TheCodingCamp

Website Documentation

Version 1.0 • 2025

1. Project Overview

TheCodingCamp is a full-stack blogging platform built to allow an admin to create, manage, and publish blog posts, while visitors can browse, read, and get in touch through a clean public-facing interface.

The website consists of six main pages — three are publicly accessible to all visitors, and three are restricted to the site admin. All pages share the same consistent header and footer for a unified user experience.

Public Pages

- Home Page — lists all published blog posts
- About Page — background and info about the blog author
- Contact Page — form for visitors to send a message to the admin

Admin-Only Pages

- Login Page — secure entry point to the admin area
- Admin Dashboard — manage posts, files, and settings
- Add New Post — form for creating and publishing new blog entries

2. Technology Stack

Here is a breakdown of the technologies used across the different layers of the application:

- **Frontend:** Bootstrap — a popular CSS framework used to build a responsive, mobile-friendly UI without writing custom CSS from scratch.
- **Backend:** Python Flask — a lightweight web framework used to handle routing, page rendering, form submissions, and authentication logic.
- **Database:** MySQL — a relational database used to store all blog posts, user credentials, and other structured data.
- **ORM / DB Connection:** SQLAlchemy (via SQLALCHEMY_DATABASE_URI) — connects the Flask app to MySQL, allowing data to be read and written using Python objects rather than raw SQL.
- **Data Exchange:** JSON — used for structured data storage and exchange between different parts of the application.

3. Page-by-Page Breakdown

3.1 Home Page

The Home Page is the first thing a visitor sees when they land on the website. At the top, there is a full-width background image with the website title 'Coding Camp' displayed prominently over it. This creates a strong visual first impression.

The header (navigation bar) sits at the very top of the page and is consistent across the entire website. On the left side of the nav bar, the website name 'TheCodingCamp' is displayed as a brand link. On the right side, there are links to the three public pages: Home, About, and Contact.

As you scroll down past the banner, you will see a list of all published blog posts displayed in card or link format. Each blog title is clickable and takes the visitor to the full blog post page where they can read the complete article.

At the bottom of the post listing section, there are two pagination buttons: a 'Previous Posts' button on the left and a 'Next Posts' button on the right. These allow visitors to navigate between pages of blog posts, which is useful as the number of posts grows over time.

At the very bottom of the page is the footer, which is shared across all pages. It contains links to the admin's social media profiles.

> 📝 *All blog posts visible on the Home Page are fully functional — clicking any post title opens the full article.*

3.2 About Page

The About Page can be accessed by clicking the 'About' link in the navigation bar. Like all pages, it shares the same header and footer.

At the top of the page is a background image with the text 'About Me' overlaid on it. Scrolling down reveals a section containing detailed information about the blog's admin — their background, interests, and any other relevant personal or professional information.

This content is static, meaning it is hard-coded into the page and cannot be changed from the admin dashboard. Any updates to the About page need to be made directly in the source code or template files by the admin or developer.

> 📝 *If you want the About page content to be editable from the dashboard in a future version, you can consider storing it in the database like blog posts.*

3.3 Contact Page

The Contact Page provides a way for visitors to reach out to the admin directly from the website. It follows the same layout pattern as other pages — background image with 'Contact Me' displayed at the top, then the main content below.

The contact form includes the following fields that visitors need to fill in:

- Name
- Email Address
- Phone Number
- Message

Once all fields are filled in, the visitor clicks the 'Send' button located at the bottom right of the form. This submits the form and delivers the message directly to the admin's email address.

> 📝 *Make sure your Flask backend is configured with correct SMTP email settings so that form submissions are actually delivered to the admin's inbox.*

3.4 Login Page

The Login Page acts as a security gate for the admin area. Whenever someone tries to access the Admin Dashboard directly — whether by typing the URL or clicking an admin link — they are automatically redirected to the Login Page first.

The page presents a simple login form with two fields: Username and Password. If both credentials are entered correctly, the user is granted access and redirected to the Admin Dashboard. If either credential is wrong, the page reloads with an error message prompting the user to check their username or password and try again.

> 📝 *It is strongly recommended to use a strong, unique password for admin access and to store credentials securely using hashing (e.g., bcrypt) in the database rather than plain text.*

3.5 Admin Dashboard

The Admin Dashboard is the control center of the website. It is only accessible after a successful login and provides the admin with all the tools needed to manage the site's content.

The dashboard includes the following features and sections:

- **Logout Button:** Allows the admin to safely end their session and return to the public site.
- **Add New Post Button:** A shortcut button that navigates the admin to the Add New Post page.
- **File Uploader:** A section that allows the admin to upload any file (such as images or documents) and save it to the server.
- **Published Posts List:** A complete list of all blog posts that have been published. Each post in the list has two action buttons: Edit (to open the post in an editable form) and Delete (to permanently remove the post from the site).

When the admin clicks the Edit button next to any post, they are taken to a dedicated edit page where the existing content of that post is pre-loaded into the form fields for easy editing and saving.

3.6 Add New Post Page

This page is where new blog content is created. It is accessed by clicking the 'Add New Post' button on the Admin Dashboard.

The form on this page includes the following fields that the admin needs to fill in before publishing:

- **Blog Title:** The main heading of the blog post.
- **Blog Subtitle:** A secondary headline that provides a short description or teaser of the post.
- **Blog Slug:** A URL-friendly version of the title used in the post's web address (e.g., /blog/my-first-post). It should be lowercase and use hyphens instead of spaces.
- **Background Image:** An image uploaded to be used as the banner or header image of the blog post.
- **Posted By:** The author's name or username to be displayed on the post.
- **Blog Content:** The full body text or content of the blog post.

Once all fields are filled in, clicking the 'Post' button publishes the blog entry, making it immediately visible on the Home Page for visitors to read.

At the top of the Add New Post page, there are also two navigation buttons: 'Back to Dashboard' to return to the admin panel, and 'Logout' to end the current session.

> 📝 *Blog slugs must be unique for each post. If two posts share the same slug, routing errors or content conflicts may occur.*

4. Shared Components

4.1 Header (Navigation Bar)

The header is displayed at the top of every page across the entire website. It contains the website brand name 'TheCodingCamp' on the left side, which typically links back to the Home Page. On the right side, the navigation links to the three public pages are listed: Home, About, and Contact.

The admin-related pages (Login, Dashboard, Add Post) do not appear in the public navigation, keeping the interface clean for regular visitors.

4.2 Footer

The footer appears at the bottom of every page. It contains links to the admin's social media accounts, giving visitors an easy way to connect or follow the author on other platforms.

5. User Roles

There are two types of users for this website:

- **Visitor (Public User):** Can view all published blog posts, read individual articles, browse the About page, and send messages via the Contact page. No login is required.
- **Admin:** Has full control over the website's content. After logging in, the admin can publish new posts, edit or delete existing posts, upload files, and manage the site through the dashboard.

6. Developer Notes & Recommendations

- Ensure your MySQL database is running and properly connected via the SQLALCHEMY_DATABASE_URI environment variable before starting the Flask application.
- Admin passwords should be stored as hashed values in the database using a library like bcrypt or Werkzeug's security module — never store plain-text passwords.
- The Contact form requires email configuration in Flask (e.g., using Flask-Mail or an SMTP server) to actually deliver messages to the admin.
- Blog slugs should be auto-generated from the blog title on the Add New Post form to avoid human error and duplicate slug issues.
- The About Page content is currently static. Consider moving it to the database for easier updates without touching the code.

- Make sure uploaded files are validated for type and size on the server side to prevent security issues.

End of Documentation