**End Semester Project**

Web and Android Technology

IT-604

**Satadru Banerjee**

**BTech 6th Sem**

**Roll: T91/IT/174111**

**Reg: DO1-1111-0167-17**

**Table of Contents:**

* Problem Statement
* Problem Analysis
* Flowchart
* Class Diagram
* Table Structure
* Input/Output
* Code Repository
* Conclusion

**Problem Statement:**

Develop a Web based client-server application for maintaining the inventory management. It accepts item detail and issue the same by checking all validations. System should have authentication.

Specific instructions:

* Follow the MVC architecture
* Define session for handling idle time.
* Design database / table with triggers
* Design web interface responsive
* Use the language / tools: Java, HTML, CSS, JavaScript, MySQL, and others.
* You may add more features

**Problem Analysis:**

Here, the inventory management system is the cart feature of a mock e-commerce platform. The user can add products to his cart and remove them if needed. Here, the user will be able to proceed to a final checkout page and post a mock payment, be redirected to the homepage. The model continuously fetches data about the products, the user’s cart as well as the user’s details and upgrades the view which is our HTML interfaces. The controllers are connected to the database tables and accordingly create, update and delete the table entries. For now, products and users are hardcoded into the database.

The solution can be subdivided into the following major steps:

* An HTML based frontend which serves as the primary user interface. The homepage is a pure HTML, CSS, JS file. Since the approach chosen is of the inventory management system of an E-commerce site, the homepage describes our products and related features.
* On clicking a particular project, the user is taken to the product details page, which is based on java servlets and fetches the product related data from a MySQL database.
* The User can select add to cart option, And will be asked to login. Post this, product and its relevant details as well as the user’s unique id will be pushed to the cart table.
* On clicking the cart icon, all entries in the cart table are shown.
* The user can click proceed to checkout. Based on his ID, the user is identified from the table of users and his delivery address and other details are displayed, with an option to edit.
* On selecting an appropriate payment option, the user can proceed to payment. Post this, the cart will be emptied and the order is placed. User is redirected to the homepage

**User Flow:**

Homepage(HTML)

Product page (Servlet)

Buy Now/Add to Cart

Authenticated user?

Cart details

Checkout

Check delivery details and pay

Payment Success?

**Class Diagram:**

addCart(Cart c), editCart(Cart c), deleteCart(Cart c)

DBConnector

String: username, password, url

MySQL DB

getUser(User u), getProduct(Product p), getCart (Cart c), getHistory(User u)

List<Cart>

getCart(User u)

ShowCart

List<Cart> cartItems

User

Int: uid

String: username, userpwd, userAddr

User

getUser(int uid)

Product

Int: pid, pprice, pcategory,pdelTime

String: pname, pcategory, pdesc, pimg

ShowUser

User u1

getDetails(Product p)

ShowProduct

Product p

showUser(User u)

Product

showCart(User u)

addtoCart(Product p)

Authenticate(User u)

Cart

Int: pid,uid,qty,price

Authentication

User: u

makePayment(User u, Cart c)

ShowMessage

Int: mcode

String: message

Checkout

User: u

Cart: c

paymentSuccess(int uid)

