# Work Scape

## An app to track tasks for a team of people

# Table of Contents

# Application Overview

The project is a full stack application *[stack to be decided by developers],* which aims to provide a platform where an individual can create a project, add users, assign and track progress of tasks for their subordinates.

# Proposed Objectives / Salient Features

- **Data Modelling** - Building relations across various types of data and organising them into a format to support APIs and the front end view.
- **File Uploads**- Uploading files and storing them in an online bucket, retrieval, modification and deletions.
- **Real Time Communication** - Chat room for project team members where everyone can contribute to the conversation.
- **Email Notifications** - The application should have emailing facilities for corresponding activities like a member being added to the team, a task being assigned, or a task being completed and so on.
- **Real-Time Notifications** - The application should also have real time notifications for logged in individuals which would appear under a bell icon at relevant points in time. *[TBD the placement and behaviour of the widget]*
- **Authentication & Authorization** - The application would employ user authentication *[technology/architecture to be decided],* which will also take care of user roles.
- **Scheduled triggers from the backend** - there will be data cleanup tasks which will occur at the backend without user intervention.
- **UX & Accessibility** - The application should preferably be compliant with **WAI-ARIA standards** of web application accessibility.

# Module wise breakdown of features

## Authentication & Authorization Mechanism

The application implements authentication and authorization through a suitable technology. There is no guest mode in the application and all users intending to use the application features must register first. To provide these abilities to the user, there will be Login and Sign-up forms provided.

The application should also manage a session based system where a user can stay logged in from a browser, without having to login on every visit. The session should persist for a suitable amount of time, for example 3-5 days, after which the user must login again to verify their identity.

The mechanism should also cover and protect all the data access points, for example, the APIs and any non-public content within the application from any unauthorised access. The only application resources publicly available are the following:

- Login / Sign-up pages.
- Legal data like Terms & Conditions / Privacy Policy.
- Developer info / profiles and contact.
- Social media links.
- Application Documentation (optional)

All application routes and pages are also protected behind the authentication. Users cannot access any views other than the aforementioned public ones if not logged in. Conversely, the login / sign-up pages should not be accessible to a logged in user and the application should automatically navigate back to the home page if an explicit navigation is attempted.

## Registration

All users must register with a unique email address. The system should not allow an email address to register twice, and should validate this either when the email field is populated or on submitting the email for registration while throwing a proper error message.

Registering to the application is a three step process.

- **Step 1:** The user visits the registration page, where they are presented with the basic info fields:
    - Full Name (free text upto 35 characters)
    - E-mail address (valid email pattern)
    - This step would also verify that the email is not already registered in the system.
- **Step 2:** The user creates a password for their account with *password & confirm password* fields. The field values must match.
- **Step 3:** Email verification. Once the fields on step 1 and 2 are satisfied, and the user clicks on the submit button and their account is created.
    - A verification email is sent to the entered email address with a unique alphanumeric 6-8 digit code, which would be valid for a finite amount of time, for example, 30-45 minutes.
    - The user enters the code received in the email within this time frame, and submits step 2. Once the code is verified, the user is registered successfully and is redirected to the login screen where they can login using their new account credentials.

**Special Scenarios to be handled.**

- **Email Does not arrive at the provided mail address.**
    - The user can request another verification email after 2 minutes, there should be a counter provided on the screen for when another mail can be requested. Upon clicking this, another email will be sent to the provided email address, and the previous one would be invalidated.
- **The user is unable to receive email on the mail address and/or the address is invalid.**
    - The user can skip the verification process, and continue to login to the system as an unverified user. However, in this scenario, the user will have limited access to the application.
    - The user can only create projects and add other members to it, but they cannot contribute to other projects.
    - An unverified user is not discoverable by the user search system.
    - An unverified user can update/verify their email address from their account settings page, which will require the same email verification process as during registration.

## Login

The login page consists of two fields, *email address* and *password.* Once the fields are populated and submitted, the system should validate that the email is registered and the password is valid.

- If not matched, it should throw appropriate error messages.
- Once validated, the user will land to the Homepage / Dashboard.

**Account Locking**

**A User's account can be locked in two scenarios, and for either, a correct password will not be enough to admit the user. The user will have to verify their email address in order to regain access.**

- **Case 1:** Too many failed login attempts
    - The system tracks login attempts by the user, and after 3-5 failed attempts, the user's account will be locked.
    - After successful verification of the email, the user will be required to create a new password before logging into the application.
- **Case 2:** Continued inactivity
    - If the user has not logged into the system for more than 60 days, the user's account will be locked.
    - Only verification is required, and there's no need to create a new password.

## Password Reset from Login Page

The user has a link on the login page which they can use to reset their password if they forget it. The password reset flow will be a three step process.

