Shopping Cart Project

Nidhi Agarwal

Functional and Technical Specification Document for the project

08

**Fall**

**Objective**

**Part 1** - Create a RESTFul shopping cart API using object oriented php.

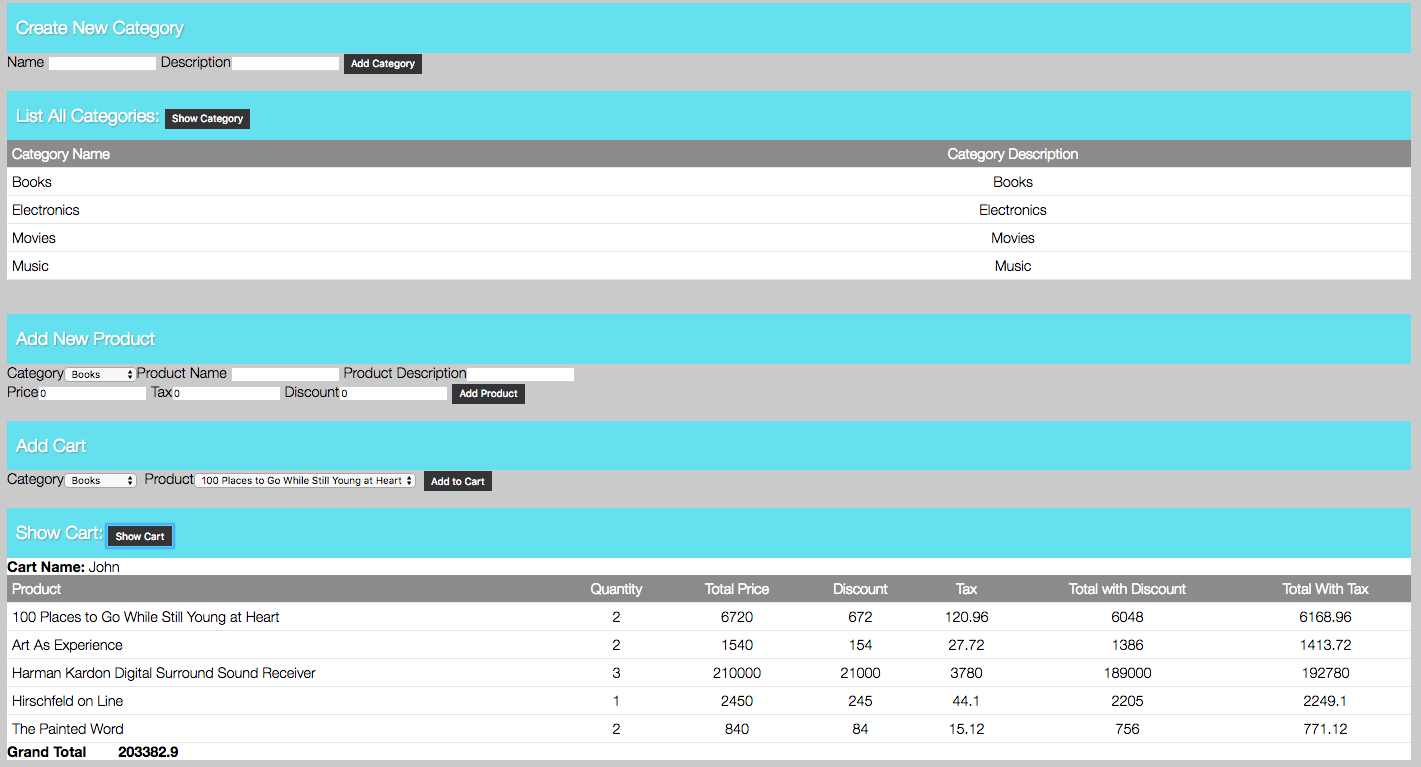
**Important Prerequisite and Instruction**:

* APIs written in core php are preferred. You can use a framework if you can’t do this in core php.
* This API should be testable with REST console or CURL.
* The Development of API must be completely Object Oriented, RESTFul and TestDriven.
* API input must be properly validated.
* API must follow all the standard REST convention.
* API should consist of below Objects:
  + Categories (Name, Description)
  + Products (Name, Description, Price, Tax (in %age), Discount (in %age), Category)
  + Carts (Name, Products, Total, Total Discount, Total with Discount, Total Tax, Total With Tax, Grand Total)
* API List
  + Add Category
  + List Categories
  + Add Product
  + List Products
  + Create cart
  + Show Cart (Includes Cart Name, Products, Total, Total Discount, Total with Discount, Total Tax, Total With Tax, Grand Total)

