mindbowser

Knowledge Sharing Platform with AI Assist

Technical Stack Expectations

Frontend: ReactJS
Backend: Node.js (Express)/Java
Database: MySQL

Editor:

- Use a Rich Text Editor (CKEditor / Quill / TipTap)
- Fallback to `<textarea>` if needed

AI Tools (Mandatory Exposure):
Use **any AI-powered developer tool** (ChatGPT, Cursor AI, GitHub Copilot, Claude, etc.) during development.

⚠️ The goal is **not just using AI**, but showing **how you used it effectively**.

Problem Statement

Build a **Knowledge Sharing Platform** where users can create, share, and discover technical articles.
The platform should also include **AI-assisted features** to enhance the authoring and reading experience.

> Think: *Medium + StackOverflow-lite + AI assist*

---

mindbowser

Functional Requirements

👥 User Roles

Registered Users

- Create, edit, and delete their own articles
- View and search all public articles
- Use AI assistance while writing

Public Users

- View and search published articles
- Read full article content

🔐 Authentication

- **Signup:** Username, email, password
- **Login:** Email + password (JWT-based auth)
- **Logout:** Token/session invalidation

📝 Article Management

Create Article

- Title
- Category (Tech, AI, Backend, Frontend, DevOps, etc.)
- Rich text content
- Tags (comma-separated)

Read Articles

- Publicly visible
- List view with:
    - Title

  - Short summary
  - Author
  - Category
  - Created date

Edit / Delete Article

- Only the original author can edit or delete

View Single Article

- Full content
- Author details
- Tags
- Created & updated timestamps

🔍 Search & Filter (Frontend-focused)

- Search by:
  - Title
  - Content
  - Tags

- Filter by category

🤖 AI-Assisted Features (Key Differentiator)

1️⃣ AI Writing Assistant (Mandatory)

While creating or editing an article:

- Button: **“Improve with AI”**

- AI can:

  - Rewrite content more clearly
  - Improve grammar
  - Make content more concise
  - Suggest a better title

Implementation can be:

- Real AI API (OpenAI, Claude, etc.)
- OR mocked AI responses (but structure should allow real integration)

2️⃣ AI Summary Generator

- Automatically generate a **short summary** for each article
- Used on the Home page article cards

3️⃣ AI Tag Suggestions (Optional Bonus)

- Based on article content, suggest relevant tags

🧭 UI & Navigation

Pages

- **Home Page**
  - List all articles
  - Search & filter
- **Article Detail Page**
  - Full article view
- **Create / Edit Article Page**
  - Rich editor + AI tools
- **User Dashboard**
  - List of user's own articles
  - Edit/Delete actions

- **Authentication Pages**

mindbowser

  - Login
  - Signup

Navigation Bar

- Home
- New Article
- My Articles
- Login / Logout

📦 Project Delivery Expectations

GitHub Repositories

- **Frontend Repo (Public)**
- **Backend Repo (Public)**

README Requirements (Both Repos)

Must include:

1️⃣ Approach

- Architecture overview
- Folder structure
- Key design decisions

2️⃣ AI Usage (Mandatory Section)

Explain:

mindbowser

- Which AI tools you used (ChatGPT, Copilot, Cursor, etc.)
- Where AI helped:
    - Code generation
    - Refactoring
    - SQL queries
    - API design
    - UI ideas
- What you reviewed or corrected manually

Example:
"Used ChatGPT to generate initial Express boilerplate, then optimized middleware and error handling manually."

3️⃣ Setup Instructions

- Prerequisites
- Environment variables
- Backend setup
- Frontend setup

📹 Demo Requirement

- Screen recording (Google Drive public link)
- Must show:
    - Signup/Login
    - Article creation

- AI assist usage
- Edit/Delete
- Search & filters

⭐ Evaluation Criteria

- Code quality & project structure
- API design
- UI/UX clarity
- Correct authorization logic
- Practical AI usage (not just buzzwords)
- Clean README and documentation

Assignment Submission

Submit your assignment details using the form link below.

https://forms.gle/oBCmSW1RoZaHvVxw7