ONLINE WEB APPLICATION J C PENNY

Submitted to

Prof.Yuan Hong

Mohan Babu Jayaprakash	Sabareesh Suresh
mjayaprakash@hawk.iit.edu	ssuresh14@hawk.iit.edu
A20405567	A20396634

Sharanheer Chhogalal Choudhari
schoudhari@hawk.iit.edu
A20398615

FACT FINDING TECHNIQUES AND INFORMATION GATHERING

LEARNING AND PERSONAL EXPERIENCE:

We have all used online shopping for purchasing our customized products in recent days and this experience gives everyone a basic understanding of how the web application operates. This includes inventory, check-out, product pricing, customer records, store details etc. Also, in this project **Sabareesh Suresh** has created the datasets using MySQL, establishing the Entity Relationship diagram, breaking down the E-R diagram to relational Schema based on the requirement goals of the project as discussed in Lecture slides and Discussion forums. **Shareenheer Chowdhari** has worked on the Java Servlets incorporating the functionalities of product pricing, checkout actions and Inventory control. **Mohan Babu Jayaprakash** has established the functionalities of Customer records, Store information and relation updates. Finally, the front-end GUI to be more user friendly and customer centric has been implemented by **Mohan, Sabareesh, Sharanheer** figuring out the modelling of Login page, selection of products, adding and removing product details, payment methods and checkout action with the help of Tomcat and Eclipse tools, based on the individual functionalities by establishing the relationship for each entity for the online web application "**J C Penny**". This overall prototype will qualify our team to build an online web application for the database system.

ENTERPRISE RESEARCH:

Types of data needed to keep track of in this system will be the information, checkout activity, inventory control, employee information, and customer activity and user registration. Products information are critical to an online electronic store it adds/removes multiple number of products each day from its inventory. Data about products which need to be incorporated into the design include: UPC (manufacture's code), ID, brand, description, price, cost, weight, shape, size and if it is taxable. This fictitious Electronic store has multiple stores. Each store will have a record in the 'store' entity including a unique ID and address. Inventory will have a store ID component and products information. This will allow an employee to see the list of products in the store, the quantity, and the value of each product in the store inventory.

Customer information is critical to any successful business and an online business is no exception. Each customer is assumed to enroll in the Frequent shopper program by registering with his email id. The customers will have their information stored within the store's database and the information will be linked to their purchases. Employees can view reports at which store they purchased, item purchased, number of transactions and they money spent on the purchases. Customers can select products for checkout and they also need to have their purchases subtotaled and taxes added so they know what they will have to pay. After the transaction is complete, a receipt should be displayed. For this project, there will be a web form a user, customer, can choose products and add them to their checkout basket. Once the 'checkout' button is pressed, the products will be deducted from inventory. There will be no actual commerce mechanism in this mock up. Employees work at a store. The employees will have a unique ID. Each Employee working in a store has the access to update the inventory for that particular store once the product gets deducted after each purchase.

PROJECT AND DATABASE SCOPE:

This project will focus on small aspect of an electronic enterprise of a Web Application simulation of a customer buying products by registering through a Login form. They can see the list of products displayed once the user logs in. It's the customer's decision on what where and how many to purchase. The transaction is complete once the customer checkouts with his card details and payment. The project removes the products once purchased from the inventory. Employees can login to use the system from a admin point of view and perform tasks of a store employee like reports on inventory, customer activity, inventory list and quantities, storage capacity from data stored in the tables.

USER:

This entity type represents all the people that shop at the grocery store. The user type is defined either to be an "Employee" or a "Customer". A customer performs a checkout. The CUSTOMER entity relates to the CHECKOUT table via the BUYS PRODUCTS relationship. The "Cust_ID" primary key is a foreign key in the checkout table.

CHECKOUT

The checkout is tagged to a unique Checkout_id**.** This entity type represents an atomic transaction of a customer purchasing products in the store. It relates to CUSTOMERS via the BUYPRODUCTS relationship. It is connected to the PRODUCTS entity through the CHECKOUT ACTION relationship. This relationship will become a table using the Primary Key from CHECKOUT and PRODUCTS to join every product on each individual checkout transaction. CHECKOUT needs a ‘subtotal’ entity as it is calculated at the time of purchase with those specific products prices.

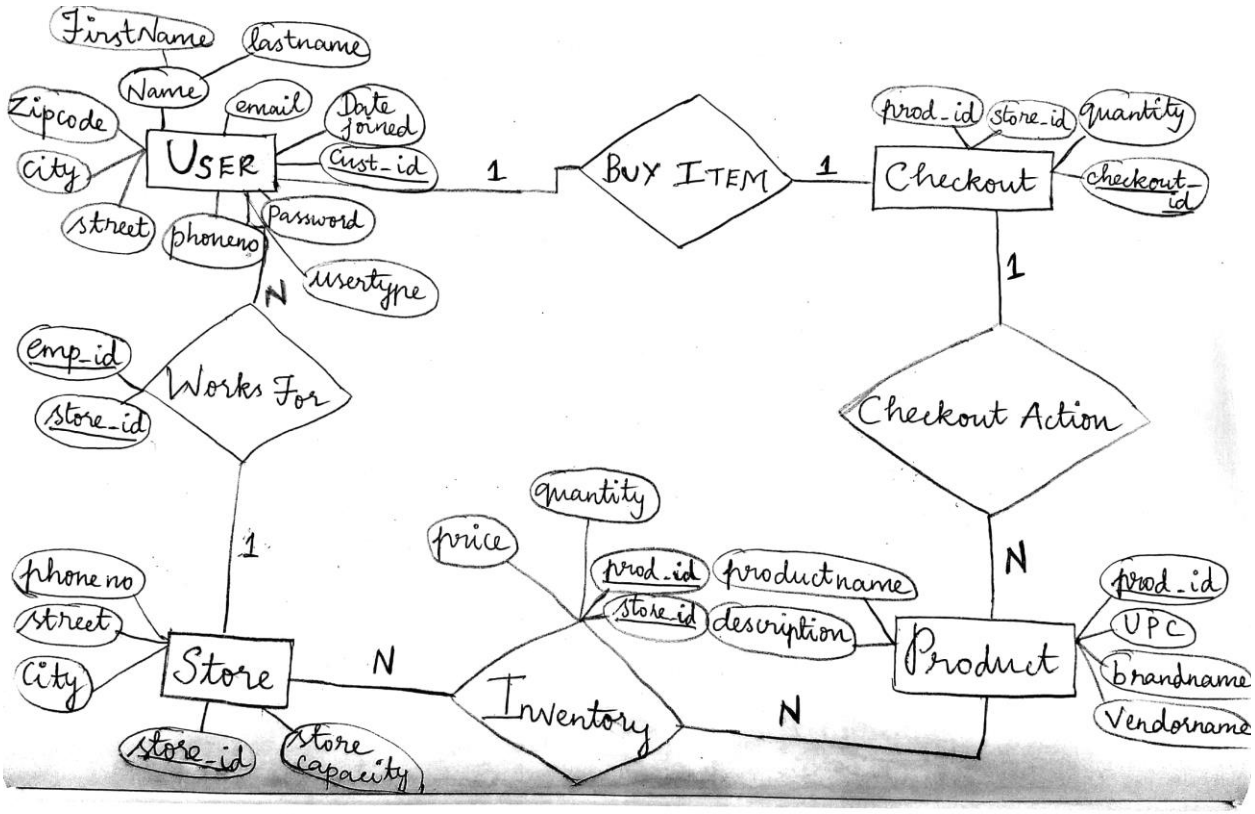
PRODUCTS

Each product is uniquely identified with a Productid**.** In this relation each product can be easily related with its Brand name, Vendor name and with the product description. This entity type represents the individual store products like Headphones, Laptops, Speakers etc., which the customers can purchase. PRODUCTS is related to CHECKOUT as described above. This entity type represents how many and what type of products are in a checkout. It relates to CHECKOUT via the CHECKOUT ACTION relationship. It is also related to STORES through the INVENTORY relationship. This relationship passes on information from the store where the transaction is process to the receipt.

STORE

This entity type represents the relation between the number of products each store has, the store location and address, based on which customers can decide which store they want to purchase from. The Store_ID primary key is used as a foreign key in multiple tables.

ER Diagram



RELATIONSHIP SET DESCRIPTION

BUY PRODUCTS

This relationship connects the CUSTOMERS entity to the CHECKOUT entity and is a binary relationship. Many customers can do many checkouts but each checkout has one customer assigned. It’s possible a customer has no checkouts yet if they just signed up. Therefore there is a 0..N mapping cardinality between CUSTOMERS:CHECKOUT. There is partial participation between the CUSTOMERS entity and the BUY PRODUCTS relationship and full participation between the CHECKOUT entity and the BUY PRODUCTS relationship. The multiplicity of CUSTOMERS in BUY PRODUCTS is 0..N. The multiplicity of CHECKOUT in BUY PRODUCTS is 1..N.

