LARAVEL ASSESSMENT

DOCUMENTATION BY:

1. MUHAMMAD AFIQ BIN ALUSRIZAN

Description

This document outlines the process I followed in completing this assessment. While there were a few mistakes made along the way, they provided valuable learning opportunities, and I was able to improve through those experiences.

Source: https://laravel.com/

Install laravel

- My Composer version was outdated (from 2021 😅),

-So I updated it.

```
C:\Users\Aifiq>composer self-update
Upgrading to version 2.8.8 (stable channel).
```
\

Create a new Laravel project

```
C:\Users\Aifiq>laravel new form_project

        |  ___ _ __ __ ___   _____| |
        | |   | '__/ _` \ \ / / _ \ |
        | |___| | | (_| |\ V /  __/ |
        |_____|_|  \__,_| \_/ \___|_|

Would you like to install a starter kit? [No starter kit]:
  [none     ] No starter kit
  [breeze   ] Laravel Breeze
  [jetstream] Laravel Jetstream
> none

Which testing framework do you prefer? [PHPUnit]:
  [0] PHPUnit
  [1] Pest
> 1

Would you like to initialize a Git repository? (yes/no) [no]:
> yes
```

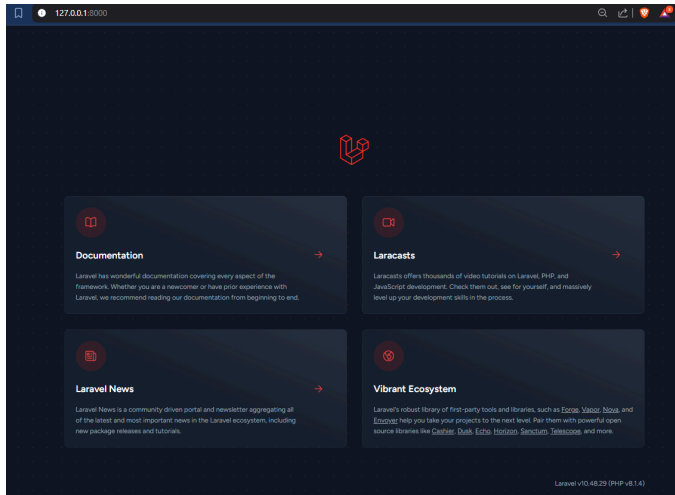After creating the project, I tested it first.

```
PS C:\Users\Aifiq\form_project> php artisan serve

   INFO  Server running on [http://127.0.0.1:8000].

  Press Ctrl+C to stop the server

  2025-04-17 13:39:33 ........................................... ~ 1s
  2025-04-17 13:39:34 /favicon.ico ........................... ~ 0s
  []
```
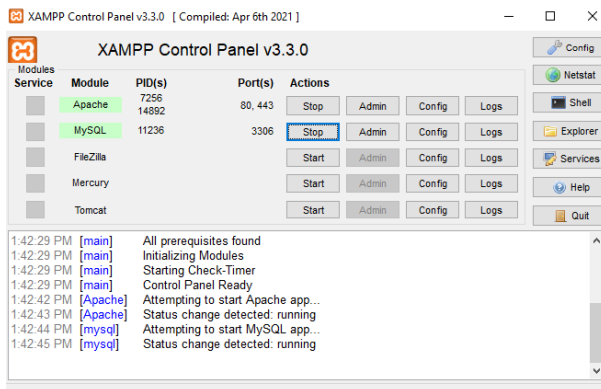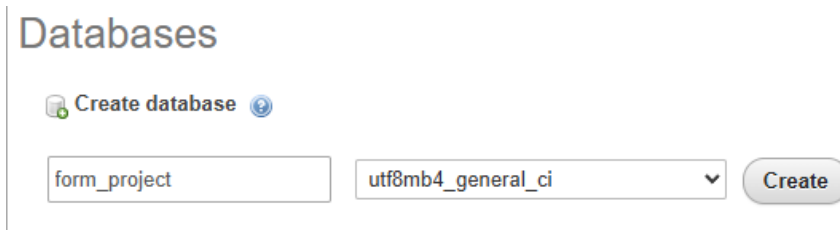
Default Laravel setup

Start XAMPP

Create a new database

Update the .env file

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

Then, check the Laravel database connection.

```
PS C:\Users\Aifiq\form_project> php artisan migrat
e

  INFO  Preparing database.

  Creating migration table .......... 38ms DONE

  INFO  Running migrations.

  2014_10_12_000000_create_users_table  44ms DONE
  2014_10_12_100000_create_password_reset_tokens_table  32ms DONE
  2019_08_19_000000_create_failed_jobs_table  117ms DONE
  2019_12_14_000001_create_personal_access_tokens_table  63ms DONE

PS C:\Users\Aifiq\form_project>
```

Create a table

P/S: I usually create tables manually, but according to the documentation, we're supposed to use a command—so let's try that:

https://laravel.com/docs/12.x/migrations#main-content

```
PS C:\Users\Aifiq\form_project> php artisan make:migration create_user

  INFO  Migration [C:\Users\Aifiq\form_project\database\migrations/2025_04_17_060328_create_user.php] created successfully.
```

The table was created successfully.

```
🐘 2025_04_17_060328_create_user.php                                    U
> coodors
```

Add code according to the requirements

```php
public function up(): void
{
    Schema::create('user', function (Blueprint $table) {
        $table->id();
        $table->string('name');
        $table->string('email')->unique();
        $table->string('password');
        $table->enum('gender', ['Male', 'Female']);
        $table->date('birthday');
        $table->boolean('status_active')->default(false);
        $table->softDeletes();
        $table->timestamps();
    });
}
```
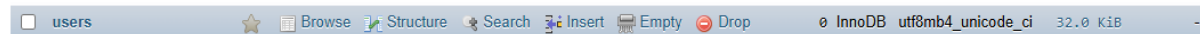
Then run the migrate command.

```
                        php artisan migrate
>> C:\Users\Aifiq\form_project>

  INFO  Running migrations.

  2025_04_17_060328_create_user ........................................................ 44ms DONE
```
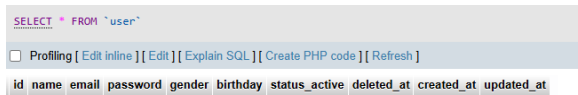
Checked the database , it's there ✅

| | users | ⭐ | 🔲 Browse | 🔧 Structure | 🔍 Search | 🔢 Insert | 🗑 Empty | ⊘ Drop | 0 | InnoDB | utf8mb4_unicode_ci | 32.0 KiB | - |

Also verified the contents inside the table.

```
SELECT * FROM `user`
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

| id | name | email | password | gender | birthday | status_active | deleted_at | created_at | updated_at |

Create a module

https://laravel.com/docs/12.x/eloquent

```
PS C:\Users\Aifiq\form_project> php artisan make:model Users

  INFO  Model [C:\Users\Aifiq\form_project\app\Models\Users.php] created successfully.
```

🐘 Users.php   U

Add the necessary code.

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\SoftDeletes;


class Users extends Model
{
    use HasFactory;

    protected $fillable = [
        'name', 'email', 'password', 'gender', 'birthday', 'status_active'
    ];

    use SoftDeletes;
}
```

Create a controller

https://laravel.com/docs/12.x/controllers#main-content

```
  INFO  Model [C:\Users\Aifiq\form_project\app\Models\Users.php] created succe

PS C:\Users\Aifiq\form_project> php artisan make:controller UserController
```

```
∨ Http
  ∨ Controllers
    🐘 Controller.php
    🐘 UserController.php
```
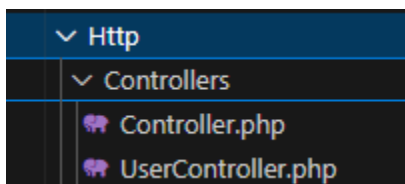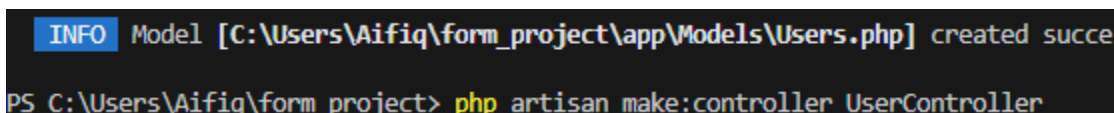
After that, I realized that I had followed the requirements incorrectly; I should have used Laravel version 8 and PHP version 8.1. After making several setups and changes, I started my process from the beginning. However, I did not have time to take individual screenshots. The image below shows the name and version changes.

```
PS C:\Users\Aifiq\form_website> php -v
PHP 8.1.4 (cli) (built: Mar 16 2022 09:33:31) (ZTS Visual C++ 2019 x64)
Copyright (c) The PHP Group
Zend Engine v4.1.4, Copyright (c) Zend Technologies
PS C:\Users\Aifiq\form_website> php artisan --version
Laravel Framework 8.83.29
PS C:\Users\Aifiq\form_website>
```

```
🐘 2025_04_17_101645_create_members_table.php
```

```
🐘 MemberController.php
```

```
∨ Models
   🐘 Member.php
```

