

Photo ABCD Project Plan

1. Introduction

- a. Primary Goal: Build a photo blogging platform that organizes blogs both alphabetically and chronologically with user role management and customization features.
- b. Tools to be used: Github, Trello, XAMPP, and Bluehost

2. Planning

- a. Milestones
 - i. Alphabetical and Chronological Blog Compilation
 - ii. Blog Post Creation for Logged-in Users
 - iii. Customizable Alphabet Book with Progress Tracking
 - iv. Search, Filter, and Dashboard options
 - v. Sharing, Collaboration, and Privacy Controls
- b. Timeline
 - i. TBD
- c. Development Framework
 - i. PHP as the server-side
 - ii. MySQL for the database
 - iii. HTML/CSS/JavaScript for front-end
- d. Database Design - Schema
 - i. Blog Data (title, description, date, privacy settings)
 - ii. User Data (email, roles: user/admin)
 - iii. Alphabet Book Progress Data
 - iv. Event Date, Blog Creation/Modification Dates

3. Design

- a. Wireframes/Mockups
 - i. Blog Creation Page for Registered Users
 - ii. Blog Listing Pages (alphabetical/chronological)
 - iii. Admin & User Dashboards
 - iv. Home Page View for Visitors and Logged-in Users
 - v. Photo-only Display View

4. Development

- a. Backend

- i. Set up MySQL tables for blog entries, users, and Alphabet Book data
 - ii. Implement blog post creation and storage in MySQL
 - iii. Create user authentication and role management in MySQL
 - iv. Develop the privacy filter for blog posts (private by default)
 - v. Build backend logic for sorting blogs alphabetically and chronologically
 - vi. Create photo upload handling with default image assignment
 - vii. Create Admin Dashboard for user management, blog management, reports, and preferences
 - b. Frontend
 - i. Develop blog listing and creation pages
 - ii. Implement search and filtering options (eg. alphabetical, date range)
 - iii. Set up customizable Alphabet Book with progress tracking
 - iv. Develop HTML compilation feature for users to export their Alphabet book or chronological blog collection
 - c. Integration with bluehost
 - i. Ensure the application runs both locally and on bluehost without the need to hardcode URLs
- 5. Testing
 - a. Test individual modules such as blog creation, user login, and privacy settings
 - b. Test the full flow for registered users (ie. Account creation, login, blog creation, search, filtering, sharing)

Base Plan

- 1. Set Up Development Environment
 - a. Install XAMPP (If you haven't already)
 - b. Initialize GitHub Repository
 - i. Prof might be taking care of this for us.
 - c. Trello Board
 - i. Break down the below steps into different project tasks
 - d. Create Project Structure

- i. Organize directories for HTML, CSS, JavaScript, PHP, and images
- 2. Database Design & Implementation
 - a. Design the MySQL Database
 - i. In progress by Justin and Logan
 - ii. Tables needed: users, blogs, alphabet_books, photos, etc
 - iii. Key columns: User roles, blog details (ie. title, description, date, privacy settings), and relationship between blogs and users
 - b. Code Database Schema
 - i. Write SQL scripts to create tables and relationships
 - 1. phpMyAdmin could be a useful tool to manage/visualize data
 - c. Create a db.php File
 - i. PHP file that will manage database connections for all interactions with the MySQL database
- 3. User Authentication
 - a. Sign-Up/Login Forms (HTML/CSS)
 - i. Build a simple HTML form for user registration and login
 - ii. Data can be hardcoded for testing until we are ready to connect it to our database
 - b. Backend (PHP)
 - i. Write PHP scripts to handle user registration (store these in the “users” table) and login (validate against the database)
 - c. Admin/User Roles
 - i. Just need to define admin/user roles in the users table so that we can manage access accordingly
- 4. Blog Creation
 - a. Create a form for blog creation that includes inputs for title, description, photo(s), and event date.
 - i. We can use MySQL or PHP functions in order to gather creation/modification dates
 - b. Need to implement file upload functionality for blog photos
- 5. Alphabetical and Chronological Blog Sorting
 - a. Alphabetical Sorting

- i. Create a blog listing page where users can filter by the first letter of the blog titles
 - ii. In the backend, query blogs using SQL's "ORDER BY" clause with "title"
 - b. Chronological Sorting
 - i. Add an option for sorting by event date or blog creation date.
 - 1. We can use SQL "ORDER BY" clause with "event_date" to display the blogs in chronological order.
- 6. Customizable Alphabet Book
 - a. Frontend
 - i. Display the blogs in a grid format where each letter corresponds to a blog entry
 - ii. Implement a progress bar to show how many letters have been covered, possibly show which ones are missing
 - b. Backend
 - i. Query blogs grouped by alphabet and calculate progress based on how many of the 26 letters have associated blogs
- 7. Privacy and Sharing Settings
 - a. Privacy Filter (Backend)
 - i. Add a "privacy" column to the "blogs" table
 - ii. Set the blogs to private by default, create a form where users can toggle the privacy setting
 - b. Public and Private Blogs (Frontend)
 - i. Ensure that public blogs are accessible to everybody, while private blogs are only accessible by the blog creator
 - ii. Implement sharing via the ability to toggle blog visibility
- 8. Search and Filter Options
 - a. Alphabetical Search
 - i. Add a search input field to filter blogs by their first letter/a specific word
 - ii. In the backend, use SQL's "Like" operator to filter the results based on input
 - b. Date Range Filter
 - i. Add a date picker for filtering blogs between two given dates

- ii. Implement a backend query using SQL's "Between" clause to retrieve blogs within the date range

9. Interactive Dashboards

- a. User Dashboard (Frontend)
 - i. Create a page where logged-in users can see all of their blogs. Include editing/deletion options.
- b. Admin Dashboard
 - i. Create an admin panel that shows summary reports such as number of blogs or user count.
 - ii. Use data tables to manage users and blogs
- c. Backend
 - i. Allow admin to manage all users and blogs (ie. have the ability to modify or delete them).

10. Photo-Only View and Chronological View

- a. Photo-Only View
 - i. Add a filter that hides all text from the blogs and only shows the photos and blog titles
- b. Chronological View
 - i. Implement a chronological view that organizes blogs based on the event date

11. Download/Print Feature

- a. HTML Compilation
 - i. Need to implement a printable view of the Alphabet Book and blog timeline using HTML
 - ii. Must ensure proper page breaks in HTML

12. Deployment

- a. Need to test the application on different screen sizes and devices.
- b. Must thoroughly test each feature for functionality, security, and responsiveness
- c. Once everything is working locally, the application must be migrated to bluehost
 - i. Functionality must once again be tested on the live server in order to find any deployment related issues.