# **Project Documentation**

## **Introduction**

This documentation provides a comprehensive overview of the Laravel application developed as part of a collage assignment. The application is a dynamic responsive website built using Laravel's MVC framework and incorporates various features such as user authentication, role-based access control, CRUD operations, server-side validation, and graphical representation of data.

This project is hosted at the following URL: [Insert Project URL]

The source code is publicly available on Github: [Insert GitHub URL]

The documentation covers all the requirements of the assignment in detail and guides you on how to navigate through the different functionalities of the application.

## **Seeded Users**

The following users have been seeded into the database:

1. Test User 1
   * Username: testuser1@example.com
   * Password: password
   * Role: User
2. Test User 2
   * Username: testuser2@example.com
   * Password: password
   * Role: User
3. Admin User
   * Username: adminuser@example.com
   * Password: password
   * Role: Admin

## **Routing**

All the routes for this Laravel application are defined in the web.php file located in the routes directory. The application uses both anonymous functions and controller methods as route handlers.

The homepage is served by the default Laravel welcome view. User authentication routes have been automatically generated using the Auth::routes() method.

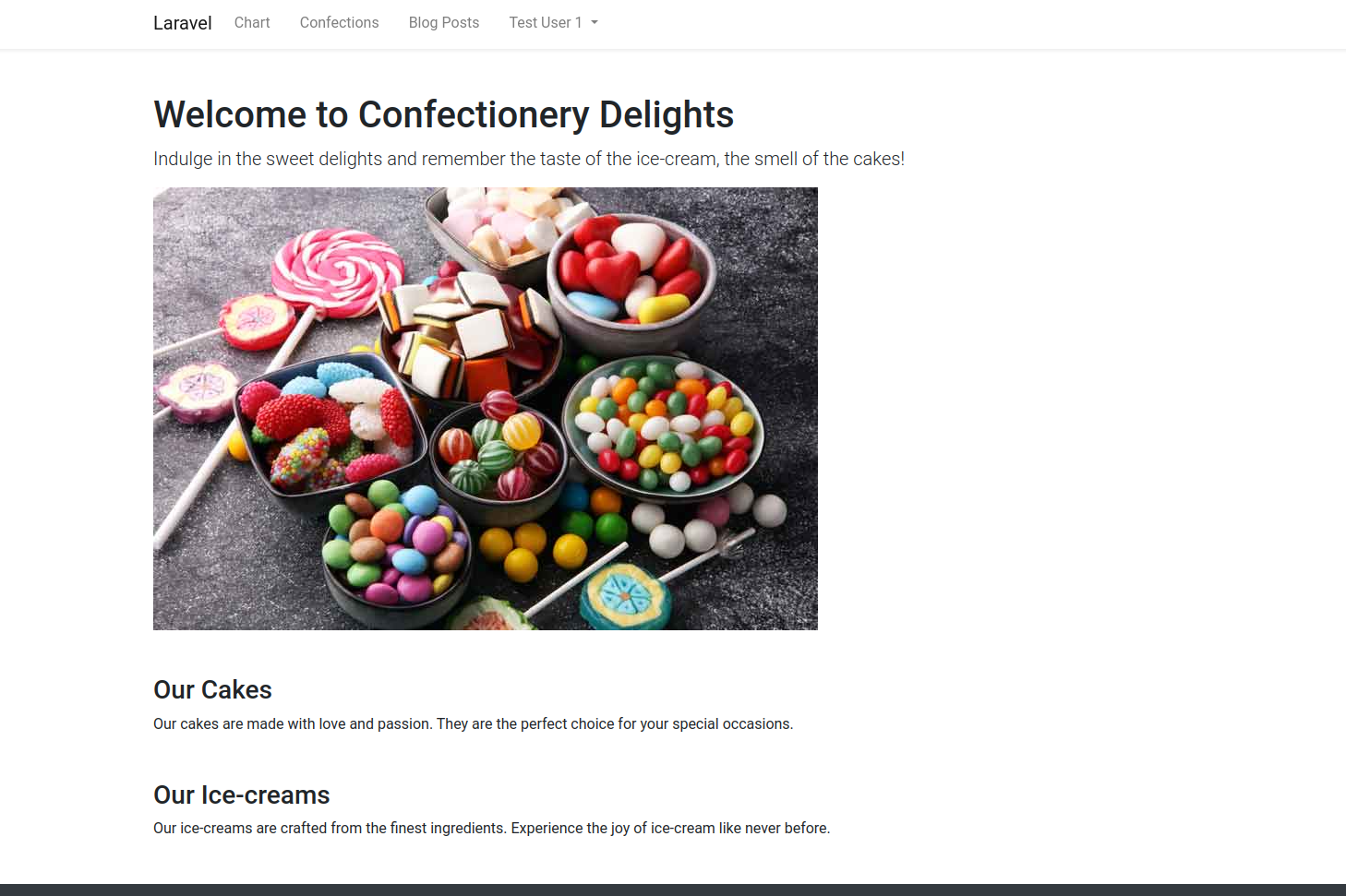
For handling blog posts, the application uses a resource route that automatically creates routes for standard CRUD operations. Similarly, a resource route is also used for user management by admins.