CHECKOUT ACTION

The CHECKOUT ACTION relationship is a binary relationship between the CHECKOUT entity and the PRODUCTS entity. A checkout transaction can contain many separate products but each products is only contained once in a specific checkout. The multiplicity between the CHECKOUT entity and the CHECKOUT_PRODUCTS entity is 1..N with full participation. The multiplicity between the PRODUCTS entity and the CHECKOUT_PRODUCTS entity is 1..N also. The ER multiplicity between the entities CHECKOUT and PRODUCTS is N..N as a checkout can contain many products and many checkouts can contain the same products.

INVENTORY

The INVENTORY relationship is a binary relationship. It connects the STORE entity with the PRODUCTS entity. Many products are contained in a store and each products will be in multiple stores. Therefore, the ER multiplicity between the entities STORE and PRODUCTS is N..N. This relationship is constructed by the combination of the PK of STORE entity (Store_ID) and the PK of PRODUCTS entity (product_ID). This PK merger is represented as a new INVENTORY entity. The multiplicity between the STORE entity and the INVENTORY entity is 1..N with full participation. The multiplicity between the PRODUCTS entity and the INVENTORY entity is 1..N also. There are two attributes added to this STORE CONTAINER relationship, ‘quantity’ and ‘price’. These two attributes and the two PK’s will be combined in the relational DB model to form the INVENTORY table.

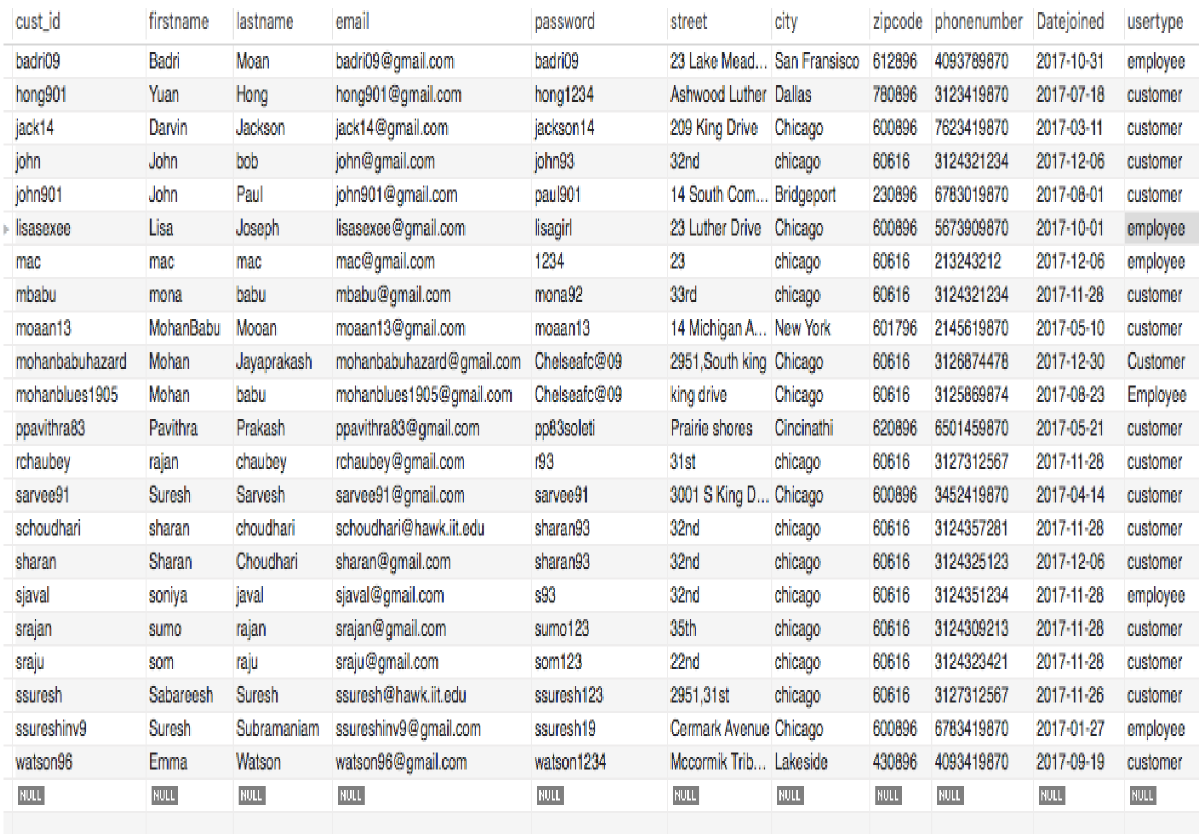
WORKS FOR:

The WORKS FOR relationship describes the interaction between the EMPLOYEE and STORE entity types. It is a binary relationship type. Many employees can work for one store. Every employee must work for a store, but only one store. Therefore there is total participation for the each side of the WORKS FOR relationship. The mapping cardinality of EMPLOYEE : STORE is N:0. The multiplicity of EMPLOYEE in WORKS FOR is 1..N as each employee must be at a store and more than one could be. The multiplicity of STORE in WORKS FOR is 0..N as a STORE could be new and have no employees initially assigned.

SQL SCRIPTS :

CUSTOMER :

This is the script used to create the Customer relation whenever a new user i.e. a new customer or an Employee try to register for the J C Penny Enterprise for the first time. Every user is generated with a unique id and a password to login. Once the user completed his check-in form then the has successfully registered and then the user is directed to the respective web page depending on the user type.

