

# 1. Project Title

Town hall board - A place for users to write and post articles to and read the published articles, with content moderation by editors and admins.

## Project Type

- Web Application
  - POC for Senior Project (No)
- 

# 2. Team Members

- Member 1: Min Khaung Kyaw Swar
  - Member 2: Kyaw Zeyar Hein
- 

# 3. Problem Statement & Motivation

## What problem does this system solve?

User that want user-written articles often rely on social media or forums, which lack structured publishing, version control, and approval before content goes public. This system would let users publish written works while ensuring quality and safety through editor approval and site moderation.

The site could even serve as an online novel site for aspiring writers.

## Who is the target user?

Primary: readers who browse and comment on published works.

Secondary: creators who want to submit works for publishing.

Staff: editors who review/approve submissions, and admins who moderate content and users.

## Why does this problem matter?

Without approval and moderation, platforms can quickly fill with low-quality or harmful content. A simple workflow (submit → review → publish) supports community publishing while keeping trust and content quality.

---

## 4. Project Scope & Features

### Core Features

- Public browsing and reading of published works
  - Search and filtering (by title/keyword, tags/categories, author, date)
  - User authentication with roles (Reader/Creator, Editor, Admin)
  - Work submission by users (create/edit drafts, submit for review)
  - Editorial review workflow:
    - Editors can approve/reject works
    - Only approved works become publicly visible
  - Comment system on published works (create/edit/delete own comments)
  - Moderation tools:
    - Admin can hide/delete comments
    - Admin can remove/hide works that violate rules
    - Admin can manage user access (e.g., suspend/ban)
  - Basic creator profile page (list of their published works)
- 

## 5. Data Models

### **Entity 1: User**

**Fields:** name, email, passwordHash, role (creator/editor/admin), bio, status (active/suspended), createdAt

**Operations:** Create, Read, Update, Delete

### **Entity 2: Work (published article)**

—

**Fields:** title, content, authorId, tags, categoryId, status (draft/submitted/approved/rejected/published/hidden), submittedAt, approvedAt, publishedAt, createdAt, updatedAt

---

**Operations:** Create, Read, Update, Delete

**Entity 3: Drafts (the 5 saved versions of an article )**

**Fields:** title, content, authorId, workId, pinned ( don't delete ), createdAt

**Operations:** Create, Read, Update, Delete

**Entity 4: Comment**

**Fields:** workId, userId, body, status (visible/hidden), createdAt, updatedAt

**Operations:** Create, Read, Update, Delete

**Entity 5: Review**

**Fields:** workId, editorId, decision (approved/rejected), feedback (optional), createdAt

**Operations:** Create, Read, Update, Delete

**Entity 6: Category**

**Fields:** name, description

**Operations:** Create, Read, Update, Delete

---

## 6. Technology Stack

**Frontend:** React

**Backend:** Next.js

**Database:** MongoDB

**Deployment:** Vercel ( Next ) + Netlify ( React )