WDD PACE University of Winnipeg

2022-May-3

Lihang Yao

Proposal for

Gardener Website Upgrade

Table of Content

1. Introduction

Gardener website is a local platform for people to interact with each other. Now the owner of website decides to upgrade the website to a dynamic changing personal blog website instead of a static website for exhibition purpose.

To meet the requirements to upgrade the current website, our company will take the responsibility to apply a new technique on this project.

We will use PHP as the primary programming language, HTML to construct the fundamental infrastructure, CSS to present a gorgeous style and JavaScript to enhance the functionality of the current website. In addition, we will use MySQL as the database for this project.

We will use MVC to dispatch every user request, by this way, we can make the program more organized and easy to maintain.

2. Features and Functions

2.1 For normal users

- normal user can login and logout
- normal user can register an account
- normal user without login can review all posts
- normal user with login can leave a comment on posts
- normal user can view all comments they left on the website
- normal user can delete their comments
- normal user can view other user's comments
- normal user can search for specific posts
- normal user can view posts by categories

2.2 For Admin

- Admin can do anything as a login normal user
- Only Admin can login to the dashboard of the administration system
- Admin can add a new post
- Admin can edit existing posts
- Admin can delete any posts
- Admin can view lists of current posts, users, comments, and categories

3. Database

There are five tables in the database. Every table will have a unique id field as primary key to target an individual row. Each table have time related fields to record the time of creation and update. Tables with a ‘deleted’ field will determine whether this record is deleted.

- users table

  Users table will collect user information on first name, last name, mailing information, phone, email, and users must create a password for login purpose.

- posts table

  Posts table contains the title, author-id, summary, content, image, category, status of post and privilege to allow comment.

- comments table

  Comments table contains the information of the comments. There will be post-id field relating to the post where this record of comment resides.

- categories table

  Categories table contains names of five categories of posts.

- log table

  Log table stores the records of all users’ entries.

4. Use Cases

- For unauthenticated users

  Unauthenticated users can view all posts and comments that other users left on the website. But if they want to leave a comment on a post, they must register an account first. On register page, they must input all required fields and click submit, then it will bring the user to profile page if registered successfully.

- For authenticate users

  Authenticate users could browse all posts on the website after login. And they can leave a comment on any allowed-comment posts. If they enter their individual profile, they will find all comments they left previously.

- For Admin

  Admin can do anything what an authenticate user can do on the website. Besides, Admin can login the dashboard to access the administration system. They can view list all posts, comments, and user information. And they can create, update, and delete any posts. On user page, Admin can delete or update any user's information. On comments page, Admin can delete any comments if possible.

5. Security

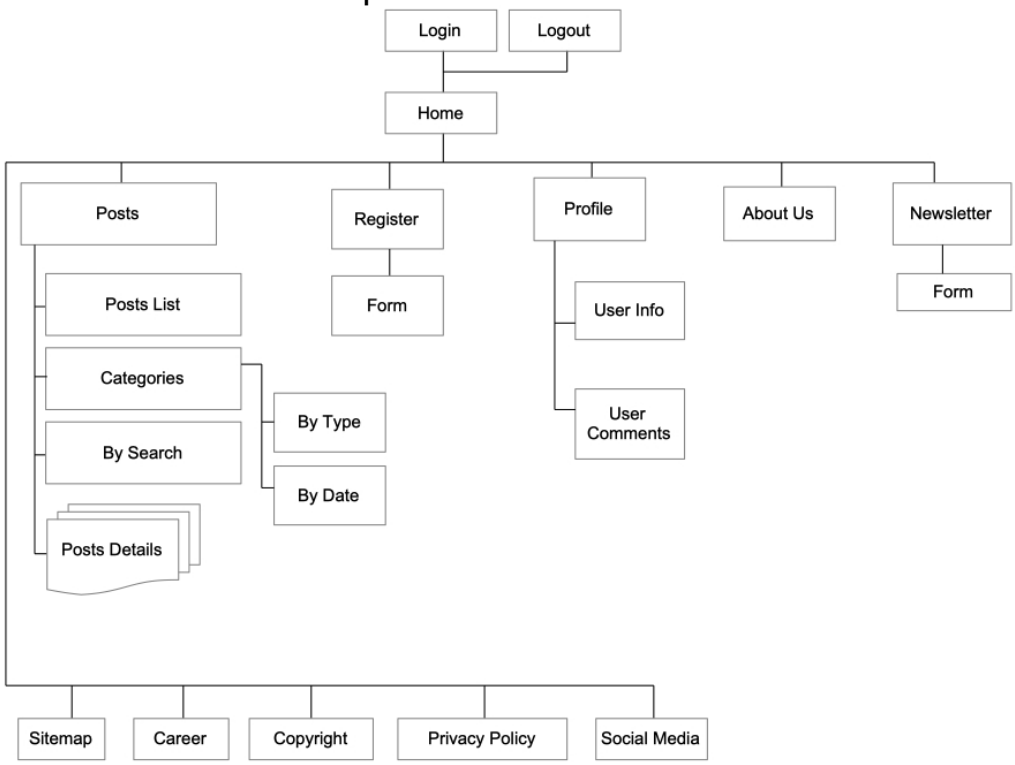
- To prevent XSS attacks, we will use PHP built-in function htmlentities() to escape any output to users' web browser. Besides, we will validate any value of form field from user's input.

