

# Blog Design

*by Jaaw*

Anthony Liang, Anthony Chen, Jerry Chang, Winston Venderbush  
*Aliang8, achencoms, jerrychang22, wvenderbush*

**Abstract:** Create a single-author blog (referred to as admin or author). Homepage shows feed of blog posts, each post pointing to its own page that displays the post as well as comments on that post. Admin can edit and post from an admin page. Posts can be separated by time stamp and category on Category page. Users can comment on posts.

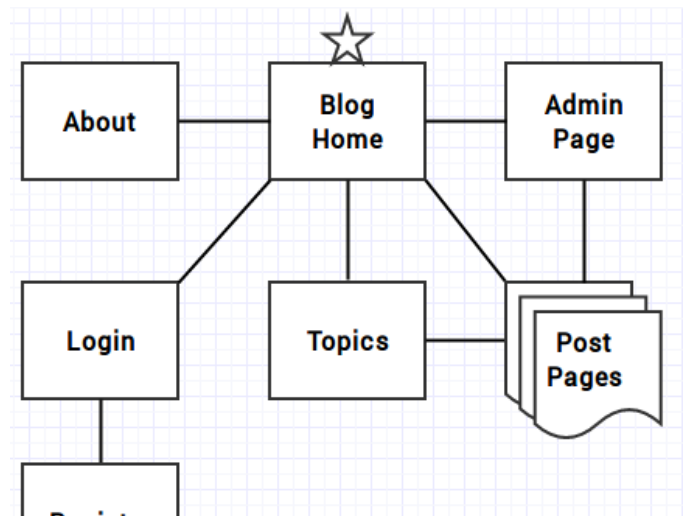
## Site Map

- Main Page / Blog Page
  - Post Feed
- Admin Page (for writing/editing posts, etc.)
- Post Pages
  - Comments under each post (on the post's own page)
- Login Page (handles admin and commenter login)
  - Register Page (designed for new commenters)
- About Page
- Topics (posts sorted by category tag)

## Site Map Linking

- Main Page / Blog Page
  - Admin Page
  - Post Pages (linked in feed)
  - Topics Page
  - Login Page
    - Register Page
  - About Page

## Visual Site Map



## Program Components

- Date Tags
- Category Tags
- Comments on Posts
- User Accounts
  - Login system
  - Registration system
- Post Creation (Admin System)

## Database Schema

- User Database
  - Username
  - Hashed Password
  - Favorite Posts
- Posts Database
  - Username of Author
  - Post ID
  - Time Tag
  - Category Tag
  - Content of Post
- Comments Database
  - Username of Commenter
  - Content of Comment
  - Time tag
  - Comment ID
  - Post ID/Comment ID of associated post
  - Comment ID (for associated replies)
    - Top level comments have an associated comment ID of 0

## Component/Database Relationships

- Posts can be viewed, organized by category tag, on the Topics Page
- Each blog post is identified by its Post ID
  - Each comment identifies the post it is linked to with that post's Post ID
  - Sub-comments (ie. replies to comments) identify the comment they are linked to with that comment's comment ID
- Each post has its own category tag and time tag
- Each comment has a time tag

# Task Delegation

- Flask
  - Create Flask App
  - Create page routes
  - Implement backend functions in routes
  - Data processing from web browser
    - Login-in, log-out
    - Comments
    - Post submissions
- HTML/CSS (Front End)
  - Writing HTML Templates
  - Creating CSS templates
  - Generally formatting blog
- SQL Databases
  - Organizing databases system
  - Work with back end programmer to:
    - Implement SQL calls
    - Ensure proper transfer of data to and from databases
- Python (Back End)
  - Creating back end functions
    - Login/logout
    - Adding posts
    - Commenting
    - Separating posts by category
    - Separating posts by time stamp (archives)
    - Admin/Posting page