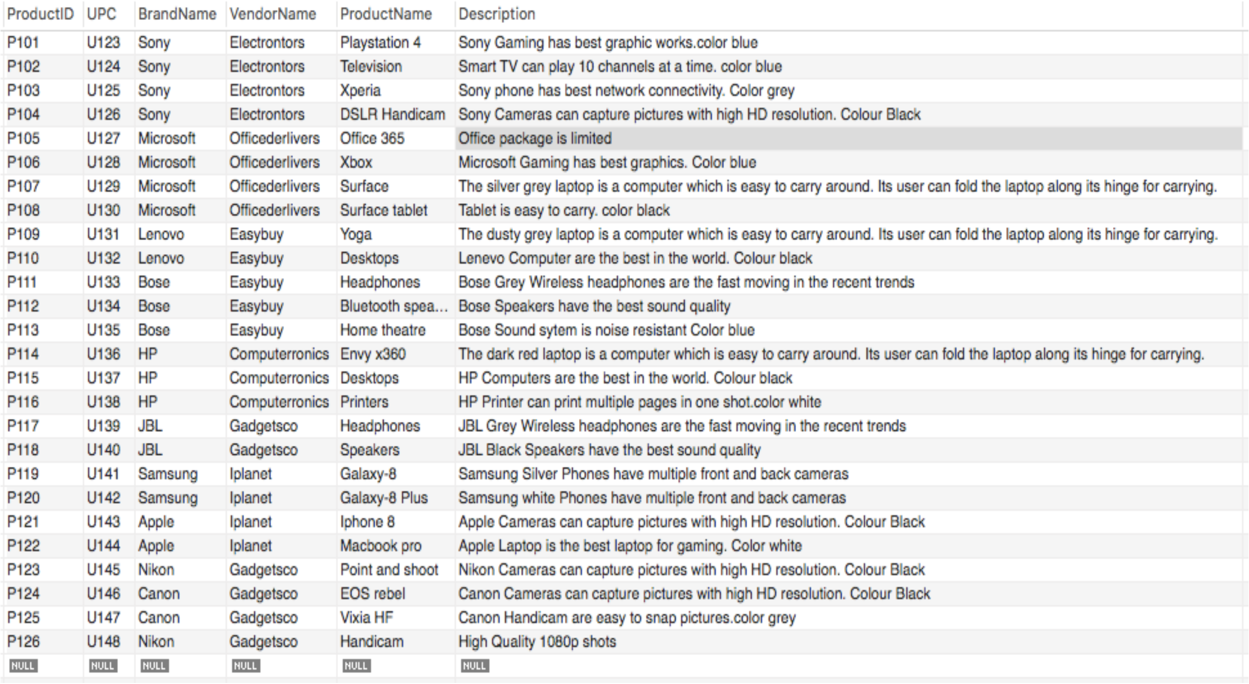


cust_id	firstname	lastname	email	password	street	city	zipcode	phonenumber	Datejoined	usertype
badri09	Badri	Moan	badri09@gmail.com	badri09	23 Lake Mead...	San Fransisco	612896	4093789870	2017-10-31	employee
hong901	Yuan	Hong	hong901@gmail.com	hong1234	Ashwood Luther	Dallas	780896	3123419870	2017-07-18	customer
jack14	Darvin	Jackson	jack14@gmail.com	jackson14	209 King Drive	Chicago	600896	7623419870	2017-03-11	customer
john	John	bob	john@gmail.com	john93	32nd	chicago	60616	3124321234	2017-12-06	customer
john901	John	Paul	john901@gmail.com	paul901	14 South Com...	Bridgeport	230896	6783019870	2017-08-01	customer
lisasexee	Lisa	Joseph	lisasexee@gmail.com	lisagirl	23 Luther Drive	Chicago	600896	5673909870	2017-10-01	employee
mac	mac	mac	mac@gmail.com	1234	23	chicago	60616	213243212	2017-12-06	employee
mbabu	mona	babu	mbabu@gmail.com	mona92	33rd	chicago	60616	3124321234	2017-11-28	customer
moaan13	MohanBabu	Mooan	moaan13@gmail.com	moaan13	14 Michigan A...	New York	601796	2145619870	2017-05-10	customer
mohanbabuhazard	Mohan	Jayaprakash	mohanbabuhazard@gmail.com	Chelseafc@09	2951,South king	Chicago	60616	3126874478	2017-12-30	Customer
mohanblues1905	Mohan	babu	mohanblues1905@gmail.com	Chelseafc@09	king drive	Chicago	60616	3125869874	2017-08-23	Employee
ppavithra83	Pavithra	Prakash	ppavithra83@gmail.com	pp83soleti	Prairie shores	Cincinathi	620896	6501459870	2017-05-21	customer
rchaubey	rajan	chaubey	rchaubey@gmail.com	r93	31st	chicago	60616	3127312567	2017-11-28	customer
sarvee91	Suresh	Sarvesh	sarvee91@gmail.com	sarvee91	3001 S King D...	Chicago	600896	3452419870	2017-04-14	customer
schoudhari	sharan	choudhari	schoudhari@hawk.iit.edu	sharan93	32nd	chicago	60616	3124357281	2017-11-28	customer
sharan	Sharan	Choudhari	sharan@gmail.com	sharan93	32nd	chicago	60616	3124325123	2017-12-06	customer
sjaval	soniya	javal	sjaval@gmail.com	s93	32nd	chicago	60616	3124351234	2017-11-28	employee
srajan	sumo	rajan	srajan@gmail.com	sumo123	35th	chicago	60616	3124309213	2017-11-28	customer
sraju	som	raju	sraju@gmail.com	som123	22nd	chicago	60616	3124323421	2017-11-28	customer
ssuresh	Sabareesh	Suresh	ssuresh@hawk.iit.edu	ssuresh123	2951,31st	chicago	60616	3127312567	2017-11-26	customer
ssureshinv9	Suresh	Subramaniam	ssureshinv9@gmail.com	ssuresh19	Cermark Avenue	Chicago	600896	6783419870	2017-01-27	employee
watson96	Emma	Watson	watson96@gmail.com	watson1234	Mccormik Trib...	Lakeside	430896	4093419870	2017-09-19	customer
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Primary Key : cust_id
Weak/Strong : Strong

PRODUCT DETAILS :

The below script depicts the Product name, its respective brand, UPC (the bar code for each product), Vendor name and Product details and its description. Every product is uniquely identified by its ProductID.

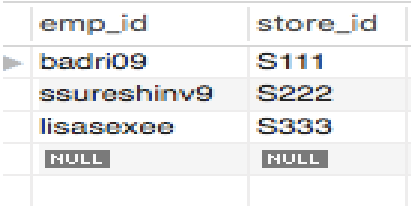


ProductID	UPC	BrandName	VendorName	ProductName	Description
P101	U123	Sony	Electrontors	Playstation 4	Sony Gaming has best graphic works.color blue
P102	U124	Sony	Electrontors	Television	Smart TV can play 10 channels at a time. color blue
P103	U125	Sony	Electrontors	Xperia	Sony phone has best network connectivity. Color grey
P104	U126	Sony	Electrontors	DSLR Handicam	Sony Cameras can capture pictures with high HD resolution. Colour Black
P105	U127	Microsoft	Officederlivers	Office 365	Office package is limited
P106	U128	Microsoft	Officederlivers	Xbox	Microsoft Gaming has best graphics. Color blue
P107	U129	Microsoft	Officederlivers	Surface	The silver grey laptop is a computer which is easy to carry around. Its user can fold the laptop along its hinge for carrying.
P108	U130	Microsoft	Officederlivers	Surface tablet	Tablet is easy to carry. color black
P109	U131	Lenovo	Easybuy	Yoga	The dusty grey laptop is a computer which is easy to carry around. Its user can fold the laptop along its hinge for carrying.
P110	U132	Lenovo	Easybuy	Desktops	Lenevo Computer are the best in the world. Colour black
P111	U133	Bose	Easybuy	Headphones	Bose Grey Wireless headphones are the fast moving in the recent trends
P112	U134	Bose	Easybuy	Bluetooth spea...	Bose Speakers have the best sound quality
P113	U135	Bose	Easybuy	Home theatre	Bose Sound sytem is noise resistant Color blue
P114	U136	HP	Computerronics	Envy x360	The dark red laptop is a computer which is easy to carry around. Its user can fold the laptop along its hinge for carrying.
P115	U137	HP	Computerronics	Desktops	HP Computers are the best in the world. Colour black
P116	U138	HP	Computerronics	Printers	HP Printer can print multiple pages in one shot.color white
P117	U139	JBL	Gadgetsco	Headphones	JBL Grey Wireless headphones are the fast moving in the recent trends
P118	U140	JBL	Gadgetsco	Speakers	JBL Black Speakers have the best sound quality
P119	U141	Samsung	Iplanet	Galaxy-8	Samsung Silver Phones have multiple front and back cameras
P120	U142	Samsung	Iplanet	Galaxy-8 Plus	Samsung white Phones have multiple front and back cameras
P121	U143	Apple	Iplanet	Iphone 8	Apple Cameras can capture pictures with high HD resolution. Colour Black
P122	U144	Apple	Iplanet	Macbook pro	Apple Laptop is the best laptop for gaming. Color white
P123	U145	Nikon	Gadgetsco	Point and shoot	Nikon Cameras can capture pictures with high HD resolution. Colour Black
P124	U146	Canon	Gadgetsco	EOS rebel	Canon Cameras can capture pictures with high HD resolution. Colour Black
P125	U147	Canon	Gadgetsco	Vixia HF	Canon Handicam are easy to snap pictures.color grey
P126	U148	Nikon	Gadgetsco	Handicam	High Quality 1080p shots
NULL	NULL	NULL	NULL	NULL	NULL

Primary Key : ProductID, UPC
Strong/Weak : Strong

WORKS FOR

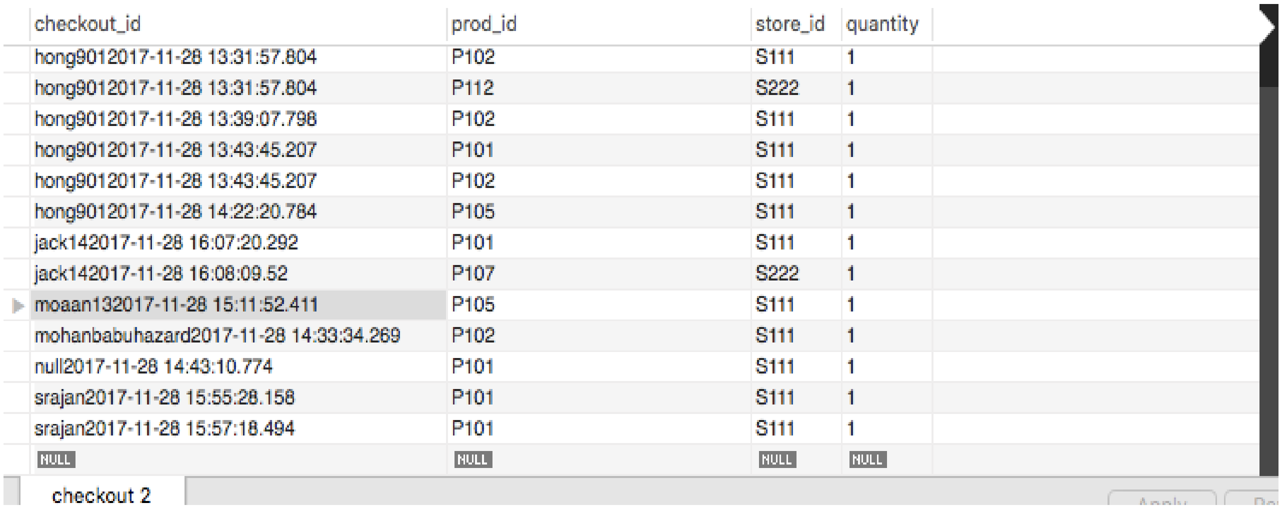
Primary Key : emp_id, store_id
Foreign Key : emp_id references Customer to cust_id, store_id references Store.



emp_id	store_id
badri09	S111
ssureshinv9	S222
lisasexee	S333
NULL	NULL

CHECKOUT

Once the user places his order in the cart and decides to checkout the user can verify the quantity, store_id, prod_id and then directed to the payment page once he completes his checkout. The checkout is uniquely identified by its checkoutid. Every user is generated with a unique checkout id and the timestamp is also generated each time a user checkouts.