I developed a function in the `MemberController` to store data.

```php
// Store members
public function store(Request $request)
{
    $request->validate([
        'name'     => 'required|string|max:255',
        'email'    => 'required|email|unique:members,email',
        'password' => 'required|string|min:6',
        'gender'   => 'required|in:Male,Female',
        'birthday' => 'required|date',
    ]);

    Member::create([
        'name'     => $request->name,
        'email'    => $request->email,
        'password' => Hash::make($request->password),
        'gender'   => $request->gender,
        'birthday' => $request->birthday,
        'status_active'   => $request->has('status_active'),
    ]);

    return redirect()->route('member.table');
}
```

I also implemented the code to display tables for both active and inactive members.

```php
// Show table page
public function table()
{
    $activeMembers = Member::where('status_active', true)->get();
    $inactiveMembers = Member::where('status_active', false)->get();

    return view('member.table', compact('activeMembers', 'inactiveMembers'));
}
```
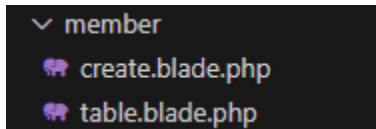
Additionally, I implemented soft delete functionality for the data.

```php
// Soft delete members
public function delete($id)
{
    $members = Member::findOrFail($id);
    $members->delete();

    return back()->with('success', 'members deleted successfully.');
}
```

After that, I created the Blade view files: `create.blade.php` and `table.blade.php`.

```
v member
    create.blade.php
    table.blade.php
```

Subsequently, I designed the user interface and integrated all the available data and functions.

```php
<form action="{{ route('member.store') }}" method="POST">
    @csrf

    <div class="mb-3">
        <label class="form-label">Name</label>
        <input type="text" name="name" class="form-control" placeholder="Enter full name" value="{{ old('name') }}">
    </div>

    <div class="mb-3">
        <label class="form-label">Email</label>
        <input type="email" name="email" class="form-control" placeholder="example@mail.com" value="{{ old('email') }}">
    </div>

    <div class="mb-3">
        <label class="form-label">Password</label>
        <input type="password" name="password" class="form-control" placeholder="********">
    </div>

    <div class="mb-3">
        <label class="form-label">Gender</label>
        <select name="gender" class="form-select">
            <option value="Male" {{ old('gender') === 'Male' ? 'selected' : '' }}>Male</option>
            <option value="Female" {{ old('gender') === 'Female' ? 'selected' : '' }}>Female</option>
        </select>
    </div>

    <div class="mb-3">
        <label class="form-label">Birthday</label>
        <input type="date" name="birthday" class="form-control" value="{{ old('birthday') }}">
    </div>

    <div class="form-check mb-4">
        <input class="form-check-input" type="checkbox" name="status_active" id="status_active" {{ old('status_active') ? 'checked' : '' }}>
        <label class="form-check-label" for="status_active">
```

