Logotipo, nombre de la empresa

Descripción generada automáticamente

|  |  |  |
| --- | --- | --- |
| **Nombre**:  Héctor Endy Pérez Navarro | | **Matrícula**:  03009066 |
| **Nombre del curso:**  Diseño de aplicaciones web | **Nombre del profesor**:  Prof. Cristopher Gerardo Gaytan Diaz | |
| **Módulo**: 3 | **Actividad**:Tarea 8 | |
| **Fecha**: 29-10-2024 | | |
| **Bibliografía**: | | |

**1. Crear el proyecto Laravel**

Abrimos la terminal y nos dirigimos a la carpeta donde queremos crear el proyecto. Ejecutamos:

laravel new actividad-9

**Esto generará el proyecto llamado “actividad-9”. Entra a la carpeta:**

cd actividad-9

**2. Inicializar un repositorio Git y crear el repositorio en GitHub**

**Inicializa el repositorio:**

git init

**Crea un nuevo repositorio en GitHub con el nombre “actividad-9” y luego enlaza el repositorio local al de GitHub:**

git remote add origin <URL\_DEL\_REPOSITORIO\_GITHUB>

**3. Instalar y configurar autenticación**

**Laravel 7 tiene un paquete llamado laravel/ui para configurar la autenticación básica. Para instalarlo, ejecutamos:**

composer require laravel/ui

**Luego, generamos la autenticación:**

php artisan ui bootstrap --auth

npm install && npm run dev

Esto generará vistas y rutas para registro e inicio de sesión y cargará los archivos necesarios para el diseño en public/css y public/js.

**4. Modificar las vistas y las rutas de acceso**

1. **Crear la vista “landingpage”**:
   * Crea un archivo en resources/views llamado landingpage.blade.php.
   * Añade contenido básico, indicando que es la página de bienvenida para usuarios no registrados.
2. **Crear la vista “dashboard”**:
   * Laravel creará una vista home.blade.php en resources/views que servirá de base para el dashboard. Renombramos este archivo a dashboard.blade.php y personalízalo como prefieras para usuarios registrados.
3. **Configurar las rutas**:
   * En routes/web.php, agregamos estas rutas para redirigir a las vistas correctas:

Route::get('/', function () {

return view('landingpage');

})->name('landingpage');

Route::get('/dashboard', function () {

return view('dashboard');

})->middleware('auth')->name('dashboard');

1. **Con esto, al ingresar a la raíz (/), se mostrará la landingpage, y solo los usuarios autenticados podrán acceder a /dashboard.**

**5. Personalizar la barra de navegación**

**Abrimos resources/views/layouts/app.blade.php y realizamos los siguientes cambios:**

* Usamos una directiva @auth para mostrar el enlace a “Dashboard” solo a usuarios autenticados.
* Usamos @guest para mostrar los enlaces de “Iniciar sesión” y “Registrarse” solo a los usuarios invitados.

<ul class="navbar-nav ml-auto">

@guest

<li class="nav-item">

<a class="nav-link" href="{{ route('login') }}">Iniciar sesión</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{{ route('register') }}">Registrarse</a>

</li>

@else

<li class="nav-item">

<a class="nav-link" href="{{ route('dashboard') }}">Dashboard</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{{ route('logout') }}" onclick="event.preventDefault(); document.getElementById('logout-form').submit();">

Cerrar sesión

</a>

<form id="logout-form" action="{{ route('logout') }}" method="POST" style="display: none;">

@csrf

</form>

</li>

@endguest

</ul>

**6. Proteger el acceso al Dashboard**

Ya que usamos el middleware auth, solo los usuarios registrados pueden ver el dashboard. Si un usuario no autenticado intenta acceder a /dashboard, será redirigido a la página de login.

**7. Modificamos el archivo README.md**

Agregamos una descripción básica del proyecto, especificando las funcionalidades de autenticación y seguridad implementadas. También incluye instrucciones de instalación y de despliegue.

Ejemplo:

# Proyecto Actividad-9

Este proyecto implementa un sistema de autenticación básico en Laravel 7 con diferentes vistas para usuarios registrados y no registrados.

- La página principal `landingpage` se muestra a usuarios no autenticados.

- La vista `dashboard` es accesible solo para usuarios registrados.

- Barra de navegación dinámica según el estado de autenticación del usuario.

## Instalación

1. Clonar el repositorio.

2. Ejecutar `composer install`.

3. Configurar las variables de entorno en `.env`.

4. Ejecutar `php artisan migrate` para configurar la base de datos.

**8. Realizar capturas de pantalla y documentación**

**Realiza un commit de tus cambios y súbelos a GitHub:**

bash

Copiar código

git add .

git commit -m "Configuración básica de seguridad y autenticación"

git push origin main