Project Description:

Please design and implement a **responsive web site** and **scalable web application** based on the **service-oriented architecture** (SOA).

Mandatory Requirements

A. **HTML/CSS/JavaScript**: You are required to build your web site's client side Graphical User Interface (GUI) using HTML/CSS/JavaScript. You are required to use responsive HTML/CSS/JavaScript templated such as Bootstrap (http://getbootstrap.com) and Foundation 3 (http://foundation.zurb.com), etc.

B. **Server-side Programming**: You can use any programming language for your web site's server-side implementation and your web application's Web Services implementation.

C. **Domain**: The students can pick any domain, application, or service of their choice for their web site/application. Example: implement a project related to e-commerce (i.e. electronic selling and buying of products or services) such as online book store, online flight reservation system, etc.

D. **Functionalities**: Your web site and web application should support the following functionalities:
1. New user registration
2. Existing user login and logout
3. User login information
i. Last date/time of valid login
ii. Number of failed logins
4. User profile information display and editing
5. An image slider of three (3) or more images. Clicking on any of the images in the slider should direct the user to the page in the web-site related to that image.
6. A navigation bar containing at least five (5) entries
7. Search functionality
i. Results should be displayed in a sortable table
8. Shopping cart and order purchase submission:
i. ability to add items
ii. ability to remove items
iii. ability to update item counts
iv. Submission of order purchase request should result in a "purchase order received" confirmation email being sent back.
You are free to use existing third-party libraries to implement the shopping cart
9. Page listing all the previous orders
10. Ability to add and view reviews to purchased items
11. Accessible any unavailable page should retrieve a pretty and generic 404 page
12. Contact-Us page
13. Submission of a request from the Contact-Us page should result in a "message received" confirmation email being sent back.

E. **Database:** It is mandatory that your project use a database to store all data. There is no restriction on what type of database to use. Any NoSQL database or RDBMS is fine.

The database SQL or ORM request and response information should be available in the Web-Service web/app server logs for the TA to review the implementation of this feature. In addition, the TA might inspect the database's content getting updated via a database SQL console.

F. **Web Services:** Any user operation (on your web site) that requires database access (to retrieve information or add/update information in the database) must be performed via Web Services. These Web Services should be hosted as a different web application and on a different web/application server than the web/application server containing the web site. However, the two different web/application servers can reside on the same machine. Your web site will directly or indirectly consume these web services. For this project, Web Services are platform/programming-language independent, unassociated, loosely coupled units of functionalities that are self-contained and implemented via SOAP/WSDL or RESTful methodologies. **All Web Services should require authentication/authorization for clients (i.e. your web site's server or client) to access a particular Web Service's functionality.**
The web service request, database query, and response information should be available in the both the Website and Web-Service web/app server logs for the TA to review the implementation of this feature.

G. **Other Required Features**: Your web site/application implementation should also include the following four (4) features:

1. High Performance: perform distributed caching. Memcached is a good option for implementing a distributed caching mechanism.
Cache miss and cache hit information should be available in the web/app server logs for the TA to review the implementation of this feature.

2. Client-Server Communication Encryption: encrypt the communication channel between the client (i.e. browser), web site server, and Web Services server using TLS/SSL.
The TA will check the implementation of this feature on the Website web/app server by checking if the URL in the browser address bar contains the HTTPS protocol. The TA will check the implementation of this feature on the Web-Service web/app server by checking in the web/app server logs if the web service request call uses the HTTPS protocol.

3. Request/Response Compression: perform compression (e.g. gzip) of:
    a. web site server's response to the client
The TA will check the implementation of this feature by looking at the "Content-Encoding" HTTP response header field either in the browser debug console (a.k.a. inspect element console) or in the Website's web/app server log file
    b. web site server's request to the Web Service server
The TA will check the implementation of this feature by looking at the "Content-Encoding" HTTP request header field in the Web-Service's web/app server log file
    c. Web Service server's response to the web site's server
The TA will check the implementation of this feature by looking at the "Content-Encoding" HTTP response header field in the Web-Service's

web/app server log file

4. Asynchronous Services: at least two (2) supported web site's functionalities should be asynchronous and implemented using AJAX.

Extra Features:
1. Object-Relation Mapping (ORM) Framework: perform mapping of object-oriented domain model to RDBMS tables using ORM frameworks such as Hibernate (Java), ADO.NET Entity Framework (.NET), Django (Python), Propel (PHP), etc.
Object serialization/unserialization and SQL query information should be available in the web/app server logs for the TA to review the implementation of this feature.