**Table Structure:**

Products

\*Int(8) prod\_id

Varchar(30) Category

Varchar(50) pname

int(3) qty

int (2) delivery

Varchar(500) description

Varchar (100) image

Users

\*Int(5) user\_id

Varchar(30) username

Varchar(50) fname

Varchar(256) userpwd

Varchar (300) userAddr

Purchase-History

\*Int(5) user\_id

\*Int(8) prod\_id

Varchar(50) pname

Int(2) qty

Int(7) price

Varchar(10) purchaseDate

Cart

\*Int(5) user\_id

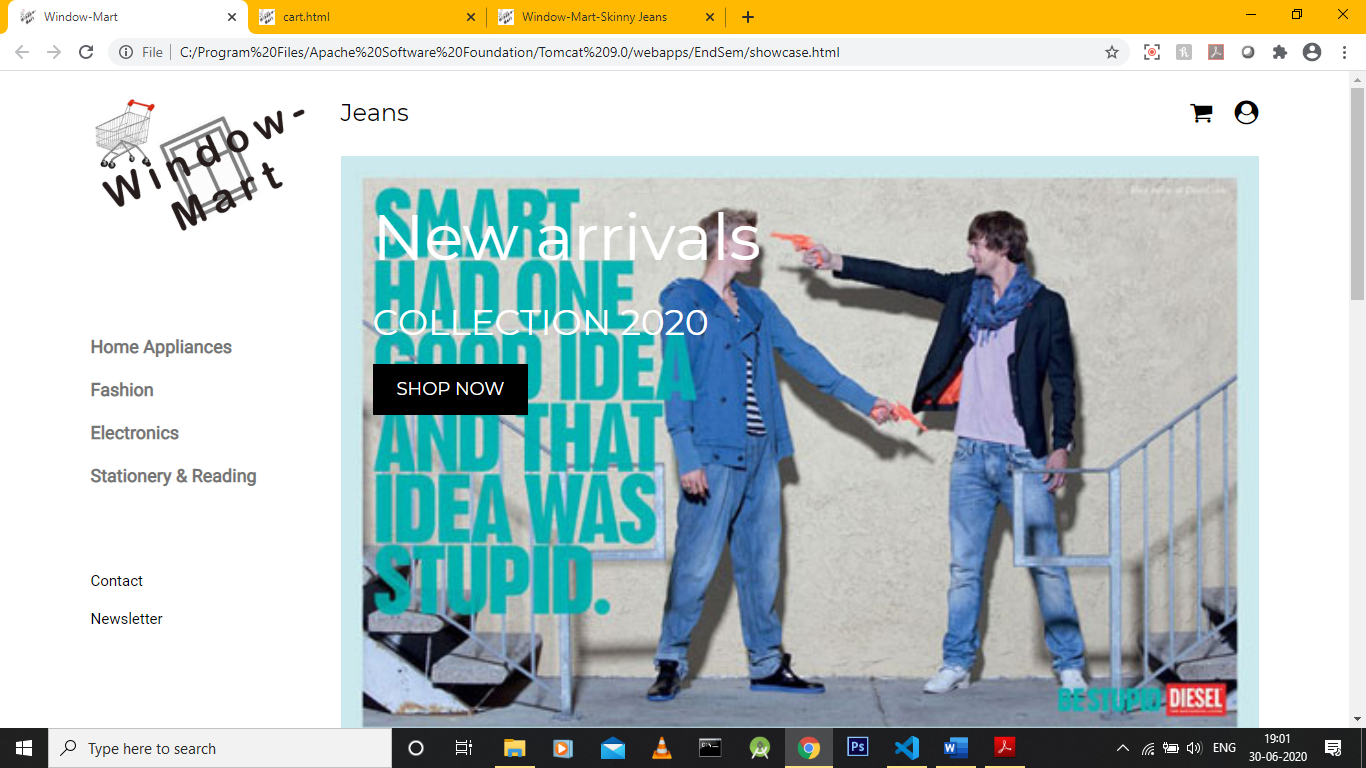
\*Int(8) prod\_id

