

- 1) **Project intro.** Review documents, initial project repository. Pass react official tutorial & create tic-tac-toe game with some of your additional features.
- 2) **Pages creation - 1 part.** Create 'members', 'member progress', 'members tasks' pages. Use json files/ local storage or [firebase](#) as data base, local storage as data cache, you can use [Faker](#) f.e. for fake data generator for your data base. **All component must be created by hands, do not add any additional library to the project. Create your own spinner.**
- 2) **Pages creation - 2 part.** Create 'tasks', 'tasks tracks' pages. Create common **header, side-bar & footer**. Add page with login/logout forms for future integration with back-end API. **All validation should be created by hands, apply your own idea how to validate sensitive data fields in forms.**
- 3) **Roles & Validation.** Add roles: 'ADMIN', 'MENTOR', 'USER'. Rewrite common components using react-bootstrap/reactstrap package. Add validation to all necessary filed in the forms, for reactstrap use [avaiility-reactstrap-validation](#) package.
- 4) **Back-end API integration - 1 part.** Contact with back-end developer, connect members, member progress & members tasks page to back-end API. Figure out how to start it and work with it accordingly. Rewrite state and data fetching in members page with redux, react-redux, redux-thunk packages. Use [axios](#) for data fetching.
- 5) **Back-end API integration - 2 part.** Connect tasks, task track pages to back-end API. Rewrite state and data fetching in tasks page with redux, react-redux, redux-thunk packages. Create awesome README with [readme-md-generator](#) package. Create loader component by yourself like PulseLoader or DotLoader. Use as a reference [react-spinners](#) package.
- 6) **Hooks experience.** Use react hook API (useState, useRef, useContext, useReducer, useEffect) in 2 places. Add 'about application' page.
- 8) Demo