M30CDE Coursework

Anthon Johansson  
[johanss2@coventry.ac.uk](mailto:johanss2@coventry.ac.uk)   
Student ID: 6770097

Innehåll

[Description 1](#_Toc437871021)

[Functions 1](#_Toc437871022)

[Regular user 1](#_Toc437871023)

[Customers (logged on users) 1](#_Toc437871024)

[Admin 1](#_Toc437871025)

[Use case 2](#_Toc437871026)

[Overview Design 3](#_Toc437871027)

[ER diagram 3](#_Toc437871028)

[Wireframe Design 4](#_Toc437871029)

[Evaluation of development process 12](#_Toc437871030)

[Evidence of implementation 13](#_Toc437871031)

[Code and database 13](#_Toc437871032)

[Screen shots 14](#_Toc437871033)

[Test plan 22](#_Toc437871034)

[Test results 23](#_Toc437871035)

# Description

The e-commerce application is a web shop selling different smartphones.

## Functions

### Regular user

* Can log in
* Can register a customer account
* Can browse the item catalogue
* Can search for items in the item catalogue page
* Can add an item to the shopping cart
* If an item already exists in the cart and another identical one is added to the cart, the quantity will increase
* Can check the shopping cart and remove items from it
* The cart page will always show the total price of all the items in the cart
* To checkout and make an order, the user has to login to a customer account

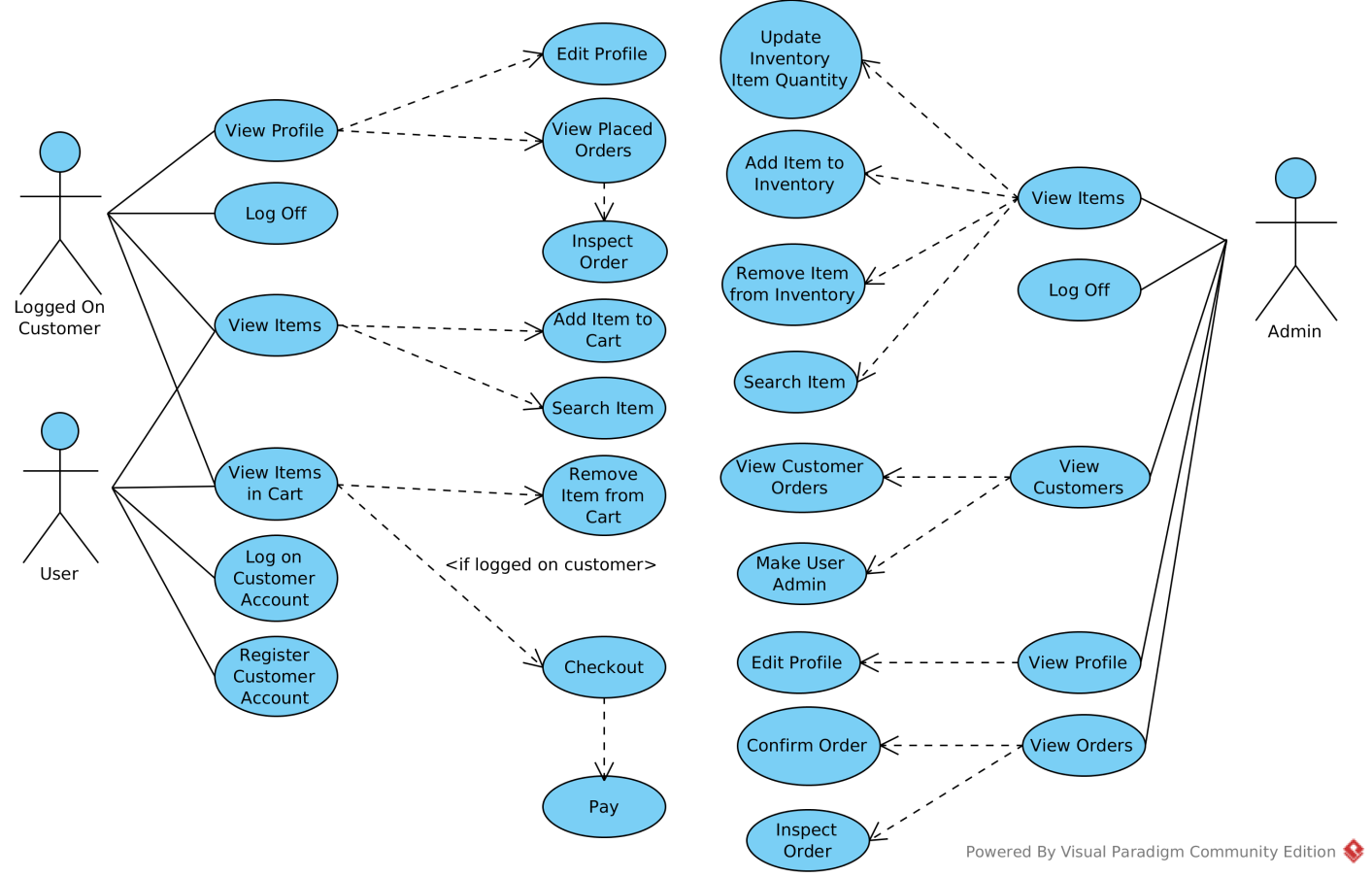
### Customers (logged on users)

* The same as regular users and the functions below
* Can log out
* Can view profile info
* Can edit the registered profile info
* Can view old orders made from this account
* Can inspect old orders to see which items that was ordered and the amount of each item in that specific order
* Can checkout the items in the shopping cart and make an order after entering a valid 10 digit credit card number, the shopping cart has to have at least one item in it to allow the checkout

### Admin

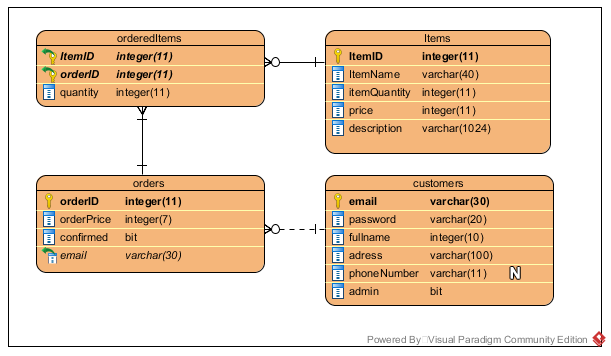
* The same as regular users, logged on customers and the functions/restrictions below
* Can't add any items to shopping cart or view the shopping cart, an admin can't shop at all
* Can't view orders made from this account, since an admin can't shop on the website
* Can view all the registered customers and their registered information, except for their password
* Can search for a customer
* Can make/unmake normal customers into an admin, can only perform this action on other accounts and not the current logged on admin account
* Can view the orders made by every registered customer
* Can inspect all placed orders to see which items were included in each order and the quantity of each included item, just like logged on customers can inspect their own orders
* Can confirm the delivery status of a placed order (received=order placed but not processed by admin, delivered=processed, packed and sent to customer by admin)
* Can add new items to the items page (the inventory) by entering a valid item id, item name, item quantity, item price and item description
* Can update the quantity of existing items by entering a valid amount and marking the add or remove radio button for the specific item
* Can remove an item from the items page (the inventory) completely

## Use case



# Overview Design

## ER diagram



The primary keys are marked with a yellow key to the left of the name. The foreign keys look about the same but have a green arrow on it. PhoneNumber in the “customers” table is the only nullable attribute.

A registered customer can make zero or many orders. An order will include at least one or many items. An item may be added to zero or many different orders. The OrderedItems table exist because of this many to many relation and has an additional quantity attribute in case several of the same item is placed in the same order.

## Wireframe Design

To see all the wireframe design pages and use buttons: <https://www.lucidchart.com/documents/view/ae564ac8-5cfa-4b49-a839-6741d3c7a9e9/6>

Figure 1. The ordinary startpage.

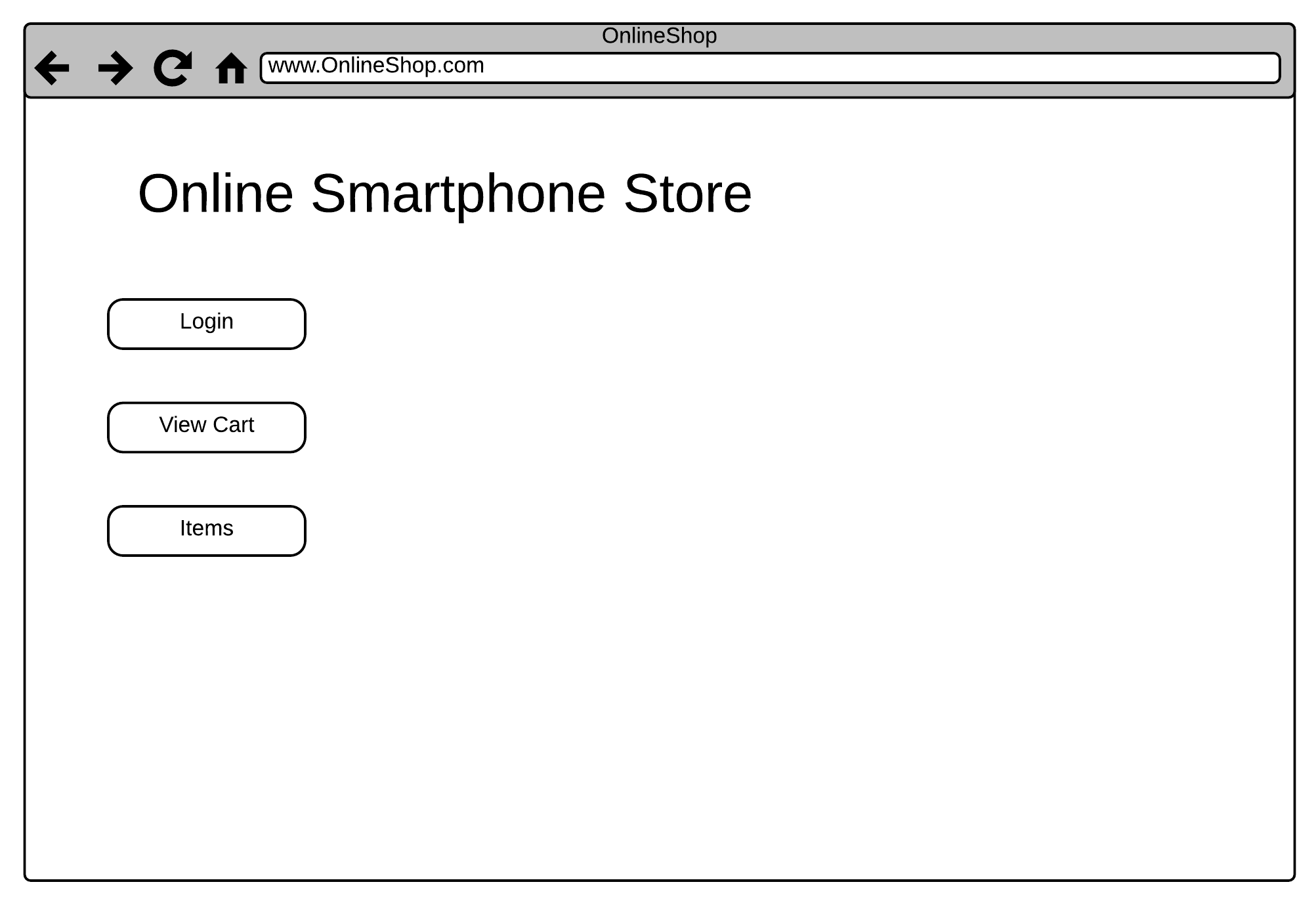
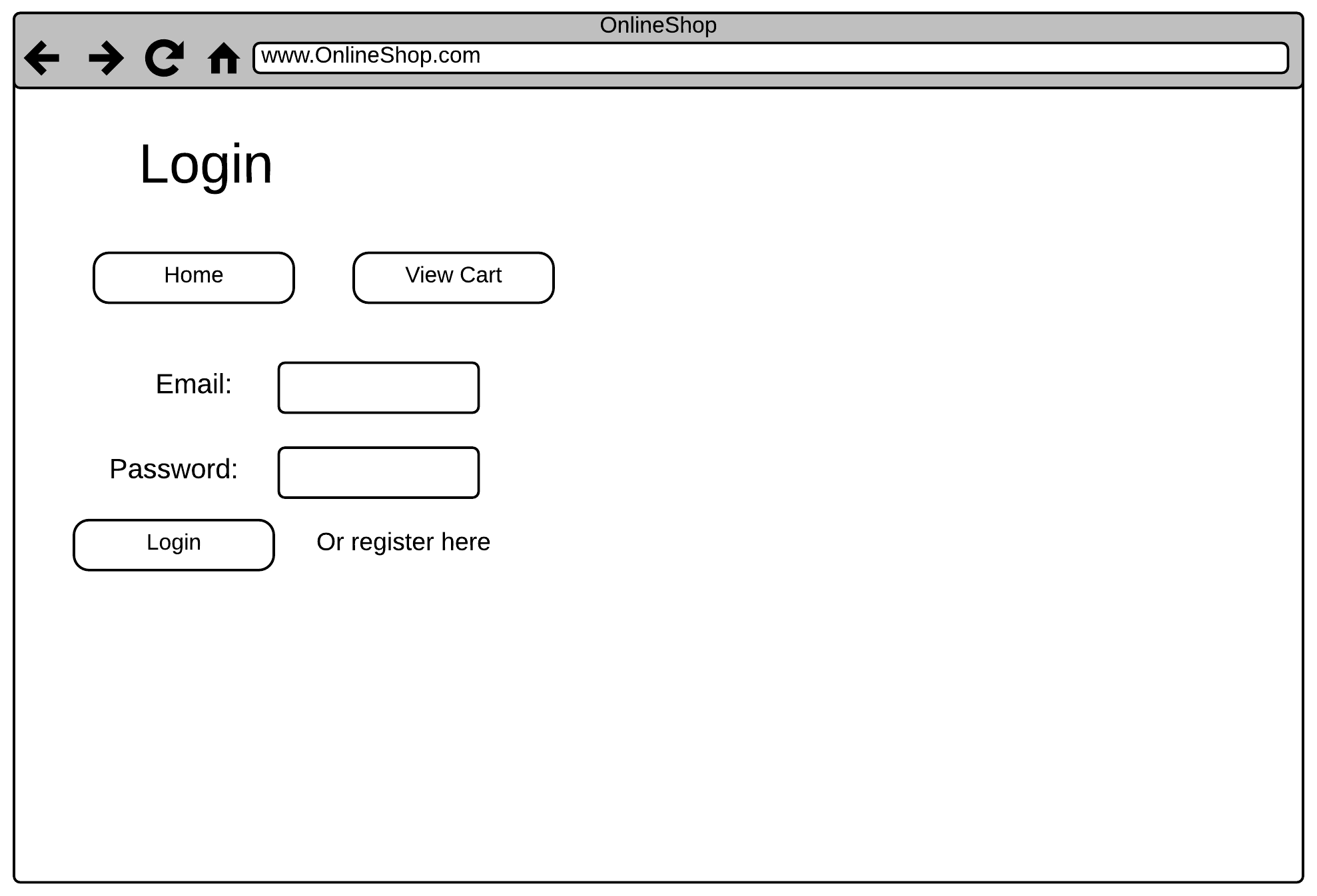


Figure 2. The login page, “here” is a link to the register page. Both users and customers can use cart.



