Contents

Chapter: 1 Introduction to Project Management Tool

1.1 Definition

Project Management Tools or Software are the solution for the one who wants to manage complex projects in team that requires appropriate planning and synchronized progress among team members to efficiently utilize time and human resource, while increasing the clarity in project progress and quality of the project work. Nowadays such tools are in primary needs of Organization to gain such ease in their projects.

1.2 Purpose and Objective

Seamless Collaboration: Around the globe in a click as they are cloud-based applications.
Task management: Create and assign tasks, deadlines and status reports.
Tracking time: Track time spent on tasks and export timesheets in a project management tool.
Team communication: Communicate with members easily with timely responses to urgent inquiries.
Team Productivity and Accountability: Brings clarity about what exactly one has to do and responsible for.
Easy Documentation: Task specific report files as well as other project plan related files can be managed within app cloud storage or even in your personal third-party drives such as gdrive or dropbox if integrated.
Track Progress: The main goal is to know the status of the project progress every now and then in order to meet the deadlines.

1.3 History

The term **"Project Management" and related techniques** were introduced after 1954 in US Air Force for military purpose and then after gained attention for business purposes. The Famous **CPM** technique was introduced in 1957 and **PERT** in 1958. In 1965, the more managed technology **WBS** is presented and then after inspired by it, in 1970 **Waterfall Method** is defined to organize workflow in phases.

Oracle and **Artemis** are the first who introduced project management software for the first time as products in 1977 and later in 1979 by **Scitor**. The improvements and features then made over the time and other tools began to available in market.

PRINCE2 with increases number of processes in 1996, Agile methodology in 2001. People also started to adopt the total cost management framework in order to reduce the cost of project management.

The cloud-based software began to popular from 2008 due to its flexibility. Project management was beginning to asked as skill in jobs. Then 2010 onwards many popular cloud-based software came in picture as the virtually created team can access the information remotely from anywhere. Hence, the first mobile app for project management was developed in 2012.

Now days in market, such tools are available on different types of accessibility that includes Desktop, Web-based, Mobile, Personal, Single User, Collaborative, Visual etc. The popular tools are MS Project, Scoro, FreedCamp, Podio, Zoho, Basecamp, Jira, Trello, Asana, Wrike, ProofHub, which comes with different monthly charges according to the plans and their features. Some of this software provides even free access with basic features or limited days trial, too.

1.4 Scope of work

Project management is "*The discipline of planning, managing, inspiring and controlling resources in order to meet specific targets.*" The major quest of the project management is to achieve project objectives and goals along with keeping the project scope, time, quality and cost in mind. This dealing seems complex without the help of a good project management tool/software. The need of such tool is even higher in today's COVID-19 situations where one has to stay at home and work in co-ordination with the peers. The tool is providing web-app interface to communicate and manage the responsibilities of tasks assigned. Also, the reporting, approvals and feedbacks becomes immediate and remote. Less rush, less paper work and more productivity.

1.5 Proposed Feature set

Following are the proposed feature set:
Users involved:

- team members
- team leader

Team leader utilities:

- create project
- create tasks with
    - name
    - priority
    - deadlines
    - dependencies
    - description
- create a team
- assign tasks to the team members conveniently
- modify project process
- monitor project in quick overview including
- current progress
- past activities

- Kanban board
- add/remove tasks as per change in requirements/deadlines
- approve submitted task completions

Team members utilities:

- View assign tasks
- create sub-tasks and personal checklists
- manage personal project board and account
- create submissions
- create/submit reports
- get notifications/reminders and messages/email

General Users utilities:

- login/log out
- account settings
- Login mode
- Reset Password
- Mail Communication within App

Chapter: 2 System Design

2.1 Class Diagram

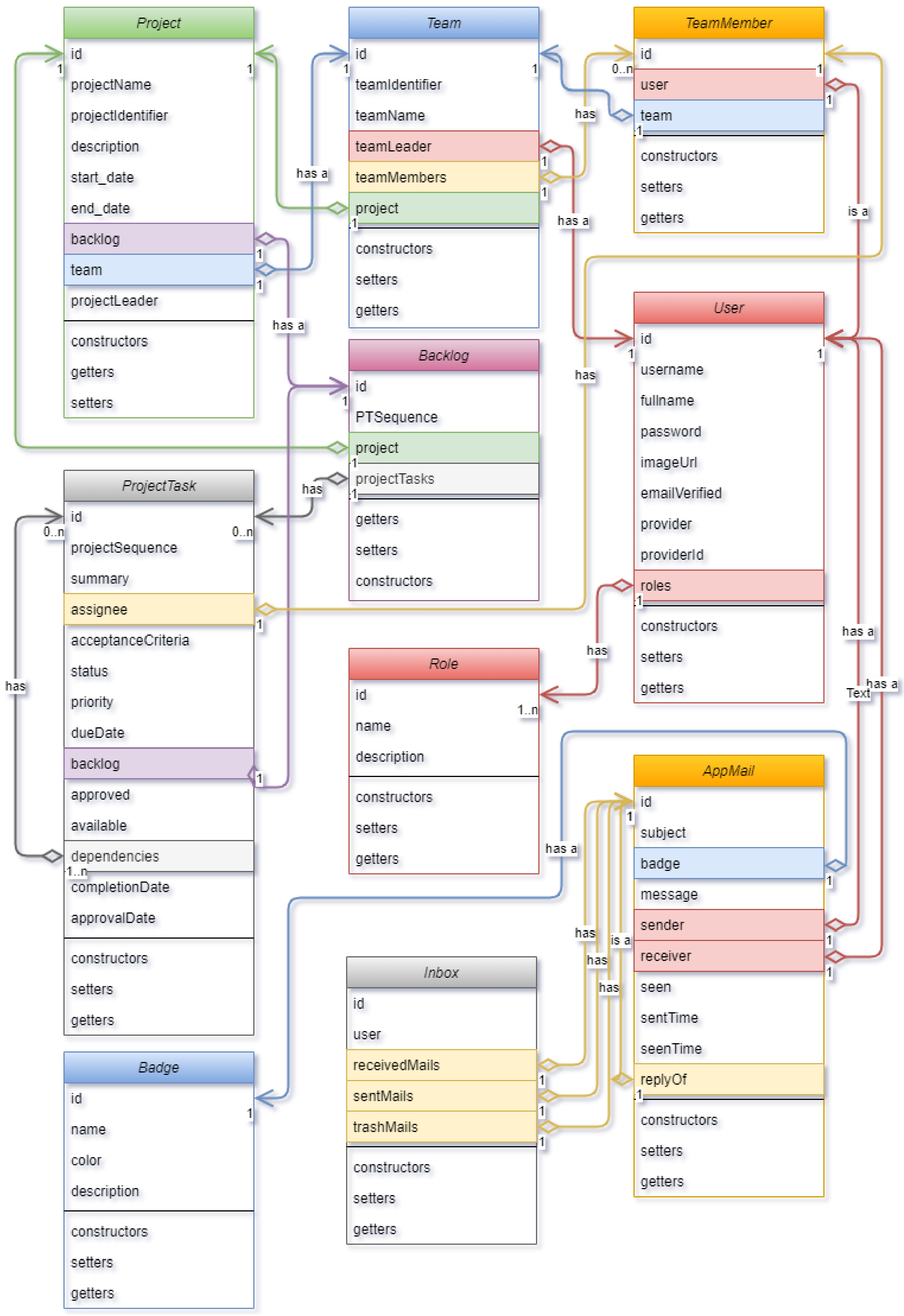


Figure 1 Class Diagram

2.2 Entity-Relationship Diagram

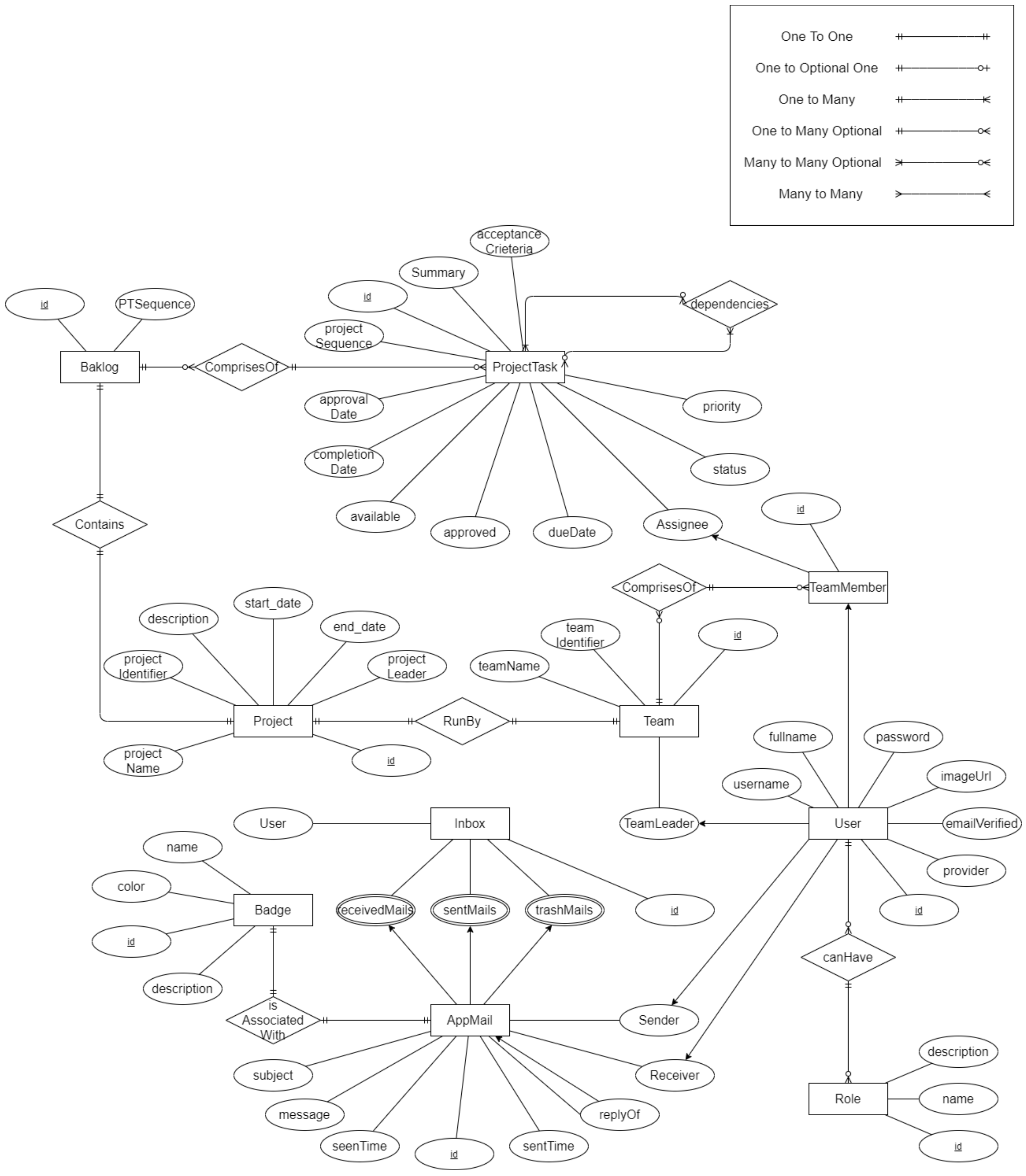


Figure 2 E-R Diagram

2.3 System Flow Chart

Following is the System Flow Chart for the **Authentication** in application. User first come to the index/landing page of the application. The application uses token-based authentication. Hence it looks for token in local storage, if it's found then it validates and logs in the user automatically, else use has to go with one of the signup/login processes. On successful login token is saved in local storage and user is prompted to choose his role for the next usage of the application and accordingly he is redirected to the team leader dashboard or team member dashboard.