There are also two additional routes for managing confections - one for listing all confections, and another for filtering them based on certain criteria.

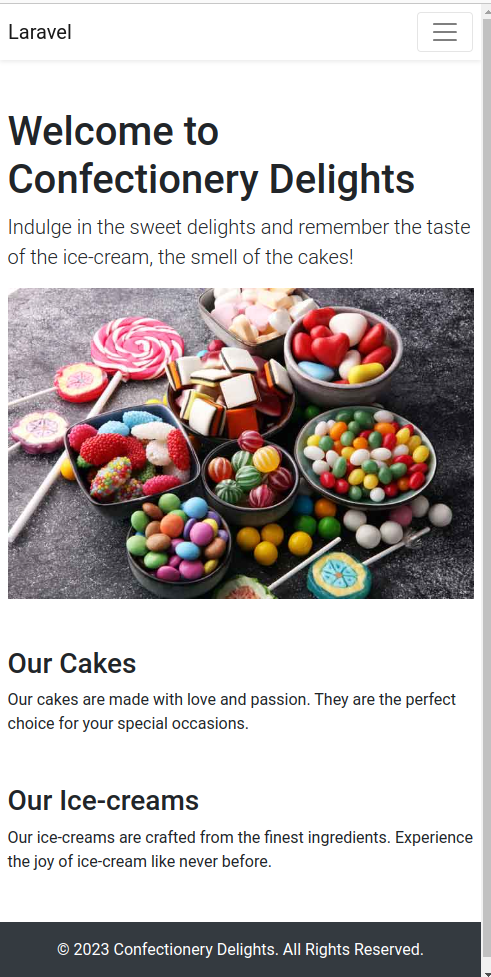
## **User Interface**

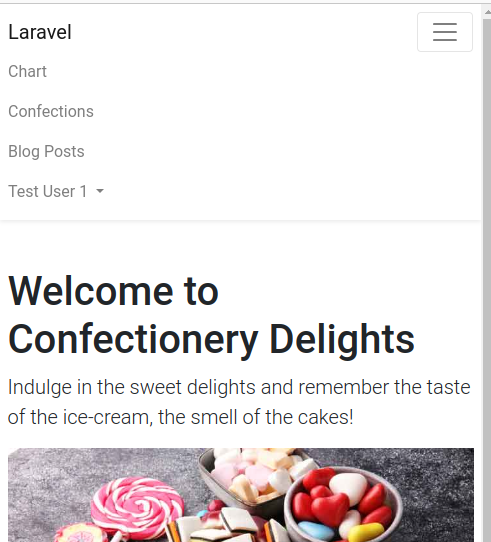
The user interface of the application is constructed using a combination of Laravel's Blade templating engine and Bootstrap's front-end framework for responsive design.

