

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

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

Project name	DRoutine
--------------	----------

1. Short project description (Business needs and system features)

Nowadays there are plenty of products and cosmetics which are being used by many people on daily basis for different reasons in order to take care of their skin health.

However, it's really hard to pick the right products and end up with the proper routine so that all products work in conjunction to resolve the persons' problems.

DRoutine is an application which helps you pick the right products for your skin type, mix multiple products together for better and faster results and essentially build a skin care routine for you, personally.

DRoutine allows users to add information about their type of skin and their problems, if they have any. The application will then pick the right products for the user based on their skin type and user's preferences on the products and will also make sure that the products can be used together in combination and if that's not the case, it will make a specific schedule what product to apply when. Anonymous users would only be able to check out the products and a few example routines.

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.

- *Anonymous User* – can only view all products which might be included in the routine.
- *Registered User* – can add their skin information, what they would like to achieve(heal acne, fight aging processes, etc) and optionally products they would like to try out, based on all of this, a personal skin care routine would be done
- *Support User* – he is responsible for viewing customers' requests and build personal skin care routines for each user
- *Administrator* – can manage all products data (create, edit and delete products), add description of the products, register/delete support users

2. Main Use Cases / Scenarios

Use case name	Brief Descriptions	Actors Involved
1. Browse products	The <i>User</i> can browse the skin products(Home page)	Registered User
2. Register	<p><i>Anonymous User</i> 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 <i>User</i> role.</p> <p><i>Administrator</i> can register Support users by entering Support <i>User Data</i> and choosing the Support User Role.</p> <p>Administrators are internally registered, no UI and API exposed for this.</p>	<i>Anonymous User</i>
3. Change User Data	<i>Registered User</i> can view and edit their personal <i>User Data</i> as well as their skin type data and products they would like to use.	<i>Registered User</i>
4. Change Products Data	<i>Administrator</i> can view and edit <i>Product Data</i> – add, edit, delete skin products	<i>Administrator</i>
5. Manage Users	<p><i>Administrator</i> can browse and manage Support Users – delete them, register new ones, edit information about them.</p> <p><i>Administrator</i> can create a new Support user using <i>Register Support User</i>.</p>	<i>Administrator</i>
6. Manage personal skin routines	<p><i>Support user</i> can browse all skin routines, the ones he created and the ones created by all other support users.</p> <p><i>Support user</i> can create new skin routines for different users, if a user request has come to his routines checklist. <i>Support user</i> can search through all requests for routines he has.</p>	<i>Support User</i>
7. Manage personal skin care routines status	<i>Administrator</i> can browse all routines created by the support users and update their status – ready for use, still in progress, to be configured.	<i>Administrator</i>
8. Add/Edit product	<i>Administrators</i> are able to add, edit or delete skin products	<i>.Administrator,</i> Registered user

	Registered users are able to add products to their “want to try out” list or remove a product from there.	
9. Add/Edit skin care routine	<i>Support user can create a skin care routine for a user, it can edit it or delete it.</i>	<i>Support user</i>
10. Add/Edit Skin progress	<i>User can add information about their skin type, they can update the information about their skin state or delete any invalid information</i>	<i>Registered User</i>
11. Complete skin care routine	<p><i>Any registered user who has been using a skin care routine can complete it once he decides to stop doing the routine.</i></p> <p>If the user goes inactive, the Support user has the right to complete and archive the skin care routine after 1 year of inactivity.</p>	<i>Registered user,</i> <i>Support User</i>
12. Track user’s progress	<i>Support users can track the progress of a user if the user updates their information accordingly.</i>	<i>Support user</i>

3. Main Views (SPA Frontend)		
View name	Brief Descriptions	URI
1. Home	Presents the introductory information for the purpose of the system.	/
2. Explore	Presents all products currently being offered	/explore
3. Enroll for a schedule	Presents all products from your lists and allows you to add more and make a request to enroll for a skin care routine	/schedule-routine
4. Check Active Routines and Track Results	Allows the user to update their skin care routine progress, track results and complete routine	/active-routines
5. User Registration	Presents a view allowing the <i>Anonymous Users</i> to register in the system.	/register
6. Login	Presents a view allowing the users to login.	/login
7. User Data	Presents ability to view and edit personal <i>User Data</i> . Allows users to add or edit information about their skin ad well.	/personal
8. Support User Routines	Presents the routines of the current support user, how many are in progress, how many are done and how many new routine requests are present.	/routines

9. Products Dashboard	Presents all current products available in the system and option to manage each of them. Administrator visible only.	<i>/products-dashboard</i>
10. Support Users	Presents ability to manage the Support Users of the system.	<i>/support-users</i>
11. About	Presents information about the <i>DRoutine</i> project and his owner.	<i>/about</i>

4. API Resources (Node.js Backend)		
View name	Brief Descriptions	URI
1. Users	GET <i>User Data</i> for all users. Available only for <i>Administrators</i> .	<i>/api/users</i>
2. User	GET, PUT, DELETE <i>User Data</i> for <i>User</i> with specified <i>userId</i> , according to restrictions deccribed in UCs.	<i>/api/users/{userId}</i>
3. Login	POST <i>User Credentials</i> (e-mail address and password) and receive a valid <i>Security Token</i> to use in subsequent API requests.	<i>/api/login</i>
4. Logout	POST a logout request for ending the active session, and invalidating the issued <i>Security Token</i> .	<i>/api/logout</i>
5. Products	GET Skin Products, and POST new <i>Skin Products</i> (Id is auto-filled and modified entity is returned as result from POST request)	<i>/api/products</i>
6. Routines	GET, PUT, DELETE Skin Routine for a <i>user</i>	<i>/api/users/{userId}/routines</i>
7. Support Users	GET support users and POST new <i>Support User</i> (Id is auto-filled and modified entity is returned as result from POST request). Available only for <i>Administrators</i> .	<i>/api/support-users</i>