CSC4370/6370

SUMMER 2018 FINAL PROJECT 4

E-COMMERCE USING PHP-MVC

ALL GROUPS EMAIL ON 07/30/2018 CLASS 12 NOON

Submission Requirements

You must turn work at the SPECIFIED TIME so you can receive credit for Homework!

You must be submitted on (ICollege), by the due date and time. Late homework will be subject to a penalty, as stated in the course grading policy. No email or hard copies of homework will be accepted.

You may discuss the assignments with other students in the class, but (as stated in the academic honesty policy) your written answers must be your own, and you must list the names of other students you discussed the assignment with.

How to Submit

Log into (ICollege), select the class to view its drop box folders, select the correct folder for the given assignment and upload the file there.

(Please submit link of project 4 to ICollege drop box all members AND TEST CASE)

Requirements

You shall choose one team member as leader for purposes of coordinating the project and reporting to the instructor.

Each team shall make a presentation lasting five minutes or less in which you present your completed project. At the beginning of the

Presentation the leader shall present to the instructor a single sheet of paper which states the following:

- **Leader's Name**
- **Project Name**
- **Description: a one-sentence description of your project**

- TOPIC **AN ONLINE VIRTUAL AGENT** that has the ability to Travel anywhere to visit a destination.

DEMO ANY ONE OF THE FEATURES OF BELOW

- Book flights (choose their seating should show the availability)
- Select rental car (Type- SUV, Compact, Midsize, and Luxury)
- Pre-pay for the parking - This system should show the availability of parking along with a variation of prices based up when time frame of purchase, space accommodation and along with a VIP section. **Bonus (Display trending now parking section)

Please be sure to create a sound method of managing the facility

This system will simulate the automation of a real scenario

Please be mind-full that this is a VIRTUAL AGENT application. There are users and an inventory. As a user selects items from the inventory, they are added to their shopping cart etc. When the user checks out, the shopping cart is turned into an order.

Use the MVC framework to model your application. This means you will use PHP for your views, control, and for the model.

***** Try to present your project on your local sever and NOT codd server*****

Project Pieces Views

Views(php/html)	Controls	Models(DB)
1. Main 2. Menu 3. Login	1. Menu 2. Login 3. Addtocart	1. DB Use the following tables 1. Customers 2. Inventory
4. Inventory 5. ViewCart 6. Profile 7. Register	4. Checkout 5. Register 6. Admin	3. Orders 2. PHP Controls 1. User 2. ShoppingCart 3. DB

Interaction Views

Views	Submits to:	Access Models
menu.php	Menu.php	User.php
login.php	Login.php	DB
inventory.php	Addtocart	Shoppingcart.php
viewcart.php	Checkout that simulates CC processing and Email conformation	DB User.php
profile.php	Register.php	Shoppingcart.php
register.php		

Interaction Controls

Control	Forward To	Models(Update/Access)
Menu.php	*	-
Login.php	main.php/login.php	User.php/DB
Addtocart.php	viewcart.php	Shoppingcart.php
Checkout.php	profile.php	DB/Shoppingcart.php
Register.php	main.php/register.php	DB, User.php

other	-	-

Descriptions

Views	Desc.	Displays
menu.php	New user	Links
login.php	Standard login page	Failed attempts
inventory.php	A reflection of the inventory, used to add to the cart	DB:Inventory table
viewcart.php	Shows the status of the current shopping cart with a total	Shoppingcart.php
profile.php	Personal info plus list of past purchases	DB: table
register.php		-

Use Cases

Case	Interacts with	Results
Login	Login.php	User shopping cart updated
Browse Inventory	DB:Inventory is read	DB:Orders updated
Add to cart	Addtocart.php	DB User.php Shoppingcart.php?
Check out	Checkout.php, Shoppingcart.php	DB:Orders updated
View Profile	Register.php	
register.php		

Auto-detecting Credit Card Type

This credit card type detection is a nice addition to the standard payment form because it frees up the user from entering what is actually redundant information

Use data structure called an inversion map from the Google Closure Library. This data structure maps integer ranges to values, a perfect fit for mapping card number prefixes to card types. This card detection code isn’t restricted to regular expression syntax, so it’s free to declaratively mirror the original card number ranges before being transformed and assembled into the final data structure.

Fields for Card Entry

- Credit Card from should have
- Your name
- Credit card type
- Credit card number
- Expiration date
- Extra Feature Coupon or discounts

Second Form fields as it relates to the Credit Card

- Address
- Billing Address
- Phone number

What's in a Credit Card Number?