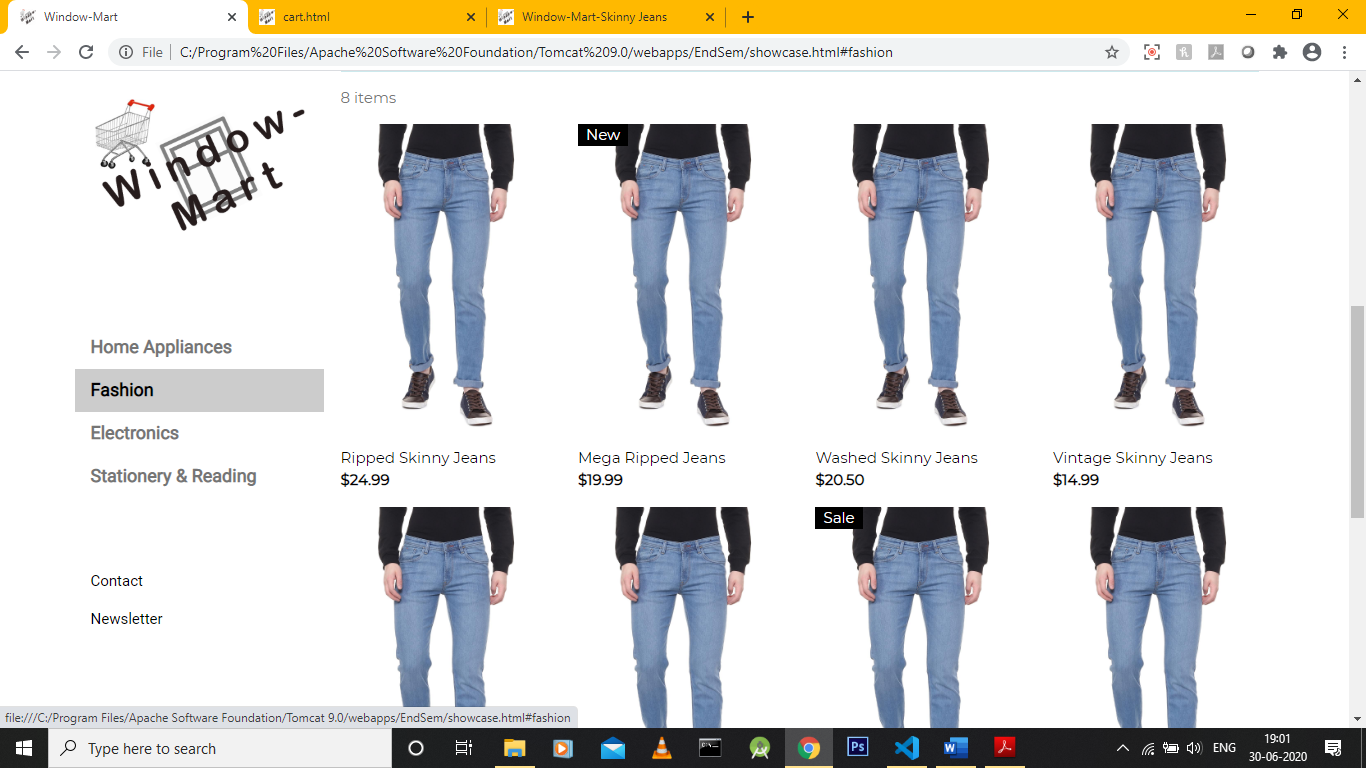
Varchar(50) pname

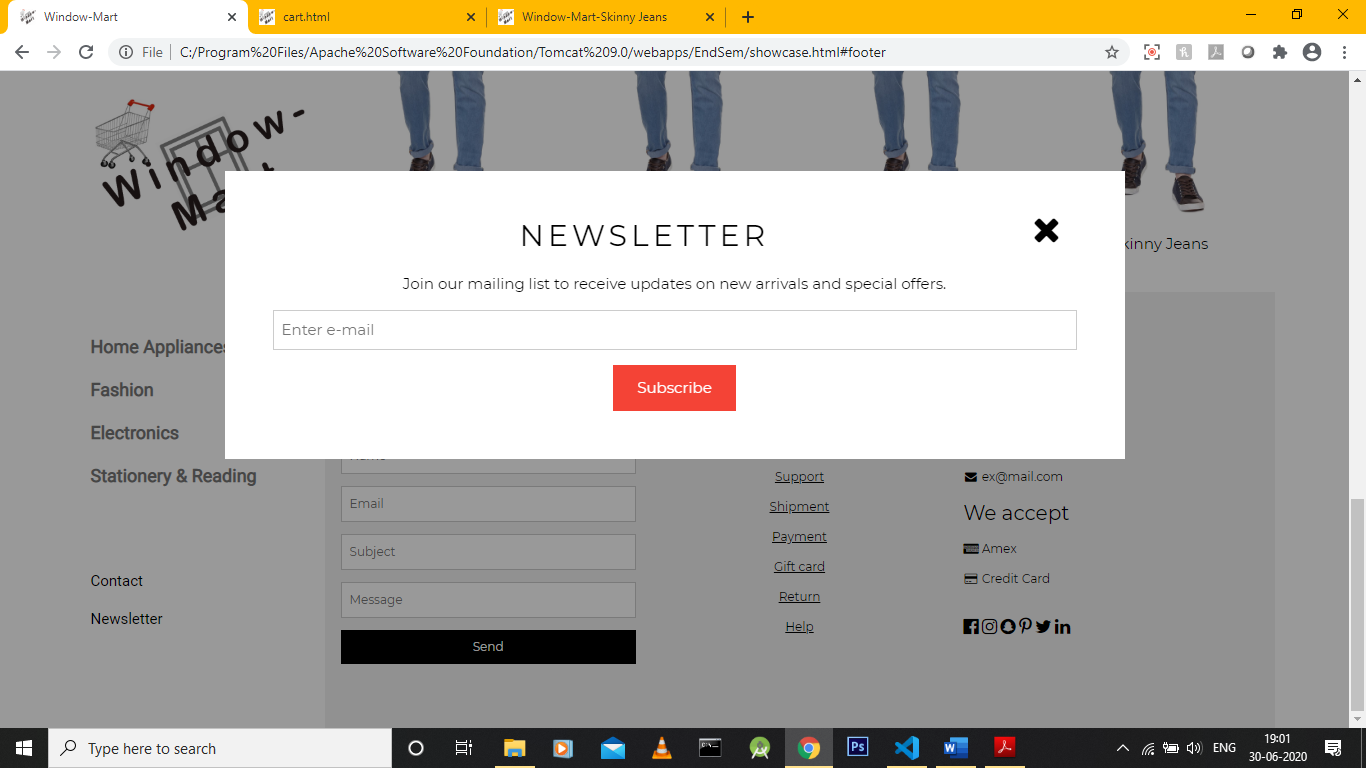
Int(2) qty

Int(7) price

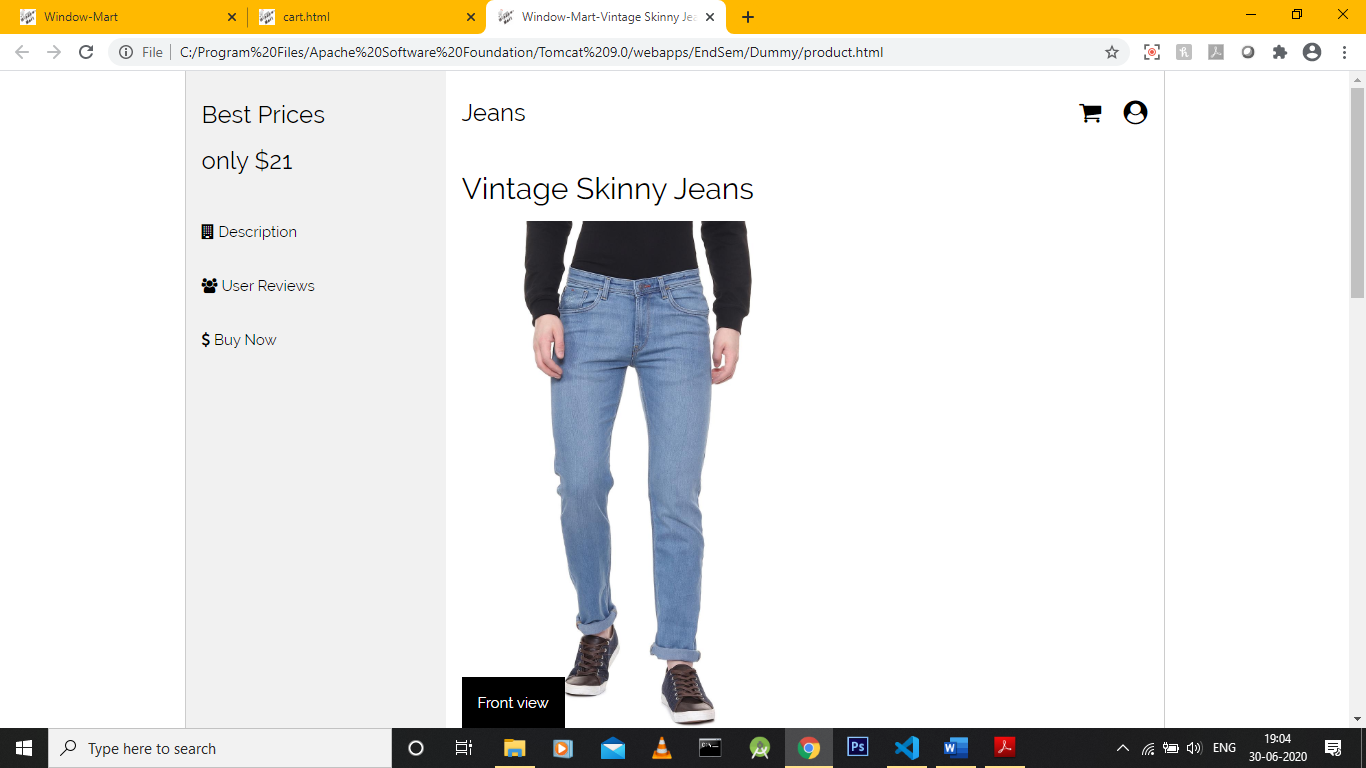
