Develop a Simple Laravel CRUD Application | December 3, 2024 | 11:47:51 PM

Objective

To develop a Laravel application that allows users to manage a list of books. The application must implement the following functionalities:

- Search
- Create
- Edit
- Update
- Delete

Application Description

You will create a Book Management System where users can perform CRUD operations on books. Each book will have the following attributes:

- Title (string)
- Author (string)
- Description (text)
- Published Year (integer)
- Genre (string)

Step-by-Step Instructions

Setup the Laravel Project
1. Install a new Laravel project:
bash
```
laravel new book-management
cd book-management
```

2. Set up the database connection in the .env file.

Create the Database and Model
1. Create a migration and model for the Book entity:
bash
```
php artisan make:model Book -m
Php artisan migrate
```

Open VS Code

```
code .
```

2. Define the database schema in the migration file located in database/migrations/:
Look for the migration file

```
Schema::create('books', function (Blueprint $table) {
```

```
            $table->id();
            $table->string('title');
            $table->string('author');
            $table->text('description');
            $table->integer('published_year');
            $table->string('genre');
            $table->timestamps();
        });
```

3. Run the migration:

```
    php artisan migrate
```

Create the Controller and Routes

1. Generate a controller for books:

bash

```
   php artisan make:controller BookConroller –resource
```

2. Define the routes in routes/web.php:

```
  <?php

  use Illuminate\Support\Facades\Route;
  use App\Http\Controllers\BookController;

  Route::get('/', [BookController::class, 'index'])->name('books.index');
  Route::resource('books', BookController::class);
```

Implement CRUD Functionalities

a. Create and Store Books

1. In the create method of BookController, return a view for adding a new book:

```
   public function create()
    {
        return view('books.create');
    }
```

2. In the store method, validate and save the book data:

Import Book model

```
   use App\Models\Book;
```

3. Create a modal resources/views/books/index.blade.php with a form to add a book.

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Search Books</title>
    <link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.min.css" rel="stylesheet">
```

```
</head>

<body class="bg-gray-200 flex flex-col items-center min-h-screen p-6">
    <!-- Search Form -->
    <form method="GET" action="{{ route('books.index') }}"
        class="flex space-x-2 p-4 bg-gray-100 rounded-md shadow-md mb-8 w-full max-w-lg">
        <input type="text" name="query" placeholder="Search books..."
            class="flex-1 border border-gray-300 rounded-md px-3 py-2 outline-none focus:border-blue-500
focus:ring-1 focus:ring-blue-500 transition duration-200">
        <button type="submit"
                class="px-4 py-2 bg-blue-500 text-white rounded-md hover:bg-blue-600 focus:outline-none
focus:ring-2 focus:ring-blue-500 transition duration-200">
            Search
        </button>
    </form>

    <!-- Button to Open Modal -->
    <button onclick="document.getElementById('bookModal').classList.remove('hidden')"
            class="px-4 py-2 bg-blue-500 text-white rounded-md hover:bg-blue-600 focus:outline-none
focus:ring-2 focus:ring-blue-500 transition duration-200">
        Add New Book
    </button>

    <!-- Book List -->
    <div class="w-full max-w-4xl mx-auto mt-6">
        <h2 class="text-3xl font-bold mb-6 text-center">Book List</h2>
        @foreach ($books as $book)
            <div class="space-y-4 mb-2">
                <div
                        class="bg-white p-6 rounded-lg shadow-lg border border-gray-200 hover:shadow-xl
transition-shadow duration-300">
                    <div class="flex justify-between items-center mb-2">
                        <h3 class="text-2xl font-semibold text-gray-800">{{ $book->title }}</h3>
                        <div class="flex space-x-2">
                            <!-- Edit Button -->
                            <a href="{{ route('books.edit', $book) }}"
                                    class="px-4 py-2 bg-yellow-500 text-white rounded-md hover:bg-yellow-600
focus:outline-none focus:ring-2 focus:ring-yellow-500 transition duration-200">
                                Edit
                            </a>
                        </div>
                    </div>
                    <p class="text-gray-700 mb-2">{{ $book->description }}</p>
                    <div class="flex justify-between items-center">
                        <span class="text-gray-600 text-sm">Published:
                            <strong>{{ $book->published_year }}</strong></span>
                                <span class="text-gray-600 text-sm">Genre: <strong>{{ $book->genre
}}</strong></span>
                    </div>
                </div>
            </div>
        @endforeach
    </div>