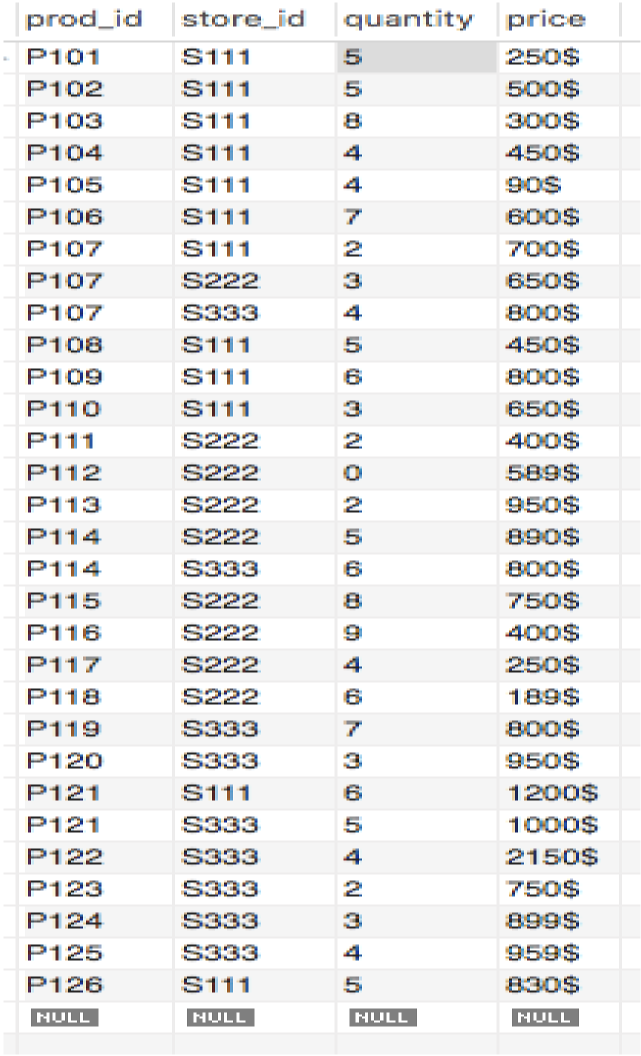


checkout_id	prod_id	store_id	quantity
hong9012017-11-28 13:31:57.804	P102	S111	1
hong9012017-11-28 13:31:57.804	P112	S222	1
hong9012017-11-28 13:39:07.798	P102	S111	1
hong9012017-11-28 13:43:45.207	P101	S111	1
hong9012017-11-28 13:43:45.207	P102	S111	1
hong9012017-11-28 14:22:20.784	P105	S111	1
jack142017-11-28 16:07:20.292	P101	S111	1
jack142017-11-28 16:08:09.52	P107	S222	1
moaan132017-11-28 15:11:52.411	P105	S111	1
mohanbabuhazard2017-11-28 14:33:34.269	P102	S111	1
null2017-11-28 14:43:10.774	P101	S111	1
srajan2017-11-28 15:55:28.158	P101	S111	1
srajan2017-11-28 15:57:18.494	P101	S111	1
NULL	NULL	NULL	NULL

Primary Key : checkout_id
Weak/Strong : Strong

INVENTORY

The inventory is an important relation to display the quantity of the number of products each store has and also the price of each quantity the user selected. The inventory is the relationship that is been established between the Store and the products. i.e. whenever an user checksout the product then the new product will be updated from the inventory to the respective store based on the store_id.

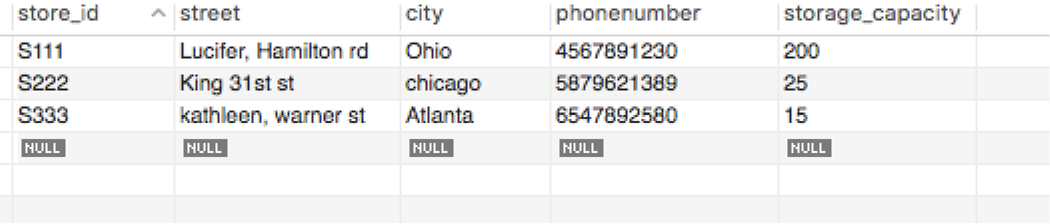


prod_id	store_id	quantity	price
P101	S111	5	250$
P102	S111	5	500$
P103	S111	8	300$
P104	S111	4	450$
P105	S111	4	90$
P106	S111	7	600$
P107	S111	2	700$
P107	S222	3	650$
P107	S333	4	800$
P108	S111	5	450$
P109	S111	6	800$
P110	S111	3	650$
P111	S222	2	400$
P112	S222	0	589$
P113	S222	2	950$
P114	S222	5	890$
P114	S333	6	800$
P115	S222	8	750$
P116	S222	9	400$
P117	S222	4	250$
P118	S222	6	189$
P119	S333	7	800$
P120	S333	3	950$
P121	S111	6	1200$
P121	S333	5	1000$
P122	S333	4	2150$
P123	S333	2	750$
P124	S333	3	899$
P125	S333	4	959$
P126	S111	5	830$
NULL	NULL	NULL	NULL

Primary Key : prod_id, store_id
Foreign Key : prod_id refernences product, store_id references store
Weak/Strong : Strong

STORE

Every Store is uniquely identified by the store_id. In our Database J C Penny could be in multiple locations and this below script makes the user easy to find the store near by their location based on the store address and the store phonenumber.



store_id	street	city	phonenumber	storage_capacity
S111	Lucifer, Hamilton rd	Ohio	4567891230	200
S222	King 31st st	chicago	5879621389	25
S333	kathleen, warner st	Atlanta	6547892580	15
NULL	NULL	NULL	NULL	NULL

Primary Key : store_id
Strong/Weak : Strong

CONVERSION FROM ER TO RELATIONAL MODEL

Step 1: Mapping of Regular Entity Types

For each regular (strong) entity type *E* in the ER schema, create a relation *R* that includes all the simple attributes of *E*. Include simple components of composite attributes. Choose one of *E*s key attributes to be the primary key of *R*. Such relations are sometimes called entity relations because each tuple represents an entity instance.

Step 2: Mapping of Weak Entity Types

For each weak entity type *W* in the ER schema with owner entity type *E*, create a relation *R* and include all simple attributes (or simple components of composite attributes) of *W* as attributes of *R*. Include as foreign key attributes of *R*, the primary key attribute(s) of the relation(s) that correspond to the owner entity type(s). If there is a weak entity type *E*2 whose owner is also a weak entity type *E*1, then *E*1 should be mapped before *E*2 to determine its primary key first.

Step 3: Mapping of Binary 1:1 Relationship Types

For each binary 1:1 relationship type *R* in the ER schema, identify the relations *S* and *T* that correspond to the entity types participating in *R*. There are three possible approaches: 1. the foreign key approach, 2. the merged relationship approach, and

3. the cross-reference or relationship relation approach. The first approach is the most useful.

A. Foreign key approach

Choose one of the relations (we'll say *S*) and include as a foreign key in *S* the primary key of *T*. It is better to pick an entity type with total participation in *R* in the role of *S*. Include all the simple attributes (or the simple components of composite attributes) of the 1:1 relationship type *R* as attributes of *S*.

B. Merged relation approach

This involves merging the two entity types and the relationship into a single relation. This may be appropriate when both participations are total.

C. Cross-reference or relationship relation approach

Set up a third relation *R* for the purpose of cross-referencing the primary keys of the two relations *S* and *T* representing the entity types. The relation *R* is called a relationship relation or lookup table, because each tuple in *R* represents a relationship instance that relates one tuple from *S* with one tuple of *T*.

Step 4: Mapping of Binary 1:N Relationship Types

For each regular binary 1:N relationship *R*, identify the relation *S* that represents the participating entity type at the *N-side* of the relationship type. Include as foreign key in *S* the primary key of the relation *T* that represents the other entity type participating in *R*. (Include attributes as done previously).

Step 5: Mapping of Binary M:N Relationship Types

For each binary M:N relationship type *R*, create a new relation *S* to represent *R*. Include as foreign key attributes in *S* the primary keys of the relations that represent the participating entity types. Their combination will form the primary key of *S*. Include in *S* all the attributes from the M:N relationship type.

Step 6: Mapping of Multivalued Attributes

For each multivalued attribute *A*, create a new relation *R*. This relation will include an attribute corresponding to *A*, plus the primary key attribute *K*—as a foreign key in *R*—of the relation that represents the entity type or relationship type that has *A* as an attribute. The primary key of *R* is the combination of *A* and *K*. If the multivalued attribute is a composite, we include its simple components.

Step 7: Mapping of *N*-ary Relationship Types

For each *n*-ary relationship type *R*, where *n* > 2, create a new relation *S* to represent *R*. Include as foreign key attributes in *S* the primary keys of the relations that represent the participating entity types. Also include any simple attributes of the *n*-ary relationship type (or simple components of composite attributes) as attributes of *S*. The primary key of *S* is usually a combination of all the foreign keys that reference the relations representing the participating entity types. However, if the cardinality constraints on any of the entity types *E* participating in *R* is 1, then the primary key of *S* should not include the foreign key attribute that references the relation *E*' corresponding to *E*.

