Church Website - Project Structure & Explanation

1. Project File Structure

/church-website

??? /public   # Publicly accessible files (CSS, JS, Images)

?   ??? /css   # Stylesheets (styles.css, admin.css)

?   ??? /js   # JavaScript files (script.js, admin.js)

?   ??? /images   # Site images

??? /views   # PHP files for each page

?   ??? [each website page e.g. activities.php, contact-us.php]

?   ??? 404.php   # Page not found

??? /includes   # Reusable components (header, footer, navbar, config)

??? /routes   # Routing system (router.php)

??? /admin   # Admin dashboard (login.php, manage-content.php)

??? /controllers   # Handles backend logic

??? /models   # Database interactions (Content.php, User.php)

??? index.php   # Main entry point (handles routing)

??? .htaccess   # URL rewriting (clean URLs)

??? config.php   # Database settings

2. Explanation of Each File

index.php: Handles all routing requests and dynamically loads pages.

config.php: Contains database configuration settings.

.htaccess: Rewrites URLs to clean format (hides .php extensions).

/views/: Contains PHP files for each website page.

/includes/header.php: Contains reusable HTML header code.

/includes/footer.php: Contains reusable HTML footer code.

/includes/navbar.php: Contains website navigation bar.

/admin/: Contains files for admin login and content management.

/controllers/: Handles business logic such as user authentication.

/models/: Contains database models for fetching and updating content.

3. Summary of Data Flow

1. When a user visits a page, the request is handled by index.php.

2. index.php checks the requested page and routes it to the correct file in /views.

3. If the page exists, it loads the header, navbar, and footer.

4. If the page does not exist, a 404 error page is shown.

5. If the page requires database content, it fetches data using the models.

6. Admins can log in via /admin and manage content dynamically.

7. The website uses .htaccess to remove .php extensions and clean URLs.