**Part 2** - Create UI to use these API’s

**Functional Specifications**

**Shopping Cart Home Page**



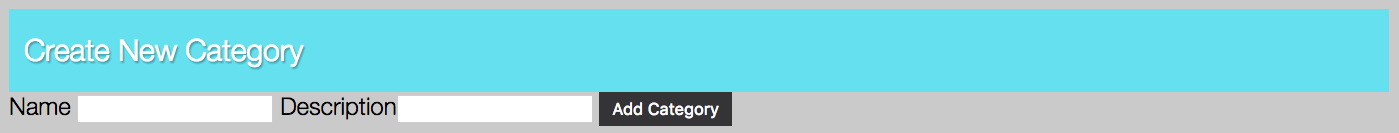
1. **Adding a new category**

* Enter the name of category.
* Enter the description of category
* Click on Add Category button which will internally call http://<servername>/ShoppingCart/addcategory/ API
* New category will be added in category table and you will get a message that “Category has been added successfully”;
* In case of blank or length exceed, proper error messages will be displayed

“Please enter a category name.” “Please limit the category to 50 characters.”

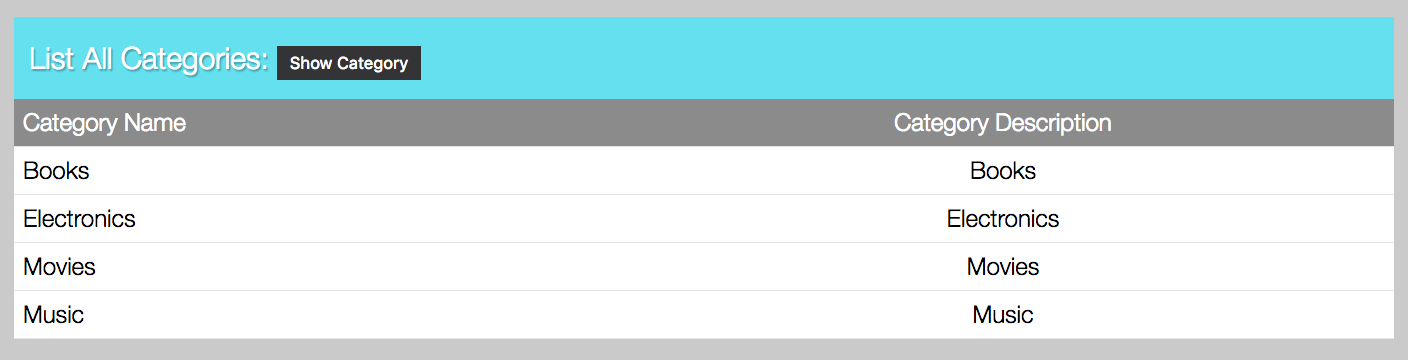
“Please enter a category description.”

“Please limit the category description to 250 characters.”



1. **Displaying the list of categories**

* Click on Show Category button which will internally call [http:// <servername>/ShoppingCart/category/](http://localhost/ShoppingCart/category/) API
* List of categories with category name and description will be displayed



1. **Adding a new product**

* Category dropdown will be populated on page load
* Select the category for which product needs to be added
* Enter the name of the product
* Enter the description of the product
* Enter the price (by default it’s value is 0)
* Enter the tax incurred on the product (in percentage without % sign)
* Enter the discount on the product (in percentage without % sign)
* Click on Add Product button which will internally call [http:// <servername>/ShoppingCart/addproduct/](http://localhost/ShoppingCart/category/) API
* New product will be added in product table and you will get a message that “Product has been added successfully”;
* In case of blank or invalid entry, proper error messages will be displayed

“'Please enter a product name.”

“Please limit the product to 50 characters.”

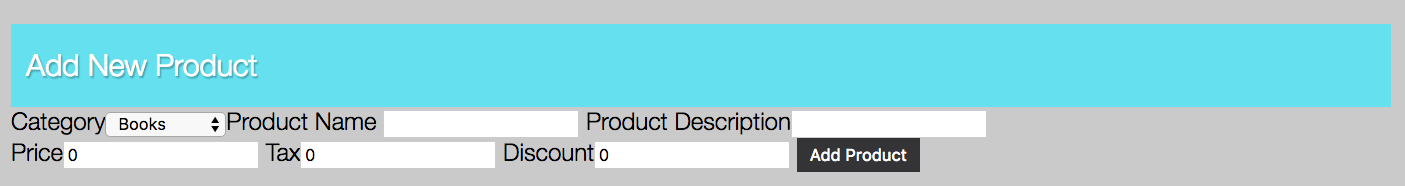
“Please enter a product description.”

“Please limit the product description to 250 characters.”

“Please enter a valid price of the product.”