Step 8: Options for Mapping Specialization or Generalization

Convert each specialization with *m* subclasses {*S1, S2,… ,Sm*} and (generalized superclass *C*, where the attributes of *C* are {*k, a1,…,an*} and *k* is the (primary) key, into relation schemas.

A. Create different relations (tables) for each sub class and one for the superclass.
B. Integrate the superclass into each ‘M’ relation so there are no more relationships.
C. Union all the attributes into one relation and add a indicator, like an integer, that specifies the type of entity it is. Obviously there would be nulls in cells that did not correspond to the type of class that entity was.
D. Similar to option ‘C’ but use binary flags to determine the entity type after you union all the attributes.

Step 9: Mapping of Categories (Union Types)

A category (or union type) is a subclass of the union of two or more superclasses that can have different keys because they can be of different entity types. For mapping a category whose defining superclasses have different keys, it is customary to specify a new key attribute, called asurrogate key, when creating the relation. Because the keys are different, we cannot use just one of them exclusively to identify all tuples in the relation. We can create a relation to correspond to the category, and include any attributes of the category in this relation. The primary key of the new relation is the surrogate key. We also include this surrogate key as a foreign key in each relation corresponding to a superclass of the category. For a category whose superclasses have the same key, there is no need for a surrogate key.

CONSTRAINTS

Entity Integrity Constraint: States that no primary key value can be NULL. This is because the primary key value is used to identify individual tuples in a relation. Having NULL values for the primary key implies that we cannot identify some tuples.

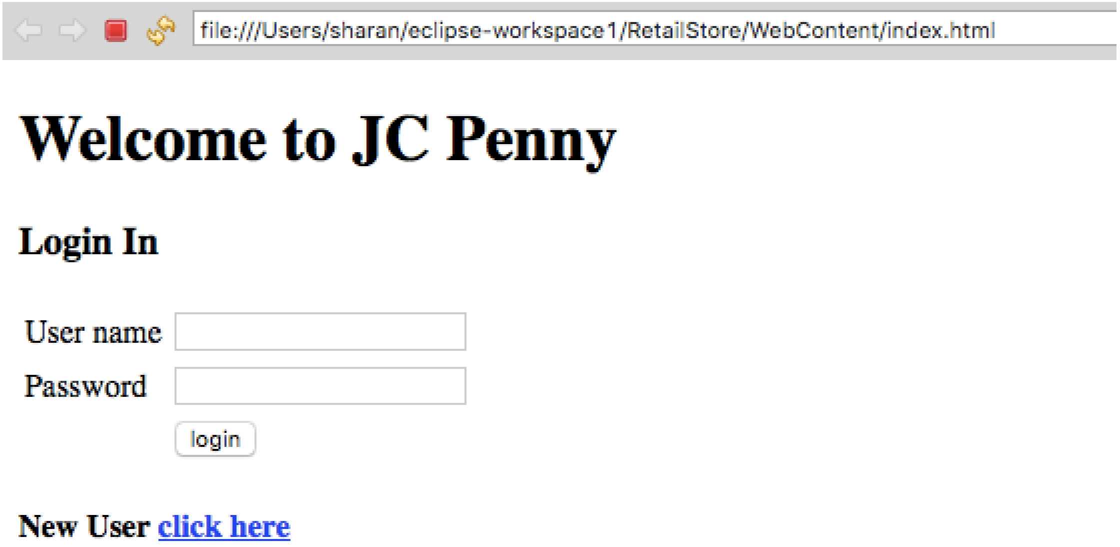
Primary Key and Unique Key Constraints: An entity type usually has an attribute whose values are distinct for each individual entity in the entity set. Such an attribute is called a key attribute, and its values can be used to identify each entity uniquely. Sometimes several attributes together form a key. This is a composite key. The composite key should be minimal, but include all component attributes to have the uniqueness property.

Referential Integrity Constraint: Is specified between two relations and is used to maintain the consistency among tuples in the two relations. A tuple in one relation that refers to another relation must refer to an existing tuple in that relation. For this we use a foreign key to reference the primary key of the tuple in another relation.

Check Constraints: A check constraint is a condition that defines valid data when adding or updating an entry in a table of a relational database. A check constraint is applied to each tuple in the table. The constraint must be a predicate. It can refer to a single or multiple column of the table. The result of the predicate can be TRUE, FALSE, or UNKNOWN, depending on the presence of NULLs.

GUI INTERFACE

LOGIN PAGE



file:///Users/sharan/eclipse-workspace1/RetailStore/WebContent/index.html

Welcome to JC Penny

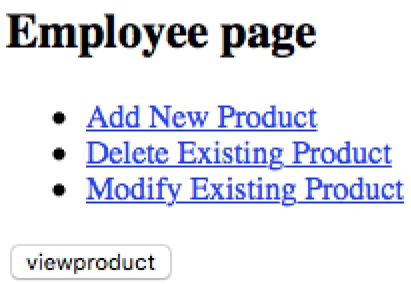
Login In

User name

Password

New User click here

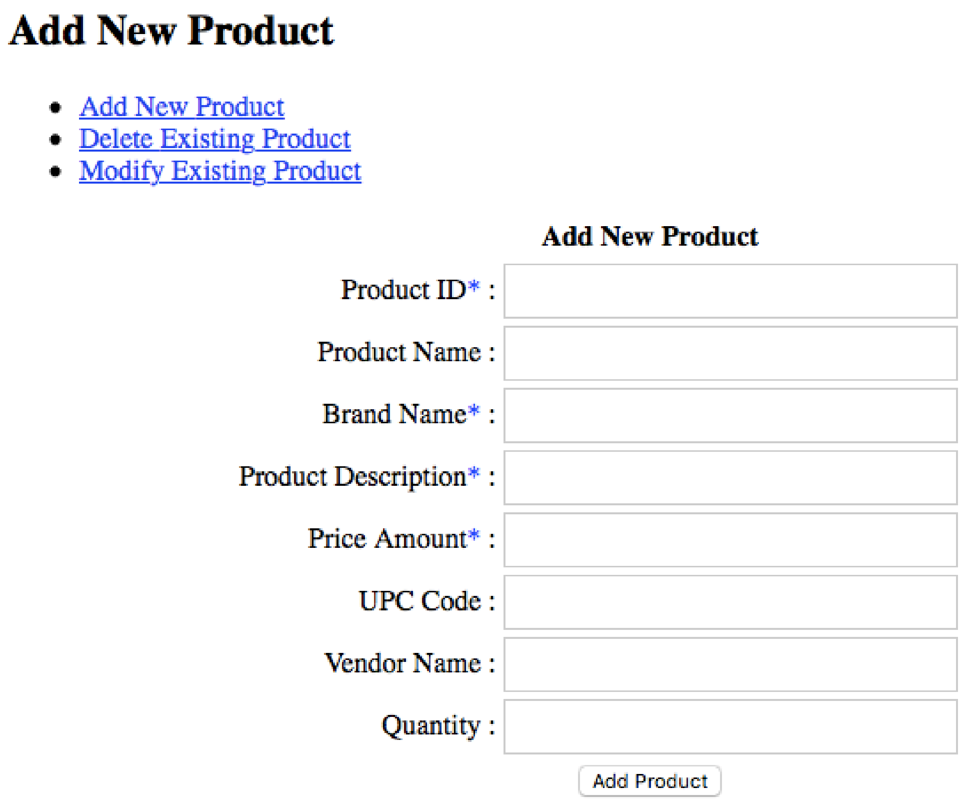
EMPLOYEE LOGIN PAGE



Employee page

- Add New Product
- Delete Existing Product
- Modify Existing Product

WHEN A EMPLOYEE MODIFIES THE INVENTORY

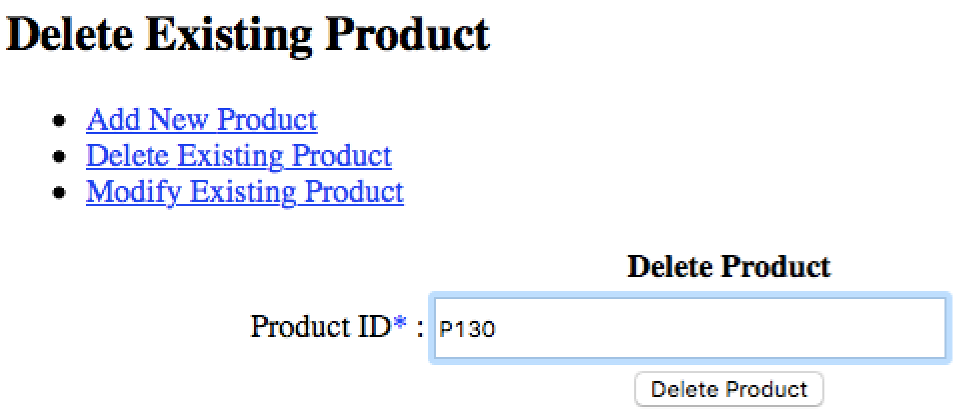


Add New Product

- Add New Product
- Delete Existing Product
- Modify Existing Product

Add New Product

Product ID* :	
Product Name :	
Brand Name* :	
Product Description* :	
Price Amount* :	
UPC Code :	
Vendor Name :	
Quantity :	

