gaining ground as an accepted and used business paradigm. More and more business houses are implementing web sites providing functionality for performing commercial transactions over the web. It is reasonable to say that the process of shopping on the web is becoming commonplace.

1.2 OBJECTIVE OF THE SYSTEM

The objective of this project is to develop a general purpose e-commerce store where electronic products like Laptops, Speakers, TV, can be bought from the comfort of home through the Internet. However, for implementation purposes, this project will deal with an online shopping for electronic projects.

An online store is a virtual store on the Internet where customers can browse the catalog and select products of interest. The selected items may be collected in a shopping cart. At checkout time, the items in the shopping cart will be presented as an order. At that time, more information will be needed to complete the transaction. Usually, the customer will be asked to fill or select a shipping address, and order details along with order number is shown to the customer as soon as the order is placed.

1.3 JUSTIFICATION

In today's market, it is extremely difficult to start a new small scale business and its sustenance with competition from the well-established and settled/brand owners. Most often, even if the quality of the product is really good, due to a lack of advertisement or business at the small scale, it just becomes another face in the sea, and the product does not reach a larger group of customers. In fast paced life of today when everyone is squeezed for time, the majority of people are finicky when it comes to doing physical shopping. Logistically, a consumer finds a product more interesting and attractive when they find it on the website of a retailer directly and are able to see item's details online. The customers of today are not only attracted because online shopping is very convenient, but also because they have broader selections, highly competitive prices, better information about the product and extremely simplified navigation for searching regarding the product.

1.4 OVERALL DESCRIPTION

1.4.1 Description:

- Any visitor can register and view available products.
- Only registered member can purchase multiple products regardless of quantity.
- AboutUs page is available to show information about company.
- ContactUs page is available to show contact information of the company.
- There are three roles available: Visitor, User and Admin.
    - Visitor can view available products.
    - User can view and purchase products.
    - An Admin has some extra privilege including all privilege of visitor and user.
        - ✓ Only Admin can access the Admin Page.
        - ✓ Admin can add user, edit user information and can remove user.
        - ✓ Admin can add products, edit product information and add/remove product.

1.4.2 Using the code:

1. Import Project folder in eclipse.
2. Configure Apache tomcat server.
3. Configure H2 console.
4. Run the Project on the server.

1.4.3 Web Pages Details:

- Home Page
- Product Page
- Product Details Page
- AboutUs Page
- ContactUs Page
- Admin Page
- Cart Page
- Shipping Address Page
- Order Confirmation Page
- Thank you Page

CHAPTER-2

DESIGN OF THE SYSTEM

2.1 System Requirements (Hardware &Software)

2.1.1 Hardware:

This web application shall provide minimum hardware requirements. The following hardware configurations are required for a PC using the online shopping-cart application:

- Pentium processor
- 32 MB of free hard-drive space
- 128 MB of RAM

2.1.2 Software:

This section lists the requirements that are needed to run the system efficiently. The operating system needed for the system to run effectively, the interface to run the application, the driver for running Java web applications, the integrated development environment to develop the application, and the third-party tool used for editing purposes are as follows:

- Operating System: Windows (Vista/Windows 7,8,10) or MAC OS 2.
- Web Brower: Internet Explorer, Mozilla Firefox, or Google Chrome
- Drivers: Java Runtime Environment
- Integrated Development Environment: Eclipse, Apache Tomcat

2.2 USE CASE DIAGRAM

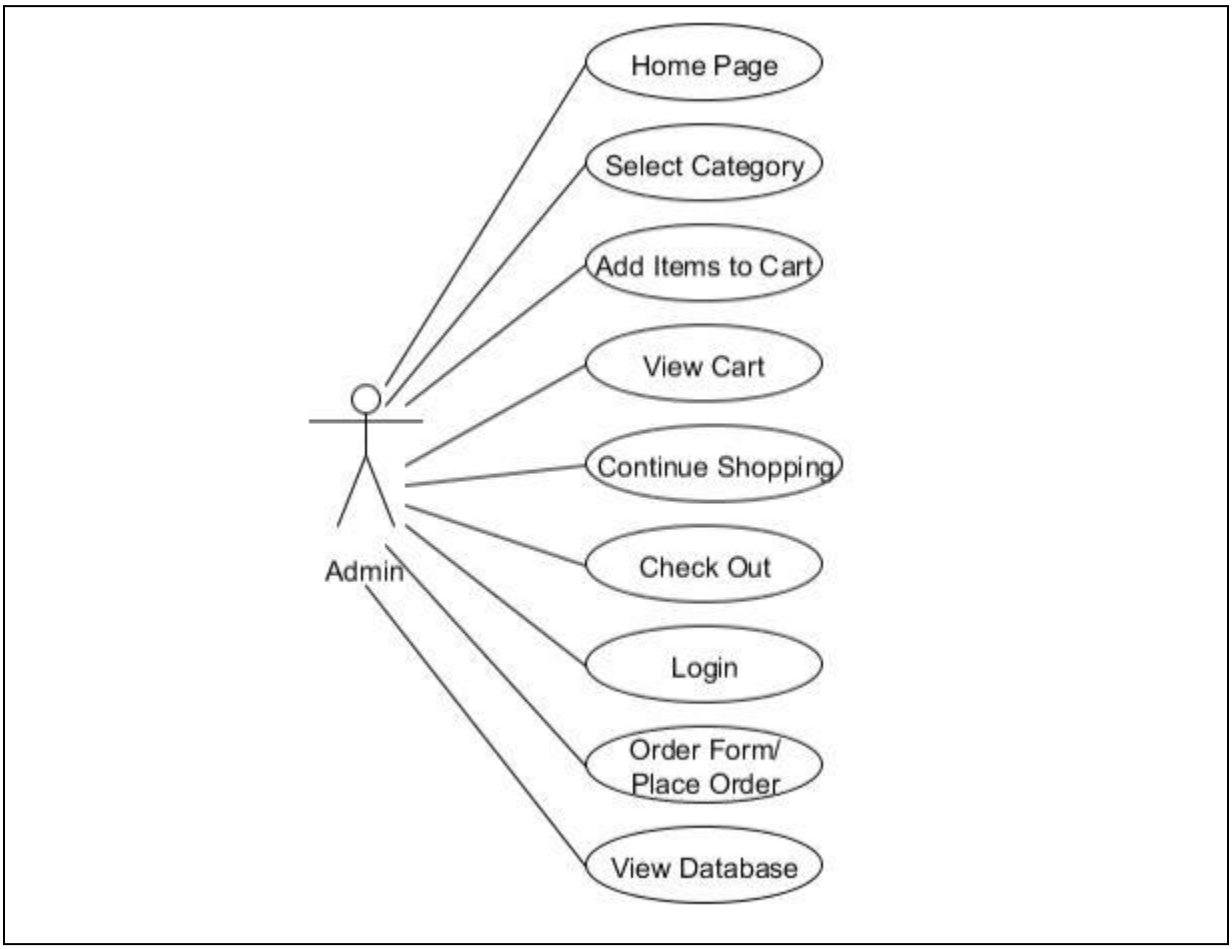

Fig. 2.1: Use Case Diagram for the Admin

Admin: The administrator is the owner of this application. One must have a basic understanding of computers and the internet as well as prior knowledge for operating the eclipse and Java programming languages. The administrator is responsible for maintaining all the training documents required for the system. The administrator can perform the following functions:

- Admin can add user, edit user information and can remove user.
- Admin can add products, edit product information and add/remove product.