**Desktop home page:**

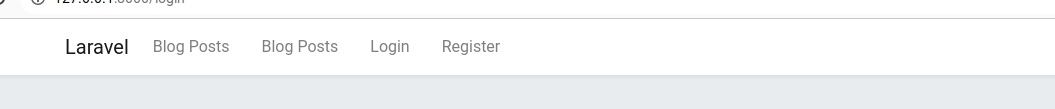


**Mobile home page:**

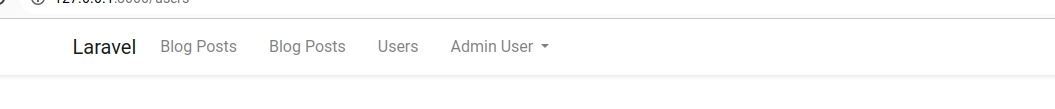
****

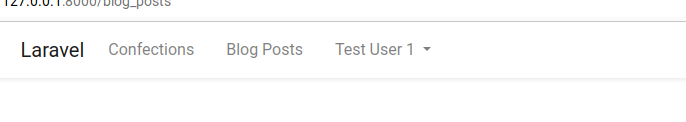
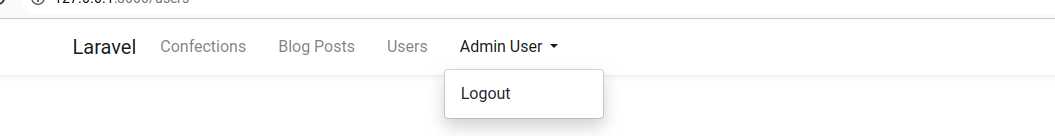
**Mobile home page (navigation open):  
**

The navbar at the top contains links to different sections of the website. Depending on the user's role (guest, logged-in user, or admin), some menu items are selectively shown or hidden. The 'Users' menu item, for example, is only visible to users with the 'manage-users' permission (typically, admins).



When logged in, users can see their name on the top-right corner of the navbar. Clicking on their name reveals a dropdown with the 'Logout' option.



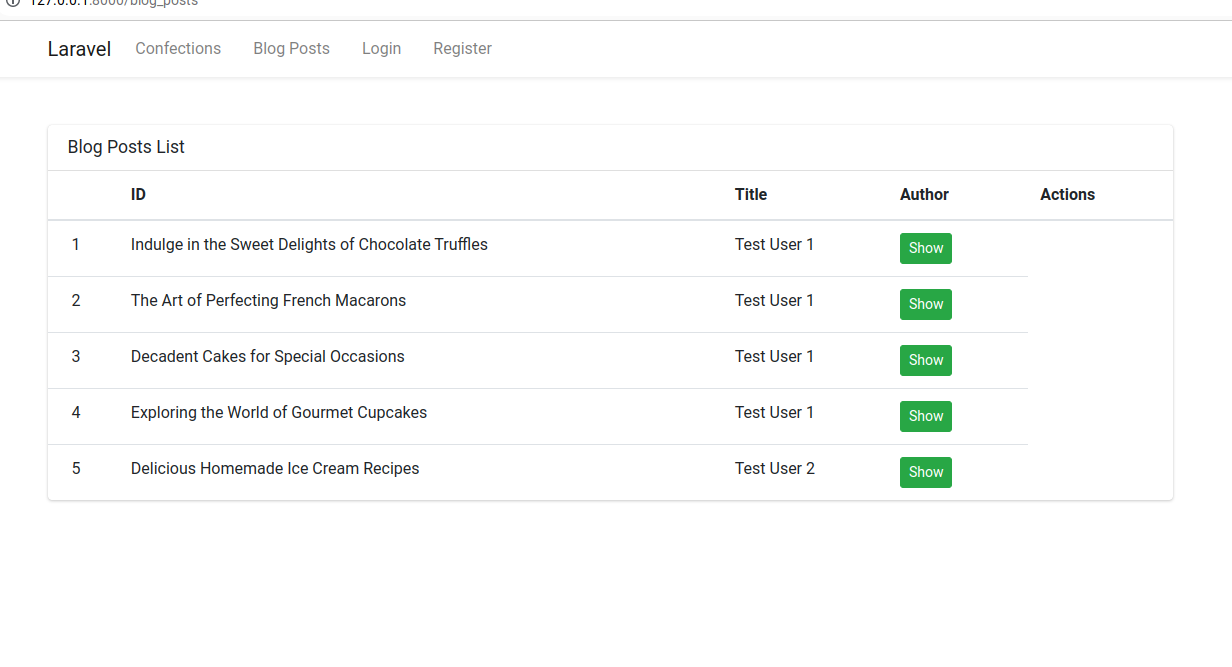


## **CRUD Applications**

The application is using the AdminLTE css library.

