

DOCUMENTATION:

(FOR THE LAB ACTIVITY 2)

I. BASIC ROUTES**a. HOMEPAGE ROUTE**

```
// Homepage route
Route::get('/homepage', [LoginController::class, 'homepage'])->name('homepage');
```

Purpose: This route is for displaying the homepage after successful login.

Controller: The homepage method in the LoginController is responsible for returning the homepage view.

b. ABOUT ROUTE

```
// About route with an optional parameter {username?}
Route::get('/about/{username?}', [LoginController::class, 'about'])
    ->name('about')
    ->where('username?', '[A-Za-z]'); // Constraint: only alphabetic characters
```

This route is for the "About" page and allows an optional username in the URL (e.g., /about/John). It uses the **LoginController's about()** method to display a personalized message. If no username is provided, it defaults to 'guest' or gets the name from the session. The route ensures the username consists only of alphabetic characters.

c. REDIRECT ROUTE

```
// Redirect /home to /
Route::redirect('/home', '/');
```

This route automatically redirects any requests to ``/home`` to the root URL ``/``. Its purpose is to ensure users accessing ``/home`` are sent to the main page, streamlining navigation.

d. CONTACT FORM ROUTE

```
// Contact form route
Route::post('/contact', function () {
    $data = request()->all();
    Mail::to('mingkai103019@gmail.com')->send(new ContactMe($data));
    return redirect('/contact')->with('flash', 'Message Sent Successfully');
});
```

This route handles form submissions for the contact page. It listens for **POST requests to /contact**, retrieves all input data from the request, and sends an email to mingkai103019@gmail.com using the ContactMe Mailable class. After sending the email, it redirects the user back to the /contact page with a success message indicating that the message was sent successfully. This provides feedback to the user about their action.

e. REDIRECT ROOT ROUTE

```
// Redirect root route to the login page
Route::get('/', function () {
    return redirect()->route('login');
});
```

Purpose: This route handles requests to the root URL (/).

Functionality: It redirects users to the login page. This is useful for ensuring that users are authenticated before accessing the application.

f. LOGIN ROUTE

```
// Login route
Route::get('/login', function () {
    return view('login');
})->name('login');

Route::post('/login', [LoginController::class, 'login']);
```

GET /login: This route responds to GET requests at the /login URL. It returns the login view, which is where users can enter their login credentials. The route is named 'login' for easy reference elsewhere in the application.

POST /login: This route handles POST requests to /login. It uses the login method from the LoginController to process the login form submission. This method typically includes validation of the credentials and logic for authenticating the user.

II. ROUTE PARAMETERS

```
Route::get('/about/{username?}', [LoginController::class, 'about'])
    ->name('about')
    ->where('username?', '[A-Za-z]'); // Constraint: only alphabetic characters

Route::get('/content/{username?}', [LoginController::class, 'content'])
    ->name('content')
    ->where('username?', '[A-Za-z]'); // Constraint: only alphabetic characters

Route::get('/contact/{username?}', [LoginController::class, 'contact'])
    ->name('contact')
    ->where('username?', '[A-Za-z]'); // Constraint: only alphabetic characters

Route::get('/user/{username?}', [LoginController::class, 'userProfile'])
    ->name('user.profile')
    ->where('username?', '[A-Za-z]'); // Constraint: only alphabetic characters
```

The **username?** in **{username?}** makes the username parameter optional.

Constraints (**where('username?', '[A-Za-z]+')**) ensure that the parameter only contains alphabetic characters.

These routes dynamically accept or ignore the username depending on the URL input, offering flexibility for your application.

```

class LoginController extends Controller
{
    public function login(Request $request)
    {
        // Validate the request (allowing optional username)
        $request->validate([
            'username' => 'nullable|alpha', // Ensure username contains only alphabetic
        ]);

        // Get the username from the request, default to 'guest' if not provided
        $username = $request->input('username');
        if (empty($username)) {
            $username = 'guest'; // Default to 'guest' if username is empty
        }

        // Store the username in the session
        $request->session()->put('username', $username);

        // Redirect to the homepage with username in the URL
        return redirect()->route('homepage', ['username' => $username]);
    }

    public function homepage(Request $request, $username = null)
    {
        // Retrieve the username from the URL, or from the session if not provided
        $username = $username ?: $request->session()->get('username', 'guest');

        // Pass the username to the homepage view
        return view('homepage', ['message' => "Welcome, $username!"]);
    }

    public function about(Request $request, $username = null)
    {
        // Retrieve the username from the URL, or from the session if not provided
        $username = $username ?: $request->session()->get('username', 'guest');

        // Pass the username to the about view
        return view('about', ['message' => "Welcome, $username!"]);
    }

    public function content(Request $request, $username = null)
    {
        // Retrieve the username from the URL, or from the session if not provided
        $username = $username ?: $request->session()->get('username', 'guest');

        // Pass the username to the content view
        return view('content', ['message' => "Welcome, $username!"]);
    }

    public function contact(Request $request, $username = null)
    {
        // Retrieve the username from the URL, or from the session if not provided
        $username = $username ?: $request->session()->get('username', 'guest');

        // Pass the username to the contact view
        return view('contact', ['message' => "Welcome, $username!"]);
    }

    public function userProfile(Request $request, $username = null)
    {
        // Default to 'guest' if no username is provided
        $username = $username ?: 'guest';

        // Store the username in the session
        $request->session()->put('username', $username);

        // Redirect to the homepage with username in the URL
        return redirect()->route('homepage', ['username' => $username]);
    }
}

```

LOGIN CONTROLLER

1. login(Request \$request)

- This method handles the login form submission.
- **Validation:** It validates that the username input (if provided) contains only alphabetic characters. If no username is provided, it is considered nullable.
- **Default Value:** If the username is not provided or is empty, it defaults to 'guest'.
- **Session Storage:** The username is stored in the session using `$request->session()->put('username', $username)`.
- **Redirection:** After storing the username, the user is redirected to the homepage, with the username passed in the URL (`redirect()->route('homepage', ['username' => $username])`).

2. homepage(Request \$request, \$username = null)

- This method is responsible for rendering the homepage view.
- **Session or URL Username:** It first checks if the username is provided via the URL. If not, it retrieves the username from the session, defaulting to 'guest'.
- **Passing Data to View:** It passes a personalized welcome message ("Welcome, \$username!") to the homepage view.
- **View:** The view rendered is `homepage.blade.php`.

3. about(Request \$request, \$username = null)

- Similar to the homepage method, this one handles the "About" page.
- **Username Handling:** It follows the same logic—if no username is in the URL, it retrieves the username from the session or defaults to 'guest'.
- **Passing Data to View:** The message is passed to the about view.

4. content(Request \$request, \$username = null)

- This method handles the "Content" page.
- **Username Handling:** It retrieves the username either from the URL or the session, with a default of 'guest' if not found.
- **Passing Data to View:** The username is embedded in a welcome message passed to the content view.

5. contact(Request \$request, \$username = null)

- This method manages the "Contact" page.
- **Username Handling:** Just like the previous methods, the username is fetched from either the URL or session.
- **Passing Data to View:** The username is passed in a personalized message to the contact view.

6. userProfile(Request \$request, \$username = null)

- This method handles the user profile page, and unlike the others, if the username is not provided, it directly defaults to 'guest'.
- **Session Storage:** It stores the username in the session.
- **Redirection:** After storing the username, it redirects to the homepage with the username included in the URL.

III.VIEWS