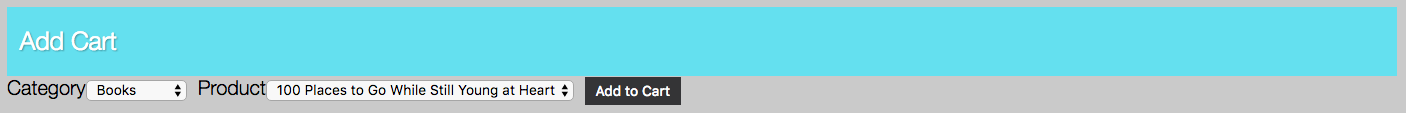
“Please enter a valid tax incurred on this product.”

“Please enter a valid discount incurred on this product.”



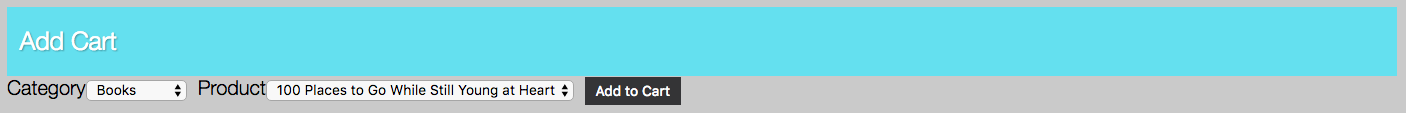
1. **Displaying the list of products**

* By default, the products corresponding to first category will be populated in Product dropdown.
* On change of category from category dropdown, product dropdown will be populated for which [http:// <servername>/ShoppingCart/listproducts/](http://localhost/ShoppingCart/listproducts/) API will be called.



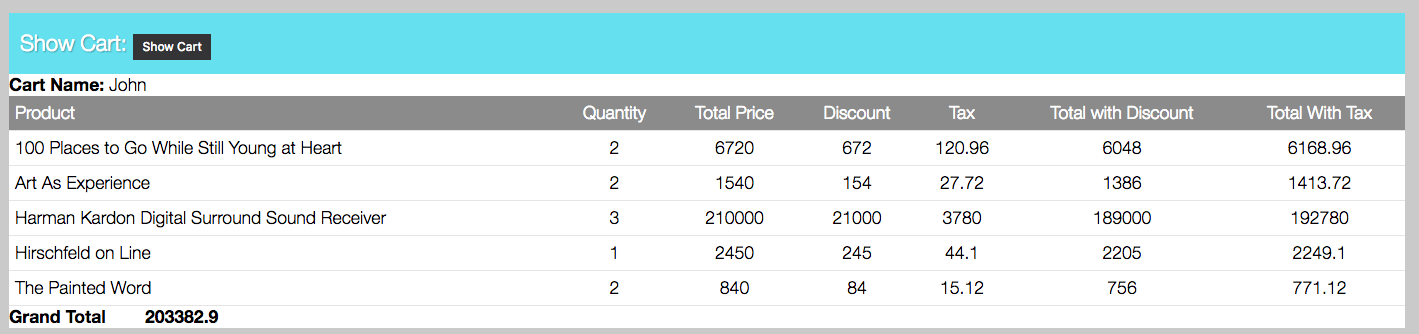
1. **Adding a product into the cart**

* Select the category from category dropdown.
* Select the product from product dropdown.
* Click on Add to Cart button which will internally call [http:// <servername>/ShoppingCart/createcart/](http://localhost/ShoppingCart/createcart/) API
* New item will be added in cart table and you will get a message that “Product has been added successfully in the cart.”



1. **Displaying the products in the cart**

* Click on Show Cart button which will internally call [http:// <servername>/ShoppingCart/showcart/](http://localhost/ShoppingCart/showcart/) API
* List of products with user name, product name, quantity, total price, discount, tax, tax with discount, total with tax and grand total will be displayed for a particular user



**Technical Specifications**

**Deployment Guidelines:**

1. Create a directory ShoppingCart in the apache server
2. Copy the contents of <ShoppingCart.zip> into the ShoppingCart directory on the Apache server
3. Use the database dump file to create and populate the database.
4. Please update the database connection details in the shopping\_cart\_store.php
5. Please copy the contents of the htaccess.txt file to .htaccess file
6. To run the UI please use the following URL

http://<servername>/ShoppingCart/index.php

**File Details**

1. **index.php**

* Consists of the html of shopping cart page

1. **RestController.php**

* Controls the RESTful services of Shopping Cart

1. **ShoppingCart.php**

* Class of all RESTful API calls

1. **ValidateFields.php**

* Class to validate input values

1. **HtmlGenerator.php**

* Class to create html elements

1. **DbConfiguration.php**

* Class having all Database related common functions

1. **shopping\_cart\_store.php**

* Parent class of all shopping cart stores, with default database connection support

