**Practical Aspnet**  
Seavus Education And Development Center

# **Goal**

This course will cover aspnet web application development, using all the technologies that were previously learned on the Academy. The goal of this course is to prepare all the students for developing aspnet web application

# **Technologies**

## Main Technology stack

* ASP.NET Web Application / C#
* MS SQL Server / Entity Framework / MongoDb
* JavaScript / jQuery

# **Project**

## Specification

**Web application for ordering food**

* Administration of restaurants (this is done by system administrators)
* Administration of a restaurant menu (this is done by the restaurant account)
* Order submission (needs to be done by the end user
* Order approval (needs to be approved by the restaurant)
* Order status tracking
* Reviews functionality
* Favorites (end user can mark a restaurant as favorite and can see them in favorites page)

## Setup

* Git repository for Application hosted on GitHub
* Git repository for Students hosted on GitHub
* Presentation
* Project execution plan
* Comments on code for students

# **Project Requirements**

## Administration

* This part of the system is dedicated for accounts with role of type "administrator"
* The system administrator should have 4 visible pages:
  + **Administrators**
  + **Restaurants**
  + **Orders**
  + **Reviews**
* On Administrators page the admin should be able to add/remove another admin
* On Restaurants page the admin should be able to execute crud operations over restaurants
* On Orders page the admin should be able to manage orders if necessary.
* On Reviews page the admin should be able to manage reviews and comments. There should be an option to approve or decline a review, because the review may contain inappropriate words.

## Restaurants

* This part of the system is dedicated for accounts with role of type "restaurant"
* The restaurant account should have 4 pages visible
  + **Profile**
  + **Menu**
  + **Orders**
  + **Reviews**
* On the **Profile** the restaurant account should be able to manage its profile and information like name, location and contact information
* On the **Menu** page it should be able to manage its menu. It should be able to execute crud operations and ordering/grouping menu items
* On the **Orders** page it should be able to manage orders like declining
* It should contain a page where the restaurant account can manage its menu, which means there should be implemented crud operations for menu items
* On the **Reviews** page it should be able to manage reviews. There should be an option to approve or decline an review, reply to a review.

## End Users

* This part of the system is dedicated for end users
* It can be accessible publicly for browsing restaurants and menus.
* For submitting an order the end user must be logged in

## Order Entity

* Is created by the end user
* End user can chose a restaurant
* End user can add/remove items in the order
* End user can change the amount of the items in the list (something like 2 Pizzas and 3 Chicken Burgers)
* End user can add a comment on the order (something like "i don't want ketchup in the Cheese Burger")
* Order can be submitter and is saved with status "under approval".
* At this moment the order should be visible on the "Orders" page of the restaurant account.
* The restaurant should be able to manage the order, for an instance to decline an order, call the end user via telephone to ask him for more information or approve the order.
* After approval the order should be saved with the status "in progress".
* After the order exits the restaurant the restaurant account marks order as "in delivery"
* After the order reaches the destination, the order is marked as "delivered"
* The user can now submit a review and/or add a comment for any order item