|  |  |  |  |
| --- | --- | --- | --- |
| Assignment 1 | | Project Summary | |
| Course | | Fullstack Application Development with Node.js + Express.js + React.js - 2024 | |
|  | | | |
| Project author | | | |
|  | Name | | FN |
|  | Roman Litvinov | | 6MI8000023 |

|  |  |
| --- | --- |
| Project name | HeatHub (“ЭлектроТерм”) |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| The internet is changing the world in many ways. The business sphere is no exception **The HeatHub (HB)** is an online shop, where customers can order heating elements for different household needs. It allows users to register, and administrators to communicate with customers via built-in chat. The system will be developed as a *Multi-Page Application (MPA)* using ***React.js*** 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 ***React Router***. The backend will be implemented as a ***REST/JSON API*** using JSON data serialization. There will be also a real-time event streaming from the server to the web client using ***Socket.IO*** for chat between customer and administrator.  The main user roles (actors in UML) are:  • *Anonymous User* – can view the information pages, items, blog of the owner etc.  • *Registered User* can buy items, comment posts in blog section.  • *Administrator* – can manage the shop (create, edit, delete) items, write posts in blog, manage users data (if it needed). |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **View information and products** | The user can view information about the store, product categories, as well as individual products. | All users |
| * 1. **Register** | *The user* can register in the system by providing the necessary data.  *As an administrator*, you can register a new user and assign them a role. | *Anonymous User,*  *Administrator* |
| * 1. **Change User Data** | *A registered user* can view and change their personal data.  *As an administrator*, you can view and edit user data. | *Registered User,*  *Administrator* |
| * 1. **Items** | *As an administrator* , you can add, edit, and remove products from your store.  As an *user* you can additems to chart, delete them.  As a *registered user* tou can buy items and leave comments under item | *Anonymous User,*  *Registered User,*  *Administrator* |
| * 1. **Manage Blog** | *As an administrator* , you can create, edit, and delete blog posts.  *The user* can view the blog and comments  *A registered user* can comment on posts | *Anonymous User,*  *Registered User,*  *Administrator* |
| * 1. **Manage Orders** | *As an administrator, you can* view and manage orders.  The *user* can save items to the shopping cart.  *A registered user* can save products to the cart, place orders, view them and track their status. | *Anonymous User,*  *Registered User,*  *Administrator* |
| * 1. **Chat** | *Users* can communicate with the administrators via the built-in chat. | All users |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (SPA Frontend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Home** | Presents the introductory information about the shop. Leftside are popular categories, Reightside are promotion and downside popular items. Here you can manage your account ( login/logout, cart overview, orders overview, profile settings ) | / |
| * 1. **Items View** | Presents *a* view allowing the users to see tubular heating elements | */tubular-heating-elements/:type* |
| Presents *a* view allowing the users to see *backup power supply items* | */backup-power-supply/:type* |
| * 1. **Dashboard** | Presents *a* view allowing the users to see and correct details about themselves. | */chat* |
| * 1. **Users** | Presents *a* view allowing the administrators to make a posts and users to comment them | */blog* |
| * 1. **About** | Presents information about the *shop* and his owner. | */about* |

|  |  |  |
| --- | --- | --- |
| 1. API Resources (Node.js Backend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Login** | POST *User Credentials* (e-mail address and password) and receive a valid *Security Token* to use in subsequent API requests. | */api/auth/login* |
| * 1. **Register** | POST *User Data* for *User* with specified *userId*, according to restrictions described in UCs. | */api/auth/register* |
| * 1. **Verification** | GET *User Data* while refreshing / first visit of website | */api/auth/me* |
| * 1. **User** | GET, DELETE, PUT *User Data.*  All requests are available for *all users.*  Mark: Only *Administrator* can delete other users. | */api/users*  */api/users/{userId}* |
| * 1. **Items** | GET *Items Data.* DELETE*,* POST and PUT requests are available only for *Administartor.* | */api/items/:type* |
| * 1. **Posts** | GET, POST, PUT, DELETE *Posts Data.* DELETE*,* POST and PUT methods are available only for *Administartor.* | */api/posts* |
| * 1. **Room** | GET, POST, DELETE *Room Data. Only Administator* can DELETE the room.  GET, POST methods via WS-protocol ara available for *ALL* Users | */api/chat/room*  */api/chat/rooms*  */api/chat/:roomId* |
| * 1. **Cart** | GET, PUT, *Cart Data*. *Registered* user can manage his/her cart in profile component | */api/cart* |