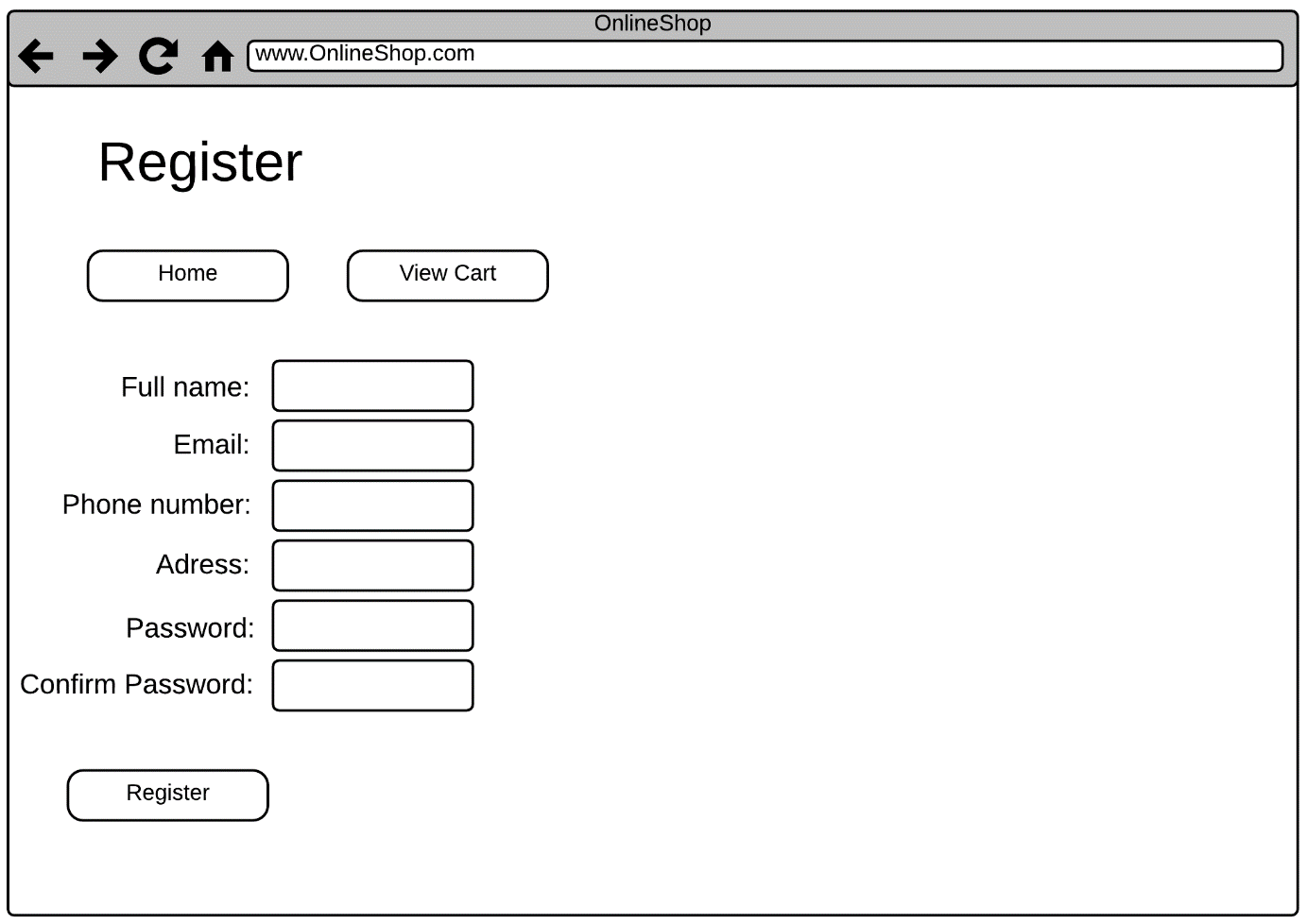


Figure 3. The register page, the cart is available from this page as well.

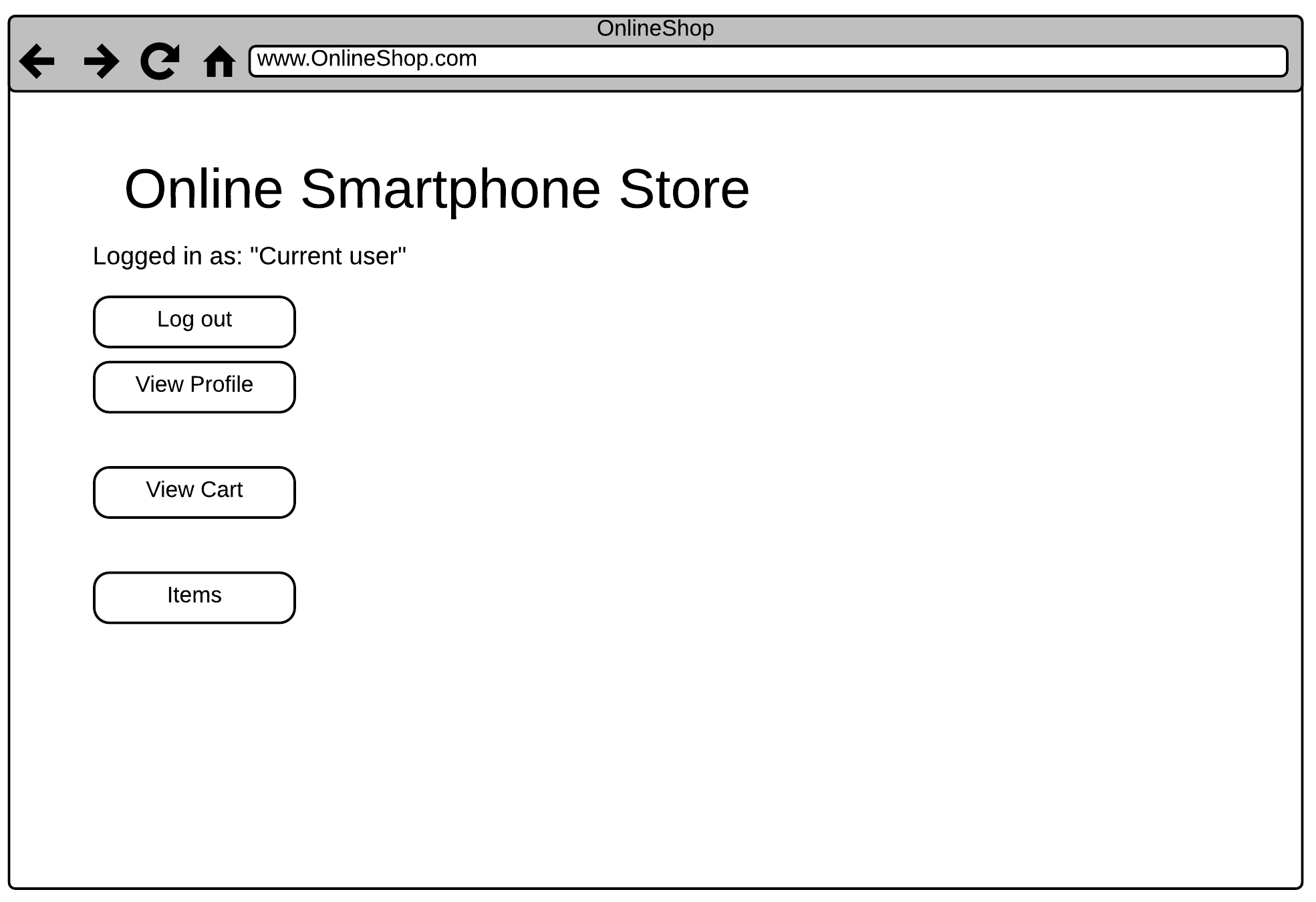


Figure 4. The startpage available for logged on customers, can log out, view profile, view cart items and view the items page.

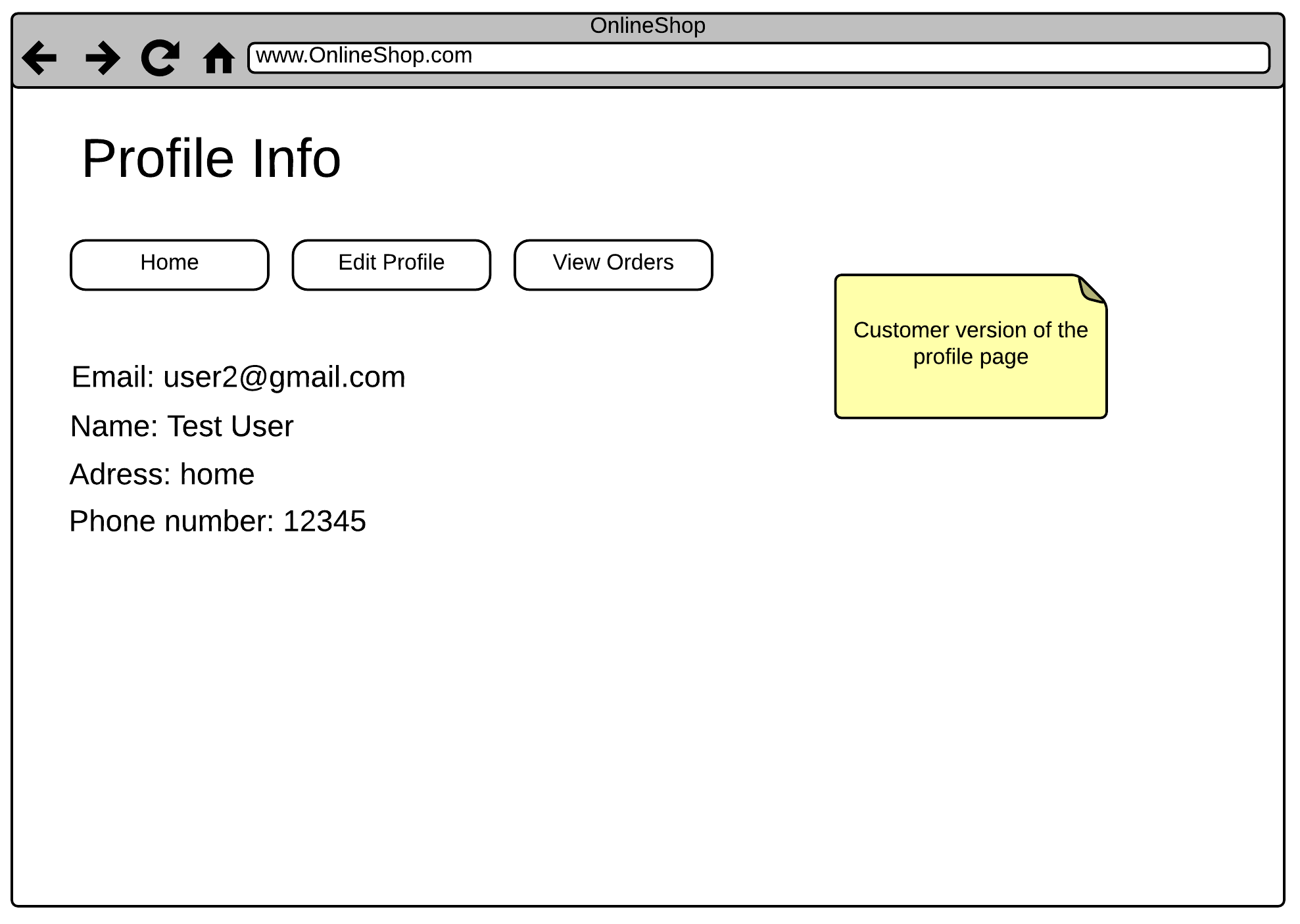


Figure 5. “View Profile” will open this page, the customer can edit profile info or view orders made from this account.

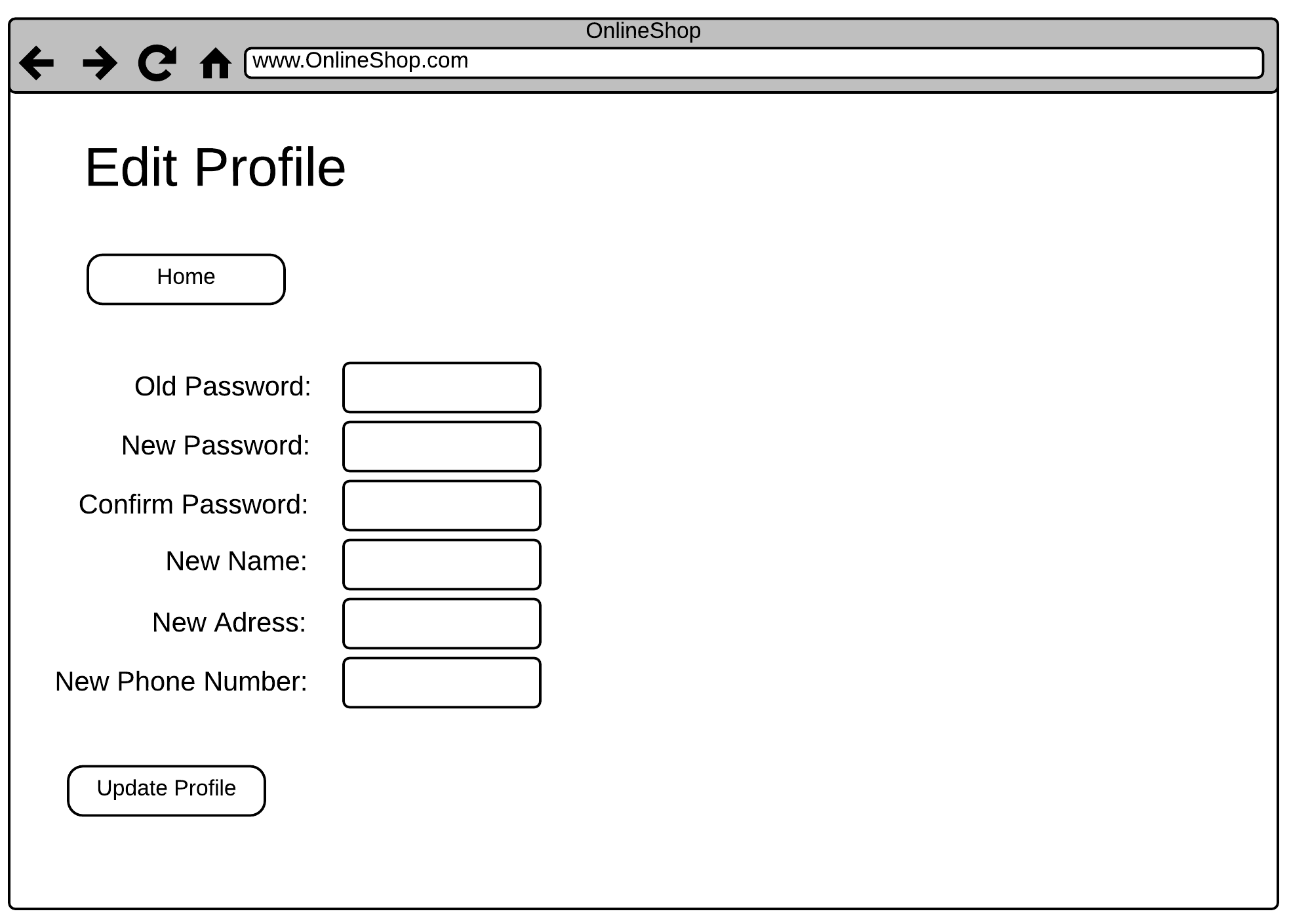


Figure 6. Page for updating profile info. Only the fields that aren’t left blank will be updated to the profile info.

