FH TECHNIKUM WIEN University of Applied Sciences

BIF2 - Web Frameworks - MEAN

What is the second attempt project?

- At the second attempt, you’re going to implement different stages, based on the part you’ll have to do.
- Goal is to create a time/room tracking app, with Login, Register, TimeTracking, Overview, … components.
- This app should be similar, to the currently used COVID time tracking apps, e.g. when visiting a restaurant.
- Angular Material provides all necessary components, to build this project. No need to look for additional controls.
- Please organize yourself in groups of **two**.
- Use the Moodle forum, when encountering problems.

Prerequisites

- You can reuse everything done during the lectures, this is basically the login/signup part in Angular and the backend part in Node.js/Express.js.
- Make sure to check Moodle for existing videos, to get started with Angular again.
- Reusing code from the semester project is also allowed.
- Please make sure to read the instructions carefully, to only implement the part needed for you.
- Various room data is provided via a JSON file in Moodle. If you don’t want to use that, you’re eligible to create your own room descriptions JSON file. But make sure to follow the structure of the JSON file given.
- The provided InMemory-DB is based on the Puzzle-Game project. You’ll have to change it, to fit your needs for this project.

Prerequisites

- Based on your current grading of the practical course part, the implementation part is split into the following:
  - In case of only (!) a negative exercise part, please implement **only Part 1** (see slide 10 & 11) of the given project and create a protocol (see slide 15).
  - In case of only (!) a negative project part please implement the **specific parts of the project (Part 1 + Part 2)** (see slide 10 & 11 & 12 & 13) and create a protocol (see slide 15).
  - In case of a negative exercise part **AND** a negative project part, please implement the **whole project (Part 1 + Part 2 + part 3)** (see slide 10 & 11 & 12 & 13 & 14) and create a protocol (see slide 15).

What to do?

- Create a time/room tracking app, where users can register, login, logout and track their time for staying in a certain room.
- Provide an overview over all tracked time data.
- Add a nav bar (either on top or on the left side) to cover login, signup, logo, FAQ, overview, track time, … and link the buttons to the corresponding pages.
- Implement the different parts of your Time-Tracking Website and make sure to test them.
- Fill parts without logic (FAQ) with realistic text. Make sure to search online for good examples.
- Create a protocol and add it to the Moodle upload.

Core Functionality

- **DON‘T** use HTML-Tables as layout elements for the pictures. Use CSS or Angular components.
- The HTML site should **NOT** contain any static elements needed for the game (layout, pictures, ...). Only necessary buttons and labels. Make sure to load content dynamically via TypeScript logic.
- **DON‘T** use external npm packages or JavaScript libraries for Non-UI tasks, like login, signup, etc...
- **DON‘T** use the innerHTML property, use DOM methods for creating/removing of necessary elements.

Core Functionality

- Check if a user visiting the site is already logged in. (hint: create a service for this!)
- If a user is currently logged in, the signup button should not be displayed.
- If a user is currently logged in, the login button should change its caption to logout. When the logout button is clicked, the user will be logged out.
- If a user is currently logged in, display a button „Overview“ which links to the users profile.
  - The “Overview” site shows information about the user, like username, additional information and the currently tracked time (room numbers + time spent in that room) as a table.
- Create a logo and always show it on the upper left.

Core Functionality

- If a user is currently logged in, display a button „Track Time“(in the navbar) which links to a page, where the user can create time tracking records.
  - The “Track Time” site provides functionality, where the users can choose a room from a **dropdown**, a date from a **datepicker** and time values via **input** (with type=“time”) and create a “Check-In” when clicking a submit button.
- After “Check-In” the “Track Time” page should only display a “Check-Out” button, to make sure the user is only in one room.
- After the user is leaving a room and clicks on the “Check-Out” button, the time tracking data (e.g. room number, username, time in seconds) should be saved in an InMemory-DB.
- Implement a timer or similar logic, to check how long the user was in a room.

Core Functionality

- Only logged in users can see the “Track Time” and “Overview” buttons in the navbar.
- Create an “Overview” page, to display all logged data from the currently logged in user. (hint: use Angular Material tables for this: https://material.angular.io/components/table/examples)
- Create an “FAQ” page (get information about how this may look on the web) and fill it with realistic text samples. (hint: use Angular Material expansion panel for this: https://material.angular.io/components/expansion/overview)
- Make sure to display at least 10 time tracking records in your “Overview” page.

