CPS630 – Winter 2023
Web Applications
Project- (iter-III & iter-IV)

Dues are defined in the guideline of It-1&2.

Submission Instructions:

- There will be only ONE submission for all iterations I-IV.
- There will be only ONE submission per team.
- Details about the submission, dues, and deliverables are defined in the guideline of It-1&2.

Smart Customer Services (SCS): This Project has been developed iteratively through several iterations.
For theses iterations [Total 30 Marks]

Design for Iteration-III: [20 Marks]
In this iteration you will apply changes on the web-application you have developed in iter-1&2 in order to cover all requirements defined in (A), (B), and (C) below (Please note that most of requirements in A and B were already developed in Iter-1&2).

A) The system should work in two separate modes: Admin-mode and User-mode as follows:

1- A d d "Admin-mode"
The Admin should be able to 'Maintain" the system for all major transactions on the databases. TheAdmin can maintain the database to **add /delete/search/edit** the data. The Admin can maintain the database to cover alltypes of data: user accounts, login ids, passwords, titles, images, addresses, descriptions, orders, prices, invoices, user reviews, user rankings, latitudes, longitudes, addresses, ...

2- "Sign Up" (This was already designed and implemented in It-1&2)
The ordinary Users should be able to "Sign-in" to work with the system, only if they already registered through "Sign-up" and created a profile.

B) The system should provide Services to the End Users as follows:

1- "Services (a) and (b)"
(These two services were already designed and implemented in It-1&2: (a) Shopping and (b) Delivery)

2- Add "Service (c)"
This is an extra Service that should be defined by the developer (each team) to be provided as an optional offer. It should be an interesting new Service (an extra one) that makes the system to be preferable by the users, for example adding "Sales Prices", "Express Shipping" or "Price Matching" and etc for a short period of time. The layout and UI for Service (c) should be designed relevant to the other UIs in the system.
After selecting and saving the items in the shopping cart through services a, b, and c, then by clicking on a specific button, the invoice should be generated and displayed on the screen along with the delivery map and the system is ready for the payment process.

3- Add "Payment"
User should be able to enter the credit card number and the banking information to do the purchase for the items saved in shopping cart and see the banking transaction result (assume that it is always a successful transaction).

4- Add "Review"
User should be able to write a short review and define a ranking number (RN) to the service they received and to the items that they purchased (1-5, fromlowest to highest). The "RN" and "Review" for all items should be saved in a separate table in the database. The "RN" and "Review" for each item should be displayed on the screen if selected by the user.

5- Add "Browser" detecting
User should be able to open and view the system on different browsers: Firefox, Internet Explorer, and Chrome (at least 3) with the same layout. The Browser information should appear on the screen.

C) The system should also be re-designed based on the Single-Page-Application **"SPA"** architecture.
Re-Design your system to work based on the **"SPA"** architecture using either of the following JavaScript Frameworks:

AngularJS, Angular or **React**.

First define the layout of your application separate pages (Views), then define elements & contents for each page (data Model) that should be in the SPA, and finally define all events, processes (Controller) to apply sync changes on the SPA (Views, Model).

Design for Iteration-IV: [10 Marks]

In this section, the Web-Application you have designed and developed in iter-III will be enhanced with the "Security" features. On this path, Part-A in iter-III should be revised with the following requirements:

1-For all users of the system, either "Administrators" or "End Users", all passwords should be saved in the database after being encoded by using both "Salting the Hash" and "Secure Hash" MD5 ( ) function. First attach a random "salt" to each password and then apply the MD5 ( ) hash function on it. You need to change your database design a bit for saving both the "salt" and MD5 hash of the salted password in the database with appropriate formats.
Also, you need to update the code that you have developed so far in order to cover the new changes in the design. For example the authentication code for validating the login & password process should be updated. First retrieve the salt from the database based on the submitted user id (login name). Then using the submitted password and the retrieved salt, create a salted password and run it through the one-way hash. Finally the generated MD5 hash value (along with its user id) should be compared with the ones saved in the database. If they are the same, the login was successful and if not there is an error.

2-Apply the same process on the other data you think should be wrapped with a higher level of security, like the user balance and credit card number.

3-Describe about the possible security issues and the algorithm you applied to detect and prevent them in your report (deliverables about it-IV).

Deliverables and Dues for Iter-III & IV:

Deliverables and Dues are defined in the guideline of It-1&2.

Note-In case of adding any extra part to your report, you should name it explicitly in the table of contents and describe it in your report. Include a *readme.txt* file in your submission, if there are any special instructions required to run your application.