

## IELM 231 IT for Logistics and Manufacturing

### Client-side programming - part I

#### Background:

Up until now, we have seen how a web-user can submit data to a server, and the server can then execute CGI programs to take some actions based on the inputs. Since *CGI programs* run on the web server, they *are caller server-side programs*.

*In this* sequence of *labs*, we will see how to enhance client-server program functionality by also asking a program to run on the client side via the web interface. Namely you will write *client-side programs*.

The main operation is as follows:

1. Web client (e.g. browser) sends a request for a web-page (URL)
2. Server sends the requested page, which contains data+program(s)
3. Client executes the program, which may (i) do something, (ii) generate some more data to display; then the client displays the resulting data as an HTML page.

In the following labs, we shall see what type of things can be done by client-side programs, and in particular, use them for some common and useful client-server tasks. In particular, you create simple applications for *the following three tasks*:

**Task 1.** Simple client-side program to *create a better GUI*, including checking for proper inputs, verification of data, etc. This exercise will also include basics of *Javascript* programming.

**Task 2.** *Sessions tracking* using cookies.

**Task 3.** A simple *secure communication* exercise, using ideas of *symmetric keys*.

In this exercise, we look at task 1.

---

#### Objectives:

- (a) Learn the basics of Javascript language, including program structure, basic data structures including variables and arrays, branching (if-then), looping, and some useful functions.
- (b) Use Javascript to input and validate some user data including Name, Address and Credit Card Number.
- (c) Use a simple CGI program to store the data into a database table on the server.

#### Step 1. Basics of Javascript

Javascript is an interpreted programming language (namely, the code is human-readable, not compiled), and Javascript code is most commonly embedded inside common HTML or XML web pages.

The advantage of using Javascript in a web page is to allow some functions to be run at the client. This has at least two important uses:

- (a) Data input into a form can be verified and corrected at the client side, without having to submit the data to the server for verifications -- therefore GUI design is better.
- (b) It allows for the client computer to encrypt the data being sent, for better security.

Javascript code is typically written inside a typical web page (e.g. inside an html file). The web browser on the client therefore must be able to read, and execute the Javascript code. Almost all modern web browsers can execute Javascript.

A simple Javascript reference sheet is linked from the Lab web-page.

For learning basic commands of JavaScript, and how to use it in an HTML page, you are encouraged to use any one of the tutorials available online (a good one is the WebMonkey tutorials linked from below) -- in particular, the lessons 4 and 5 of the webmonkey tutorial are useful for today's exercise.

**Step 2.** Create a table on your web (and DB) server, using MS Access, as follows:

**Table name:** ClientData

**Fields:**

LastName char(30),

OtherNames char(50),

CCno char(16),

CCtype char(15) which should be one of [ "Master Card", "Visa", "American Express", "Diners Club"]

ExpiryMMYY char(5) which should be of the form "xx/yy", where xx is 00,,12 and yy is 00,99

**Step 3.** Write an HTML file, called SendCCdata.html, which has an HTML form to collect the data required for a record of the ClientData table. This HTML file should contain JavaScript code that will confirm that the data sent by the user is correct, as follows:

- (i) LastName should be a string with only alphabets or hyphen characters.
- (ii) OtherNames should be a string (it should not be NULL, and should have at least one alphabet or numeric character).
- (iii) CCtype should be selected from a selection list as given above.
- (iv) CCno should be a string with only numbers in it; if the user types any spaces, your program should delete them. Then your program should make the following tests for the CCno
  - (a) Length of CCno based on CCtype [Diners Club: 13; Amex: 15, Others: 16]
  - (b) First digit(s): Visa: '4', Master Card: 51-55, Amex: 34-37
  - (c) Luhn test (as described by the example below):  
CC number[digits to be doubled are in red]:    1234 5678 8765 4321  
Sum up all the black digits:   A= 2 + 4 + 6 + 8 + 7 + 5 + 3 + 1 = 36  
Multiply each red digit by 2:    2, 6, 10, 14, 16, 12, 8, 2

Add digits of any double digit numbers: 2, 6, (1+0)=1, (1+4)=5, (1+6)=7, (1+2)= 3, 8, 2  
Sum up all the result:  $B = 2 + 6 + 1 + 5 + 7 + 3 + 8 + 2 = 34$   
 $A + B = S = 36 + 34 = 70$   
If  $S \bmod 10 = 0$ , then the CC number is valid.

For more details, please see the Wikipedia article on Credit card numbers.

(v) The ExpiryMMYY should be in the correct form as described above.

**Step 4.** Write a CGI program that will receive the data sent by the Form in Step 3; it will then check if this credit card number is already in the DB, and if not, it will create the record for this credit card in the ClientData table.

---

**References:**

Webmonkey Javascript tutorials:

<http://www.webmonkey.com/webmonkey/programming/javascript/tutorials/tutorial1.html>

Wikipedia article on Credit card number protocols

[http://en.wikipedia.org/wiki/Credit\\_card\\_number](http://en.wikipedia.org/wiki/Credit_card_number)

---