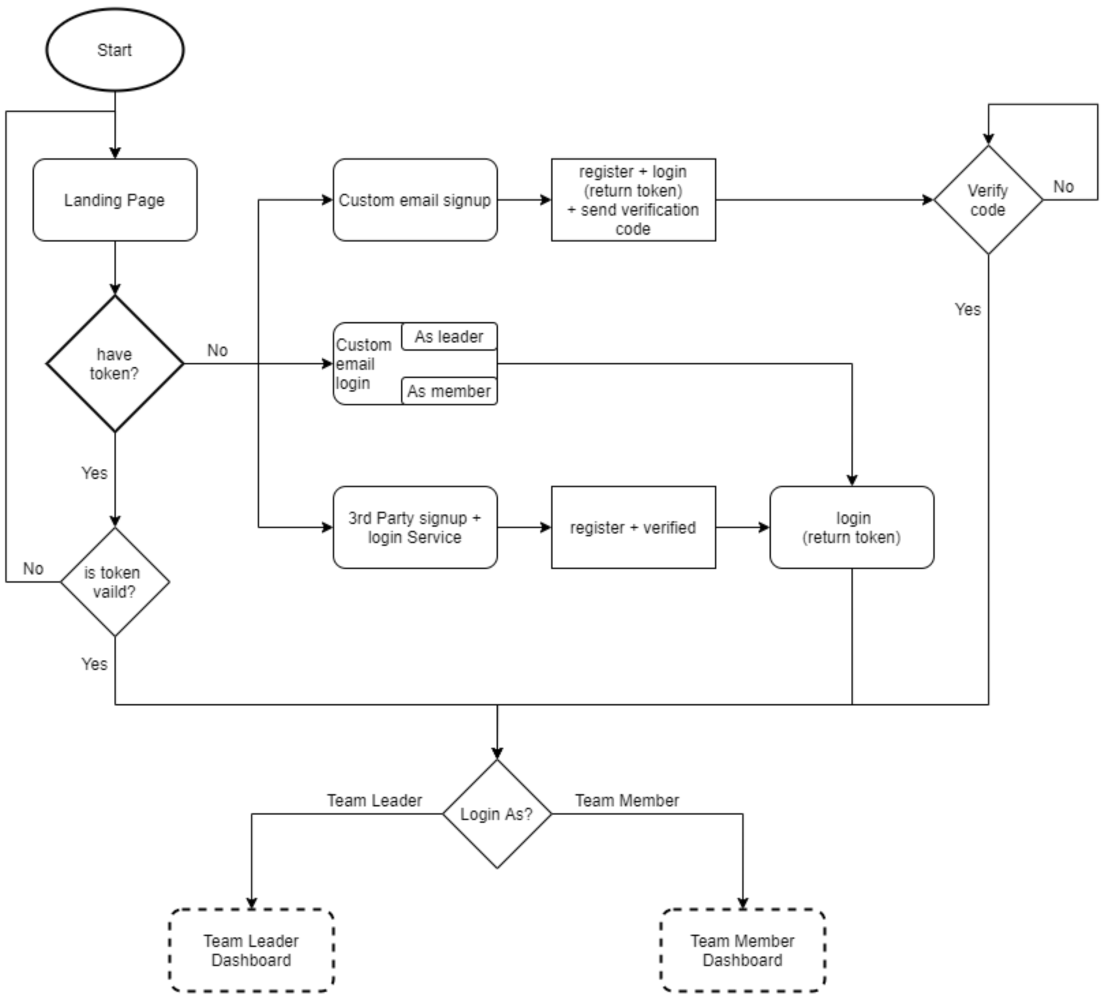


Figure 3 Authentication Flow Chart

Following is the flow chart for **team leader dashboard**, which can be accessible after successful authentication. The flow chart mainly describes the utilities of the team leader. However, he can switch his role to team member and go to the team member dashboard. Also, he can access and go to his MailBox, which is for communication purpose within application.

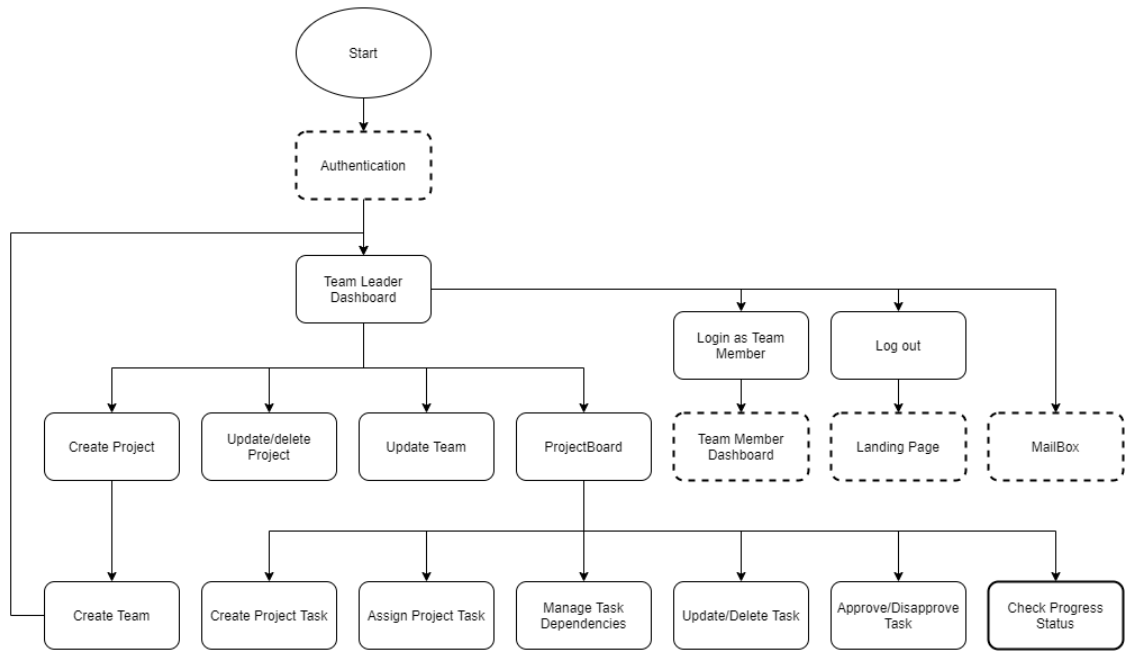


Figure 4 Team Leader Dashboard Flow Chart

Following is the Flow chart for **team member dashboard**, which also accessible after successful authentication. Similar to the team leader dashboard, team member can also switch the role to go to team leader dashboard and check for his own projects. As a user, member too can access Mailbox.

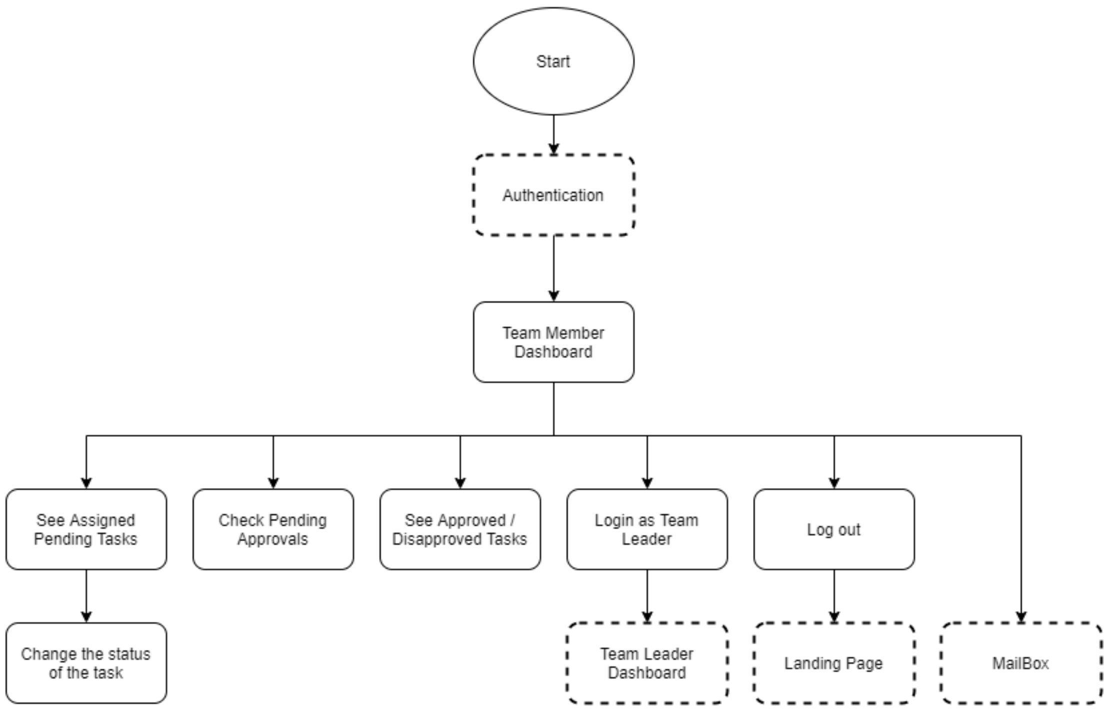


Figure 5 Team Member Dashboard Flow Chart

Following is the Flow chart for **Mailbox**, which is general user functionality. Flow chart describes the features of Mailbox.

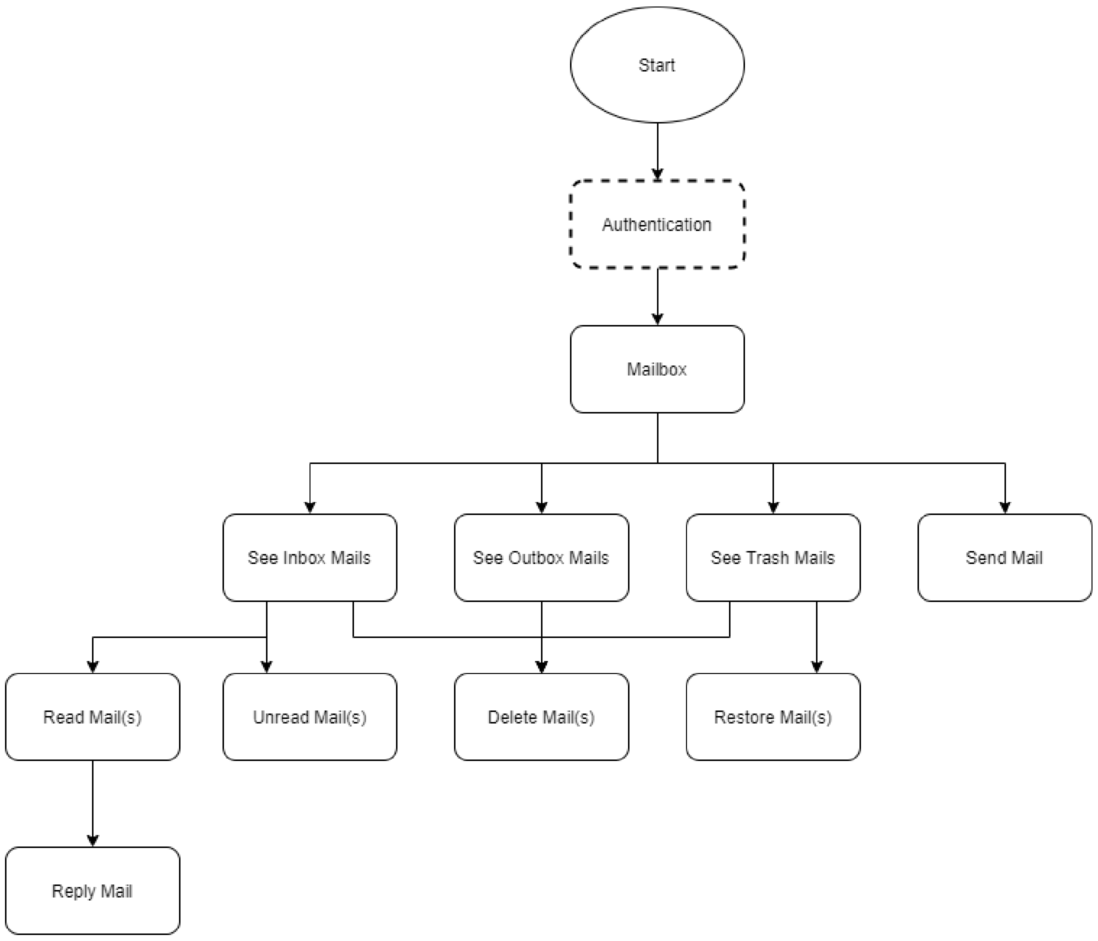


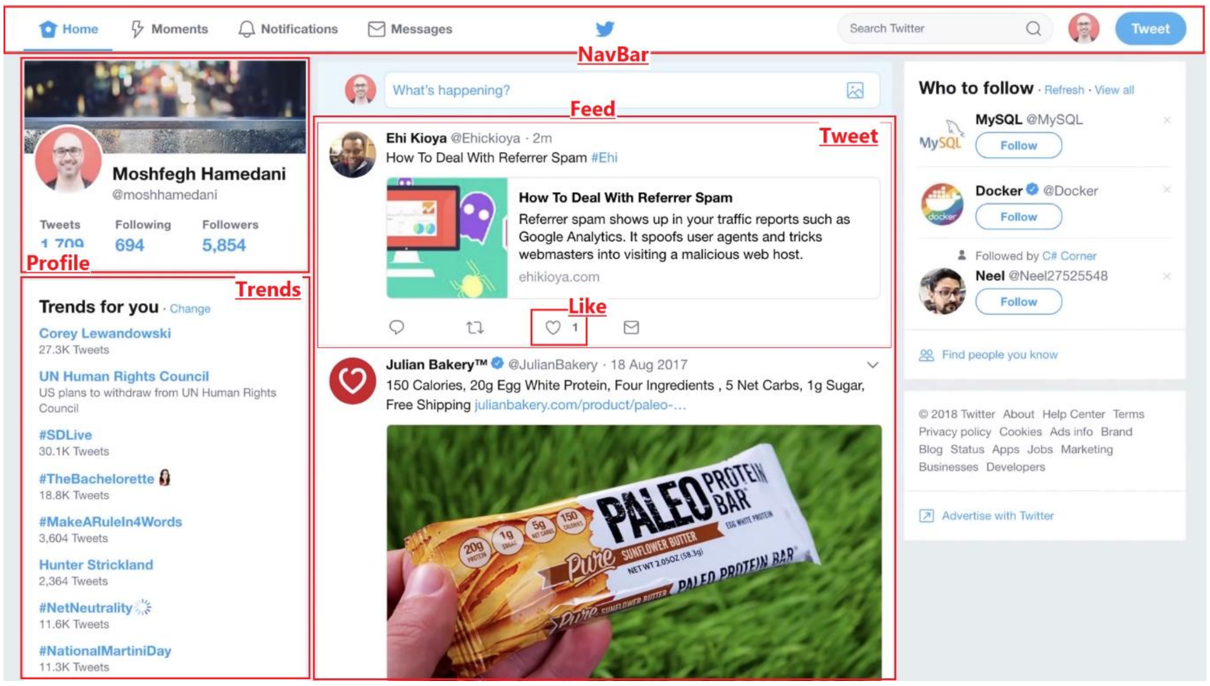
Figure 6 Mailbox Flow Chart

Chapter: 3 Toolsets

3.1 ReactJS

ReactJS is one of the dominating JavaScript Library for developing quick and interactive UI. At the heart of the react application is component (a piece of the UI). We build bunch of reusables, independent, isolated components and aggregate them to develop complex UI. Each React application has at least one root component containing other such child components.

Following tweeter page is the example which is divided in the component:



Moshfegh Hamedani
@moshhamedani
Tweets 1,709 Following 694 Followers 5,854

Trends for you · Change
Corey Lewandowski
27.3K Tweets
UN Human Rights Council
US plans to withdraw from UN Human Rights Council
#SDLive
30.1K Tweets
#TheBachelorette
18.8K Tweets
#MakeARuleIn4Words
3,604 Tweets
Hunter Strickland
2,364 Tweets
#NetNeutrality
11.6K Tweets
#NationalMartiniDay
11.3K Tweets

Ehi Kioya @Ehickioya · 2m
How To Deal With Referrer Spam #Ehi
How To Deal With Referrer Spam
Referrer spam shows up in your traffic reports such as Google Analytics. It spoofs user agents and tricks webmasters into visiting a malicious web host.
ehikioya.com

Julian Bakery™ @JulianBakery · 18 Aug 2017
150 Calories, 20g Egg White Protein, Four Ingredients , 5 Net Carbs, 1g Sugar, Free Shipping julianbakery.com/product/paleo-…

MySQL @MySQL
Docker @Docker
Followed by C# Corner
Neel @Neel27525548

© 2018 Twitter About Help Center Terms Privacy policy Cookies Ads info Brand Blog Status Apps Jobs Marketing Businesses Developers

Figure 7 Tweeter Page Divided in Components

Hence the corresponding React app can be tree of components basically as following:

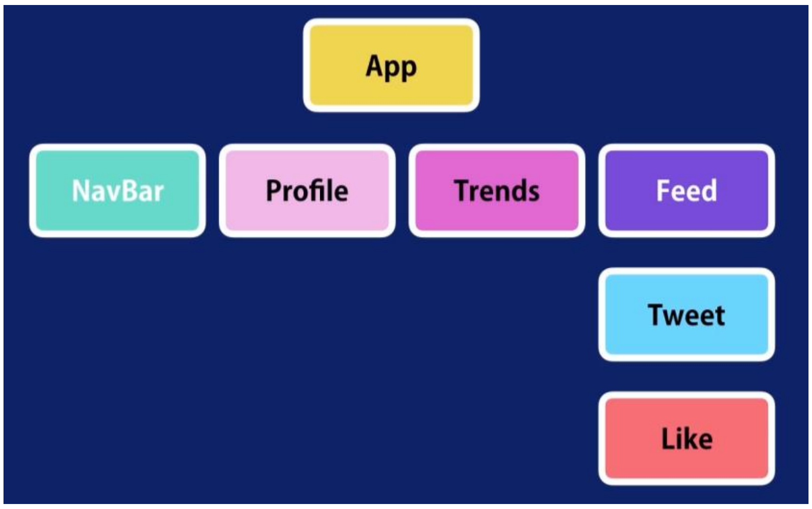
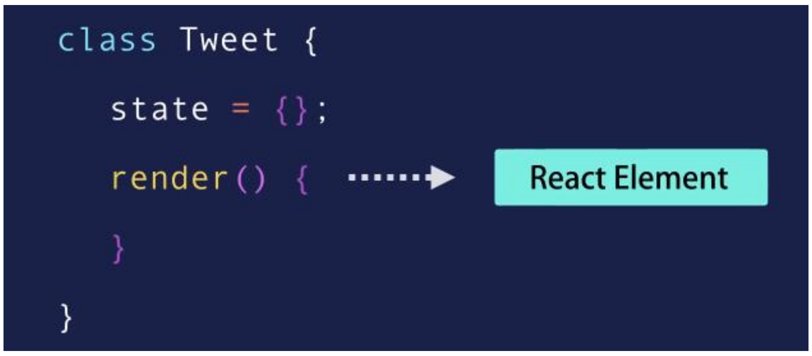


Figure 8 React Component Tree

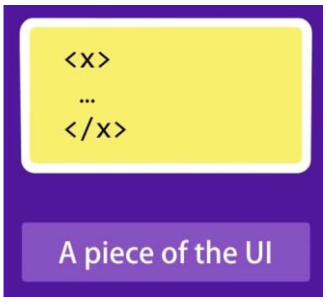
In terms of implementation, basically such components have its own local state that stores the necessary variables for that particular component and a render method that returns a JSX element (HTML components which can be controlled by JavaScript we can say) to be rendered on the screen that describes how the UI should look like.



```
class Tweet {
   state = {};
   render() {
   }
}
```

Figure 9 Implementation of a React Component

Such Component can basically be used similar to the HTML tags to create piece of UI:



```
<x>
 …
</x>
```

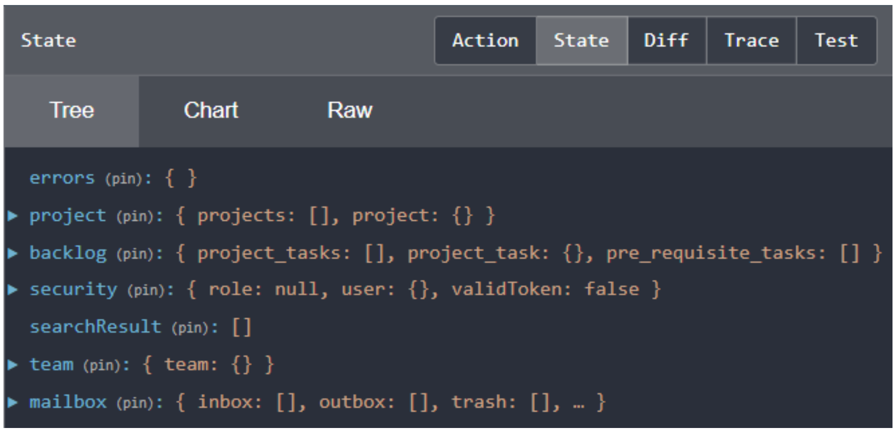
Figure 10 Representation of Component Usage

React keeps Virtual DOM in memory which basically is a light-weight representation of the DOM element. The JSX elements such as above one is then compared with the existing DOM representation and if changes are found in states, we get new updated react element and replaces that particular child component in the tree to sync with Virtual DOM. Hence eventListners are not needed to update the UI. Instead, we create functions that changes these states based on actions and react will automatically react to the state changes as React's primary task is to take care of rendering a view and makes sure that view is in sync.

3.2 ReduxJS

ReduxJS is state management library for JavaScript application. It can be used with React, Angular, Vue and also VanillaJS. It works well with react comparatively and adopted widely by the community. In frontend, we want to keep updating different UI part in Synchronization. In complex UI, data flows from one part to another and in some cases, unpredictable changes may occur and cause an unhandled error.

Hence to figure out how the data changed and even to keep track of changes, the answer is state management and the solution is state management library such as ReduxJS.



```
  errors (pin): { }
▸ project (pin): { projects: [], project: {} }
▸ backlog (pin): { project_tasks: [], project_task: {}, pre_requisite_tasks: [] }
▸ security (pin): { role: null, user: {}, validToken: false }
  searchResult (pin): []
▸ team (pin): { team: {} }
▸ mailbox (pin): { inbox: [], outbox: [], trash: [], … }
```

Figure 11 Redux State Object in Tree View

With Redux, we store all the application state inside a central repository which basically is only one JS Object, and this way it accomplishes the concept of single source of truth. We can consider it as a database of frontend we can say. All UI parts gets data from this store. Hence updating is required only in one single place. This will solve the problem of synchronization between different parts of UI. This way Redux is centralizing the state of the app in order to make the flow of the data transparent and predictable. Following is the state object inspecting using Redux Dev Tool Extension, where each slice of the state object is supposed to updated by the corresponding function know as reducer:

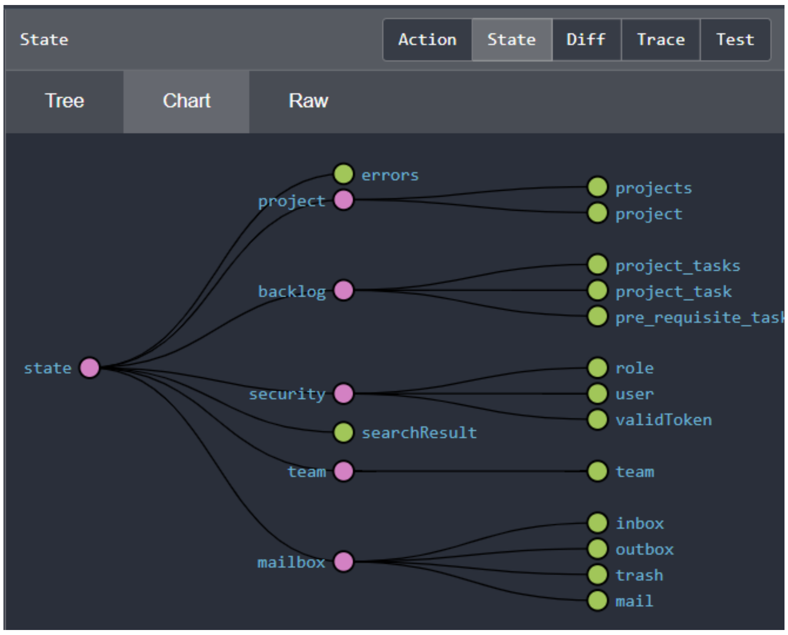


Figure 12 Redux State Object in Chart View

3.3 Spring Boot Framework

The Spring Framework is basically the inversion of control container for the Java platform and the framework to build an application. The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the Java EE platform. (1) Spring is so powerful and high level that it is also known as "Framework of Frameworks".

With Spring Boot, it is easy to build stand-alone, production-level Spring-based Apps which can be "just run". It provides an opinionated view of the Spring-platform and third-party libraries so developers have to spend less efforts to get started. Almost every Spring Boot application does not require much configuration. (2)

Features:

- Spring Boot provides in order to utilize the power of Spring Framework:
- Create stand-alone Spring applications
- Embed Undertow, Tomcat or Jetty straight forward
- We can have opinionated 'starter' dependencies in order to minimize the initial build configuration
- Auto-configures the Spring and other 3rdparty libraries whenever require
- Gives production-ready utilities like health checks, metrics, and externalized configuration
- Literally zero code generation and eliminates the requirements of the XML configuration (2)

The Spring Boot is used to develop the REST API for the Application.

3.4 Spring Security and JWT

Spring Security is a high level, powerful and almost fully customizable authentication and access-control framework. It's the de-facto standard for securing any Springbased apps. Spring-Security is the framework which aims to provide the authentication and the authorization in Java-based applications. As every Spring-based project, the true usefulness of Spring Security can be seen in how easily it can be customized to achieve custom requirements. (3) Features are as following:

- For both Authentication and Authorization, it provides comprehensive and extensible support
- Prevents attacks such as session fixation, CSRF, click-jacking etc
- Servlet Application Programming Interface integration
- Optional integration with Spring Web MVC (2)

Spring Security is used to secure the REST API of the app and JWT is used to manage user's login and subsequent requests to the API.

