✅ 1. Missing Route Definitions for Controller

You've only defined routes for views (/ and /home), but you never routed any HTTP requests to your RegisterController methods, such as createUser, updateUser, etc.

✅ Fix:

Add these routes in your web.php (or api.php, if using API):

php

CopyEdit

use App\Http\Controllers\RegisterController;

Route::post('/user', [RegisterController::class, 'createUser']);

Route::put('/user/{id}', [RegisterController::class, 'updateUser']);

Route::get('/users', [RegisterController::class, 'allUsers']);

Route::delete('/user/{id}', [RegisterController::class, 'deleteuser']);

✅ 2. Incorrect createUser() and updateUser() method calls in Controller

You're calling these methods with no arguments:

php

CopyEdit

public function createUser()

{

return $this->registerService->createUser(); // ❌ Missing $request

}

But your service expects an array $user, and ultimately the repository expects a Request|array.

✅ Fix:

Inject the Request object:

php

CopyEdit

public function createUser(Request $request)

{

return $this->registerService->createUser($request);

}

public function updateUser($id, Request $request)

{

return $this->registerService->updateUser($id, $request);

}

✅ 3. Incorrect usage of findOrFail() in Repository

These lines are incorrect:

php

CopyEdit

return $this->registerModel->findOrFail($email); // ❌

You're trying to find by email or username using findOrFail, but it uses primary key (id). Instead, use where():

✅ Fix:

php

CopyEdit

public function getRegisteredUserByEmail(string $email)

{

return $this->registerModel->where('email', $email)->firstOrFail();

}

public function getRegisteredUserByUsername(string $username)

{

return $this->registerModel->where('name', $username)->firstOrFail();

}

✅ 4. deleteUser() method is empty in the Repository

You haven't implemented deletion logic:

php

CopyEdit

public function deleteUser($id)

{

// TODO: Implement deleteUser() method.

}

✅ Fix:

php

CopyEdit

public function deleteUser($id)

{

$user = $this->registerModel->findOrFail($id);

$user->delete();

return response()->json(['message' => 'User deleted successfully.']);

}

✅ 5. createUser() and updateUser() expect $user->input()

But $user might just be an array. This logic will break if not handled properly:

✅ Fix: Normalize input handling

In RegisterRepositoryImp, do something like this:

php

CopyEdit

public function createUser(Request|array $user)

{

$data = is\_array($user) ? $user : $user->all();

return $this->registerModel::create($data);

}

public function updateUser($id, Request|array $user)

{

$updateUser = $this->registerModel::findOrFail($id);

$data = is\_array($user) ? $user : $user->only(['name', 'email']);

$updateUser->fill($data);

$updateUser->save();

return $updateUser;

}

✅ 6. You Should Hash the Password Before Saving

Laravel does not hash passwords automatically. You must manually hash it:

✅ Fix:

In RegisterRepositoryImp@createUser:

php

CopyEdit

use Illuminate\Support\Facades\Hash;

$data['password'] = Hash::make($data['password']);

Add that before saving.

✅ 7. You Didn't Register the Bindings for Interfaces

If you don’t register RegisterRepositoryInterface with RegisterRepositoryImp in a service provider, Laravel won’t resolve the dependency.

✅ Fix:

In App\Providers\AppServiceProvider.php:

php

CopyEdit

use App\Repository\Contracts\RegisterRepositoryInterface;

use App\Repository\RegisterRepositoryImp;

use App\Services\Contracts\RegistrationServiceInterface;

use App\Services\RegistrationServiceImp;

public function register()

{

$this->app->bind(RegisterRepositoryInterface::class, RegisterRepositoryImp::class);

$this->app->bind(RegistrationServiceInterface::class, RegistrationServiceImp::class);

}

✅ 8. Blade Files Not Used

You said you have “4 blade files” but haven’t connected them. If you're aiming to render views, you'll need to call return view('filename'); from the controller methods.

✅ Summary of Fixes Needed

| Issue | Fix |
| --- | --- |
| Missing routes | Define routes for your controller |
| Wrong findOrFail() usage | Replace with where()->firstOrFail() |
| Incomplete method deleteUser() | Implement logic |
| Missing Request injection | Pass $request into controller methods |
| Missing binding | Bind interfaces in AppServiceProvider |
| Password hashing | Hash the password before saving |
| Blade integration | Add view rendering if needed |