Laravel migration

Laravel Migrations are fundamental for managing database changes in Laravel applications. They offer an organized approach to synchronize your database with your application's codebase across various environments

What are laravel migration ?

Laravel Migration is a set of instructions that define the changes you want to make to your database schema. These changes can include creating new tables, altering existing tables, adding or modifying columns, and seeding the database with initial data

By encapsulating these changes within migration files, Laravel ensures that your database schema remains synchronized with your application's codebase, making it easier to manage database changes across different development environments and deployment stages.

Laravel allows you to carry out migrations without worrying about the specific database system you're using, whether it's MySQL, PostgreSQL, SQLite, or

others supported by Laravel. The framework abstracts away the database-specific syntax, making migrations both portable and adaptable to different database systems.

Why Would You Use Laravel Migrations?

	Explanation
Database Schema Management	Laravel Migrations help manage and version-control database schemas, allowing for structured and organized changes to the database structure over time.
Version Control	Migrations are tracked in version control systems like Git, ensuring that changes to the database can be easily reviewed, rolled back, or applied across development environments.
Database Portability	Laravel provides a database-agnostic approach, allowing you to write schema changes once and easily switch between different database systems, such as MySQL, PostgreSQL, SQLite, or SQL Server.
Rollback and Recovery	Migrations offer tools for rolling back or resetting database changes, enabling developers to recover from errors, adjust the schema, and maintain a stable database state.
Documentation	Migrations serve as documentation for the database schema evolution, with each migration file containing a timestamp and a description of the changes made, improving codebase transparency and maintainability.
History and Rollback Control	Laravel keeps track of executed migrations and offers rollback limitations, ensuring that migrations are idempotent and controlled in their application and rollback.
Seamless Integration with Testing	This integration allows developers to create test databases with the same schema as the application’s main database, making it easier to write and execute database-related tests

Codebase Consistency	By encapsulating database changes in migration files, you ensure that every developer working on the project can apply these changes uniformly across different environments, from local development setups to production servers.
Dependency Management	Dependency management feature simplifies the process of managing complex database changes that rely on the existence of certain structures.
Collaboration	Since the database schema is defined in code, multiple developers can work on it simultaneously, and any schema changes can be easily shared and merged using version control systems like Git. This streamlines the development process and reduces conflicts.

Create migration

```
1. php artisan make:migration create_example_table
```