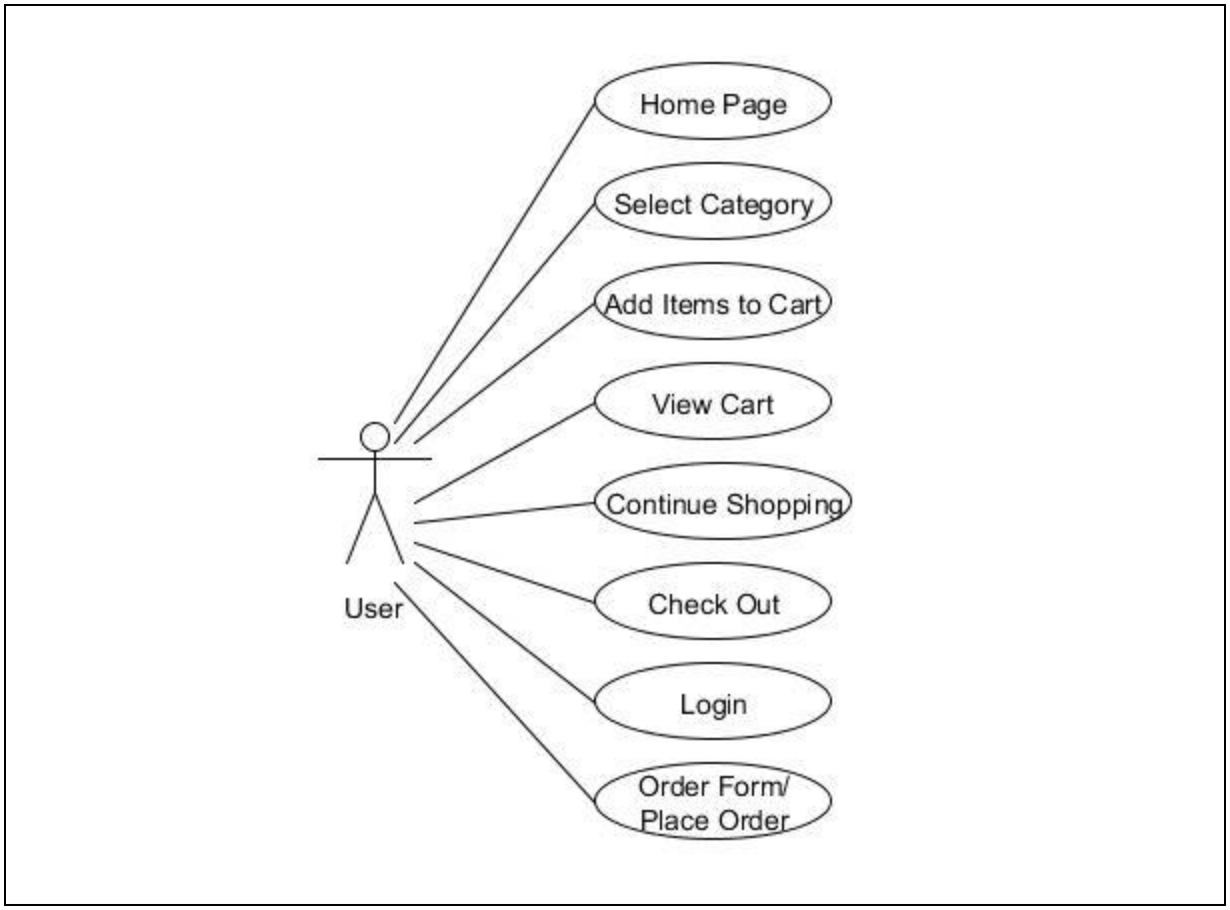

Fig. 2.2: Use Case Diagram for the User

Users: The users of this application are all customers who would shop to test the application. These users are anyone with shopping experience and the know-how to browse through a shopping-cart application. They must have basic understandings about computers and the internet. The users should be able to perform the following functions using this system:

- View, browse, and select a category on the home page.
- View, add, and update items in the cart.
- Delete items from the cart.
- Check out the items from the application or continue shopping.
- Sign-on/login using a username and password.
- Place the order by completing the order form

2.3 SOFTWARE ARCHITECTURE

The diagram below shows the Software's broad architecture.

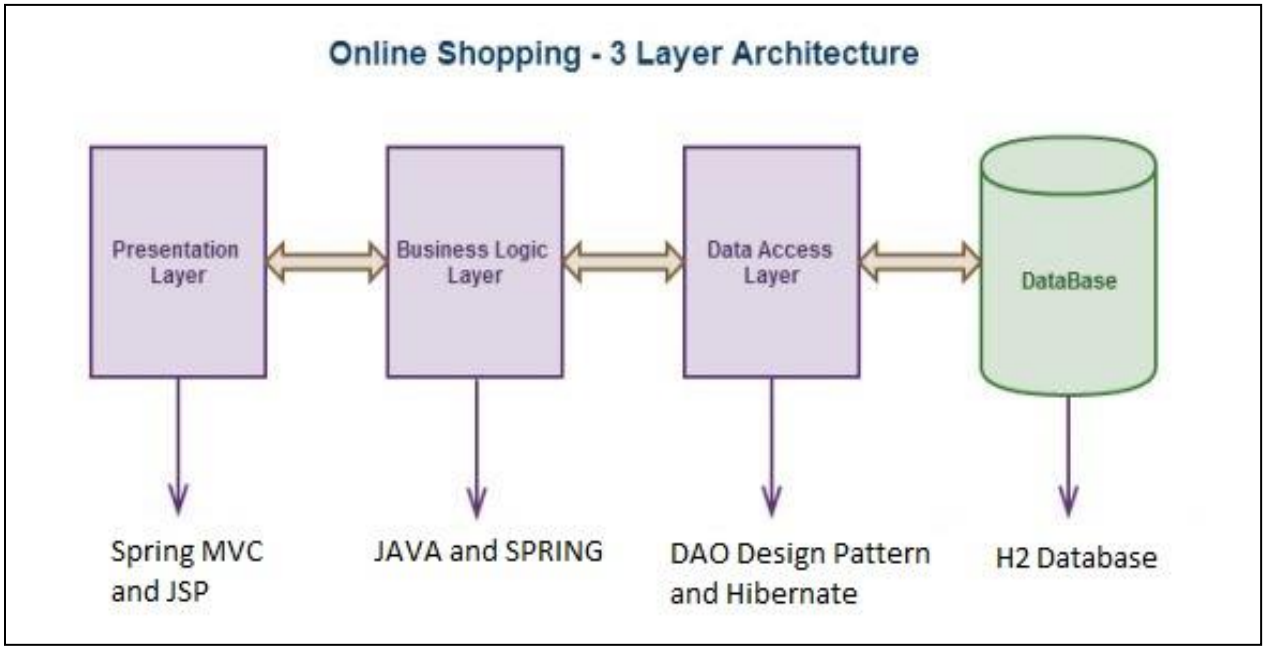

Fig. 2.3: Three Layer Architecture of Web Application

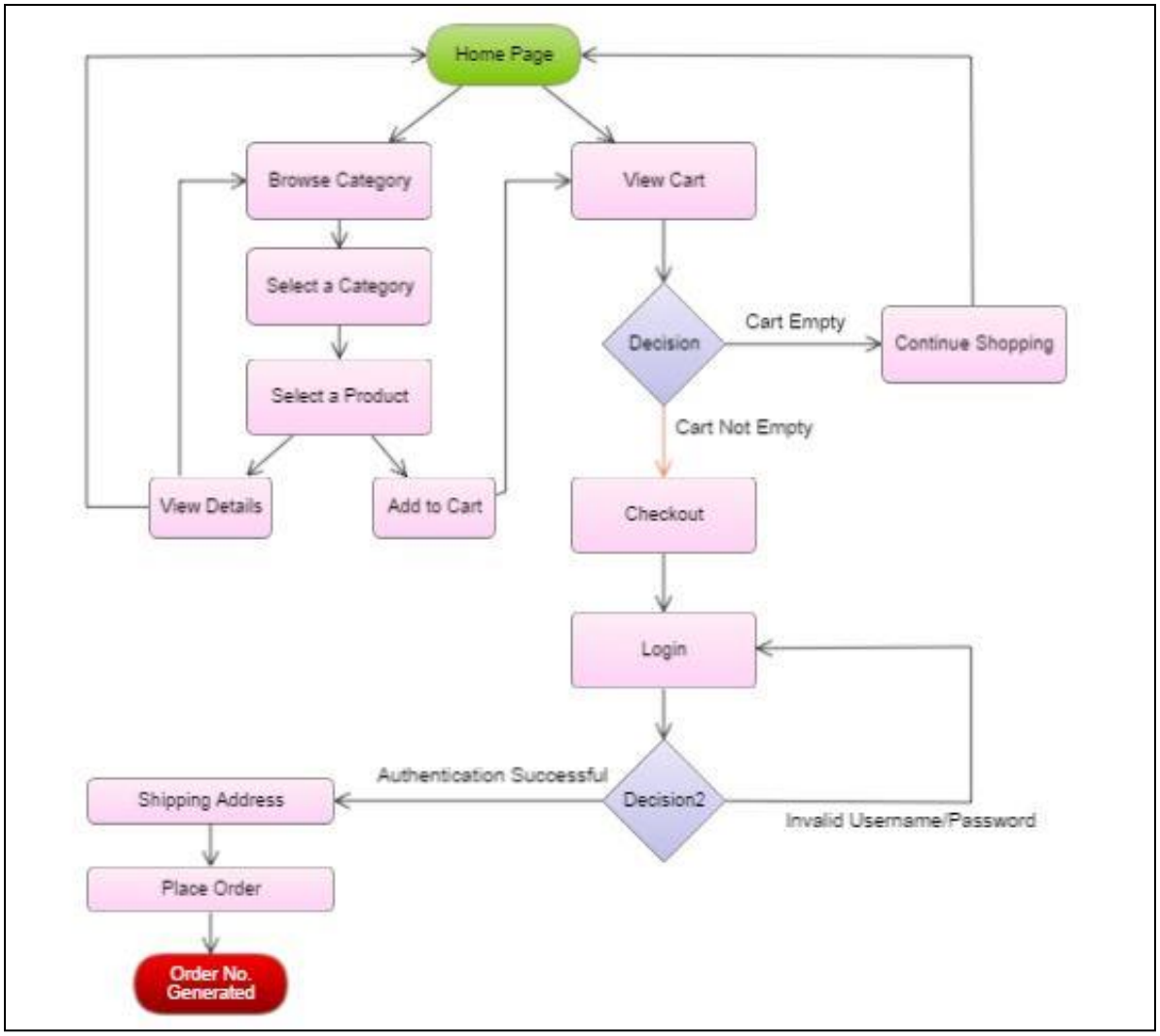

Fig. 2.4: Activity Diagram for Web Application

CHAPTER-3

IMPLEMENTATION

This project is divided into two part: ve as in the frontend of the project and veBackend as in the Backend of the project. The veBackend is used for database configuration. Here, DAO Design Pattern is used to do CRUD operation on the Database. H2 Database is used to store data in Database for this project. JSP, JavaScript, AngularJS and CSS is used in the frontend (ve) and JSON is used for Data interchange in this project. Restful Service is used on the cart page to increase/decrease the quantity of the item in cart, to refresh the cart and delete items from cart.

3.1 To implement Spring MVC

The Spring web MVC framework provides model-view-controller architecture and ready components that can be used to develop flexible and loosely coupled web applications. The MVC pattern results in separating the different aspects of the application inputlogic, businesslogic,and UIlogic, while providing a loose coupling between these elements.

- The **Model** encapsulates the application data and in general they will consist of POJO.
- The **View** is responsible for rendering the model data and in general it generates HTML output that the client's browser can interpret.
- The **Controller** is responsible for processing user requests and building appropriate model and passes it to the view for rendering.

In ve (Frontend),

- **Dependencies:** Spring-web, Spring-webmvc
- **Configuration:** WebConfig.java, WebAppIntializer.java
- **Controllers added in com.ve.controller package.**

3.2 To configure Database

JDBC stands for **Java Database Connectivity**. It provides a set of Java API for accessing the relational databases from Java program. These Java APIs

enables Java programs to execute SQL statements and interact with any SQL compliant database.

ORM stands for **Object-Relational Mapping (ORM)** is a programming technique for converting data between relational databases and object oriented programming languages such as Java, C#, etc.

Hibernate is an Object-Relational Mapping(ORM) solution for JAVA. It is an open source persistent framework created by Gavin King in 2001. It is a powerful, high performance Object-Relational Persistence and Query service for any Java Application.

Hibernate maps Java classes to database tables and from Java data types to SQL data types and relieves the developer from 95% of common data persistence related programming tasks.

In veBackend (Backend),

- **Dependencies:** Hibernate-core, H2, Commons-dbcp, Spring-orm
- **Configuration:** AppContext.java

In ve (Frontend),

- **Dependencies:** jstl, javax.servlet-api, veBackend
- **Plugins:** maven compiler, maven war

3.3 To implement Validations

In veBackend (Backend),

- **Dependencies:** validation-api, hibernate-validator

3.4 To implement Image Upload

In veBackend (Backend),

- **Dependencies:** commons-io, commons-fileupload

In ve (Frontend),

- **MultipartResolver added in WebConfig.java**

3.5 To implement Spring Security

In ve (Frontend),

- **Dependencies:** Spring-security-web, Spring-security-config
- **Configuration:** SecurityConfig.java, SecurityWebAppIntializer.java
- **Few changes in WebConfig.java**

3.6 To implement Spring Webflow

In ve (Frontend),

- **Dependencies:** Spring-webflow
- **Configuration:** WebflowConfig.java
- **Demo-flow.xml added in folder flow along with other jsp files.**

