

## TASK MANAGEMENT DRAFT: BLOG APPLICATION

---

### 1. Feature: User Authentication (Login Page)

- **Ticket 1 (Backend):** Develop User Authentication API
    - **Description:** Create a POST endpoint to authenticate users using email and password. Return a JWT token upon success.  
+1
    - **API Path:** `POST /api/v1/auth/login`
  - **Ticket 2 (Frontend):** Implement Login Page UI and Integration
    - **Description:** Build the login form with email/password validation and integrate with the authentication API. Store token in local storage/cookies.
- 

### 2. Feature: Post Management (List & Search)

- **Ticket 3 (Backend):** Create Get Posts API with Search Functionality
    - **Description:** Develop an API to fetch a list of blog posts. Support filtering/searching by keywords via query parameters.  
+1
    - **API Path:** `GET /api/v1/posts?search={keyword}`
  - **Ticket 4 (Frontend):** Develop Post List Page with Search Bar
    - **Description:** Display posts in a list/grid view. Implement a search input that triggers the API call based on user keywords.
- 

### 3. Feature: Blog Content View

- **Ticket 5 (Backend):** Develop Get Post Detail API
    - **Description:** Create an endpoint to retrieve full details of a specific blog post by its ID.  
+1
    - **API Path:** `GET /api/v1/posts/{id}`
  - **Ticket 6 (Frontend):** Build Blog Content View Page
    - **Description:** Create a detailed view page to display post title, content, author, and date fetched from the API.
- 

### 4. Feature: Blog Content Edit

- **Ticket 7 (Backend):** Implement Update Post API
  - **Description:** Create a PUT/PATCH endpoint to allow authorized users to edit and update existing blog content.  
+1
  - **API Path:** `PUT /api/v1/posts/{id}`
- **Ticket 8 (Frontend):** Build Post Edit Form and Logic

- **Description:** Develop an edit interface with a Rich Text Editor. Populate the form with current post data and handle the update submission.
-