**Sample Input/Output Operations:**

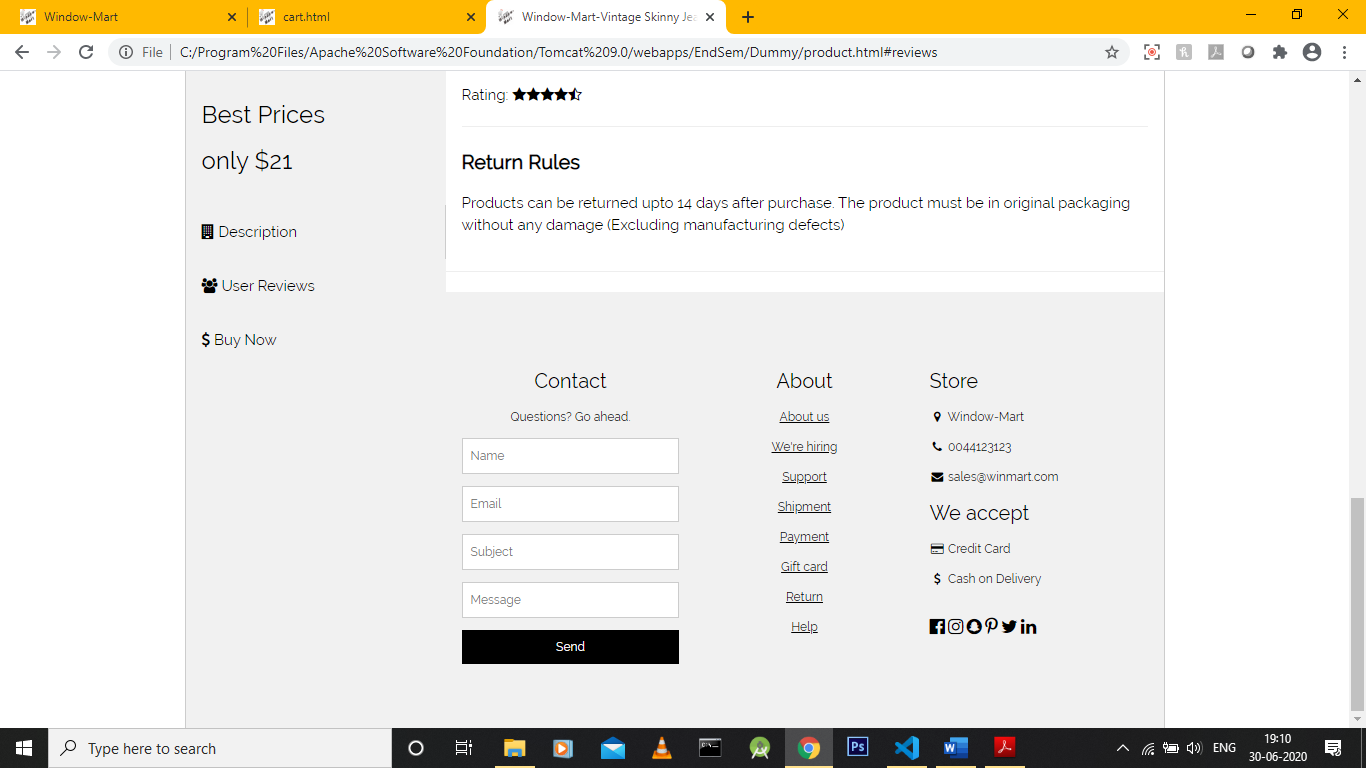


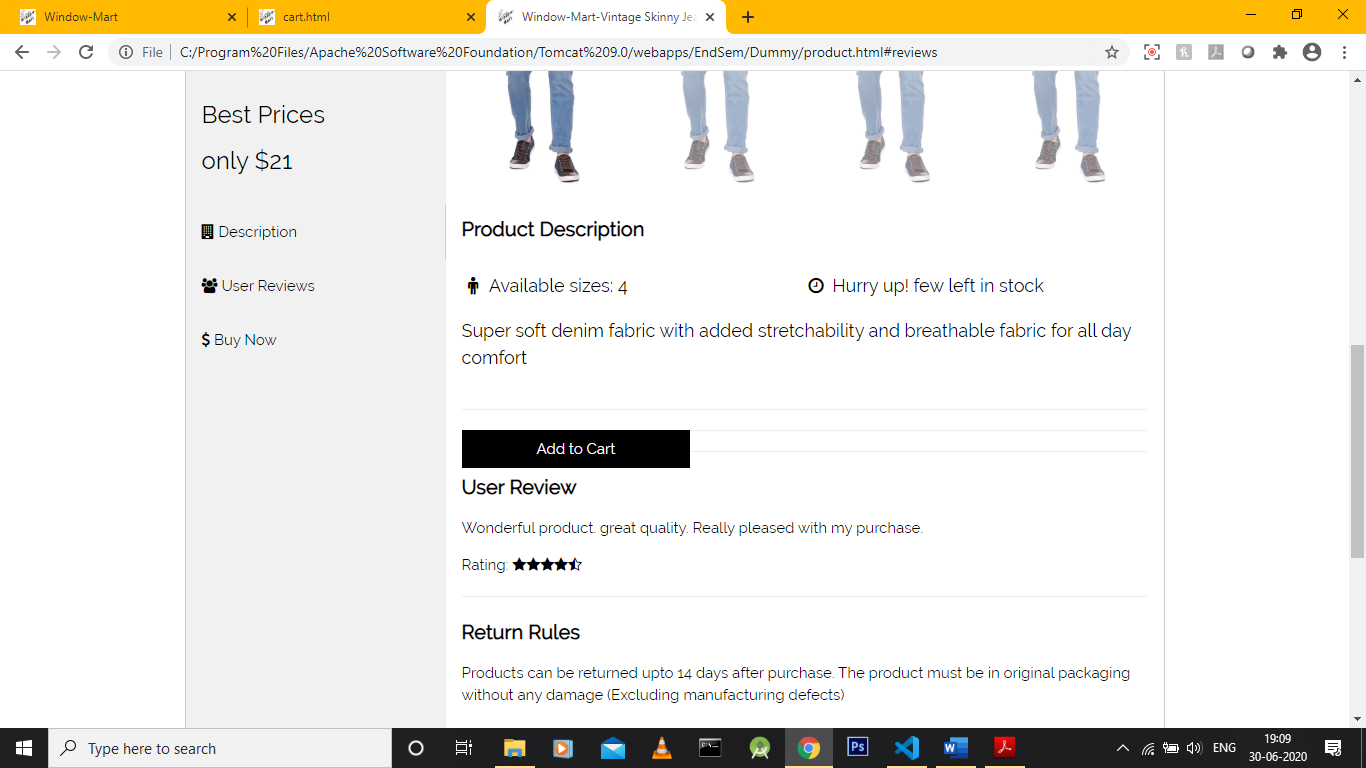


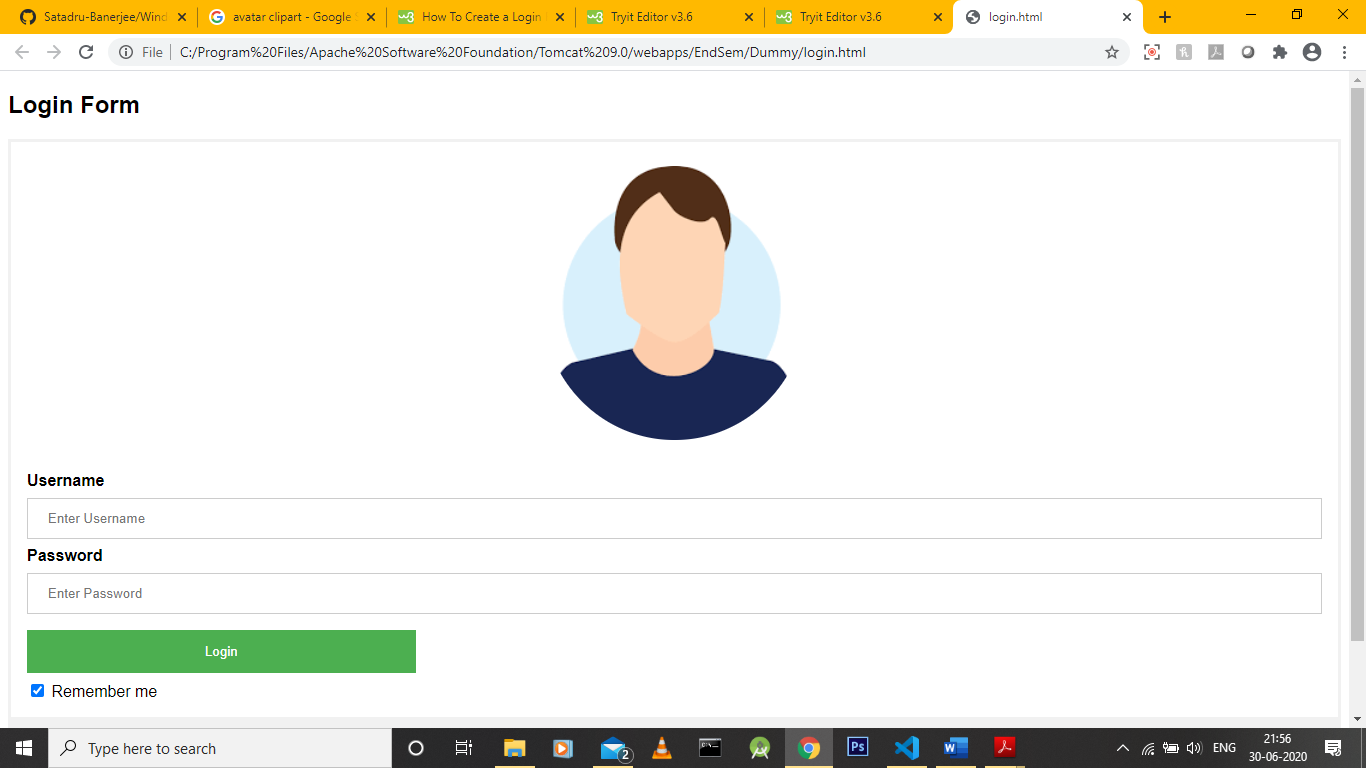


Product page:

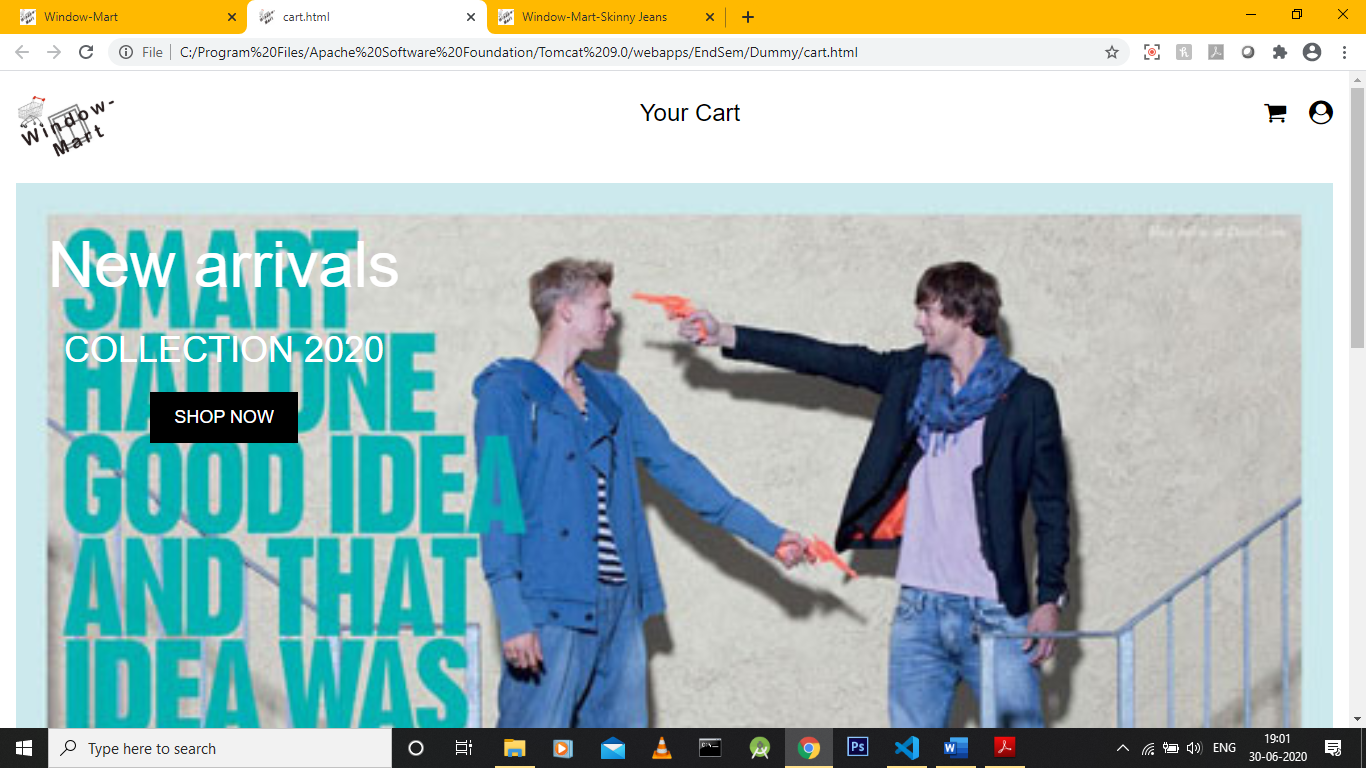


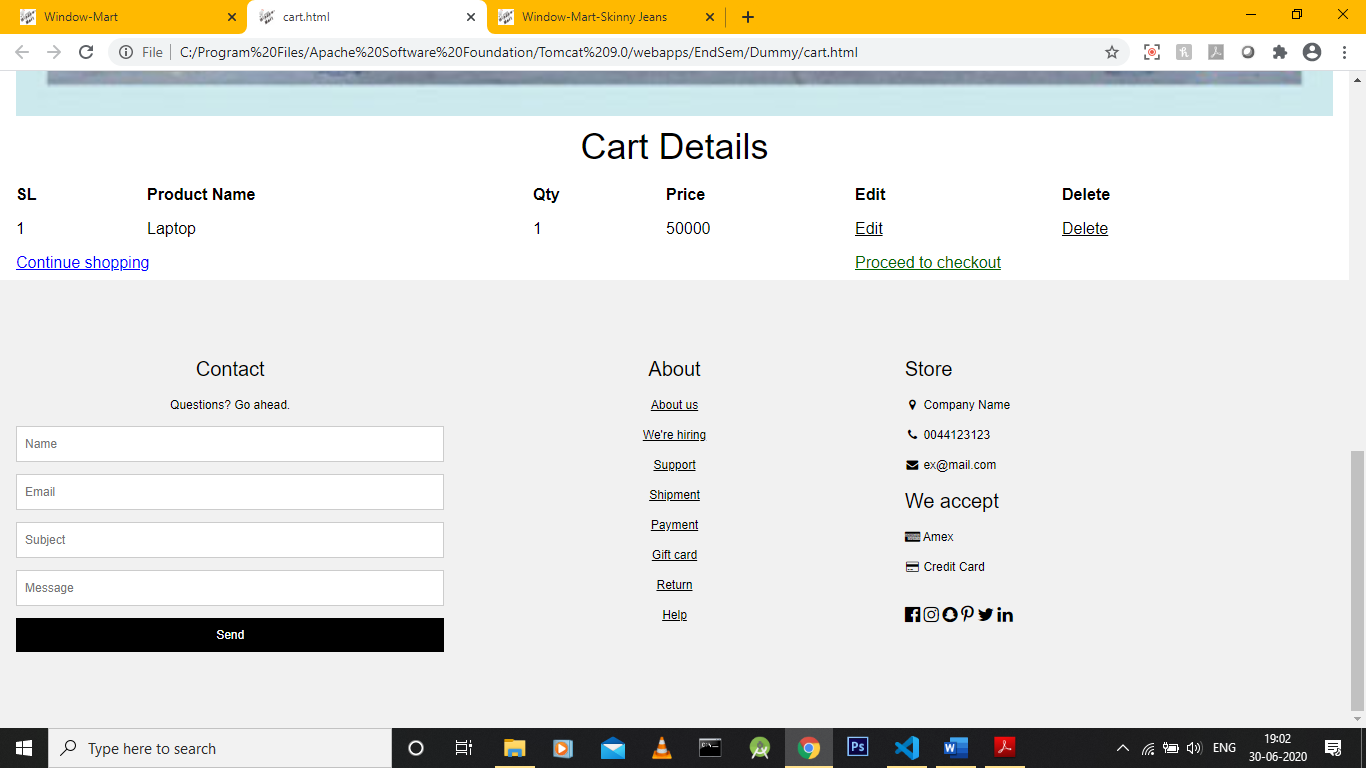




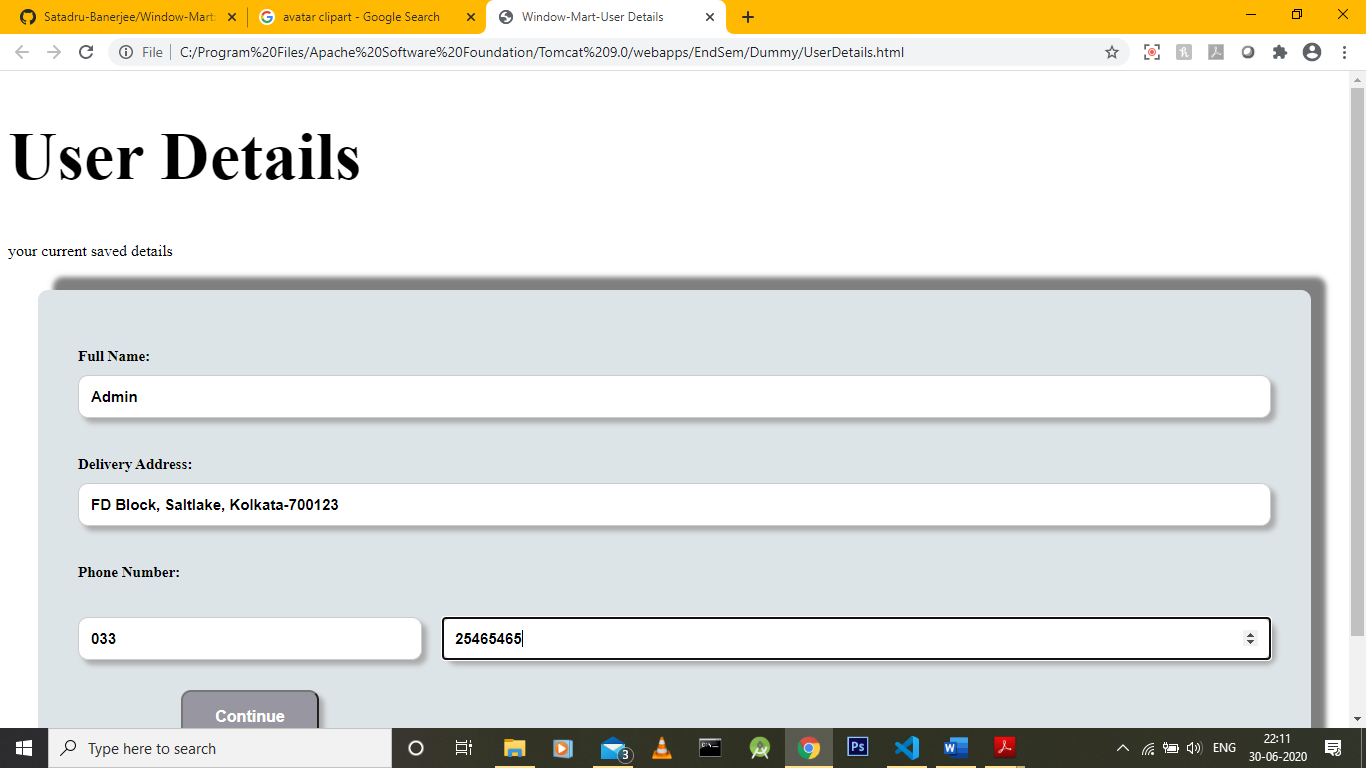


The cart:





Checkout details:



**Code Repsitory:**

<https://github.com/Satadru-Banerjee/Window-Mart>

**Conclusion:**

This project is an ideal example of how the MVC architecture can be easily implemented, making some of the most complicated problems modular and easy to edit. The project also has a number of features disabled at the moment which add a degree of scalability and the ability to expand this application vastly. Further features will include a product management system and a user editing panel and addition of new features. There is also the case of adding more products and online analytics features for traffic and engagement monitoring. However, in all, this project represents a perfect representation of the MVC architecture in a real-world scenario.