CHAPTER-4

SCREENSHOTS

4.1 Output screen

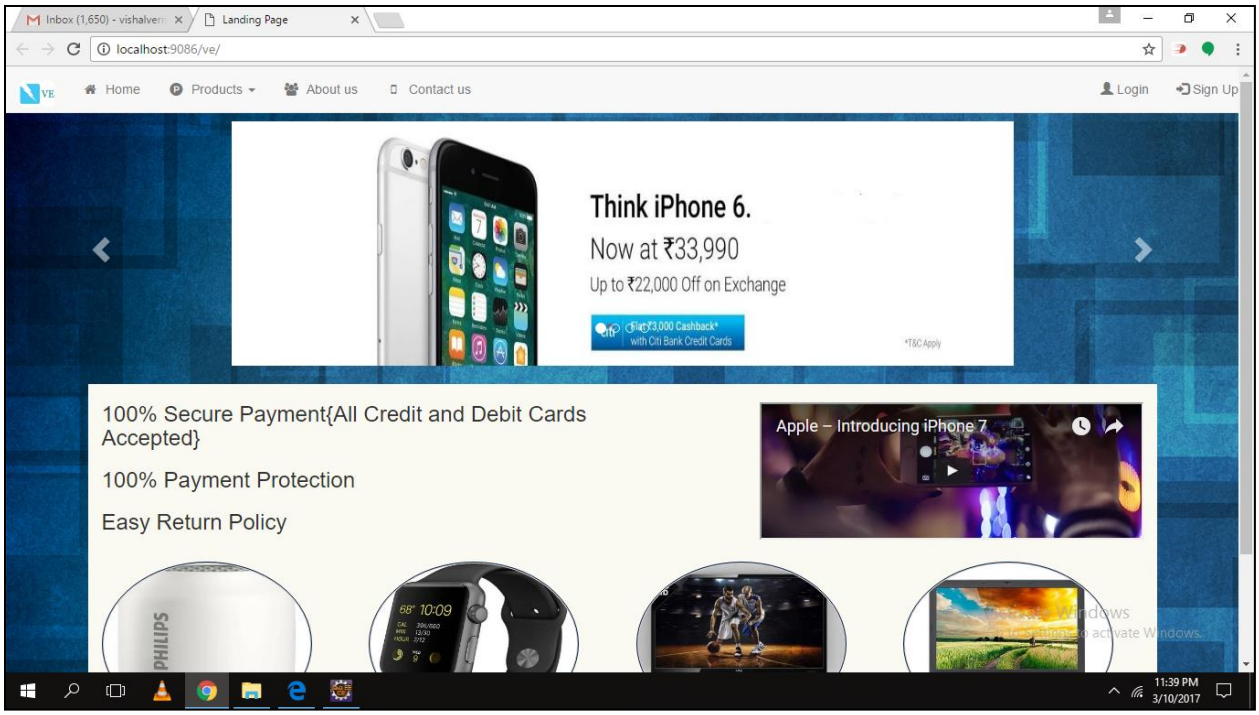

Fig. 4.1: Home Page

Home Page: The home page of the application (Figure 5.1) is common to all the visitors/users/administrators. At the time of logging into the home page, the page shows the navigation bar with a dropdown menu on Products, the carousel with images of products available for shopping. In the middle section, it shows an embedded youtube video and below that there are images of category of products.

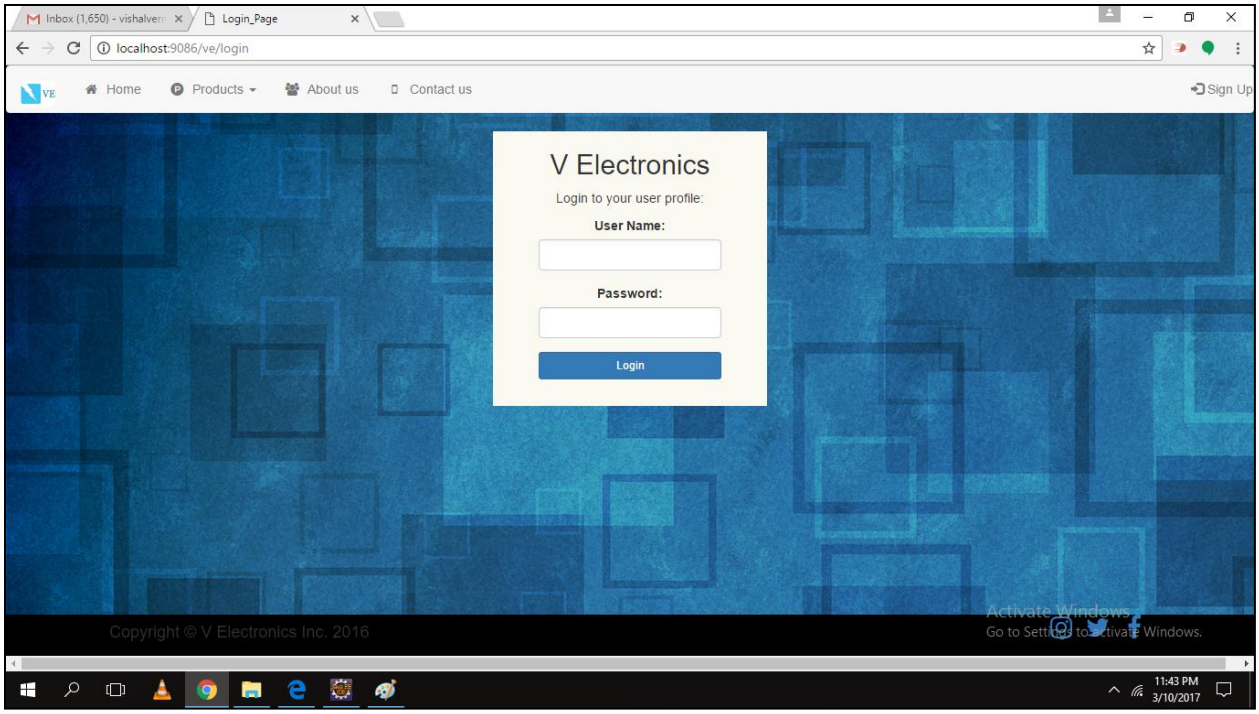

Fig. 4.2: Login Page

This is the login page users will see when they'll click on Login in navigation bar. This login page acts as a security feature for the application so that unauthorised access can be prevented.