3.5 Database

In this project, Relational Database is taken in use. During the development of the API for the app, the H2 database is used. H2 database is embedded Relational Database which can be taken in use on the go. We can easily test the API server with small amount of data in database and even we can configure it to flush the data on every restart of the server. In deployment and actual use, the used database is Open-Source Relational Database MySQL, However PostgreSQL can also be used. Spring Boot can be configured to use

MySQL or PostgreSQL as database located on some remote server by including Hibernate and JPA in dependencies.

3.6 Postman

Postman is the coolest tool if one need to analyse the RESTful APIs created by other developers or to test the one, we have developed. It provides an easy, featureful UI that can be used to send requests, without the efforts for creating snippets of code just for testing any API's functionalities. (4) It completely avoids the need of building HTML form pages just to test the API endpoints. This tool become very handy for the API developer. Moreover, it has functionality to run custom code snippets before and after the request sent, which helps further to perform custom testcases. We can create different collections for different projects and different folders for different features inside the collection. Due to the facility of environment, we can set some global variables and run such collection or a folder in order to automatically test the bunch of sequentially dependent/independent requests. Moreover, using code snippets, we can also customize the flow of the request-sending apart from linear manner. It saves lots of time in testing when the API becomes more complex as just the one click can re-test the existing modules in order stay assured that previous modules are not affected after integrating new functionalities in the API.

3.7 NodeJS

Node.js is an environment developed on Chrome's JS run-time in order to create quick and scalable network apps with ease. It utilizes the event-driven, non-blocking I/O model which is making it accurate and light-weight, that of course makes it suitable for intense data driven real-time apps which is executed across distributed-devices. (5) Node Module Package Manager allows to include some other useful libraries in the current project. These some other libraries include axios, redux-logger, redux-thunk, react-router-dom and many more. Features:

- Asynchronous and Event-Driven
- Very Fast
- Single Threaded but Highly Scalable
- No Buffering
- Released under MIT license

3.8 Visual Studio Code

VSCode is one of the best opensource code editor that is built by Microsoft for Windows, Linux and macOS. It provides aid in syntax-highlighting, debugging, smart auto code completion, source-code refactoring based on language, snippets and built in Git Support. (6) It comes with built-in support for JavaScript, TypeScript and Node. Moreover, there are plenty of extensions are available in VSCode's Marketplace which helps developer to carry out their projects using different frameworks, libraries or languages. Due to that, in addition to the react project for frontend, Spring boot REST API is built using VSCode extensions and eliminated the needed to install heavy editors for java projects such as eclipse or IntelliJ.

Chapter: 4 Technical Brief of Implementation

4.1 Frontend Diagram

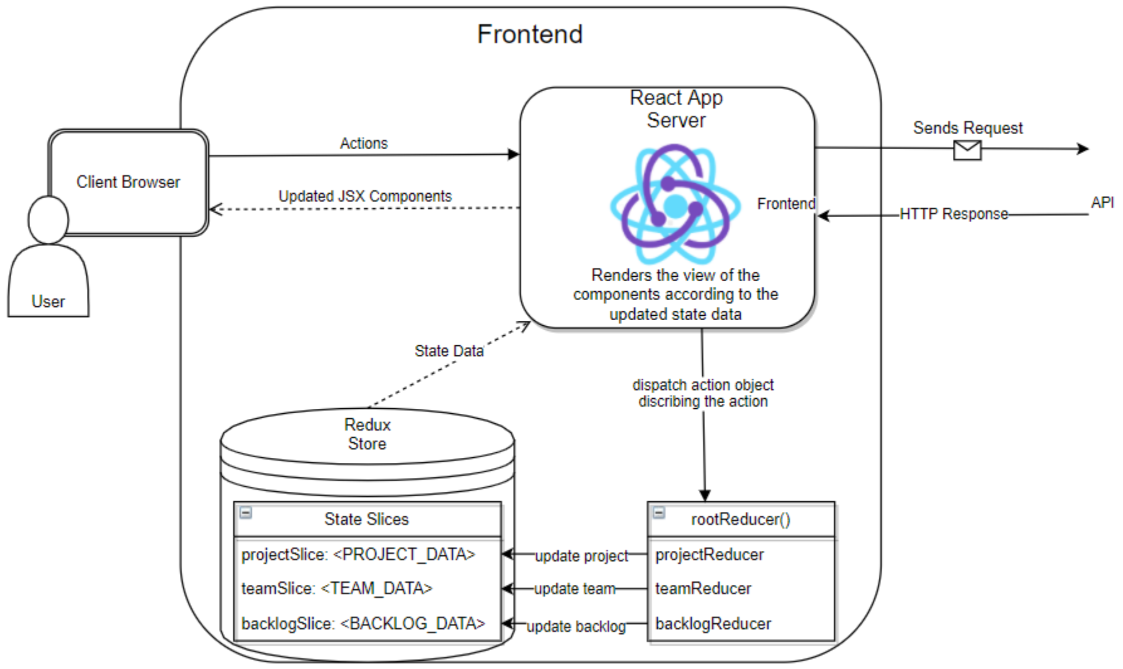


Figure 13 Frontend Diagram

Frontend uses ReactJS and ReduxJS. The above image depicts the idea of the flow of the data and operation briefly using these two libraries.

The application is available to client via frontend server, which is actually the server run by react. React server presents the application in forms of components to the client. These components are JSX elements basically. According to the actions of the client, the view of the app is constantly updated by replacing new JSX elements by React Server.

Now to decide what to change in a view (components), the clients' actions are priorly defined and managed using reduxJS. ReduxJS provides a store to hold the state of the entire application. The state of the application is basically a JavaScript object, which most of the time is divided in state-slices. In the diagram, there are three state slices are mentioned: projectSlice, teamSlice and backlogSlice. These slices are responsible to hold the data of project, team and backlog respectively which is fetched according to the need from backend by using axios (JS library).

The view of the application utilizes the data from the state. Whenever a user performs any action, an action object is created, which holds the information regarding the operation. Such operations are already priorly classified when the redux store is designed. The action objects are then pass down to the root-reducer, which comprises of further reducers particular to the state-slices. In diagram, three reducers: projectReducer, teamReducer and backlogReducer is shown. These reducer functions are responsible to update the corresponding state slice based on the action object provide in their arguments. Whenever the state slices get changed, react is notified to check and render the view with new JSX element. One data-part of the state may be consumed by different components on the screen, hence whenever such data is changes, all these components get the effect. This way different part of the component-based web page stays in sync by using redux with react.

4.2 Backend Diagram

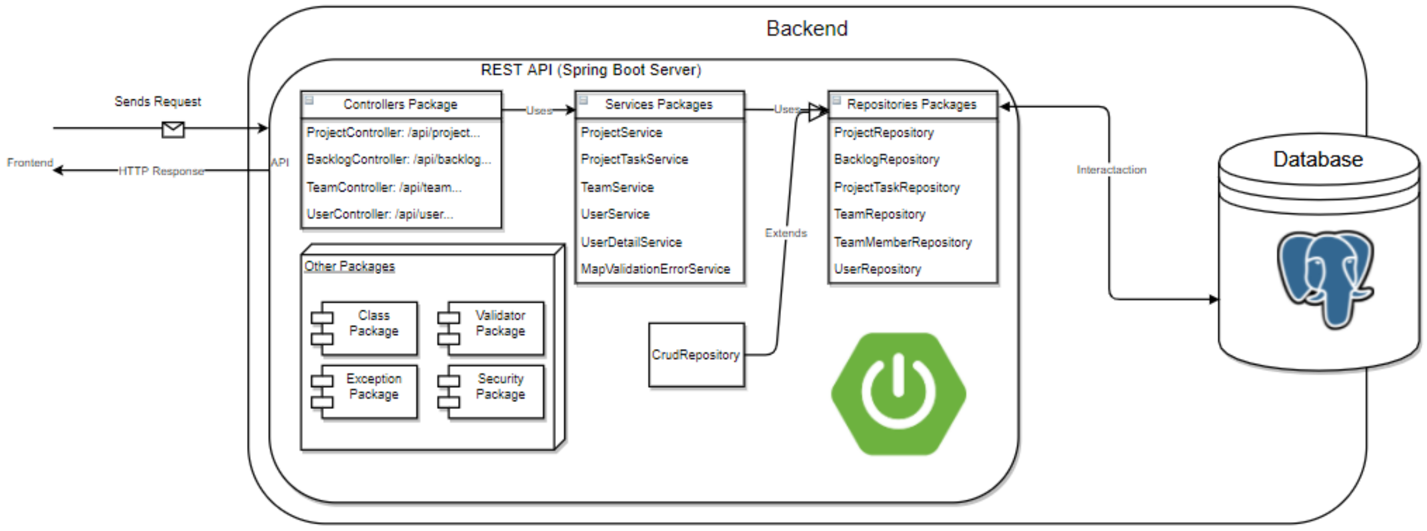


Figure 14 Backend Diagram

The above diagram represents the heart of the REST API very briefly. The API is following the architecture follows the MVC (Model View Controller) pattern developed in Java using Spring Boot. To handle the different requests made to the API, there are different controllers defined for different endpoints of the API. In diagram few of them are shown and senses the following basic endpoints:

1) **ProjectController:** senses all requests on '/api/project' endpoint
2) **BacklogController:** senses all requests on '/api/backlog' endpoint
3) **TeamController:** senses all requests on '/api/team' endpoint
4) **UserController:** senses all requests on '/api/user' endpoint

A particular controller senses further request on further sub endpoint using different functions. Such functions utilize the functions provided by different Services like projectService is providing the functions to save, update, retrieve or delete the project.

Such services use the functions provided by the corresponding repository interfaces and are implemented by Spring Boot JPA. Such repository functions are providing a way to interact with the database without even writing a query.

Each particular repository utilizes the class annotated as entity designed by the developer and creates the table with corresponding attributes in the database on the timing of the starting the server if not created previously. Spring Boot is able to provide an implementation for the custom entity repositories since while defining them, we extend them from CrudRepository as shown in the diagram.

In addition to this all, there are other packages at aid for different logic handling. That includes Class Package, Exception Package, Security Package, Validator Package. The security package further holds the packages for Custom authentication and OAuth2 Authentication services.

4.3 Testing of the API

While developing such application where frontend is completely separated from backend and both are running on different servers, and frontend relies on the backend API for data through requesting, the backend is required ready prior to do coding for frontend. In such case for testing of the API, we ideally need to create basic forms that send data to the API endpoints. To avoid such efforts, there are tools to send the request directly to the API. One of best is Postman. In this application, the testing of the API is done using Postman. Postman is way comforting for testing. We can create folder of requests of a particular module where one request may rely on the response of the previous request, define the order of the requests and then on one click we can test the newly designed or previously designed feature without putting much effort every time.

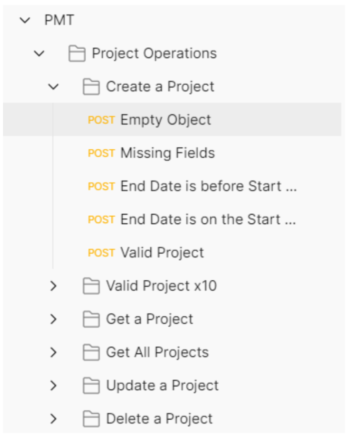
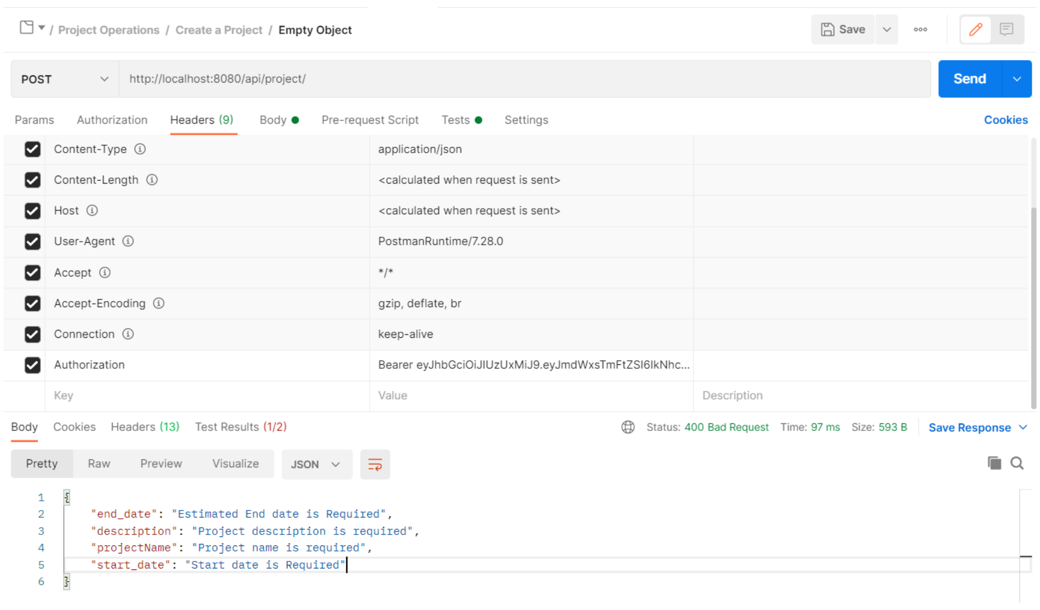


Figure 15 Folder structure of Project Operations in Postman

In the image, the folder 'Project Operations' contains 6 more folders of request to test the entire ProjectController. When we run the folder 'Project Operations' all the endpoints of the controller will be tested by the requests kept inside these six folders. Following is one example request to test the create project form. We can see the errors we got in response. Also, it says that 1 out of 2 tests are passed. This way after each feature developed in the API, all the modules are easily re-tested.



http://localhost:8080/api/project/

	Key	Value	Description
✓	Content-Type	application/json	
✓	Content-Length	<calculated when request is sent>	
✓	Host	<calculated when request is sent>	
✓	User-Agent	PostmanRuntime/7.28.0	
✓	Accept	*/*	
✓	Accept-Encoding	gzip, deflate, br	
✓	Connection	keep-alive	
✓	Authorization	Bearer eyJhbGciOiJIUzUxMiJ9.eyJmdWxsTmFtZSI6IkNhc...	