Implementation - Part 1

- Create the Time-Tracking App webpage with Angular, Angular Material and Node.js/Express.js.
- Implement a working Login/Logout and SignUp functionality.
- Make sure, the navbar button visibility is correct. (Overview, Track Time, Logout, etc…)
- Use the provided InMemory-DB for handling user data.
- The usage of MongoDB is allowed, but not necessary. Using the provided InMemory-DB is sufficient.
- Create a logo and always show it on the upper left.

Implementation - Part 1

- Create an “Overview” page, to display an table with all logged time data from the currently logged in user. For **Part 1**, it’s enough to only show fake data.
- Create an “FAQ” page (get information about how this may look on the web) and fill it with realistic text samples.
- Create a “Track Time” page with fake controls (buttons, date picker, input). No logic required, but make sure it looks realistic!

Implementation - Part 2

- Implement the Time-Tracking functionality of **Part 1**.
- A user can choose a room from the dropdown. The data of the dropdown is based on the room data JSON provided via Moodle. (hint: use Angular AutoComplete for the room selection - https://material.angular.io/components/autocomplete/examples)
- For the time tracking form use a combination of Angular Material AutoComplete, DatePicker, Input (of type="time", to get the time selection) and Button to get all necessary data.
- Save the Time-Tracking data in the In-Memory DB. You can use the provided DB and change it according to the requirements, or implement your own InMemory-DB.

Implementation - Part 2

- Implement the Overview functionality (hint: use Angular Table for this - https://material.angular.io/components/table/examples)
- The table should at least display the following data (data should be provided by the InMemory-DB):
    - Room Name
    - Check-In time
    - Check-Out time
    - Time spent in room

Implementation - Part 3

- Extend the table functionality of **Part 2**
  - Make sure, to use pagination.
  - Make sure, to make the table rows are sortable (sort them by clicking on the header).
  - Implement functionality to select rows and delete them.
  - Make sure, to delete the rows from the InMermory-DB too!

Protocol

- Add a protocol as **PDF** with the following content:
  - protocol about the technical steps you made (designs, failures and selected solutions). Describe which Angular Material components you used and where problems occurred.
  - explain how your application works and describe the process from login, to time tracking, to overview, to logout with screenshots and text.
  - track the time spent with the project.

Upload to Moodle

- Upload a ZIP file with your UID into the moodle course (submission can be found at section **Wiederholungsprüfung**) **BEFORE** the end of submission!!!
- Give the ZIP file a name following this schema: webframeworks-timetracking-ifxxbxxx.zip (e.g. webframeworks-timetracking-if20b001.zip)
- Make sure that everything (except the **node_modules** folder!!!) is within the ZIP file.
- The protocol should be part of the ZIP file too!!!

Grading - Part 1

- If implementing only **Part 1**, the project will be graded with 24pts.
  - Login (Authentication token, logout, …), signup -> 4 pts
  - Overview page, Time Tracking page, site navigation -> 4 pts
  - InMemory DB attachment -> 4 pts
  - good UI styling (usage of material libraries, useful design & theming [colors]) -> 4pts
  - Logo, "realistic" FAQ -> 4pts
  - Protocol -> 4pts

Grading - Part 1 & Part 2

- If implementing **Part 1 & Part 2**, this project will be graded with 40pts.
  - Design & implementation of core functionality (time tracking) -> 8 pts
  - Login (Authentication token, logout, …), signup -> 4 pts
  - Overview page, Time Tracking page, site navigation -> 8 pts
  - InMemory DB attachment -> 8 pts
  - good UI styling (usage of material libraries, useful design & theming [colors]) -> 4pts
  - Logo, “realistic” FAQ -> 4pts
  - Protocol -> 4pts

Grading - Part 1 & Part 2 & Part 3

- If implementing **Part 1 & Part 2 & Part 3**, this project will be graded with 45pts.
  - Design & implementation of core functionality (time tracking) -> 8 pts
  - Login (Authentication token, logout, …), signup -> 4 pts
  - Overview page, Time Tracking page, site navigation -> 8 pts
  - Extended overview table functionality -> 5pts
  - InMemory DB attachment -> 8 pts
  - good UI styling (usage of material libraries, useful design & theming [colors]) -> 4pts
  - Logo, “realistic” FAQ -> 4pts
  - Protocol -> 4pts