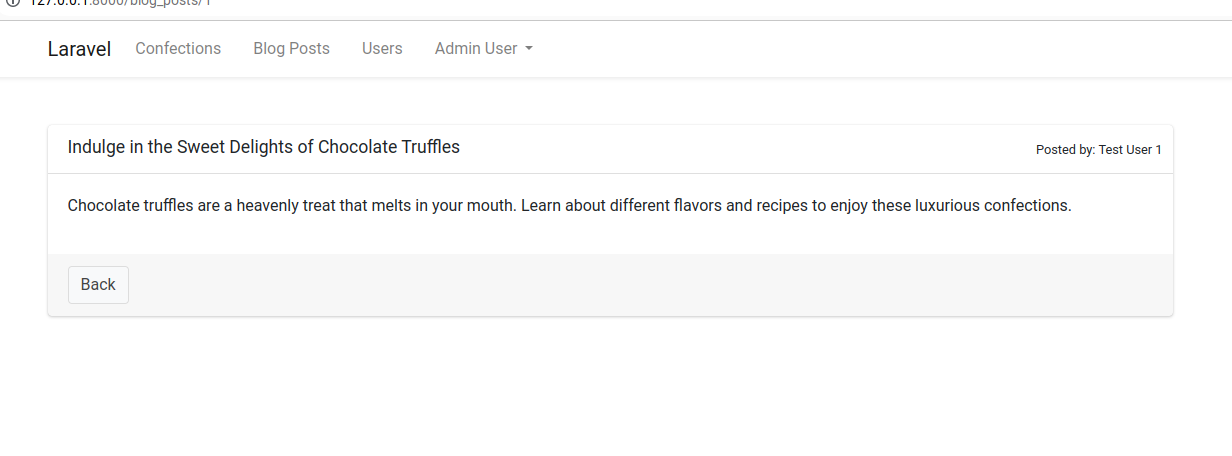
### **Blog Posts**

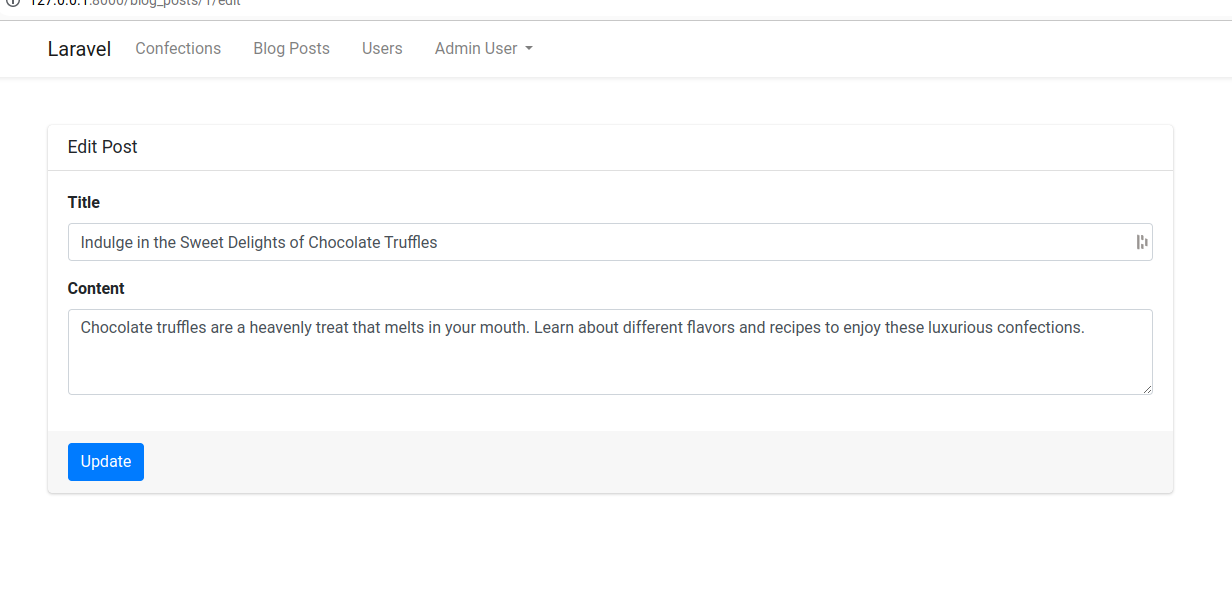
Users can view all blog posts on the site. Logged-in users can modify and delete only their own contents, while the admin can modify and delete all contents.

