

社内教育用ウェブアプリの製作

アプリ概要

- 目的
 - 本アプリは、X(旧Twitter)のようなシステムをイメージした、社内教育用のツールです。
 - 社長が普段考えていることや感じていることを投稿し、幹部メンバーがそれを閲覧することで、会社として同じ方向性を共有することを目的としています。
 - 利用方法
 - 幹部メンバー(一般ユーザー)や管理者は、投稿に対して以下のアクションを行います:
 - いいねを押すことで、投稿を確認したことを示す。
 - コメントがオンになっている投稿に対して、自分の考えや気づきをコメントとして記録する。
-

2. 主な機能要件

2.1 ログイン/ログアウト機能

1. ログイン
 - メールアドレス + パスワードでログイン。
 - 一度ログインしたらブラウザを閉じるまでセッションは失効しない。
 2. ログアウト
 - 画面のどこかに「ログアウトボタン」を設置し、ユーザーが明示的にログアウトできるようにする。
-

2.2 ユーザー管理

1. 管理者アカウント
 - 初期パスワードを管理者が任意に設定可能。
 2. 一般ユーザーアカウント
 - 管理者が手動で登録(メールアドレス・初期パスワードを設定)。
 - パスワードを忘れた場合、管理者が手動でリセット(初期パスワードを再設定)。
 3. ユーザー削除
 - 管理者がユーザーを削除可能。
 - 削除後も、そのユーザーが行った「いいね」や「コメント」は残す。
-

2.3 投稿機能(管理者専用)

1. 投稿内容
 - タイトル、本文、ファイル添付（以下に対応）：
 - 画像、動画、テキスト、**PDF**形式。
 - 最大**50MB**までアップロード可能。
 - 投稿本文内のURLは自動的にハイパーリンク化し、クリック時に別タブで開く。
 - 投稿日時を投稿ごとに表示。
 2. コメント有無の切り替え機能
 - 投稿時に「コメント欄のオン/オフ」を設定できるスイッチを設置。
 - コメント欄がオンの場合、ユーザーと管理者がコメント投稿可能。
 - コメント欄がオフの場合、いいねのみ可能。
 3. 投稿編集/削除
 - 管理者が投稿内容を修正・削除可能。
 4. 投稿の並び順
 - 投稿一覧は**時系列順（新しい投稿が上）**で表示。
-

2.4 コメント機能

1. コメント投稿（一般ユーザーおよび管理者）
 - コメント欄がオンになっている投稿にのみ、コメント可能。
 - 文字数制限なし。
 - コメントにファイル添付が可能（投稿と同じ形式・容量制限）。
 - コメント日時をコメントごとに表示。
 2. コメントの表示
 - コメントが多い場合、最初の**2～3件**のみを表示。
 - 「すべてのコメントを見る」ボタンを押すと全コメントが展開され、スクロール可能になる。
 3. コメントの編集/削除
 - ユーザー自身が投稿したコメントのみ、編集・削除可能。
 - 管理者もコメントを投稿可能で、不適切なコメントを削除できる。
-

2.5 いいね機能

1. いいねボタン
 - 各投稿に「いいね」が可能。再度押すと取り消し（トグル式）。
 - いいね数を投稿一覧画面で表示。
 2. いいねしたユーザー一覧
 - いいね数をクリックすると、別ページに遷移し、誰がよかったかをリスト表示。
 - 一覧が多い場合はスクロール可能。
-

2.6 管理者専用フィルター機能

1. コメントがオンになっている投稿の絞り込み

- 管理者がコメント欄がオンになっている投稿のみを一覧表示できるフィルター機能を追加。
 - フィルターをオンにすると、該当する投稿のみ時系列で表示される。
-

2.7 検索機能

- 投稿のタイトルや本文をキーワード検索可能。
 - 高度なフィルタリングは不要。
-

3. データ管理・バックアップ・ログ

1. データベース
 - 投稿、コメント、いいね、ユーザー情報を保存。
 - ファイル(画像や添付ファイル)はサーバー上にアップロードし、データベースにはパスを記録。
 2. バックアップ
 - 自動バックアップを設定し、定期的にデータベース全体をエクスポート。
 - 復元手順をエンジニアにドキュメント化してもらう。
 3. ログ管理
 - アクセスログや操作ログをデータベースに記録。
 - 1ヶ月分を保持し、それ以降は古いデータを削除。
-

4. デザインとUI/UX

1. モバイル優先デザイン
 - 画面幅が狭い環境を想定し、装飾は最低限に抑える。
 2. シンプルで軽快なUI/UX
 - 必要最低限のメニューとボタン配置。
 - コメントやファイル添付を含めても動作が軽快であることを重視。
-

5. 運用環境

1. ホスティング
 - 無料または低コストのホスティングサービス(RenderやVercelなど)または安価なVPS。
2. 納品形態
 - エンジニアが設定・設置まで対応し、利用可能な状態で納品。
 - ドメインやサーバー設定も含め、初心者でもそのまま利用できるようにする。

6. 納期・予算

- 希望納期: 相談
 - 予算: 10万～20万円(応相談)
 - 必要機能を優先し、それ以上の実装が必要な場合は別途見積もり。
-

7. 開発条件・注意点

1. 技術スタック
 - バックエンド: Node.js + Express
 - データベース: SQLite または MongoDB
 - フロントエンド: HTML/CSS/JavaScript
 2. セキュリティ
 - 高度なセキュリティ対策は不要だが、最低限のパスワードハッシュ化や CSRF/XSS対策を実施。
-

8. 納品物

- **Web**アプリケーション一式(動作する形でデプロイ済み)
- 操作マニュアル(管理者向け・ユーザー向け)
- バックアップと復元手順書(簡単な手順で対応できる内容)

