

1. Menu

- Buat database
- Buat koneksi database di file '.env' dengan mengisi informasi seperti 'DB_CONNECTION', 'DB_HOST', 'DB_PORT', 'DB_DATABASE', 'DB_USERNAME', DAN 'DB_PASSWORD'
- Buat model dan migrasi untuk produksi, categories, tags dan users
 - “ php artisan make:model Product -m
 - php artisan make:model Category -m
 - php artisan make:model Tag -m
 - php artisan make:model User -m “
- Buka masing-masing file migrasi (di 'database/migrations') dan definisikan struktur tabel untuk setiap entitas
- Jalankan migrasi untuk membuat tabel di database “ php artisan migrate “
- Buat controller untuk mengelola operasi terkait dengan setiap entitas
 - “ php artisan make:controller DashboardController
 - php artisan make:controller ProductController
 - php artisan make:controller CategoryController
 - php artisan make:controller TagController
 - php artisan make:controller UserController “
- Buka controller yang baru dibuat (di 'app/Http/Controllers') dan implementasikan logika bisnis untuk setiap operasi
- Tentukan rute di file ' web.php'
 - “ Route::get('/dashboard', 'DashboardController@index');
 - Route::resource('/products', 'ProductController');
 - Route::resource('/categories', 'CategoryController');
 - Route::resource('/tags', 'TagController');
 - Route::resource('/users', 'UserController');
- Buat tampilan untuk setiap bagian dari aplikasi, seperti halaman dashboard, produk, kategori, tag, dan pengguna
- Jalankan aplikasi laravelnya “ php artisan serve “

2. Inputan Produk

- Buat model migrasi untuk produk “ php artisan make:model Product -m “
Buka file migrasi yang baru dibuat (di ' database/migrations') dan tambahkan kolom yang diperlukan untuk tabel produk
- Jalankan perintah migrasi untuk membuat tabel produk di database “ php artisan migrate “
- Buat controller untuk mengelola operasi pada produk “ php artisan make:controller ProductController “
- Buka controller yang baru dibuat (di ' app/Http/Controllers ') dan implementasikan logika bisnis untuk operasi pada produk
 - “ // Contoh fungsi untuk menampilkan semua produk
 - public function index()
 - {
 - \$products = Product::all();
 - return view('products.index', compact('products'));
 - } “
- Buat tampilan untuk menampilkan produk, menambah produk baru, mengedit, dan menghapus. Letakkan file tampilan di direktori
 - “ ' resources/views/products ' “

- 'index.blade.php' untuk menampilkan semua produk
- 'create.blade.php' untuk formulir penambahan produk baru
- 'edit.blade.php' untuk formulir pengeditan produk
- Tentukan rute di file 'web.php' untuk setiap fungsi controller
"Route::resource('/products', 'ProductController');" "
- Jalankan aplikasi laravel "php artisan serve"

3. Inputan Kategori

- Buat model dan migrasi untuk kategori "php artisan make:model Category -m"
Buka file migrasi yang baru dibuat (di 'database/migrations') dan tambahkan kolom yang diperlukan untuk tabel kategori
- Jalankan perintah migrasi untuk membuat tabel kategori di database "php artisan migrate"
- Buat controller untuk mengelola operasi pada kategori "php artisan make:controller CategoryController"
- Buka controller yang baru dibuat (di 'app/Http/Controllers') dan implementasikan logika bisnis untuk operasi pada kategori
"// Contoh fungsi untuk menampilkan semua kategori
public function index()
{
 \$categories = Category::all();
 return view('categories.index', compact('categories'));
}
- Buat tampilan untuk menampilkan kategori, menambah kategori baru, mengedit, dan menghapus. Letakkan pada tampilan di direktori
"resources/views/categories"
• 'index.blade.php' untuk menampilkan semua kategori
• 'create.blade.php' untuk formulir penambahan kategori baru
• 'edit.blade.php' untuk formulir pengedit kategori
- Tentukan rute di file 'web.php' untuk setiap fungsi di controller
"Route::resource('/categories', 'CategoryController');" "
- Jalankan aplikasi laravel "php artisan serve"

4. Inputan Tag

- Buat model dan migrasi untuk tabel tag "php artisan make:model Tag -m"
- Buka file migrasi yang baru dibuat (di 'database/migrations') dan tambahkan kolom yang diperlukan
"public function up()
{
 Schema::create('tags', function (Blueprint \$table) {
 \$table->id();
 \$table->string('name');
 \$table->timestamps();
 });
}"
- Jalankan migrasi untuk membuat tabel tag di database "php artisan migrate"
- Buat controller untuk mengelola operasi terkait "php artisan make:controller TagController"
- Buka controller yang baru dibuat (di 'app/Http/Controllers/TagController.php') dan tambahkan logika untuk menambahkan tag

```
“ use App\Models\Tag;
```

```
public function addTag(Request $request)
```

```
{
```

```
    $tagName = $request->input('name');
```

```
    Tag::create(['name' => $tagName]);
```

```
    return response()->json(['message' => 'Tag added successfully']);
```

```
} “
```

- Tentukan rute untuk controller di file ‘ web.php ‘
“ Route::post('/add-tag', 'TagController@addTag'); “
- Buat formulir/ antarmuka pengguna untuk mengirimkan permintaan endpoint yang ditentukan pada langkah diatas
- Tambahkan validasi pada controller untuk memastikan data yang dimasukkan sesuai gunakan validator class
- Jalankan aplikasi laravel untuk melihat hasilnya “ php artisan serve “

Nama : Adinda Asri Azizah

Nim :20220801303

Fakultas Ilmu Komputer

Teknik Informatika