|  |  |
| --- | --- |
| Project name | Agro nursery online store |

|  |  |  |
| --- | --- | --- |
| Project author | | |
| № | Names of Participants | FN |
| 1 | Ivan Petrov | 3MI0700035 |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| Project Description: Agro Nursery Online Store  **The Agro Nursery Online Store** is designed to allow customers to purchase nursery products. This platform will enable users to browse a wide range of plants, seeds, and gardening supplies while providing a seamless shopping experience.  Key Features:   * **User-Friendly Interface**: Using ***html/template,*** ***tailwindcss, Alpine.js*** for the UI and interactivity. Customers can easily navigate through the product catalog, view detailed descriptions, and make purchases. * **Real-Time Customer Support**: Integrated chat support using ***gorilla/websockets*** allows users to connect with customer service representatives for immediate assistance. * **Secure User Registration**: Users can register, manage profile data, and track their orders. Using ***gorilla/sessions*** as an auth service on the server and DB * **Product Management**: Administrators can add, update, and manage products within the store efficiently. As well as see real-time statistics with ***e-charts*** * **Dynamic Routing**: The application will utilize **(TO BE DECIDED)** *gin // gorilla/mux // chi*for server-side routing, ensuring smooth navigation between product pages and main views * **Robust Database Management**: Leveraging ***sqlc*** and ***pgx*** for PostgreSQL database interactions, the platform will ensure efficient data handling and security. |

[[1]](#footnote-2)

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Browse Information** | The *User* can browse the information views (Home, Products, About) in *the system* and can choose to order products | All users |
| * 1. **Register** | *Anonymous User* can register in the system by providing a valid e-mail address, first and last name, and choosing password. By default, all new registered users have *Customer* role.  *Administrator* can register new users by entering *User Data* and choosing a Role (*Customer, Support*, or *Administrator*). | *Anonymous User, Administrator* |
| * 1. **Change User Data** | *Registered User* can view and edit their personal *User Data.*  *Support* can view User Data of all users  *Administrator* can view and edit *User Data* of all *Users* and assign them *Roles*: *Customer, Support*, or *Administrator*. | *Registered User, Support, Administrator* |
| * 1. **Manage Users** | *Administrator* can browse and filter users based on different criteria: first and last name, email, Role.  *Administrator* can choose a *User* to manage, and can manage the chosen User - edit (using Change User Data UC) or delete.  *Administrator* can create a new user using *Register UC*. | *Administrator* |
| * 1. **Manage Products** | *Administrator* can browse and filter *Products* based on different criteria: name, category*,* creation date, etc.  *Administrator* can view statistics about products  *Administrator* can create a new Product.  *Administrator* can choose a *Product* to edit and delete if necessary | *Administrator* |
| * 1. **Manage Orders** | *User* can browse their orders, add new *Order* using *Cart View*  *Administrator, Support* can browse *Orders* of all *Users*, update and delete them. | *User, Support, Administrator* |
| * 1. **Order a product** | *User* can add/remove items from their *Cart* and order them by processing their cart order. | *All users* |
| * 1. **Contact Customer Service** | *Registered users* may contact the support of the site for issues they came across  *Support and Administrators can answer users* | Registered users |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (Frontend) | | |
|  | **Brief Descriptions** | **URI** |
| * 1. **Home** | Show special offers and products catalog | / |
| * 1. **Products category** | Shows all subcategories of the product category | */products/{category}/* |
| * 1. **Products subcategory** | Presents all store products from the subcategory | */products/{category}/{subcategory}* |
| * 1. **Product** | View with full information for a product and option to order it | /product/{productId} |
| * 1. **Customer service** | Dialog chat window / Page view to contact customer support | */chat/{chatId}* |
| * 1. **User Registration** | Presents a view allowing the *Anonymous Users* to register in *the System* | */register* |
| * 1. **Login** | Presents a view allowing the users to login. | */login* |
| * 1. **User Data** | Provides ability to view and edit personal *User Data* | */profile* |
| * 1. **Users** | Presents ability to manage (CRUD) Users and their User Data (available for *Administrators* only, as described in UCs). | */users* |
| * 1. **About** | Presents information about the the project and his owner. | */about* |

|  |  |  |
| --- | --- | --- |
| 1. API Resources (Backend) **Not sure it’s needed as it’s server-side only app** | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Users** | GET *User Data* for all users, and POST new *User Data* (Id is auto-filled by *the server* 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*, according to restrictions described in UCs. | */api/users/{userId}* |
| * 1. **Login** | POST *User Credentials* (e-mail address and password) and receive a valid *Session Id* | */api/login* |
| * 1. **Logout** | POST a logout request for ending the active session*,* and invalidating the issued *Session*. | */api/logout* |
| * 1. **Products** | GET *Product Data* for all products, and POST new *Product Data* (Id is auto-filled by *the server* and modified entity is returned as result from POST request). Available only for *Administrators*. | */api/products* |
| * 1. **Product** | GET, PUT, DELETE *Product Data* for *Product* with specified *productId*, according to restrictions described in UCs. | */api/products/{productId}* |
| * 1. **Chats** | GET *Chat Data* for all chats, and POST new *Chat Data* (Id is auto-filled by *the server* and modified entity is returned as result from POST request). Available only for *Administrators*. | */api/chats* |
| * 1. **Chat** | GET, PUT, DELETE Chat Data *Chat* with specified *chatId*. | */api/chats/{chatId}* |
| * 1. **Messages** | GET *Message data* for all messages for specified *chatId* Chat, and POST new *Message data* (Id is auto-filled by *the server* and modified entity is returned as result from POST request). Available only for *Administrators*. | */api/chats/{chatId}/messages* |
| * 1. **Test Result** | GET, PUT, DELETE *Message data for Message* with specified *messageId* in specified c*hatId Chat* | */api/chats/{chatId}/messages/{messageId}* |
| * 1. **Orders** | GET Order *data* for all orders, and POST new *Order data* (Id is auto-filled by *the server* and modified entity is returned as result from POST request). Available only for *Administrators*. | */api/orders* |
| * 1. **Order** | GET, PUT, DELETE *Order Data* for *Order* with specified *orderId*, according to restrictions described in UCs. | /api/orders/{orderId} |

1. Can add like Manage subscriptions use case for automatic orders of materials [↑](#footnote-ref-2)