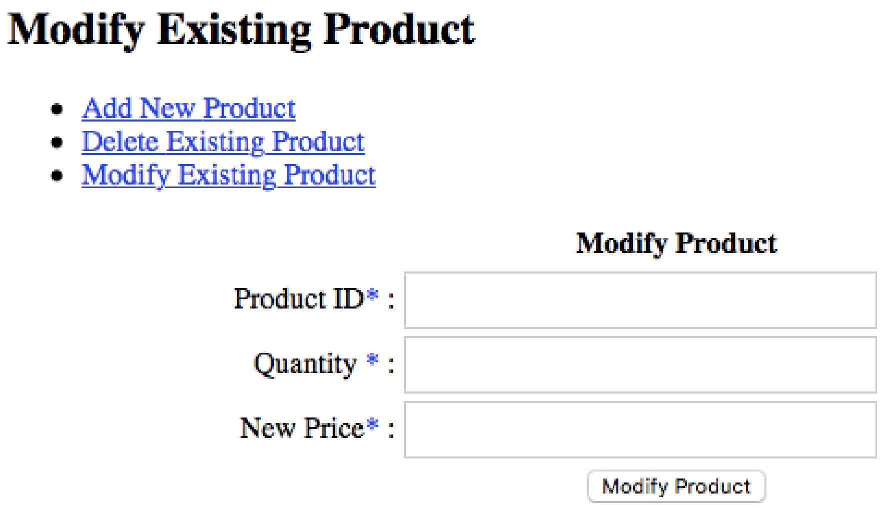


Delete Existing Product

- Add New Product
- Delete Existing Product
- Modify Existing Product

Delete Product

Product ID* :	P130



Modify Existing Product

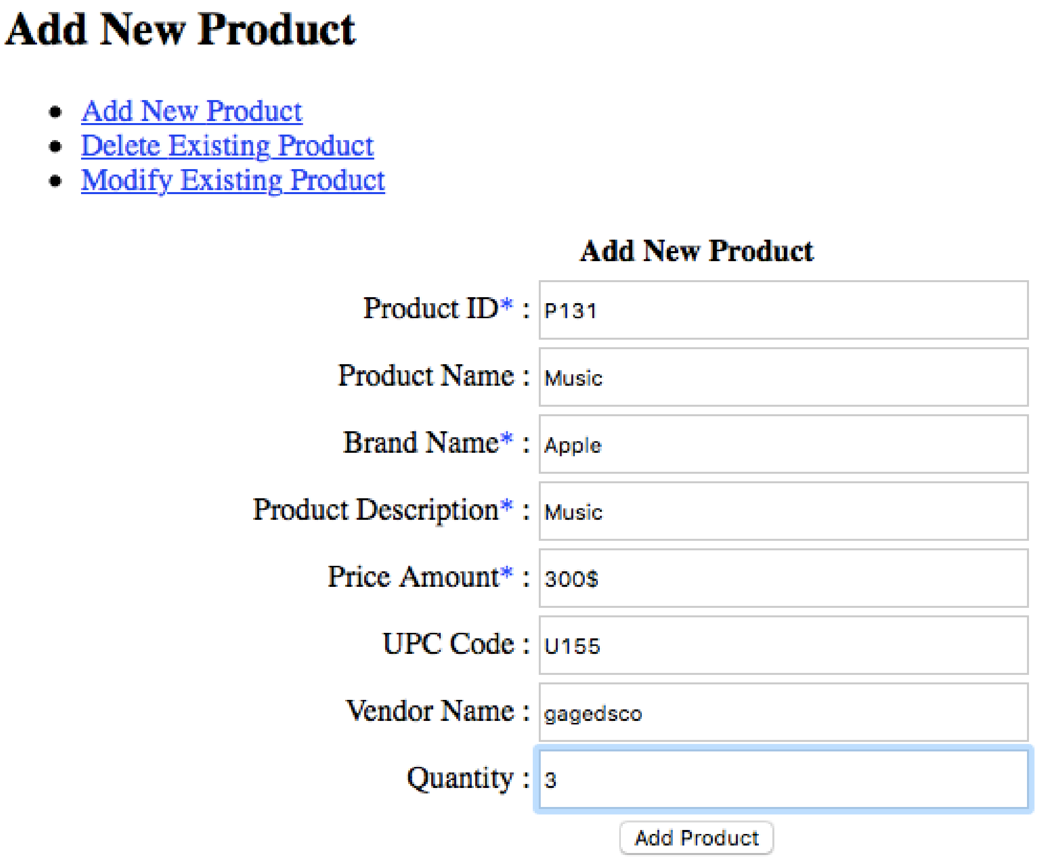
- Add New Product
- Delete Existing Product
- Modify Existing Product

Modify Product

Product ID* :

Quantity * :

New Price* :



