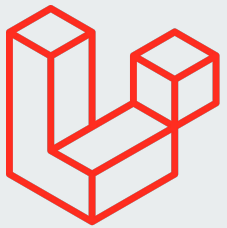
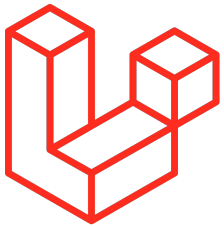
Dokumentasi Laravel

M Ikhsan Adriansyah



Tech Stack




tailwindcss


Chart.js


sweetalert2

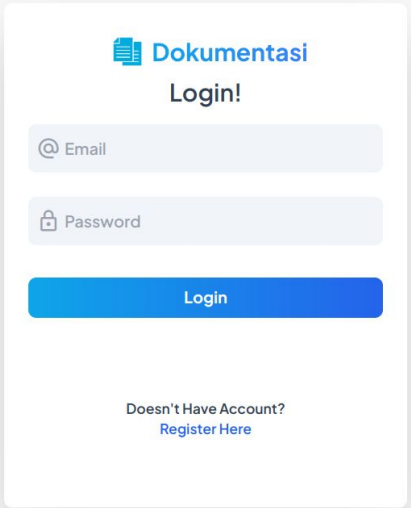

jQuery
write less, do more.

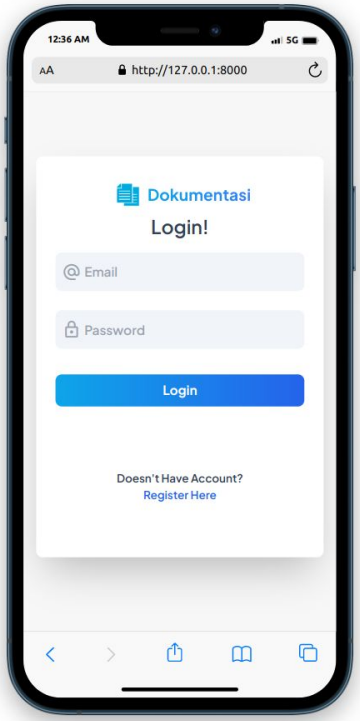

MySQL


NGINX

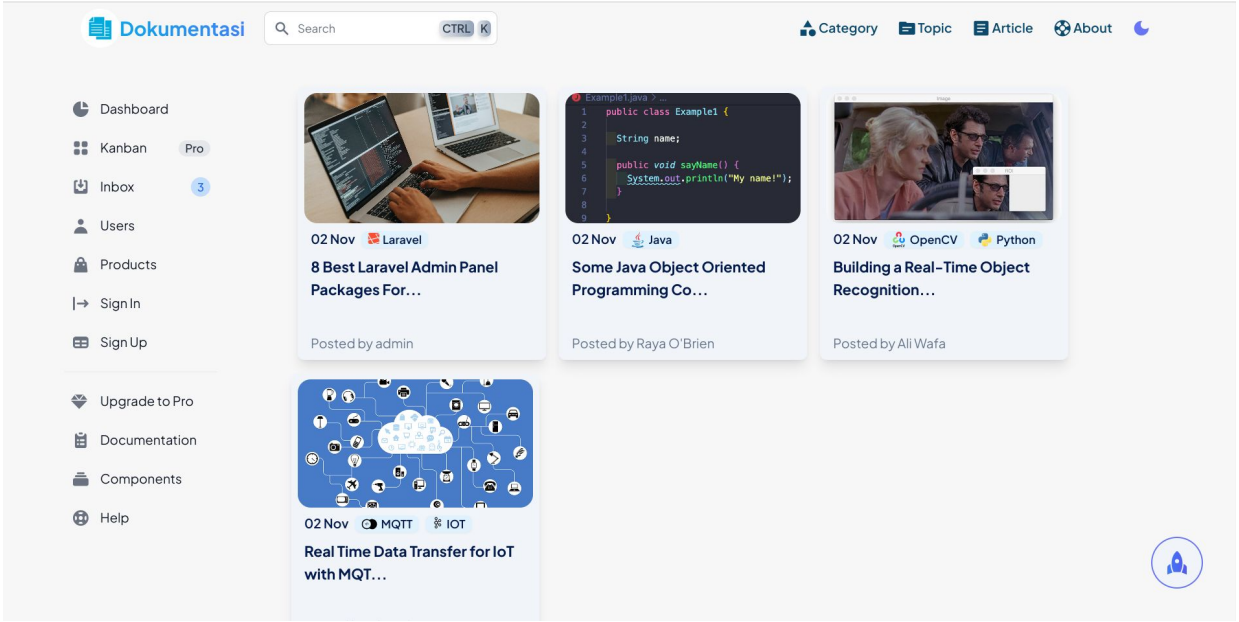

4
CKEditor 4

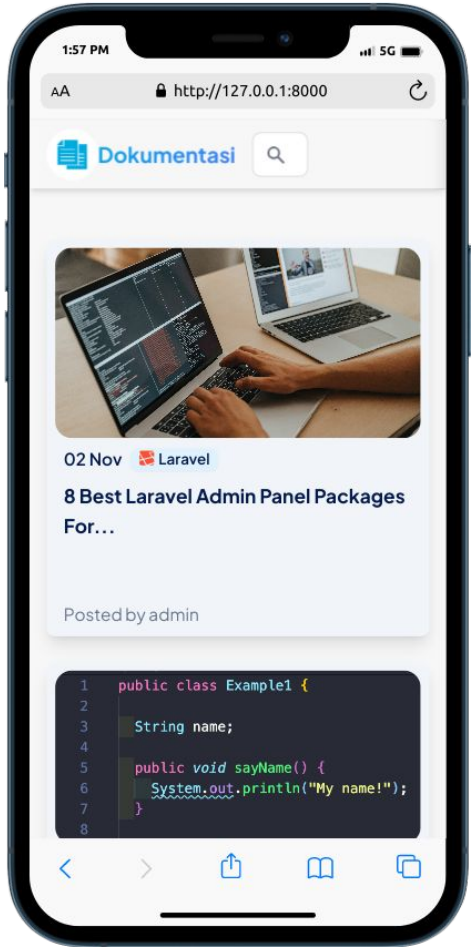
User Auth UI


Dokumentasi
Login!
Email
Password
Login
Doesn't Have Account?
Register Here


12:36 AM
5G
AA
http://127.0.0.1:8000
Dokumentasi
Login!
Email
Password
Login
Doesn't Have Account?
Register Here

Ui Dan UX


Dokumentasi
Search
CTRL K
Category
Topic
Article
About
Dashboard
Kanban
Pro
Inbox
3
Users
Products
Sign In
Sign Up
Upgrade to Pro
Documentation
Components
Help
02 Nov Laravel
8 Best Laravel Admin Panel Packages For...
Posted by admin
02 Nov Java
Some Java Object Oriented Programming Co...
Posted by Raya O'Brien
02 Nov OpenCV Python
Building a Real-Time Object Recognition...
Posted by Ali Wafa
02 Nov MQTT IOT
Real Time Data Transfer for IoT with MQT...

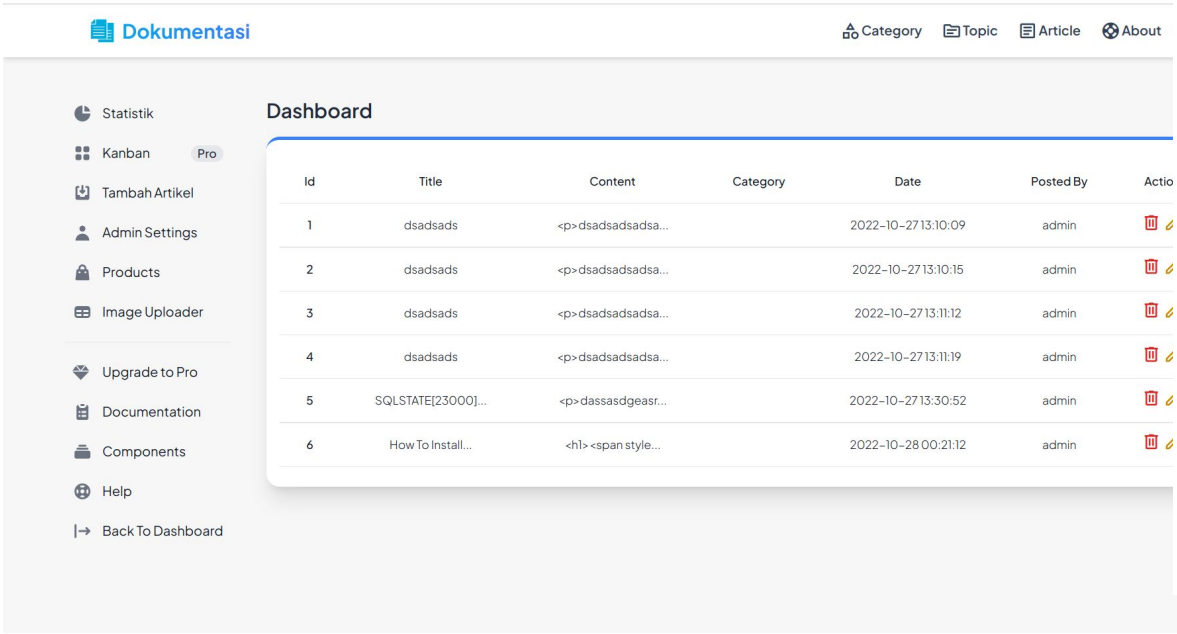

1:57 PM
5G
AA
http://127.0.0.1:8000
Dokumentasi
02 Nov Laravel
8 Best Laravel Admin Panel Packages For...
Posted by admin
public class Example1 {
String name;
public void sayName() {
System.out.println("My name!");
}

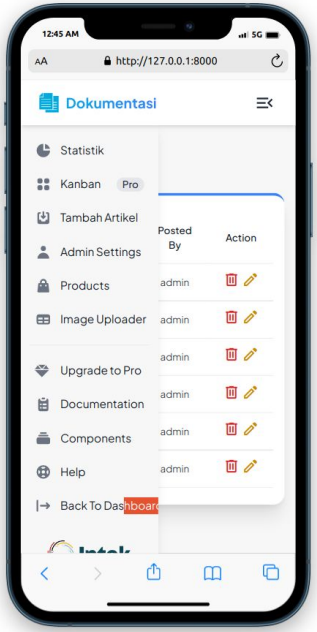
Ui Dan UX


Dokumentasi
Search
CTRL K
Category
Topic
Article
About
Dashboard
Kanban
Pro
Inbox
3
Users
Products
Sign In
Sign Up
Upgrade to Pro
Documentation
Components
Help
Raya O'Brien
Published on Nov 02 , 2022
Some Java Object Oriented Programming Concepts
Java
Example1.java > ...
public class Example1 {
String name;
public void sayName() {
System.out.println("My name!");
}
}
» Find More Post!
Bill Gates
How To Change User In MySQL
Bill Gates
How Create Controller In Laravel
Bill Gates
React Native Vs Flutter , Which is the best?
Bill Gates In Linux
Python OpenCV for beginners
Bill Gates In Flutter
Laravel eloquent Lazy Vs. Eager Loaded
Object Oriented Programming (OOP) might sound intimidating at first, especially if meaningless 15 letter words are just being thrown at you, but the goal of this post is to clear up some of the more common OOP concepts.
Objects are created with a C lass and that class has variables and methods that say what that object can

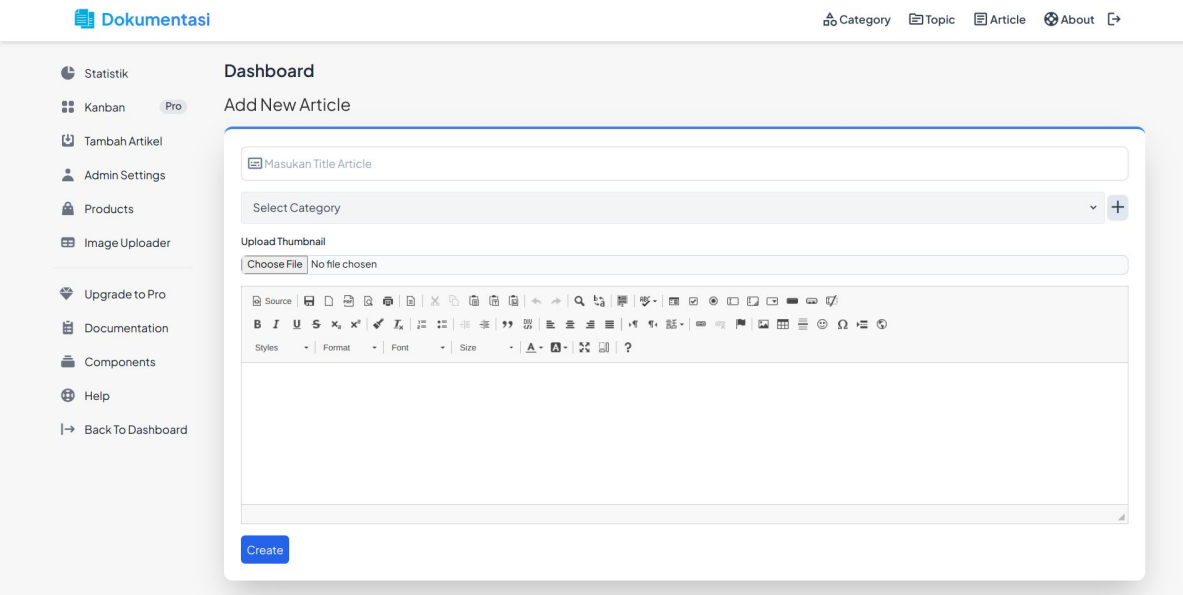

2:04 PM
5G
AA
http://127.0.0.1:8000
Dokumentasi
Raya O'Brien
Published on Nov 02 , 2022
Some Java Object Oriented Programming Concepts
Java
Example1.java > ...
public class Example1 {
String name;
public void sayName() {
System.out.println("My name!");
}
}
Object Oriented Programming (OOP) might sound intimidating at first, especially if meaningless 15 letter words

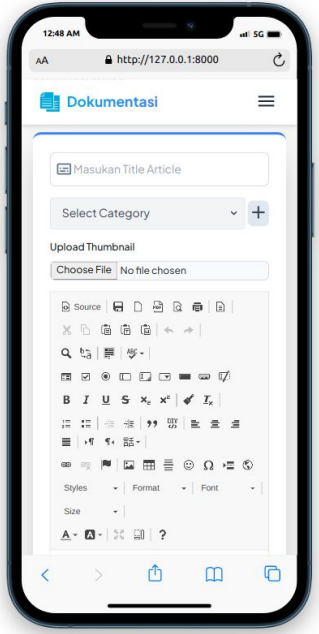
Dasboard UI


Dokumentasi
Category
Topic
Article
About
Statistik
Kanban
Pro
Tambah Artikel
Admin Settings
Products
Image Uploader
Upgrade to Pro
Documentation
Components
Help
Back To Dashboard
Dashboard
Id
Title
Content
Category
Date
Posted By
Actio
1
dsadsads
<p>dsadsadsadsa...
2022-10-27 13:10:09
admin
2
dsadsads
<p>dsadsadsadsa...
2022-10-27 13:10:15
admin
3
dsadsads
<p>dsadsadsadsa...
2022-10-27 13:11:12
admin
4
dsadsads
<p>dsadsadsadsa...
2022-10-27 13:11:19
admin
5
SQLSTATE[23000]...
<p>dassasdgeasr...
2022-10-27 13:30:52
admin
6
How To Install...
<h1><span style...
2022-10-28 00:21:12
admin

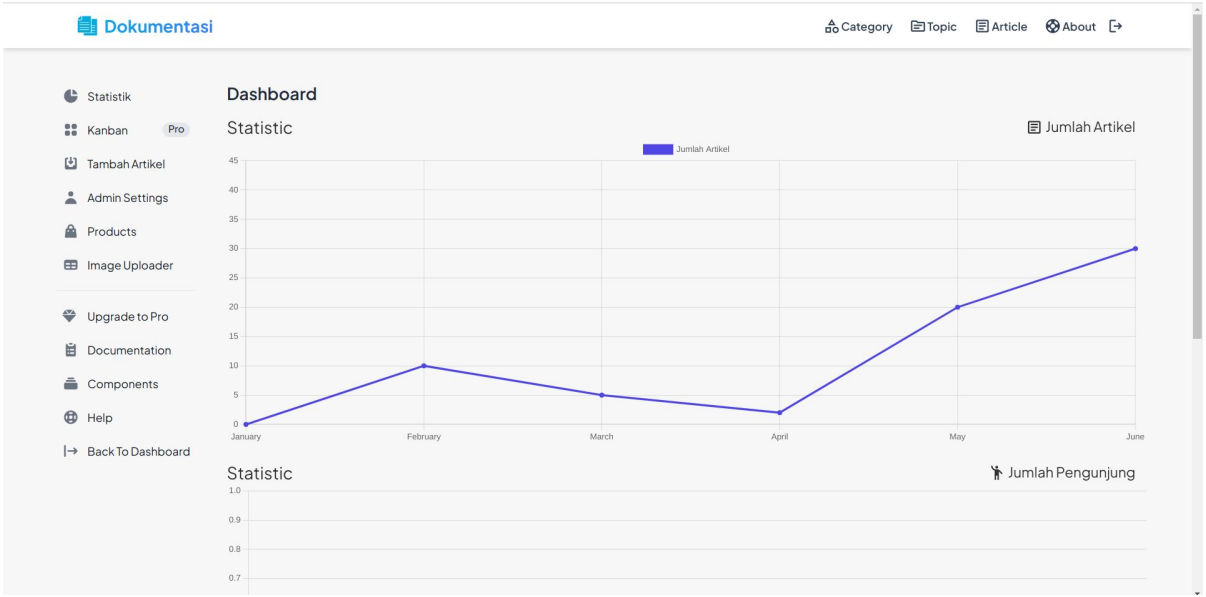

12:45 AM
5G
AA
http://127.0.0.1:8000
Dokumentasi
Statistik
Kanban
Pro
Tambah Artikel
Admin Settings
Products
Image Uploader
Upgrade to Pro
Documentation
Components
Help
Back To Dashboard
Posted By
Action
admin

Dashboard Ui

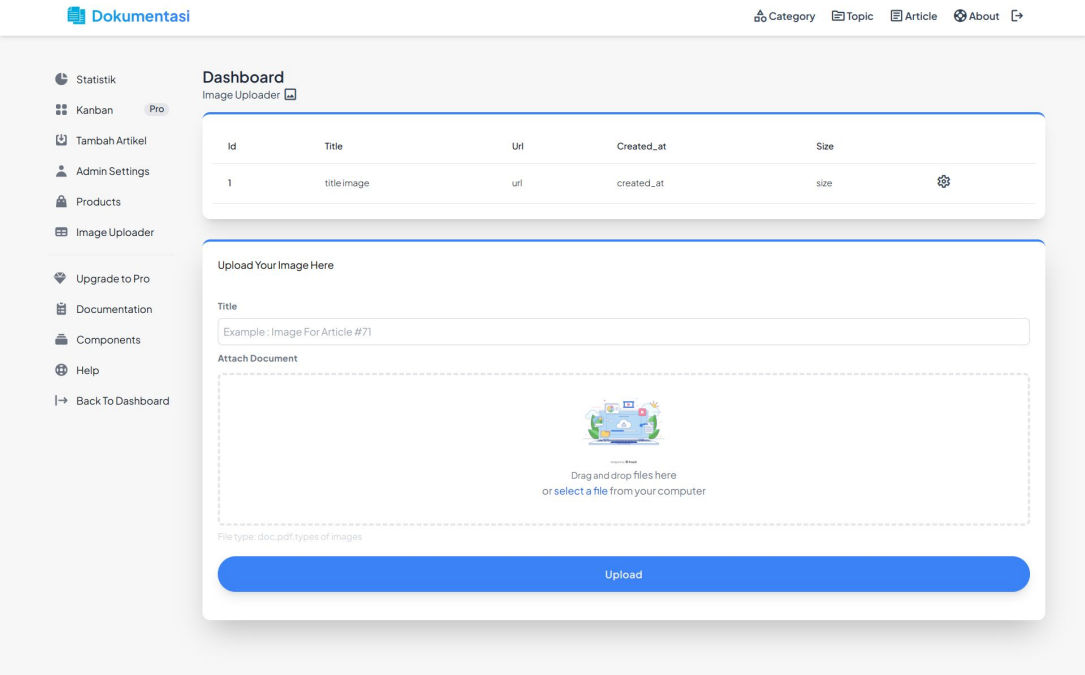


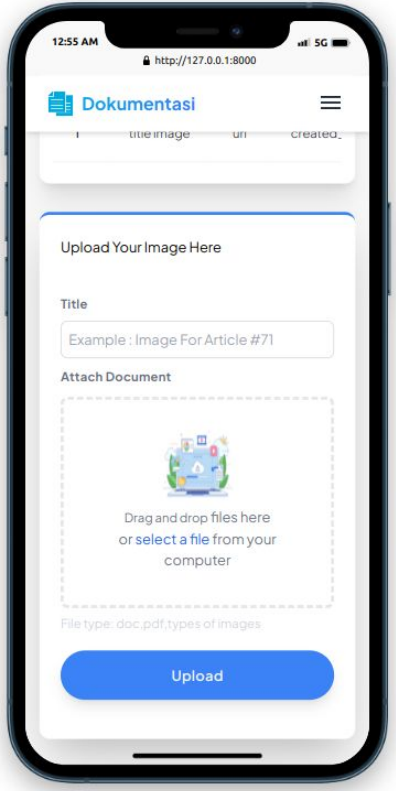


Dashboard UI

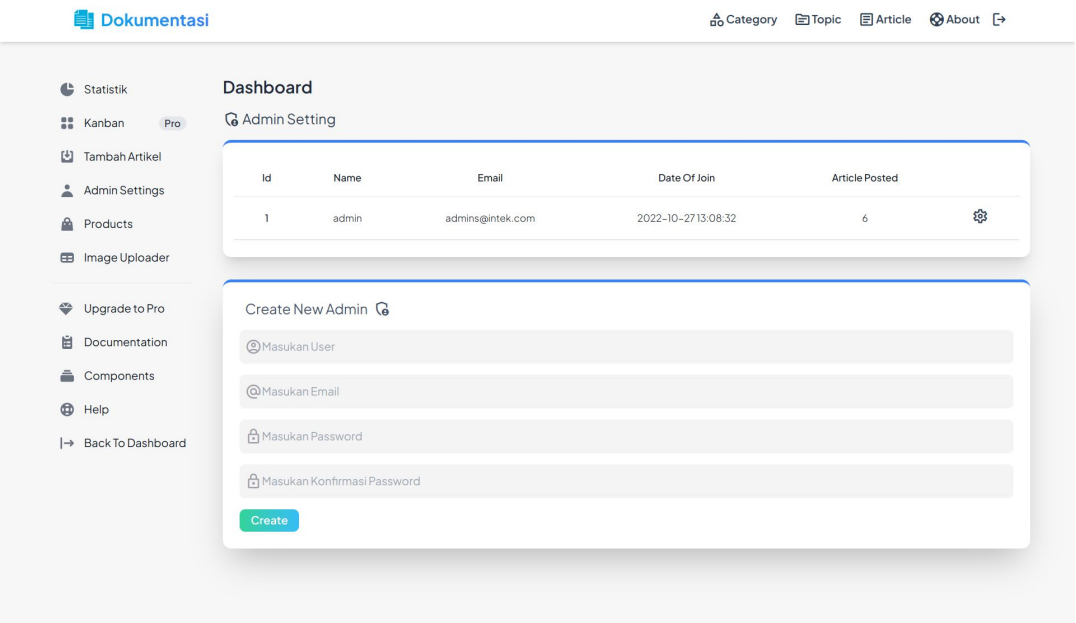


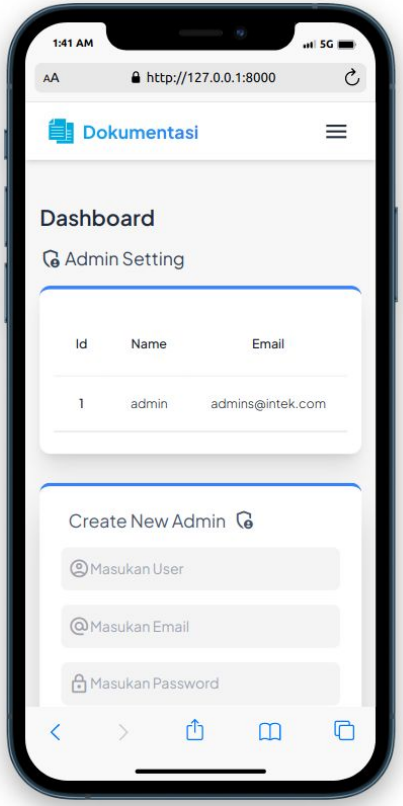
Dashboard Ui



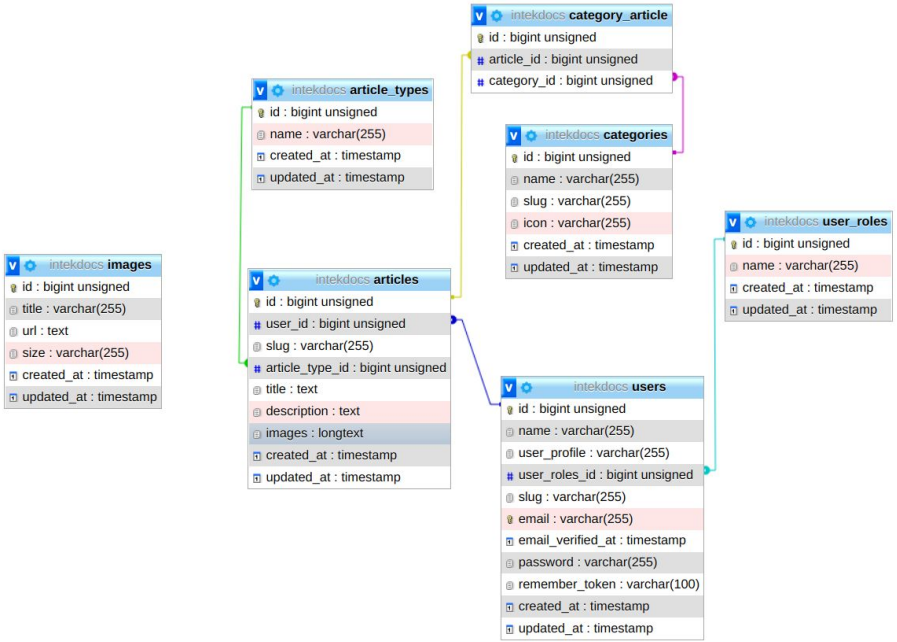


Dashboard Ui





Database Design



Setup Composer, Nginx , PHP FPM , MySQL , NodeJS , PHPMyAdmin

Sebelum Masuk Ke Project Laravel Pastikan Sudah Terinstall Bebarapa Tools Berikut,

Composer sebagai Dependency Manager untuk laravel, PHP FPM adalah PHP yang berjalan di Nginx,

Mysql adalah Database Server, NodeJS Menghandle Javascript Framework Dan Utility Javascript yang terinstall pada laravel , Nginx Sebagai Web Server , dan PHPMyAdmin Untuk Mengelola MySQL

Menginstall Composer

Ketikan ini Pada Terminal Ubuntu

sudo apt-get install curl php-cli php-mbstring git unzip

curl –sS https://getcomposer.org/installer | php

sudo mv composer.phar /usr/local/bin/composer

composer -V (cek versi composer)

composer self-update (Untuk Mengupdate Composer)

Menginstall Nginx

sudo apt update

sudo apt install nginx

sudo apt status nginx (untuk mengecek apa nginx sudah terinstall)

sudo ufw allow ‘Nginx full’

sudo ufw reload

Jika berhasil akan muncul page seperti ini pada localhost

Menginstall PHP-FPM

sudo apt-get install php8.1-fpm -y

sudo systemctl status php8.1-fpm (Mengecek apakah berhasil terinstall)

sudo nano /etc/php/8.1/php.ini

tambahkan script berikut

cgi.fix_pathinfo = 0

sudo systemctl restart nginx

Menginstall MySQL dan PhpMyAdmin

Install MySQL

sudo apt -get install mysql-server

sudo apt-get install php-fpm php-mysql

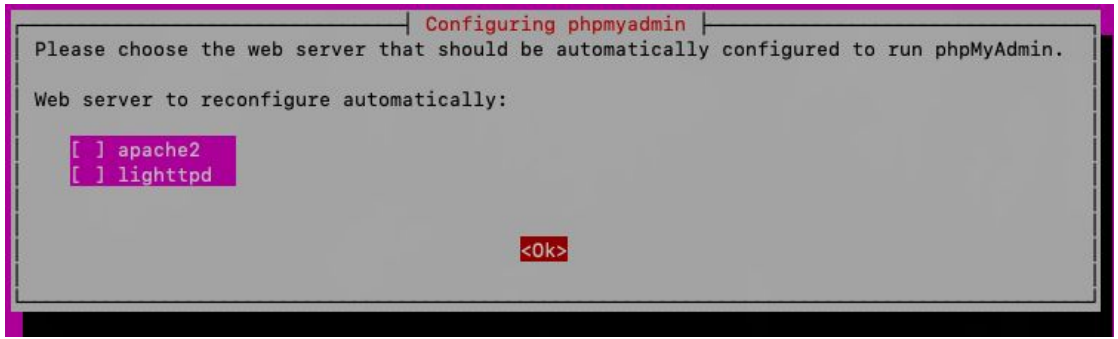
sudo systemctl restart php8.1-fpm

Menginstall MySQL dan PhpMyAdmin

Install PhpMyAdmin

sudo apt install phpmyadmin

apabila ada apache dan lightspeed klik TAB untuk skip



Setelah memilih Skip , Skip step step setelahnya dengan pilih No Saja

Sudo nano /etc/nginx/phpmyadmin.conf

masukan script berikut

```
       location /phpmyadmin {
    root /usr/share/;
    index index.php index.html index.htm;
    location ~ ^/phpmyadmin/(.+\.php)$ {
        try_files $uri =404;
        root /usr/share/;
        fastcgi_pass unix:/run/php/php8.1-fpm.sock;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include /etc/nginx/fastcgi_params;
      }

    location ~* ^/phpmyadmin/(.+\.(jpg|jpeg|gif|css|png|js|ico|html|xml|txt))$ {
        root /usr/share/;
    }
}
```

sudo nano /etc/nginx/sites-available

Tambahkan Script Berikut Pada Line Server

```
server {
       include snippets/phpmyadmin.conf;
      }
```

Membuat Project Laravel

-Pastikan Composer Telah Terinstall

composer create-project laravel/laravel ExampleProject

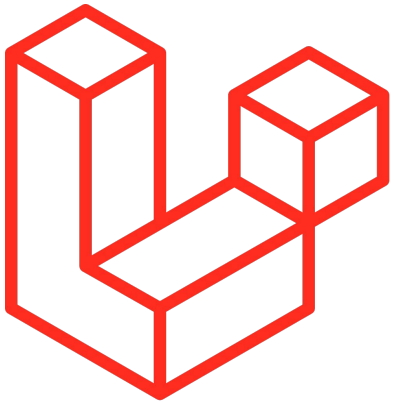
cd ExampleProject

-Menjalankan Laravel

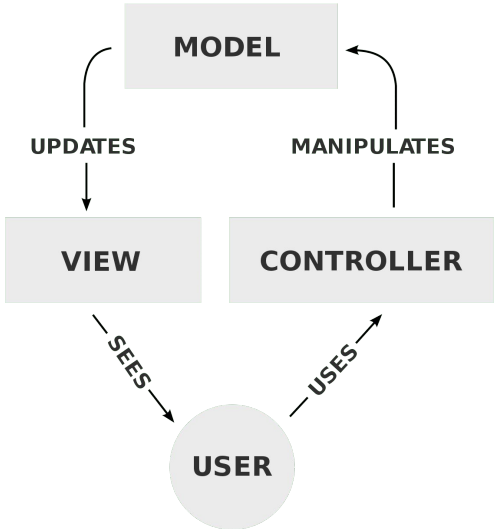
php artisan serve

Apa itu Laravel?

Laravel adalah *framework* berbasis bahasa pemrograman PHP yang bisa digunakan untuk membantu proses pengembangan sebuah website agar lebih maksimal. Dengan menggunakan Laravel, website yang dihasilkan akan lebih dinamis.



Cara Kerja Laravel



Laravel Menggunakan Konsep MVC (Model View Controller) dan juga Menggunakan Konsep Pemrograman Berorientasi Objek (Object Oriented Programming) , MVC Membuat file file dalam laravel tersusun rapi dan untuk codingan backend tidak akan di taruh di View melainkan akan di olah melalui controller sehingga diharapkan lebih aman secara security (Menghindari Penetration Testing)

Struktur Folder Pada Laravel

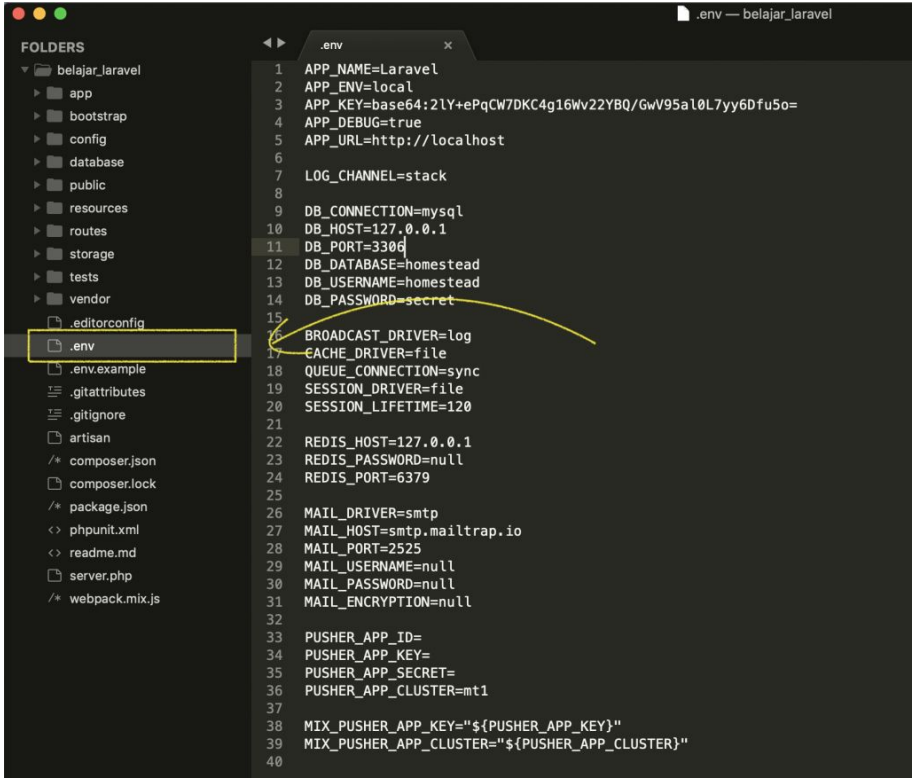
Directory	Deskripsi
App	Menhandle Semua Kode Untuk Berjalannya laravel contoh : controller, Model
Bootstrap	Menhandle Bootstraping Script
Config	Berisi file config aplikasi
Database	Berisi File File yang berhubungan dengan db contoh : Migration,seeder,
Public	Bisa di isikan gambar, javascript file ,CSS file dapat diakses dengan asset()
resources	Berisi file views , template layout

Routes	berisi file yang berhubungan dengan rute url , seperti web.php dan api.php
Storage	Menyimpan Session , hasil compiled dari laravel dan file file lain yang dihasilkan oleh laravel
Test	berisi file file pengujian berbagai case oleh laravel
Vendor	Menghandle Semua Composer Dependency

Folder yang Sering diPakai

.Env Pada Laravel

.env adalah file untuk konfigurasi pada laravel seperti untuk menyambungkan laravel dengan mysql, redist, postgresql dll



Cara Menghubungkan Laravel Dengan MySQL

Buka File .Env

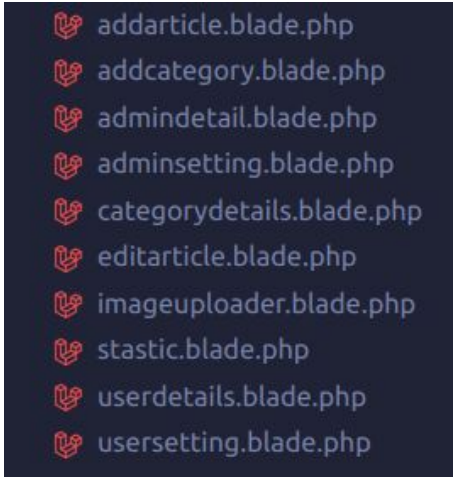
Konfigurasikan Yang

ada di kotak merah

berikut

View Pada Laravel (Blade)

Laravel menggunakan extension khusus untuk Menhandle View atau tampilan depannya yaitu dengan menggunakan blade , blade memungkinkan developer untuk menggunakan sintax sintax khusus yang tidak ada di PHP native atau PHP murni



View Pada Laravel (Blade)

Regular PHP:

```
<?php if ($user->isLogged()): ?>
    Welcome back, <strong><?= $user->name; ?></strong>
<?php endif; ?>
```

Blade:

```
@if ($user->isLogged())
    Welcome back, <strong>{{ $user->name }}</strong>
@endif
```

Blade membuat penulisan kode php pada frontend lebih efisien dan singkat,

{{ }} adalah syntax blade untuk echo

View Pada Laravel (Blade Templating)

Untuk Membangun suatu aplikasi website , kita harus memiliki sebuah layout yang sama antara satu page dengan page lainnya oleh karena itu blade templating sangat penting agar layout dan view efisien tidak membutuhkan banyak kode dan juga head website bisa ditaruh di master layout sehingga diharapkan website lebih cepat tanpa harus meload head dari setiap masing masing page

View Pada Laravel (Blade Templating)

File Layout Master (master.blade.php) ,

@yield berguna untuk kita mengisikan isi kontent dari child page seperti kita mendeklarasikan bahwa tag body ini adalah sebuah canvas lukis yang bisa kita isikan lukisan

```
<!DOCTYPE html>
<html>
<head>
<title>Ini Layout Master</title>
</head>
<body>

 @yield('content')

</body>
</html>
```

View Pada Laravel (Blade Templating)

File page child contoh : about.blade.php

@extends berarti kita ingin menggunakan template atau parent

page yang telah disediakan oleh master.blade.php (layouts master)

@section berarti kita ingin memasukan kontent pada

bagian @yield yang telah di deklarasikan tadi pada layouts.master

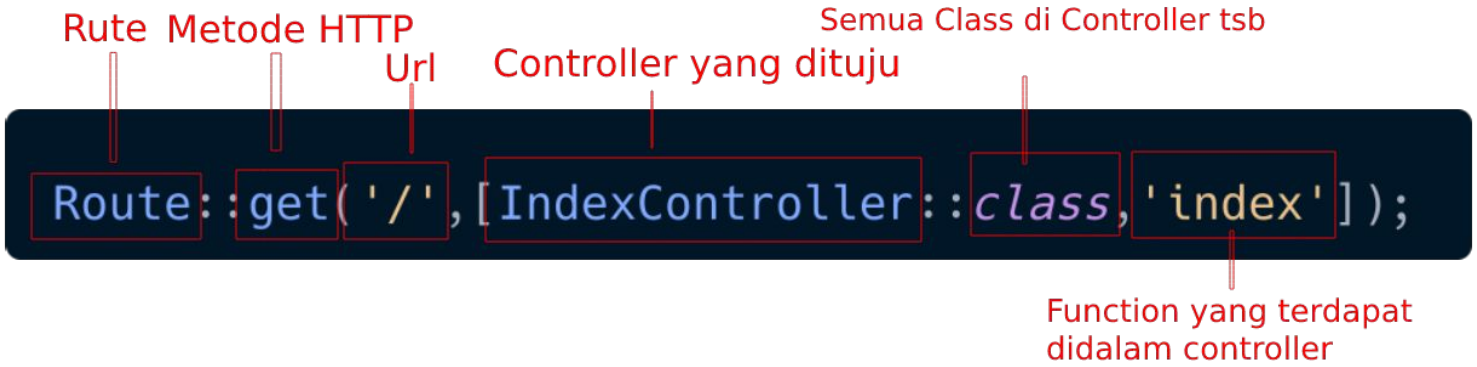
```
@extends('layouts.master')
@section('content')
<h1>Ini Isi Konten Dari Child
Page</h1>
@endsection
```

Routes

Route Berfungsi untuk mengatur rute dari url akan diarahkan ke page atau logic backend mana , melalui method pada controller

```php
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\IndexController;
use App\Http\Controllers\RedirectHandlesController;
use App\Http\Controllers\AdminAuthController;
use App\Http\Controllers\IotController;
use App\Http\Controllers\ArticleController;
use App\Http\Controllers\UserAuthController;

Route::get('/',[IndexController::class,'index']);
Route::post('/',[IndexController::class,'index']);
Route::get('loginadmin',[AdminAuthController::class,'index']); //login admin
Route::get('/article/{slug}',[IndexController::class,'show'])->name('show');
Route::get('dashboard',[AdminAuthController::class,'dashboard']);
Route::get('login',[UserAuthController::class,'index'])->name('userlogin'); //loginuser
Route::post('login',[UserAuthController::class,'auth'])->name('userloginpost');//loginuser
Route::get('register',[UserAuthController::class,'registerindex'])->name('registerindex');
Route::post('register',[UserAuthController::class,'register'])->name('registerpost');
Route::post('loginadmin',[AdminAuthController::class,'auth'])->name('authpost'); //login admin
Route::get('/logout',[AdminAuthController::class,'signOut'])->name('signout');
Route::delete('/article/{id}/delete',[ArticleController::class,'destroy'])->name('articledestroy');
Route::get('/article/{id}/edit',[ArticleController::class,'edit']);
Route::put('article/{id}/update',[ArticleController::class,'update'])->name('update');
Route::get('dashboard/tambahartikel',[ArticleController::class,'index'])->name('addarticle');
Route::post('/dashboard/tambahartikel',[ArticleController::class,'create'])->name('addarticlepost');
Route::get('/dashboard/statistik',[AdminAuthController::class,'statistic'])->name('statisticindex');
Route::get('/dashboard/admin',[AdminAuthController::class,'setting'])->name('adminsetting');
Route::post('/dashboard/admin',[AdminAuthController::class,'createnewadmin'])->name('createadmin');
Route::get('/dashboard/admin/{user:slug}/details',[AdminAuthController::class,'admindetails'])->name('admindetails');
Route::post('/dashboard/admin/{user:slug}/details',[AdminAuthController::class,'adminUpdatePicture'])-
Route::get('/dashboard/imageuploader',[AdminAuthController::class,'imageuploaderview'])->name('imageuploader');
Route::post('/dashboard/imageuploader',[AdminAuthController::class,'imageuploader'])->name('imageuploaderpost');
Route::delete('/dashboard/image/{id}/delete',[AdminAuthController::class,'imagedestroy'])->name('imagedestroy');
Route::get('/dashboard/addcategory',[AdminAuthController::class,'addcategoryindex'])->name('addcategory');
Route::get('/dashboard/category/{slug}/details',[AdminAuthController::class,'categorydetails'])->name('categorydetails');
Route::post('/dashboard/addcategory',[AdminAuthController::class,'addcategory'])->name('addcategorypost');
Route::get('/dashboard/usersetting',[AdminAuthController::class,'usersettingindex'])->name('usersetting');
Route::get('/dashboard/usersetting/{user:slug}/details',[AdminAuthController::class,'userdetails'])->name('userdetails');
Route::post('/dashboard/usersetting/{user:slug}/details',[AdminAuthController::class,'userupdatepic'])->name('userupdate');
Route::get('/dashboard/iot',[IotController::class,'index'])->name('iotindex');
Route::get('category/{slug}',[ArticleController::class,'category'])->name('categoryindex');
Route::get('404',[RedirectHandlesController::class,'index']);
```

Struktur dari Route Pada Laravel



Javascript File

Untuk Confirm Password Validation Pada Register dengan cara membandingan value antara 2 input dan mengeluarkan output tertentu apabila password dan confirm password tidak sama

```
$(document).ready(function () {
    $("#submits").attr("disabled",true);
    $("#passwords").on('keyup',function(){
    let valuepassword =  $(this).val()
    $("#confirmpasswords").on('keyup',function(){
    let confirmvalue =  $(this).val()

   if(valuepassword != confirmvalue){
       $("#submits").attr("disabled",true);
       $("#message").show()
       $("#message").text("Password Tidak Cocok!")
    } else if ( valuepassword == "" && confirmvalue == "") {
      $("#message").hide()
      $("#submits").attr("disabled",true);
    } else {
      $("#submits").removeAttr("disabled");
      $("#message").empty()
    }


   });
   });



});
```

Javascript File

Untuk Hamburger Menu Pada Mobile

Apabila Ada perubahaan pada checkbox

tambahkan atau hapus class

```
$(document).ready(function () {
    $('#hamburgercheckbox').change(function(){
        if($(this).is(':checked')){
            let hamburgericon = $('#hamburgericon')
              if(hamburgericon.find('menu')){
                $("#hamburgericon").empty();
                $("#hamburgericon").html("menu_open");
                $(".mobile-dashboard-nav").addClass("slide");
              }
            } else {
                $(".mobile-dashboard-nav").removeClass("slide");
                $("#hamburgericon").empty();
                $("#hamburgericon").html("menu");
            }

    });


});

   });


});
```

Javascript File

Upload Image Logic

```
$(document).ready(function () {
    let fileInput = document.querySelector('#input');
    let dropArea = document.querySelector('#dropimagehere')
    let thumbnail = document.querySelector("#dropimagethumbnail");
    let borderDrop = document.querySelector("#borderdropimage")
    let file
    $("#dropimagehere").on('dragover',function(event){
    event.preventDefault();
     $("#borderdropimage").removeClass("border-dashed");
})

$("#dropimagehere").on('dragleave',function(event){
    event.preventDefault()
    $("#borderdropimage").addClass(".border-dashed");
})
fileInput.addEventListener("change",(event) => {
    event.preventDefault();
    if(!borderDrop.classList.contains("border-dashed")){
        borderDrop.classList.add("border-solid")
        borderDrop.classList.remove("border-dashed");
    } else {
        borderDrop.classList.remove("border-dashed");
        borderDrop.classList.add("border-solid")
    }
    file = event.dataTransfer.files[0];
    showFile();
})


dropArea.addEventListener("drop",(event) => {
    event.preventDefault();
    if(!dropArea.classList.contains("border-dashed")){
        borderDrop.classList.add('border-solid"')
        borderDrop.classList.remove("border-dashed");
    } else {
        borderDrop.classList.remove("border-dashed");
        borderDrop.classList.add('border-solid"')
    }
    file = event.dataTransfer.files[0];
    showFile();
})



function showFile(){
    let fileType = file.type;
    let validExtensions = ["image/jpeg","image/jpg","image/png"];
    if(validExtensions.includes(fileType)){
        let fileReader = new FileReader();
        fileReader.onload = ()=> {
            let fileURL = fileReader.result;
            let imgTag = `<img src="${fileURL}" alt="image"
class="object-cover">`
            thumbnail.innerHTML = imgTag;
        }
        fileReader.readAsDataURL(file);
    } else {
        file = ""
        Swal.fire({
            icon: 'error',
            title: 'Oops...',
            text: 'File Bukan Gambar',
          })

       $("#borderdropimage").addClass("border-dashed");

    }
}

});
```

Javascript File

Copy To ClipBoard Logic

Mencari Semua Id dari class tablerows

dan melakukan “for of” untuk mendapat id satu per satu lalu di cocokan

```
const tablerows = document.getElementsByClassName('table-rows');
for (const tbrow of tablerows) {
    $('.copy-button').on('click', function(e) {
       let singletbrow = tbrow.querySelector('#image-id')
       let tbrows = tbrow.querySelector('#image-id').innerText
       if($(this).attr("id") == tbrows){

 navigator.clipboard.writeText(singletbrow.parentNode.querySelector('#ima
ge-url').getAttribute('title'))
          tippy('.copy-button', {
            content: 'Copied!',
            trigger:'click'
          });

        }

});
```

Javascript File

Toggle Dark Javascript

jika button setLight di click

dan setLight Mempunyai text dark_mode

kosongkan class dan tambah class baru

atau sebaliknya

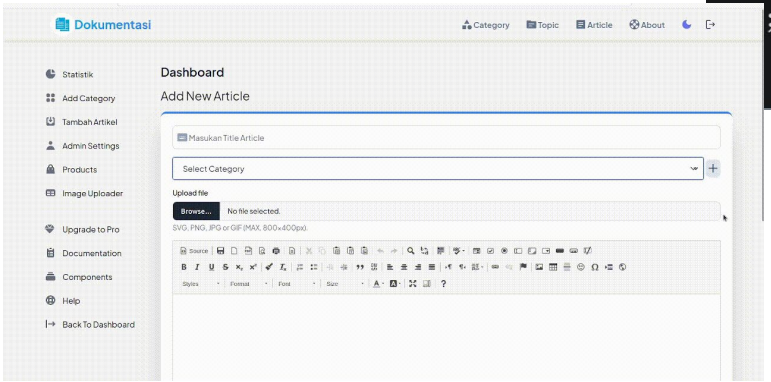
```
$(document).ready(function () {
  $('#setlight').on('click',function(){
    if($('#setlight').text().includes('dark_mode')){
        $('#setlight').empty();
        $('#setlight').text('light_mode');
        $('.toggle').removeClass('text-indigo-500')
        $('.toggle').addClass('text-red-500')
    } else {
        $('#setlight').empty()
        $('#setlight').text('dark_mode');
        $('.toggle').removeClass('text-red-500')
        $('.toggle').addClass('text-indigo-500')
    }
  })



})
```

Javascript File

Add New Category Di

Dashboard Tambah Article

```
$('#addnewcategory').on('click',()=>{

      let clone = $('#inputcategory').clone(true);
      $('.category-input ').after(clone)
      let categoryevery =  $('.category-every')
      if(categoryevery.length > 2){
         $("#addnewcategory").hide()
      }

;
```



Javascript File

Search Ajax Jquery

Jika Keyboard diketik , get value kirimkan melalui ajax request menuju controller dan dikembalikan lagi dalam bentuk json data kemudian di append ke html

```
$(document).ready(function () {
    $("#default-search").on('keyup',function(){
        let searchQuest = $(this).val();
        var url = window.location.pathname
        $.ajax({
            method: 'get',
            url: url,
            dataType: 'json',
            data: {
                "searchQuest" : searchQuest,
                "_token" : $('#token').val()
            },
            success: (data) => {

                if(searchQuest == ""){
                    $('#result').html("No Recent Searches")
                } else {
                    $('#results').html(data)
                }


          }
        })

    })

})
```

Models Dan Migrations

Untuk User

Model / Eloquent pada laravel digunakan untuk **mewakili pemanggilan tabel**. Dengan demikian, model merupakan bagian yang utama dimana kita akan berinteraksi ke tabel yang bersangkutan melalui model yang dibuat.

Migration Adalah Konsep Isi Table Yang Akan dibuat

Untuk Membuat Model Sekaligus Migrations

```
php artisan make:model NamaModel -m
```

-m artinya migrations

```php
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('user_profile')->nullable();
            $table->string('slug')->nullable();
            $table->string('role')->nullable();
            $table->string('email')->unique();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->rememberToken();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('users');
    }
};
```

```php
<?php

namespace App\Models;

// use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Laravel\Sanctum\HasApiTokens;

class User extends Authenticatable
{
    use HasApiTokens, HasFactory, Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    protected $fillable = [
        'name',
        'slug',
        'user_profile',
        'role',
        'email',
        'password',
    ];

    /**
     * The attributes that should be hidden for serialization.
     *
     * @var array<int, string>
     */
    protected $hidden = [
        'password',
        'remember_token',
    ];

    /**
     * The attributes that should be cast.
     *
     * @var array<string, string>
     */
    protected $casts = [
        'email_verified_at' => 'datetime',
    ];

    public function article(){
        return $this->hasMany(Article::class);
    }

    public function getRouteKeyName()
    {
       return 'slug';
    }
}
```

Model Dan Migrations Categories

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use App\Models\Article;
use App\Models\ArticleCategory;
use Illuminate\Database\Eloquent\Model;

class Category extends Model
{
    use HasFactory;

    protected $fillable = [
        "name",
        "icon"
    ];

    public function articles(){
        return
$this->belongsToMany(Article::class,'category_article')->withPivot('categ
ory_article');
    }


    public function category(){
        return $this->hasMany(Category::class);
       }

}
```

```php
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('categories', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('icon');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('categories');
    }
};
```

Model Dan Migrations

Untuk Artikel

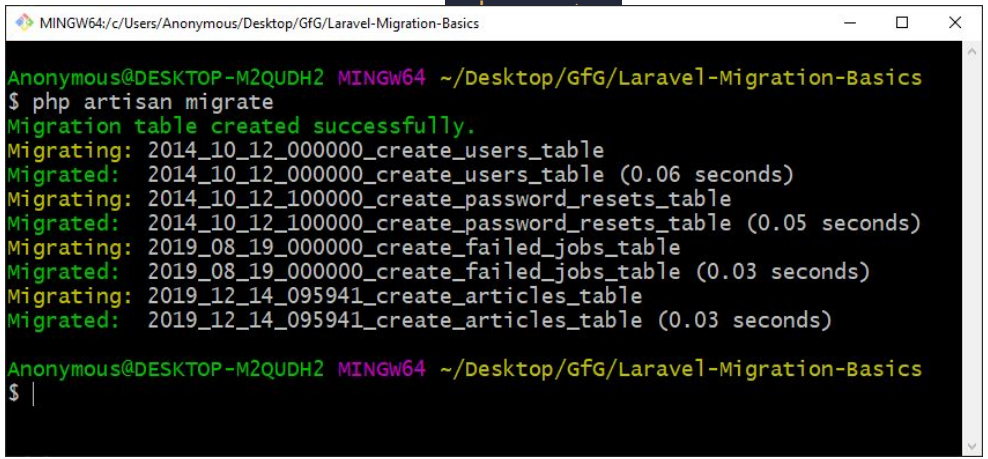
```php
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;



return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('articles', function (Blueprint $table) {
            $table->id();
            $table->foreignId('user_id')->constrained();
            $table->string('slug');
            $table->text('title');
            $table->text('description');
            $table->string('images');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('articles');
    }
};
```

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use App\Models\Category;
use Illuminate\Support\Str;

class Article extends Model
{
    use HasFactory;


    protected $fillable = [
        "user_id",
        "title",
        "slug",
        "category_id",
        "description",
        "images",
    ];

    public function user(){
        return $this->belongsTo(User::class);
    }

    public function categories(){
        return $this->belongsToMany(Category::class,'category_article');
    }


    public function getShortDescriptionAttribute()
    {
        return Str::limit(
            nl2br(strip_tags($this->description)),
            90
        );
    }

}   * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('users');
    }
};
```