I then updated my model accordingly.

```php
class Member extends Model
{
    use HasFactory;
    use SoftDeletes;

    protected $fillable = [
        'name',
        'email',
        'password',
        'gender',
        'birthday',
        'status_active'
    ];

    protected $casts = [
        'birthday' => 'date',
    ];

    protected $dates = ['deleted_at'];
}
```
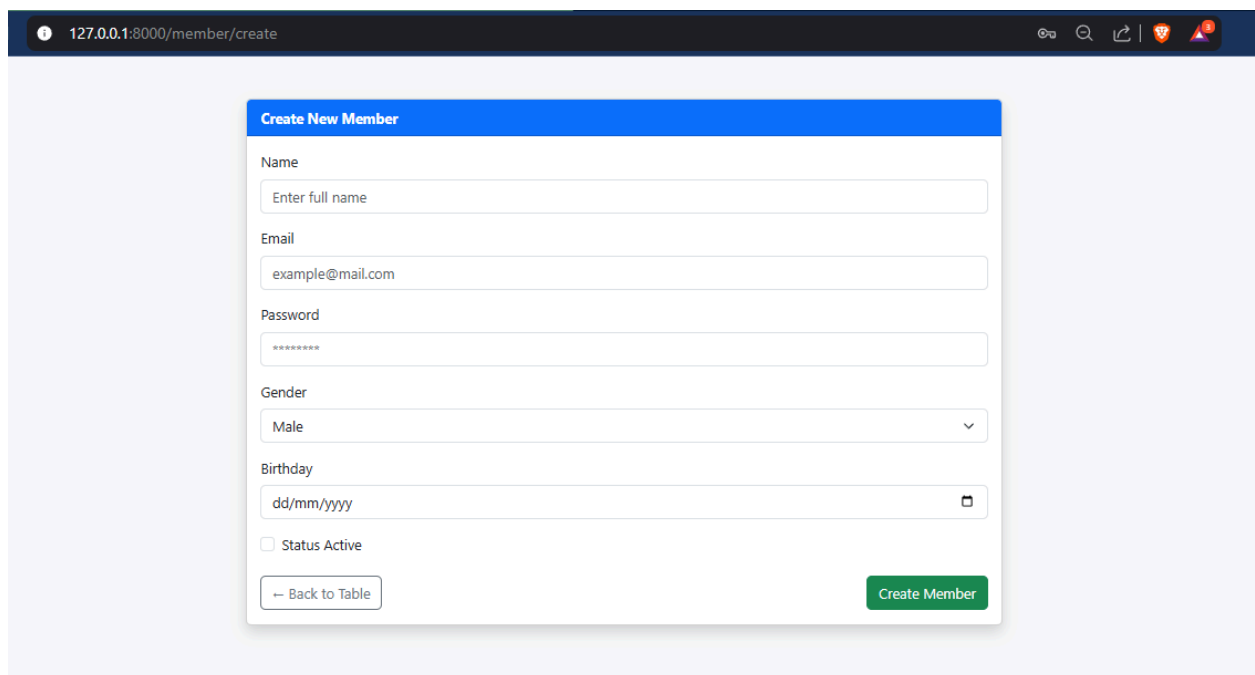
I also modified my routes.

```php
Route::get('/', function () {
    return view('member.create');
});

Route::get('/member/create', [MemberController::class, 'create'])->name('member.create');
Route::post('/member/store', [MemberController::class, 'store'])->name('member.store');
Route::get('/member/table', [MemberController::class, 'table'])->name('member.table');
Route::delete('/member/delete/{id}', [MemberController::class, 'delete'])->name('member.delete');
```

Below are the results and the user interface.

Member Management

Active Members

#	Name	Email	Gender	Birthday	Created At	Action
1	Ikhwan	lala@gmail.com	Male	15/01/2025	17 Apr 2025, 11:02 AM	Delete
2	Adam	Adam@gmail.com	Male	08/04/2025	17 Apr 2025, 11:02 AM	Delete
3	Aiman	Aiman@gmail.com	Male	17/02/2015	17 Apr 2025, 11:15 AM	Delete
4	Afwan	afwan@gmail.com	Male	27/12/1999	17 Apr 2025, 11:42 AM	Delete

Non-Active Members

#	Name	Email	Gender	Birthday	Created At	Action
1	Fatin	Fatin@gmail.com	Female	06/06/2000	17 Apr 2025, 11:01 AM	Delete
2	Zubaidah	baidah@gmail.com	Male	09/02/2021	17 Apr 2025, 11:03 AM	Delete
3	Udin	Udin@gmail.com	Male	04/02/2019	17 Apr 2025, 11:35 AM	Delete

Add New Member

Final Note

To be completely honest, I was unable to complete all aspects of the project due to time constraints and other commitments. This is the extent of what I was able to accomplish within the given timeframe. Nevertheless, I found this assessment to be an engaging and insightful experience.

I plan to continue updating and enhancing this project over time, potentially adding various new features and improvements.