# Teamwork Project Assignment for the [Web Development Basics @ SoftUni](https://softuni.bg/trainings/1211/Web-Development-Basics-September-2015)

[Design and implement a](https://softuni.bg/trainings/1211/Web-Development-Basics-September-2015) **Shopping cart/CMS/RPG Game** **using PHP and HTML / CSS / JavaScript**. Your project must meet all the requirements listed below.

## Requirements

* **Use PHP** – the major part of your work should be PHP written
  + You should create custom PHP Framework
  + You should create the project over your framework
  + You must additionallyuse **HTML5, CSS3** to create the content and to stylize your web application
  + You may optionally use **JavaScript, jQuery, Bootstrap**
  + Use **PHP 5 or higher (7 is a bonus)**
* **User source control system**
  + **Use GitHub** or other source control systemas project collaboration platform
* Publish your project **live in Internet** – your project should be public in Internet
  + You may **share your project** to get external feedback
  + Most shared and commented projects will get additional **bonus score**
* **Valid and high-quality PHP, HTML and CSS**
  + Follow the best practices fr PHP development: [http://www.phptherightway.com](http://www.phptherightway.com/).
  + Validate (when possible) your HTML ([http://validator.w3.org](http://validator.w3.org/)) and CSS code ([http://css-validator.org](http://css-validator.org/))
  + Follow the best practices for **high-quality PHP, HTML and CSS**: good formatting, good code structure, consistent naming etc.
* **Usability, UX and browser support**
  + Your web application should be easy-to-use, with intuitive UI, with good usability
  + Ensure your web application works correctly in the latest HTML5-compatible browsers: Chrome, Firefox, IE, Opera, Safari (latest versions, desktop and mobile versions)
  + You do not need to support old browsers like IE9
* **Framework requirements**
  + **ABSOLUTELY REQUIRED** (without those ones the project is invalidated)
    - Default routing system
    - Custom routing system
      * Easy way to define custom routes with parameters and parameter types
    - Areas
      * Custom routing for areas; It should override the global routing configuration
    - Strongly typed views
    - POST request actions using Binding Models
    - ViewHelpers for Forms, Ajax submitted Forms, CSRF Token
      * Form ViewHelper
        + Radio buttons
        + Checkboxes
        + Dropdowns
        + Text fields
        + Password fields
        + Textareas
    - Out of the box escaping the html chars unless stated otherwise
  + **Other**
    - Data annotations
      * @Authorize – the method is not called if the user is not logged in
      * @Admin – the method is not called if the user is not logged and is not administrator
    - Overriding default route config with annotations
      * @POST – the method can be called only on POST
      * @PUT – the method can be called only on PUT
      * @DELETE – the method can be called only on DELETE
      * @GET – can be omitted. If declared or not, the method is called only on GET
      * @Route(“route/here”) – declares the route action is called upon
    - Dependency injection configuration (config file or annotation)
    - User roles and config/annotation for them (method is not called if user is not in the role)
    - Upload files ViewHelper
    - Help page that scans all routes (configured, annotated and basic ones) and shows them (e.g. PUT users/{id}). Opening the route shows the serialized Binding Model that accepts if any or primitive parameters if any. (e.g. {name: name, email: email}, or ?id={id}&name={name})

### Shopping cart

**Required** functionalities:

* User registration / login and user profiles.
* User roles (user, administrator, editor)
* Initial cash for users
* Product categories
* Listing products in categories
* Add to cart functionality
* Promotions for certain time interval
  + Promotions on certain products (% discount)
  + Promotions on all products (% discount)
  + Promotions on certain categories (% discount)
  + Promotions for certain user criteria (registered more than X days, have more than X cash, etc…)
  + If two or more promotions collide on a date period for certain product – the biggest one applies only
* Visibility only of available products
* Quantity visibility
* Checkout the cart
* View cart
* Users can sell bought products
* Editors can add/delete product categories
* Editors can add/delete products
* Editors can move products between categories
* Editors can change quantities
* Editors can reorder products
* Administrators have full access on products, categories, users and their possessions

**Bonus** functionalities

* Managing the cart
* Users can sell products and put them promotions
* Users can make comments on products (review)
* Administrators: ban users
* Administrators: ban IP’s

## Public Project Defense

Each student will have to deliver a **public defense** of its work in front of the SoftUni team. The students will have **only ~20 minutes** for the following:

* **Demonstrate** the web application (very shortly).
* Show the **source code** and explain how it works.

Be **well prepared** for presenting maximum of your work for minimum time. Open the project assets beforehand to save time.

## Assessment Criteria

* **Functionality** (all the required functionalities according to the type of project you choose) – **0…50**
* **Framework functionality – 0..70**
* **Overview** (HTML / CSS / Usability / UX) – **0…10**
* **Code quality** (correct naming, code formatting, separation of concerns, etc.) – **0…30**
* **Security** (XSS, SQL Injection, CSRF…) **– 0…25**
* **Bonus** (bonus point are given for implementing optional functionalities according to the type of project you choose) – **0..35**