|  |  |  |  |
| --- | --- | --- | --- |
| Documentation | | Project Summary | |
| Course | | Fullstack Application Development with Node.js + Express.js + React.js - 2022 | |
|  | | | |
| Project author | | | |
|  | Name | | FN |
|  | Iskra Bozhkova | | 62410 |

|  |  |
| --- | --- |
| Project name | People for everything |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| In our busy everyday life, we need someone who can do some work for us just with one click. **People for everything** provides the ability for users to find a person for different type of things they need. Also, the system will allow other type of users (freelancers) to share their abilities and skills and find a job easily.  The system is developed as a *Single Page Application (SPA)* using ***React.js*** as front-end, and ***Express.js*** as backend technology. Each view has a distinct URL, and the routing between pages will be done client side using ***React Router***. For database it is used Mongoose library for MongoDB. The backend will be implemented as a ***REST API***.  The main user roles are:   * Unregistered user – only can view the main page and the description about the app * Registered user * User searching for help (extends *Registered user*) – can view different types of services, choose what they need and make an appointment * Freelancer (extends *Registered user*) – can describe what skills they have and what they can do for other people |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Home page** | All type of users can browse the information views – Home page and About. | All users |
| * 1. **Register** | *Unregistered user* can register in the system by choosing in which role they what to register and providing a valid e-mail address, first and last name, and choosing password and adding profile picture. | *Unregistered User* |
| * 1. **Login** | *Registered user* can login in the system after they are registered. | *User searching for help, Freelancer* |
| * 1. **Add skills information** | *Freelancer* can add information about them - their skills and abilities. | *Freelancer* |
| * 1. **View advertisement list** | *User searching for help and Freelancer* can view list with advertisements ordered by different categories. | *User searching for help,*  *Freelancer* |
| * 1. **Make an appointment** | *User searching for help and Freelancer* can view free hours of freelancers and make an appointment. | *User searching for help,*  *Freelancer* |
| * 1. **List of appointments** | *Freelancer* can view list of user’s appointments for their uploaded advertisements. | *Freelancer* |
| * 1. **Write comments for freelancers** | *User searching for help* *and Freelancer* can write comments under advertisement post of a freelancer. | *User searching for help,*  *Freelancer* |
| * 1. **Give rating** | *User searching for help and Freelancer* can give rating to given freelancer’s advertisement. | *User searching for help,*  *Freelancer* |
| * 1. **Add to Favorites** | *User searching for help and Freelancer* can add the freelancer they like to Favorites list. | *User searching for help,*  *Freelancer* |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (SPA Frontend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Home** | Presents the main page and introductory information for the purpose of the system as well as instructions how to start using it. | / |
| * 1. **Registration** | Presents a view allowing the users to register in the system. | */register* |
| * 1. **Login** | Presents a view allowing the users to login in the system. | */login* |
| * 1. **User information update** | Presents ability to change the user information - password. | */update-info* |
| * 1. **Advertisements list** | Presents a view allowing the users to see freelancers of some category and to choose one of them. | */advertisements* |
| * 1. **Profile** | Presents view allowing freelancer and users to see their profile. | */profile* |
| * 1. **Favorites list** | Presents list of freelancers that user is added to their favorites list. | */favs* |
| * 1. **Detailed Advertisement** | Presents the view allowing user to see detailed information about the advertisement and the freelancer. It gives the ability to users and freelancers to write comments, give rating and make an appointment. | */advertisements/:id* |
| * 1. **List of appointments** | Presents the view allowing freelancer to see all of the user appointments. | */appointments* |
| * 1. **About** | Presents information about the project and his owner. | */about* |

|  |  |  |
| --- | --- | --- |
| 1. API Resources (Node.js Backend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Users** | GET *User Data* for all users *User data.* | */api/users* |
| * 1. **User** | GET *User Data* for *User* with specified *userId*. | */api/users/{userId}* |
| * 1. **Registration** | POST *User Data* for *User* to register them in the system | */api/auth/register* |
| * 1. **Login** | POST *User Credentials* (e-mail address and password) and receive a valid *Security Token*. | */api/auth/login* |
| * 1. **Logout** | POST a logout request for ending the active session with *system* and invalidating the issued *Security Token*. | */api/logout* |
| * 1. **Appointments** | POST new user appointments. | */api/appointments* |
| * 1. **Appointment** | GET given appointment by its appointmentId. | */api/appointments/{appointmentId}* |
| * 1. **Advertisements** | POST new freelancers advertisements. | */api/advertisements* |
| * 1. **Advertisement** | GET given advertisement by its advertisementId. | */api/advertisements/{advertisementId}* |
| * 1. **Comments** | POST and GET comments | */api/comments* |
| * 1. **Comment** | GET given comment by specified commentId | */api/comments/{commentId}* |
| * 1. **Rating** | GET given comment by specified advertisement id | */api/comments/{advertisementId}* |
| * 1. **Favorites list** | POST favorites list. | */api/favs* |
| * 1. **Favorites** | GET given favorite freelancer by specified userId | */api/favs/{userId}* |