※開発条件などについて

AI上で出てきたものなので、それ以外でも問題ありません。

AIに書かせたコードがどの程度役に立つのかは分かりませんが、時間短縮など役に立つのであればそちらも共有可能です。

9. イメージ

```
import React, { useState } from 'react';
```

```

import {
  ThumbsUp,
  MessageCircle,
  MoreVertical,
  Search,
  LogOut,
  Filter
} from 'lucide-react';
import { Card, CardContent, CardHeader } from '@components/ui/card';
import { Avatar } from '@components/ui/avatar';
import { Button } from '@components/ui/button';
import { Input } from '@components/ui/input';
import { Textarea } from '@components/ui/textarea';
import { Switch } from '@components/ui/switch';

const MainFeed = () => {
  const [showAllComments, setShowAllComments] = useState(false);

  const samplePost = {
    id: 1,
    title: "今月の経営方針について",
    content: "私たちの会社が目指すべき方向性について共有したいと思います。\\n\\n昨今のデジタル化の波を踏まえ、...",
    author: "社長",
    timestamp: "2024年12月31日 10:30",
    likes: 15,
    commentsEnabled: true,
    comments: [
      {
        id: 1,
        author: "営業部長",
        content: "理解しました。営業部でも共有させていただきます。",
        timestamp: "2024年12月31日 11:00"
      },
      {
        id: 2,
        author: "技術部長",
        content: "技術部門としての具体的な取り組みについて、補足させていただきます。",
        timestamp: "2024年12月31日 11:15"
      }
    ]
  };

  return (
    <div className="min-h-screen bg-gray-50">
      { /* ヘッダー */ }
      <header className="bg-white shadow-sm fixed w-full top-0 z-10">
        <div className="max-w-4xl mx-auto px-4 py-3 flex justify-between items-center">

```

```

<h1 className="text-xl font-bold">社内教育アプリ</h1>
<div className="flex items-center space-x-4">
  <Search className="w-5 h-5 text-gray-500" />
  <Filter className="w-5 h-5 text-gray-500" />
  <LogOut className="w-5 h-5 text-gray-500" />
</div>
</div>
</header>

{/* メインコンテンツ */}
<main className="max-w-4xl mx-auto pt-16 px-4">
  {/* 投稿作成フォーム (管理者のみ) */}
  <Card className="mb-6">
    <CardContent className="pt-4">
      <Input
        type="text"
        placeholder="タイトル"
        className="mb-3"
      />
      <Textarea
        placeholder="投稿内容を入力..."
        className="mb-3"
      />
      <div className="flex justify-between items-center">
        <div className="flex items-center space-x-2">
          <Button variant="secondary" size="sm">
            ファイルを添付
          </Button>
          <div className="flex items-center space-x-2">
            <span className="text-sm">コメント許可</span>
            <Switch />
          </div>
        </div>
        <Button>投稿する</Button>
      </div>
    </CardContent>
  </Card>

  {/* 投稿一覧 */}
  <Card className="mb-4">
    <CardHeader className="flex flex-row items-start justify-between space-y-0">
      <div className="flex space-x-4">
        <Avatar>
          <div className="w-10 h-10 rounded-full bg-gray-200 flex items-center justify-center">
            社
          </div>
        </Avatar>

```

```

<div>
  <h2 className="font-bold">{samplePost.title}</h2>
  <p className="text-sm text-gray-500">
    {samplePost.author} • {samplePost.timestamp}
  </p>
</div>
</div>
<MoreVertical className="w-5 h-5 text-gray-500" />
</CardHeader>
<CardContent>
  <p className="whitespace-pre-wrap mb-4">{samplePost.content}</p>

  {/* アクション */}
  <div className="flex items-center space-x-4 mb-4">
    <button className="flex items-center space-x-1 text-gray-500">
      <ThumbsUp className="w-5 h-5" />
      <span>{samplePost.likes}</span>
    </button>
    <button className="flex items-center space-x-1 text-gray-500">
      <MessageCircle className="w-5 h-5" />
      <span>{samplePost.comments.length}</span>
    </button>
  </div>

  {/* コメント */}
  {samplePost.commentsEnabled && (
    <div className="space-y-4">
      {samplePost.comments.slice(0, showAllComments ? undefined : 2).map(comment
=> (
        <div key={comment.id} className="flex space-x-3">
          <Avatar>
            <div className="w-8 h-8 rounded-full bg-gray-200 flex items-center
justify-center text-sm">
              {comment.author[0]}
            </div>
          </Avatar>
          <div className="flex-1">
            <div className="bg-gray-50 rounded-lg p-3">
              <p className="font-medium text-sm">{comment.author}</p>
              <p className="text-sm">{comment.content}</p>
            </div>
              <p className="text-xs text-gray-500 mt-1">{comment.timestamp}</p>
            </div>
          </div>
        </div>
      )
    )}

    {samplePost.comments.length > 2 && !showAllComments && (
      <button

```

```

        className="text-sm text-blue-600"
        onClick={() => setShowAllComments(true)}
      >
        すべてのコメントを表示
      </button>
    )}

    {/* コメント入力 */}
    <div className="flex space-x-3">
      <Avatar>
        <div className="w-8 h-8 rounded-full bg-gray-200 flex items-center
justify-center text-sm">
          私
        </div>
      </Avatar>
      <div className="flex-1">
        <Textarea
          placeholder="コメントを入力..."
          className="min-h-[60px]"
        />
        <div className="flex justify-between mt-2">
          <Button variant="secondary" size="sm">
            ファイルを添付
          </Button>
          <Button size="sm">
            コメントする
          </Button>
        </div>
      </div>
    </div>
  </div>
)}
</CardContent>
</Card>
</main>
</div>
);
};

export default MainFeed;

```