Add New Product

- Add New Product
- Delete Existing Product
- Modify Existing Product

Add New Product

Product ID* : P131

Product Name : Music

Brand Name* : Apple

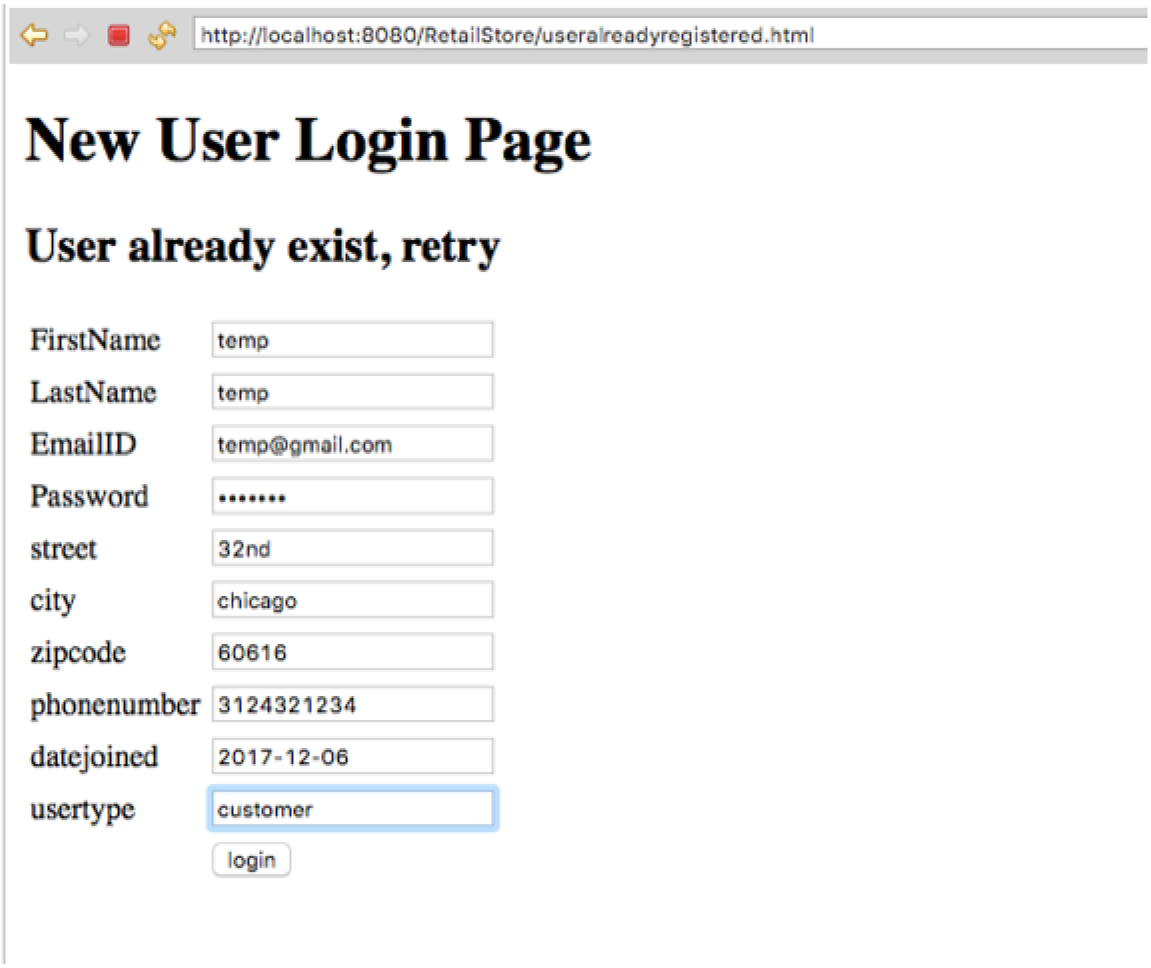
Product Description* : Music

Price Amount* : 300$

UPC Code : U155

Vendor Name : gagedsco

Quantity : 3

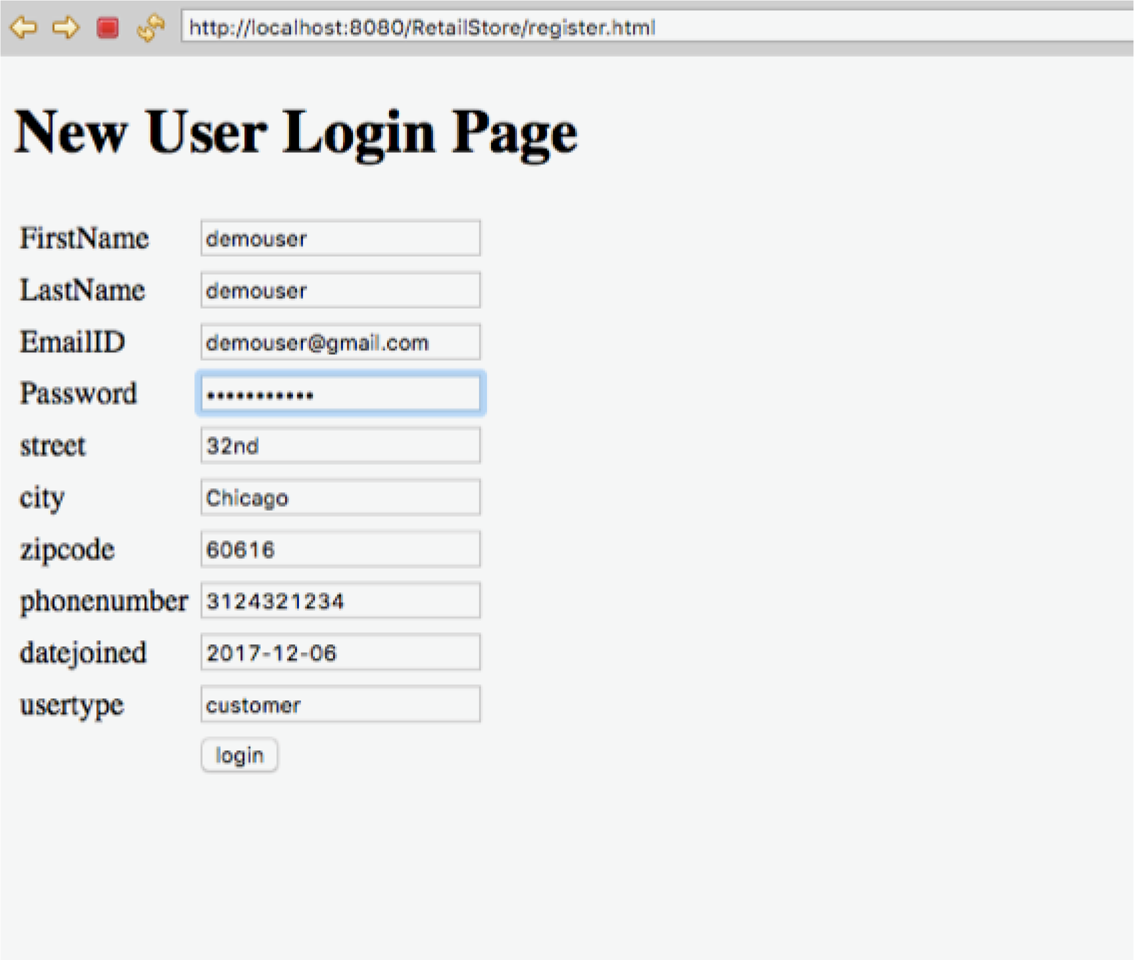


http://localhost:8080/RetailStore/useralreadyregistered.html

New User Login Page

User already exist, retry

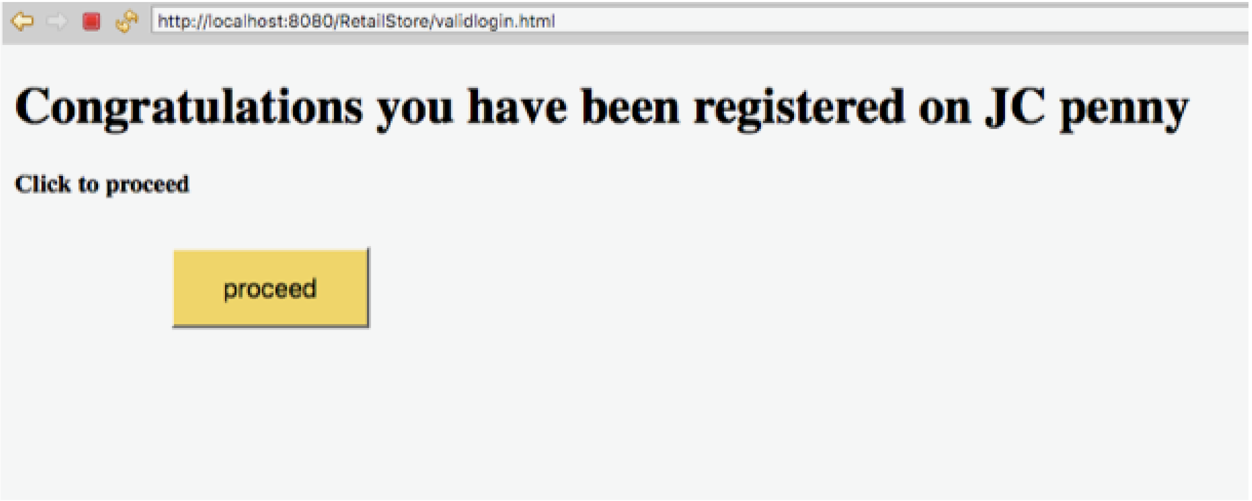
FirstName	temp
LastName	temp
EmailID	temp@gmail.com
Password	•••••••
street	32nd
city	chicago
zipcode	60616
phonenumber	3124321234
datejoined	2017-12-06
usertype	customer



http://localhost:8080/RetailStore/register.html

New User Login Page

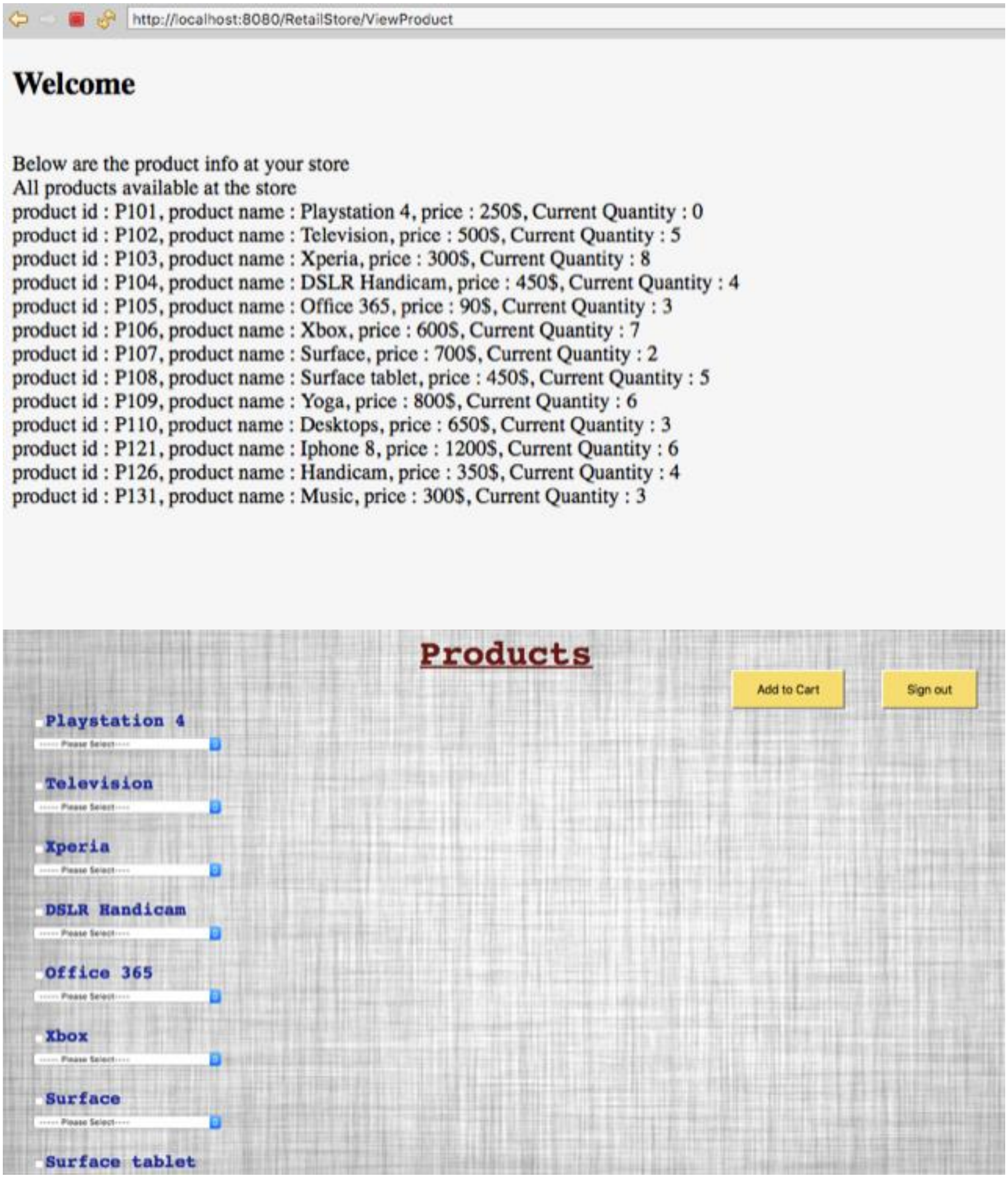
FirstName	demouser
LastName	demouser
EmailID	demouser@gmail.com
Password	••••••••••••
street	32nd
city	Chicago
zipcode	60616
phonenumber	3124321234
datejoined	2017-12-06
usertype	customer



