

Project Name - Real Time To do List

(you can also give it a name of your choice)

Deadline - 15 Days

Project Description -

This project is aimed to create a ready to deploy Live TODO List management system. It must have all the features mentioned below and it must be deployed on a server before submission. There should be two separate parts of the application. A Frontend developed and deployed using the technologies mentioned below and a REST API (with realtime functionalities) created using the technologies mentioned below.

Frontend Technologies allowed - HTML5, CSS3, JS, Bootstrap and Angular

Backend Technologies allowed - NodeJS, ExpressJS and Socket.IO

Database Allowed - MongoDB and Redis

Features of the Application -

1) User management System -

- a) Signup - User should be able to sign up on the platform providing all details like FirstName, LastName, Email and Mobile number. Country code for mobile number (like 91 for India) should also be stored. You may find the country code data on these links (<http://country.io/phone.json>, <http://country.io/names.json>)
- b) Login - user should be able to login using the credentials provided at signup.
- c) Forgot password - User should be able to recover password using a link or code on email. You may use [Nodemailer](#) to send emails. (Please use a dummy gmail account, not your real account).

2) To do list management (single user) -

- a) Once user logs into the system, he should see an option to create a ToDo List
- b) User should be able to create, a new empty list, by clicking on a create button

- c) User should be able to add, delete and edit items to the list
- d) User should also be able to add sub-todo-items, as child of any item node. Such that, complete list should take a tree shape, with items and their child items.
- e) User should be able to mark an item as "done" or "open".
- f) User should be able to see his old ToDo Lists, once logged in.

3) Friend List -

- a) User should also be able to send friend requests, to the users on the system. Once requests are accepted, the friend should be added in user's friend list. Friends should be Notified, in real time using notifications.

4) To do List management (multi-user) -

- a) Friends should be able to edit, delete, update the list of the user.
- b) On every action, all friends should be notified, in real time, of what specific change is done by which friend. Also the list should be in sync with all friends, at any time, i.e. all actions should be reflected in real time.
- c) Any friend should be able to undo, any number of actions, done in past. Each undo action, should remove the last change, done by any user. So, **history of all actions should be persisted in database**, so as, not to lose actions done in past.
- d) Undo action should happen by a button on screen, as well as, through keyboard commands, which are "ctrl+z" for windows and "cmd+z" for mac.

- 5) **Error Views and messages** - You have to handle each major error response (like 404 or 500) with a different page. Also, all kind of errors, exceptions and messages should be handled properly on frontend. The user should be aware all the time on frontend about what is happening in the system.

A few important conditions-

- 1) Design of the application must be done entirely by you from scratch using Bootstrap. You are not allowed to use templates that are already available on internet. Use of such templates will be considered plagiarism and your submission will be rejected.

2) You must follow all the industry practices taught to you in the training. You have to structure your application in the same way as taught in your training and write all the components, services, middlewares, libraries etc yourself.

3) Rate limiting - Your APIs should have pagination or rate limiting to avoid send huge chunks of messages as API response.

Evaluation Basis

This project will be evaluated on following basis -

- 1) **Quality of JavaScript code** - Your application's Javascript code should be optimized to be readable with proper indentation and comments. It should be broken down into functions for better maintainability and it should not contain any logical bugs. It should use modern javascript as much as possible.
- 2) **Responsive Design** - Your website should be full responsive. It will be checked thoroughly for all the things that have been taught to you in the level.
- 3) **Intuitive Thinking** - You have thinking intuitively and make the platform as easy to understand as possible. You have think about all the possible error cases and you have to handle them by giving alert messages to user. You must use elements like progress bars and loaders to handle the UX better.
- 4) **Originality of code** - Your code will be checked for plagiarism and if it's not original, it will be discarded with a negative skill score.
- 5) **Quality of Frontend application(Angular application)** - All the best practices associated with Angular must be followed.
- 6) **Quality of Backend application** - All the best practices associated with Advance rest API development must be followed.

- 7) **Documentation** - Documentation should be done properly as taught to you in training. Documentations of REST API endpoints and Socket Endpoints should be separate.
-

Deliverables from Candidate

- 1) Project description - A TXT/Doc file containing the description of your project and all your assumptions. It should also describe the features of the project and also any extra features that you have coded to get extra marks for intuitive thinking.
 - 2) Github repository link of this project should be mentioned in the TXT file.
 - 3) You have to host the built versions of the applications on AWS and mention the URL of that application in the TXT file. URL of both Frontend and REST API must be mentioned along with documentation link(if any).
-

Always remember these evaluation basis, and your deadline. And your aim is to meet the deadline.

Warning - Do not submit incomplete projects or projects that are not running. They will result in negative skill score. Do not seek help from your peer learners or anyone else. Your code will be checked for plagiarism at multiple stages and if you are found to be guilty of cheating, your submission will be discarded.