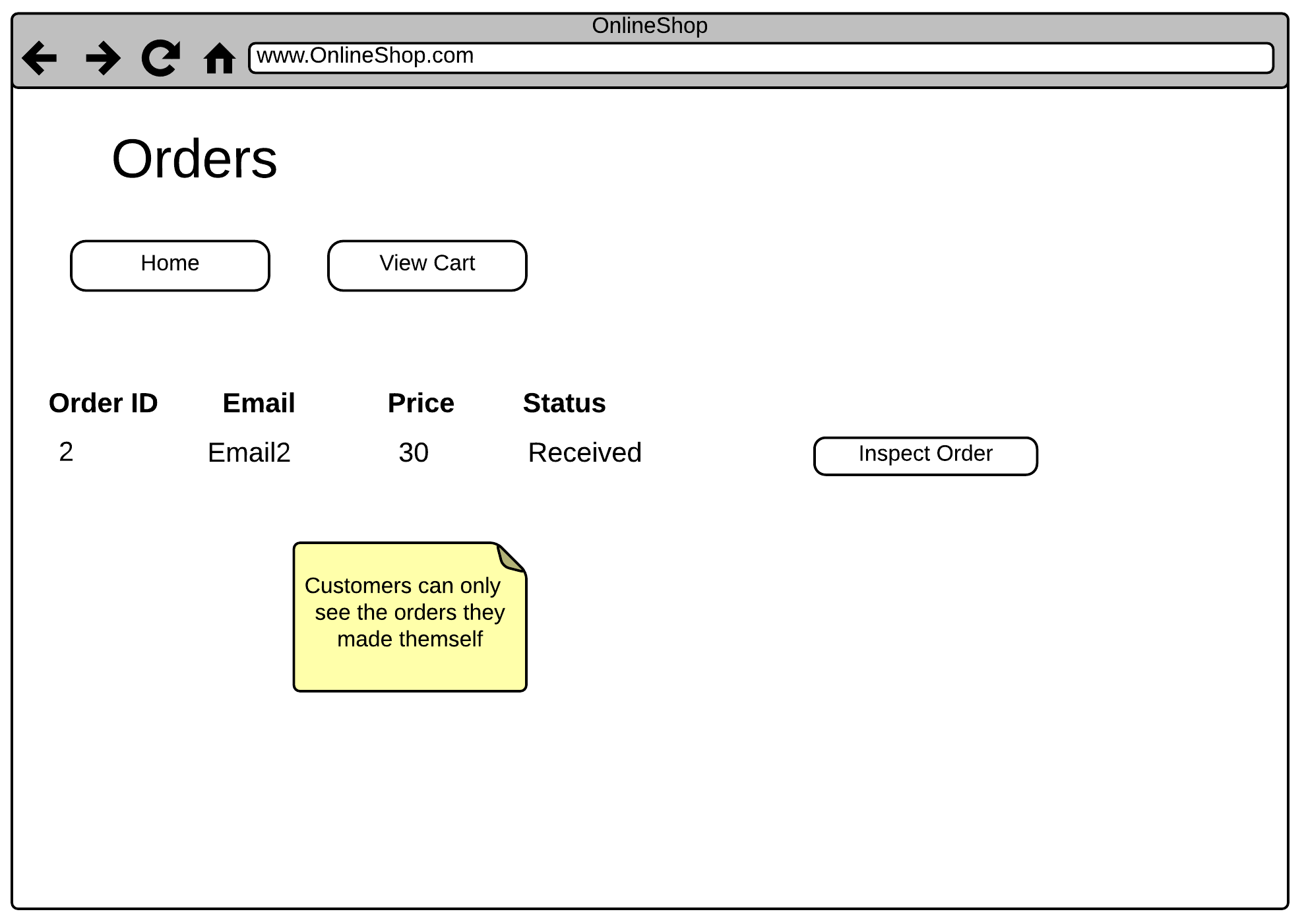


Figure 7. This page will show all orders made from this customer account. Each customer can only see his/her own orders.

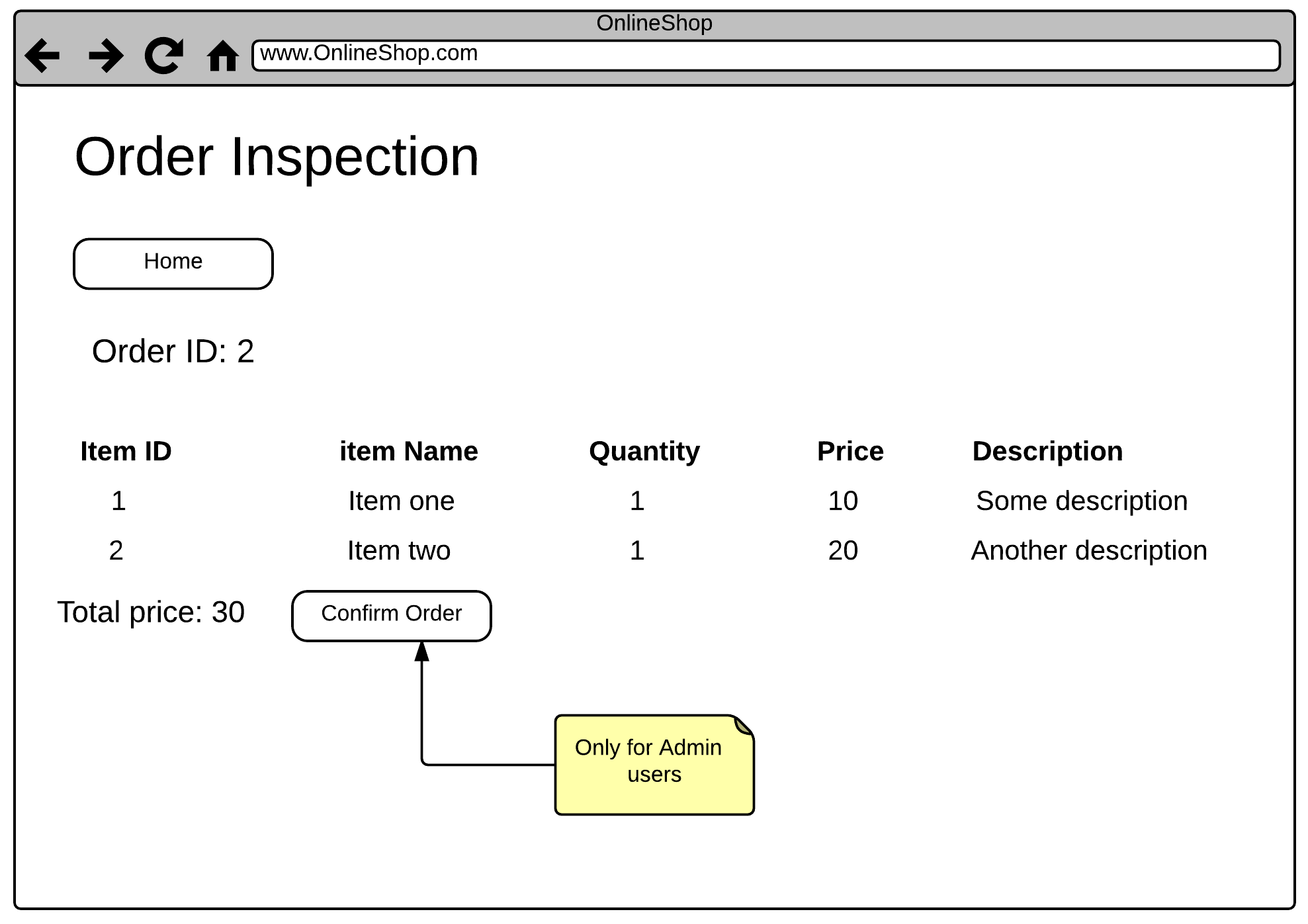


Figure 8. This page will show the items and total price of an order. Admins has the option to confirm the order status.

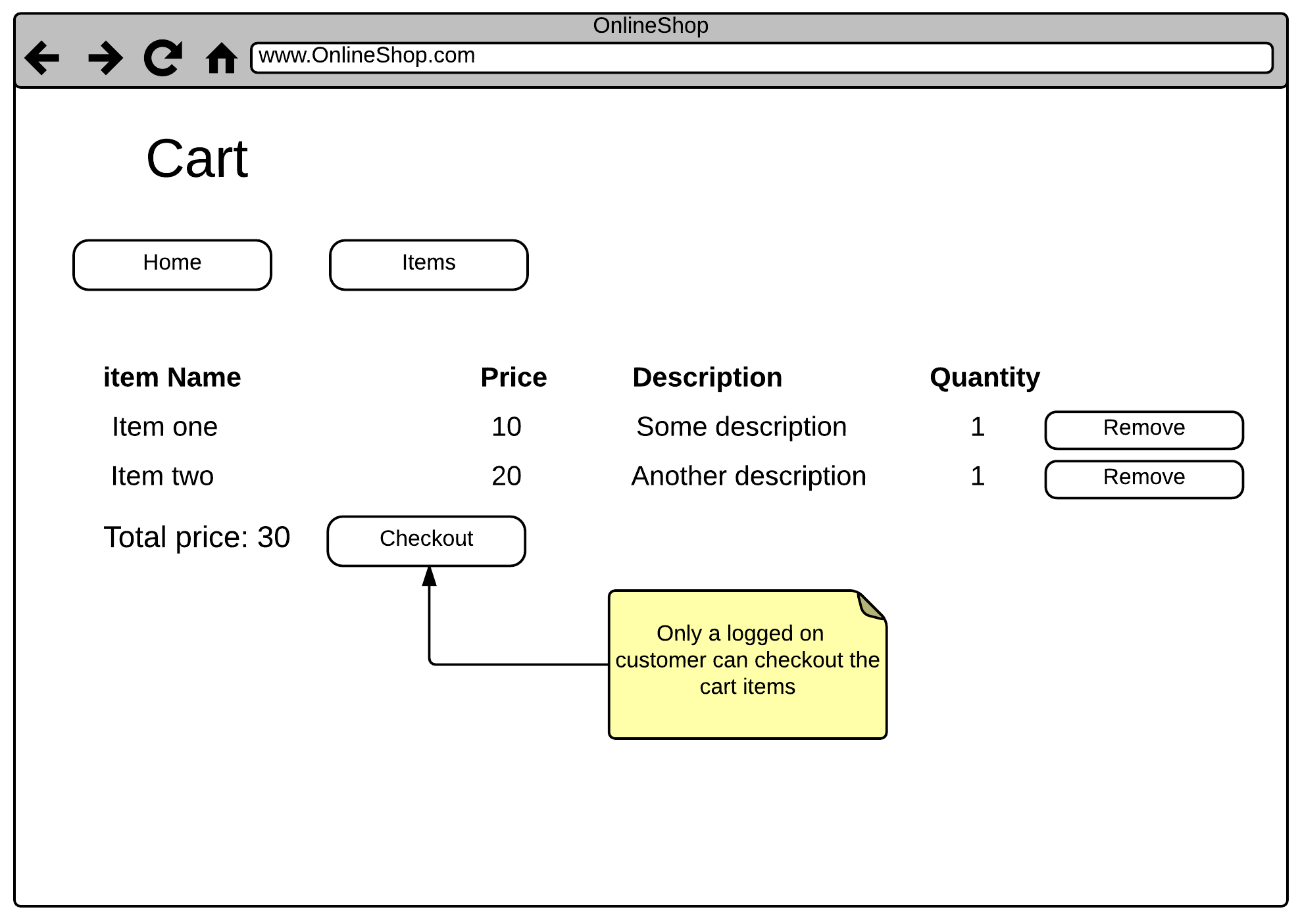


Figure 9. This page displays the items in the cart and the total price. Only a logged on customer can make the order.

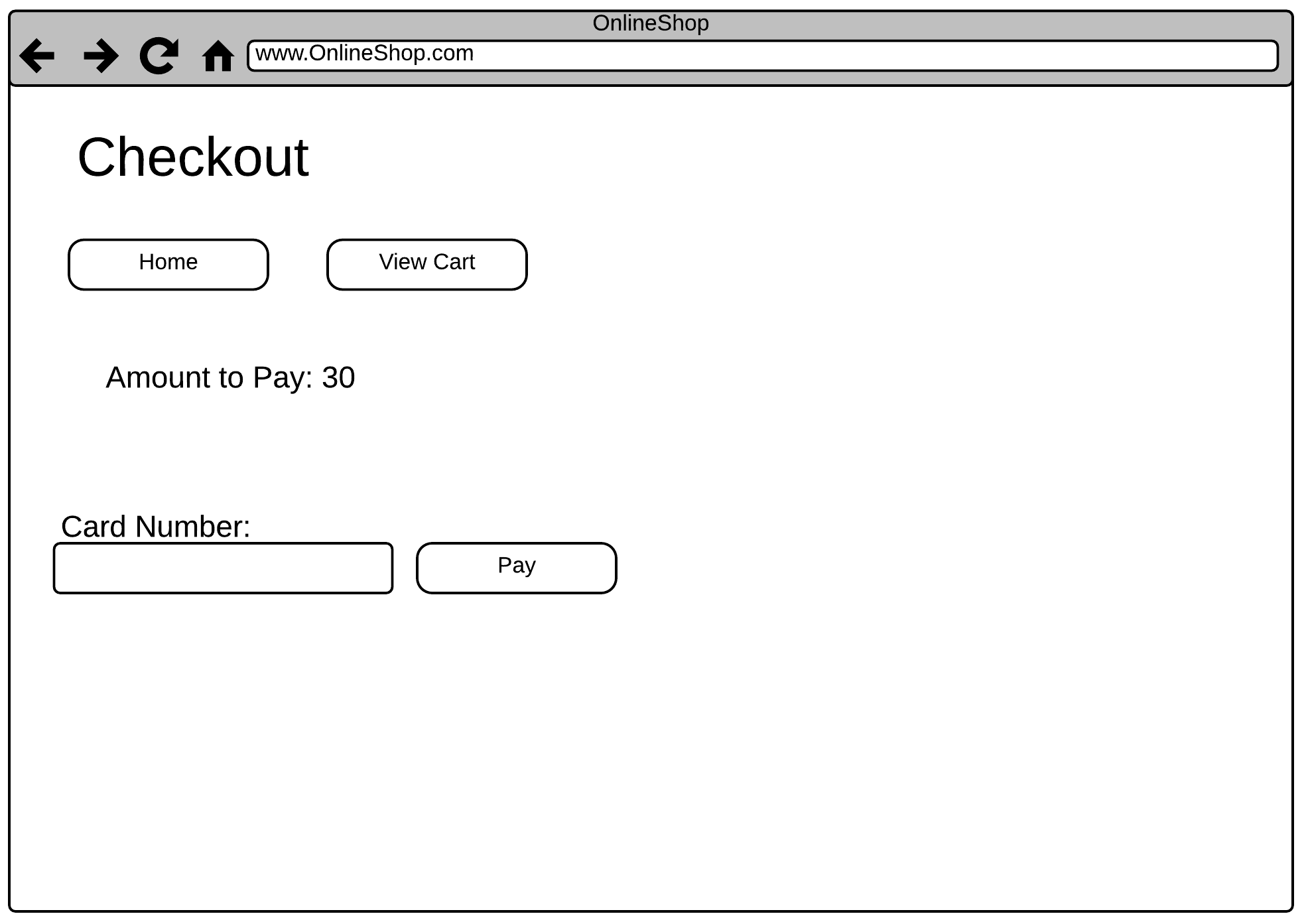


Figure 10. The checkout button will redirect to this page. A 10 digit card number needs to be entered to make the order.

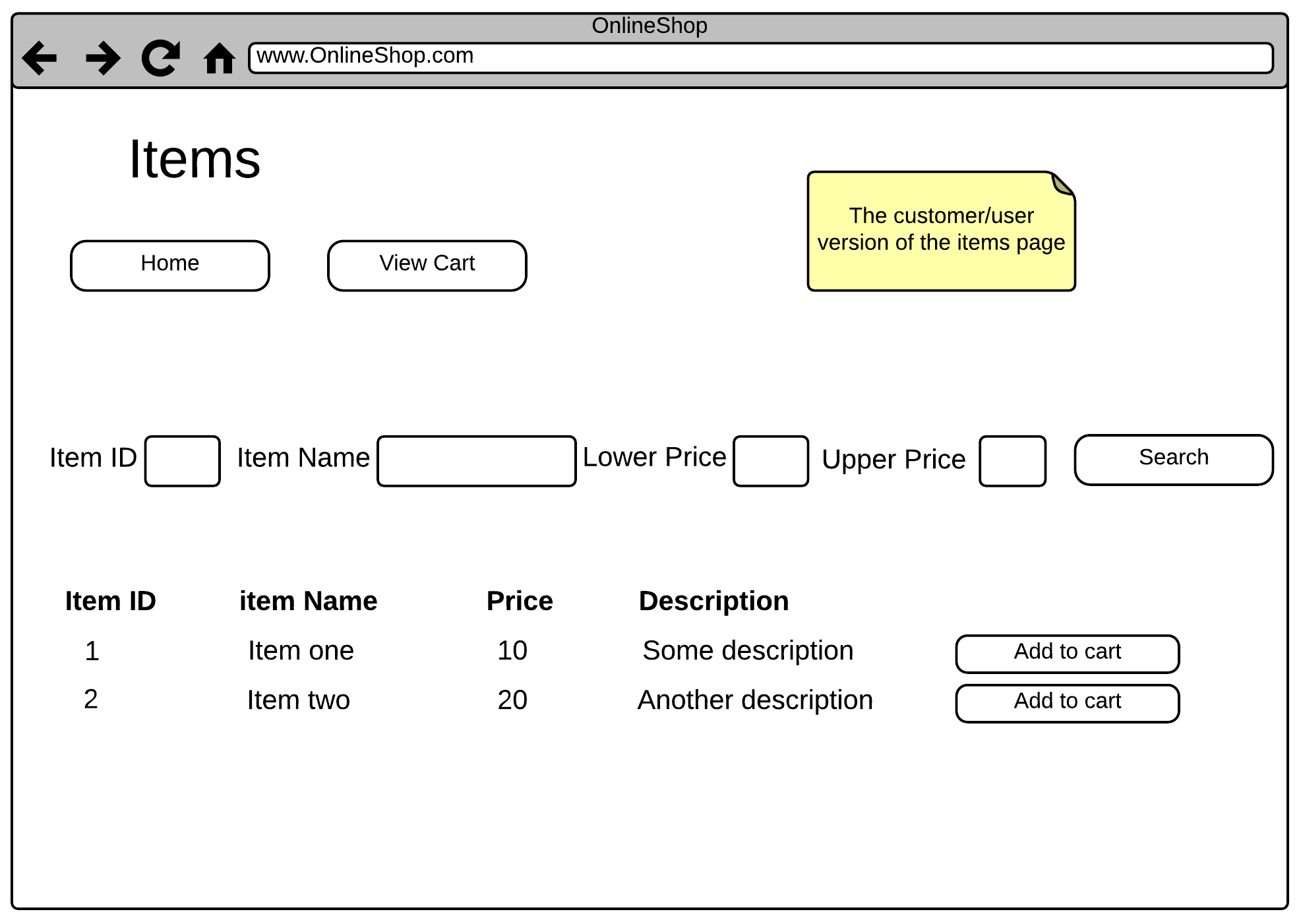


Figure 11. The items page for customers and normal users. Both can add items to their carts or search for an item.

