Weekly Detailed Activities for Blog Platform Development

Week 1: Partner Management Admin Portal Development

Objective: Build the admin portal for partners to manage their content, including database setup and CRUD functionalities.

Activities:

1. **Database Design and Setup:**
    - Define the table structure for storing content:
        - id (Primary Key, Auto-increment)
        - title (VARCHAR, Not Null)
        - summary (TEXT, Null)
        - content (LONGTEXT, Not Null)
        - content_category (VARCHAR, Null)
        - uploaded_date (TIMESTAMP, Default CURRENT_TIMESTAMP)
        - active_date (DATE, Null)
        - keywords (VARCHAR, Null)
        - categories (VARCHAR, Null)
        - partner_id (Foreign Key, Null) — to link content to specific partners
        - status (VARCHAR, Default 'draft') — to track content status ('draft', 'published', 'archived')
        - views_count (INT, Default 0) — to track the number of views per content
        - tags (VARCHAR, Null) — for additional categorization
        - is_featured (BOOLEAN, Default False) — to highlight special content
    - Set up database migrations and run them to create the tables.
2. **Backend Development:**
    - **Authentication & Authorization:**
        - Implement partner login and session management.
        - Set up role-based access control (RBAC) for different partner roles (e.g., author, editor).
    - **Content Management:**
        - Create endpoints for CRUD operations (Create, Read, Update, Delete) for content.
        - Implement content validation and sanitization.
    - **File Upload Handling:**
        - Set up a file upload system for images or media associated with content.
        - Store file paths in the database and handle file storage securely.
3. **Frontend Development for Partner Portal:**

- **Dashboard:**
  - Build a dashboard for partners to view content metrics (views, engagement).
- **Content Editor:**
  - Develop a rich text editor for partners to create and format content.
  - Include fields for title, summary, content, categories, keywords, and active date.
- **Content List & Management:**
  - Implement a list view for partners to manage existing content.
  - Include features to filter and search content.

4. **Testing:**
   - **Unit Testing:**
     - Test individual components and backend endpoints for functionality.
   - **Integration Testing:**
     - Ensure that all components work together seamlessly, especially CRUD operations.

Week 2: Super Admin Portal Development

Objective: Build the portal for site owners to manage partners, content moderation, and site-wide settings.

Activities:

1. **Database Design and Setup:**
   - Define additional tables:
     - partners (id, name, email, status, created_at, updated_at)
     - users (id, email, password, role, partner_id, created_at, updated_at)
     - subscriptions (id, user_id, partner_id, start_date, end_date, status)
     - site_settings (id, setting_key, setting_value, created_at, updated_at)
2. **Backend Development:**
   - **Admin Authentication & Authorization:**
     - Implement admin login and secure session management.
     - Define roles and permissions (e.g., super admin, moderator).
   - **Partner Management:**
     - Create endpoints to manage partners (CRUD operations).
     - Implement features for approving, suspending, or deleting partners.
   - **Content Moderation:**
     - Develop tools for reviewing, approving, or rejecting content.
     - Set up notifications for partners regarding content status changes.

    - **Site Settings:**
        - Implement a module for managing site-wide settings (e.g., themes, contact info).
3. **Frontend Development for Admin Portal:**
    - **Dashboard:**
        - Develop an admin dashboard displaying key metrics (total content, active partners, user activity).
    - **Partner Management Interface:**
        - Build a UI for managing partner accounts and viewing partner content.
    - **Content Moderation Tools:**
        - Create an interface for reviewing and moderating content submissions.
    - **Settings Management:**
        - Develop a form-based UI for updating site settings.
4. **Testing:**
    - **Unit Testing:**
        - Test admin functionalities, including partner and content management.
    - **Security Testing:**
        - Conduct security audits to ensure the admin portal is protected against unauthorized access.

Week 3: User-Facing Website Development

Objective: Develop the front-facing website where users can read content based on their subscriptions.

Activities:

1. **Frontend Development:**
    - **Homepage:**
        - Design a homepage featuring a mix of content, highlights, and featured partners.
    - **Content Browsing:**
        - Develop category-based and keyword-based content browsing features.
    - **Subscription and User Login:**
        - Implement user registration and login systems.
        - Create a subscription flow allowing users to choose and subscribe to partners.
    - **Content Display:**

        - Design a content display page with options for related content, comments, and sharing.
        - Implement content view count increment functionality.

2. **Backend Integration:**
    - **Content Delivery:**
        - Develop APIs to fetch and deliver content to the frontend based on user subscriptions.
    - **Subscription Management:**
        - Integrate backend logic to handle user subscriptions and access control.
3. **UI/UX Design:**
    - **Responsive Design:**
        - Ensure the website is fully responsive and accessible on all devices.
    - **User Experience Optimization:**
        - Focus on clean, intuitive navigation and fast content loading times.
4. **Testing:**
    - **Usability Testing:**
        - Test the website with actual users to gather feedback on usability and design.
    - **Performance Testing:**
        - Optimize the website for performance, focusing on loading speeds and resource usage.

Week 4: Authentication and User Management

Objective: Build and refine authentication and user management features, ensuring secure access and a smooth user experience.

Activities:

1. **Authentication System:**
    - **User Registration:**
        - Implement a secure user registration process with email verification.
    - **User Login:**
        - Develop a login system with session management and secure password handling.
    - **Password Management:**
        - Add features for password reset and account recovery.
2. **User Profile Management:**
    - **Profile Dashboard:**
        - Allow users to view and edit their profile information.

   - **Subscription Management:**
      - Provide users with tools to manage their subscriptions, including renewals and cancellations.
3. **Security Enhancements:**
   - **Data Encryption:**
      - Implement data encryption for sensitive user information.
   - **Session Security:**
      - Use secure cookies and implement measures to prevent session hijacking.
4. **Testing:**
   - **Security Testing:**
      - Conduct thorough testing to identify and address security vulnerabilities.
   - **User Testing:**
      - Test the authentication flow and profile management features with real users to ensure ease of use.

Week 5: Payment Integration and Final Preparations

Objective: Integrate payment processing for subscriptions, finalize the platform, and prepare for launch.

Activities:

1. **Payment Gateway Integration:**
   - **Research and Selection:**
      - Choose a payment gateway that supports your region and platform requirements.
   - **Integration:**
      - Implement the payment gateway APIs for handling subscriptions and payments.
   - **Testing:**
      - Test payment flows, including subscription activation, renewals, and cancellations.
2. **Final Testing and Quality Assurance:**
   - **Comprehensive Testing:**
      - Conduct final testing of all features, focusing on edge cases and potential issues.
   - **Load Testing:**
      - Ensure the platform can handle expected traffic levels, especially during peak times.
3. **Deployment Preparation:**
   - **Server Setup:**

        - Prepare the production environment, including servers, databases, and security configurations.
    - **Data Migration:**
        - Ensure that any necessary data is migrated to the live environment.
4. **Launch and Post-Launch Support:**
    - **Launch Plan:**
        - Coordinate the platform launch, including marketing and user onboarding.
    - **Monitoring and Support:**
        - Monitor the platform for any issues post-launch and provide prompt support.