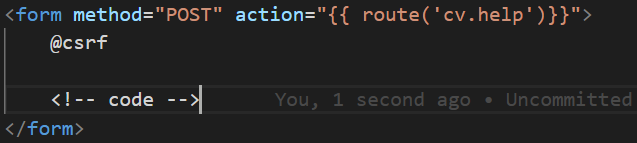
1. **What is CSRF protection in Laravel, and how does it work?**

* CSRF stands for Cross-Site Request Forgery. CSRF protection is used to protect from malicious attack that make the authorized user perform any unauthorized action on a website they are already logged into.

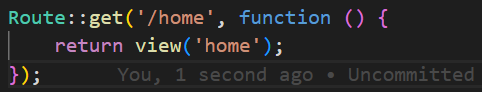
Laravel generates CSRF token for each user session that is active. This token verifies that the user making the request is an authenticated user in the application. This token is stored in user’s session that changes everytime the session is regenerated, malicious attacker cannot access it.



1. **What are routes in Laravel?**

* In Laravel, routes are used to define the URL patterns for your web application and map them to specific controller actions or views. When a user visits a URL, Laravel’s routing system determines which controller or action should handle that request.
* Routes are usually defined in routes/web.php file for web-based routes.
* Each route is associated with an HTTP request method (GET, POST, PUT, DELETE) and a URL pattern.
* Routes can be linked to controller methods, closures, or views.

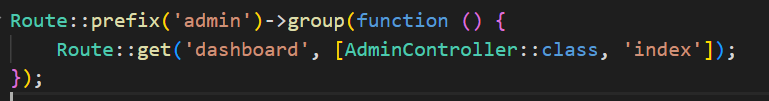
Basic route defining:



Route with controller:



Grouping routes with common namespace, middleware or prefix:

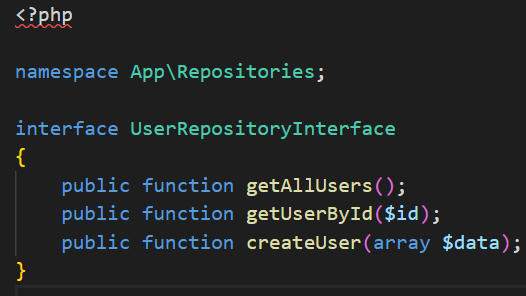


1. **What is the Repository Pattern, and how is it used in Laravel?**

* The Repository pattern is a design pattern used to abstract the data layer of an application. It helps in separating the logic that interacts with the database (such as fetching records, inserting, or updating data) from the rest of the application.
* To use Repository pattern we follow the following steps:

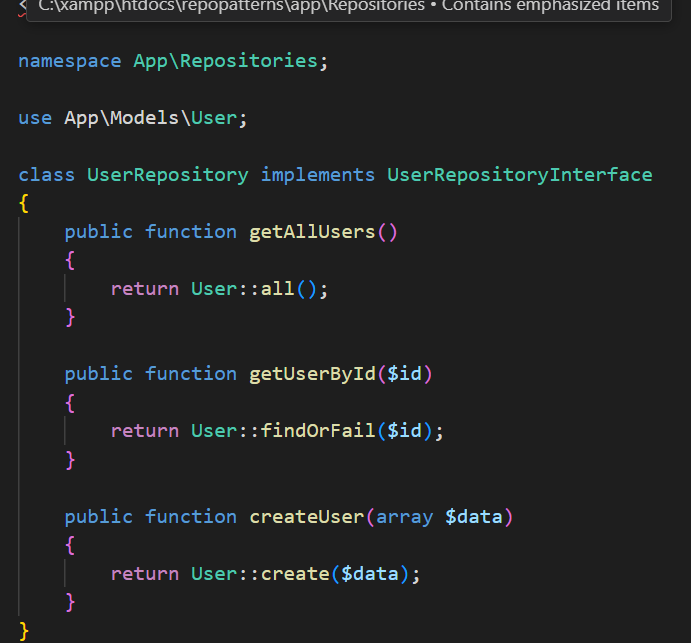
**1. Create a Repository Interface**: This defines the methods that your repository will implement.

Example: app/Repositories/UserRepositoryInterface.php



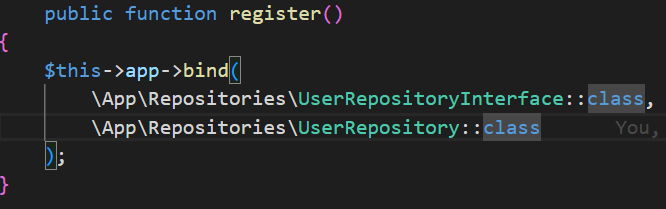
2. **Create the Repository Class**: This class implements the methods defined in the interface and handles the actual database interaction.

Example: app/Repositories/UserRepository.php



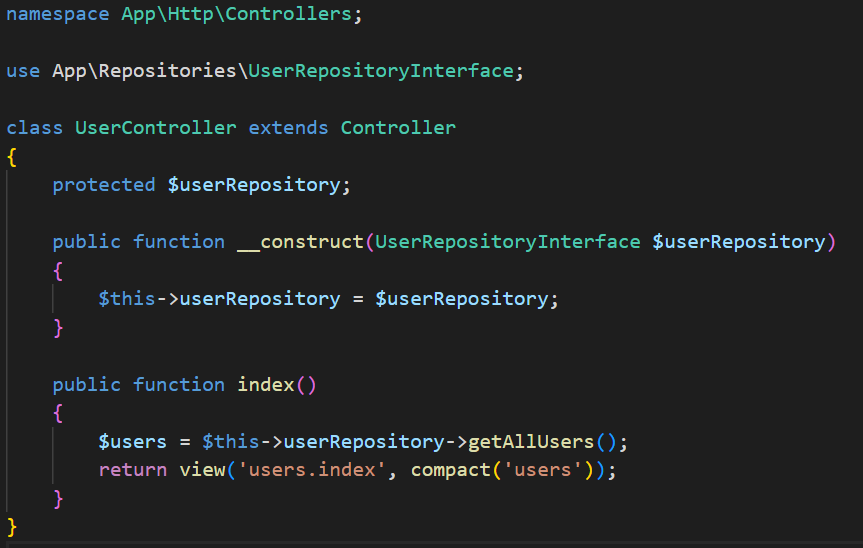
3. **Bind the Repository to the Interface**: You need to bind the repository class to its interface in a service provider (usually in AppServiceProvider).

