Frontend Development Assignment

Dream Journal

Domain

The Dream Journal is an application where a user can view, add, edit or delete dreams.
The Dream is an entity with next fields**:**

- **Title -** `String`
- **Description -** `String`
- **Date -** `String`
- **Type -** `Enum(happy | sad | exciting | scary)`

An app should provide three pages (Add, View, Edit Dreams) for the user to manage his or her dreams. Dreams should be saved inside the user's browser (redux state) and they should appear if he or she revisits the web page.

Tasks

1. Create an empty React.JS project with a UI library configured
2. Add redux to the project alongside all configurations needed for it (e.g. persistor etc...)
3. Create a "Add Dream" page with all necessary input fields and validations
4. Create an "View Dreams" page that displays dreams saved inside Redux
5. Create an "Edit Dreams" page that will allow a user to edit a specific dream and save it
6. Create a delete functionality on a "View Dreams" page which will enable a user to delete a certain dream
7. Create import/export .json functionality on the "View Dreams" page. Export all dreams saved inside the redux store as a single .json which can be imported later at any time.
8. Create pagination on the "View Dreams" page

Note: All tasks should be done in order as they are presented.

Tech Details

- You can choose between plain Javascript and Typescript.
- For the UI library, you can choose the one you're most comfortable with. ChakraUI or React Bootstrap would be our suggestions.
- All dreams should always be inside the redux store. Use standard redux data flow for storing and fetching.
- The project should use only functional components. You can freely use custom hooks, there are no limits in regards to that.
- You can include any dependency (e.g. momentJS for date-time manipulation).
- You can assume that the app should always start in the dev mode (`react-scripts start`)

Deliverables

- **Public** code repository which should include
  - Source code
  - README.md file with instruction on how to start the app
  - ASSIGNMENT.pdf (this very file)

Notes

- The assignment is a one-day activity. The solution should be submitted before midnight, and any commits to the repository made after that period will not be considered. Even though it can be a whole-day activity, we highly encourage working on the assignment within the 8 hour time scope and pushing commits to a remote repository in real-time as you complete each task.

- Each task should be a new commit. This will allow for partial completion of the assignment which will still be valuable.
  Example: Finishing task #2 and not finishing task #3 will still provide value. On the other hand, combining them and ultimately having them unfinished provides almost no value.

- You should not be discouraged if you do not finish all tasks - that is pretty much expected since the number of tasks and their complexity is not balanced with the timeframe allowed for the completion. The assignment is intentionally challenging to

allow a better understanding of your current capabilities and skills. The **quality** of completed tasks will be of primary concern, the number of completed tasks of secondary concern, and the look and feel of the pages *tertiary* concern.

- If something is not clearly defined within this document, use common sense to deduce the correct approach which will produce a solution that matches the constraints of the assignment the most

- Use a public Github repository as a remote.

- Access to the repository should be given to user ***@Milutin-P***

- If you are satisfied with the solution you provided, we highly recommend to keep (and improve if needed) your solution on your Github profile whatever the outcome of the position we are offering