http://localhost:8080/RetailStore/validlogin.html

Congratulations you have been registered on JC penny

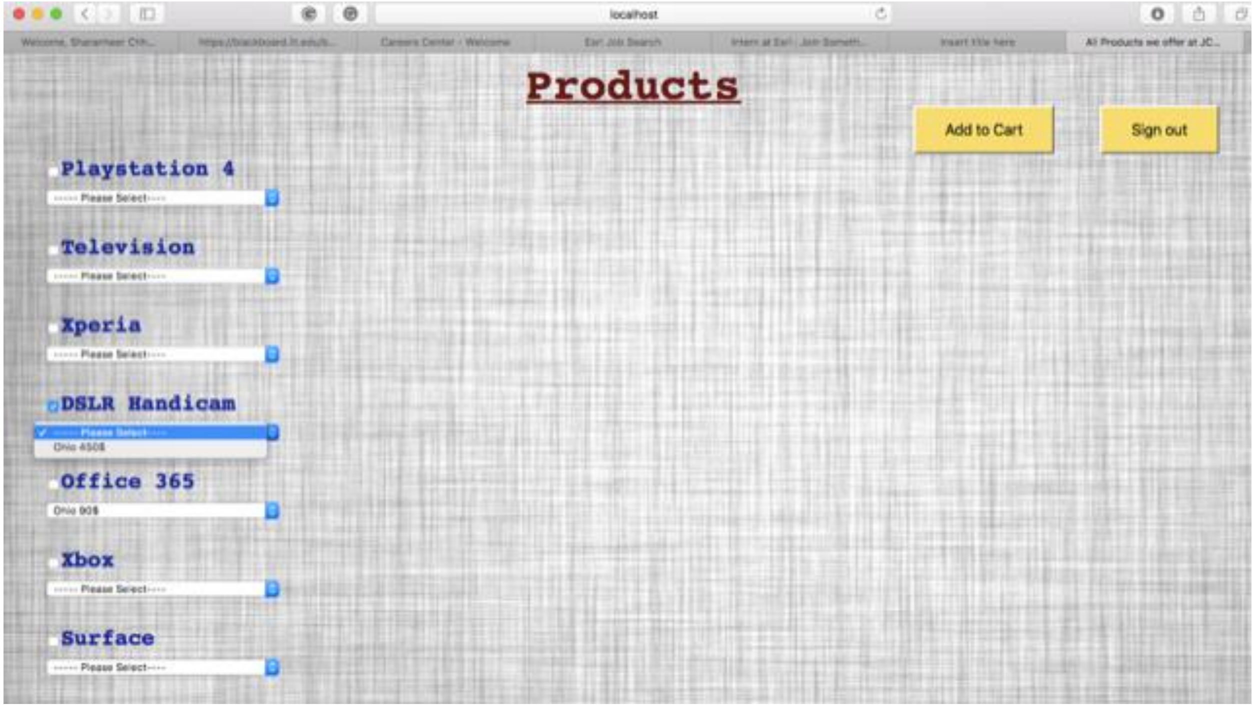
Click to proceed



Welcome

Below are the product info at your store
All products available at the store
product id : P101, product name : Playstation 4, price : 250$, Current Quantity : 0
product id : P102, product name : Television, price : 500$, Current Quantity : 5
product id : P103, product name : Xperia, price : 300$, Current Quantity : 8
product id : P104, product name : DSLR Handicam, price : 450$, Current Quantity : 4
product id : P105, product name : Office 365, price : 90$, Current Quantity : 3
product id : P106, product name : Xbox, price : 600$, Current Quantity : 7
product id : P107, product name : Surface, price : 700$, Current Quantity : 2
product id : P108, product name : Surface tablet, price : 450$, Current Quantity : 5
product id : P109, product name : Yoga, price : 800$, Current Quantity : 6
product id : P110, product name : Desktops, price : 650$, Current Quantity : 3
product id : P121, product name : Iphone 8, price : 1200$, Current Quantity : 6
product id : P126, product name : Handicam, price : 350$, Current Quantity : 4
product id : P131, product name : Music, price : 300$, Current Quantity : 3

Products
Add to Cart
Sign out
Playstation 4
------ Please Select-----
Television
------ Please Select-----
Xperia
------ Please Select-----
DSLR Handicam
------ Please Select-----
Office 365
------ Please Select-----
Xbox
------ Please Select-----
Surface
------ Please Select-----
Surface tablet



Products

□ Playstation 4
------ Please Select-----

□ Television
------ Please Select-----

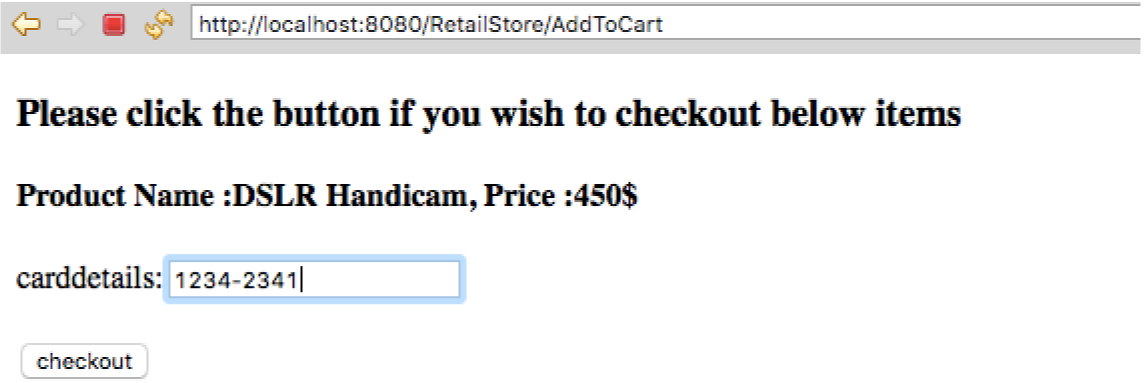
□ Xperia
------ Please Select-----

☑ DSLR Handicam
✓ ------ Please Select-----
Ohio 450$

□ Office 365
Ohio 90$

□ Xbox
------ Please Select-----

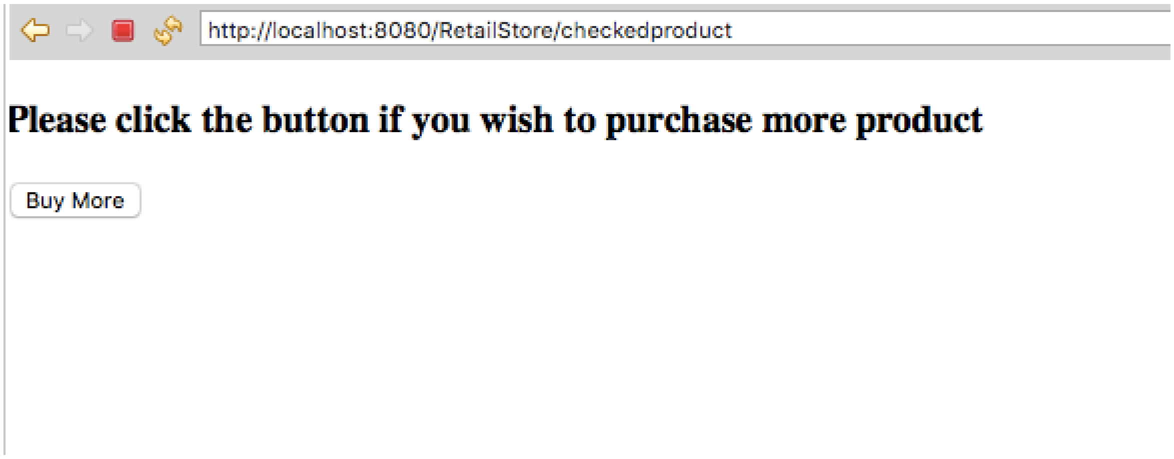
□ Surface
------ Please Select-----



Please click the button if you wish to checkout below items

Product Name :DSLR Handicam, Price :450$

carddetails: 1234-2341



http://localhost:8080/RetailStore/checkedproduct

Please click the button if you wish to purchase more product

Conclusion

JC penny serves as an online store for customers to buy electronic products from the stores at different locations. We offer the best deals for the customers at different locations at their preferred location.