- **Step 1:** User enters their email and submits. The system verifies whether the email is registered in the system. If they're not found, they will be notified of the same and presented with a link to the registration page for creating a new account.
- **Step 2:** The system sends the user an email with a verification code, similar to the registration flow. And the user has to enter the same into the input field provided. Once the code is verified, the user is taken to step three.
- **Step 3:** Here the user needs to set up a new password and enter it again into the confirm password field. Once the passwords match, the user's password will be updated in the system, and they will be required to login with their new password through the login screen.

## User Account Settings

The user account settings is a globally accessible page for a logged in user. The link is provided in the global navigation at all times. This page allows the logged in user to edit certain parameters of their account. The account setting page will have the following sections:

**Note:** The user account settings page should clearly indicate whether their email address is validated or not, with a quick access to validate it on the fly.

**Note:** Any changes related to a user's account will require a password validation.

### Basic Details

The user can edit basic details of their account namely, **Phone number** and **full name**.

These will be an in-place edit, meaning, once the user clicks on the edit button, the read only fields will transform into input fields with an additional field for password which will be mandatory. After filling all the fields, the user can click on *save* to update their entries permanently in the system.

## Email Verification / Change

User’s can change their email address provided in the system. For this, an edit button would be provided along with the email address field. Upon clicking this, the user can enter a new email address. Saving the change would require validation of the new email address.

**The Email address provided by the user can be unverified.** In this case, there would be an additional button along with the edit button which enables the user to verify the email address. Upon clicking the button an email will be triggered to the mail ID, and an input will appear for the user to enter the verification code.

**Note:** From one logged in session, the user can generate only one verification code, i.e. a button to resend the code on the email address will not be provided.

## Password Update

Updating password requires 3 fields, namely, **current password, new password** and **confirm new password.** Once all three fields are satisfied, the user can submit the form.

**Note:** This change requires an email verification for additional security.

## Display Picture

The system will generate a default display picture using the user's initials, which the user can change with an actual image. The user can upload an image file which will be associated with the user's account and will be shown at all places, wherever the user's info is required, for example, in chat window and project cards etc. For this the user can upload an image file of their choice with a suitable file format and file size rules provided so as to make the application work with the images properly.

**Proposed and optional feature:** The system may implement file compression for quick access to images in places where a full sized image is nor required.

### User Logout

The logout button is always available at the user account settings page, which the user can click on to logout of their session. The application should show a confirmation dialog prior to actually logging the user out of the system. Once logged out, the application would redirect to the login page.

**Proposed Feature:** Application colour theme preference settings *(dark / light mode)*.

# Global Navigation

The global navigation would be a 2 part item, with one static throughout the application and another appearing when required.

**Static Block:**

This block conventionally appears on the top of the application body and will contain 3 items:

- Application Title *[also optionally a breadcrumb of the user's location]*
- Notification icon *[covered in detail in the Notification section]*
- A link to user account settings.

**Dynamic Block:**

This block appears when the user is viewing a particular project. It contains 4 items.

- Link to project details page ***(default)***
- Link to Tasks List
    - **For Creator:** Shows all tasks grouped by status
    - **For Contributor:** Shows only my tasks grouped by status
    - [covered in detail in the tasks section]
- Link to Project Group Chat
- Link to Project Members List
    - **For Creator:** Shows all members with actions
    - **For Contributor:** Shows all members with hierarchy
    - [covered in detail in the project workflow section]

# Home Page / Dashboard

The dashboard is the first page the user will see upon logging into the application. It will consist of the following sections.

- A list of projects the user is part of separated into three groups, namely, *my projects, contributing to* and *completed.* The completed list is shown on demand and is behind a toggle switch.
- The first list (my projects) is a list of projects that the current user has created and is an owner to.
- The second list (contributing to) is a list of active projects in which the current user was added by someone else.
- The third list (completed) is a list of all projects which are marked as completed and owned by the logged in user.
- A button to add a new project from the dashboard. This button can be part of the owned projects list as well, considering accessibility. More details on the Projects section.

## Owned Projects.

The owned project cards will contain the following:

- The Project Title
- Creation Date, *[optionally age]*
- Number of users, *[preferably as an avatar stack]*
- Number of tasks, with link to tasks view
- A Direct link to chat link

## Contributed projects.

Contributing project card will have the following blocks:

- The Project Title
- Creation Date, *[optionally age]*
- Number of users, *[preferably as an avatar stack]*
- Number of tasks assigned to me, with link to tasks view.
- A Direct link to chat link

## Completed Projects.

Completed project card will be similar to the owned project card, but the colour will be greyed out. All the navigation will still work, but any actions within the inner page will not be accessible. There will be a single button to delete the project.

**Note:**

All these lists will be horizontally scrolling containers or carousels to keep screen space usage to a minimum. The lists would preferably be sorted based on their latest task activities, although the sort criteria can be modified by the development team as per their convenience.

