|  |  |  |  |
| --- | --- | --- | --- |
| Assignment 1 | | Project Summary | |
| Course | | Fullstack Application Development with Node.js + Express.js + React.js - 2019 | |
|  | | | |
| Project author | | | |
| № | Pseudonym | | Face-to-face/ online |
| 1 | Dimitar Marinov | | face-to-face |
| 2 | Milen Rabadzhiev | | face-to-face |

|  |  |
| --- | --- |
| Project name | Online shop system |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| Internet has evolved a lot in the past years, so have selling goods, more and more organisations started to rely on selling their goods online.  We would like to offer a system that makes it easier for end users to see and check-out a product. The system will be developed as a Single Page Application (SPA) using React.js as front-end, and Node.js + express as backend technologies.  • Guest users – can view the information about products and search for them  • Registered users - can view the information about products and add them to cart  • Products - have different varieties and can be bought by registered users  • Cart - registered users can add the products that they would like to buy, it has a check-out button which finalizes the purchase  • Administrator - can manage (create, edit user data and delete) all products and users |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Browse information about different products** | Users can browse the existing products and search products with a keyword | Registered and guest users |
| * 1. **Register** | *Guest users can register by providing a valid e-mail, password and username. By default all users are guests.* | *Guest users* |
| * 1. **Manage users** | *Administrators can choose an users and edit his data* | *Administrator* |
| * 1. **Manage products** | *Administrators can make CRUD operations on products* | *Administrator* |
| * 1. **Add products to cart** | *Registered users can add different products to their cart* | *Registered users* |
| * 1. **Check-out** | *Registered users can check-out the products that are in their cart* | *Registered users* |
| * 1. **Search information by different category of a product** | *Users can select a different category and see all the products that belong to this category(product listing)* | *Registered users* |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (SPA Frontend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Home** | Presents the home page with a set of a top products, footer, menu and search tab | / |
| * 1. **Product details** | Presents the information about product details | */product-details* |
| * 1. **Cart** | Presents a list of products which are already in the cart | */cart* |
| * 1. **Users** | Presents a list of users and some information about them | */users* |
| * 1. **Products** | Presents information about all products after filtering them by some way. | */products* |

|  |  |  |
| --- | --- | --- |
| 1. API Resources (Node.js Backend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Users** | GET *User Data* for all users, and POST new *User Data* (Id is auto-filled by *OKTS* and modified entity is returned as result from POST request). Available only for *Administrators*. | */api/users* |
| * 1. **User** | GET, PUT, DELETE *User Data* for *User* with specified *userId*. | */api/users/{userId}* |
| * 1. **Products** | GET all products and POST new *product* (Id is auto-filled by *OKTS* and modified entity is returned as result from POST request). Also, the endpoint supports query params to search by category or keyword (e.g. /api/products?category=Books&title=Introduction) | */api/products* |
| * 1. **Test** | GET, PUT, DELETE *Product Data* (for *Product* with specified *productId)*. | */api/products/{productId}* |
| * 1. **Cart** | GET and POST to get the products in the cart for an user with {userId} and add a product to his cart | */api/users/{userId}/cart* |
| * 1. **Remove from cart** | DELETE to remove a product from user’s cart*.* | */api/users/{userId}/cart/{productId}* |
| * 1. **Checkout** | POST to checkout | */api/users/{userId}/cart/checkout* |