|  |  |  |  |
| --- | --- | --- | --- |
| Assignment 1 | | Project Summary | |
| Course | | Fullstack Application Development with Node.js + Express.js + React.js - 2021 | |
|  | | | |
| Project author | | | |
| F№ | Name | | Email |
| 82057 | Damian Ivanov | | damian.ivanovv@gmail.com |

|  |  |
| --- | --- |
| Project name | Project Manager Tool |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| The Project Manager Tool will be application for organizing everything around a certain project (Task, Comments, Deadlines, etc.). The system will be developed as a *Single Page Application (SPA)* 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*** and ***Server Sent Events (SSE)/WebSocket*** in order to allow the users to see the project state and all of the tasks in real time. The main user roles are:  • *Anonymous User* – can only view the information pages and simple examples.  • *User* (extends *Registered User*) – can join project and complete tasks.  • *Administrator* (extends *Registered User*) – can manage (create, edit user data and delete) all *Registered Users, Projects, Tasks* |
|  |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Browse information** | The *User* can browse the information views (Home, Projects, Tasks, About) | 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 *User* role.  *Administrator* can register new by entering *User Data* and choosing a Role (*User, PM(Project Manager)*, or *Administrator*). | *Anonymous User, Administrator* |
| * 1. **Change User Data** | *Registered User* can view and edit her personal *User Data* as well.  *Administrator* can view and edit *User Data* of all *Users* and assign them *Roles*: *User, PM*, or *Administrator*. | *Registered User, 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 Projects** | *Administrator* can browse and filter *projects* based on different criteria: name of *Project, name of the PM,* creation date, etc.  *Administrator* or *PM* can create a new *Project,* and assign a *Token Key* for automatic *User* enrollment  *Administrator* can choose a *Project* to manage, and can add or remove *Users* to the *Project,* or delete it if necessary. | *Administrator* |
| * 1. **Manage Task** | *PM* can browse all of the *Tasks for a Project*, add new using *Add/Edit Task UC*, and delete a *Task*.  *Administrator* can browse the *Tasks* of all *PM*, edit and delete them. | *PM, Administrator* |
| * 1. **Add/Edit** | *PM* or *Administrator* specifies/edits *Tasks* meta-data such as: Task name, subject, description, tags, and assign it to specific user or let it be picked. | *PM, Administrator* |
| * 1. **Complete Task** | *User* or *PM* browses Tasks available and can sort and filter them using different Task metadata fields. *User* or *PM* chooses a Task to complete. *PMT* presents the *Task Information to the User and the button for assign it to himself.* | *User, PM* |
| * 1. **Monitor Task Progress** | *PM* or *Administrator* monitors in real time the progress of *Users* on completion of Tasks. | *PM, Administrator* |
| * 1. **Browse Task Results** | *User* can browse her own completed *Tasks*.  *PM* can browse the *Tasks* created by her/him and for particular *User.* | *User, PM* |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (SPA Frontend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Home** | Presents the introductory information for the purpose of the system as well as detailed instructions how to start using it. Prominently offers ability to register. | / |
| * 1. **Projects** | Presents projects available according to *User's Role* and identity. Offers abilities to browse, choose, create, read, update, delete (CRUD) Projects, as defined by *UCs* (for *Administrators* and *Moderators* only). | */projects* |
| * 1. **Complete Tasks** | Presents Tasks available to particular User (according to what the *User* was assigned to) allowing him/her to choose a Task and complete it. | */complete-task* |
| * 1. **Completed Tasks** | Presents *Task Results* available according to *User's Role* and identity as defined by *UCs*. | */completed-tasks* |
| * 1. **User Registration** | Presents a view allowing the *Anonymous Users* to register in *PMT* | */register* |
| * 1. **Login** | Presents a view allowing the users to login. | */login* |
| * 1. **User Data** | Presents ability to view and edit personal *User Data*, enroll to more *User Projects* by presenting a *Token Key* for particular *Project*, as well as deregister from *PMT*. | */personal* |
| * 1. **User’s Projects** | Presents ability to browse, filter and manage *Users in Projects (*for *Administrators),* and assign Users to them (for *Administrators* only), as described in UCs. | */project-participants* |
| * 1. **Dashboard** | Presents in real time the Users’ progress on active Tasks, subject to Role restrictions described in UCs. | */dashboard* |
| * 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 *PMT* project and his owner. | */about* |

|  |  |  |
| --- | --- | --- |
| 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 *PMT* 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 *Security Token* to use in subsequent API requests. | */api/login* |
| * 1. **Logout** | POST a logout request for ending the active session with *RC,* and invalidating the issued *Security Token*. | */api/logout* |
| * 1. **Projects** | GET *Projects*, and POST new *Project* (Id is auto-filled by *PMT* and modified entity is returned as result from POST request), according to *User's Role* and identity security restrictions. | */api/projects* |
| * 1. **Project Info** | GET, PUT, DELETE *Project* (including assigned Users) for *Project* | */api/projects/{projectId}* |
| * 1. **Tasks** | GET users (according to *User's Role* and identity) and POST new *Task* (Id is auto-filled by *PMT* and modified entity is returned as result from POST request). | */api/tasks* |
| * 1. **Task** | GET, PUT, DELETE *Task Data* for *Task* with specified *TaskId*. | */api/tasks/{taskId}* |
| * 1. **Task Results** | GET *Task Results* (according to *User's Role* and identity) for *Task* with specified *TaskId*, and POST new *Task Result* (Id is auto-filled by *PMT* and modified entity is returned as result from POST request). | */api/tasks/{taskId}/results* |
| * 1. **Task Result** | GET, PUT, DELETE *Task Result* (according to *User's Role* and identity) for *Task* with specified *TaskId* and *Task Result* with specified *TaskResultId.* | */api/tasks/{taskId}/results/ {taskResultId}/* |
| * 1. **Task Comments** | GET,POST Comment about particular task, Available only for PM and the User assign to it | */api/tasks/{taskId}/comments* |
| * 1. **Active Tasks** | SSE event streaming of Users’ progress on currently active *Tasks* (separate event pushed for each update) according to *User's Role* and identity security restrictions. | */api/active-tasks* |