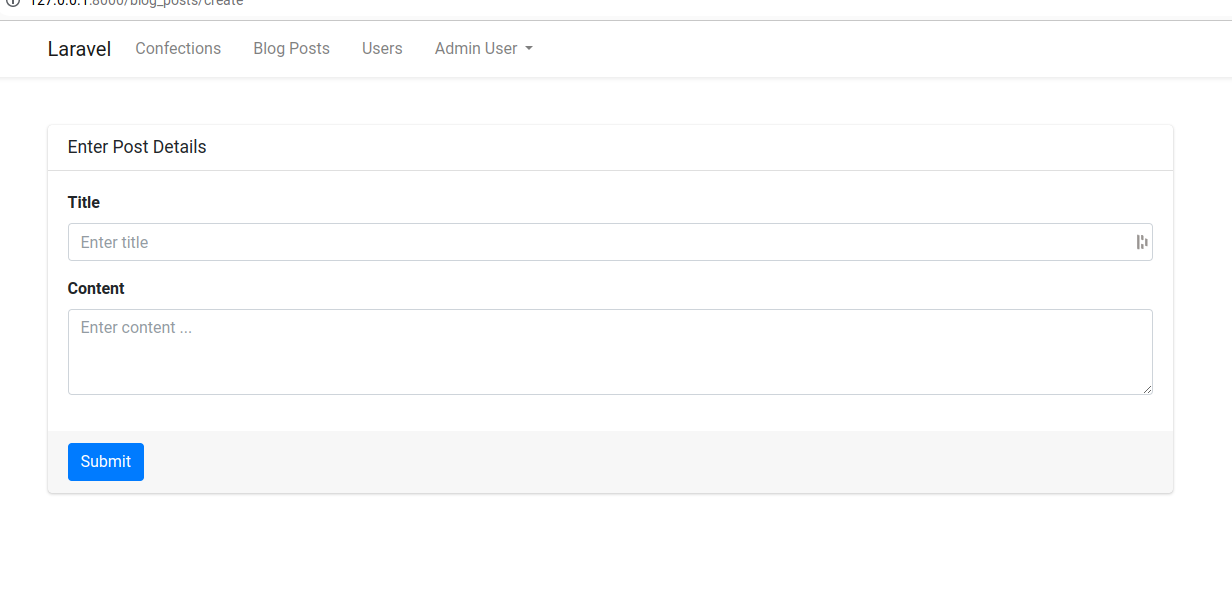


### 

### 

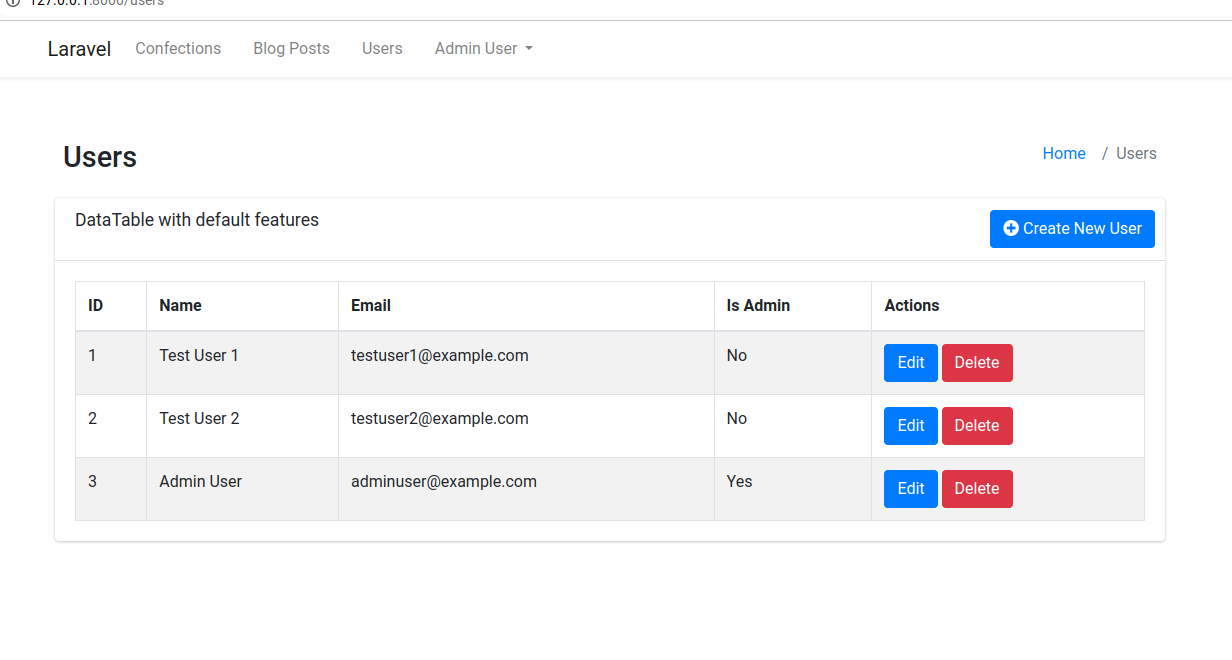
Any user can view any blog post:  
  