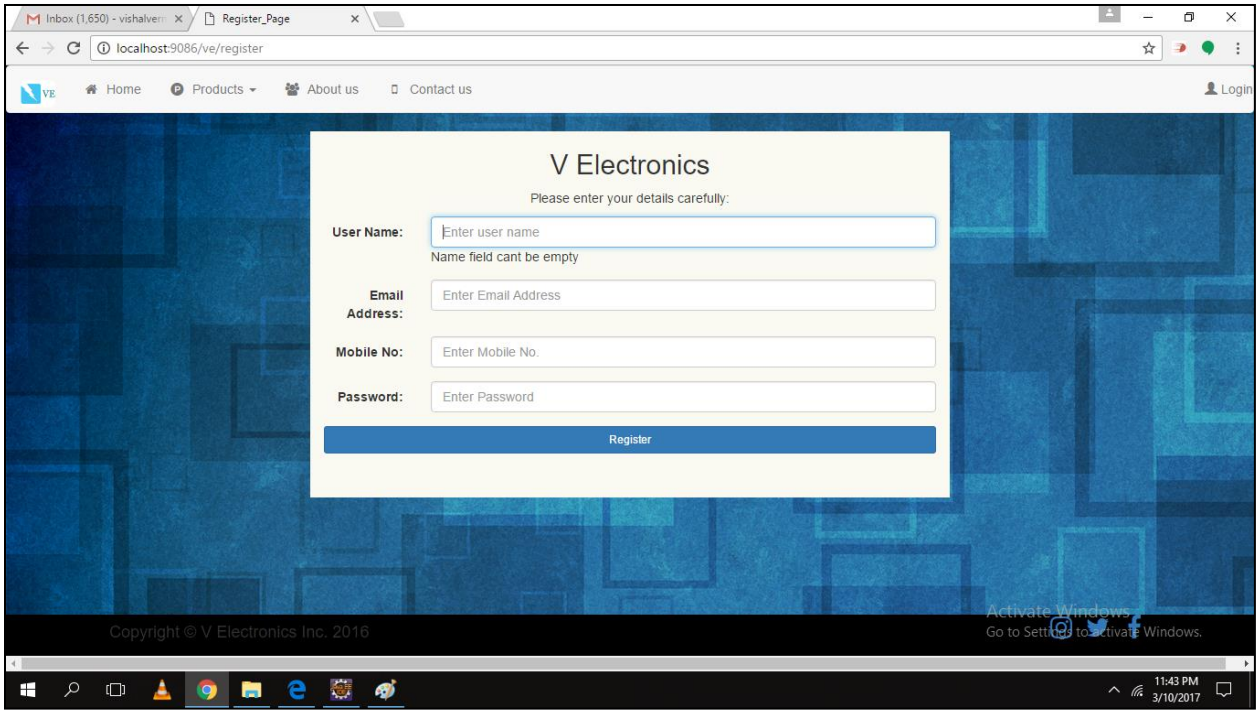

Fig. 4.3: Register Page

First time users need to sign up in order to use the Web Application.

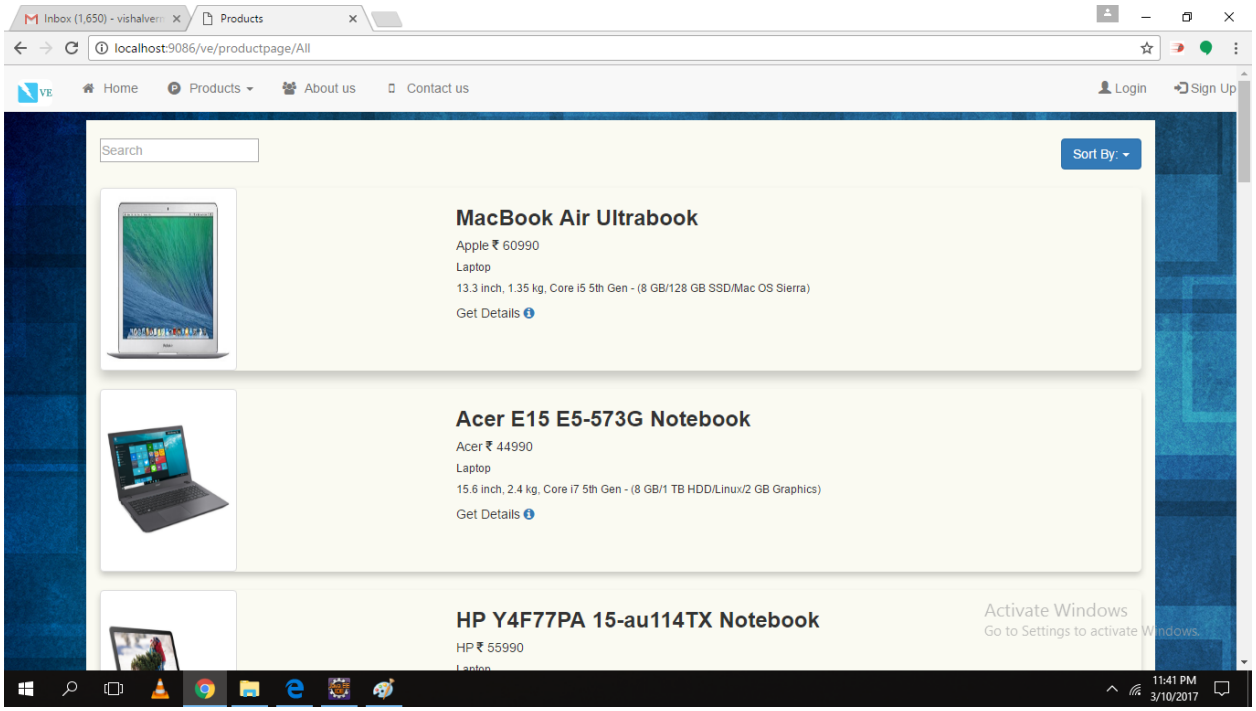

Fig. 4.4: Product Page

User can see all the products as well as products on the basis of category. The user can search for a specific product and can sort list of the products by using Sort By dropdown.

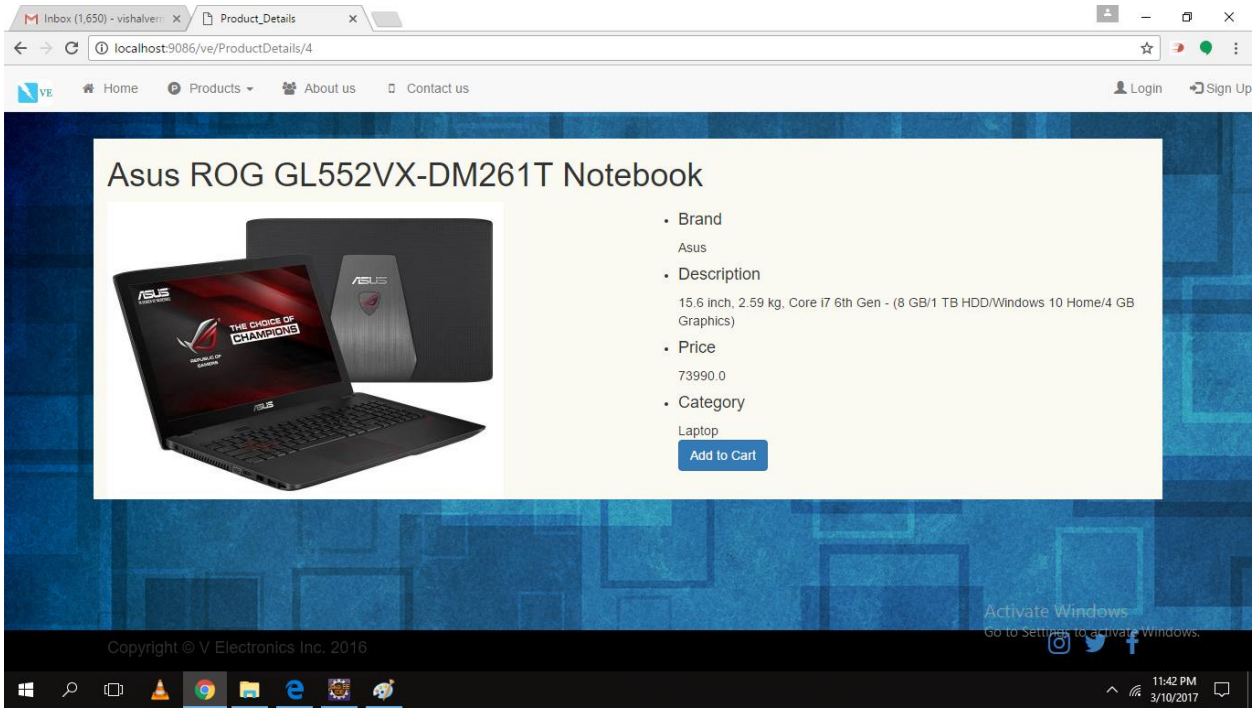

Fig. 4.5: Product Details Page

This is the product details page which tells user every detail of the product, user showed interest in from the product page. User can also add product inside the cart from here.