Despite looking somewhat random, credit card numbers are actually governed by strict conventions. There is a standard called ISO/IEC 7812 that specifies the format for identification numbers on credit cards as well as other card-based identification numbers. The entire identification number is separated into three parts:

Issuer Identification Number (IIN). The IIN is the first four to six digits of the overall identification number and it represents the company that issued the card. In the case of credit cards, the IIN represents the issuing bank.

Account Number. The next few numbers are your personal identification number. For credit cards, this is your account number.

Check Digit. The very last digit is used to verify the overall validity of the identification number. Calculations are used with the preceding numbers to determine that the number format is correct.

Consider the sample MasterCard number 5555-5555-5555-4444 (don't worry, all banks have sample credit card numbers you can use for testing purposes). The first four digits, 5555, is the IIN representing the fake bank issuing the MasterCard. The

numbers 5555-5555-444 are the individual account number and the last 4 is the check digit.

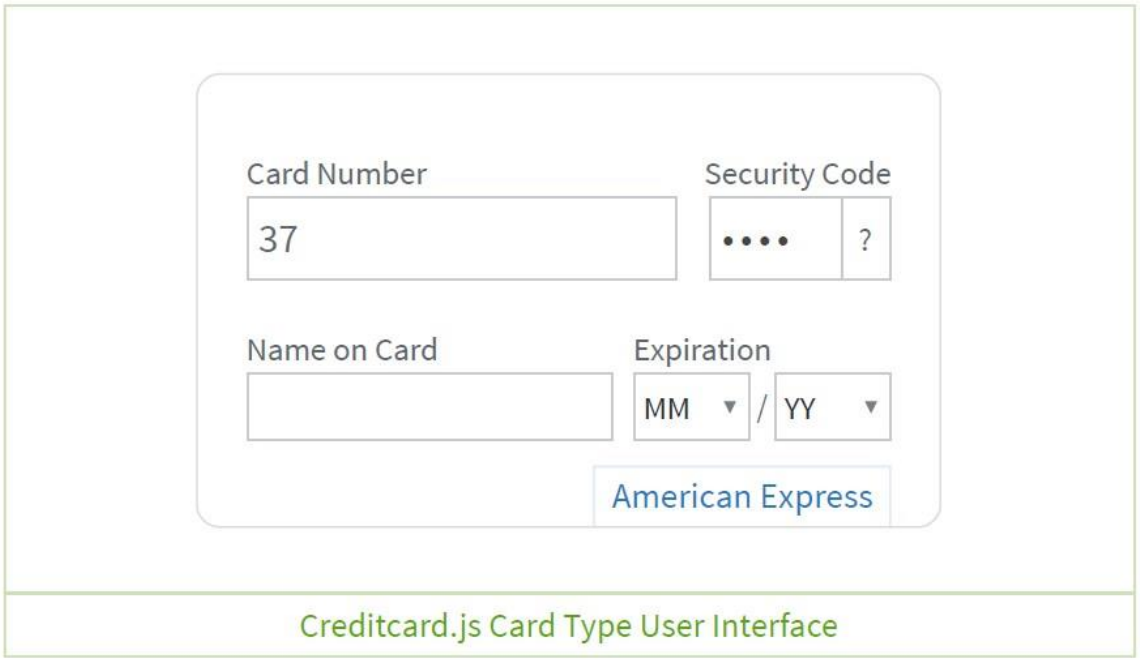
Detecting Credit Card Type

The interesting thing about the IIN is that it also determines the type of credit card.

- Here are some common IIN patterns:
- MasterCard IINs have the first two digts in the range 51-55
- Visa IINs always begin with a 4
- American Expression IINs always begin with 34 or 37

Knowing this, it's possible to write a simple JavaScript function to determine the type of credit card given an account number.

Please Implement this approach by displaying the detected card type in the bottom right of the credit card form



Creditcard.js Card Type User Interface

The use of large text instead of small icons creates a more readable interface. The text is styled just enough to be noticeable experiment with transitioning so when you start typing the Card logo can now be displayed at the bottom of the form.

Here are some demo images you can use to display

*********PLEASE NOT ALL GRADUATE STUDENT ARE REQUIREED TO CREATE AN EXTRA FEATURE (CREATE YOU OWN FUNCTIONALITY WHAT IS NOT DISCRIBED BY ME)

Requirements:

1. See above
    1. Stuff
    2. Things
    3. Project requirements

Be prepared to present in class.