**Database Migration and Seeding**

**Migration:**

Migrations are a type of version control for your database

1. First Create the database in localhost/phpmyadmin

path : localhost/phpmyadmin

click on database

Enter database name and click on create button

2. Go to the project folder

path : project\_name/config/database.php

**database.php**

Set default database:

'default' => env('DB\_CONNECTION', '**mysql**'),

or

'default' => 'mysql',

Set Connection of database:

'mysql' => [

'driver' => 'mysql',

'host' => 'localhost',

'database' => 'database\_name',

'username' => 'root',

'password' => '',

'charset' => 'utf8',

'collation' => 'utf8\_unicode\_ci',

'prefix' => '',

'strict' => false,

],

3. Go to the command prompt for creating a table

c:\>wamp\www\project\_name\php artisan make:migration create\_users\_table

4. File is created in migrations folder

Path : project\_name/database/migrations/filename.php

open the file

<?php

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Database\Migrations\Migration;

class CreateUserTable extends Migration

{

/\*\*

\* Run the migrations.

\*

\* @return void

\*/

public function up()

{

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

$table->increments('id');

$table->string('user\_name');

$table->string('password');

$table->timestamps();

});

}

/\*\*

\* Reverse the migrations.

\*

\* @return void

\*/

public function down()

{

Schema::drop('user');

}

}

5. Run the migration use these "php artisan migrate" command

Go to the Command prompt

c:/>wamp/www/project\_name/ php artisan migrate

|  |  |
| --- | --- |
| **Command** | **Description** |
| $table->bigIncrements('id'); | Incrementing ID using a "big integer" equivalent. |
| $table->bigInteger('votes'); | BIGINT equivalent for the database. |
| $table->binary('data'); | BLOB equivalent for the database. |
| $table->boolean('confirmed'); | BOOLEAN equivalent for the database. |
| $table->char('name', 4); | CHAR equivalent with a length. |
| $table->date('created\_at'); | DATE equivalent for the database. |
| $table->dateTime('created\_at'); | DATETIME equivalent for the database. |
| $table->decimal('amount', 5, 2); | DECIMAL equivalent with a precision and scale. |
| $table->double('column', 15, 8); | DOUBLE equivalent with precision, 15 digits in total and 8 after the decimal point. |
| $table->enum('choices', ['foo', 'bar']); | ENUM equivalent for the database. |
| $table->float('amount'); | FLOAT equivalent for the database. |
| $table->increments('id'); | Incrementing ID for the database (primary key). |
| $table->integer('votes'); | INTEGER equivalent for the database. |
| $table->json('options'); | JSON equivalent for the database. |
| $table->jsonb('options'); | JSONB equivalent for the database. |
| $table->longText('description'); | LONGTEXT equivalent for the database. |
| $table->mediumInteger('numbers'); | MEDIUMINT equivalent for the database. |
| $table->mediumText('description'); | MEDIUMTEXT equivalent for the database. |
| $table->morphs('taggable'); | Adds INTEGER taggable\_id and STRING taggable\_type. |
| $table->nullableTimestamps(); | Same as timestamps(), except allows NULLs. |
| $table->rememberToken(); | Adds remember\_token as VARCHAR(100) NULL. |
| $table->smallInteger('votes'); | SMALLINT equivalent for the database. |
| $table->softDeletes(); | Adds deleted\_at column for soft deletes. |
| $table->string('email'); | VARCHAR equivalent column. |
| $table->string('name', 100); | VARCHAR equivalent with a length. |
| $table->text('description'); | TEXT equivalent for the database. |
| $table->time('sunrise'); | TIME equivalent for the database. |
| $table->tinyInteger('numbers'); | TINYINT equivalent for the database. |
| $table->timestamp('added\_on'); | TIMESTAMP equivalent for the database. |
| $table->timestamps(); | Adds created\_at and updated\_at columns. |

**Seeding the Data into the table**

**Seeding: Seeding** is used for insert the data into the table

1. Open the command prompt and run these command ' php artisan make:seeder UserTableSeeder '

path : c:/>wamp/www/projectname/php artisan make:seeder UserTableSeeder

then create the file in seeds folder with the name 'UserTableSeeder'

Path : projectfolder/database/seeds/UserTableSeeder.php

2. write the code in UserTableSeeder.php

Path : projectfolder/database/seeds/UserTableSeeder.php

<?php

use Illuminate\Database\Seeder;

class UserTableSeeder extends Seeder

{

/\*\*

\* Run the database seeds.

\*

\* @return void

\*/

public function run()

{

DB::table('user')->insert([

'user\_name' => 'admin',

'password' => 'admin123',

]);

}

}

and call the file 'UserTableSeeder' in DatabaseSeeder file

Path: projectfolder/database/seeds/DatabaseSeeder.php

<?php

use Illuminate\Database\Seeder;

class DatabaseSeeder extends Seeder

{

/\*\*

\* Run the database seeds.

\*

\* @return void

\*/

public function run()

{

$this->call('UserTableSeeder');

}

}

3. Run the seeding open the command prompt

c:/>wamp/www/project\_name/php artisan db:seed