|  |  |  |  |
| --- | --- | --- | --- |
| Assignment 1 | | Project Summary | |
| Course | | Fullstack Application Development with Node.js + Express.js + React.js - 2023 | |
|  | | | |
| Project author | | | |
|  | Name | | FN |
|  | Viktoria Cholakova | | 6MI0700110 |

|  |  |
| --- | --- |
| Project name | Fitness Challenge Tracker |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| ***The Fitness Challenge Tracker (FCT)*** is a web application designed to facilitate fitness challenges among users. It encourages a healthy lifestyle by allowing users to create, join, and participate in fitness challenges. The system will be developed as a Single Page Application (SPA) using React.js for the front end and Node.js + Express.js for the backend. The backend will expose a RESTful API for seamless communication between the frontend and backend.GitHub link: [Viktoria75/FCT: Fitness Challenge Tracker Application made with React (github.com)](https://github.com/Viktoria75/FCT) The main user roles (actors in UML) are:  • *Anonymous User* – can only access public information, view featured challenges, and browse the system without participating.  • *Challenge Participant* (extends *Registered User*) – Can log daily workout activities, track personal progress within a challenge, remove participants, view the challenge leaderboard.  • *Administrator* (extends *Registered User*) – can manage (create, edit user data and delete) all *Registered Users*, as well as *Challenges*. |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Browse information and view leaderboard** | The *User* can browse the information views (Home, Challenges, Leaderboard) in *FCT*.  *Users* can view the leaderboard for each fitness challenge, showcasing the top participants based on their overall performance. | 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.  *Administrator* can register new users by entering *User Data.* | *Anonymous User, Administrator* |
| * 1. **Change User Data** | *Registered User* can view and edit her personal *User Data.*  *Administrator* can view and edit *User Data* of all *Users.* | *Registered User, Administrator* |
| * 1. **Create Fitness Challenge** | *Registered User or Administrator* can create new fitness challenges by specifying details such as challenge name, duration, type of activity and goal.  This action gives them the role of *Challenge Creator*. | *Registered User, Administrator* |
| * 1. **Manage Fitness Challenges** | *Administrator* or *Challenge Creator* have the ability to manage their challenges by editing challenge details, removing participants, and closing the challenge once completed.  *Administrator* can choose a *Challenge* to manage, and can add or remove *Challenge Participants* to the *Challenge,* or delete it if necessary. | *Administrator, Challenge Creator* |
| * 1. ****Track Progress**** | *Challenge Participants* can log their daily workout activities and track their progress throughout the challenge duration. Real-time updates on fellow participants' progress are displayed.  *Administrator* can view any *Challenge Participant*’s progress. | *Challenge Participant, Administrator* |
| * 1. ****Join Fitness Challenge**** | *Registered Users* can join any fitness challenge. This action gives them the role *Challenge Participant*. | *Registered User* |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (SPA Frontend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Home** | Provides an overview of featured challenges, encouraging users to join and participate. Includes a section for recently completed challenges. | / |
| * 1. **Challenges** | Lists all available challenges with filter options based on type, duration. Users can join challenges from this view. | */challenges* |
| * 1. **My Challenges** | Displays challenges that the user has joined or created. Participants can log their daily activities and view the challenge leaderboard. | */participants/:userId* |
| * 1. **Leaderboard** | Shows the overall leaderboard for all challenges, highlighting the top participants based on their performance. | */leaderboard* |
| * 1. **User Registration** | Presents a view allowing the *Anonymous Users* to register in *FCT*. | */register* |
| * 1. **Login** | Presents a view allowing the users to login. | */login* |
| * 1. **User Data** | Presents ability to view and edit personal *User Data*, as well as deregister from *FCT*. | */user-profile* |
| * 1. **Create Challenge** | Allows users to create a new fitness challenge by providing necessary details such as challenge name, duration, goal, and type of activity. | */create-challenge* |
| * 1. **Challenge Details** | Displays detailed information about a specific challenge, including participants, progress, and the leaderboard. | */challenges/{challengeId}* |
| * 1. **Users** | Presents ability to manage (CRUD) Users and their User Data (available for *Administrators* only, as described in UCs). | */users* |

|  |  |  |
| --- | --- | --- |
| 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 *FCT* 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/users/login* |
| * 1. **Challenges** | GET Retrieve challenge data for all challenges.  POST Create a new fitness challenge (Id auto-filled by FCT, modified entity returned as result from POST request). | */api/challenges* |
| * 1. **Participants** | GET Retrieve participants for a specific challenge.  POST Add a new participant to the challenge (Id auto-filled by FCT, modified entity returned as result from POST request).  PUT Update participant data for a specific challenge.  DELETE Remove a participant from the challenge. | */api/challenges/{challengeId}/participants* |
| * 1. **Progress Tracking** | GET Retrieve progress data for a specific challenge.  POST Log new progress for a participant in the challenge (Id auto-filled by FCT, modified entity returned as result from POST request).  PUT Update progress data for a specific challenge and participant. | /api/challenges/{challengeId}/progress |
| * 1. **Leaderboard** | GET Retrieve the leaderboard data. | */api/challenges/leaderboard* |
| * 1. **View and Edit Challenge** | GET Retrieve challenge data for a specific challenge.POST Add yourself to a challenge  If you are *Administrator/Challenge creator*: PUT Update challenge details for a specific challenge.DELETE Delete an existing challenge. | /api/challenges/{challengeId} |
| * 1. **My Challenges** | GET Retrieve challenges for the authenticated user | */api/participants/:userId* |
| * 1. **Achievements** | GET: Retrieve achievements/results data.  POST: Create a new achievement/result (Id auto-filled by FCT, modified entity returned as a result from POST request). | */api/achievements* |