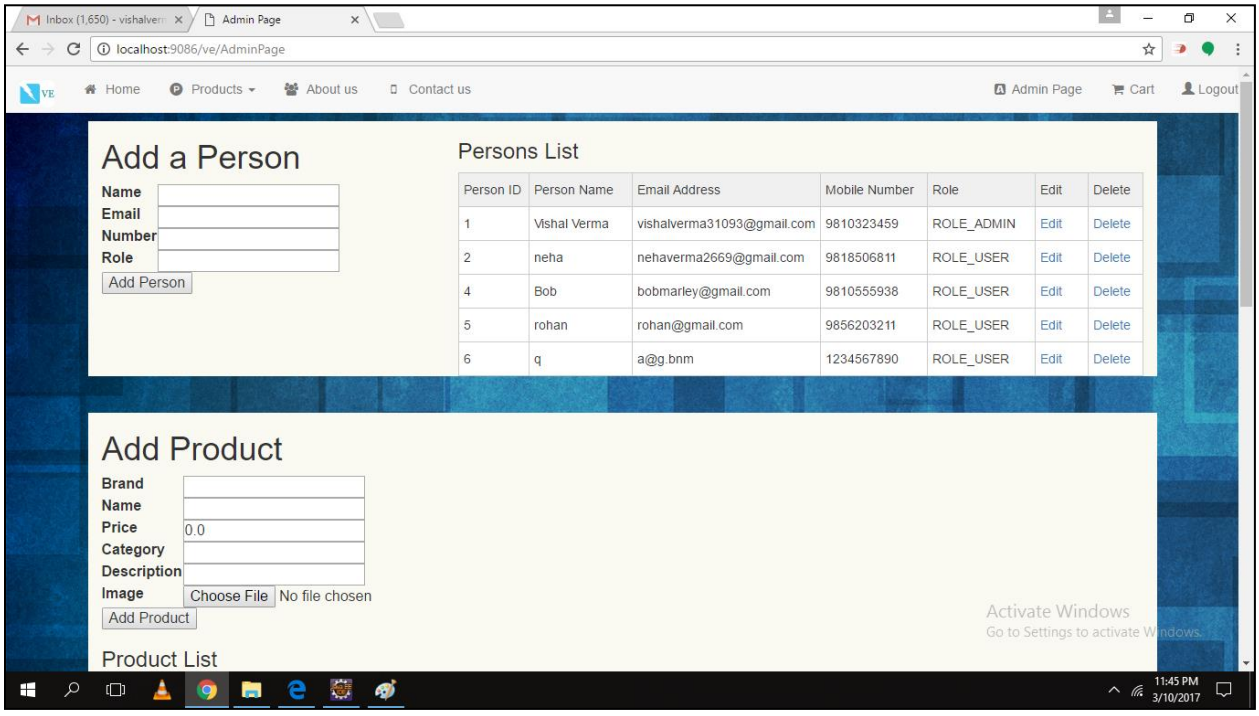

Fig. 4.6: Admin Page(List of Users)

This is Admin Page, from where Admin can add new user, edit the information about the user, delete the user and view list of all the users.

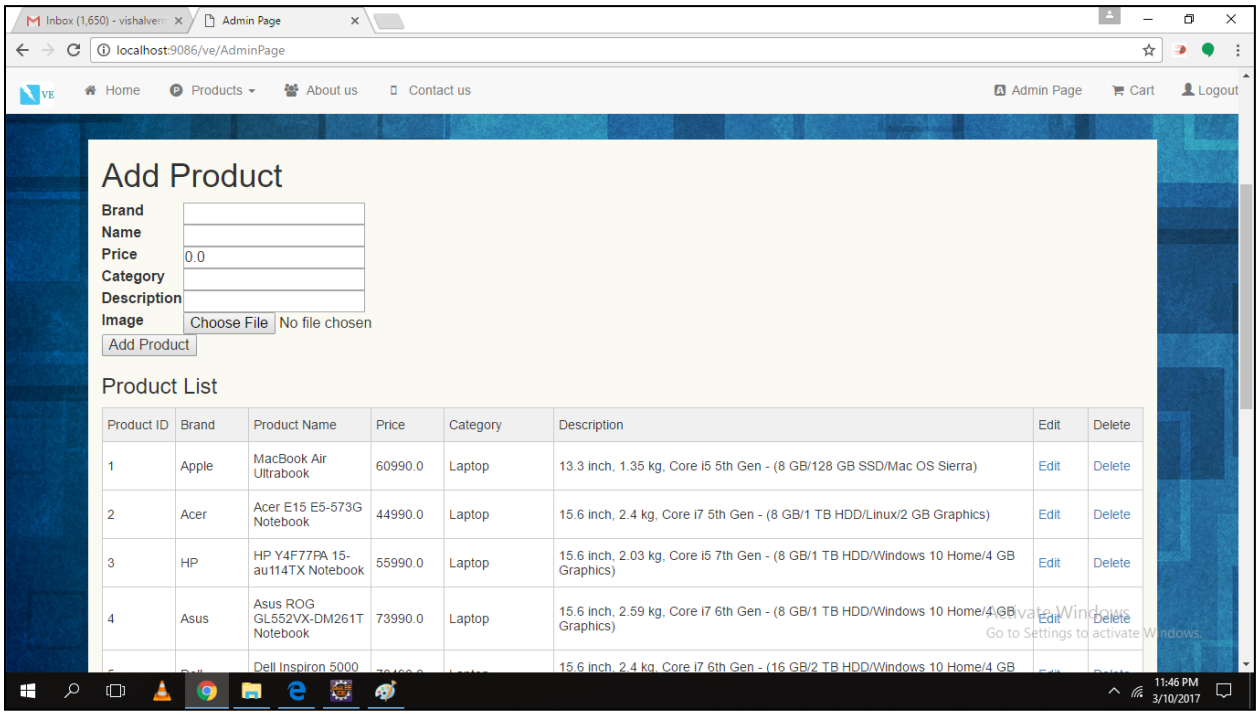

Fig. 4.7: Admin Page

This is Admin Page, from where Admin can add new product with image of the product, edit the information about the product, delete the product and view list of all the products.