Users can edit their own blog posts and admins can edit any blog post:  
  


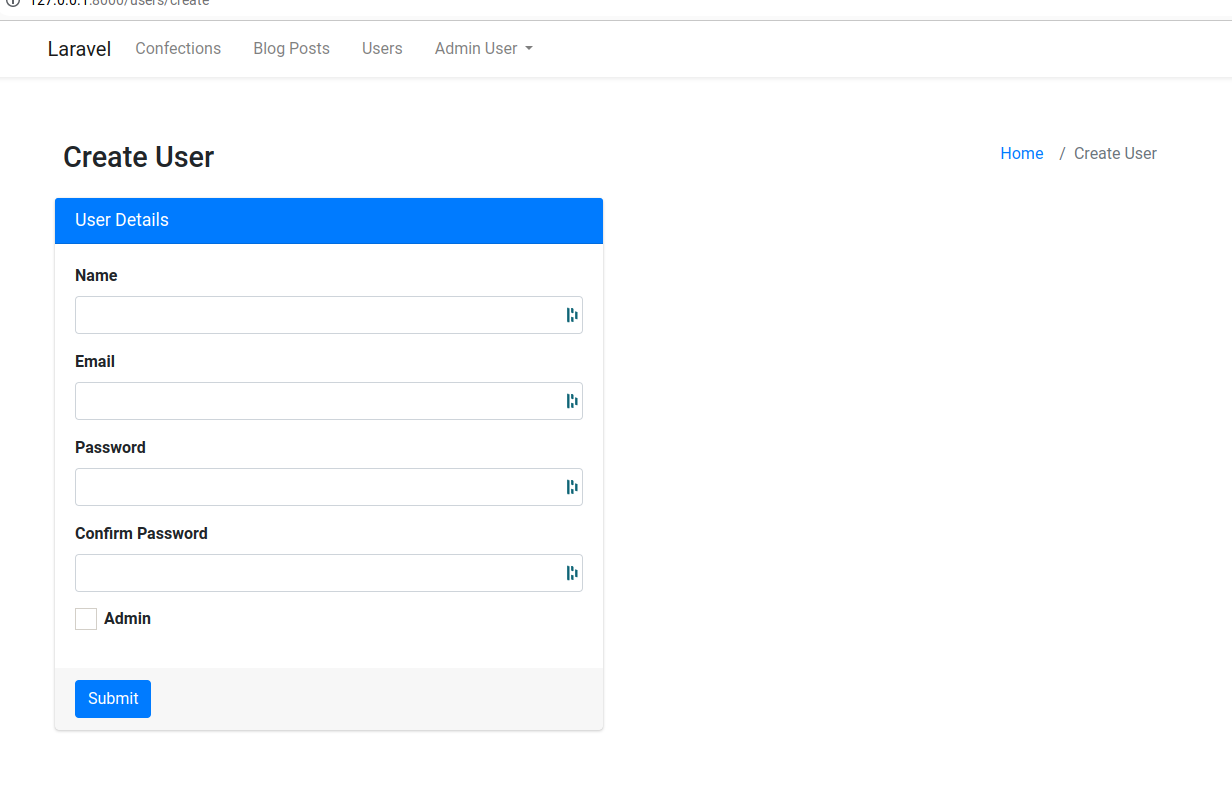
Signed in users can create new blog posts by clicking on the New Post button:  