Status: 400 Bad Request Time: 97 ms Size: 593 B

```
{
    "end_date": "Estimated End date is Required",
    "description": "Project description is required",
    "projectName": "Project name is required",
    "start_date": "Start date is Required"
}
```

Figure 16 Postan Request example

Chapter: 5 User Manuals

5.1 Login Functionalities

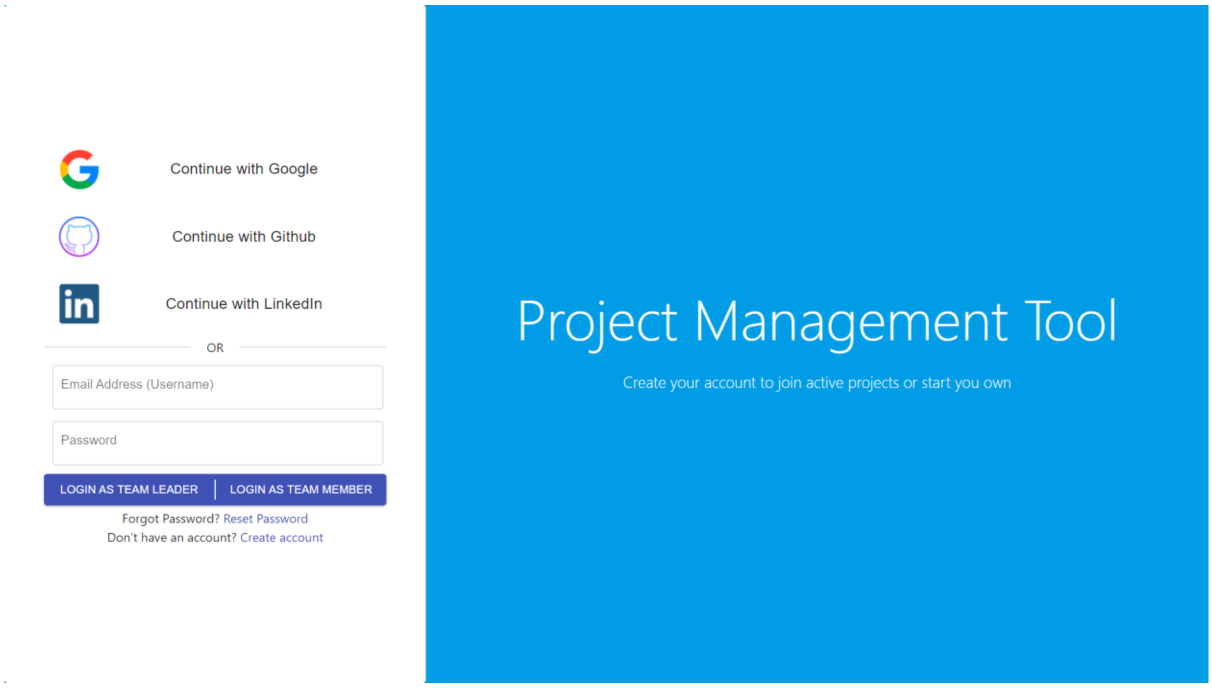

Continue with Google
Continue with Github
Continue with LinkedIn
OR
Email Address (Username)
Password
LOGIN AS TEAM LEADER | LOGIN AS TEAM MEMBER
Forgot Password? Reset Password
Don't have an account? Create account
Project Management Tool
Create your account to join active projects or start you own

Figure 17 Landing Page

When the new user came to the site, first thing he sees is the following landing page. On the left of the page is login/signup functionality and in the right the tool name and a short line describing the app.

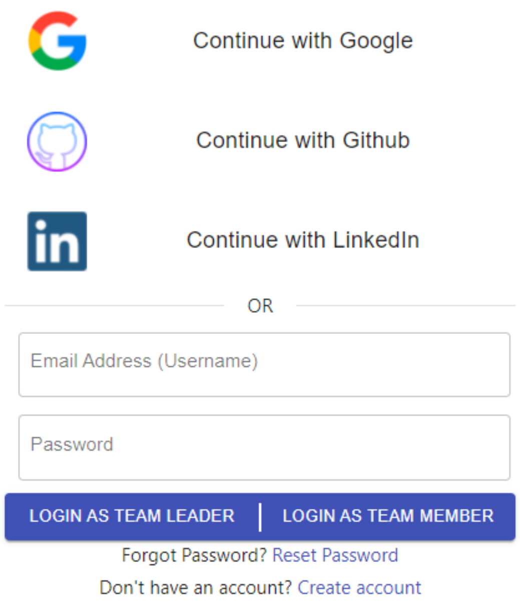

Continue with Google
Continue with Github
Continue with LinkedIn
OR
Email Address (Username)
Password
LOGIN AS TEAM LEADER | LOGIN AS TEAM MEMBER
Forgot Password? Reset Password
Don't have an account? Create account

Figure 18 Login Form

In the left, there are three different 3rd party login services:

1) Google
2) Github
3) LinkedIn

The user of such application is most likely to have accounts on such platform.

Moreover, A login form requiring custom email address and password is also there. User who doesn't want to associate any of the login service with this app can go with custom email and password.

But first he has to register and for that a link "Create Account" is provided bellow the login button, upon clicking it, two more fields: full name and confirm password will added to the form and form will become signup form. This way, we can toggle signup and login forms. Also, on login form there is a link to reset the password if the user forgets the password.

The following is the signup form:

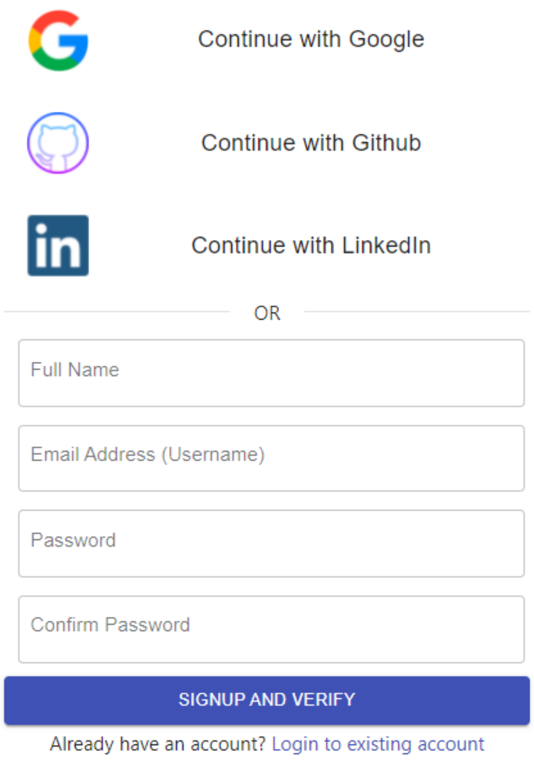


Figure 19 Signup form

Upon successful login he has to choose the role of team leader or team member for the next activity in order to bring up the appropriate dashboard.

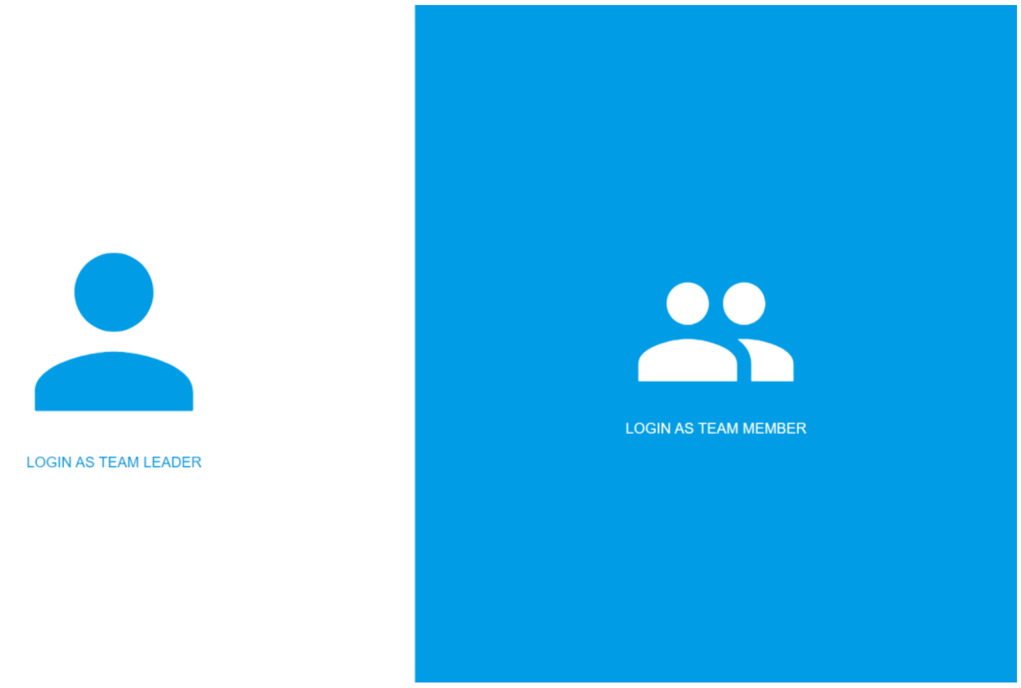


Figure 20 Choose role for login

5.2 Header Functionalities

On the top of the window, there is a navigation bar is there to aid.

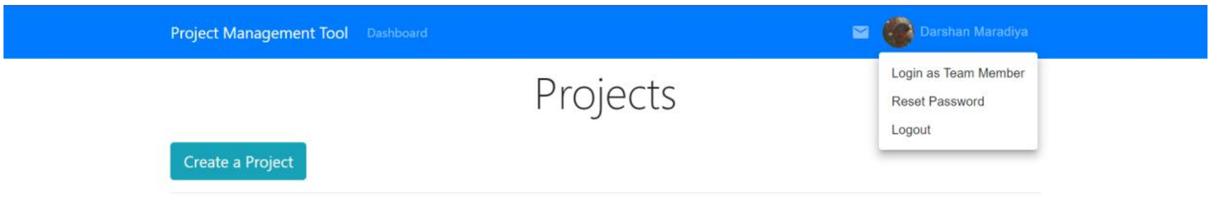


Figure 21 Navbar

On the left App Name and Dashboard links to navigate to the dashboard.

On the right, mail icon is to navigate to the application-mailbox of the user.

