# Teamwork Project Assignment for the [Web Services and Cloud Course @ SoftUni](https://softuni.bg/courses/web-services-and-cloud/)

This teamwork project assignment is designed to develop skills for creating RESTful Web services (using ASP.NET Web API) and to deploy the Web services in cloud environment like Azure and AppHarbor.

## Project Description

Design and implement RESTful Web services. Deploy the ASP.NET Web API application in the cloud. Implement a JavaScript client application, which consumes the services from the cloud.

You are free to choose what kind of services and application to build. You could implement project like:

* **Web Chat Application**
  + The app holds users, messages, notifications, …
  + Users can send messages between each other
  + Users can send files, photos, etc.
  + Users receive notification when another user send them a message
  + …
* **LinkedIn-like Application**
  + The app holds users, groups, skills, endorsements, …
  + Users can have connections
  + Users can have skills
  + Users can endorse their connections for skill
  + Users can create groups
  + Users can join groups
  + …
* **Facebook-like Application**
  + The app holds users, posts, comments, …
  + Users can have friends
  + Users have wall
  + Users can write a post on his or on his friend wall
  + Users can create groups
  + Users can join groups
  + Users can comment his friends posts
  + …
* **Twitter-like Application**
  + The app holds users, tweets, …
  + Users can follow users and can be followed by another users
  + Users can post tweets
  + Users can reply to tweets
  + …
* **Image Gallery Application**
  + The app holds users, galleries, albums, comments, notifications, …
  + Users can create gallery
  + The gallery can have many albums
  + Users can subscribe for gallery
  + Users can upload images in albums in their own galleries
  + Images have title
  + Users can leave a comment about an image
  + Users receive notifications when somebody comments an image of theirs
  + …
* **Tic-tac-toe Game**
  + The app holds users, games, …
  + Users can create game
  + Users can join an existing game
  + Users can perform moves in a started game
  + Users receive notifications when a user in a game of theirs has made their move
  + …
* **Bulls and Cows Game**
  + The app holds users, games…
  + Users can create game
  + Users can join a random game
  + Users can perform moves in a started game
  + Users receive notifications when a user in a game of theirs has made their move
  + …
* **Application by Choice**
  + You can design and implement your own application
  + You can modify one of sample applications described above

## General Requirements

All projects should implement **authentication** (user registration, login and logout). Some services should be public, while others should be private (accessed after successful login).

### Server-Side Application

* Your application must be implemented using **ASP.NET Web API** framework.
* Expose your public services as **RESTful Web services**.
* Host the application in a **cloud environment**, e.g. in **AppHarbor** or **Azure**.
* Use **database in the cloud**, e.g. **MS SQL**, **MySQL**, **MongoDB**, **Redis** or other.
* Optionally use a **file storage cloud API**, e.g. **Dropbox**, **Google Drive** or other
* Optionally use a **real-time push notification service**, e.g. **PubNub**, **Azure Notifications Hub** or other

### Client Application (JavaScript)

* Implement a **simple client UI** for your application:
  + Do not put too much time on a beautiful UI.
  + Use UI libraries and frameworks to save time.
* The client application should consume the RESTful services using HTTP requests.
* The application should run on a modern Web browser.

## Additional Requirements

* Follow the best practices for OO design and **high-quality code** for the RESTful service app:
  + Use data encapsulation.
  + Use exception handling properly.
  + Use inheritance, abstraction and polymorphism properly.
  + Follow the principles of strong cohesion and loose coupling.
  + Correctly format and structure your code, name your identifiers and make the code readable.
* Use a **source control system** by choice, e.g. Git, SVN, GitHub, CodePlex.
  + Submit a link to your public repository.

## Public Project Defense

Each team will have to deliver a **public defense** of its work in front of the other students, trainers and assistants. Teams will have **only 10 minutes** for the following:

* **Demonstrate** how the application works (very shortly).
* Show the **source code** and explain how it works.
* Explain how each team member has **contributed**: display the commit logs in the Source Control System you are using.
* Optionally you might prepare a **presentation** (3-4 slides).

Please be **strict in timing**! On the 10th minute you **will be interrupted**! It is good idea to leave **the last 2 minutes for questions** from the other students, trainers and assistants.

Be **well prepared** for presenting maximum of your work for minimum time. Bring your own laptop. Test it preliminary with the multimedia projector. Open the project assets beforehand to save time.

## Assessment Criteria

* **Service Authentication** (register / login / logout) **– 0…10**
* **Service Functionality** (at least 5 service endpoints) – **0…25**
* **Client UI** (the focus is on the services, not on the client, so the UI gives less points) – **0…10**
* **Code Quality** (well-structured code, split into classes and files, good naming, formatting, etc.) – **0…5**
* **Teamwork\*** (source control; each team member contributed in 5 different days; distribution of tasks) – **0…5**
* **Bonus** (bonus points are given for implementing optional functionalities / original approach) – **0..5**

\* If not all team members have contributed to the project, this **does not affect** the Teamwork points.

## Give Feedback about Your Teammates

You will be invited to **provide feedback** about all your teammates, their attitude to this project, their technical skills, their team working skills, their contribution to the project, etc. The feedback is important part of the project evaluation so **take it seriously** and be honest.