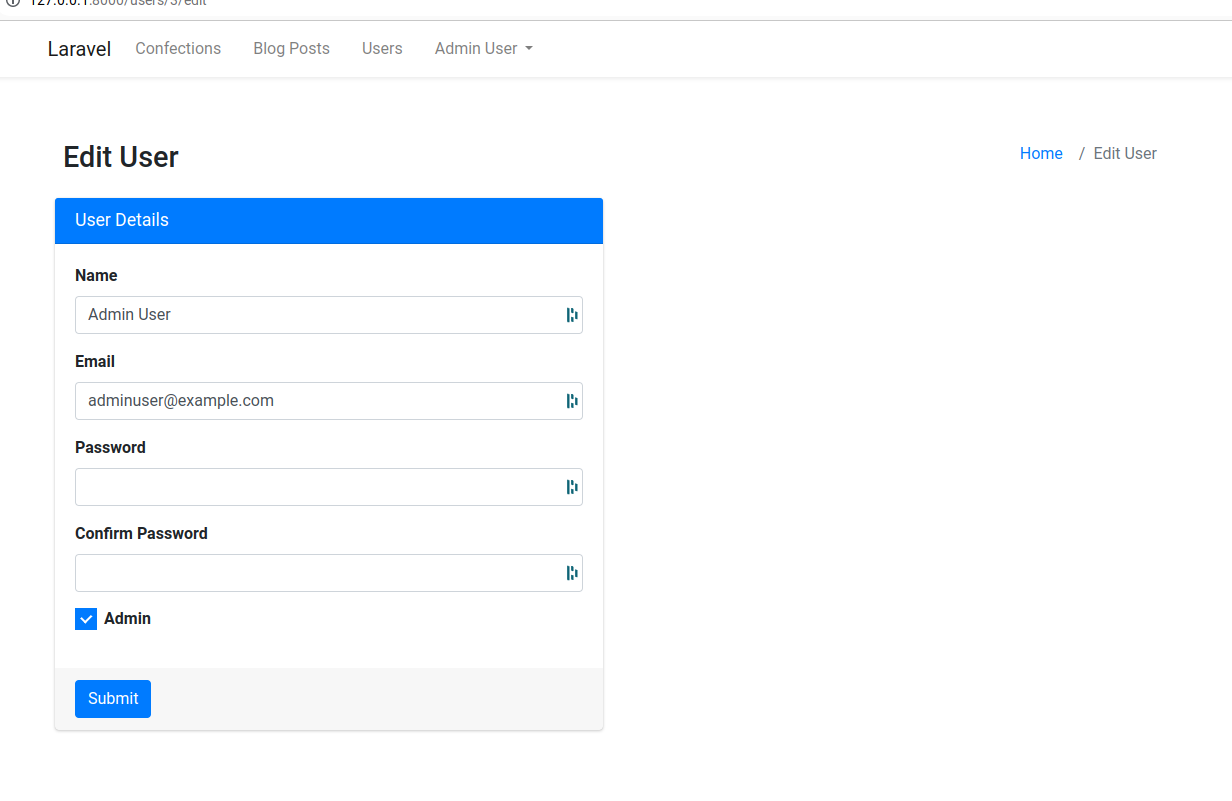
Users can delete their blog posts, admins can delete any blog posts.

### **User Management**

The admin can manage all the users with another CRUD application accessible via the 'Users' link in the navbar.  
  


New users can be created by pressing the Create New User button:

  
  
An admin user can be created by checking the Admin checkbox.   
  
The user data can be edited later:



The admin role can be turned on and off anytime for any user.  
  
Users can be deleted on the listing page.

## **Server-side Validation and Data Manipulation**

The application includes a form where users can retrieve data from the database or add new data to one of the tables. Server-side validation is implemented to ensure data integrity.