

Dokumentasi Job Friends Web Application Final Assignment Web Programming

Nama : Albert Tanubarata

NIM : 2602091961

```
routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use App\Http\Controllers\AuthController;
5  use App\Http\Controllers\UserController;
6  use App\Http\Controllers\PaymentController;
7  use App\Http\Controllers\FriendController;
8  use App\Http\Controllers\NotificationController;
9
10 /*
11 |-----
12 | Web Routes
13 |-----
14 |
15 |
16 |
17 |
18 */
19
20 // Home Route
21 Route::get('/', [UserController::class, 'index'])->name('home');
22
23 // Authentication Routes
24 Route::get('/register', [AuthController::class, 'showRegistrationForm'])->name('register');
25 Route::post('/register', [AuthController::class, 'register']);
26 Route::get('/login', [AuthController::class, 'showLoginForm'])->name('login');
27 Route::post('/login', [AuthController::class, 'login']);
28 Route::post('/logout', [AuthController::class, 'logout'])->name('logout');
29
30 // User Profile Routes
31 Route::middleware(['auth'])->group(function () {
32     Route::get('/profile', [UserController::class, 'showProfile'])->name('profile');
33     Route::post('/profile/update', [UserController::class, 'updateProfile'])->name('profile.update');
34
35     // Payment Routes
36     Route::get('/payment', [PaymentController::class, 'showPaymentPage'])->name('payment');
37     Route::post('/payment/submit', [PaymentController::class, 'submitPayment'])->name('payment.submit');
38
39     // Friends Routes
40     Route::post('/friends/add/{id}', [FriendController::class, 'addFriend'])->name('friends.add');
41     Route::post('/friends/remove/{id}', [FriendController::class, 'removeFriend'])->name('friends.remove');
42
43     // Messaging/Notifications
44     Route::get('/notifications', [NotificationController::class, 'index'])->name('notifications');
45 });
46
47 // Search and Filtering
48 Route::get('/search', [UserController::class, 'search'])->name('search');
```

Route::get('/', [UserController::class, 'index'])->name('home'); berfungsi untuk menampilkan halaman utama dan Route::get('/search', [UserController::class, 'search'])->name('search');

Berfungsi untuk menyediakan fitur pencarian.

```

database > migrations > 2025_01_09_000000_create_users_table.php
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  class CreateUsersTable extends Migration
8  {
9      public function up()
10     {
11         Schema::create('users', function (Blueprint $table) {
12             $table->id();
13             $table->string('name');
14             $table->string('email')->unique();
15             $table->string('password');
16             $table->enum('gender', ['Male', 'Female']);
17             $table->json('fields_of_work');
18             $table->string('mobile_number');
19             $table->integer('registration_fee');
20             $table->integer('wallet_balance')->default(0);
21             $table->timestamps();
22         });
23     }
24
25     public function down()
26     {
27         Schema::dropIfExists('users');
28     }
29 }

```

Kodingan di atas berfungsi untuk membuat tabel users di database untuk menyimpan data pengguna.

app > Http > Controllers > AuthController.php

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\Models\User;
7  use Illuminate\Support\Facades\Hash;
8  use Illuminate\Support\Facades\Auth;
9
10 class AuthController extends Controller{
11     public function showRegistrationForm()
12     {
13         return view('auth.register');
14     }
15
16     public function register(Request $request)
17     {
18         $request->validate([
19             'name' => 'required|string|max:255',
20             'email' => 'required|string|email|max:255|unique:users',
21             'password' => 'required|string|min:8|confirmed',
22             'gender' => 'required|in:Male,Female',
23             'fields_of_work' => 'required|array|min:3',
24             'fields_of_work.*' => 'string',
25             'linkedin_username' => 'required|url|regex:/https:\/\/www\.linkedin\.com\/in\/.+\/',
26             'mobile_number' => 'required|digits_between:10,15',
27         ]);
28
29         $user = User::create([
30             'name' => $request->name,
31             'email' => $request->email,
32             'password' => Hash::make($request->password),
33             'gender' => $request->gender,
34             'fields_of_work' => json_encode($request->fields_of_work),
35             'linkedin_username' => $request->linkedin_username,
36             'mobile_number' => $request->mobile_number,
37             'registration_fee' => rand(100000, 125000),
38         ]);
39
40         Auth::login($user);
41
42         return redirect()->route('profile');
43     }
44
45     public function showLoginForm()
46     {
47         return view('auth.login');
48     }
49 }
```