    <!-- Modal -->
        <div id="bookModal" class="fixed inset-0 bg-black bg-opacity-50 flex justify-center items-center
hidden">
```

```
    <div class="bg-white p-6 rounded-lg shadow-md w-full max-w-lg">
        <h1 class="text-2xl font-bold mb-4">Add New Book</h1>
        <form method="POST" action="{{ route('books.store') }}">
            @csrf
            <div class="mb-4">
                <label for="title" class="block text-gray-700 font-semibold mb-2">Title</label>
                <input type="text" id="title" name="title" placeholder="Enter book title"
                    class="w-full border border-gray-300 rounded-md px-3 py-2 outline-none focus:border-blue-500 focus:ring-1 focus:ring-blue-500 transition duration-200"
                    required>
            </div>
            <div class="mb-4">
                <label for="author" class="block text-gray-700 font-semibold mb-2">Author</label>
                <input type="text" id="author" name="author" placeholder="Enter author's name"
                    class="w-full border border-gray-300 rounded-md px-3 py-2 outline-none focus:border-blue-500 focus:ring-1 focus:ring-blue-500 transition duration-200"
                    required>
            </div>
            <div class="mb-4">
                <label for="description" class="block text-gray-700 font-semibold mb-2">Description</label>
                <textarea type="text" id="author" name="description" placeholder="Enter book's description"
                    class="w-full border border-gray-300 rounded-md px-3 py-2 outline-none focus:border-blue-500 focus:ring-1 focus:ring-blue-500 transition duration-200"
                    required></textarea>
            </div>
            <div class="mb-4">
                <label for="published_year" class="block text-gray-700 font-semibold mb-2">Published year</label>
                <input type="number" id="author" name="published_year"
                    placeholder="Enter book's publish year"
                    class="w-full border border-gray-300 rounded-md px-3 py-2 outline-none focus:border-blue-500 focus:ring-1 focus:ring-blue-500 transition duration-200"
                    required>
            </div>
            <div class="mb-4">
                <label for="genre" class="block text-gray-700 font-semibold mb-2">Genre</label>
                <input type="text" id="author" name="genre" placeholder="Enter book's genre"
                    class="w-full border border-gray-300 rounded-md px-3 py-2 outline-none focus:border-blue-500 focus:ring-1 focus:ring-blue-500 transition duration-200"
                    required>
            </div>
            <div class="flex justify-end space-x-2">
                <button type="button" onclick="document.getElementById('bookModal').classList.add('hidden')"
                    class="px-4 py-2 bg-gray-500 text-white rounded-md hover:bg-gray-600 focus:outline-none focus:ring-2 focus:ring-gray-500 transition duration-200">
                    Cancel
                </button>
                <button type="submit"
                    class="px-4 py-2 bg-blue-500 text-white rounded-md hover:bg-blue-600 focus:outline-none focus:ring-2 focus:ring-blue-500 transition duration-200">
                    Add Book
                </button>
            </div>
        </form>
```

```
        </div>
    </div>
</body>

</html>
```

b. Read and List Books

1. In the index method of BookController, retrieve all books and return a view:

```
    public function index(Request $request)
    {
        $query = $request->input('query');

        $books = Book::when($query, function ($q) use ($query) {
            return $q->where('title', 'like', "%$query%")
                     ->orWhere('author', 'like', "%$query%");
        })->get();

        return view('books.index', compact('books'));
    }
```

2. Create the view resources/views/books/index.blade.php to display the list of books.

c. Edit and Update Books

1. In the edit method, retrieve the book data and return a view:

```
     public function edit(Book $book)
    {
        return view('books.edit', [
            'book' => $book
        ]);
    }
```

2. In the update method, validate and update the book data:

```
    public function update(Request $request, Book $book)
    {
        $validated = $request->validate([
            'title' => 'required|string|max:255',
            'author' => 'required|string|max:255',
            'description' => 'required',
            'published_year' => 'required|integer',
            'genre' => 'required|string|max:255',
        ]);