Then next, there is a button with a profile photo and the full name of the user, upon clicking it the menu pops-up with three options:

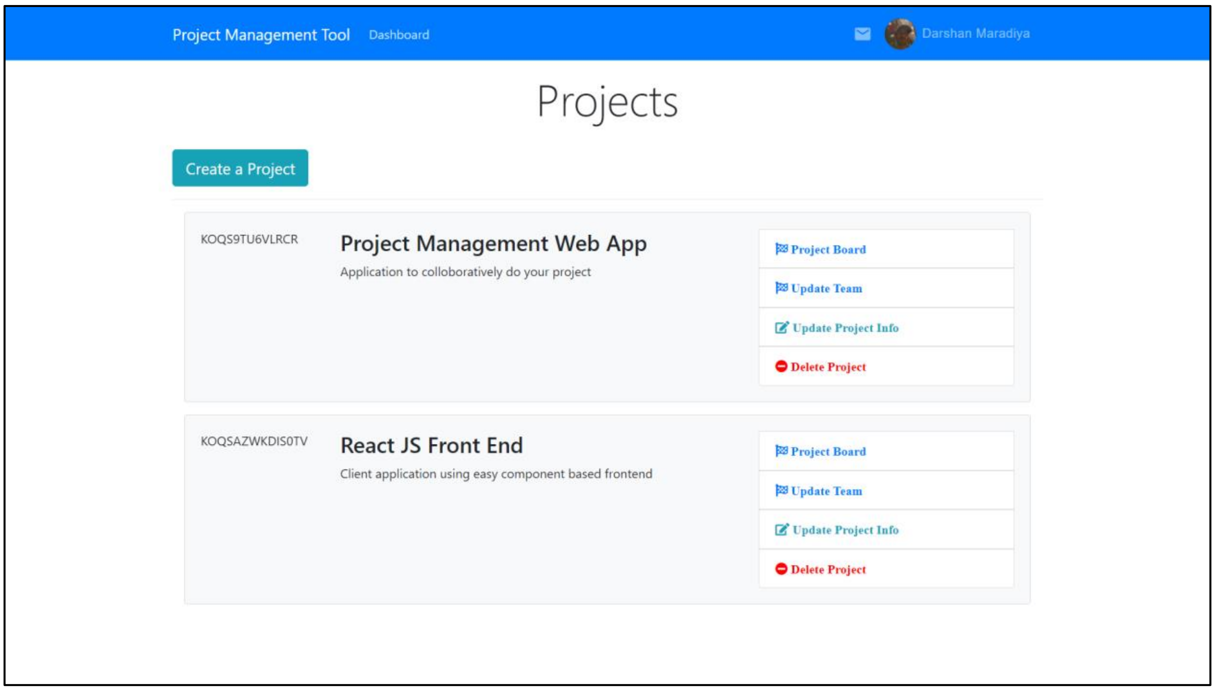
1) Login as Team Member ("Login as Team Leader" if currently logged in as "Team Member")
2) Reset Password
3) Logout

Here the profile photo shown is same as the one in the user's social account.

5.3 Dashboard Functionalities

As application has two types of user-roles, there are two types of dashboard, too.

5.3.1 Team Leader Dashboard



KOQS9TU6VLRCR
Project Management Web App
Application to colloboratively do your project

KOQSAZWKDIS0TV
React JS Front End
Client application using easy component based frontend

Figure 22 Team Leader Dashboard

On the leader's dashboard, all the projects are shown as list, which are going on or done under his leadership. Each project is showing its project name, project-identifier and project description.

Moreover, there are four operation for each project:
1) Project Board: To navigate to the project board
2) Update Team: To navigate to the update team form
3) Update project Info: To navigate to edit the project info
4) Delete Project: To delete the project and all the related data

Following is the view of project board:

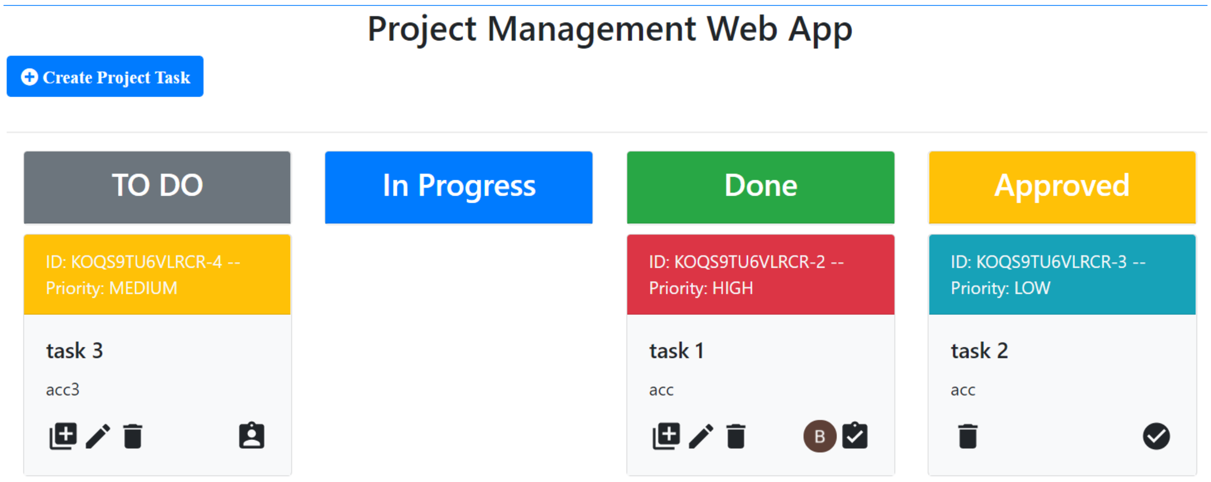

Project Management Web App
Create Project Task
TO DO
In Progress
Done
Approved
ID: KOQS9TU6VLRCR-4 -- Priority: MEDIUM
task 3
acc3
ID: KOQS9TU6VLRCR-2 -- Priority: HIGH
task 1
acc
B
ID: KOQS9TU6VLRCR-3 -- Priority: LOW
task 2
acc

Figure 23 Project Board

On the top-left, there is a button to create new project task, upon clicking it, user will be able to naviagate to the create project task form.

On the main board, there are four trays for containing tasks with different progress status/phase:
- To Do
- In Progress
- Done
- Approved

The task can have priority, too:
- Low: Marked with blue color
- Medium: Marked with yellow color
- High: Marked with red color

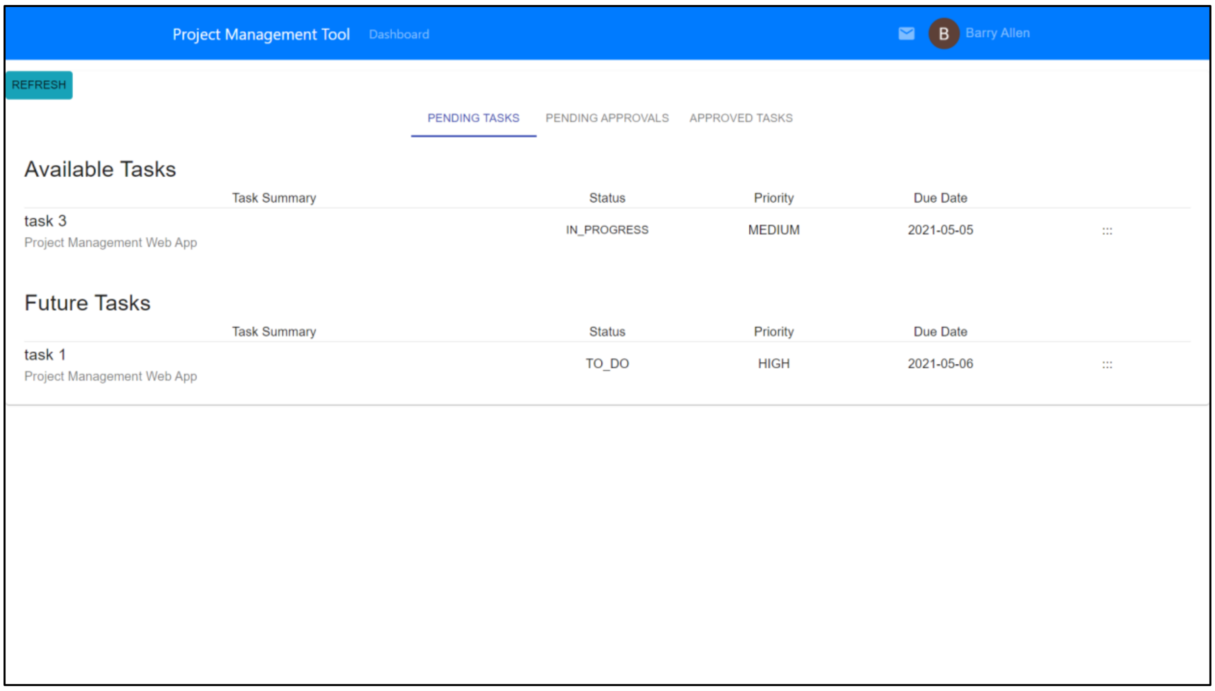
In each task card, there is an operation tray containing following operations respectively:
1) Manage Dependencies: To add pre-requisite tasks
2) Update Task: To edit the information of the task
3) Delete Task: To delete the task and all the related informations
4) Assign Task: To assign task to team member, if assigned then the user's avatar is displayed
5) Approve/Disapprove: To approve or disapprove the submitted tasks

5.3.2 Team Member Dashboard

On the Team Member Dashboard, there are 3 panels:

1) Pending Tasks

- In this panel, there are two sections: Available Tasks and Future Tasks. Available tasks are those tasks, which has no pre-requisite tasks pending and Future tasks are those which is not yet ready because their pre-requisite tasks are still not done.



Available Tasks

Task Summary	Status	Priority	Due Date	
task 3 Project Management Web App	IN_PROGRESS	MEDIUM	2021-05-05	:::

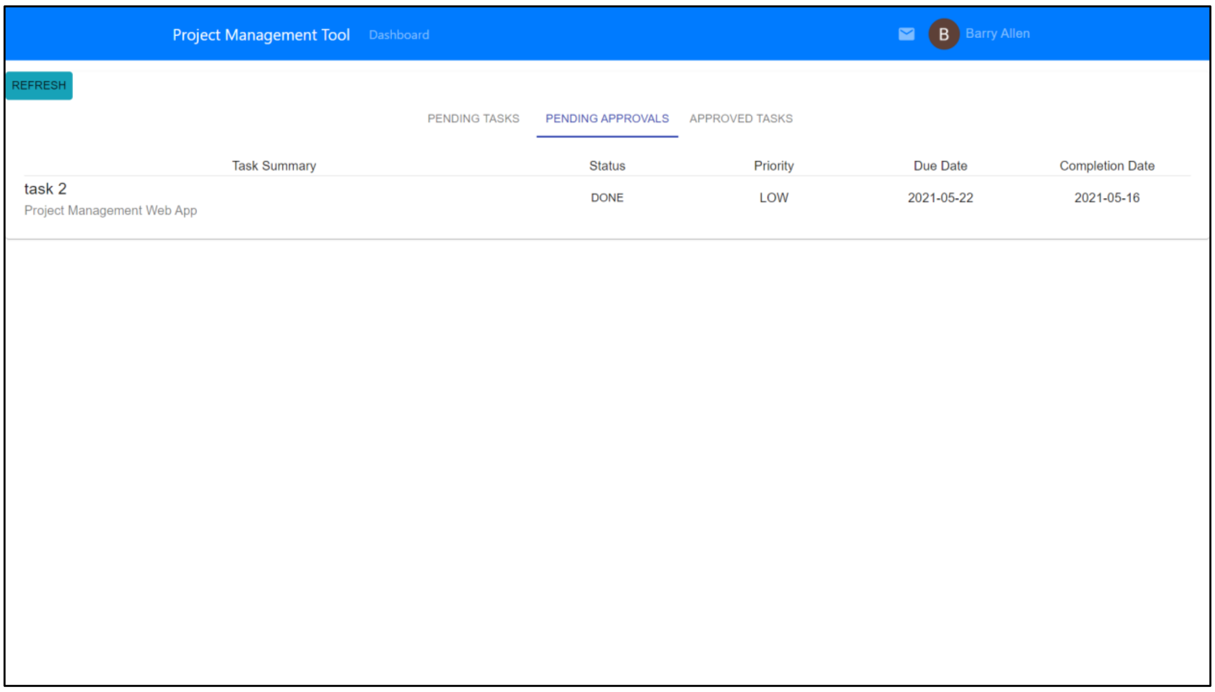
Future Tasks

Task Summary	Status	Priority	Due Date	
task 1 Project Management Web App	TO_DO	HIGH	2021-05-06	:::

Figure 24 Pending Tasks on Team Member Dashboard

2) Pending Approvals

- In this panel, there are tasks which are completed (marked as Done) and waiting for approval from the team leader.