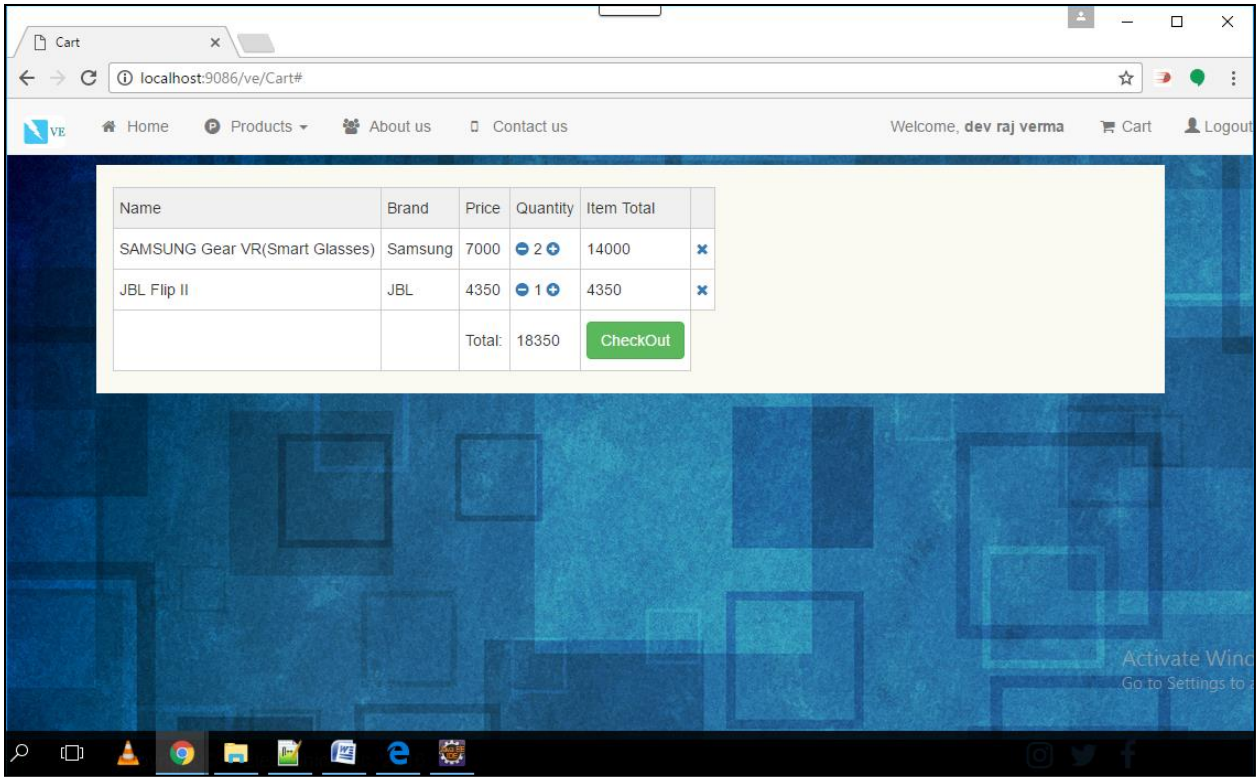

Fig. 4.8: Cart Page

Finally when user clicks on ‘add to cart’ in the product details page, he gets to see his Cart Page as shown here. User can increase as well as decrease the quantity of the product by clicking upon plus and minus icons. User can also remove the item from the cart by clicking upon the cross icon.

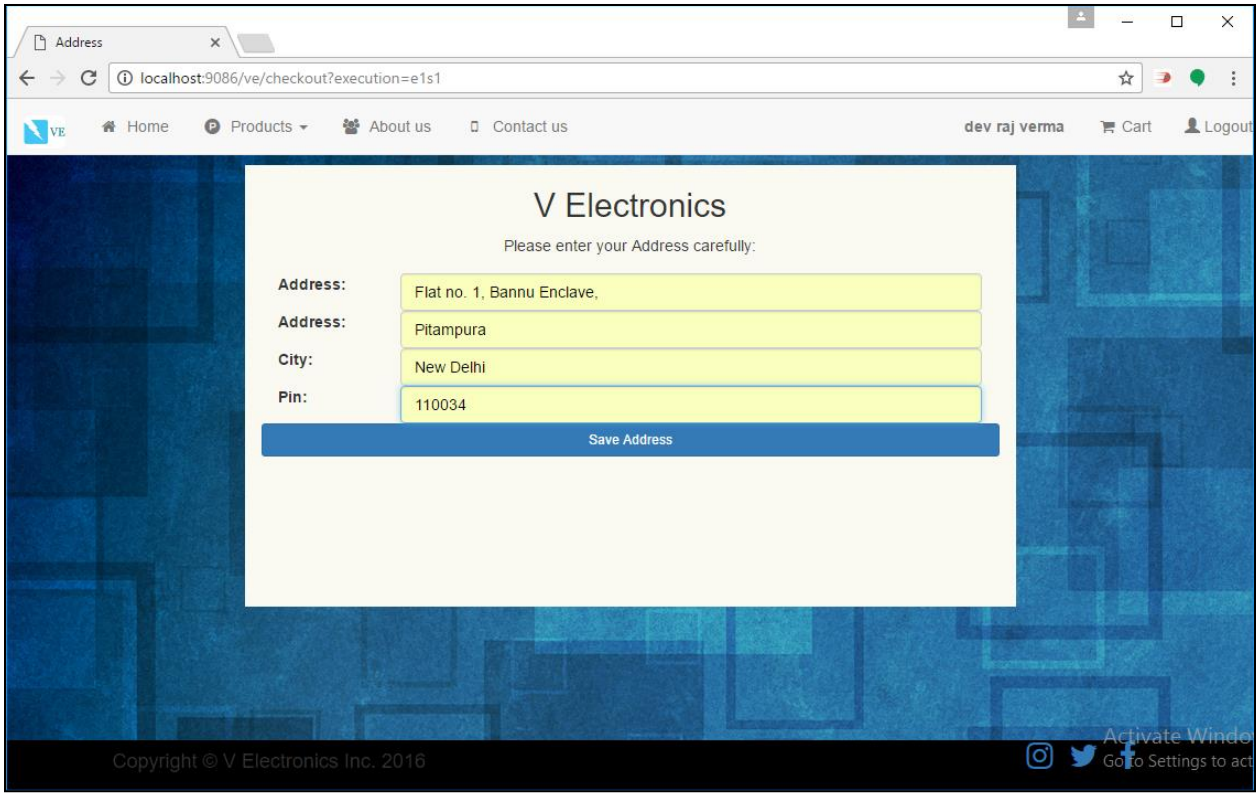

Fig. 4.9: Shipping Address Page

Shipping Address of the user is taken here as shown as above.

