# Team Grapes

# Project Assignment for the [JavaScript Core Module @ SoftUni](https://softuni.bg/trainings/1686/practical-teamwork-javascript-may-2017)

Design and implement a **Single-Page Application**, using JavaScript and tools of your choice.

## Technologies

Your project must demonstrate the use of the following:

* **JavaScript** – your main functionality must be in JS
* **Structured Code** – split your code into modules and make use of **programming patterns**, such as MVC
* **Templating and Routing** – use a library of your choice (**mustache** and **sammy**, **React** and **React-Router**)
* **HTML** & **CSS** – provide a UI for the user; **bootstrap** and other libraries are **optional**
* **Backend Service –** your app must communicate with a **REST** service or **API** via **AJAX** calls or a web socket (**Kinvey** or a provider of your choice)

You are allowed to mix and match libraries as you see fit, but be prepared to motivate their usage.

## Project Scope

Your project should implement **at least** the following **functionality**:

* User **registration**, **login** and **logout**
* **View** some content (e.g. photos and short description of products)
* **Create** new content (e.g. post new order)
* **Edit** existing content(e.g. edit order (add other products to his order)))
* **Delete** existing content(e.g. for example cancel/delete his order(we can make this option to be available only 8 hours after making the order)
* An **admin role**, able to modify other users’ profiles or data (an admin will manage status change)

Your project should keep its data in a **Kinvey** or some other **backend service**. You may implement any additional functionality as you see fit.

## Sample Assignments

All assignments will require a certain level of **research** and **additional knowledge**, not covered in the main course. You may consult the trainers for guidance and advice.

# Orders System

Create a Web-based application for creating and managing orders. The web application is supposed to include three parts:

* Products page
* Registered users page – for making/editing the order and review order status.
* Administration interface for managing order status

Provide the following functionality to all visitors (without authentication):

* Preview functionality
* Registration form

Registered users can:

* Create orders
* Customize their profile
* Assign due dates (we can limit them with minimal time of production needed for concrete product)
* Edit orders
* Cancel orders

Authenticated administrators should be able to:

* Manage order status
* Leave a comment/message for a concrete order/user

In addition to the normal functionality, project creators can:

* Add manager role and grant and revoke permissions to them by the administrator.
* View time-tracking details (how long each order was with given status).

## Team Organization

You **must** assign a **project manager** who will be in charge of keeping the team synchronized. It doesn’t have to be the most knowledgeable person, but rather the person who can **spare the most time** and can communicate with his teammates, delegate tasks and resolve conflicts.

Maintaining some form of **source control** is **mandatory** (GitHub recommended) – each team member must commit regularly.

A **task management tool** is **advisable** (Trello recommended) – try to split the tasks so that a single person can work on each semi-independently. Mark progress regularly.

## Deliverables

Your main deliverable is the **project repository** – commit to it the following:

* Complete **source code** of your project (JS, HTML, CSS, media assets, etc.)
* Units tests (**optional**)
* Documentation (**optional**)
* Design prototypes (**optional**)
* Any other project assets you need for the final presentation

During your **final presentation**, you will need to **demonstrate** your product – either with a local server on your machine, or deployed online (GitHub pages or hosting of your choice).

## Public Project Defense

Each team will have to deliver a **public defense** of their work. At least **the team manager** should be present. The teams will have **only ~20 minutes** for the following:

* **Demonstrate** the application’s functionality (in a concise manner)
* Show the **source code** and explain briefly how it works
* Answer **questions** from the trainers and audience

Hints for better presentation:

* Be **well prepared** for presenting maximum of your work for minimum time
* **Open all project assets** beforehand to **save time**: open your site in the browser, open your project repository page, boot up your IDE, etc.
* Test your laptop with the **multimedia projector** and ask for any output adapters in advance
* Prepare a short PowerPoint presentation (**optional**)

Answering questions is an **important part of the evaluation process** – if you do not leave enough time after your presentation, your **score** **will be reduced**!

## Assessment Criteria

* **Project – 25**
  + **Code Quality** (correct use of patterns, following language conventions, etc.)
  + **Functionality** (rich and seamless functionality, demonstrating application of course material will be scored higher)
* **Teamwork – 50**
  + **Source Control** (use of Issues, Branches, regular commits by all team members)
  + **Task management** (coherent implementation process, following incremental steps)
  + **Project Scope** (you’ve met all goals that you started with)
  + **Team Communication** (team members are kept up to date with each other’s progress)
* **Defense – 25**
  + **Answer questions** (defend your work!)
* **Bonus – 10** (awarded for achievements outside this assignment)

## Deadline

The public defense and evaluation will be held on **14-September-2017**. You must submit a link to your repository **no later** than the time you’re scheduled to present your work.