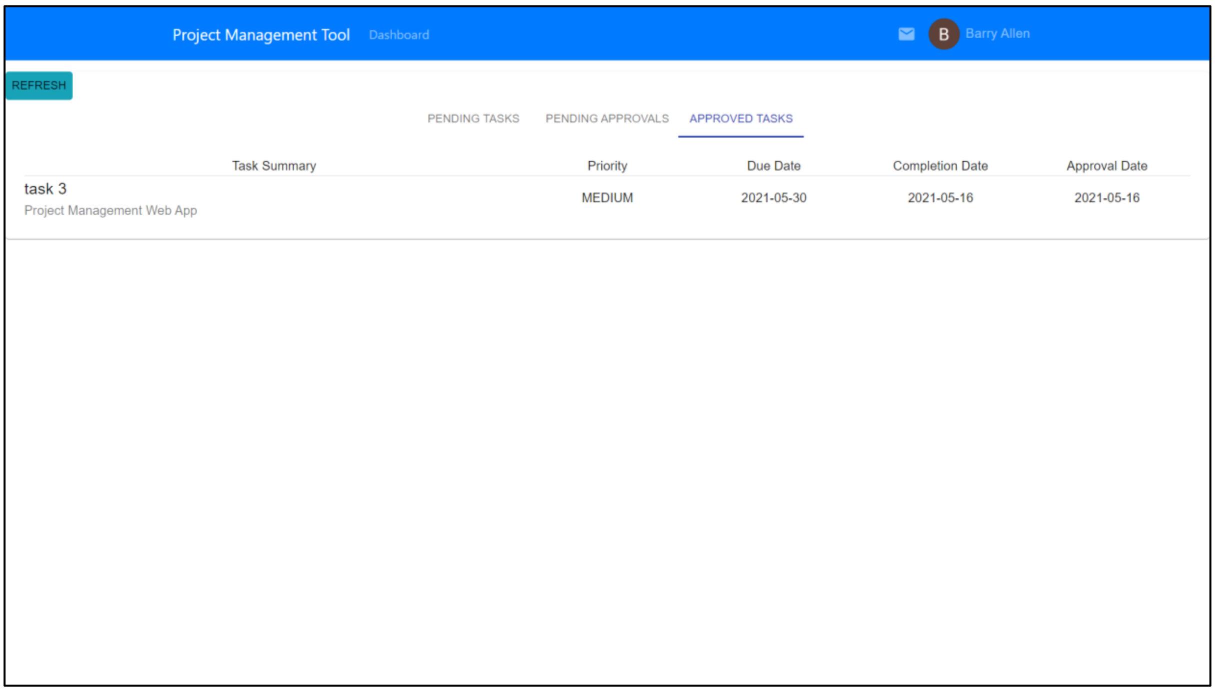


Task Summary	Status	Priority	Due Date	Completion Date
task 2 Project Management Web App	DONE	LOW	2021-05-22	2021-05-16

Figure 25 Pending Approvals on Team Member Dashboard

3) Approved Tasks

- In this section, Team Member can see their assigned and completed tasks which got approvals from the leader.

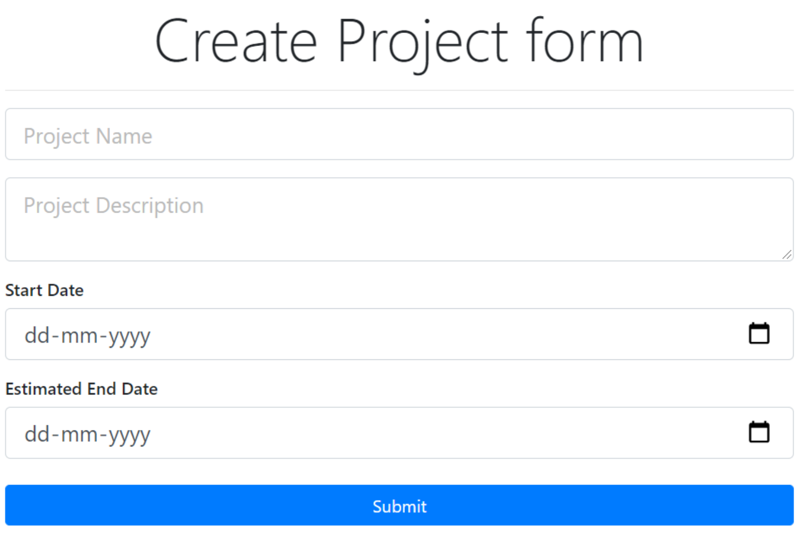


Task Summary	Priority	Due Date	Completion Date	Approval Date
task 3 Project Management Web App	MEDIUM	2021-05-30	2021-05-16	2021-05-16

Figure 26 Approved Tasks on Team Member Dashboard

5.4 Different Forms in the application

5.4.1 Create Project Form


Create Project form

Project Name

Project Description

Start Date

dd-mm-yyyy

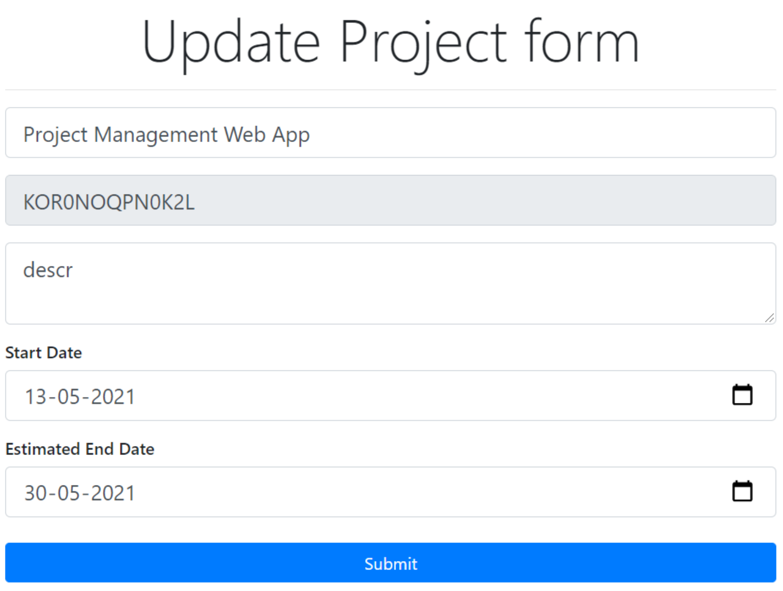
Estimated End Date

dd-mm-yyyy

Figure 27 Create Project Form

While creating a new project, Project Name – a String, small Project Description – a String, Start Date and End date of the project is required. Form validates that you enter the dates properly that is start date is before end date.

5.4.2 Update Project Form


Update Project form

Project Management Web App

KOR0NOQPN0K2L

descr

Start Date

13-05-2021

Estimated End Date

30-05-2021

Figure 28 Update Project Form

While updating the project, form gets the existing field values and then leader can change accordingly. Moreover, we can see the auto generated project identifier field and also disabled.

5.4.3 Create Team Form

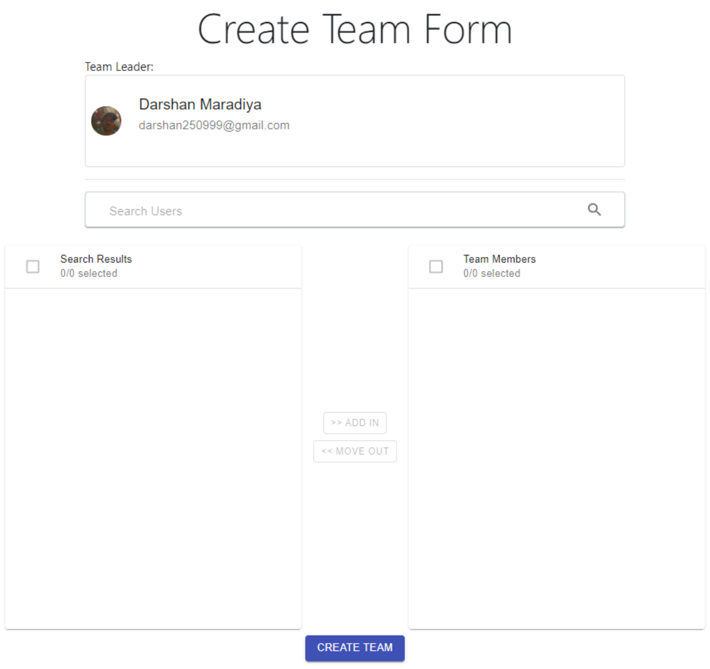


Figure 29 Create Team Form

After clicking submit on ‘Create Project Form’, leader get to see is ‘Create Team Form’. There are two trays and a search bar. Search bar will be helpful to find any particular user from existing registered users. The search results will be shown in left tray. Then we can select users from this tray, more than one at a time using checkbox and Add them in to the Team Members list shown in the right tray. Upon clicking create team button, team will be created. Here Team with no team members is still allowed if someone wants to do project alone as the one who creates the project is the leader anyway.

5.4.4 Update Team Form

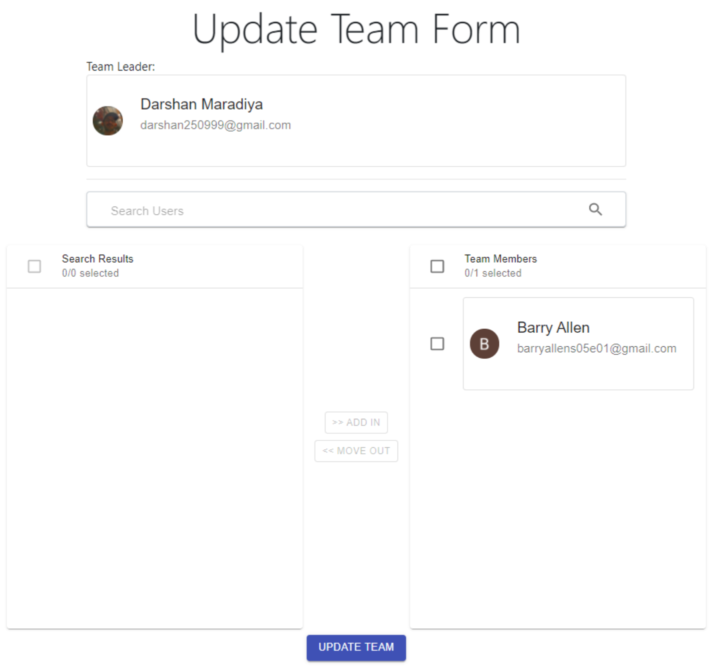
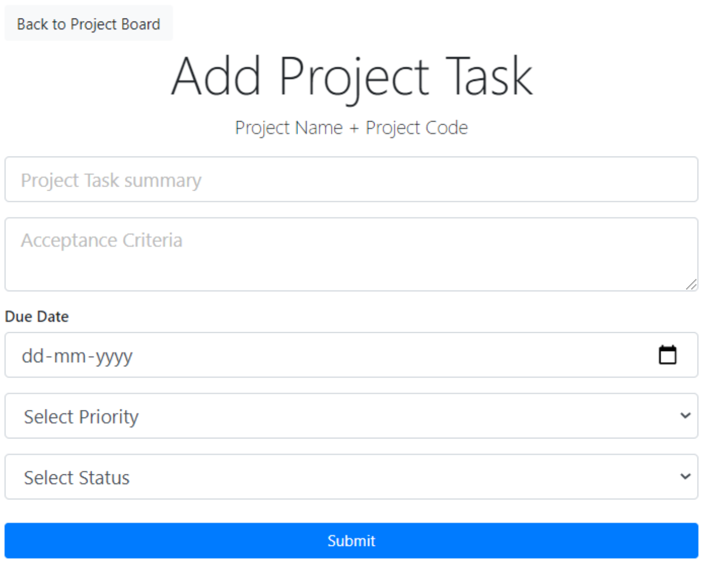


Figure 30 Update Team Form

Update Team Form is same as the create team form, except it shows the members in right tray who are already part of the team.

5.4.5 Create Project Task Form


Back to Project Board

Add Project Task

Project Name + Project Code

Project Task summary

Acceptance Criteria

Due Date

dd-mm-yyyy

Select Priority

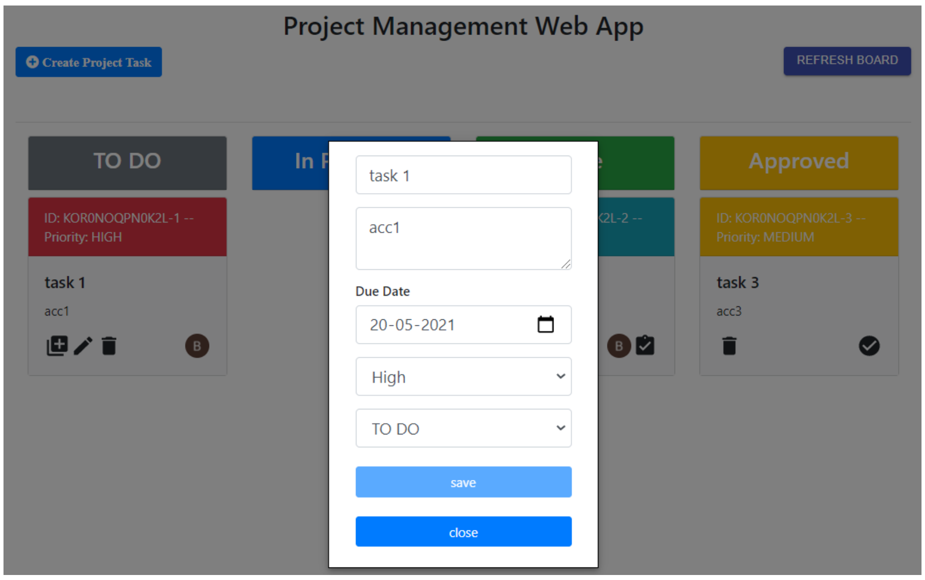
Select Status

Submit

Figure 31 Add Project Task Form

The form has two mandatory fields: ‘Project Task Summary’ and ‘Due Date’. Here Due Date must be within project’s life span. Optionally, we can write Acceptance criteria of the task. In priority, we can select from Low, Medium and High where Low is the default priority. And in Status, we can select from ‘To Do’, ‘In Progress’ and ‘Done’. The default is ‘To Do’.