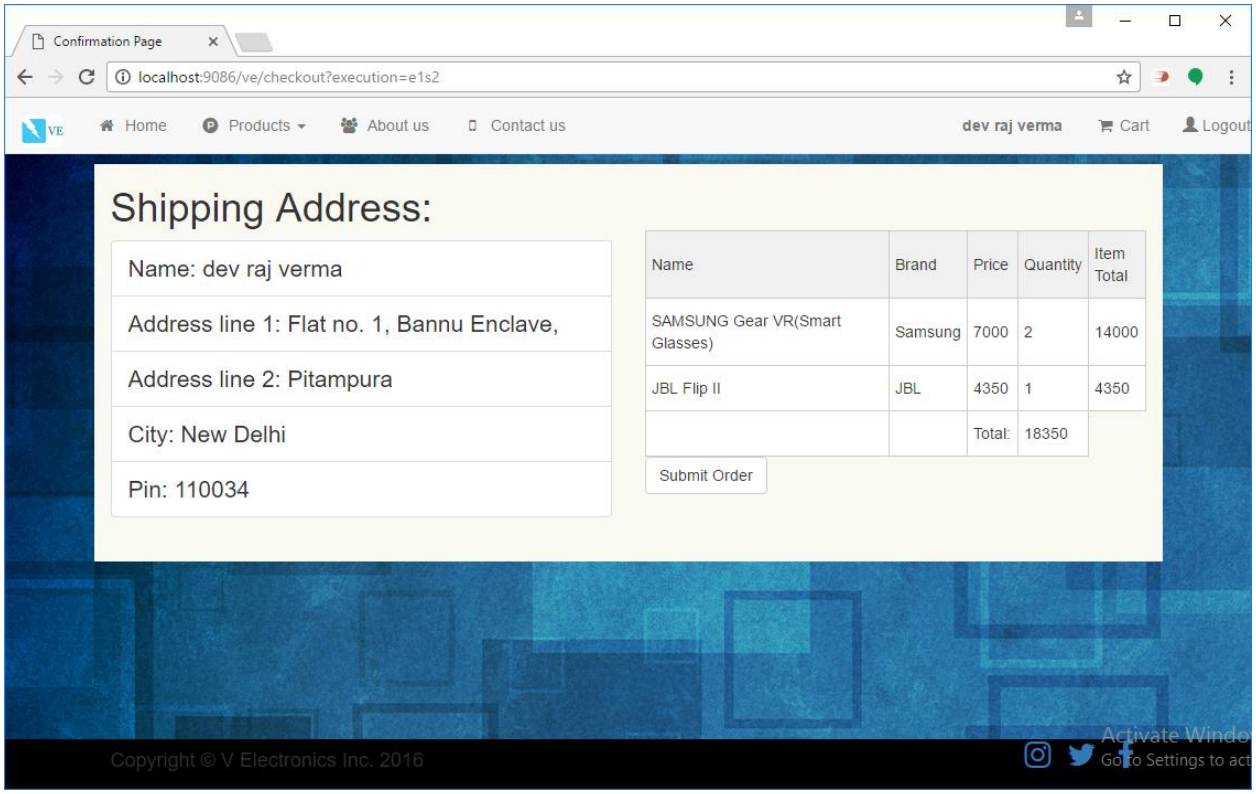

Fig. 4.10: Order Confirmation Page

On this page, the details of user as well as the order is shown and user can confirm his/her order by clicking on Submit Order Button.

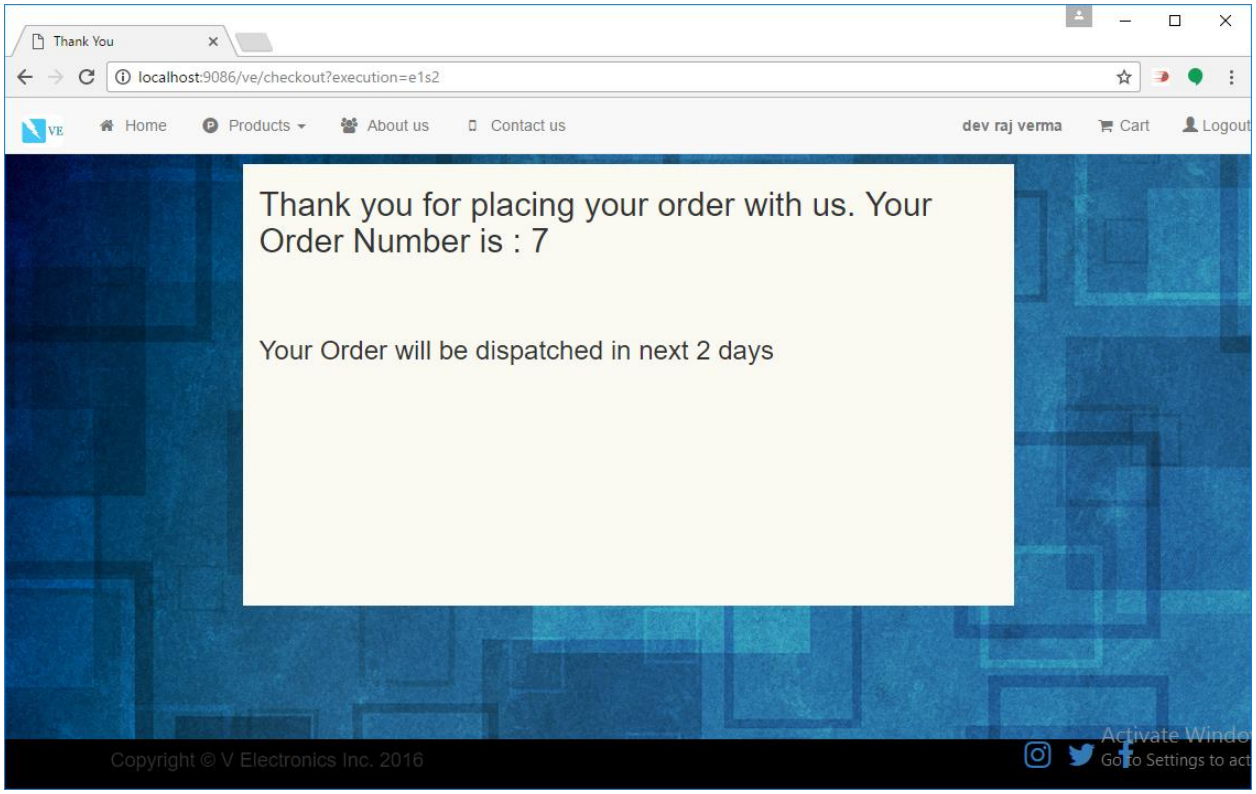

Fig. 4.11: Thank You Page

The Order Number is shown to the user as shown above.

CHAPTER-5

CONCLUSION

5.1 CONCLUSION

This project helps in understanding the creation of an interactive web page and the technologies used to implement it. The different types of technologies used while making this Web Application are HTML5, BOOTSTRAP, Maven, Spring MVC, Spring Security, Spring WebFlow, Spring ORM, Hibernate, H2 database, JQuery, JSON and AngularJS.

A good shopping cart design has been accompanied with user-friendly shopping cart application logic. It should be convenient for the customer to view the contents of their cart and to be able to remove or add items to their cart. The shopping cart described in this project provides a number of features that are designed to make the customer more comfortable.

5.2 FUTURE SCOPE

The following section discusses the work that will be implemented with future releases of the software.

- **Detailed categories**: Future work could involve adding more categories which are more detailed and have additional items.
- **Watch/Wish List**: Work can add a watch list or wish list so that users can add an item to a list to watch for item prices to go down or to see when there is a sale on any of those items.
- **Enhanced User Interface**: Work on enhancing the user interface by adding more userinteractive features.
- **Recommended Items**: Add a bar that would display the most-recommended items which would depend on the number of times an item has been purchased by any users.
- **Payment Options**: Add different payment options, such as Visa, MasterCard, PayPal, etc., where a user can also save the card information for later checkouts.

- **Shipping Options**: Add different types of shipping options: regular shipping, expedited shipping, international shipping, etc.
- **Recent History**: Display the user's recently browsed items in the recent-history tab.

REFERENCES

http://stackoverflow.com/

https://google.co.in/

https://www.w3schools.com/