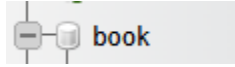


LABORATORY ACTIVITY: Populating From a Database

DATABASE:



We created a table named 'book' in our chosen MySQL server, Xampp.

.env:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=book
DB_USERNAME=root
DB_PASSWORD=
```

We modified the .env file to update the necessary configuration settings for our application.

CONTROLLERS: LandingPageController.php

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5
6  use Illuminate\Http\Request;
7  use App\Models\Landing;
8
9  class LandingPageController extends Controller
10 {
11     /**
12      * Display a listing of the resource.
13      */
14     public function index()
15     {
16         //
17         $popularBooks = Landing::all(); |
18         // Add dynamic asset path (Assets 1, Assets 2, etc.)
19         foreach ($popularBooks as $index => $book) {
20             // Dynamically build the asset image path based on the book index (1, 2, 3, etc.)
21             $book->image_url = asset('assets/img/Asset ' . ($index + 1) . '.png');
22         }
23
24         return view('LandingPage', compact('popularBooks'));
25     }
26
27     /**
28      * Show the form for creating a new resource.
29      */
30     public function create()
```

This LandingPageController is a resource controller that handles CRUD operations for the Landing model. The index method fetches all records from the database using the Landing::all() function and stores them in the popularBooks variable. It also assigns an image_url to each book by looping through the records and generating an asset path (e.g., Asset 1.png, Asset 2.png) based on their index. This ensures each book has a corresponding image for display in the view.

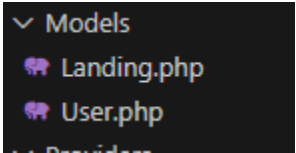
CONTROLLERS: UserController.php

```
1 namespace App\Http\Controllers;
2
3 use App\Models\UserLogin;
4 use Illuminate\Http\Request;
5 use Illuminate\Support\Facades\Hash;
6 use Illuminate\Support\Facades\Log;
7 use Illuminate\Support\Facades\Auth;
8
9 class UserController extends Controller
10 {
11     public function index()
12     {
13         return view('login-page');
14     }
15
16     public function login(Request $request)
17     {
18         // Validate input
19         $request->validate([
20             'username' => 'required',
21             'password' => 'required',
22         ]);
23
24         // Check user credentials
25         $user = UserLogin::where('username', $request->username)->first();
26
27         if ($user && Hash::check($request->password, $user->password)) {
28             // Successful login, store user info in session
29             session(['user' => $user->username]);
30             return redirect('/landings'); // Redirect to a dashboard or home page
31         }
32     }
33 }
```

This UserController manages user authentication and registration. The index method shows the login page, while the login method validates user input, checks credentials in the UserLogin model, and redirects to the landing page if successful or back with an error if invalid. The register method ensures input is valid, the username is unique, and securely stores the user's password using Hash. After registration, users are redirected to the login page with a success message.

MODELS: Landing.php

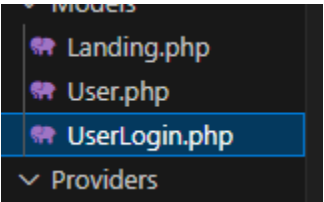
```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Landing extends Model
9 {
10     use HasFactory;
11     //table name
12     protected $table = 'landings';
13
14     //primary key
15     public $primaryKey = 'id';
16
17     //Timestamps
18
19     public $timestamps = true;
20 }
21
22
```



The Landing model maps to the 'landings table', uses id as its primary key, and manages timestamps (created_at and updated_at) automatically.

MODELS: UserLogin.php

```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use Illuminate\Foundation\Auth\User as Authenticatable;
8 use Illuminate\Notifications\Notifiable;
9
10 class UserLogin extends Authenticatable
11 {
12     use HasFactory, Notifiable;
13
14     protected $fillable = [
15         'username', 'password', 'birthday'
16     ];
17
18     protected $hidden = [
19         'password', 'remember_token',
20     ];
21
22     public $primaryKey = 'id';
23 }
```



This UserLogin model handles user data and extends Laravel's Authenticatable class, providing built-in authentication features. It uses the HasFactory trait for database seeding and Notifiable for notifications. The \$fillable property allows secure mass assignment for the username, password, and birthday fields, while \$hidden hides sensitive fields like password and remember_token. The model represents a database table with id as the primary key and ensures secure management of user attributes.

MIGRATIONS: landings & user_logins

When php artisan migrate is executed this migration creates a landings table in the database with columns: id (primary key), title, rating (default 0), author, and timestamps for tracking changes. The down method drops the table when the migration is rolled back.

