|  |  |
| --- | --- |
| Assignment 1 | Project Summary |
| Course | Fullstack with Node, Express and React |

|  |  |  |
| --- | --- | --- |
| Project author | |  |
| № | Pseudonym | Face-to-face/ online |
| 1 | ddd | face-to-face |

|  |  |
| --- | --- |
| Project name | WorkoutCreator |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| There are numerous resources on the Internet about leading a healthy lifestyle and working out, but every fitness newbie gets overwhelmed by the massive amount of information, contradicting advice and the countless ready-made exercise routines. WorkoutCreator’s goal is to provide an easy to use tool for creating a custom made workout plan, based on the user’s level, goals, available equipment and time. The user would be able to save the created workout plan, make changes to it if needed and delete it.  The system will be developed as a *Single Page Application (SPA)* using ***React*** as front-end, and ***Node.js + express*** as backend technologies. Each view will have a distinct URL. 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 exercises and signup * Registered User – can view and filter exercises, create new workouts, save workouts, edit their own saved workouts, delete their own workouts |

|  |  |  |
| --- | --- | --- |
| 2. Main Use Cases / Scenarios | |  |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| **2.1. Create a workout plan** | The user is able to create a workout plan. | *Registered User* |
| **2.2. Save a created workout plan** | After creating a workout plan, the user has the option to save the workout to ‘My Workouts’. | *Registered User* |
| **2.3. View ‘My**  **Workouts’** | The user can see a list of all of his saved workouts on the page ‘My Workouts’. | *Registered User* |
| **2.4. Edit a workout** | The user can edit (add or delete an exercise) his saved workouts. | *Registered User* |
| **2.5. Delete a workout** | The user can delete his saved workouts. | *Registered User* |
| **2.6. View exercises** | All users can view and filter the whole list of exercises. | *Anonymous* |
| **2.7. View exercise details** | The users can view details about each exercise. | *Anonymous* |
| **2.8. Sign Up** | The users can create a new account in the system, if they haven’t already. | *Anonymous* |
| **2.9. Log in** | The users can log in by providing correct credentials. | *Anonymous* |

|  |  |  |
| --- | --- | --- |
| 3. Main Views (SPA Frontend) | |  |
| **View name** | **Brief Descriptions** | **URI** |
| **3.1. Home** | Introductory page | */* |
| **3.2. Log In** | View that allows users to enter their credentials to log in. | */login* |
| **3.3. Sign Up** | Presents a view that allows users to register | */signup* |
| **3.4. Exercise** | Presents a view that allows users to get more information about a particular exercise. | */exercises/* |
| **3.5. Create Workout** | Presents a view that allows the user to create their own workout plan by choosing from existing exercises. | */create* |
| **3.6. Workout** | Allows the user to view a particular workout of his saved workouts. | */workouts/{{wor koutId}}* |
| **3.7. Profile** | Presents a list of all workouts of the logged in user. | */profile* |

|  |  |  |
| --- | --- | --- |
| 4. API Resources (Node.js Backend) | |  |
| **View name** | **Brief Descriptions** | **URI** |
| **4.1. Users** | GET all users, POST new user data (register) | */api/users* |
| **4.2. User** | GET, PUT, POST user data | */api/users/{{userId}}* |
| **4.3. Sign Up** | POST user credentials | */api/register* |
| **4.4. Login** | POST user credentials | */api/login* |
| **4.6. Exercises** | GET all exercises | */api/exercises* |
| **4.7. Exercise** | GET data about a particular exercise, based on the provided exerciseId | */api/exercises/{{exerciseId*  *D}}* |
| **4.8. Workout** | GET, POST, PUT, DELETE workout data based on the provided workoutId | */api/workout/{{workoutId}}* |