Autumn Assignment

2023

— Project Management App

Information Management Group

For- Developers

Overview

IMG, as the most prominent tech group of the institute, manages a lot of large applications such as the institute main website, the channel-i portal and all the applications that come with it. As you already know, we also actively develop new applications/enhance existing

ones to meet the campus needs. Since we work on so many apps concurrently, keeping track of the current status and goals for each application is a hefty task. Currently, management of the tasks and future goals related to an application is done either verbally by each team, or by verifying with the application design every now and then. Even in our General Body Meetings, the track of projects is kept using Google Docs. As part of your Autumn Assignment, you are going to build an in-house application for managing the tasks and future goals related to a project, specifically catering to IMG.

Goals

Account Creation and Log In

The user must be able to register and log in using the Channel i OAuth. Only the IMG Members (Maintainers) [The OAuth will provide this data] must be allowed to use your app.

Authorization

There are two roles the users must be categorized in, namely, Admins(should not be made by changing the default admin panel of Django), Normal Users. Normal users can do tasks like creating a Project Board, creating lists for the project, adding cards to lists, assigning cards to a project maintainer etc. Admin can do whatever Normal Users and can disable the users (A disabled user shouldn't be able to log in again), change the role of a user from Normal to Admin and vice-versa. Admin can perform any action in the app.

Add/ Edit/Delete Project

Anyone can create a project to start progress tracking. Some required fields for the project can be a name, a wiki, and team members. Wiki can be used to add a basic description of the project. The creator of this project can now add more team members in the project. Now, the creator, team members and the admin can edit the project details (Wiki, team members and other editable fields) and even delete the project.

Suggestion: You may use RichTextField for the Wiki.

Add/ Edit/Delete Lists and Cards

Team members and admins can add/edit/delete lists to projects. Each list must have a name and will contain one or many cards. Each card can have a title (short description), detailed description and assignees (if any).

My Dashboard

There must be a page where the user can see the cards assigned to them and the projects they are part of.

Members (For admins only)

The admin view allows users to see all the members with related details and perform tasks restricted only to admins.

Commenting (Additional Feature)

Anyone must be able to comment under a card. This can help in the discussion on the feature/ task. The commenting system must be real-time (with the help of Websockets).

Search (Additional Feature)

Anyone must be able to search for a card that is created under a list, in cases where you have a lot of cards created, it will be quicker to access the cards quickly (even when you want to look up the resolved cards of a user, or in a project dashboard).

Specifications

I. Database Architecture

Use either PostgreSQL or MySQL for storing your data and not SQLite. SQLite is a file system database and hence is not suitable for storing data in production. Make sure to do this.

II. API with Django Rest Framework:

Tech stack: Python; Django, Django REST framework

This will be the framework that should be used to write the backend part of your application. You can find all the information and tutorial about this framework at this link. You need to use ModelSerializers and ModelViewSets for making your API. Keep this in mind when developing the app. The permissions must be in a separate file named permissions.py and try to reduce code repetition.

III. Frontend with React-Typescript:

Tech stack: React-TypeScript, Redux

For the frontend of the application, you will be using React.js. Since most of you are new to it, it is recommended to first learn the basics of React by going through the

official documentation. The official documentation is more than enough for you to be able to complete this application. The benefit of using React over conventional HTML and javascript will be clear once you start learning and using it. Bootstrap is for kids; you can check out Material UI or Tailwind CSS.

IV. Search:

Tech Stack: Elastic Search

For efficient search, rather than coming up with something from scratch, you can use ElasticSearch, which would provide you the feature for indexing at and efficient searching across multiple indexes that you create for the pages that will be created on your application. You can refer to this for more details, and ask your mentors for help.

Milestones

Since this is your first full fledged project, it becomes essential that we evaluate and help you at times. Milestones will also help you to plan things and work more efficiently. Below are tentative dates for you to report your progress to your mentors. In case of doubt, you can contact any of the final or pre-final years.

I. DB Structure and Screens (21st September):

This is the very first and foremost step in your app development. You need to list down all the features in a document. Next, decide the database structure. For this, it is advisable to make ER diagrams; a verbal explanation will work as well :p. For better understanding, you can make the hand-drawn layout of the screens that you plan to build. All this has to be shared and will be reviewed by second and third years.

II. APIs / Backend (2nd October):

Now comes the backbone of your app, the APIs :'). Read up on the Django Rest Framework. Needless to say, your code should be git tracked, and you will be evaluated on the progress that you make coding your API. It is expected that you follow all the specifications mentioned above strictly.

III. Frontend (20th October):

The most interesting and probably the most challenging part of your app. Once you are ready with your APIs, you should get started with the frontend. You are free to

decide the design of your apps and make sure that it is simple to use. The hand-drawn screens made in the 1st part will come in handy and try to build on them. For inspiration, you can look at various bug tracking apps online.