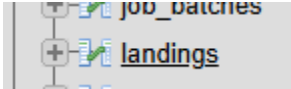
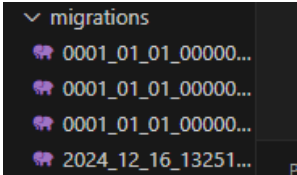
Landings:

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('landings', function (Blueprint $table) {
            $table->increments('id');
            $table->string('title');
            $table->integer('rating')->default(0);
            $table->string('author');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('landings');
    }
};
```



Show all

Number of rows: 25

Filter rows:

Search this table

Sort by key: None

Extra options

id

title

rating

author

created_at

updated_at

Edit

Copy

Delete

1

Quis molestiae eum aut porro aut rerum.

4

Suscipit culpa vitae quis voluptas ipsa qui iure q...

2024-12-16 16:01:58

2024-12-16 16:01:58

Edit

Copy

Delete

2

Facere qui fuga exercitationem est id voluptates r...

2

Quia voluptatem laboriosam aut voluptatem officia ...

2024-12-16 16:01:58

2024-12-16 16:01:58

Edit

Copy

Delete

3

Soluta ut distinctio quam consequatur.

1

Animi et id tempore excepturi vel omnis.

2024-12-16 16:01:58

2024-12-16 16:01:58

Edit

Copy

Delete

4

First Book Title

4

First Book Author

2024-12-16 16:01:58

2024-12-16 16:01:58

Check all

With selected:

Edit

Copy

Delete

Export

User_logins

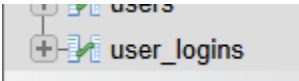
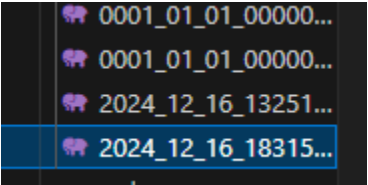
```
base 7 migrations > 2024_12_16_183154_create_user_logins_table.php

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('user_logins', function (Blueprint $table) {
            $table->id();
            $table->string('username')->unique(); // Username field (unique constraint)
            $table->date('birthday'); // Birthday field
            $table->string('password'); // Password field
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('user_logins');
    }
};
```



<input type="checkbox"/>				id	username	birthday	password	created_at	updated_at
<input type="checkbox"/>				1	admin	2000-01-01	\$2y\$12\$Zxs0XhNG5ZEJKOa580IMOADeFA1B1YLVLsM2gkjAvg...	2024-12-16 16:41:47	2024-12-16 16:41:47
<input type="checkbox"/>				2	karylle	2003-10-31	\$2y\$12\$w2fF9Z5LAIdzypJLYdz3eLgKXKKBvrryQd3hr86ir...	2024-12-16 21:49:07	2024-12-16 21:49:07
<input type="checkbox"/>				3	telle	2003-11-23	\$2y\$12\$SGTB1XFae1uESEGss7h.N8B0WK36uLqd287T.SGm...	2024-12-16 21:52:54	2024-12-16 21:52:54
<input type="checkbox"/>				4	karel	2003-06-11	\$2y\$12\$sdDptc.4oNBv.Vebppj1zay9KMDwhf6FUuzZdVnW.TB8...	2024-12-16 22:02:49	2024-12-16 22:02:49

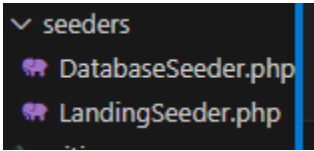
When php artisan migrate is executed this migration creates a user_logins table in the database with an auto-incrementing id, a unique username, birthday, and password fields, along with timestamps for tracking creation and updates. The down method ensures the table is dropped if the migration is rolled back.

SEEDING: LandingSeeder.php and UserLoginSeeder.php

```
database > seeders > LandingSeeder.php

1 <?php
2
3 namespace Database\Seeders;
4
5 use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6 use Illuminate\Database\Seeder;
7 use App\Models\Landing;
8
9 class LandingSeeder extends Seeder
10 {
11     /**
12      * Run the database seeds.
13      */
14     public function run(): void
15     {
16         //
17         Landing::factory()->count(3)->create();
18
19         Landing::create([
20             'title' => 'First Book Title',
21             'rating' => 4,
22             'author' => 'First Book Author',
23             'created_at' => now(),
24             'updated_at' => now(),
25         ]);
26     }
27 }
```

LandingSeeder:



Show all

Number of rows:

25

Filter rows:

Search this table

Sort by key:

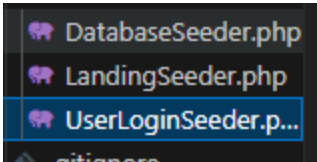
None

Extra options

UserLoginSeeder:

```
use App\Models\UserLogin;

class UserLoginSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        //
        UserLogin::create([
            'username' => 'admin',
            'birthday' => '2000-01-01',
            'password' => bcrypt('admin123'),
            'created_at' => now(),
            'updated_at' => now(),
        ]);
    }
}
```



	id	username	birthday	password	created_at	updated_at
<input type="checkbox"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="checkbox"/>	1	admin	2000-01-01	\$2y\$12\$HpmzEb5DWQnojYMI3Vbag.zrVAD59HKCTX.WBtraN4k...	2024-12-16 18:45:28	2024-12-16 18:45:28

Using php artisan db:seed --class=SeederName, it will automatically populate the tables with the initial data. This seeder populates both the landings and user_logins tables. For the landings table, it uses a factory to create 3 random records, then manually adds one specific entry with predefined values for the title, rating, author, and timestamps. For the user_logins table, it inserts a new user with the username 'admin', a birthday of '2000-01-01', and a hashed password of 'admin123' using the bcrypt function, with the created_at and updated_at fields set to the current timestamp.

Define the routes and register the controllers.

```
use App\Http\Controllers;

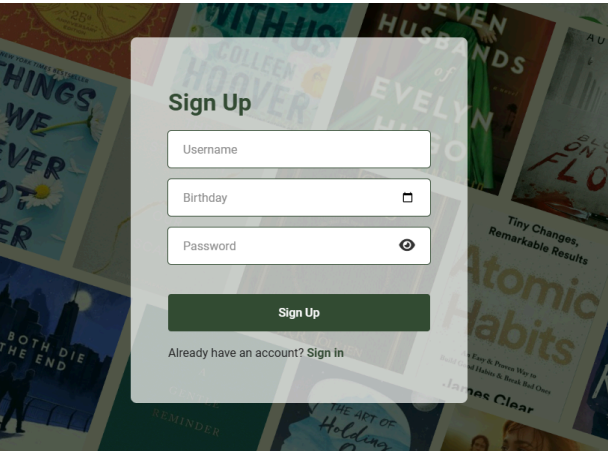
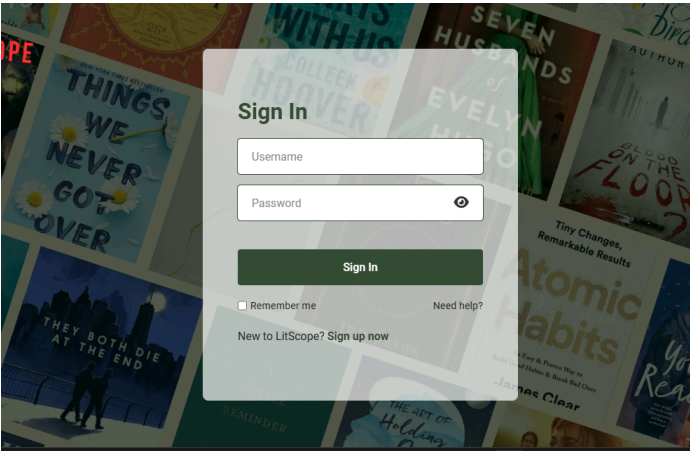
// Display the login form (GET request)
Route::get('/', [UserController::class, 'showLoginForm'])->name('login');

// Handle the login form submission (POST request)
Route::post('/login', [UserController::class, 'loginRequest'])->name('login.submit');

// Handle signup
Route::post('/signup', [UserController::class, 'register'])->name('signup.submit');

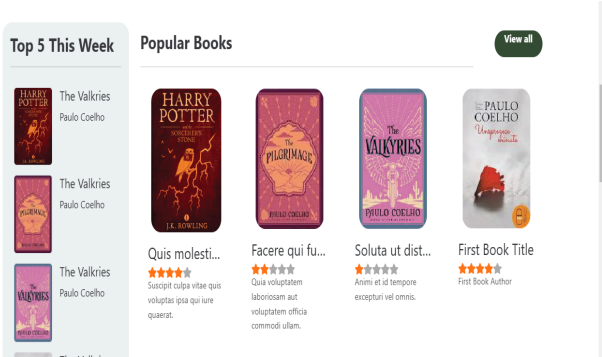
// Landing page (after successful login)
Route::get('/landing', [LandingPageController::class, 'index'])->name('landing');
```

The '/' and 'signup' routes are handled by a controller with the showLoginForm and register functions. The showLoginForm function displays the login form, while the register function displays the signup form.



Once the user enters their username and password, Laravel checks if the credentials are stored in the database. If they match, the user is redirected to /landing, which corresponds to the homepage.

Homepage:



If a new user creates an account, their information is validated and securely stored in the database. After successful registration, the user is redirected to /, where the login form is displayed. They can then log in using their newly created credentials, and the authentication process will begin again. If the login is successful, they will be redirected to /landing the homepage. This cycle ensures that only authenticated users can access protected pages on the site.