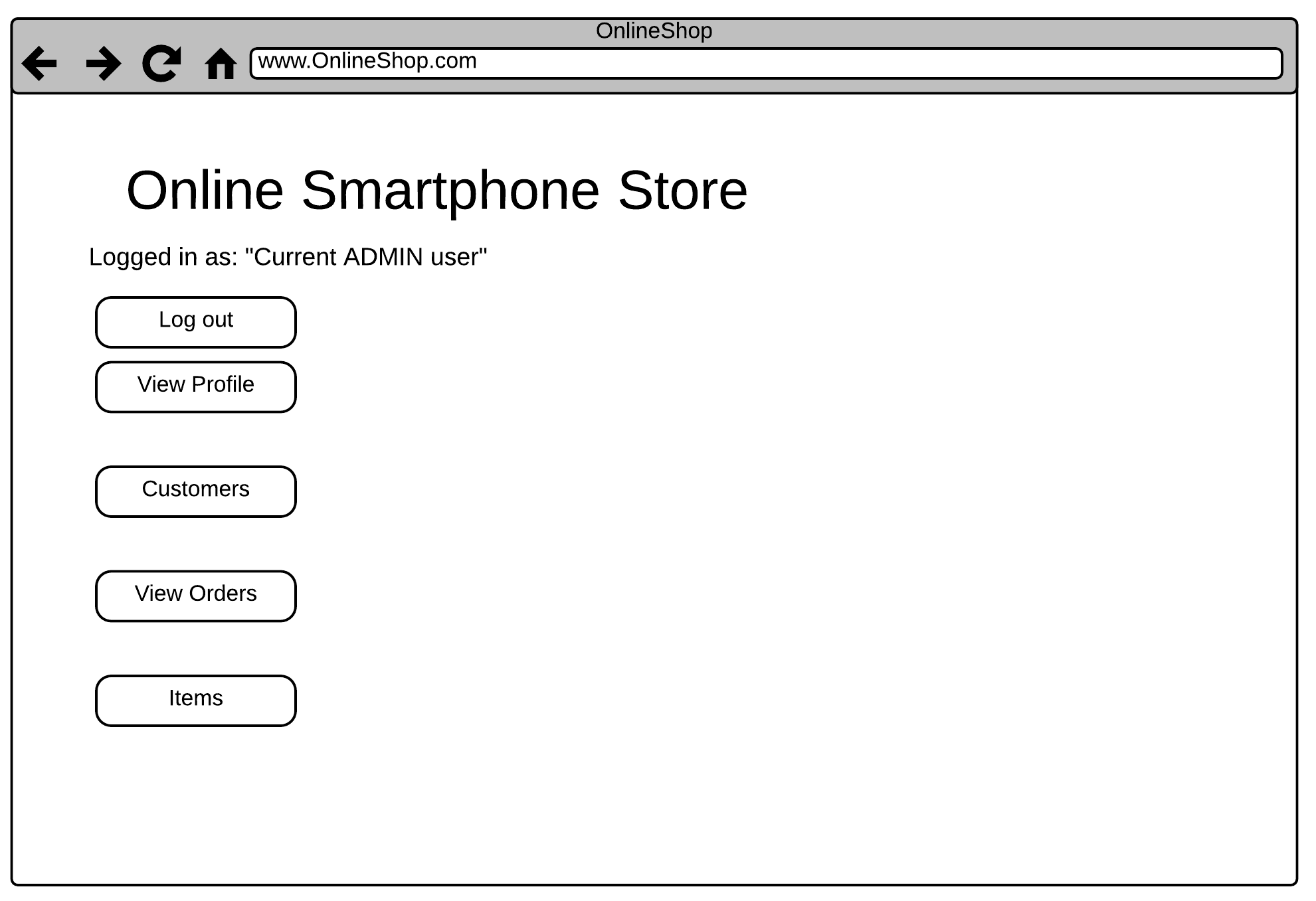


Figure 12. The start page for Admin users. Admins can View all orders from every customer and the customer information from all customers. Admins can view the items page and view their profile page as well.

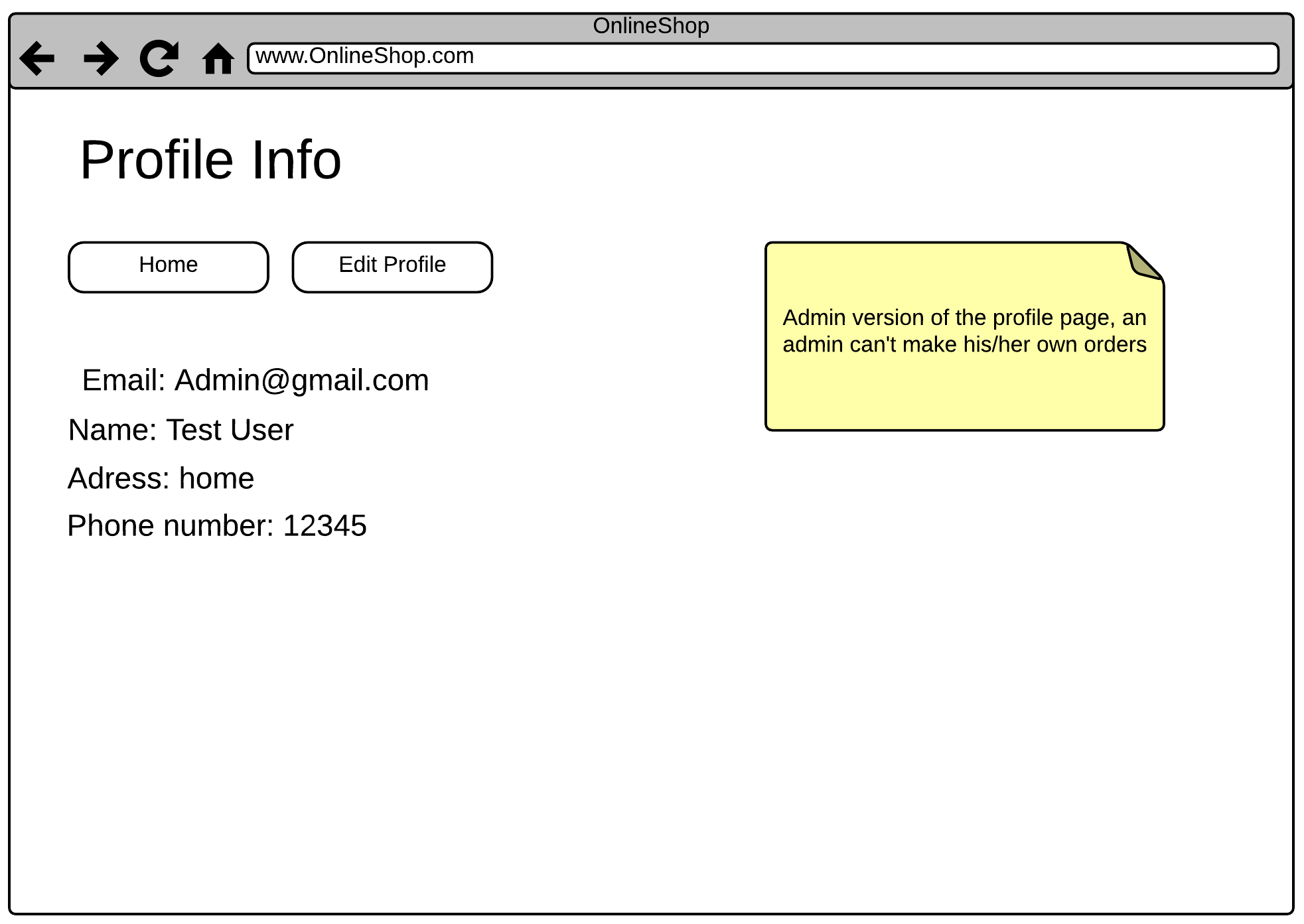


Figure 13. The profile page for Admin users. Admins can’t view their own orders since they can’t shop at all.

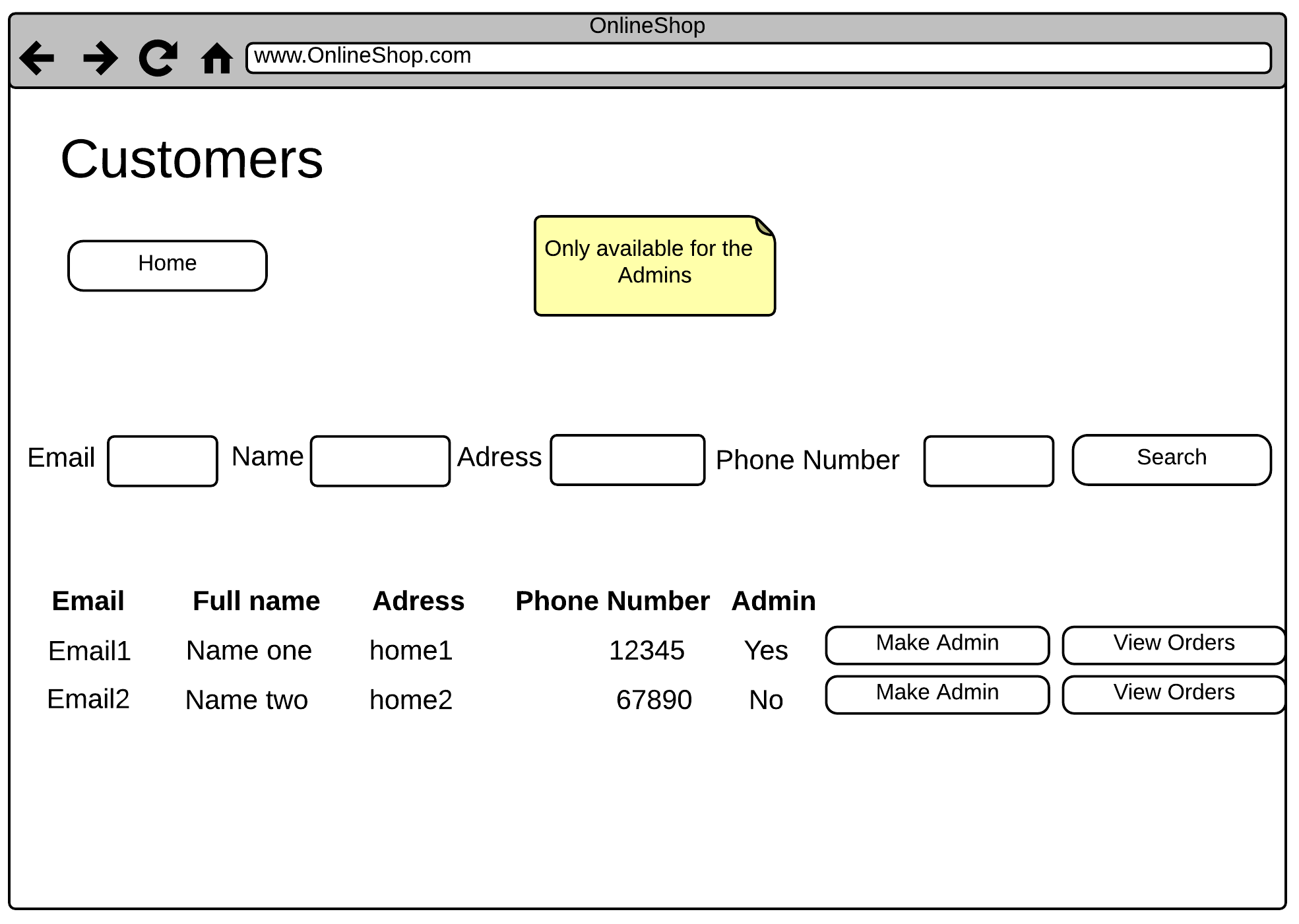


Figure 14. Admins can use this page to view details of every customer or other admins. “View orders” will show the orders made by that customer (like Figure 7).

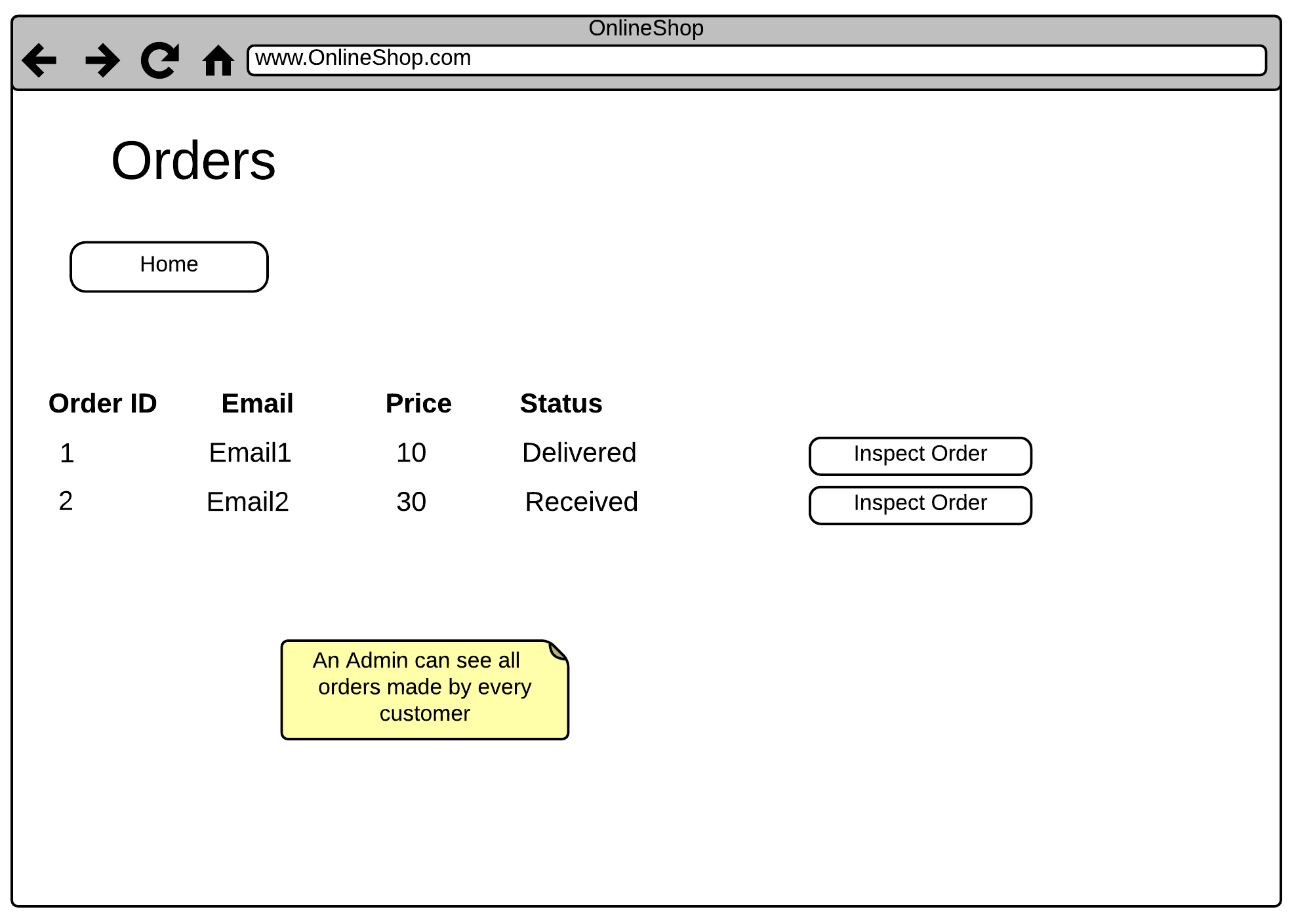


Figure 15. The “View Orders” button on figure 12 will lead to this page. The orders are made by two different customers.

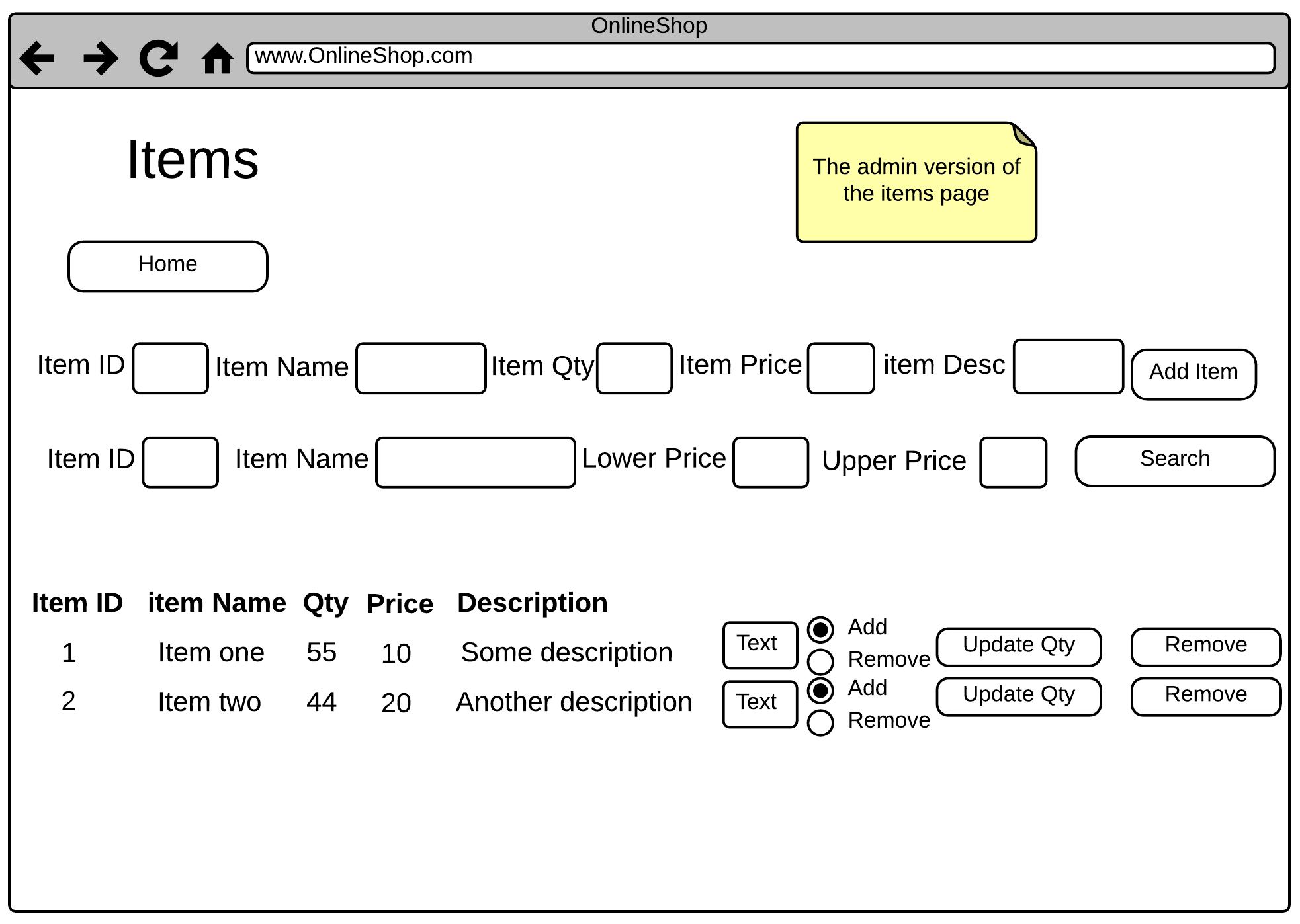


Figure 16. The items page for admins. Admins can’t add items to a cart, but can update/modify the item inventory and add new items to the items page.

# Evaluation of development process

The development process started with the creation of all the jsp pages, some are still used while others were removed because of redundancy. To make the code more readable and to separate html code from java code, a servlet was made to handle all the logic and computation. There is some java code in the jsp pages but only for looping trough lists stored in the session object and for determining which buttons that should be visible to the current user (different privileges). A MySQL database is used instead of the java database since it offers more functionality.

There are both normal users, registered customers and admins that can use the e-commerce website. Each of these types of users has different privileges, java “if” statements were used in the jsp pages to separate them without adding several different jsp pages for each privilege level.

Whenever a button is used in the jsp pages the request will be sent to the servlet, which will perform any necessary computation then redirect to the correct page. All buttons has a certain “action” name which will activate a different “case” in a java “case statement”, the case will in turn use private functions in the servlet class to perform computations.

Any content from the database will be passed from the servlet to the jsp pages through the session object, for example item lists, contents of an order, customer list etc. This also means that the servlet is the only object that has access to the database, this makes the application more secure since the servlet can check if the input it received is valid or invalid (or sql/script injection).

The first part that was finished was the customer login and registration functionality. A few arguments are set in the session object on each login to store info about the current user and the privilege level of the user (used by the if statements in the jsp pages). All the validation of input data are performed in the servlet after login information or registration info has been entered in all text boxes. If invalid info has been entered the servlet will redirect to the requesting jsp page and display error messages next to each text box with invalid data.

After login and registration was functional the jsp pages for items (products), customers and orders were finished. All of these pages are very similar in functionality since they request lists from the servlet and offer the ability to search for specific rows. The only differences between them was the additional buttons that appear next to each listed row, depending on privilege level different buttons will appear to add customer or admin functionality.

Most of the coding was done in the servlet since it performs validation controls and modifications to the database. To make the code more readable and easy to understand, I added a java case statement to the servlet method that receives and processes the incoming requests. The case statement will invoke a specific (small) private function located in the servlet class depending on the incoming request parameters. This is by far better than having one huge servlet method since you can add or modify functionality simply by adding or modifying a small private function. One thing that was problematic was to redirect from one case to a different case in the servlet case statement (no “goto” functionality exists in java). This is because it is not possible to modify an existing parameter in the request object sent from the jsp page. It is however possible to add attributes to the request object. So before performing any action requested by the request parameters the servlet will check if it requested a certain case in the case statement by looking at the request attributes first.

# Evidence of implementation

## Code and database

The code and database files (and documentation files) will be available in the following github repository: <https://github.com/antjo/M30CDE-App.git>

Some useful login information:

MySql Database:  
Username=”root”  
password=”password”

You can also find this info in the servlet constructor method where the database is initiated. Please note that the servlet thinks the database is named “OnlineShopDB” and that it will run on “localhost:3306”. This is a MySql database which means that you may have to download some external library for the MySql drivers.

The database files can be found in the “Database” folder, either as a “\*.sql” file or in the “OnlineShopDB” folderwhich contains the actual database files (\*.frm and \*.opt files).

Application:

Username=”[admin@gmail.com](mailto:admin@gmail.com)” Only existing admin account for now.  
password=”password”

Username=”[user\*@gmail.com](mailto:user*@gmail.com)” ‘\*’ is an integer in the range 1-6, existing customer accounts.  
password=”password”

## Screen shots

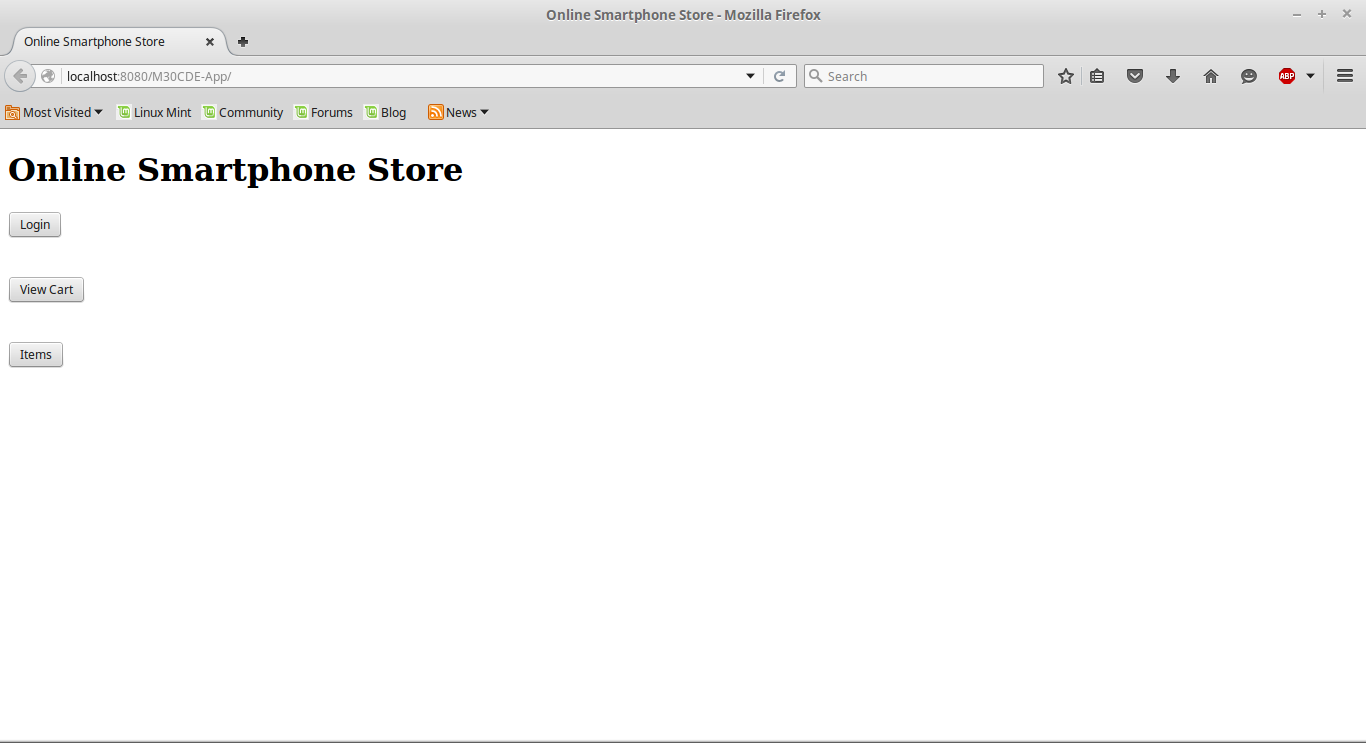


Figure 17. Normal start page for users who haven’t logged on yet.

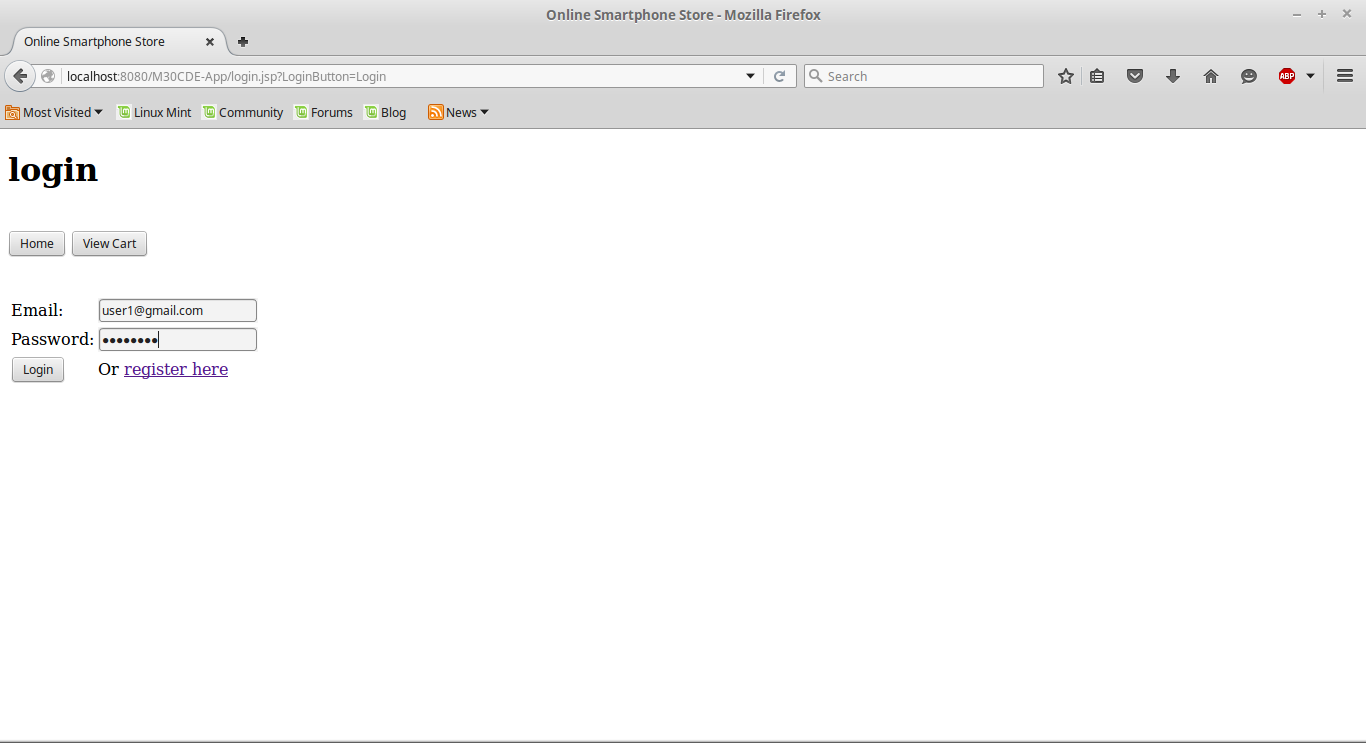


Figure 18. How it might look like when a customer logs on.

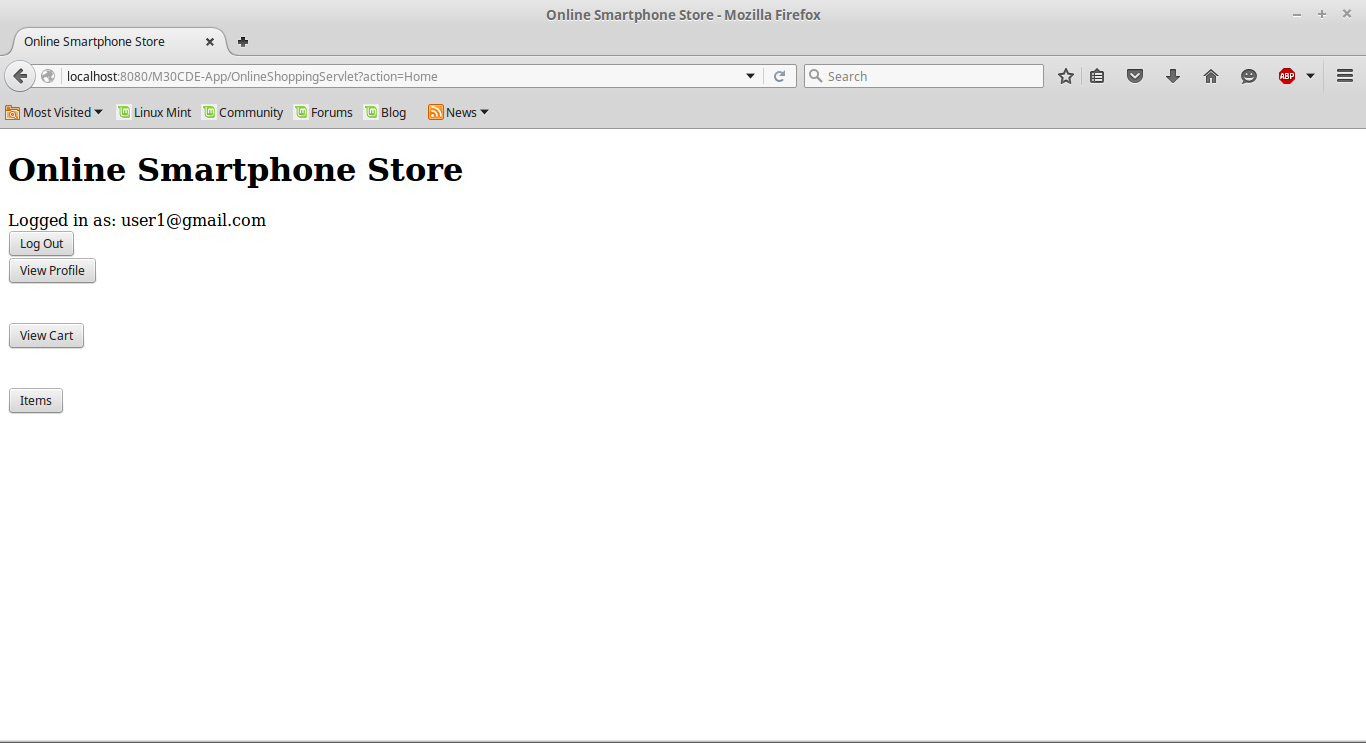


Figure 19. The screen after a customer successfully logs onto his/her customer account.

