|  |  |  |  |
| --- | --- | --- | --- |
| Course Project | | Online Food Order System | |
| Course | | Multimedia Technologies with Angular and TypeScript - 2024 | |
| Github Repository | | <https://github.com/shakevv/food-order-system> | |
|  | | | |
| Project author | | | |
| № | Name | | FN |
| 1 | Georgi Shakev | | 45692 |

|  |  |
| --- | --- |
| Project name | Online Food Order System |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| The Internet age changed our everyday life. The Online Food Order System (OFOS) provides ability for restaurant to sell food online, and for clients to order food directly to their home with a few clicks.  In addition to that it allows users to see the menu of the restaurant, filter the products by categories, use a cart functionality to prepare their orders and create new orders.  Restaurant can manage their products, categories, orders and staff accounts.  The system will be developed as a Single Page Application (SPA) using Angular as front-end and Node.js + Express as backend technologies. Each view will have a distinct URL and the routing between pages will be done client side using Angular Router. The backend will be implemented as a REST/JSON API using JSON data serialization. The main user roles (actors in UML) are:  • Anonymous User – can view Home page, which contains menu and about sections, Cart page and Order creation page.  • Staff (Registered User) – can view Home page, manage products, categories, orders and staff. |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Browse information pages** | *User* can browse information pages and sections (Menu, About, Login) | User |
| * 1. **Login** | *Anonymous User* can login in the system by providing a valid e-mail address and password. | *Anonymous User* |
| * 1. **Change User Data** | *Registered User* can edit his personal *User Data* and can view all users personal User Data. | *Registered User* |
| * 1. **Manage Users** | *Registered User* can register new Users, browse *Users* and choose a *User* to manage and remove his account. | *Registered User* |
| * 1. **Manage Products** | *Registered User* can add new *Products* and also can view and remove existing *Products.* | *Registered User* |
| * 1. **Manage Orders** | *Restaurant* can view all *Orders*, see details about them and change their status. *Anonimous User* can create new *Orders*. | *Registered User, Anonimous User* |
| * 1. **Manage Categories** | *Registered User* can manage *Categories* – add new or delete existing ones. *User* can filter *Products* by *Category*. | *Registered User, Anonimous User* |
| * 1. **Use Cart** | *Anonimous User* can manage his cart - add and remove items as well as he can see all cart details summarized. | *Anonimous User* |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (SPA Frontend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Home** | Offers ability to view the Home page – Menu and About sections. | /home |
| * 1. **Manage Orders** | Provides ability to see restaurant’s orders and manage them. | */orders* |
| * 1. **Create Order** | Provide ability to user to create his order. | */create-order* |
| * 1. **Cart** | Provide cart details*.* | */cart* |
| * 1. **Manage Users** | Presents a view allowing the *Registered Users* to view User data for other users, manage them and also update their own data. | */manage-users* |
| * 1. **Login** | Presents a view allowing the users to login. | */login* |

|  |  |  |
| --- | --- | --- |
| 1. API Resources (Node.js Backend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Users** | GET *User Data* for all users, Available only for *Registered Users*. | */users* |
| * 1. **User** | PATCH, DELETE *User Data* for *User* with specified *userId*. | */users/{userId}* |
| * 1. **Login** | POST *User Credentials* (e-mail address and password) and receive a valid *Security Token* to use in subsequent API requests. | */auth/login* |
| * 1. **Logout** | POST a logout request for ending the active session with *OKTS,* and invalidating the issued *Security Token*. | */auth/logout* |
| * 1. **Register** | POST *User Credentials* (e-mail address and password) and create new User profile. | */auth/register* |
| * 1. **Categories** | GET all *Categories* data andPOST new *Category*. | */categories* |
| * 1. **Category** | DELETE Category with specified categoryId. | */categories/{categoryId}* |
| * 1. **Products** | GET Products data and POST new *Product*. | */products* |
| * 1. **Product** | PATCH, DELETE *Product* with specified *productId*. | */products/{productId}* |
| * 1. **Orders** | GET *Orders* (according to *User's Role* and identity) for *User* with specified *userId*, and POST new *Order*. | */orders* |
| * 1. **Order** | PATCH *Order* with specified *orderId.* | */orders/{orderId}/* |