

Blog App Project Plan

Project Overview

This is a **learning-based project** to build a full-stack blog application using the following technologies:

- **Frontend:** EJS + Tailwind CSS
 - **Backend:** Express + Node.js
 - **Database:** MongoDB + Mongoose
 - **Authentication:** JWT (JSON Web Tokens)
 - **File Upload:** Multer
 - **Form Validation:** Express Validator
 - **Environment Variables:** Dotenv
-

Features

Authentication

1. **Login**
2. **Register**
3. **Password Reset** (Optional)

Blog Management

4. **Post Blog**
 - Title, Content, Thumbnail Image, Categories/Tags
5. **Blogs List Page**
 - Pagination
 - Search by Title, Author, or Tags
 - Filters (e.g., Most Recent, Most Popular)
6. **Blog Details**
 - Display Full Blog Content
 - Comments Section
 - Like/Bookmark Button
7. **Update Blog**
 - Edit Title, Content, Thumbnail, Categories/Tags
8. **Delete Blog**
9. **Categories/Tags for Blogs**
 - Organize blogs into categories or tags for better navigation

User Management

10. **User Profile**
 - Display User Information (Name, Email, Profile Picture)
 - Display User's Blogs
11. **Edit Profile**

- Update Name, Email, Profile Picture

12. Delete Profile

- Option to delete the user account and all associated blogs

Additional Features

13. Responsive Design

- Ensure the app works well on mobile, tablet, and desktop

14. Form Validation

- Validate inputs for login, register, blog creation, etc.

15. Image Upload

- Use **multer** for uploading blog thumbnails and profile pictures

16. Environment Variables

- Use **dotenv** to store sensitive data (e.g., JWT secret, database credentials)

17. Like/Bookmark Blogs

- Allow users to like or save blogs for later

Technology Stack

Frontend

- **EJS:** For server-side rendering of dynamic content.
- **Tailwind CSS:** For styling and responsive design.

Backend

- **Express:** For building the server and handling routes.
- **Node.js:** For running the backend logic.

Database

- **MongoDB:** For storing data (users, blogs, comments, etc.).
- **Mongoose:** For defining schemas and interacting with MongoDB.

Authentication

- **JWT (JSON Web Tokens):** For user authentication and protected routes.

Additional Tools

- **Multer:** For handling file uploads (profile pictures, blog thumbnails).
- **Express Validator:** For validating form inputs.
- **Dotenv:** For managing environment variables.

Learning Goals

Frontend (EJS + Tailwind CSS)

1. **Dynamic Rendering:** Using EJS to render dynamic content.

2. **Responsive Design:** Building a mobile-friendly UI with Tailwind CSS.
3. **Form Handling:** Creating and validating forms (login, register, blog post, etc.).
4. **File Uploads:** Handling image uploads for profile pictures and blog thumbnails.

Backend (Express + Node.js)

1. **Routing:** Creating RESTful routes for your app.
2. **Middleware:** Using middleware for authentication, validation, and error handling.
3. **Authentication:** Implementing JWT-based authentication (login, register, protected routes).
4. **CRUD Operations:** Performing Create, Read, Update, and Delete operations for blogs and users.
5. **Database Integration:** Using Mongoose to interact with MongoDB.

Database (MongoDB + Mongoose)

1. **Schema Design:** Creating schemas for users and blogs.
2. **Relationships:** Establishing relationships between collections (e.g., user and blogs).
3. **Querying:** Writing efficient queries to fetch and filter data.

Advanced Features

1. **Pagination:** Implementing pagination for the blogs list page.
2. **Search and Filtering:** Adding search and filter functionality for blogs.
3. **Comments:** Allowing users to comment on blogs.
4. **Likes/Bookmarks:** Implementing a like or bookmark feature for blogs.
5. **Environment Variables:** Using `dotenv` to manage sensitive data.

Project Roadmap

Phase 1: Frontend Development

1. Design the **Login Page**.
2. Design the **Register Page**.
3. Design the **Blogs List Page**.
4. Design the **Blog Details Page**.
5. Design the **Post Blog Page**.
6. Design the **Update Blog Page**.
7. Design the **User Profile Page**.
8. Design the **Edit Profile Page**.

Phase 2: Backend Development

1. Set up the Express server.
2. Implement authentication (login, register, JWT).
3. Create routes and controllers for blogs and users.
4. Set up MongoDB and Mongoose.
5. Define schemas for users and blogs.
6. Perform CRUD operations.

Phase 3: Advanced Features

1. Add pagination to the blogs list page.
 2. Implement search and filtering for blogs.
 3. Add a comments section to the blog details page.
 4. Implement a like or bookmark feature for blogs.
 5. Use `dotenv` to manage environment variables.
-