# Projects

At the heart of the application lies **Projects.**

A Project is an individual entity under which all other features of the application take place and work. A project can be created, modified and deleted by an individual who would by default be the project owner. Contributors to a project have restricted access to the project and its tasks. More details in the tasks section.

**Projects cannot be transferred i.e. the owner of a project cannot be changed.**

## Project Parameters

Each project will have the following details:

- **Title** - populated/altered by the creator.
- **Description** - populated/altered by the creator.
- **Team Size** - populated/altered by the system, updates when a member is added or removed from a project team.
- **Status** - populated/altered by the system and the creator in one case.
- **Progress** - Dynamic field, populated/altered by the system, contains a count summary of the tasks.

## Project Workflow

- To create a project an individual must be a registered user. *See authentication module for more details.*
- Once registered, any user can create a project through the create project button on the dashboard.
- The creator of the project is also the owner of the project, and has the top level privileges of the project as a whole, namely, managing team, creating tasks and assigning to members, altering the details of the project and ultimately, deleting the project.
- A Project can have the following states:
    - **Created:** The project is created, but all the details have not been populated or members have not been added to the team.
    - **Defined:** The project is said to be defined once it is created, all the details have been populated, and a team has been created by adding members to the project.
    - **In Progress:** Once a task has been created under a project, and assigned to any of the members in the project team, the project is said to be in progress. **Note:** The project is in progress only if at least one task has been created, assigned and in progress. Conversation in the project chat will not have any impact on the project status.
    - **Completed:** The transition from *In-Progress* to *Completed* is not automatic and has to be done by the project owner. A project can be marked completed only if all tasks in it are marked as completed. If there are any tasks in the pending state, the project cannot be marked

as completed. **Note:** Once the project is marked as completed, it is not deleted from the system. The data is persisted until the project owner explicitly deletes it.

- **Creating a project:**
    - The project creation window is accessible from the dashboard and has the following fields.
        - Title *[mandatory]* - Free text, upto 80 characters
        - Description *[mandatory]* - Free text, upto 256 characters
        - Team *[optional]* - Autocomplete, searches in the users database for registered and verified users. If no user is found matching the search, the field should provide a button to add an external user.
    - If a registered user is added to the team, they will get an email informing them the same.
    - If an external user is added, they will get a link to create an account after which, they will be automatically added to the team.
    - Once all the fields are populated, the form will be submitted and a project will be created which will show up in the home page.
- **Marking a project as completed:**
    - A project can be marked as completed only after all tasks in the project are in completed status.
    - A menu button will be present in the project menu to mark it as completed.
    - Once a project has been marked as completed, all the actions on the project will be disabled, including tasks, member management, chat room etc. but the data will not be deleted.
    - Completed projects will not be shown in the dashboard, and will be accessible by a toggle switch that will fetch the completed projects in the dashboard.
- **Deleting a project:**
    - Only completed projects can be deleted.
    - Clicking on the delete button will open a confirmation dialog box stating that the action cannot be undone.
    - If the user confirms the delete actions, all data associated to the project will be removed permanently. This includes all project details, tasks data, chat messages, any files uploaded, the members list *(users themselves will not be removed from the system.)*.

# Task

A task is an individual piece of work that is part of a project. It is created and assigned to one of the members added to the team by the project owner. Only the project owner has the authority to create and assign a task.

**A Task may contain the following information:**

- **Title:** required, upto 80 characters.
- **Description:** required, free text, upto 500 characters.
- **Assignee:** One of the members added to the project.
- **Estimated Effort:** Time duration in days / hours etc.
- **Due Date:** The deadline, before which the task is supposed to be completed.
- **Attachments:** One or more files of any type relevant to the task. Should have an upper limit on the number of attachments, and may also have an optional upper limit of file size.
- **Status:** One of New, In Progress, On Hold, Blocked, Completed.

## Tasks List View

The tasks view is one of the 5 views inside the project details view. The common fields in each card are as follows. These cards will be grouped by their status. Each group will be visible on a need-to-show basis

- **Task Title**
- **Assignee** *[shown in case of creator, for contributor, it is not required.]*
- **Due Date:** will follow a relative date style.
    - For a task due in 2+ days, it should show the date in dd-mmm-yyyy format. For example **due on 24-Aug-2022.**
    - For a task due in 2 days, it should show **due in 2 days.**
    - For a task due in 1 day, it should show **due tomorrow.**
    - For a task due on the current day, it should show **due today.**
    - For all tasks which are past their due date, it should always show the number of days it is past due date, a**s n days overdue.**
- **Blocking Task**: shown for a task which is in the blocked list, with the blocking task title.
- **Number of attachments:** Optional

**Order of groups:**

From top to bottom, the lists should follow the order of priority as follows.

**Past due ⇢ Blocked ⇢ In Progress ⇢ New ⇢ Completed**