- To prevent SQL injection attacks, we will use parameters to bind the value with the placeholders. By this way, we can prevent any malicious code will be injected into our databases.

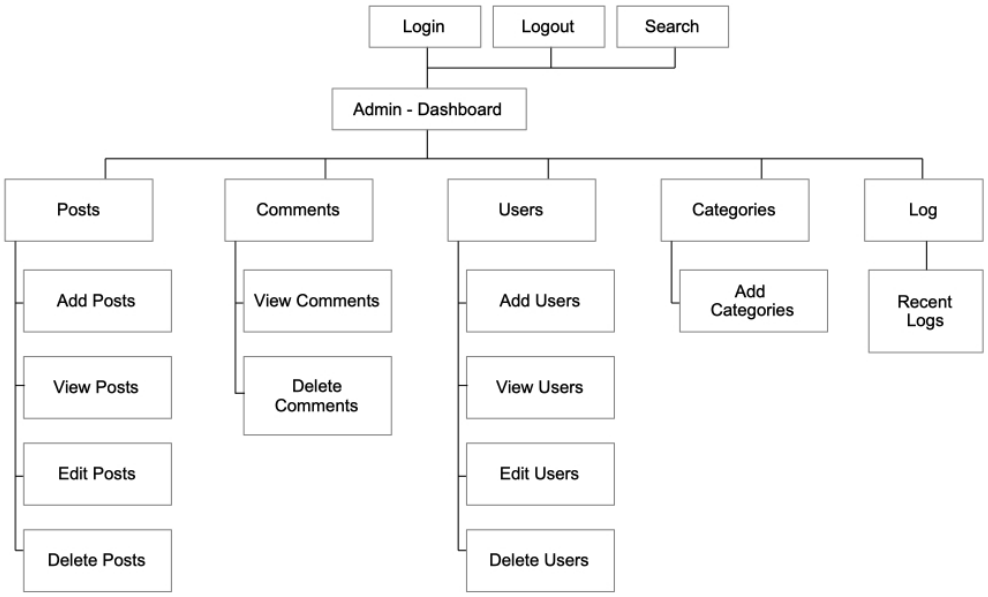
- To prevent CSRF attacks, we will use csrf token to insert into a hidden field in every form. We will validate the csrf token every time on a POST request method. If the csrf token doesn't match with the one on our server, we will end this session immediately.