        $book->update($validated);

        return redirect()->route('books.index')->with('success', 'Book updated successfully!');
    }
```

3. Create the view resources/views/books/edit.blade.php with a form to edit a book.

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
    <title>Search Books</title>
    <link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.min.css" rel="stylesheet">
</head>

<body class="bg-gray-200 flex flex-col items-center min-h-screen p-6">
    <div class="fixed inset-0 bg-black bg-opacity-50 flex justify-center items-center">
        <div class="bg-white rounded-lg shadow-lg p-6 w-full max-w-md">
            <h2 class="text-2xl font-bold mb-4">Edit Book</h2>
            <form method="POST" action="{{ route('books.update', $book->id) }}">
                @csrf
                @method('PUT')
                <div class="mb-4">
                    <label for="title" class="block text-gray-700 mb-2">Title</label>
                    <input type="text" name="title" id="title" value="{{ $book->title }}"
                           class="w-full border border-gray-300 rounded-md px-3 py-2 focus:border-blue-500
focus:ring-1 focus:ring-blue-500">
                </div>
                <div class="mb-4">
                    <label for="author" class="block text-gray-700 mb-2">Author</label>
                    <input type="text" name="author" id="author" value="{{ $book->author }}"
                           class="w-full border border-gray-300 rounded-md px-3 py-2 focus:border-blue-500
focus:ring-1 focus:ring-blue-500">
                </div>
                <div class="mb-4">
                    <label for="description" class="block text-gray-700 mb-2">Description</label>
                    <textarea name="description" id="description"
                           class="w-full border border-gray-300 rounded-md px-3 py-2 focus:border-blue-500
focus:ring-1 focus:ring-blue-500">{{ $book->description }}</textarea>
                </div>
                <div class="mb-4">
                    <label for="published_year" class="block text-gray-700 mb-2">Published Year</label>
                           <input type="number" name="published_year" id="published_year" value="{{
$book->published_year }}"
                           class="w-full border border-gray-300 rounded-md px-3 py-2 focus:border-blue-500
focus:ring-1 focus:ring-blue-500">
                </div>
                <div class="mb-4">
                    <label for="genre" class="block text-gray-700 mb-2">Genre</label>
                    <input type="text" name="genre" id="genre" value="{{ $book->genre }}"
                           class="w-full border border-gray-300 rounded-md px-3 py-2 focus:border-blue-500
focus:ring-1 focus:ring-blue-500">
                </div>
                <div class="flex justify-end space-x-2">
                    <button type="submit"
                                   class="px-4 py-2 bg-blue-500 text-white rounded-md hover:bg-blue-600
focus:outline-none focus:ring-2 focus:ring-blue-500">Save</button>
                    <a href="{{ route('books.index') }}">
                        <button type="button"
                                class="px-4 py-2 bg-gray-300 text-gray-800 rounded-md hover:bg-gray-400
focus:outline-none">Cancel</button>
                    </a>
                    <!-- Delete Button -->
                     <form action="{{ route('books.destroy', $book->id) }}" method="POST" onsubmit="return
confirm('Are you sure you want to delete this book?');">
                        @csrf
                        @method('DELETE')
                        <button type="submit"
```

```
                                    class="px-4 py-2 bg-red-500 text-white rounded-md hover:bg-red-600
focus:outline-none focus:ring-2 focus:ring-red-500 transition duration-200">
                        Delete
                    </button>
                </form>
            </div>
        </form>
    </div>
</body>
</html>
```

d. Delete Books

1. In the destroy method, delete the selected book:

```
public function destroy(Book $book)
{
    $book->delete();
    return redirect()->route('books.index')->with('success', 'Book deleted successfully!');
}
```

2. Add delete buttons in the index view for each book.

Implement Search Functionality

1. Add a search form in the index inside view/books directory:

Copy the HTML code below

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Search Books</title>
    <link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.min.css" rel="stylesheet">
</head>

<body class="bg-gray-200 flex flex-col items-center min-h-screen p-6">
    <!-- Search Form -->
    <form method="GET" action="{{ route('books.index') }}"
        class="flex space-x-2 p-4 bg-gray-100 rounded-md shadow-md mb-8 w-full max-w-lg">
        <input type="text" name="query" placeholder="Search books..."
            class="flex-1 border border-gray-300 rounded-md px-3 py-2 outline-none focus:border-blue-500
focus:ring-1 focus:ring-blue-500 transition duration-200">
        <button type="submit"
                class="px-4 py-2 bg-blue-500 text-white rounded-md hover:bg-blue-600 focus:outline-none
focus:ring-2 focus:ring-blue-500 transition duration-200">
            Search
        </button>
    </form>

