myTECHCareers

React Frontend testing procedures – Timesheet record manipulation

## Prerequisites

1. Running database and API server => Consult README.md in Github repository for more detail on installation instructions.
2. Sample data from sampledata.sql file should be loaded into the database. Alternatively, multiple records will have to be generated manually by using a suitable client/admin tool (SSMS, dbeaver, …)
3. Being able to access the system at Super Admin or Staff level to create a Student Record for later access.
   1. Alternatively, email addresses in table Users can be adjusted so that the Google authenticated address you would like to use matches the desired user access level.
   2. For this purpose, there are three predefined users, “Ordinary User” with UserID 15, “Staff Member” with UserID 14 and “Super Admin” with the UserID 1.

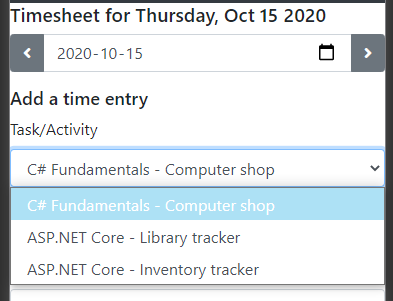
## Managing timesheet records

### Testing time constraints

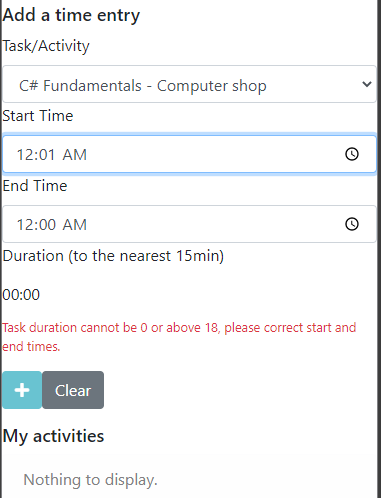
**Objective: Test that the task cannot be added to the database if its duration is 0h or more than 18h.**

Steps:

1. Ensure that the user intending to access the app will be using a Google email address associated with a student record (i.e. the UserID has an associated Student record in the database). This is important as Staff members do not have access to the student timesheet management page.
2. Open the site by accessing its URL in the browser.
3. After logging in using student-level credentials, use the site Navigation menu on the top right (click on your profile picture) and then click on timesheets.
4. Select a task/activity from the dropdown list. If no tasks are visible, change the date to October 15 2020 for which matching records exist in the Task table:



1. Select the same start time and end time for a particular task/activity.   
   (Note that on initial page load, the two times are equal and there is no error message reported, however the user is unable to add this task (+ button is disabled). This is by design to prevent immediate error reporting without any user input.)
2. User should not be able to proceed, and a validation message will be displayed (+ button is disabled – light green/turquoise in colour):

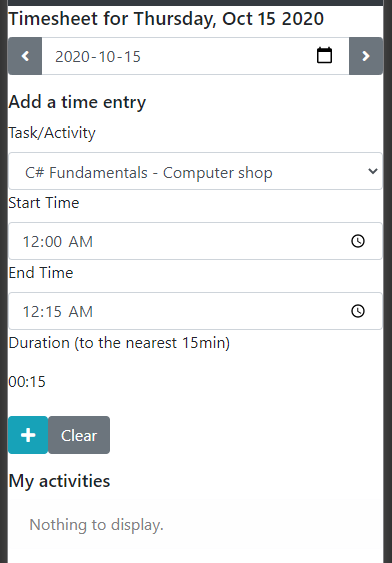


### Testing timesheet record addition

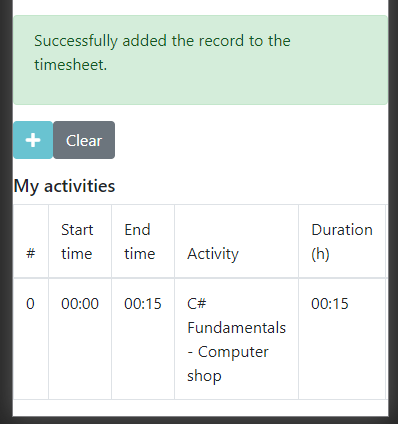
**Objective: Test successful addition of multiple timesheet entries.**

Steps:

1. As for the time constraints check, access the Timesheets page as the user with Student-level access.
2. Fill out the timesheet capture form with data that is within set constraints, such as:



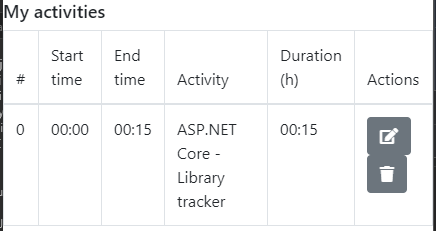
1. Note that there are no errors displayed, and the + (addition) button is not disabled. Click on the + button to add the record.
2. Feedback is displayed as soon as it is received from the API:



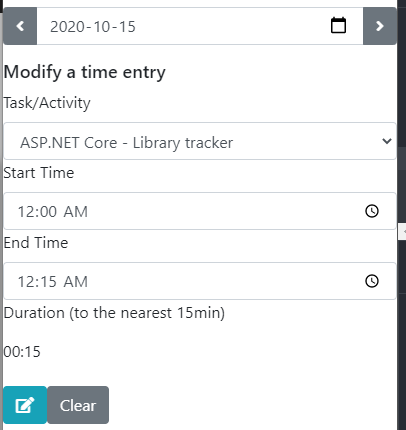
### Modifying a timesheet entry

**Objective: Test successful modification of a timesheet record**

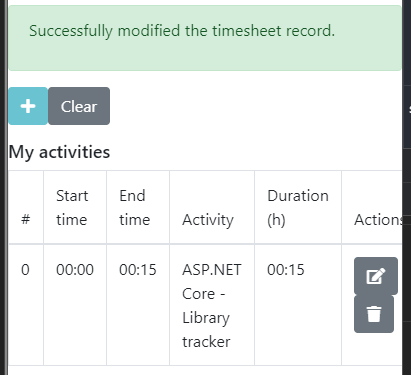
1. Using the sample entry from the previous example, click on the Edit Entry button in the Actions column of the My activities table:



1. Selected entries will appear in the entry fields on the top of the page. Select a different task from the initial one, and click on the button with the Pencil icon to save changes:



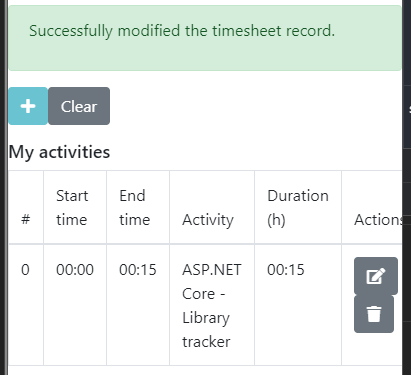
1. Check that the following message is displayed by the system and the My activities table now reflects the newly chosen data:



### Deleting a timesheet entry

**Objective: Demonstrating the ability to delete a record from the database**

1. Using the previous test as a starting point, click on the Delete Entry button shown in the Actions column of the My activities table:



1. Verify that the record is deleted from the database:

