

Nama: Valeska Valentin Ekklesia

NIM: 2602093992

Dokumentasi EduFun

Di dalam project EduFun terdapat beberapa bagian:

-View: bagian untuk menampilkan tampilan dari website. Menggunakan Blade serta Bootstrap, dan beberapa custom css. Dan masing-masing dari view sendiri sudah mengimplementasikan sebuah layout template yang sudah dibuat

-Routing: proses yang menghubungkan URL (alamat web) yang diminta oleh pengguna dengan fungsi atau logika tertentu di aplikasi

-Controller: komponen yang berfungsi untuk menangani logika aplikasi saat menangani permintaan HTTP. Controller bertindak sebagai penghubung antara model (data) dan view (tampilan)

-Model: digunakan untuk berinteraksi dengan database

-Seeders: fitur yang memungkinkan kita untuk mengisi tabel database dengan data template secara otomatis

-Database: database yang digunakan pada project ini menggunakan MySQL dan dapat di download pada folder [MYSQL DATABASE] dengan nama edufun.sql

-View

1. navbar.blade.php

```
1 <style>
2     navbar-brand.active {
3         text-decoration: underline;
4         color: blue;
5     }
6 </style>
7
8 <nav class="navbar navbar-light shadow p-3 bg-body rounded mb-0">
9     
10
11     <div class="d-flex justify-content-end gap-3 me-3">
12         <a class="navbar-brand" href="{{ Request::is('/') ? 'active' : '' }}" href="/home">
13             <div class="dropdown">
14                 <a class="navbar-brand dropdown-toggle p-0 m-0 d-flex align-items-center" href="#" id="categoryDropdown" role="button" data-bs-toggle="dropdown" style="width: 100%; height: 100%;">
15                     Category
16                 </a>
17                 <div class="dropdown-menu">
18                     <li><a class="dropdown-item" href="{{ route('category.show', 1) }}">Software Engineering</a></li>
19                     <li><a class="dropdown-item" href="{{ route('category.show', 2) }}">Interactive Multimedia</a></li>
20                 </div>
21             </div>
22         <a class="navbar-brand" href="{{ Request::is('writers') ? 'active' : '' }}" href="/writers/writers">
23             <a class="navbar-brand" href="{{ Request::is('about-us') ? 'active' : '' }}" href="/about-us/about-us">
24             <a class="navbar-brand" href="{{ Request::is('popular') ? 'active' : '' }}" href="/popular/popular">
25         </div>
26 </nav>
```

2. footer.blade.php

```

1  <style>
2      footer {
3          width: 100%;
4          position: fixed;
5          bottom: 0;
6      }
7  </style>
8
9  <footer>
10     <div class="container-fluid bg-dark m-0 text-center d-flex align-items-center justify-content-center" style="height: 60px;">
11         <p class="text-white m-0 fw-5">© EduFun 2024 | Web Programming | Valeska Valentin Ekklesia | 2602093992</p>
12     </div>
13 </footer>

```

3. master.blade.php

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>EduFun</title>
7      <link rel="icon" type="image/x-icon" href="https://encrypted-tbn0.gstatic.com/images?q=tbn:AND9GcRktX6nTxzKjKFyxezMcl87x4w6ZtxYkbyt5g6s">
8      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
9
10     <style>
11         .custom-bg-body {
12             background-color: #f2f0f0;
13         }
14     </style>
15 </head>
16 <body class="custom-bg-body">
17     <div class="container-fluid p-0 custom-height">
18         @include('layout.navbar')
19
20         @yield('content')
21
22         @include('layout.footer')
23     </div>
24
25     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>
26 </body>
27 </html>

```

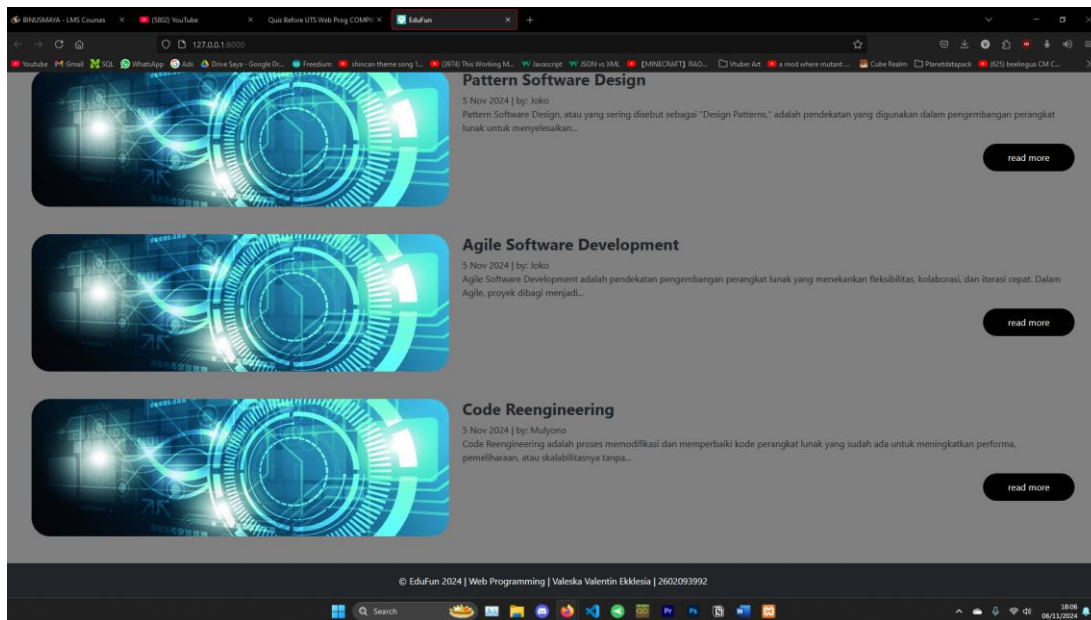
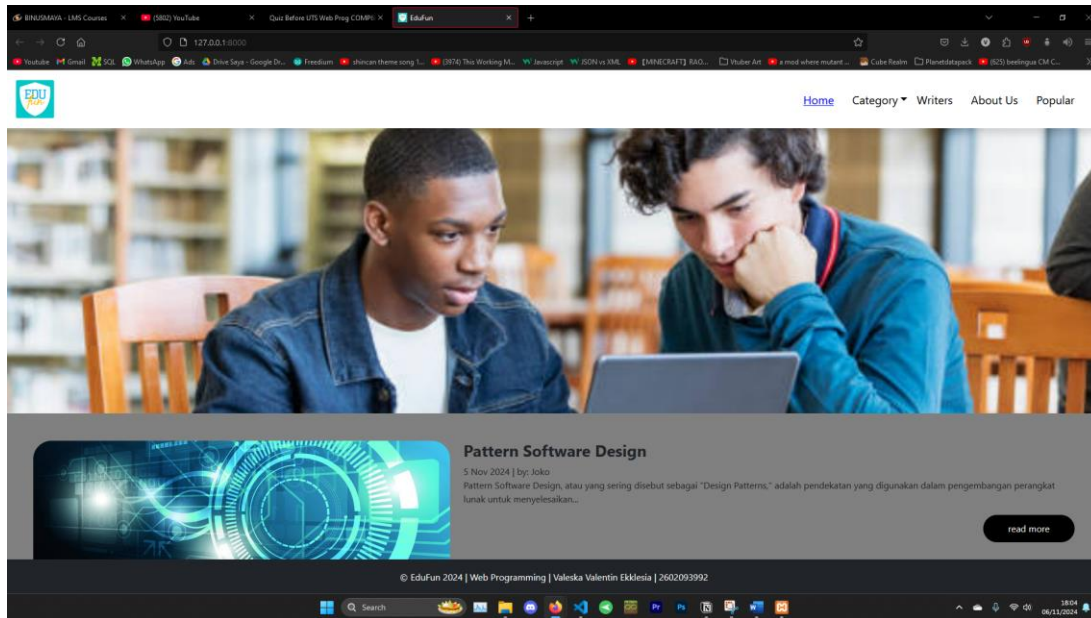
4. home.blade.php

```

1  @extends('layout.master')
2  @section('content')
3  <style>
4  .header-image {
5      width: 100%;
6      height: 500px;
7      overflow: hidden;
8  }
9
10 .header-image img {
11     width: 100%;
12     height: 100%;
13     object-fit: cover;
14 }
15
16 .custom-bg-1 {
17     background-color: gray;
18 }
19
20 .header-image-2 {
21     width: 40%;
22     height: 15rem;
23     flex-shrink: 0;
24 }
25
26 .header-image-2 img{
27     object-fit: cover;
28     border-radius: 2rem;
29 }
30
31 .custom-button {
32     height: 3rem;
33     width: 10rem;
34     border-radius: 3rem;
35 }
36
37 .custom-button-container {
38     width: 100%;
39 }
40
41 .custom-button p{
42     height: 100%;
43 }
44 </style>
45
46 <div class="header-image">
47 
48 </div>
49
50 <div class="d-flex flex-column container-fluid p-5 custom-bg-1 gap-5">
51 @foreach ($articles as $article)
52 <div class="d-flex flex-column">
53     <div class="d-flex">
54         <div class="header-image-2">
55             
56         </div>
57         <div class="d-flex flex-column ms-4" style="width:100%;">
58             <h3 class="fw-bold">{{ $article->title }}</h3>
59             <p class="m-0">{{ date('j M Y', strtotime($article->date_published)) }} | by: {{ explode(' ', $article->writer->name)[0] }}</p>
60             <p>{{ Str::words($article->content, 20) }}</p>
61             <div class="d-flex justify-content-end align-items-end custom-button-container">
62                 <a href="{{ route('articles.show', $article) }}" class="text-decoration-none">
63                     <div class="bg-black custom-button">
64                         <p class="d-flex justify-content-center align-items-center text-white">read more</p>
65                     </div>
66                 </a>
67             </div>
68         </div>
69     </div>
70 </div>
71 @endforeach
72 </div>
73
74 @endsection

```

Output home.blade.php



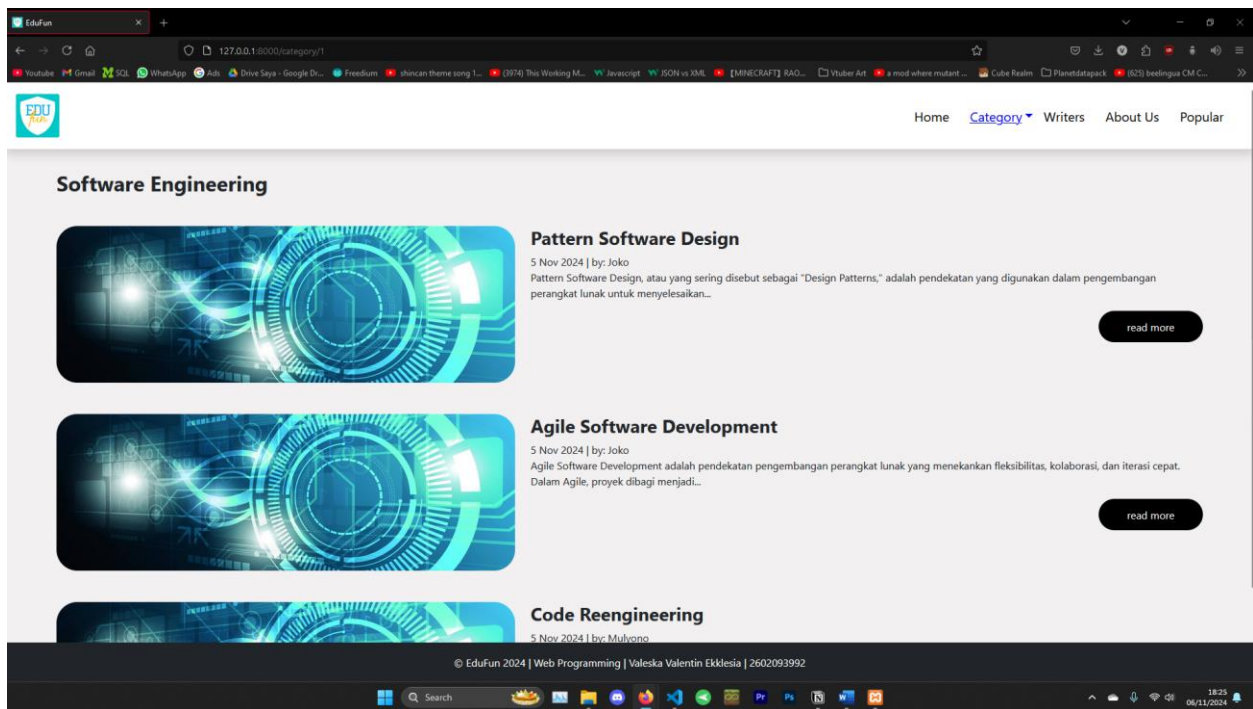
5. ./categories/show.blade.php

```

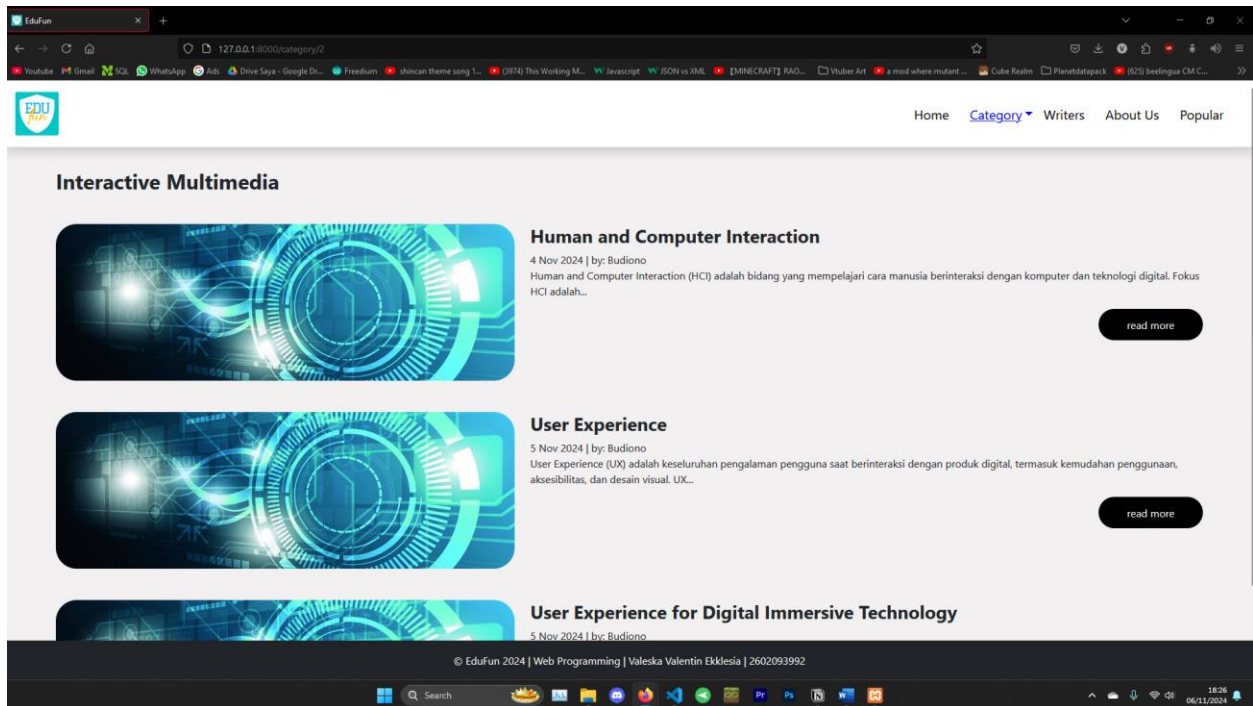
1  @extends('layout.master')
2
3  @section('content')
4  <style>
5      .custom-padding {
6          padding: 2rem 5rem;
7      }
8
9      .header-image {
10         width: 40%;
11         height: 15rem;
12         flex-shrink: 0;
13     }
14
15     .header-image img{
16         object-fit: cover;
17         border-radius: 2rem;
18     }
19
20     .custom-button {
21         height: 3rem;
22         width: 10rem;
23         border-radius: 3rem;
24     }
25
26     .custom-button-container {
27         width: 100%;
28     }
29
30     .custom-button p{
31         height: 100%;
32     }
33 </style>
34 <div class="custom-padding">
35     <h1 class="fw-bold fs-2">{{ $category->name }}</h1>
36     @foreach ($courses as $course)
37         @foreach ($course->articles as $article)
38             <div class="d-flex flex-column gap-3 mt-5">
39                 <div class="d-flex">
40                     <div class="header-image">
41                         
42                     </div>
43                     <div class="d-flex flex-column ms-4" style="width:100%;">
44                         <h3 class="fw-bold">{{ $article->title }}</h3>
45                         <p class="m-0">{{ date('j M Y', strtotime($article->date_published)) }} | by: {{ explode(' ', $article->writer->name)[0] }}</p>
46                         <p>{{ Str::words($article->content, 20) }}</p>
47                         <div class="d-flex justify-content-end align-items-end custom-button-container">
48                             <a href="{{ route('articles.show', $article) }}" class="text-decoration-none">
49                                 <div class="bg-black custom-button">
50                                     <p class="d-flex justify-content-center align-items-center text-white">read more</p>
51                                 </div>
52                             </a>
53                         </div>
54                     </div>
55                 </div>
56             @endforeach
57         @endforeach
58     @endforeach
59 </div>
60
61 @endsection

```

Output /categories/show.blade.php (Software Engineering category)



Output /categories/show.blade.php (Interactive Multimedia category)



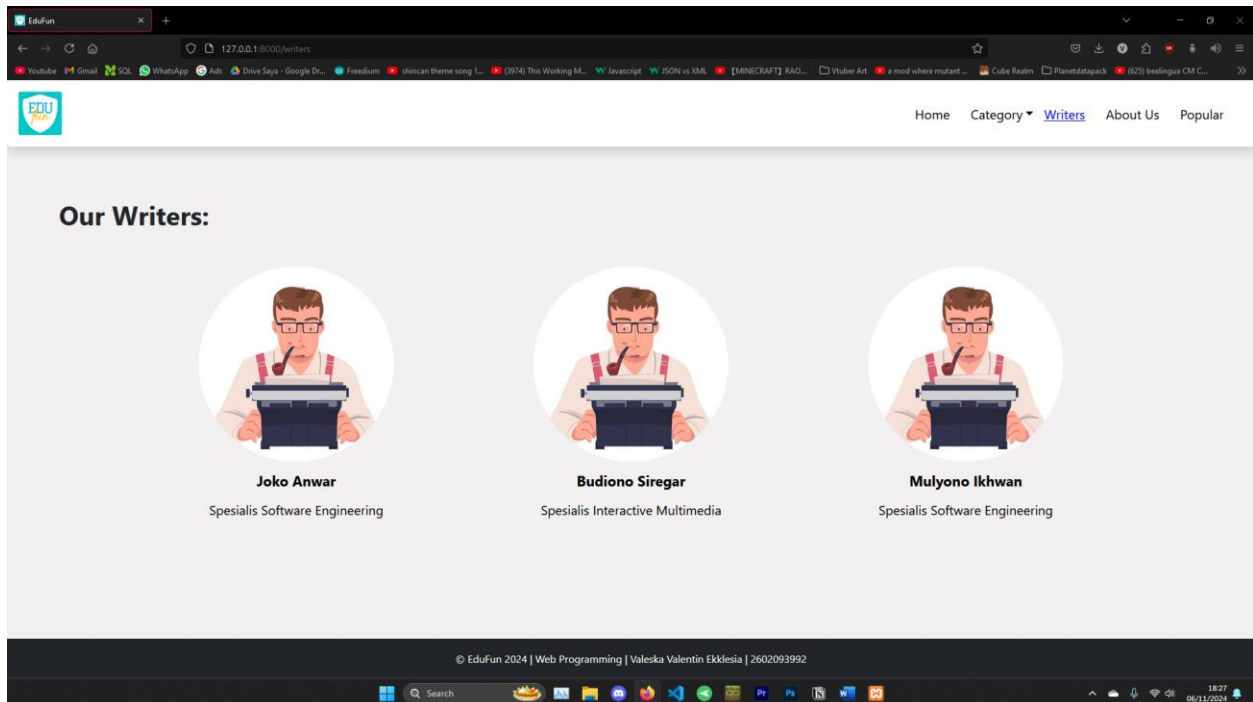
6. /writers/index.blade.php

```

1  @extends('layout.master')
2
3  @section('content')
4  <style>
5    .custom-padding {
6      padding: 5rem 5rem;
7    }
8
9    .custom-h2-writers {
10     margin-top: 1rem;
11     font-size: 22px;
12   }
13
14   .custom-p-writers {
15     margin-top: 1rem;
16     font-size: 20px;
17   }
18 </style>
19
20 <div class="custom-padding">
21   <h1 class="fw-bold">Our Writers:</h1>
22   <div class="pt-5 pb-5 d-flex justify-content-evenly">
23     @foreach ($writers as $writer)
24       <a href="{{ route('writer.show', $writer) }}" class="text-decoration-none color text-black">
25         <div class="text-center">
26           
27           <h2 class="custom-h2-writers fw-bold">{{ $writer->name }}</h2>
28           <p class="custom-p-writers">{{ $writer->bio }}</p>
29         </div>
30       </a>
31     @endforeach
32   </div>
33 </div>
34
35 @endsection

```

Output /writers/index.blade.php



7. /writers/show.blade.php

```

1  @extends('layout.master')
2
3  @section('content')
4  <style>
5      .custom-padding {
6          padding: 2rem 5rem;
7      }
8
9      .header-image {
10         width: 40%;
11         height: 15rem;
12         flex-shrink: 0;
13     }
14
15     .header-image img{
16         object-fit: cover;
17         border-radius: 2rem;
18     }
19
20     .custom-button {
21         height: 3rem;
22         width: 10rem;
23         border-radius: 3rem;
24     }
25
26     .custom-button-container {
27         min-width: 100%;
28     }
29
30     .custom-button p{
31         height: 100%;
32     }
33
34 </style>
35
36 <div class="custom-padding">
37     <div class="d-flex">
38         
39         <div class="d-flex flex-column justify-content-center ps-3">
40             <h3 class="fw-bold">{{ $writer->name }}</h3>
41             <p class="m-0">{{ $writer->bio }}</p>
42         </div>
43     </div>
44     @foreach ($articles as $article)
45         <div class="d-flex flex-column mt-5 gap-4">
46             <div class="d-flex">
47                 <div class="header-image ">
48                     
49                 </div>
50                 <div class="d-flex flex-column ms-4" style="width:100%;">
51                     <h3 class="fw-bold">{{ $article->title }}</h3>
52                     <p class="m-0">{{ date('j M Y', strtotime($article->date_published)) }} | by: {{ explode(' ', $article->writer->name)[0] }}</p>
53                     <p>{{ Str::words($article->content, 20) }}</p>
54                     <div class="d-flex justify-content-end align-items-end custom-button-container">
55                         <a href="{{ route('articles.show', $article) }}" class="text-decoration-none">
56                             <div class="bg-black custom-button">
57                                 <p class="d-flex justify-content-center align-items-center text-white">read more</p>
58                             </div>
59                         </a>
60                     </div>
61                 </div>
62             </div>
63         </div>
64     @endforeach
65 </div>
66
67 @endsection


```

Output /writers/show.blade.php


EduFun

127.0.0.1:8000/writers/1


HomeCategoryWritersAbout UsPopular



Joko Anwar
Spesialis Software Engineering



Pattern Software Design
5 Nov 2024 | by: Joko
Pattern Software Design, atau yang sering disebut sebagai "Design Patterns," adalah pendekatan yang digunakan dalam pengembangan perangkat lunak untuk menyelesaikan...
[read more](#)




Agile Software Development
5 Nov 2024 | by: Joko
Agile Software Development adalah pendekatan pengembangan perangkat lunak yang menekankan fleksibilitas, kolaborasi, dan iterasi cepat. Dalam Agile, proyek dibagi menjadi...
[read more](#)

© EduFun 2024 | Web Programming | Valeska Valentin Ekklesia | 2602093992


EduFun

127.0.0.1:8000/writers/2


HomeCategoryWritersAbout UsPopular



Budiono Siregar
Spesialis Interactive Multimedia

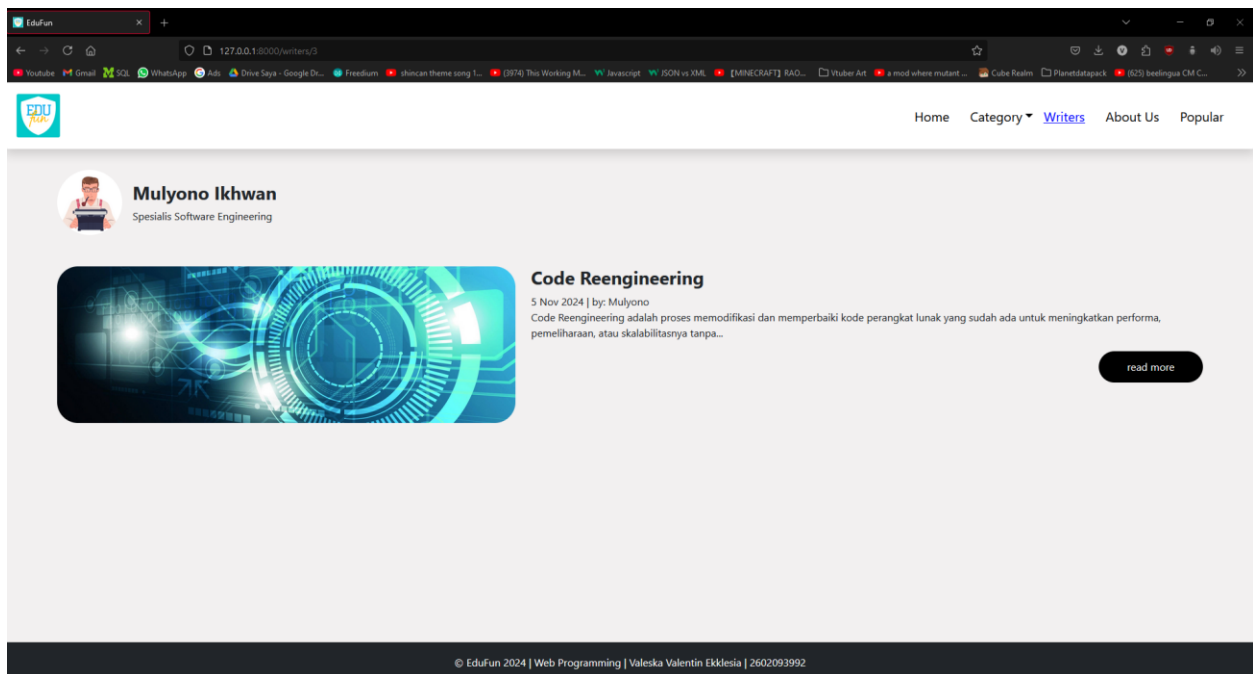


Human and Computer Interaction
4 Nov 2024 | by: Budiono
Human and Computer Interaction (HCI) adalah bidang yang mempelajari cara manusia berinteraksi dengan komputer dan teknologi digital. Fokus HCI adalah...
[read more](#)



User Experience
5 Nov 2024 | by: Budiono
User Experience (UX) adalah keseluruhan pengalaman pengguna saat berinteraksi dengan produk digital, termasuk kemudahan penggunaan, aksesibilitas, dan desain visual. UX...
[read more](#)

© EduFun 2024 | Web Programming | Valeska Valentin Ekklesia | 2602093992



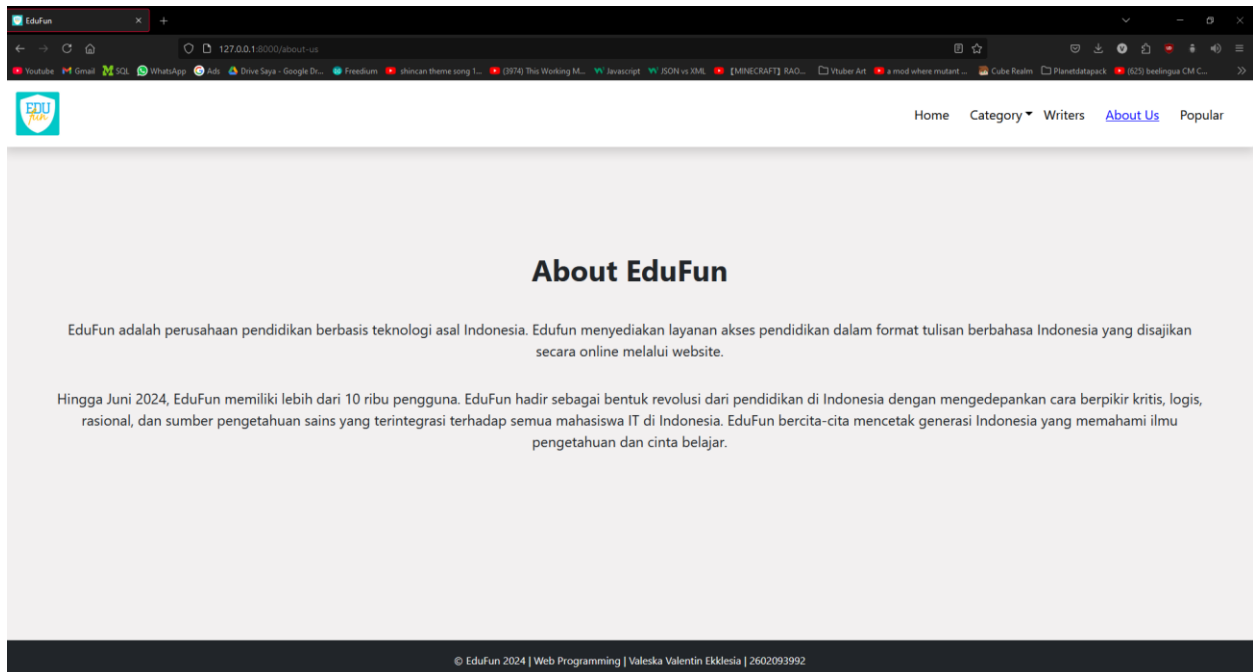
8. about.blade.php

```

1  @extends('layout.master')
2
3  @section('content')
4  <style>
5      .custom-padding {
6          padding: 10rem 5rem;
7      }
8
9      .custom-p {
10         font-size: 22px;
11     }
12
13     .custom-h1 {
14         font-size: 45px;
15     }
16 </style>
17
18 <div class="mx-auto text-center custom-padding">
19     <h1 class="fw-bold custom-h1">About EduFun</h1>
20     <div class="mt-5">
21         <p class="custom-p">EduFun adalah perusahaan pendidikan berbasis teknologi asal Indonesia. EduFun menyediakan layanan akses pendidikan dalam format
22         tulisan berbahasa Indonesia yang disajikan secara online melalui website.
23         </p><br>
24         <p class="custom-p">Hingga Juni 2024, EduFun memiliki lebih dari 10 ribu pengguna. EduFun hadir sebagai bentuk revolusi dari pendidikan di Indonesia
25         dengan mengedepankan cara berpikir kritis, logis, rasional, dan sumber pengetahuan sains yang terintegrasi terhadap semua
26         mahasiswa IT di Indonesia. EduFun bercita-cita mencetak generasi Indonesia yang memahami ilmu pengetahuan dan cinta belajar.
27         </p>
28     </div>
29 </div>
30
31 @endsection

```

Output about.blade.php



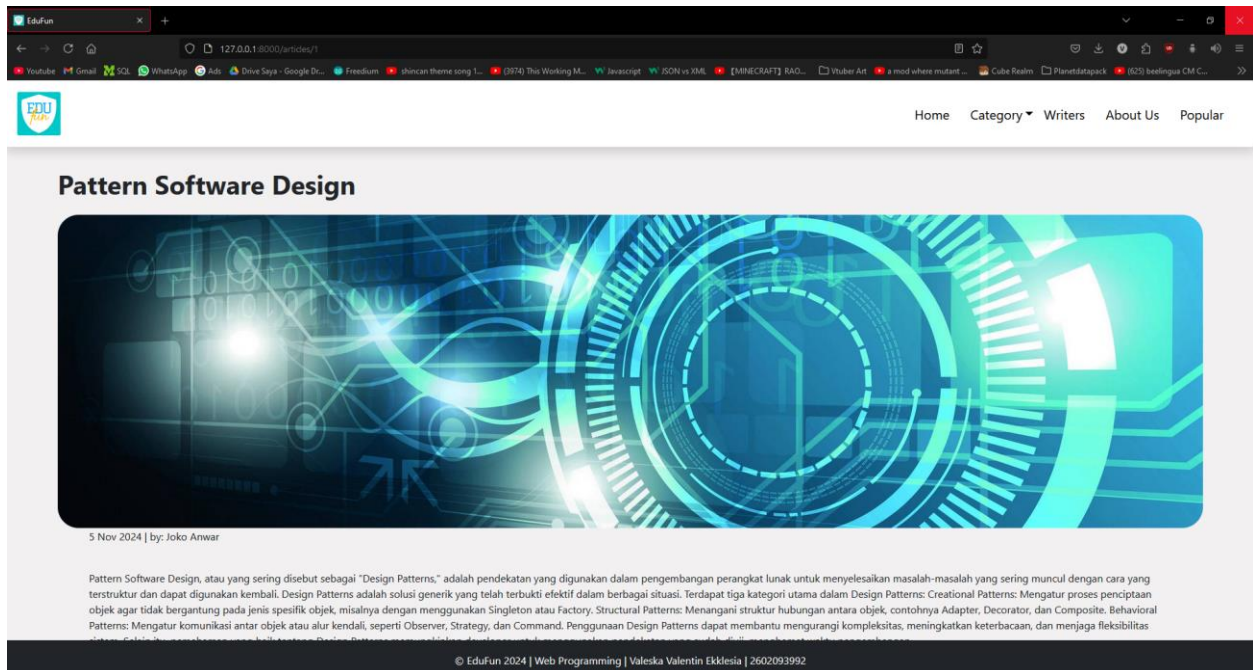
9. /articles/show.blade.php

```

1  @extends('layout.master')
2
3  @section('content')
4  <style>
5      .custom-padding {
6          padding: 2rem 5rem;
7      }
8
9      .header-image {
10         width: 100%;
11         height: 30rem;
12         flex-shrink: 0;
13     }
14
15     .header-image img{
16         object-fit: cover;
17         border-radius: 2rem;
18     }
19 </style>
20
21 <div class="d-flex flex-column custom-padding">
22     <h1 class="fw-bold">{{ $article->title }}</h1>
23     <div>
24         <div class="header-image mt-3">
25             
26         </div>
27         <div class="article-content mx-5">
28             <p>{{ date('j M Y', strtotime($article->date_published)) }} | by: {{ $article->writer->name }}</p><br>
29             <p>{{ $article->content }}</p>
30         </div>
31     </div>
32 </div>
33
34 @endsection

```

Output /articles/show.blade.php



-Routing:

web.php

```
1 <?php
2
3 use App\Http\Controllers\CategoryController;
4 use App\Http\Controllers\DetailController;
5 use App\Http\Controllers\HomeController;
6 use App\Http\Controllers\WritersController;
7 use Illuminate\Support\Facades\Route;
8
9 Route::get('/', [HomeController::class, 'index'])->name('home');
10
11 Route::get('/category/{category}', [CategoryController::class, 'show'])->name('category.show');
12
13 Route::get('/writers', [WritersController::class, 'index'])->name('writers')
14 ;
15 Route::get('/writers/{writer}', [WritersController::class, 'show'])->name('writer.show');
16
17 Route::get('/about-us', function() {
18     return view('pages.about');
19 });
20
21 Route::get('articles/{article}', [DetailController::class, 'show'])->name('articles.show');
22
23 Route::get('/popular', function() {
24     return view('pages.popular');
25 });
```

-Controller:

1. CategoryController.php

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\Category;
6  use Illuminate\Http\Request;
7
8  class CategoryController extends Controller
9  {
10     public function index()
11     {
12         $categories = Category::all();
13         return view('pages.categories.index', compact('categories'));
14     }
15
16     public function show(Category $category)
17     {
18         $courses = $category->courses()->with('articles')->get();
19         return view('pages.categories.show', compact('category', 'courses'));
20     }
21 }
22
```

2. DetailController.php

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\Article;
6  use Illuminate\Http\Request;
7
8  class DetailController extends Controller
9  {
10     public function show(Article $article)
11     {
12         return view('pages.articles.show', compact('article'));
13     }
14 }
15
```

3. HomeController.php



```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\Article;
6  use Illuminate\Http\Request;
7
8  class HomeController extends Controller
9  {
10     public function index()
11     {
12         $articles = Article::latest()->limit(4)->get();
13         return view('pages.home', compact('articles'));
14     }
15 }
16

```

4. WritersController.php



```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\Writer;
6  use Illuminate\Http\Request;
7
8  class WritersController extends Controller
9  {
10
11     public function index()
12     {
13         $writers = Writer::all();
14         return view('pages.writers.index', compact('writers'));
15     }
16     public function show(Writer $writer)
17     {
18         $articles = $writer->articles()->get();
19         return view('pages.writers.show', compact('writer', 'articles'));
20     }
21 }
22

```

-Model:

1. Article.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 Article extends Model
9  {
10     use HasFactory;
11     public function writer()
12     {
13         return $this->belongsTo(Writer::class);
14     }
15
16     public function course()
17     {
18         return $this->belongsTo(Course::class);
19     }
20 }
21
```

2. Category.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 Category extends Model
9  {
10     use HasFactory;
11     public function courses()
12     {
13         return $this->hasMany(Course::class);
14     }
15 }
16
```

3. Course.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 Course extends Model
9  {
10     use HasFactory;
11     public function articles()
12     {
13         return $this->hasMany(Article::class);
14     }
15
16     public function category()
17     {
18         return $this->hasMany(Category::class);
19     }
20 }
21
22

```

4. Writer.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 Writer extends Model
9  {
10     use HasFactory;
11
12     public function articles()
13     {
14         return $this->hasMany(Article::class);
15     }
16 }
17

```

-Migration:

1. 2024_11_05_062849_create_writers_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  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12     public function up(): void
13     {
14         Schema::create('writers', function (Blueprint $table) {
15             $table->id();
16             $table->string('name');
17             $table->string('bio');
18             $table->timestamps();
19         });
20     }
21
22     /**
23      * Reverse the migrations.
24      */
25     public function down(): void
26     {
27         Schema::dropIfExists('writers');
28     }
29 };
30
```

2. 2024_11_05_084116_create_categories_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  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12     public function up(): void
13     {
14         Schema::create('categories', function (Blueprint $table) {
15             $table->id();
16             $table->string('name');
17             $table->timestamps();
18         });
19     }
20
21     /**
22      * Reverse the migrations.
23      */
24     public function down(): void
25     {
26         Schema::dropIfExists('categories');
27     }
28 };
29

```

3. 2024_11_05_084150_create_courses_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  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12     public function up(): void
13     {
14         Schema::create('courses', function (Blueprint $table) {
15             $table->id();
16             $table->foreignId('category_id')->constrained()->cascadeOnDelete();
17             $table->string('name');
18             $table->timestamps();
19         });
20     }
21
22     /**
23      * Reverse the migrations.
24      */
25     public function down(): void
26     {
27         Schema::dropIfExists('courses');
28     }
29 };
30

```

4. 2024_11_05_084230_create_articles_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  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12       public function up(): void
13       {
14           Schema::create('articles', function (Blueprint $table) {
15               $table->id();
16               $table->string('title');
17               $table->text('content');
18               $table->date('date_published');
19               $table->foreignId('course_id')->constrained()->cascadeOnDelete();
20               $table->foreignId('writer_id')->constrained()->cascadeOnDelete();
21               $table->timestamps();
22           });
23       }
24
25       /**
26       * Reverse the migrations.
27       */
28       public function down(): void
29       {
30           Schema::dropIfExists('articles');
31       }
32   };
33

```

-Seeders:

DatabaseSeeder.php

```

1  1. Introduction
2  1.1. Background
3  1.2. Objectives
4  1.3. Scope
5  1.4. Limitations
6  1.5. Significance
7  1.6. Methodology
8  1.7. Organization of the Report
9  2. Literature Review
10 2.1. Conceptual Framework
11 2.2. Theoretical Background
12 2.3. Empirical Studies
13 2.4. Research Gaps
14 3. Methodology
15 3.1. Research Design
16 3.2. Sampling Method
17 3.3. Data Collection
18 3.4. Data Analysis
19 4. Results and Discussion
20 4.1. Descriptive Statistics
21 4.2. Inferential Statistics
22 4.3. Regression Analysis
23 4.4. Correlation Analysis
24 4.5. Factorial Analysis
25 4.6. Cluster Analysis
26 4.7. Discriminant Analysis
27 4.8. Principal Component Analysis
28 4.9. Structural Equation Modeling
29 4.10. Path Analysis
30 4.11. Bayesian Networks
31 4.12. Decision Trees
32 4.13. Support Vector Machines
33 4.14. Artificial Neural Networks
34 4.15. Genetic Algorithms
35 4.16. Ant Colony Optimization
36 4.17. Particle Swarm Optimization
37 4.18. Simulated Annealing
38 4.19. Tabu Search
39 4.20. Genetic Programming
40 4.21. Evolutionary Algorithms
41 4.22. Memetic Algorithms
42 4.23. Hybrid Algorithms
43 4.24. Metaheuristic Algorithms
44 4.25. Stochastic Algorithms
45 4.26. Heuristic Algorithms
46 4.27. Local Search Algorithms
47 4.28. Global Search Algorithms
48 4.29. Metaheuristic Search Algorithms
49 4.30. Evolutionary Search Algorithms
50 4.31. Genetic Search Algorithms
51 4.32. Artificial Neural Network Search Algorithms
52 4.33. Support Vector Machine Search Algorithms
53 4.34. Decision Tree Search Algorithms
54 4.35. Bayesian Network Search Algorithms
55 4.36. Cluster Analysis Search Algorithms
56 4.37. Factorial Analysis Search Algorithms
57 4.38. Correlation Analysis Search Algorithms
58 4.39. Regression Analysis Search Algorithms
59 4.40. Inferential Statistics Search Algorithms
60 4.41. Descriptive Statistics Search Algorithms
61 4.42. Conceptual Framework Search Algorithms
62 4.43. Theoretical Background Search Algorithms
63 4.44. Empirical Studies Search Algorithms
64 4.45. Research Gaps Search Algorithms
65 4.46. Methodology Search Algorithms
66 4.47. Research Design Search Algorithms
67 4.48. Sampling Method Search Algorithms
68 4.49. Data Collection Search Algorithms
69 4.50. Data Analysis Search Algorithms
70 4.51. Results and Discussion Search Algorithms
71 4.52. Descriptive Statistics Search Algorithms
72 4.53. Inferential Statistics Search Algorithms
73 4.54. Regression Analysis Search Algorithms
74 4.55. Correlation Analysis Search Algorithms
75 4.56. Factorial Analysis Search Algorithms
76 4.57. Cluster Analysis Search Algorithms
77 4.58. Discriminant Analysis Search Algorithms
78 4.59. Principal Component Analysis Search Algorithms
79 4.60. Structural Equation Modeling Search Algorithms
80 4.61. Path Analysis Search Algorithms
81 4.62. Bayesian Networks Search Algorithms
82 4.63. Decision Trees Search Algorithms
83 4.64. Support Vector Machines Search Algorithms
84 4.65. Artificial Neural Networks Search Algorithms
85 4.66. Genetic Algorithms Search Algorithms
86 4.67. Ant Colony Optimization Search Algorithms
87 4.68. Particle Swarm Optimization Search Algorithms
88 4.69. Simulated Annealing Search Algorithms
89 4.70. Tabu Search Search Algorithms
90 4.71. Genetic Programming Search Algorithms
91 4.72. Evolutionary Algorithms Search Algorithms
92 4.73. Memetic Algorithms Search Algorithms
93 4.74. Hybrid Algorithms Search Algorithms
94 4.75. Metaheuristic Algorithms Search Algorithms
95 4.76. Stochastic Algorithms Search Algorithms
96 4.77. Heuristic Algorithms Search Algorithms
97 4.78. Local Search Algorithms Search Algorithms
98 4.79. Global Search Algorithms Search Algorithms
99 4.80. Metaheuristic Search Algorithms Search Algorithms
100 4.81. Evolutionary Search Algorithms Search Algorithms
101 4.82. Genetic Search Algorithms Search Algorithms
102 4.83. Artificial Neural Network Search Algorithms Search Algorithms
103 4.84. Support Vector Machine Search Algorithms Search Algorithms
104 4.85. Decision Tree Search Algorithms Search Algorithms
105 4.86. Bayesian Network Search Algorithms Search Algorithms
106 4.87. Cluster Analysis Search Algorithms Search Algorithms
107 4.88. Factorial Analysis Search Algorithms Search Algorithms
108 4.89. Correlation Analysis Search Algorithms Search Algorithms
109 4.90. Regression Analysis Search Algorithms Search Algorithms
110 4.91. Inferential Statistics Search Algorithms Search Algorithms
111 4.92. Descriptive Statistics Search Algorithms Search Algorithms
112 4.93. Conceptual Framework Search Algorithms Search Algorithms
113 4.94. Theoretical Background Search Algorithms Search Algorithms
114 4.95. Empirical Studies Search Algorithms Search Algorithms
115 4.96. Research Gaps Search Algorithms Search Algorithms
116 4.97. Methodology Search Algorithms Search Algorithms
117 4.98. Research Design Search Algorithms Search Algorithms
118 4.99. Sampling Method Search Algorithms Search Algorithms
119 5. Conclusion
120 5.1. Summary
121 5.2. Implications
122 5.3. Recommendations
123 5.4. Limitations
124 5.5. Future Research
125 6. References
126 6.1. Books
127 6.2. Journals
128 6.3. Conferences
129 6.4. Theses
130 6.5. Online Sources
131 6.6. Other Sources
132 7. Appendix
133 7.1. Tables
134 7.2. Figures
135 7.3. Charts
136 7.4. Maps
137 7.5. Other Figures
138 8. Glossary
139 8.1. Terms
140 8.2. Definitions
141 8.3. Explanations
142 8.4. Examples
143 8.5. Other Terms
144 9. Index
145 9.1. Subjects
146 9.2. Names
147 9.3. Numbers
148 9.4. Other Symbols
149 10. Abstract
150 10.1. Background
151 10.2. Objectives
152 10.3. Methodology
153 10.4. Results
154 10.5. Conclusion
155 10.6. Keywords
156 10.7. Summary
157 10.8. Implications
158 10.9. Recommendations
159 10.10. Limitations
160 10.11. Future Research
161 10.12. References
162 10.13. Books
163 10.14. Journals
164 10.15. Conferences
165 10.16. Theses
166 10.17. Online Sources
167 10.18. Other Sources
168 10.19. Appendix
169 10.20. Tables
170 10.21. Figures
171 10.22. Charts
172 10.23. Maps
173 10.24. Other Figures
174 10.25. Glossary
175 10.26. Terms
176 10.27. Definitions
177 10.28. Explanations
178 10.29. Examples
179 10.30. Other Terms
180 10.31. Index
181 10.32. Subjects
182 10.33. Names
183 10.34. Numbers
184 10.35. Other Symbols
185 10.36. Abstract
186 10.37. Background
187 10.38. Objectives
188 10.39. Methodology
189 10.40. Results
190 10.41. Conclusion
191 10.42. Keywords
192 10.43. Summary
193 10.44. Implications
194 10.45. Recommendations
195 10.46. Limitations
196 10.47. Future Research
197 10.48. References
198 10.49. Books
199 10.50. Journals
200 10.51. Conferences
201 10.52. Theses
202 10.53. Online Sources
203 10.54. Other Sources
204 10.55. Appendix
205 10.56. Tables
206 10.57. Figures
207 10.58. Charts
208 10.59. Maps
209 10.60. Other Figures
210 10.61. Glossary
211 10.62. Terms
212 10.63. Definitions
213 10.64. Explanations
214 10.65. Examples
215 10.66. Other Terms
216 10.67. Index
217 10.68. Subjects
218 10.69. Names
219 10.70. Numbers
220 10.71. Other Symbols
221 10.72. Abstract
222 10.73. <
```

-Database (MySQL):