```

49
50     public function login(Request $request)
51     {
52         $credentials = $request->only('email', 'password');
53
54         if (Auth::attempt($credentials)) {
55             return redirect()->intended('profile');
56         }
57
58         return back()->withErrors([
59             'email' => 'The provided credentials do not match our records.',
60         ]);
61     }
62
63     public function logout()
64     {
65         Auth::logout();
66
67         return redirect()->route('login');
68     }
69 }

```

Codingan di atas berfungsi untuk mengelola proses registrasi dan login pengguna.

showRegistrationForm() berfungsi untuk menampilkan halaman formulir registrasi.

- **showLoginForm()** berfungsi untuk Menampilkan halaman login.
- **login()** berfungsi untuk memproses login dan mengautentikasi pengguna.
- **logout()** berfungsi untuk memproses logout pengguna.

app > Http > Controllers > PaymentController.php

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\Models\User;
7  use Illuminate\Support\Facades\Hash;
8  use Illuminate\Support\Facades\Auth;
9
10 class PaymentController extends Controller{
11     public function showRegistrationForm()
12     {
13         return view('auth.register');
14     }
15
16     public function register(Request $request)
17     {
18         $request->validate([
19             'name' => 'required|string|max:255',
20             'email' => 'required|string|email|max:255|unique:users',
21             'password' => 'required|string|min:8|confirmed',
22             'gender' => 'required|in:Male,Female',
23             'fields_of_work' => 'required|array|min:3',
24             'fields_of_work.*' => 'string',
25             'linkedin_username' => 'required|url|regex:/https:\/\/www\.linkedin\.com\/in\/.+/ ',
26             'mobile_number' => 'required|digits_between:10,15',
27         ]);
28
29         $user = User::create([
30             'name' => $request->name,
31             'email' => $request->email,
32             'password' => Hash::make($request->password),
33             'gender' => $request->gender,
34             'fields_of_work' => json_encode($request->fields_of_work),
35             'linkedin_username' => $request->linkedin_username,
36             'mobile_number' => $request->mobile_number,
37             'registration_fee' => rand(100000, 125000),
38         ]);
39
40         Auth::login($user);
41
42         return redirect()->route('profile');
43     }
44
45     public function showLoginForm()
46     {
47         return view('auth.login');
48     }
49 }
```

```

49
50     public function login(Request $request)
51     {
52         $credentials = $request->only('email', 'password');
53
54         if (Auth::attempt($credentials)) {
55             return redirect()->intended('profile');
56         }
57
58         return back()->withErrors([
59             'email' => 'The provided credentials do not match our records.',
60         ]);
61     }
62
63     public function logout()
64     {
65         Auth::logout();
66
67         return redirect()->route('login');
68     }
69 }

```

showPaymentPage() berfungsi untuk menampilkan halaman pembayaran untuk pengguna.

submitPayment() berfungsi untuk memvalidasi jumlah pembayaran, jika kurang untuk membayar maka akan menampilkan pesan kesalahan dan jika ada kelebihan bayar maka akan menambahkan kelebihan pembayaran ke saldo dompet.

app > Http > Controllers > FriendController.php

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\Models\User;
7  use Illuminate\Support\Facades\Hash;
8  use Illuminate\Support\Facades\Auth;
9
10 class FriendController extends Controller{
11     public function showRegistrationForm()
12     {
13         return view('auth.register');
14     }
15
16     public function register(Request $request)
17     {
18         $request->validate([
19             'name' => 'required|string|max:255',
20             'email' => 'required|string|email|max:255|unique:users',
21             'password' => 'required|string|min:8|confirmed',
22             'gender' => 'required|in:Male,Female',
23             'fields_of_work' => 'required|array|min:3',
24             'fields_of_work.*' => 'string',
25             'linkedin_username' => 'required|url|regex:/https:\/\/www\.linkedin\.com\/in\/.+/ ',
26             'mobile_number' => 'required|digits_between:10,15',
27         ]);
28
29         $user = User::create([
30             'name' => $request->name,
31             'email' => $request->email,
32             'password' => Hash::make($request->password),
33             'gender' => $request->gender,
34             'fields_of_work' => json_encode($request->fields_of_work),
35             'linkedin_username' => $request->linkedin_username,
36             'mobile_number' => $request->mobile_number,
37             'registration_fee' => rand(100000, 125000),
38         ]);
39
40         Auth::login($user);
41
42         return redirect()->route('profile');
43     }
44
45     public function showLoginForm()
46     {
47         return view('auth.login');
48     }
49 }
```

