# TellMyStory

# Question Scenario

The application allows the story writers to connect with publishers. The user can write the stories using available templates. He can select the category (such as horror, rom-com, fictional etc.) and post it. User will be able to see all the posts shared by other users along with the publisher name that it approved by. User can send request to any publisher, asking to publish his post in their newspaper or magazines. It is a role based login, where publisher can also see the requests sent to him. He can approve or reject the post.

After the registration/login, user will be able to see below tabs.

1. Feed -> It displays all the posts share by all the users. It remains same for both user and publisher.
   1. User can filter the posts on their categories.
2. Settings -> It consists of below tabs
3. My profile -> It consists of basic details. It will be same for both user and publisher.
4. My posts -> It displays all the posts shared to logged in user. It displays all the posts that approved by him to logged in publisher.
5. Approved -> It displays all the posts that got approved by publisher. It will be displayed only to the user.
6. Rejected -> It displays all the posts that got rejected by publisher. It will be displayed only to the user.
7. Logout -> It will be same for both user and publisher.

# Frontend Objectives

# Instructions

1. Implement the filter functionality in feed page to filter based on the categories selected by user.
2. Implement get API’s to both User and Publisher flow.
3. Integrate postNewStory API into new post screen.

***A Project with the desired folder structure has been already created***. ***You are not allowed to change folder structure here.***

You are allowed to make any changes to the below files.

1. feed.component.html
2. feed.component.ts
3. postNewStory.component.html
4. postNewStory.component.ts
5. myPost.component.html
6. myPost.component.ts

**Build, Run and Deploy**

* Load the project in VS Code.
* Install the packages.
* You can use the build command and deploy the application on the server.
* Start the development server.

# Backend Objectives

The application uses the APIs mentioned below. where some of the controllers and services have definitions and others have blank definitions, requiring you to develop logic to fulfil the objectives.

1. We have defined the below methods in the ***Controller*** Layer. You must complete the body of it.

@GetMapping("/getPost")

**public** ResponseEntity<ResponseDto> getMyPosts (@RequestParam Integer userId)

@PutMapping("/updateStatus")

**public** ResponseEntity<ResponseDto> updateStatus (@RequestParam Integer postId, @RequestParam String status)

1. We have defined below methods in ***Service*** Layer. You must complete the body of them.

List<NewPostDto> getPost (Integer userId)

Fetch all the posts which are uploaded in case of user. Fetch all the posts which are approved by him in case of publisher

NewPostDto updateStatus (Integer postId, String status)

Fetch the post by passing postId and update its status as specified

# Instructions

You are allowed to make changes only in below files.

1. TellMyStoryController.java

2. TellMyStoryServiceImple.java

You are not allowed to make any changes in below files.

1. application.properties
2. pom.xml

You are not allowed to make any changes in the files present in below packages.

1. com.example.tellmystory.constant

2. com.example.tellmystory.dto

3. com.example.tellmystory.entity

4. com.example.tellmystory.exception

5. com.example.tellmystory.handler

6. com.example.tellmystory.repository

# Build, Deploy and Run

1. Load the project in the IDE
2. Update the maven dependency
3. mvn life cycle to build & deploy
4. Run the project