1) **Project intro.** Hangouts & initial project repository review. Passing through react official tutorial. Repeating tic-tac-toe game tutorial with some of your additional features.

2) **Pages 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. as fake data generator. **All component must be created by hands, do not add any additional library to the project. Create your own spinner.**

3) **Pages 2 part.** Create ‘tasks’,’ tasks tracks’ pages. Create common **header, side-bar & footer components**. Add page with login & logut forms for future integration with back-end API. **All validation should be created by hands, apply your own validation approach to sensitive data fields in forms.**

4) **Roles & Validation.** Add roles: ‘ADIMN’, ‘MENTOR’, ‘USER’. Rewrite common components using react-bootstrap or reactstrap package. **Add validation to all necessary fields in the forms**, for reactstrap use availity-reactstrap-validation package.

5) **Back-end API integration 1 part.** Contact with back-end developer, connect members, member progress & members tasks pages to back-end API. Figure out how to start 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.

6) **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.

7) **Hooks experience.** Use react hook API (useState, useRef, useContext, useReducer, useEffect) in 2 places. Add ‘about application’ page.

8) Demo in English with complete application.