5.4.6 Update Project Task Form


Project Management Web App

Create Project Task

REFRESH BOARD

TO DO

ID: KOR0NOQPN0K2L-1 -- Priority: HIGH

task 1

acc1

Approved

ID: KORONOQPNOK2L-3 -- Priority: MEDIUM

task 3

acc3

task 1

acc1

Due Date

20-05-2021

High

TO DO

save

close

Figure 32 Update Project Task Form Modal

This is different from the others; this form opens as Modal and have fields same as create project task form with pre-loaded values. Leader can update and then save them by clicking on save button. The validation still remains same as the previous form, but one more thing is that the status of the task cannot be change from ‘To Do’ if the task has pre-requisite tasks still remains to be done.

5.5 Mailbox Functionality

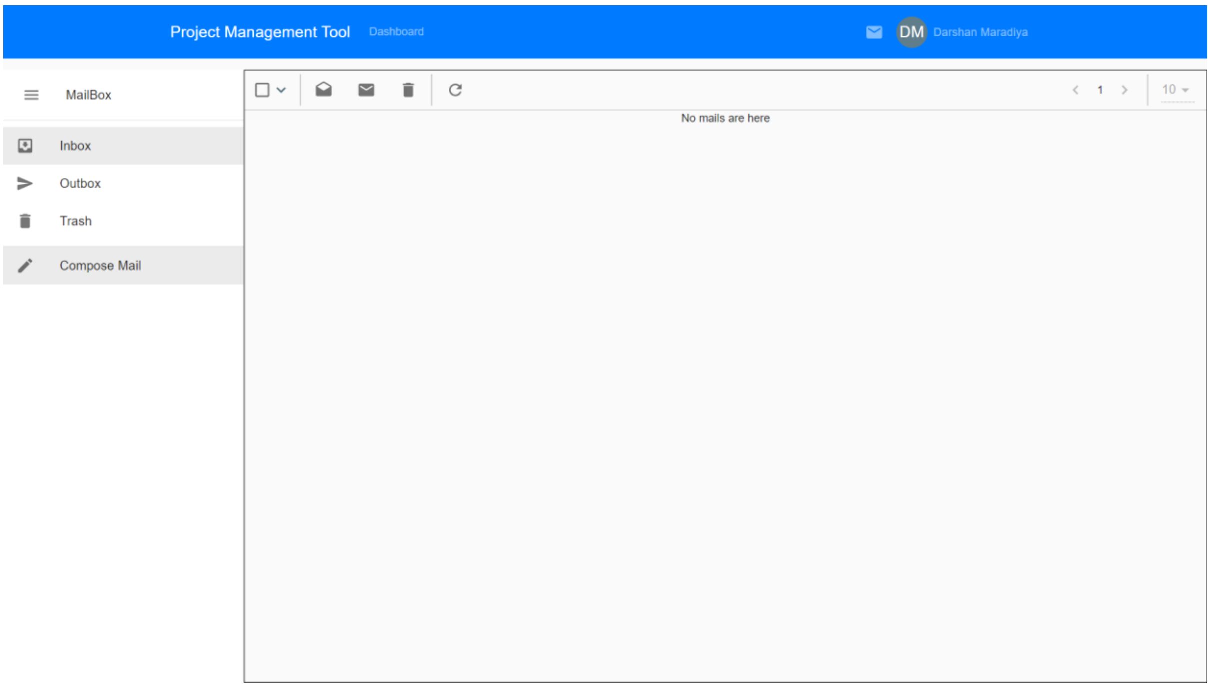


Figure 33 Mailbox

This is the view of the Mailbox. On the left there are three navigation buttons to got to the inbox, outbox or trash. Moreover, there is a button to compose a mail. In the right section, first there is a header with few operations are available to perform on mails selected in mass. This operations read/unread and delete. User can refresh the mailbox and navigate to the previous or next page.

Following is the modal that opens when we click on compose mail button. This form has 3 fields for recipient, subject (optional) and a Message.

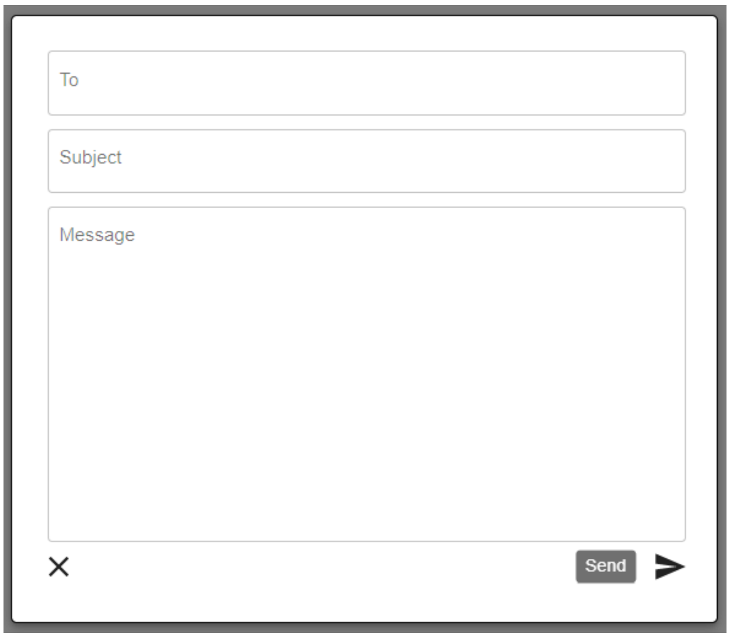


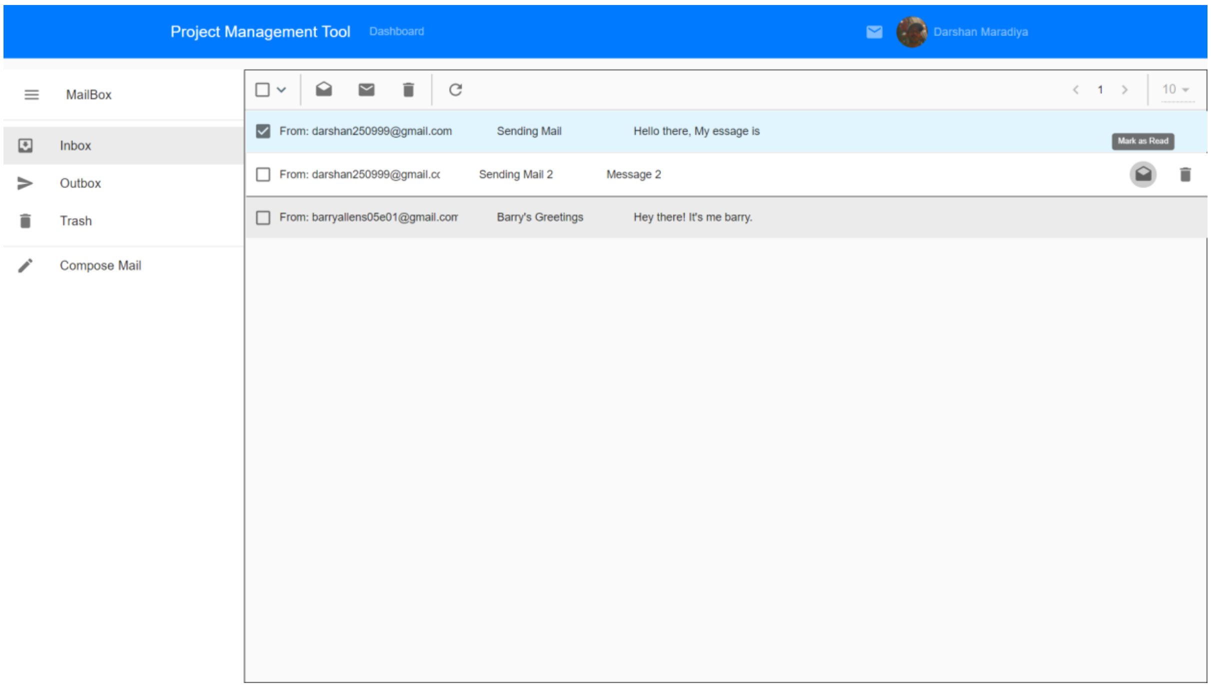
Figure 34 Compose Mail Form Modal

In the following image, there are three mails are shown:

First mail is selected, which shows that when a mail is selected then it is highlighted as light-blue.

Second mail is white, which means it is unread and since the cursor is hovering over this mail, mark as read and delete operations are available to perform this particular mail.

Third mail is not selected and it is already read. Hence it is highlighted in grey.



From: darshan250999@gmail.com Sending Mail Hello there, My essage is
From: darshan250999@gmail.cc Sending Mail 2 Message 2
From: barryallens05e01@gmail.corr Barry's Greetings Hey there! It's me barry.

Figure 35 Inbox with mails

Chapter: 6 Future Enhancement

In addition to the features added so far in the application, it is still far from the state, where it can be actually used for business in organizations. Though, as it was developed using the libraries (ReactJS and ReduxJS) and Framework (Spring Boot) which is widely accepted and used by the community, the enhancement can be easily done by anyone if wanted. For enhancement, practically there is no limit in it. Some of these ideas for modules that can be added are mentioned as below:

1) Add more roles in addition to the Team Leader and Team Member
2) Task assignment can be made convenient by checking schedules of members
3) Prepare Different types of charts for progress
4) Implement modern project management techniques and suggest the flow of tasks
5) Create history of each movements in the projects
6) Team Members can customize their own dashboard
7) Team Leader and Members can manage privacy
8) Integrate drives or provide own cloud storage to store documents
9) Auto-generated reports
10) A good messaging service can be integrated
11) If the history is managed properly, user can have timeline where they can get latest updates about projects in which they are involved
12) Specific templates like "Agile methodology" and custom templates can be provide for the project
13) Team members can raise issues they are facing
14) Comments and feedback on decided plan/activity/other events
15) Task Buckets can be personalized on Kanban board
16) Better filter/sorting/view of the tasks can be added
17) Track of deliveries
18) Integration of GitHub
19) and many more...

CONCLUSION

The Major Project was a full stack development project where I was supposed to develop a tool by building REST API at backend and Frontend for client application. By developing the application, I come up with the ideas and efforts made, coding patterns followed, problems faced and difficulties that may arise in the development phase for the actual production of any business level application. I believe that if next I will have to work in team on some project, rather than alone, then I will be able to learn and perform well on the tasks on my hands that is I can focus more on the feature I will be developing just because the work will be divided properly and as I already have this little experience of working on entire system and know that how one by one feature takes time while coding them and contributes eventually to a quality product. Through this project, I got chances to learn new libraries and famous frameworks, which brought me confidence for learning new things. Hence the project was fun with many lessons.

REFERENCES

1. Spring Framework. *Google Arts & Culture.* [Online] https://artsandculture.google.com/entity/spring-framework/m0dhx5b?hl=en.

2. Spring Boot. *spring.* [Online] https://spring.io/projects/spring-boot.

3. Spring Security Tutorials | JavaInUse. *JavaInUse.* [Online] https://www.javainuse.com/spring/sprsec.

4. What is Postman, and Why Should I Use It? *digitalcrafts.* [Online] https://www.digitalcrafts.com/blog/student-blog-what-postman-and-why-use-it.

5. Node.js - Introduction. *tutorials point.* [Online] https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm.

6. Visual Studio Code - Wikipedia. - *Wikipedia.* [Online] https://en.wikipedia.org/wiki/Visual_Studio_Code#:~:text=Visual%20Studio%20Code%20is%20a,code%20refactoring%2C%20and%20embedded%20Git..

USEFUL LINKS

Spring Full Course - Learn Spring Framework in 4 Hours | Spring Framework Tutorial | Edureka

Spring Boot Quick Start Java Brains

ReactJS Tutorial for Beginners Codevolution

Java Spring Boot JPA

React Redux Tutorial

Project Management Tools - ProjectManager.com

6 Essential Features of Project Management Software | Scoro

Originality Report