# CPS630 – Winter 2022
# Web Applications
# Project- (iter-I & iter-II)

Due on Sunday March 6th, 11:59pm

**Submission Instructions:**

- Include a *readme.txt* file in your submission, if there are any special instructions for running your Application.
- Create a zipped folder of all required files, including image and sound files, to execute your Application.
- Projects must be received by the due date and time through D2L, in order to get full marks.
- Projects received within 24 hours after the due date/time, will be marked with %20 marks penalty.
- No projects will be accepted 24 hours after the due date/time.
- Students should work within their groups, and have ONLY one submission per group on D2L.
- Discussion on your projects is recommended, while copying of the works is strictly disallowed.

**Smart Customer Services (SCS):** This project will be developed iteratively through several iterations. [Total 40 Marks]

**Design for Iteration-I:** [15 Marks]

In this iteration you will design, develop and test a "Smart Customer Services" (SCS) Web-Application. SCS is an online system that aims to plan for smart green trips inside the city and its neighborhood for online shopping and then delivery to the destinations. Considering the traffic as a serious threat to the quality of life these years, the world has been looking for various solutions to decrease the stress, frustration, delays and terrible air pollutions being caused through it. SCS attempts to provide a smart green solution on this regard by providing online shopping services and then delivery of the purchased items from the warehouses selected/close to the destination address.

SCS plans to provide services for: selecting items in a shopping cart, selecting branch/date for delivery of items, reviewing/accepting the whole invoice, processing payment, and finally leaving the system.
The SCS online system iteration-I has the following requirements that should be implemented through several web-pages using (but not limited to) HTML5, CSS3, JavaScripts, JQuery, and other required languages/scripts at the client-side:

1-The main page contains options at the top including "Home", "System logo", "About us", "Contact us", "Sign up", "Sign-in", "Reviews", "Shopping Cart", and "Types of Services" as follows:

a) Shopping online from a selected department (only one department should be developed by each team)
b) Delivery to a destination from the selected branch (assume both services a & b are requested at the same time)

Both services (a) and (b) can be provided through various pages (according to your design). The above options should appear at each page too. The "System logo" is a name and/or a small image, and no need to add "Reviews" in It-1 & 2.

2-For both services provide: menus, sub-menus, images, input/output boxes and other required items to go through the following scenario steps:

- System: showing available items, prices for the specific department (only one department should be developed by each team, such as: Furniture, Kitchen, Electrics, Appliances, Beauty, Cloths, Shoes/Bags, Food, ...)
- Customer: Sign-in, Selecting items in a shopping cart; Selecting branch location, date and time for delivery,
- System: Showing the invoice summary and the path on a map from the selected branch to the destination address,
- Customer: Reviewing/accepting the invoice summary; Processing the payment,
- System: Defining a truck for delivery; completing the order with a message to the customer on the page.

3-By selecting the "About us" and "Contact us", the names and contact points of the team members will appear on the screen respectively. A small bio can also appear about each team member.

4-Saving items in the shopping cart should be done via drag & drop method.

5-To use the system for the first time, a "Sign-up" mechanism should be used. Each user should be able to create an id & password and a profile including personal information (name, tel no, mailing address, email address).

**Notes-** 1) Feel free to design the layout of the UIs at each page, 2) Test your application on FireFox and IE browsers.

**Design for Iteration-II:** [20 Marks]
In iteration-II you will enhance and extend your work on iteration-I to add a Database to your SCS Web-Application. You should design, develop, and test a Relational Database working in your application. Start your design by creating the following tables and then extend it to contain the other tables, records, and fields as required in your system:

Order table:
-Order-Id (unique key), Date issued, Date received, Total Price, Payment Code, User-Id, Trip-Id, Receipt-Id,…

Item table:
-Item_Id (unique key), Item_name, Price, Made_in, Department Code, …

User table:
-User-Id (unique key), Name, Tel no, Email, Address, City Code, Login-Id, Password, Balance, …

Trip table:
-Trip-Id (unique key), Source Code, Destination Code, Distance (km), Truck-Id, Price, …

Truck table:
-Truck-Id (unique key), Truck Code, Availability Code, …

Shopping table:
-Receipt-Id (unique key), Store Code, Total Price, …

The following requirements will make your dynamic web-pages to work with multiple database tables and will make your solution to minimize the code duplication. Your database tables should be created at the server side of your network and should be accessed through MySql commands:

1- Combine your code from It-1 with PHP scripts (using PHP classes as much as possible), extract the common headers into separate include files and include them in the PHP files.
2- Create major tables with the above structures with proper name and type for each field; define proper connectivity among the tables. Also add more tables, records and fields if required. Since all web-pages might need to access all tables, you should generalize your database retrieval code through separate classes. Tables should be related preferably based on numeric codes (Ids).
3- Each of the images and information shown through drop-down menus must be linked with appropriate query string.
4- Use proper keys in tables (numeric Id is recommended) to facilitate storing, retrieving, sorting, filtering and manipulating of data in tables.
5- Add another option at the top of the main page as “Search” which brings your system to search mode. By clicking on the “Search”, a search dialogue box will be displayed at the top right of the main page for the users to search for specific order (User-Id, Order-Id) if that order is done or not through the data base tables. The results of the search should be displayed at the bottom right of the main page.
6- Add another option at the top of the main page as “db Maintain” which brings your system to database maintain mode. By clicking on the “db Maintain” a drop-down list will appear that indicates operations on the database including: “Insert”, “Delete”, “Select”, and “Update”. By clicking on each operation, a new page should appear with proper dialogue boxes to get required parameters. This option should be used only by the database administrator.

**Notes-** 1) Feel free to design the layout of the UIs at each page, 2) Test your application on FireFox and IE browsers.

**Project Iterations I & II Deliverables:**
Iterations I & II will be marked in two parts:

Part1) You should submit your technical report along with all your codes and other relevant files through D2L by Sunday March 6th, 11:59pm. Your submission should be in a zipped folder under the name: Team_xx_1, where xx is replaced with your team# (01 to 34). Your technical report should contain: [5Marks]

1- Cover page including: course title, project title, team#, name and student id of all team members, and a table at the bottom part indicating the type and percentage of the tasks being done by each member in your team.
2- Describe briefly about the project objectives, languages and tools you have used. Describe briefly about the design and implementation of your database maintain mode; which MySql commands you have used to implement this mode in order to: create, add, delete, modify, and search through database (you should add drawings or screen-shots at this part).
3- Describe briefly about design and layout of your application user interfaces UIs: main page, second page and any other pages that can be opened while running application using a browser (add either drawings or screen-shots).
4- Describe briefly about the design and structure of your application Databases: name of tables, their fields, primary key (and any other keys), how each table is related to the other tables, how the tables can be accessed via their keys (you should add drawings or screen-shots at this part). Also add your database schema diagram here (similar to the diagrams in slides 57, 58, or 59 of "Lec3-3-Chap14-WorkingWithDatabases").
5- Draw the whole architecture of your application using a class diagram including: the classes you have identified at your design, and how these classes are related to each other. In this diagram, indicate how you have applied MVC pattern on your system architecture. (see the diagrams in Lec5-1, slides 12, 13, 26, 31)

Part2) You should demo your application for both iterations I & II after March 6th. The schedule for the demos will be announced later on D2L.