MERN - Final Project	
Project Name	*Exercise Tracker*
Duration	● You will have **60 hours** in total to complete this project.
General Objectives	● **Create** an Exercise *Tracker* web app from scratch. ● **Demonstrate** the technical and non-technical skills developed during the Junior Full Stack Developer program.
Details	● You will work in **groups of 2-3 people** as assigned by the instructor. ● This project has **10 tasks** (~3 hours each) divided into **3 Sprints**. ● Each Sprint will have a **demo** and a **retrospective** at the end following the Scrum methodology. ● At the end of the project, you and your group will do a **final project presentation** to the entire class and potentially to a group of employers.
Assessment	● In each task you will be assessed by the instructors on the “Assessment Criteria” in this rubric.
Materials	● All participant guides, assets, and possible solutions can be found here -> GitHub practice.
RR-FP - Sprint 1 - Task 1: Design your App Wireframes	
Session Objectives	● The objective of this task is to create the *Exercise Tracker* App wireframes to understand how it will work and look.
Assessment Criteria	1. Contains an Form with the required information (name, description, activity type [run, bicycle ride, swim, walk and hike], duration, date) 2. Contains an Activity Layout that displays the information (name, description, activity type [run, bicycle ride, swim, walk and hike], duration, date) 3. Wireframe solves all the UI challenges to represent the Activity Tracker App and a clear view of an activity with the required fields(name, description, activity type [run, bicycle ride, swim, walk

	and hike], duration, date)
RR-FP - Sprint 1 - Task 2: Implement your Wireframes using Bootstrap	
Session Objectives	● Implement the basic HTML structure of your Wireframes design. ● Create a private Github repository for your project that is shared with your instructor.
Assessment Criteria	● Activity Form fields are displayed in an organised way and with the proper label and input types. ● All input fields are mapped in the exercise activity form (name, description, activity type [run, bicycle ride, swim, walk and hike], duration, date) ● A Select type is used for the different activity types (run, bicycle ride, swim, walk and hike).
RR-FP - Sprint 1 - Task 3: Create an Exercise Activity Card layout and an Exercise Activities List component	
Session Objectives	● Implement the card layout bootstrap component that contains the exercise activity information: ○ Name ○ Description ○ Activity type [run, bicycle ride, swim, walk and hike] ○ Duration ○ Date
Assessment Criteria	● The Card Layout contains all the exercise activity information (name, description, activity type [run, bicycle ride, swim, walk and hike], duration, date). ● The List group organises multiple Card Layouts accordingly. ● The List group contains at least 5 exercise activities sample cards.
RR-FP - Sprint 2 - Task 4: Migrating to React Components: exercise activity form	
Session Objectives	For this task, we'll be migrating the code you created with Bootstrap to create a new exercise activity to become a React component using the state, style and properties as needed.
Assessment Criteria	● Classes and styling is migrated correctly using React. ● React component stores all exercise activity properties: ○ Name ○ Description ○ Activity type [run, bicycle ride, swim, walk and hike] ○ Duration

<table>
<tr><td></td><td>○ Date
● JSX is used correctly to map the structure created before with Bootstrap.</td></tr>
<tr><th colspan="2">RR-FP - Sprint 2 - Task 5: Migrating to React Components: exercise activities list</th></tr>
<tr><td><b>Session Objectives</b></td><td>For this task, we'll be migrating the code you created with Bootstrap to display the list of exercise activities to become a React component using the state, style and properties as needed.</td></tr>
<tr><td><b>Assessment Criteria</b></td><td>● You must implement a React component to represent an exercise activity.
● You must implement a React component to represent the list of exercise activities.
● JSX is used correctly to map the structure created before with Bootstrap.</td></tr>
<tr><th colspan="2">RR-FP - Sprint 2 - Task 6: Implementing the Exercise Activity tracker API with Express</th></tr>
<tr><td><b>Session Objectives</b></td><td>For this task, we'll write the code to create a new Express project to implement the REST API for the Exercise Activity Tracker App.</td></tr>
<tr><td><b>Assessment Criteria</b></td><td>● A new express project must be created and committed to a separate Github repository.
● Implement the different Routing to support the CRUD operations for the Exercise Activity Tracker App:
○ Read the exercise activities of a given user.
○ Read the information of a given exercise activity using its ID.
○ Create a new Exercise Activity.
○ Update an Exercise Activity using its ID and the new information to be updated.
○ Delete an Exercise Activity using its ID.
● The server can be started with no errors displayed.</td></tr>
<tr><th colspan="2">RR-FP - Sprint 2 - Task 7: Connecting with MongoDB database</th></tr>
<tr><td><b>Session Objectives</b></td><td>For this task, we'll write the code to connect your REST API created with Express with a MongoDB database instance using Mongoose.</td></tr>
<tr><td><b>Assessment Criteria</b></td><td>● All the CRUD operations must interact with the MongoDB database.
● Mongoose is installed and included as dependency on the application using NPM.
● Mongoose and the database setup is done properly and no errors occur when the application server is started.</td></tr>
</table>

RR-FP - Sprint 3 - Task 8: Connecting your Frontend with your Backend	
Session Objectives	In this task, we'll connect the two projects created so the React App can consume and use the Express API and the data is persisted on the MongoDB Database.
Assessment Criteria	● The React application consumes the REST API created with express using the fetch function to: ○ Read the exercise activities of a given user. ○ Read the information of a given exercise activity using its ID. ○ Create a new Exercise Activity. ○ Update an Exercise Activity using its ID and the new information to be updated. ○ Delete an Exercise Activity using its ID. ● The application works as expected and no error happens when you perform the CRUD operations.
RR-FP - Sprint 3 - Task 9: Deploying and testing your application	
Session Objectives	In this task, we'll deploy the final solution to the Cloud so you can share your project URL with the instructor and other classmates.
Assessment Criteria	● The React application should be deployed and accessible via a public URL over the internet. ● The Express API should be deployed and accessible via a public URL over the internet. ● The React application and express application must communicate and work as expected.
RR-FP - Sprint 3 - Task 10: Final Presentation	
Session Objectives	Present your final project.
Assessment Criteria	● The Exercise Activity Form fields use the correct input type ● The Activity Type Form input uses the select input type to prevent the user from entering the wrong data type. ● All form fields are validated (name, description, activity type [run, bicycle ride, swim, walk and hike], duration, date). ● A meaningful error message is displayed when a form field is invalid. ● A React component is used to represent an Exercise Activity. ● A React component is used to represent a list of Exercise Activities. ● The delete and update feature is consistent with the exercise activity card and delete the data stored on the MongoDB database.

	● The update feature updates both the exercise card and the data stored on the MongoDB database. ● The Tasks data persists using the MongoDB database via the Express API. ● The express API support all the CRUD operations: ○ Read the exercise activities of a given user. ○ Read the information of a given exercise activity using its ID. ○ Create a new Exercise Activity. ○ Update an Exercise Activity using its ID and the new information to be updated. ○ Delete an Exercise Activity using its ID.