1. **product\_store.php**

* Class to control every interactions to product table

1. **cart\_store.php**

* Class to control every interactions to cart table

1. **category\_store.php**

* Class to control every interactions to category table

1. **user\_store.php**

* Class to control every interactions to user table

1. **style.css**

* style of shopping cart page

1. **.htaccess**

* Re-direction URLs for all REST APIs

1. **jquery-3.1.1.min.js**

* Standard class of jquery

1. **request.js**

* Consists of REST API calls through AJAX

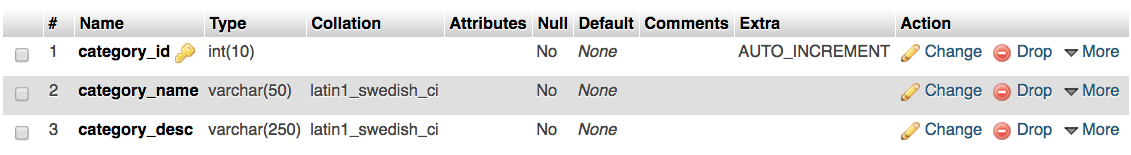
**Database Details** (shopping\_cart)

**Database Dump:** please find the database creation script in <shopping\_cart.sql> file in the attachments.

**Table Structure:**

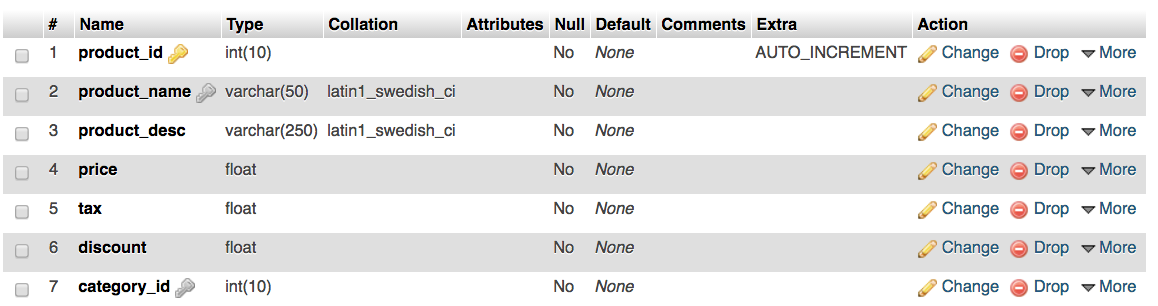
1. **category table**

* This table is used to save category details.

****

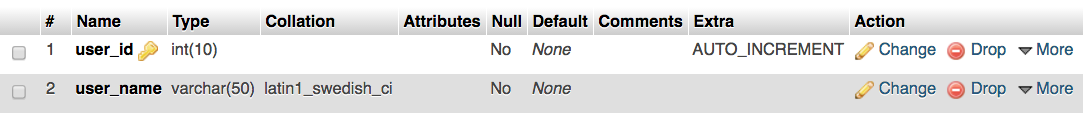
1. **product table**

* This table is used to save product details corresponding to each category.

****

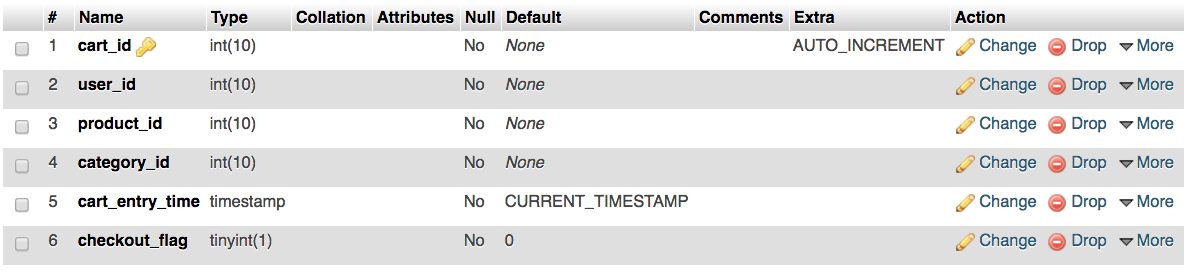
1. **user table**

* This table is used to save user details.

****

1. **cart table**

* This table is used to save cart details.

****

**Assumptions**

1. Database has been created in mysql with MyISAM engine.
2. While adding an item in the cart, the checkout flag remains 0. As the user checks out and completes the payment, checkout flag becomes 1. This part is not yet coded into the application.
3. Only entries with checkout flag as 0 will be displayed in the cart.
4. Cart is being displayed per user basis. Each user would have a single cart and would be unique to him/her.
5. We have created a user and hardcoded it for the demo purpose. The details come from User table.