6. Sitemap

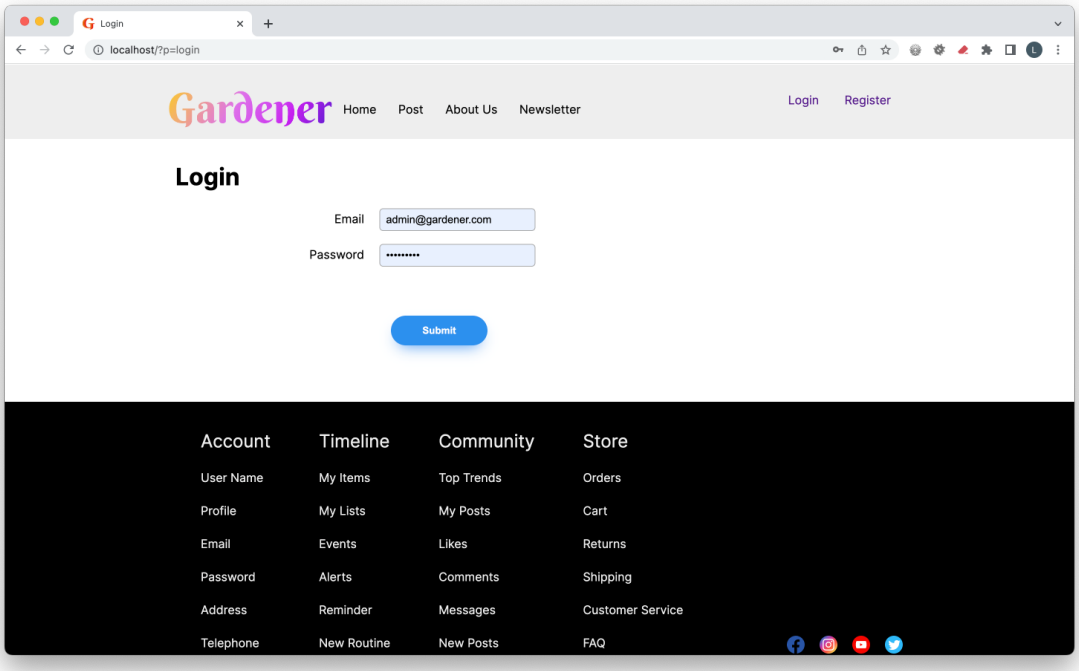
- Front-end Sitemap

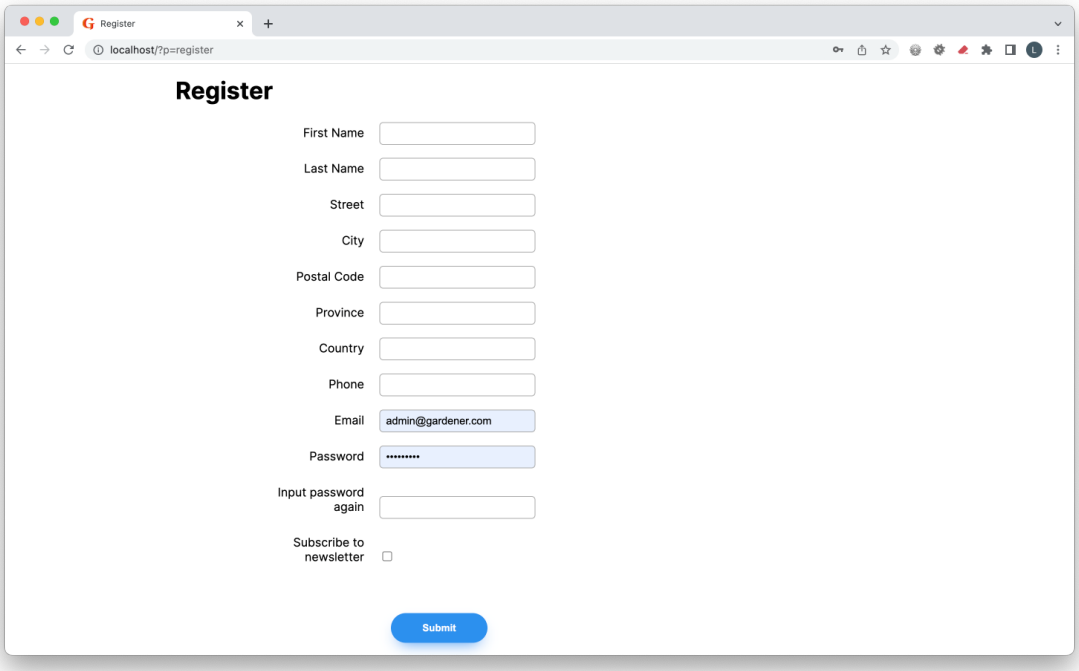


- Back-end Sitemap

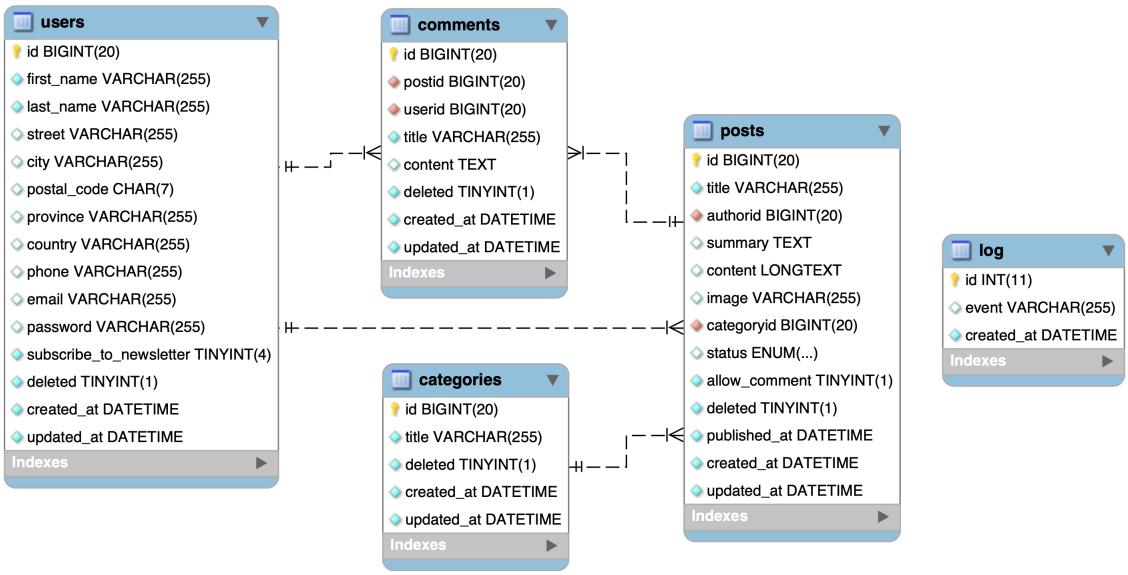


7. Login and Register





8. Physical data model



9. Timetable

Stage	Date
Kickoff meeting	May 2
Initial Wireframe	May 3
Design Mockup	May 7
Copywrite	May 7
Programming	May 31
Testing	June 2
Website Launch	June 15

10. Site bible

- **Hosting Server**

  AWS EC2

- **PHP Version**

  PHP 8.1.5

- **Database**

  MySQL 10.4.21

- **IP of Service**

  52.60.51.49

- **URL of Project**

  http://capstone.leon-web-dev.com

- **Admin User Login**

  Email: admin@gardener.com

  Password: P@ssw0rd?

- **Sample Normal User Login**

  Email: chris@gardener.com

  Password: P@ssw0rd?