```
49
50 public function login(Request $request)
51 {
52     $credentials = $request->only('email', 'password');
53
54     if (Auth::attempt($credentials)) {
55         return redirect()->intended('profile');
56     }
57
58     return back()->withErrors([
59         'email' => 'The provided credentials do not match our records.',
60     ]);
61 }
62
63 public function logout()
64 {
65     Auth::logout();
66
67     return redirect()->route('login');
68 }
69 }
```

Kodingan di atas berfungsi untuk :

addFriend(\$id): Menambahkan pengguna lain sebagai teman.

removeFriend(\$id): Menghapus pengguna dari daftar teman.

app > Http > Controllers > NotificationController.php

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\Models\User;
7  use Illuminate\Support\Facades\Hash;
8  use Illuminate\Support\Facades\Auth;
9
10 class NotificationController extends Controller{
11     public function showRegistrationForm()
12     {
13         return view('auth.register');
14     }
15
16     public function register(Request $request)
17     {
18         $request->validate([
19             'name' => 'required|string|max:255',
20             'email' => 'required|string|email|max:255|unique:users',
21             'password' => 'required|string|min:8|confirmed',
22             'gender' => 'required|in:Male,Female',
23             'fields_of_work' => 'required|array|min:3',
24             'fields_of_work.*' => 'string',
25             'linkedin_username' => 'required|url|regex:/https:\/\/www\.linkedin\.com\/in\/.+\/',
26             'mobile_number' => 'required|digits_between:10,15',
27         ]);
28
29         $user = User::create([
30             'name' => $request->name,
31             'email' => $request->email,
32             'password' => Hash::make($request->password),
33             'gender' => $request->gender,
34             'fields_of_work' => json_encode($request->fields_of_work),
35             'linkedin_username' => $request->linkedin_username,
36             'mobile_number' => $request->mobile_number,
37             'registration_fee' => rand(100000, 125000),
38         ]);
39
40         Auth::login($user);
41
42         return redirect()->route('profile');
43     }
44
45     public function showLoginForm()
46     {
47         return view('auth.login');
48     }
49 }
```

```

49
50     public function login(Request $request)
51     {
52         $credentials = $request->only('email', 'password');
53
54         if (Auth::attempt($credentials)) {
55             return redirect()->intended('profile');
56         }
57
58         return back()->withErrors([
59             'email' => 'The provided credentials do not match our records.',
60         ]);
61     }
62
63     public function logout()
64     {
65         Auth::logout();
66
67         return redirect()->route('login');
68     }
69 }

```

Kodingan index(): berfungsi untuk mengambil dan menampilkan daftar notifikasi untuk pengguna yang sedang login.

```

app > Models > User.php
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class User extends Model
9  {
10     use HasFactory;
11
12     protected $fillable = [
13         'name',
14         'email',
15         'password',
16         'gender',
17         'fields_of_work',
18         'linkedin_username',
19         'mobile_number',
20         'registration_fee',
21         'wallet_balance',
22     ];
23
24     public function friends()
25     {
26         return $this->belongsToMany(User::class, 'friends', 'user_id', 'friend_id');
27     }
28
29     public function notifications()
30     {
31         return $this->hasMany(Notification::class);
32     }
33 }

```

Codingan fillable berfungsi untuk menentukan kolom yang dapat diisi massal (mass-assignment). Lalu code friends() berfungsi untuk melakukan relasi many-to-many dengan tabel friends. Dan code notifications() berfungsi untuk melakukan relasi one-to-many dengan tabel notifications.