    <!-- Button to Open Modal -->
    <button onclick="document.getElementById('bookModal').classList.remove('hidden')"
            class="px-4 py-2 bg-blue-500 text-white rounded-md hover:bg-blue-600 focus:outline-none
focus:ring-2 focus:ring-blue-500 transition duration-200">
        Add New Book
```

```
    </button>

    <!-- Book List -->
    <div class="w-full max-w-4xl mx-auto mt-6">
        <h2 class="text-3xl font-bold mb-6 text-center">Book List</h2>
        @foreach ($books as $book)
            <div class="space-y-4 mb-2">
                <div
                        class="bg-white p-6 rounded-lg shadow-lg border border-gray-200 hover:shadow-xl
transition-shadow duration-300">
                    <div class="flex justify-between items-center mb-2">
                        <h3 class="text-2xl font-semibold text-gray-800">{{ $book->title }}</h3>
                        <div class="flex space-x-2">
                            <!-- Edit Button -->
                            <a href="{{ route('books.edit', $book) }}"
                                    class="px-4 py-2 bg-yellow-500 text-white rounded-md hover:bg-yellow-600
focus:outline-none focus:ring-2 focus:ring-yellow-500 transition duration-200">
                                Edit
                            </a>
                        </div>
                    </div>
                    <p class="text-gray-700 mb-2">{{ $book->description }}</p>
                    <div class="flex justify-between items-center">
                        <span class="text-gray-600 text-sm">Published:
                            <strong>{{ $book->published_year }}</strong></span>
                                <span class="text-gray-600 text-sm">Genre: <strong>{{ $book->genre
}}</strong></span>
                    </div>
                </div>
            </div>
        @endforeach
    </div>

    <!-- Modal -->
        <div id="bookModal" class="fixed inset-0 bg-black bg-opacity-50 flex justify-center items-center
hidden">
        <div class="bg-white p-6 rounded-lg shadow-md w-full max-w-lg">
            <h1 class="text-2xl font-bold mb-4">Add New Book</h1>
            <form method="POST" action="{{ route('books.store') }}">
                @csrf
                <div class="mb-4">
                    <label for="title" class="block text-gray-700 font-semibold mb-2">Title</label>
                    <input type="text" id="title" name="title" placeholder="Enter book title"
                                class="w-full border border-gray-300 rounded-md px-3 py-2 outline-none
focus:border-blue-500 focus:ring-1 focus:ring-blue-500 transition duration-200"
                        required>
                </div>
                <div class="mb-4">
                    <label for="author" class="block text-gray-700 font-semibold mb-2">Author</label>
                    <input type="text" id="author" name="author" placeholder="Enter author's name"
                                class="w-full border border-gray-300 rounded-md px-3 py-2 outline-none
focus:border-blue-500 focus:ring-1 focus:ring-blue-500 transition duration-200"
                        required>
                </div>
                <div class="mb-4">
                                    <label for="description" class="block text-gray-700 font-semibold
mb-2">Description</label>
```

```
                        <textarea type="text" id="author" name="description" placeholder="Enter book's description"
                                  class="w-full border border-gray-300 rounded-md px-3 py-2 outline-none focus:border-blue-500 focus:ring-1 focus:ring-blue-500 transition duration-200"
                            required></textarea>
                </div>
                <div class="mb-4">
                     <label for="published_year" class="block text-gray-700 font-semibold mb-2">Published year</label>
                    <input type="number" id="author" name="published_year"
                        placeholder="Enter book's publish year"
                                  class="w-full border border-gray-300 rounded-md px-3 py-2 outline-none focus:border-blue-500 focus:ring-1 focus:ring-blue-500 transition duration-200"
                            required>
                </div>
                <div class="mb-4">
                    <label for="genre" class="block text-gray-700 font-semibold mb-2">Genre</label>
                    <input type="text" id="author" name="genre" placeholder="Enter book's genre"
                                  class="w-full border border-gray-300 rounded-md px-3 py-2 outline-none focus:border-blue-500 focus:ring-1 focus:ring-blue-500 transition duration-200"
                            required>
                </div>
                <div class="flex justify-end space-x-2">
                                                                        <button type="button" onclick="document.getElementById('bookModal').classList.add('hidden')"
                                   class="px-4 py-2 bg-gray-500 text-white rounded-md hover:bg-gray-600 focus:outline-none focus:ring-2 focus:ring-gray-500 transition duration-200">
                            Cancel
                        </button>
                        <button type="submit"
                                   class="px-4 py-2 bg-blue-500 text-white rounded-md hover:bg-blue-600 focus:outline-none focus:ring-2 focus:ring-blue-500 transition duration-200">
                            Add Book
                        </button>
                    </div>
                </form>
            </div>
        </div>
    </body>