Example: app/Providers/AppServiceProvider.php



4. **Use the Repository in Controllers**: Inject the repository interface into your controller, and use the repository methods instead of directly interacting with the model.

Example: app/Http/Controllers/UserController.php

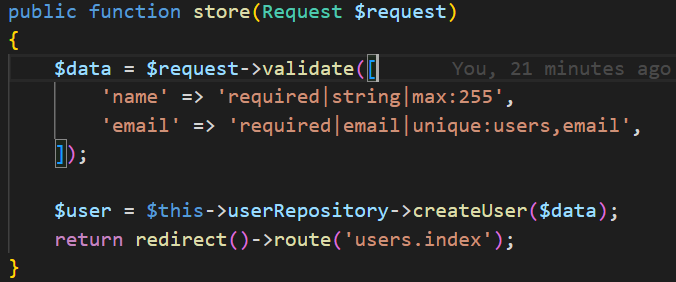


1. **How do you perform CRUD operations using the Repository Pattern in Laravel?**

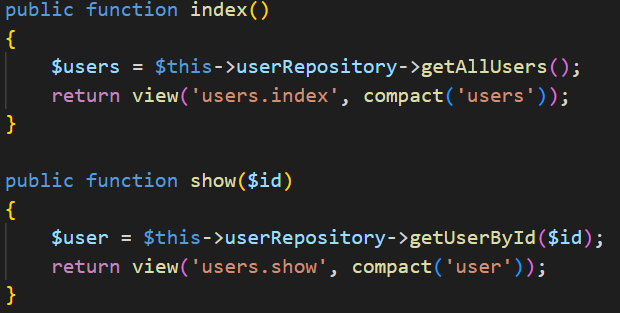
* Once you've set up the Repository Pattern as described above, performing **CRUD (Create, Read, Update, Delete)** operations becomes easy.

**Example:**

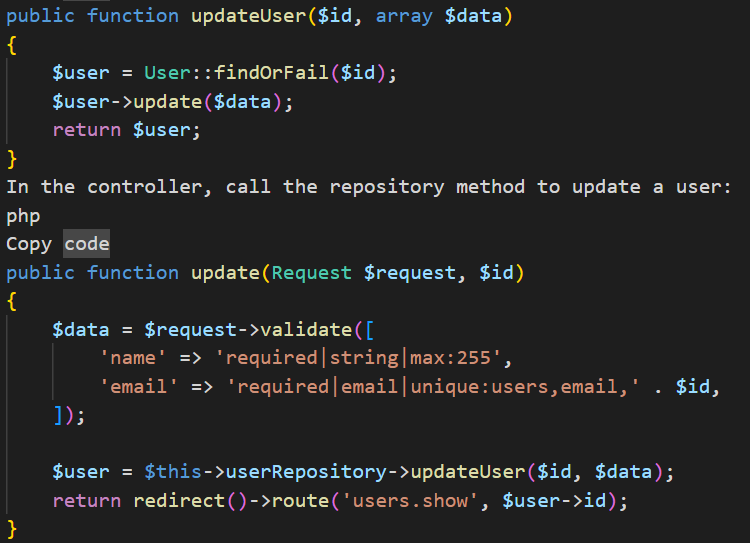
1. **Create**:
   * In your repository, implement the createUser() method (already shown above).
   * In the controller, call the repository's createUser() method to create a new user:



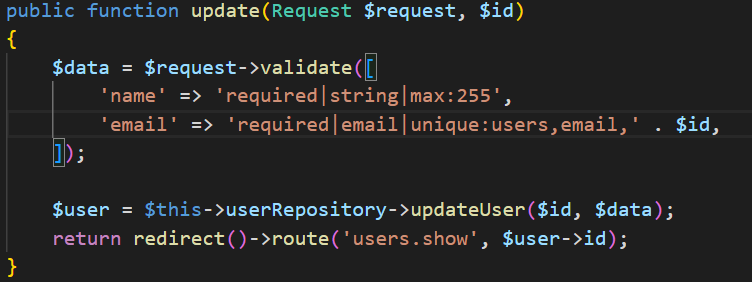
1. **Read** (Fetching Data):
   * The getAllUsers() method will retrieve all users.



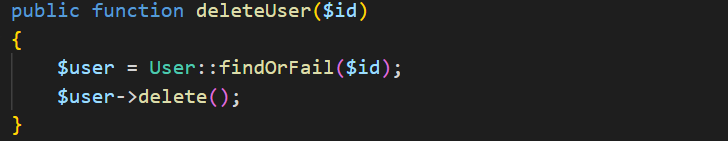
1. **Update**:
   * Create an updateUser() method in the repository.



In the controller, call the repository method to update a user:



1. **Delete**:
   * Add a deleteUser() method to the repository:

****

In the controller, call the deleteUser() method to delete a user:

