Assignment 1	Project Summary
Course	Fullstack Application Development with Node.js + Express.js + React.js - 2020

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

Project name	WorkoutGenerator

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. WorkoutGenerator's goal is to provide an easy to use tool for generating a custom made workout plan, based on the user's level, goals, available equipment and time. The user would be able to save the generated workout plan, make changes to it if needed and delete it. In addition to the custom-made workout plans, the users will have access to the full database of exercises, which could be filtered by body part, intensity, type.

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. The main user roles (actors in UML) are:

- Anonymous User – can only view and filter exercises
- Registered User – can view and filter exercises, generate 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. Generate a workout plan	The user is able to get a generated workout plan upon submitting his criteria: level, intensity, equipment, time, goal	*Registered User*
2.2. Save a generated workout plan	After the system has generated a workout plan based on the user's criteria, 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, Registered User*
2.7. View exercise details	The users can view details about each exercise.	*Anonymous, Registered User*
2.8. Register	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. Register	Presents a view that allows users to register	*/register*
3.4. Exercises	Presents a view that allows users to view all exercises and filter them by different criteria.	*/exercises*

3.5. Exercise	Presents a view that allows users to get more information about a particular exercise.	*/exercises/{{exerxiseName}}*
3.6. Generate Workout	Presents a view that prompts the user to answer several questions that would allow the system to generate an appropriate workout plan.	*/generate*
3.7. Workout	Allows the user to view a particular workout of his saved workouts.	*/workouts/{{workoutId}}*
3.8. Edit Workout	Allows the user to add or delete an exercise from a saved workout.	*/workouts/{{workoutId}}/edit*
3.9. My Workouts	Presents a list of all workouts of the logged in user.	*/workouts*
3.10. My Profile	Presents a view that allows the user to change his email or password.	*/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. Login	POST user credentials	*/api/login*
4.4. Logout	POST	*/api/logout*
4.5. Exercises	GET all exercises	*/api/exercises*
4.6. Exercise	GET data about a particular exercise, based on the provided exerciseId	*/api/exercises/{{exerciseIdD}}*
4.7. Generate	GET workout	*/api/generate*
4.8. Workout	GET, POST, PUT, DELETE workout data based on the provided workoutId	*/api/workout/{{workoutId}}*