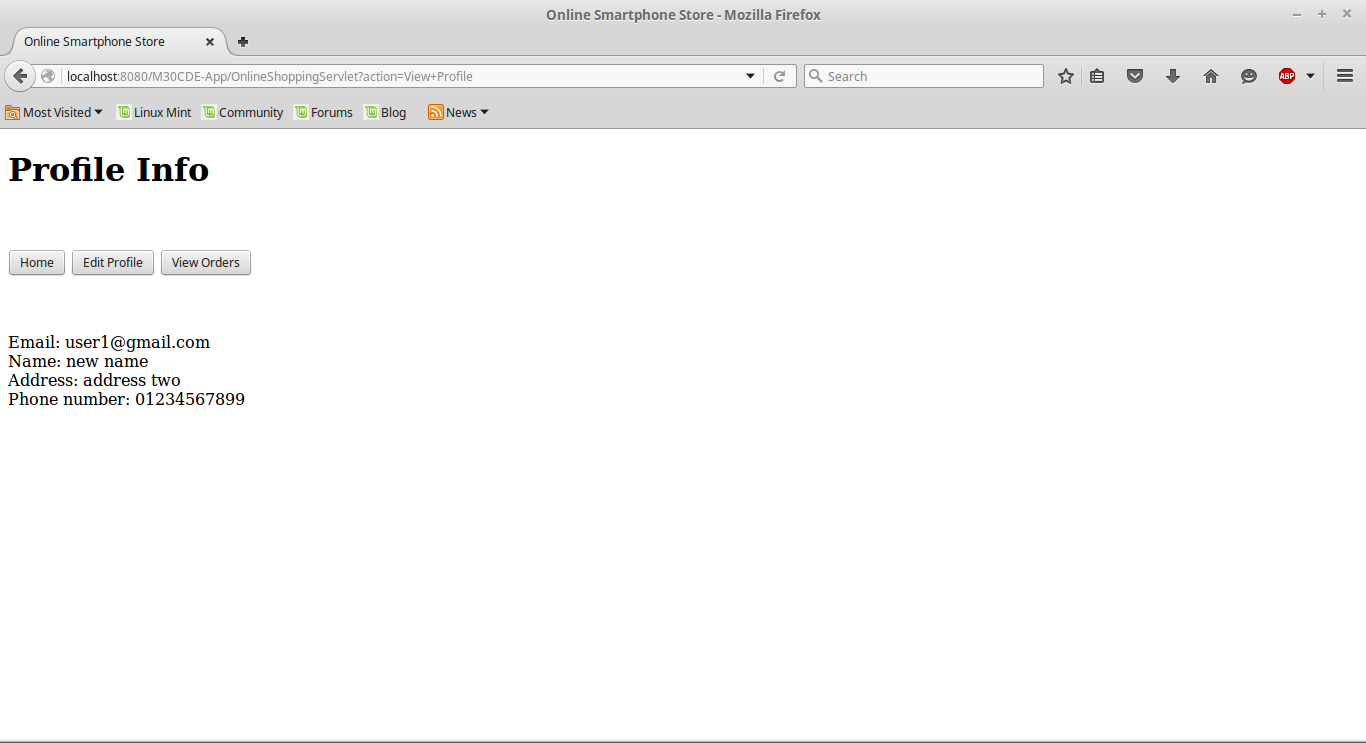


Figure 20. How it looks like when a customer views his/her profile info. An admin will have the same page but without the “view orders” button.

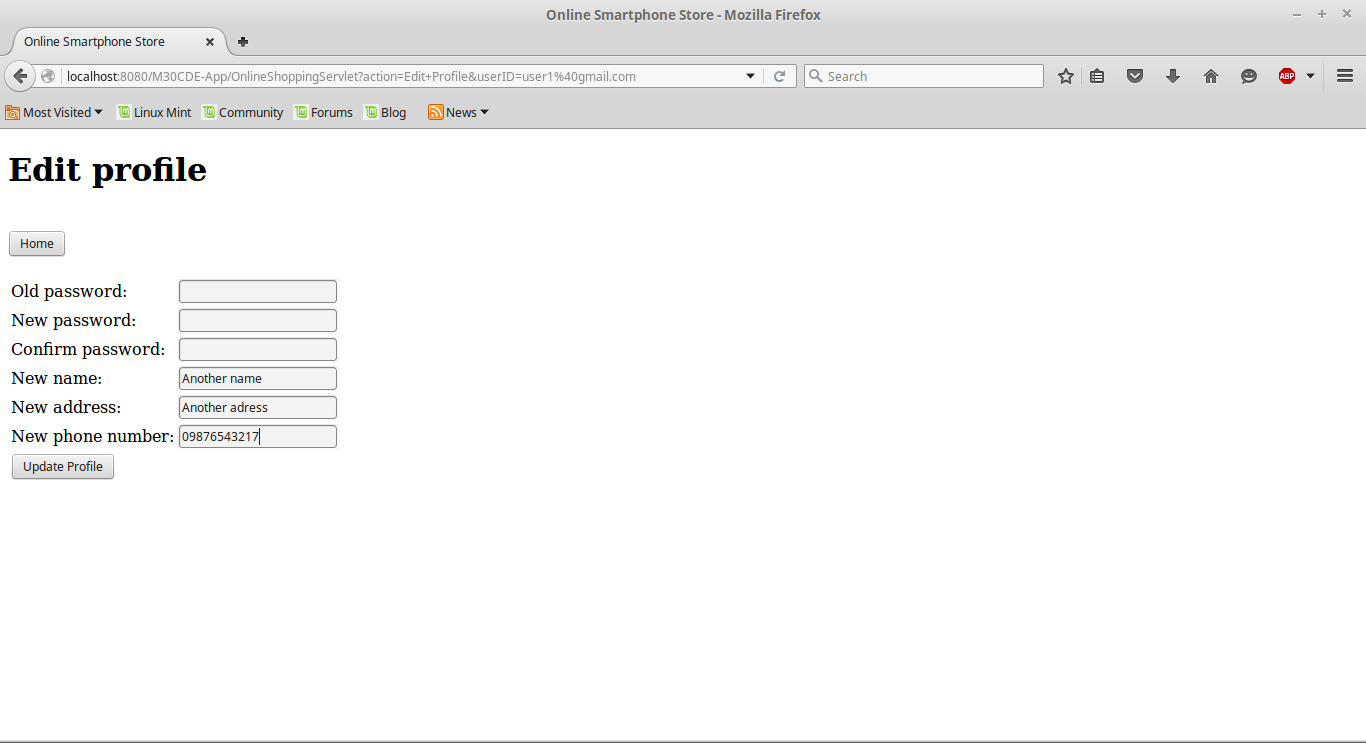


Figure 21. The page for editing profile info and some typical valid input.

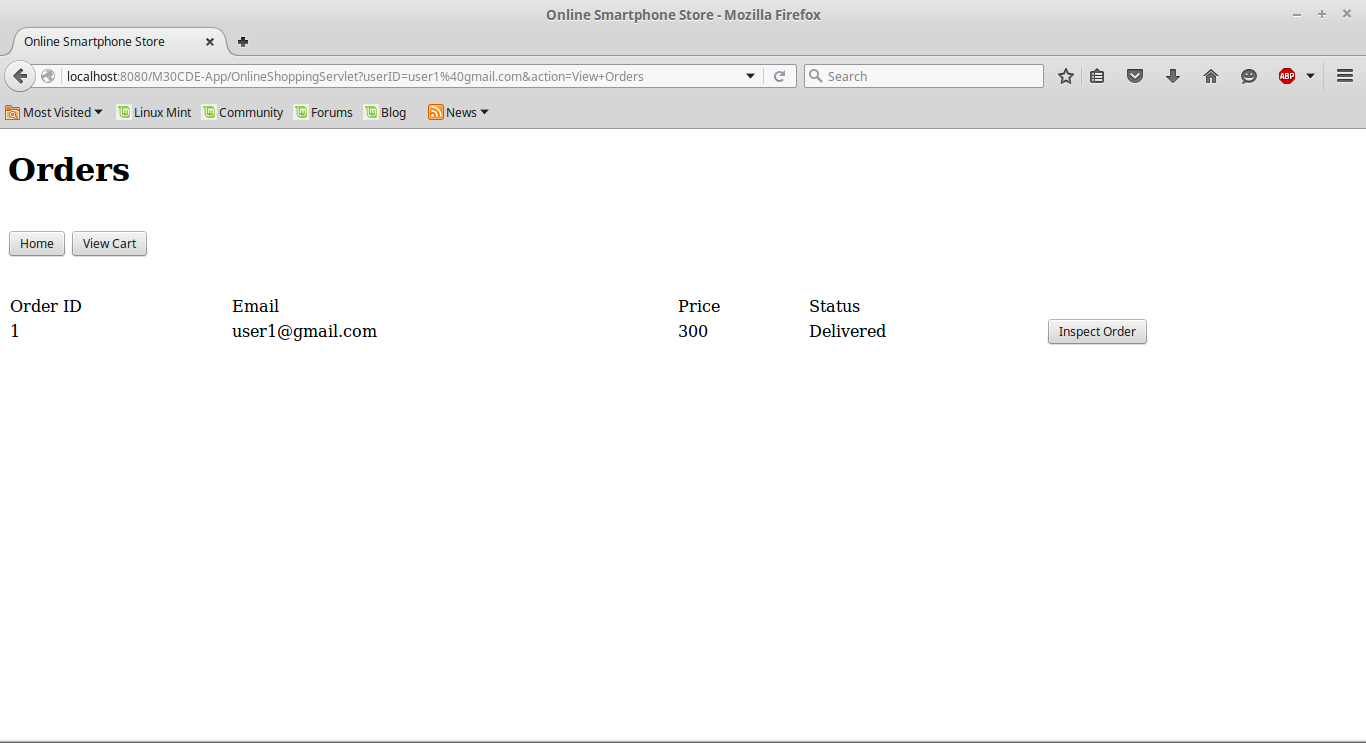


Figure 22. This picture shows how a logged on customer can view old orders made from the account.

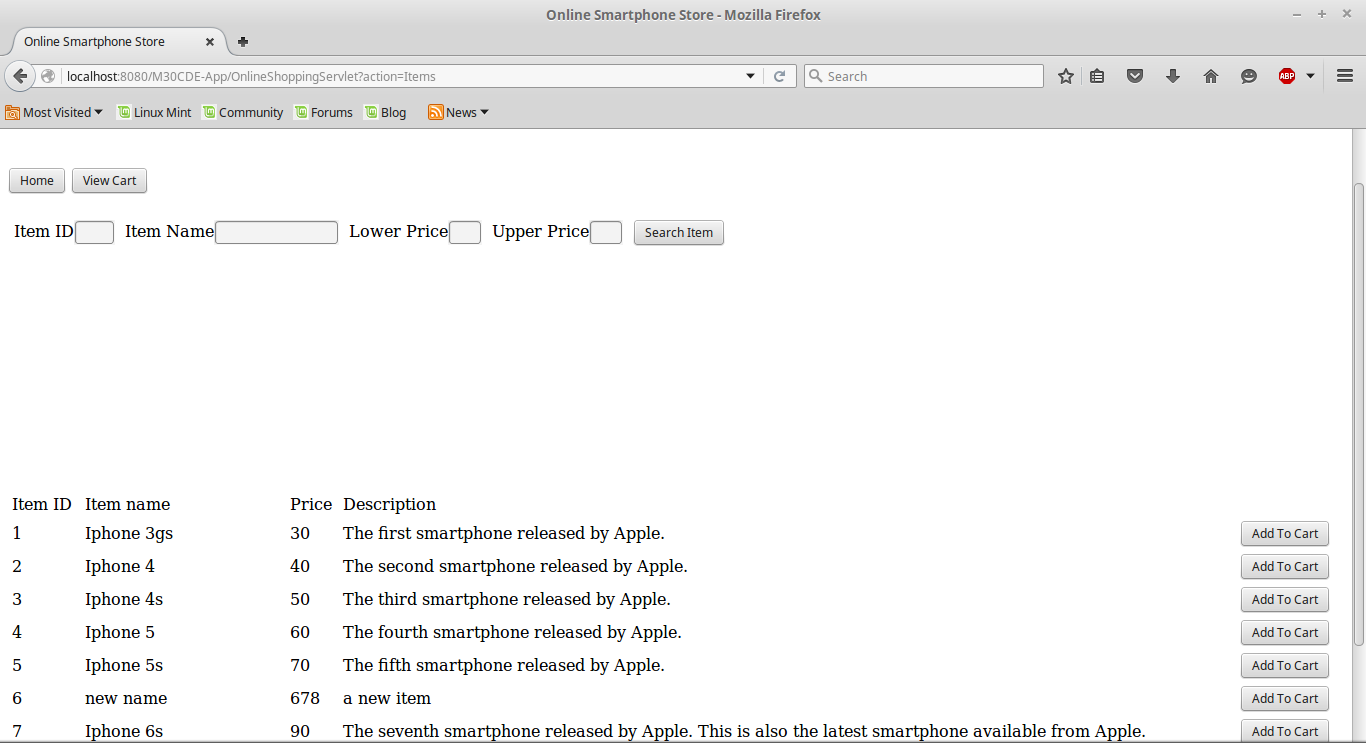


Figure 23. The items page for users and logged on customers. Both can add an item to their cart.

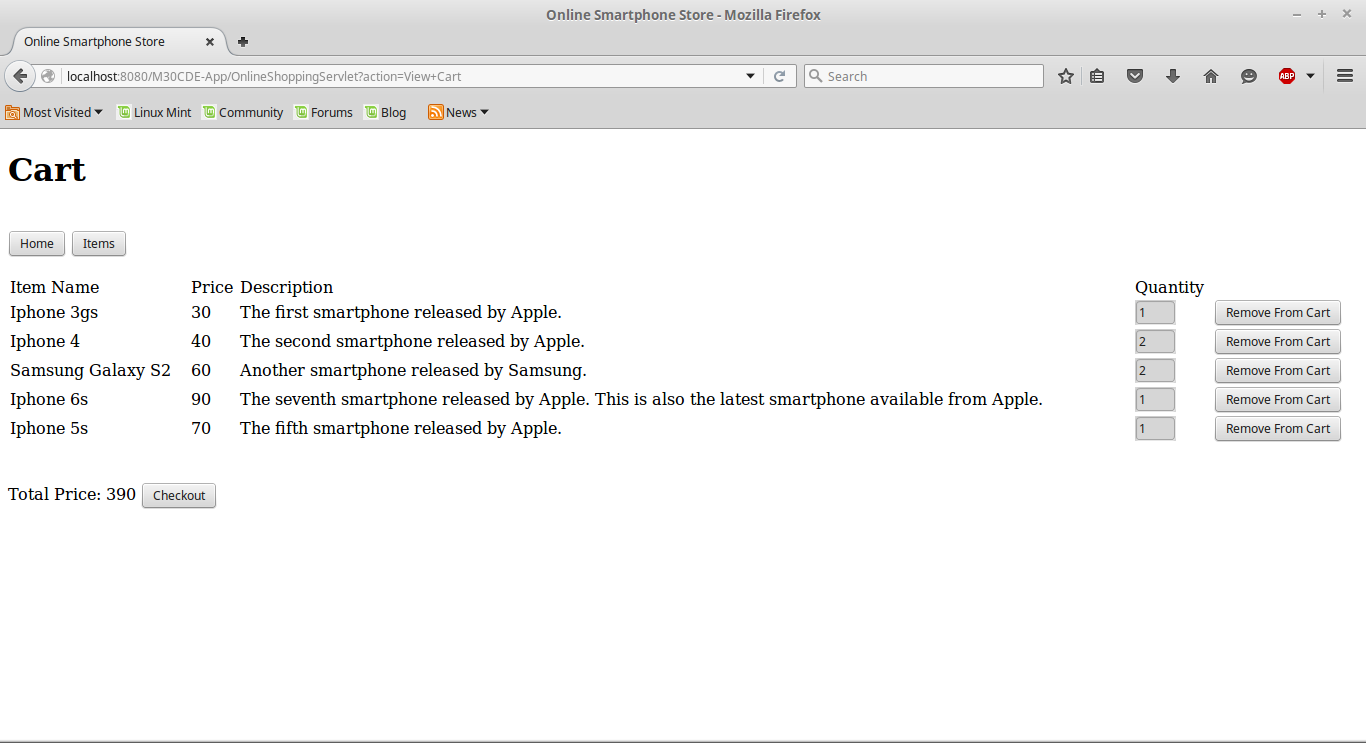


Figure 24. The cart page for users and customers. Both can remove items from the cart but only customers can checkout the cart items and make an order.

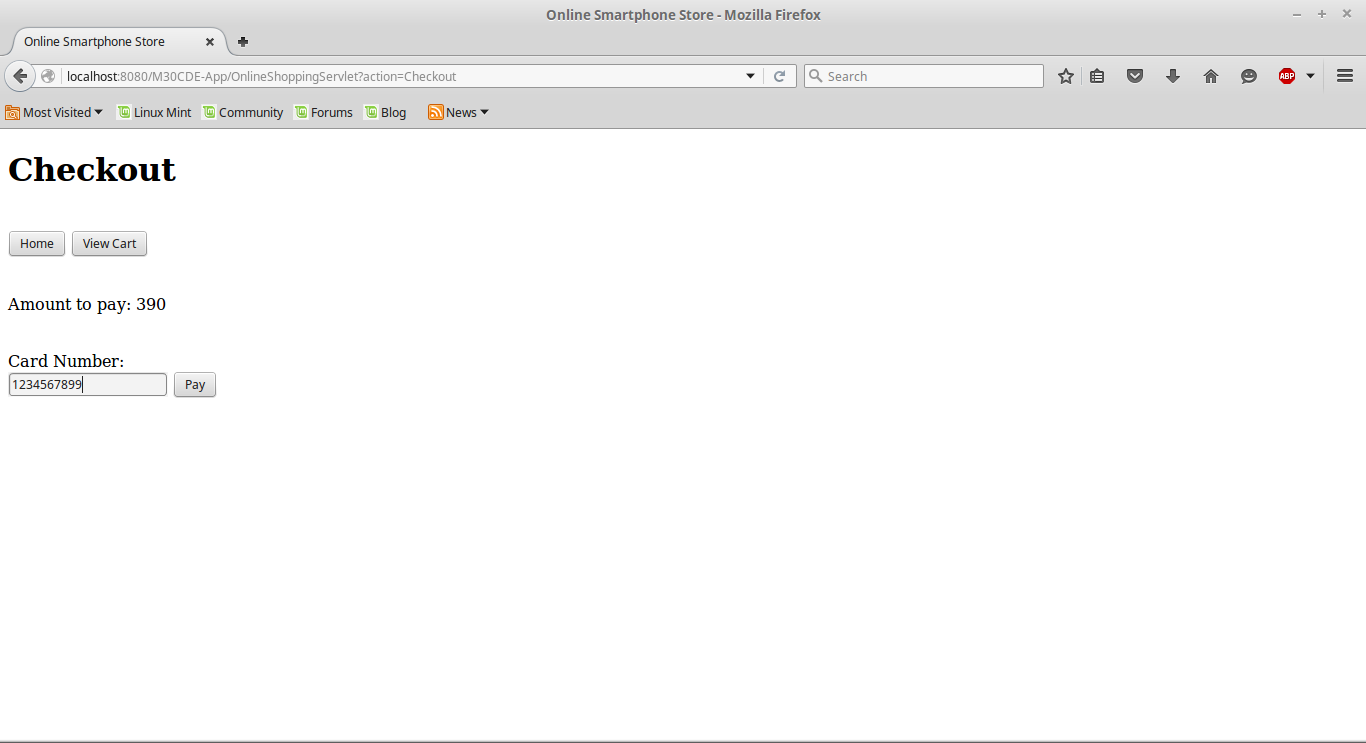


Figure 25. The checkout page. Customers has to enter a 10 digit card number to successfully make the order.

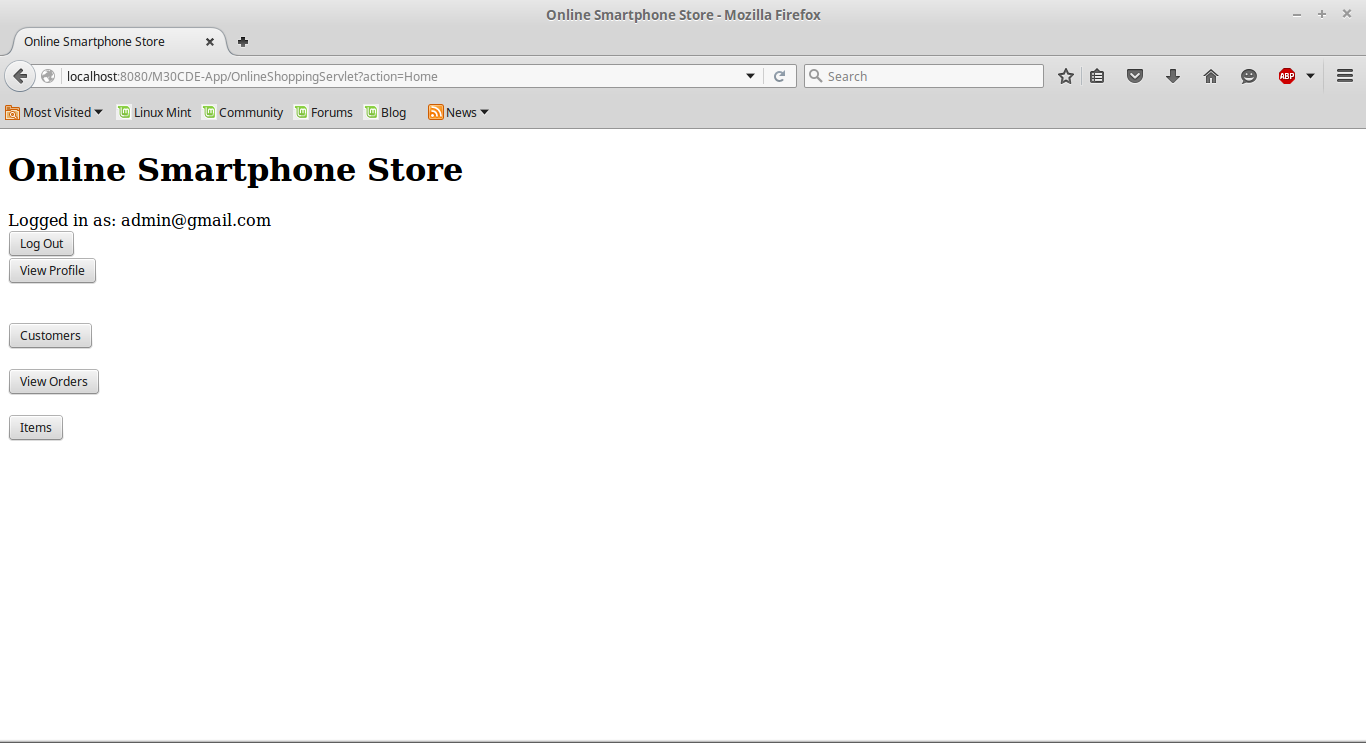


Figure 26. The start page for all the admins. Admins can view all registered customers and all placed orders from this page.

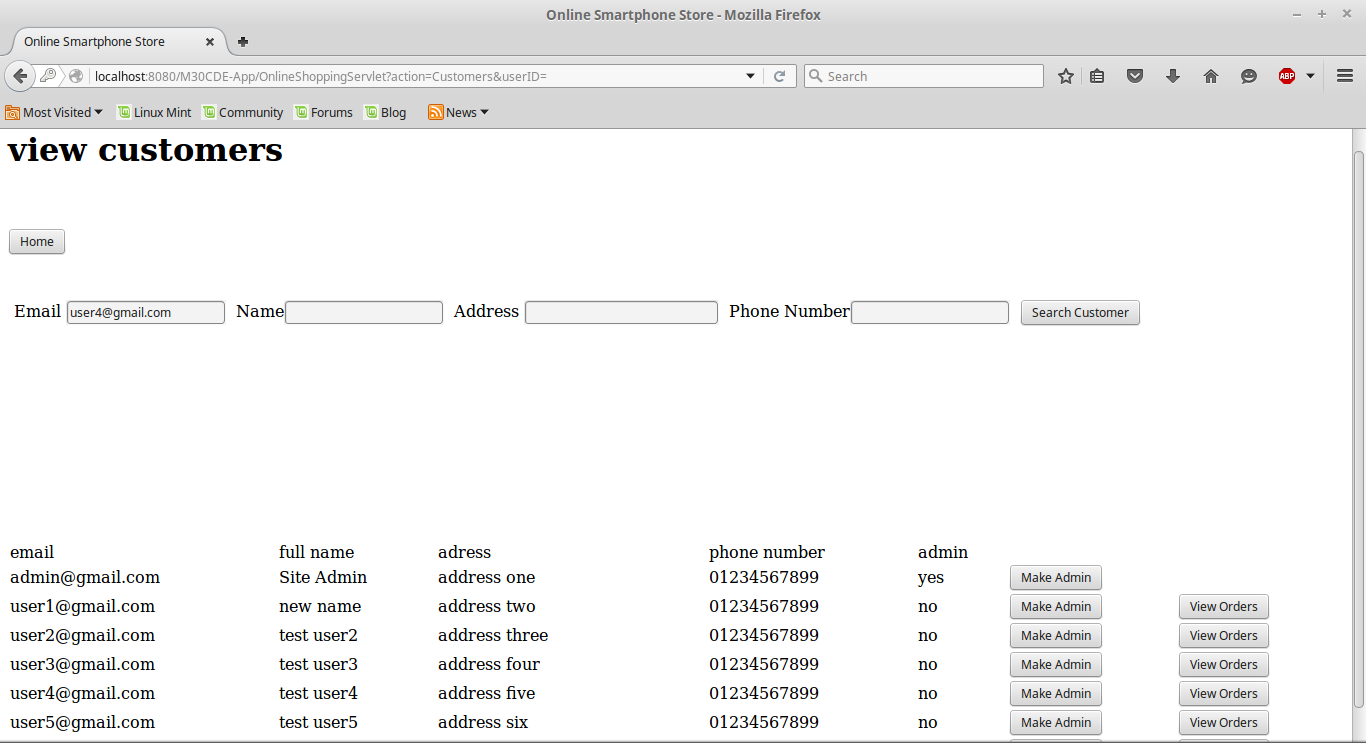


Figure 27. This page will display all the customers and their details for the admins. Admins can also make a certain customer into an admin or view all orders made by a certain customer.

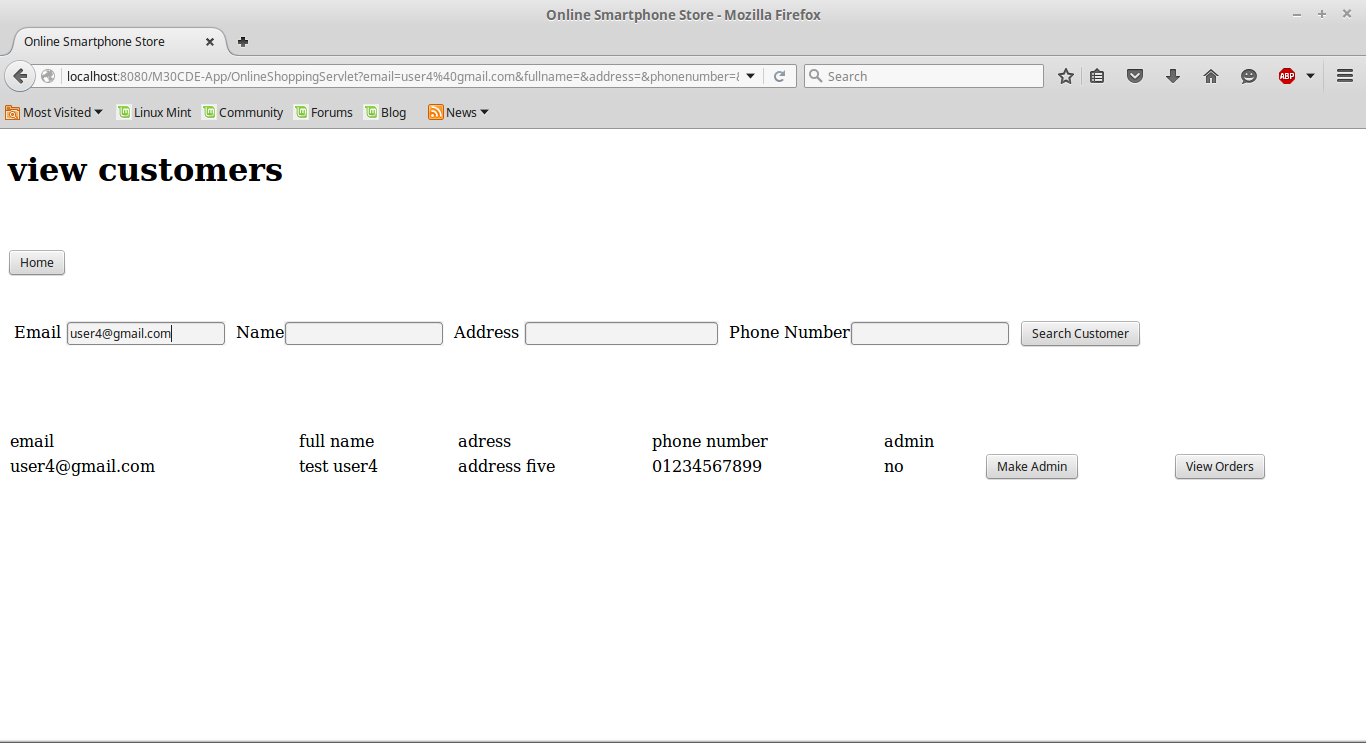


Figure 28. This is the result of searching for a specific customer in the page where admins can see all customers. Please note the search fields on top of figure 27.