    </html>
```

2. Update the index method to handle search queries:

```
    public function index(Request $request)
    {
        $query = $request->input('query');

        $books = Book::when($query, function ($q) use ($query) {
            return $q->where('title', 'like', "%$query%")
                     ->orWhere('author', 'like', "%$query%");
        })->get();

        return view('books.index', compact('books'));
    }
```

Update the web.php

Test Your Application
- Add sample data to the database.
- Verify that all functionalities (search, create, edit, update, delete) work as expected.

Deliverables
1. A GitHub repository containing the Laravel project.
2. A short document explaining the functionality and how to run the project.

Criteria for Grading

Here's a rubric template for evaluating web development projects using Laravel and Tailwind CSS. This rubric divides evaluation into three categories with detailed criteria:

Web Development Rubric for Laravel & Tailwind CSS Projects

1. Functionality (40 points)

- **Excellent (36-40 points)**: The application fully meets all functional requirements. All features, such as CRUD operations, form validation, authentication, and other Laravel functionalities, work seamlessly without any bugs.
- **Good (28-35 points)**: The application meets most functional requirements, with only minor issues or incomplete features that do not significantly impact the user experience.
- **Satisfactory (20-27 points)**: The application meets some functional requirements, but several features have issues or are incomplete. This affects usability.
- **Needs Improvement (0-19 points)**: The application fails to meet key functional requirements, with major bugs or incomplete core features that impact overall usability.

2. Design and UI/UX (30 points)

- **Excellent (27-30 points)**: The application has an appealing, responsive, and intuitive design using Tailwind CSS. The layout is well-structured, and elements are styled consistently. Accessibility considerations are taken into account.
- **Good (21-26 points)**: The design is visually appealing and mostly responsive, but there are minor inconsistencies in styling or a few usability issues.
- **Satisfactory (15-20 points)**: The design is functional but may lack polish or have issues with responsiveness or accessibility. Styling inconsistencies are present.
- **Needs Improvement (0-14 points)**: The design is not user-friendly or responsive, and styling is inconsistent or not aligned with best practices. Accessibility considerations are lacking.

3. Code Quality and Best Practices (30 points)

- **Excellent (27-30 points)**: Code is clean, well-organized, and adheres to Laravel and Tailwind CSS best practices. Proper naming conventions, modular structure, and efficient use of resources (e.g., blade components and Tailwind utility classes) are evident. Documentation is clear.
- **Good (21-26 points)**: Code is mostly well-organized and follows best practices, with minor deviations. Most components are modular, and documentation is adequate but could be improved.
- **Satisfactory (15-20 points)**: Code structure is somewhat organized but contains noticeable issues like code duplication or improper use of Laravel and Tailwind CSS conventions. Documentation is sparse or unclear.
- **Needs Improvement (0-14 points)**: Code is disorganized, difficult to read, or violates core best practices for Laravel and Tailwind CSS. Significant issues with code modularity or documentation.

Total: 100 points

This rubric can be adjusted as needed based on project specifications or additional criteria

Instructor A Deliverables
Save you file as *[Laravel Student Task]-[students last name]-[student_first name]-[date today].pdf*
Example: *Laravel Student Task*-Millena-Jay-December 9, 2024.pdf

Deadline of Submission
The deadline is set for December 5, 2024. Please send it to me via private message on my Messenger at *www.fb.com/knightofdaraga* | Jay Millena - Instructor A