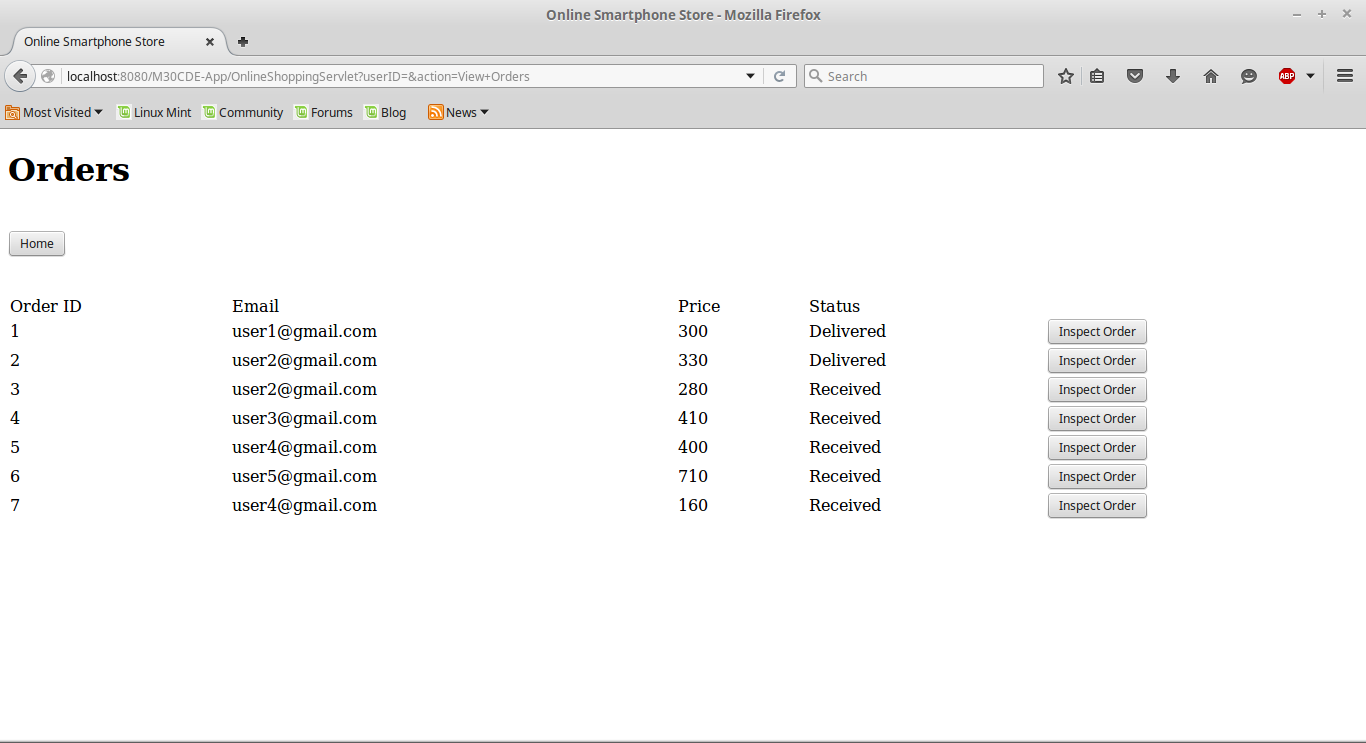


Figure 29. All the placed orders that only the admins can see. Admins can also inspect an order to see what items it contains and confirm it’s order status.

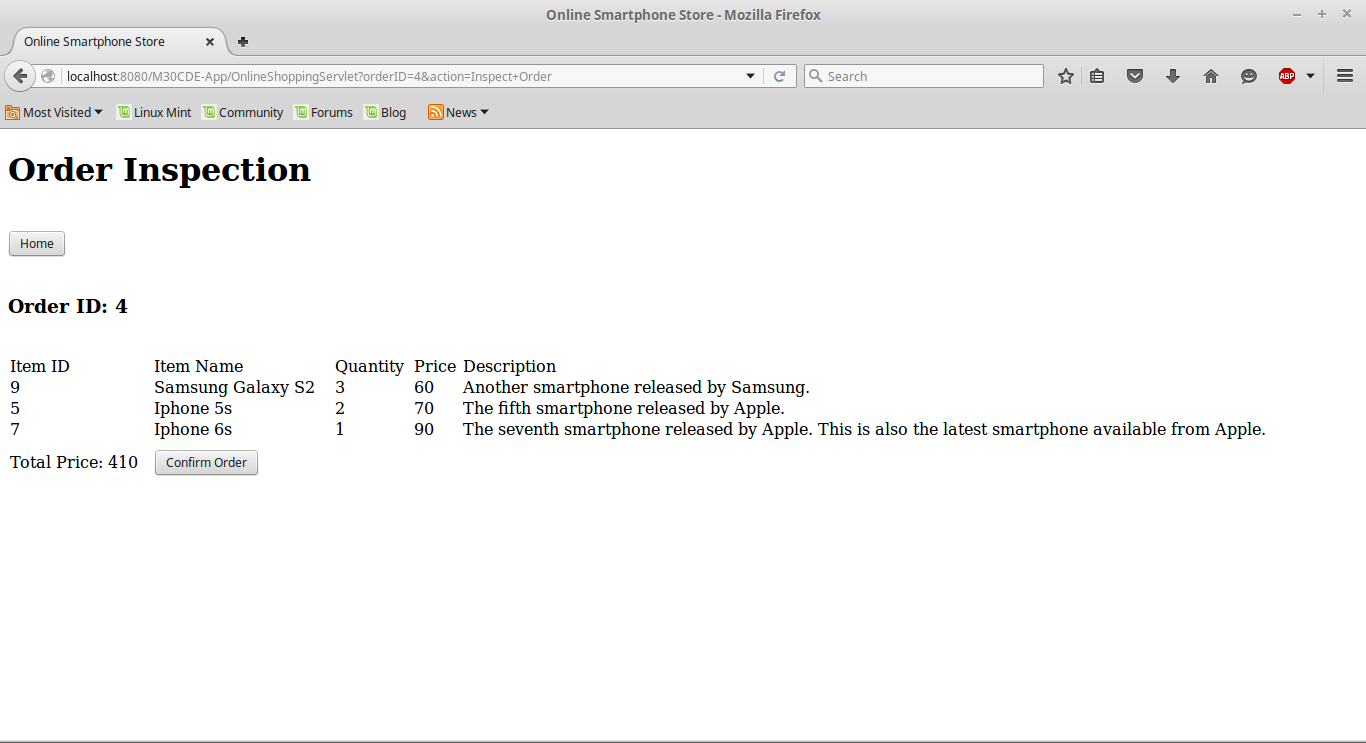


Figure 30. The order inspection page. Both admins and the customer can see this page but the confirm button (changes the order status) is only visible to the admins.

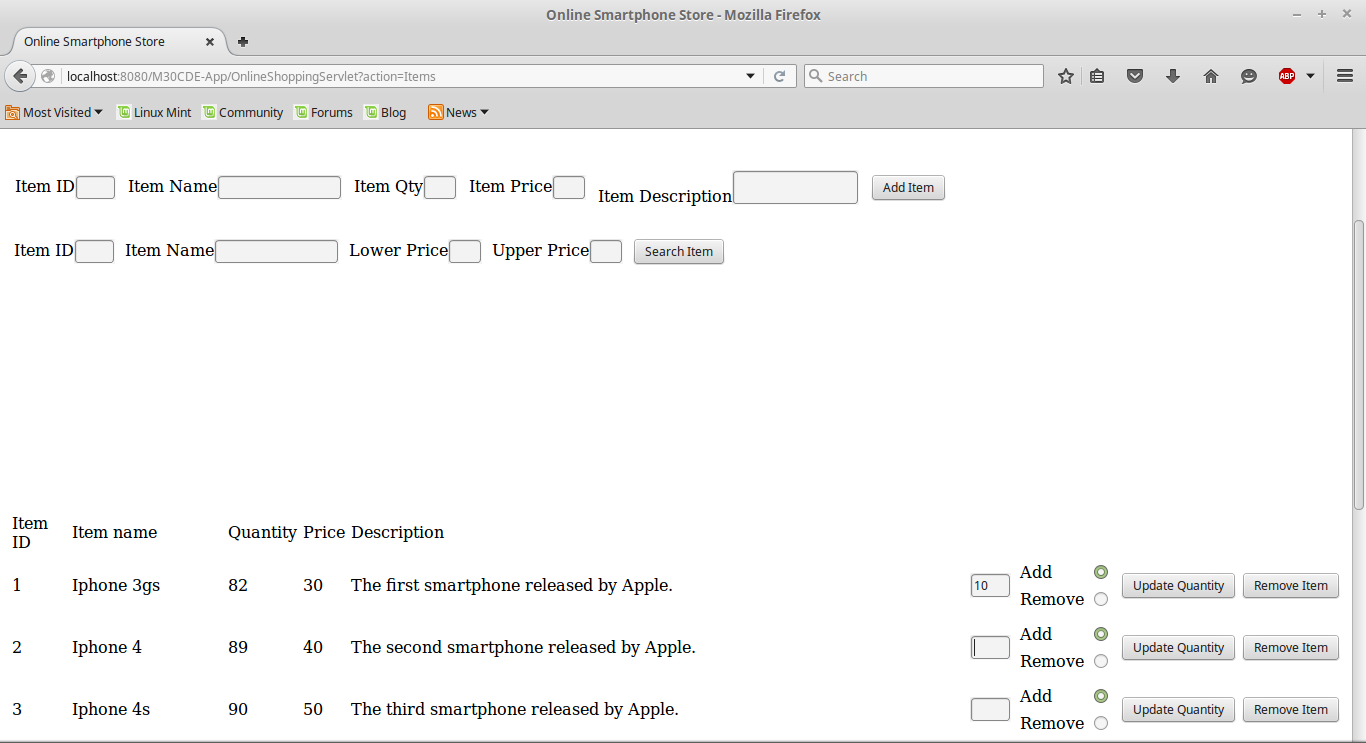


Figure 31. The items page for admins. Admins can add items and update the item quantities of existing items. Also delete existing items from the items page completely.

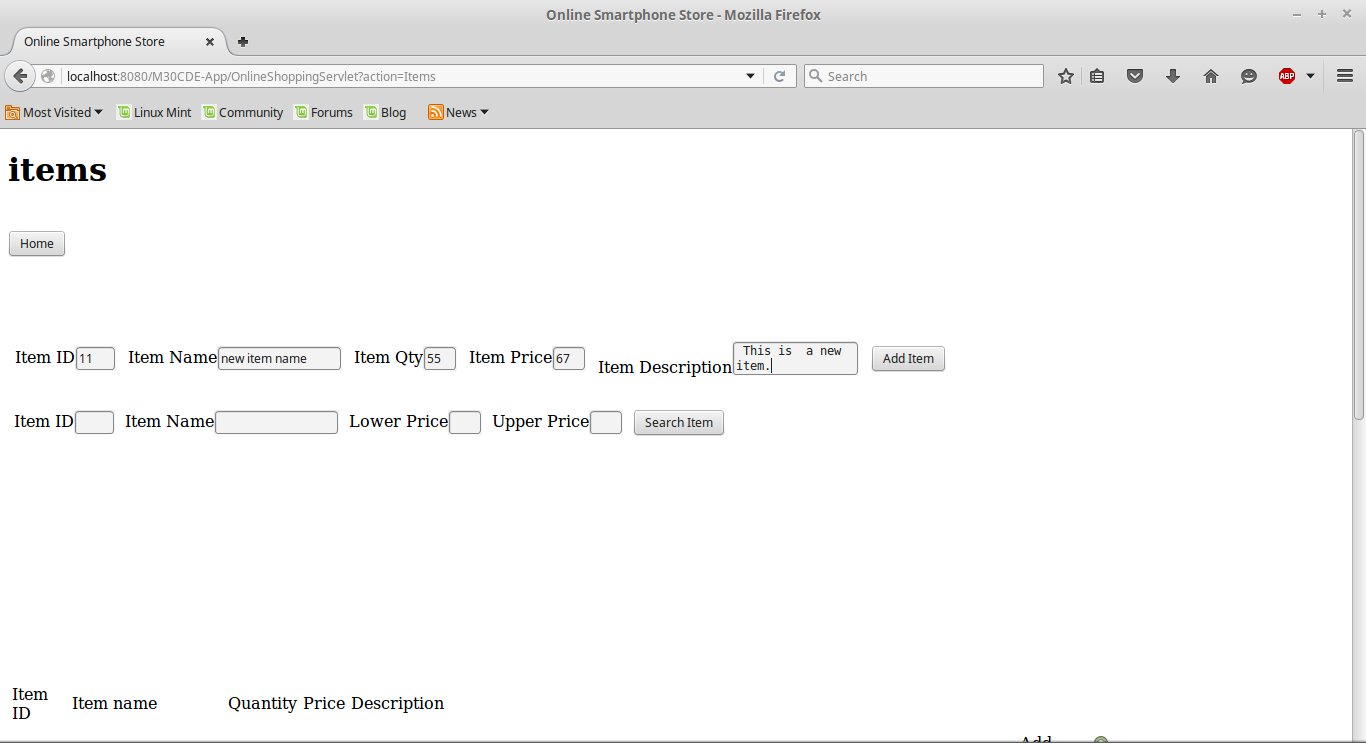


Figure 32. Example of how an item can be added to the items page by an admin.

## Test plan

Every page that needs a text input in order to continue will be tested. And also some other functions.

Test 1: Entering the wrong email or password in the login page then pressing the login button.

Test 2: Entering the correct email or password in the login page then pressing the login button.

Test 3: Register page, entering the wrong password in the password confirmbox, entering just a first name, entering an email address without “@” in it, entering a phone number with less than 11 digits.

Test 4: Entering everything the correct way in the register page.

Test 5: Items page, Search for a letter instead of valid item ID.

Test 6: Items page, search for unique item name.

Test 7: Items page, search for items within a specific price range.

Test 8: Cart page, add an item from the items page to the cart twice.

Test 9: Cart page, remove an item from the cart and control the total price.

Test 10: (Customers)Checkout page, enter a non-digit card number or card number with less than 10 digits.

Test 11: (Customers)Checkout page, enter valid card number containing 10 digits.

Test 12: (Admins)Customers page, Search for unique customer email.

Test 13: (Admins)Customers page, make a customer into Admin and back.

Test 14: (Admins)Items page, enter invalid info when adding new item to the items page. Item ID that is busy or that contains non-digits, item quantity non-digits, item price containing letters, any name and description.

Test 15: (Admins)Items page, enter everything correctly in the add item text boxes and add item.

Test 16: (Admins)Items page, enter a non-digit value when updating the quantity of an item.

Test 17: (Admins)Items page, enter a digit value when updating the item quantity of the item.

Test 18: (Admins)Items page, try removing an item from the items page completely.

Test 19: (Admins)Inspect order page, Confirm an order and check if the order status changed.

Test 20: Edit profile page, enter invalid info when updating the profile info. Enter the wrong old password, enter the wrong confirmation password, enter a only a first name as new name, enter new phone number with non-digits or less than 11 digits.

Test 21: Edit profile page, enter correct info when updating the profile info.

## Test results

Test 1: Nothing will happened, the user will stay on the login page until the login is successful or other buttons are pressed.

Test 2: The user will be successfully logged in and redirected to the customer start page (wireframe figure 4).

Test 3: The following error message will appear (in red text) next to relevant textbox on the register page. “Both first and last name is required!”,”Not a valid email address!”, “Not a valid phonenumber! Enter 11 digits”, “Passwords do not match!”, “Passwords do not match!”.

Test 4: The user will be redirected to the user start page and can start using the created customer account.

Test 5: Don’t show any results at all, no items with a letter as item ID exists.

Test 6: Show only the item that the user searched for.

Test 7: Show every item that costs as much as or more/less then the lower/upper price limit.

Test 8: The item that was added should appear in the cart on one row only and the item quantity should be 2.

Test 9: The item quantity in the cart should decrease by one if there was more than one of the item. The item should disappear from the cart if the quantity was 1. The total price displayed should be subtracted by the item price.

Test 10: A red error message will tell the customer to enter a valid 10 digit card number.

Test 11: The cart should now be empty and the costumer should be able to see a new order in his/her profile page.

Test 12: Show the customer that was searched for, if he/she exists.

Test 13: The customer that was made into an admin should no longer have any old orders. The admin should not be able to view orders made by this new admin either. The same “make admin” button should turn the new admin back to customer again if pressed twice, the old “view orders” button will reappear so that the admin can view the orders made by this customer.

Test 14: The following error messages will appear (in red text) above the “add item” text fields. “Item ID already in use!” or “Invalid item ID!”, “Invalid item quantity!”, Invalid item price!”.

Test 15: The item will appear among the other items. If the enter quantity is zero the item will only be visible to the admins until the item quantity is above 0.

Test 16: Nothing will happened, only a digit value will activate the update process.

Test 17: The item quantity will change, the new value can become negative if the original amount is smaller than the update amount and the update choice is to remove. The item will still be visible for the Admin in this case.

Test 18: The item should not be visible in the items page anymore, even for the Admin.

Test 19: The Admin should be redirected to the page displaying all the orders (wireframe figure 15) and see that the order status changed. Either Received -> Delivered or the other way around.

Test 20: The following errors should be displayed (in red text) net to relevant text box. “Wrong password!”, “Passwords does not match!”, ”Passwords does not match!”, “Both first and last name are required!”, “Invalid phone number! Enter 11 digits.”. Text boxes left blank will not be updated to the profile info, the update will not be completed until all errors disappear.

Test 21: The user will be redirected to the customer/admin start page, depending on which kind of user the current user is.