Lakukan Migrate

Migrate bertujuan untuk memasukan rancangan table kita ke dalam database asli

Ketikan ini pada terminal `php artisan`



Model & Migration Article_Category

Article_Category adalah sebuah pivot table

yang berguna menghubungkan antar 2 table

pada relasi many to many

```php
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('category_article', function (Blueprint $table) {
            $table->bigIncrements('id');
            $table->unsignedBigInteger('article_id');
            $table->unsignedBigInteger('category_id');

 $table->foreign('article_id')->references('id')->on('articles')->onDelet
e('cascade');

 $table->foreign('category_id')->references('id')->on('categories')->onDe
lete('cascade');
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('category_article');
    }
};
```

Melakukan first seeder

Seperti artinya, Seeder (benih) ini adalah sebuah **fungsi untuk menanamkan Benih berupa Data pada Database kita**. Jadi saat nanti kita melakukan migrations , Database kita sudah terisi Data dari hasil Seeder kita

cara melakukan seeder

```
php artisan make:seeder NamaSeeder
```

buka folder /database/seeders/FirstSeeder.php

masukan script berikut ke function run() pada file FirstSeeder.php

```
$this->call([
            NamaSeeder::class,
          ]);
```

Value FirstSeeder

```
Params::create([
   “Isi Table Pertama” => “isi”,
  “Isi Table Kedua” => “isi”
 ]);
```

Params adalah nama table yang akan diakses

create adalah method untuk table yang akan kita akses , selain create terdapat method lain seperti update,delete dll

Untuk Mengakses isi table dapat menggunakan string nama struktur table kemudian dilanjut dengan `=>`

Dan di isi dengan valuenya bisa juga menggunakan **$request->**ataupun langsung menggunakan string dsb

Memasukan value Ke FirstSeeder

-buka file namaSeeder.php pada folder database/seeder

-use terlebih dahulu model yang akan kita masukan value

contoh : `use App\Models\User;`

-Kemudian masuk pada function Run()
-masukan script berikut

```
  User::create([
            "name" => "admin",
            "Slug" => Str::slug('admin'),
            "role" => "moderator",
            "email" => "admins@intek.com",
           "password" => bcrypt("admin123")

        ]);
```

Lakukan Migrate Fresh beserta dengan seed

Migrate Fresh bertujuan untuk merancang kembali rancangan table kita ke dalam database asli serta memasukan seeder `php artisan migrate:fresh --seed`

Cek Kembali PhpMyAdmin , maka value dan struktur table akan terbentuk

Membuat Controller

Controller merupakan jembatan atau penghubung antara view dan model. jadi secara mudah nya, controller bisa kita pahami sebagai pengatur view dan model. controller sendiri biasanya berperan sebagai pengolah data.

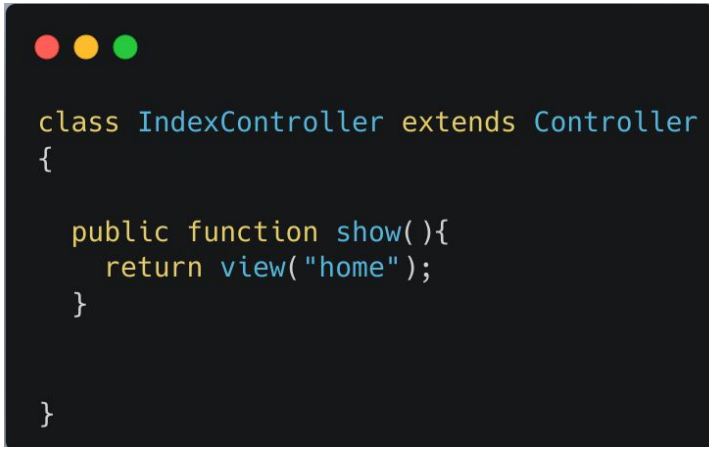
Controller memiliki banyak method tergantung

berapa banyak method yang akan kita buat

method method tersebut terbungkus kedalam class

NamaController yang merupakan Inheritences dari

class controllers



contoh controller

Menghubungkan Controller Dengan Route

Cara menghubungkan controller dengan route dapat dengan mengetik

Use *App/Http/Controllers/NamaController* *pada file web./php*

lalu dapat digunakan dengan

Route::get('/',[NamaController::class,'index']);

otomatis setiap url atau route / yang dibuka akan menjalankan method index pada NamaController

Request HTTP

Apa itu Request HTTP? ***HTTP Request*** yaitu dimana server membaca apa yang dikirimkan oleh client melalui aplikasi web server.

Metode Dalam Http Request : **Get** (Mendapatkan data dari Server)

Post (Mengirim data ke Server)

Put (Mengupdate data ke server)

Delete (Menghapus data dari server)

Penggunaan Request HTTP pada Laravel

```
<form action="{{ route('users.store') }}" method="POST">
    @csrf
                <input type="text" name="name" class="form-control"
placeholder="Name">
<button type="submit">Submit</button>

</form>
```

action adalah tujuan dari form kita, sedangkan POST adalah metode pengirimannya

@csrf adalah untuk melindungi website dari serangan heker . yang masuk melalui input field

name : untuk menamai input yang kita buat agar dapat di baca oleh $request

Mengambil Value Dari Request Yang Dikirim

Setiap Request yang dikirimkan oleh kita melalui form akan masuk ke url sesuai dengan action dari form tersebut , request ini bisa ditangkap oleh Controller dengan cara seperti ini

Jangan lupa tambahkan `use Illuminate\Http\Request;`

```
public function ambilIsiRequest(Request $request)
$request->namafield;
}
```

namafield adalah nama yang kita berikan pada input field

request tersebut dapat dimasukan ke dalam variable dan dan di kembalikan ke view atau dimasukan ke dalam database

Compact View

Compact bertujuan untuk mengembalikan value dari sebuah variable pada controller untuk dilempar ke dalam view

```
public function kembalikanValue(Request $request)
$nama = $request->nama;
return view("index",compact("nama"))
}
```

Variable nama akan dikembalikan ke view dengan menggunakan petik dua lalu pada view bisa di foreach

Compact View With Models

Kita dapat memanggil sebuah table dari database dan mengembalikan isinya ke view dengan menggunakan compact

```
public function kembalikanValueTable(Request $request)
$namatable = NamaTable::all();

return view('namaview',compact("namatable"))
}
```

Kita dapat memanggil sebuah table dan dapat melakukan aksi pada table tersebut , function all() berfungsi untuk mengambil semua data yang terdapat pada table NamaTable

Memperlihatkan Hasil Dari Controller Ke View

```
@foreach($namatable as $pertable)
{{ $pertable->propertytabel }}
@endforeach
```

@foreach bertujuan untuk memformat $namatable yang tadinya berbentuk assosiative array ke class yang dapat diakses dengan panah , {{ }} pada laravel berguna untuk print apa saja dari laravel ke dalam html

Folder Controllers - AdminAuthController

```php
namespace App\Http\Controllers;
use Hash;
use App\Models\Article;
use App\Models\User;
use App\Models\UserRole;
use App\Models\Image;
use Illuminate\Support\Facades\Storage;
use App\Models\Category;
use Session;
use Illuminate\Support\Str;
use Illuminate\Support\Facades\Auth;
use Illuminate\Http\Request;
use
RealRashid\SweetAlert\Facades\Alert;
```

Import Beberapa Utility seperti Hash,Str,Session, Auth,Http Request,dan SweetAlert

Use Beberapa Model Yang diperlukan : Article,User,UserRole,Image

Folder Controllers - AdminAuthController

```php
class AdminAuthController extends Controller
{
    public function index(){
        return view('Auth.login');
    }

    public function auth(Request $request){

         $request->validate([
            'email' => 'required',
            'password' => 'required',
         ]);
         $credentials = $request->only('email','password');
         $remember_me = $request->has('remember') ? true : false;

         if(Auth::attempt($credentials,$remember_me)) {
            return redirect()->intended('dashboard')->with('success','Berhasil Login');
         } else {
            return redirect('/login');
         }
    }


    public function dashboard(){
        $article = Article::all();
        if(!Auth::check() || Auth::user()->user_roles_id != 1){
            return redirect('loginadmin')->withErrors(['msg' => 'User Atau Password
Salah'])} else if (Auth::user()->user_roles_id == 1) {
        return view('Auth.dashboard',compact('article'));
        }
    }
```

Tambahkan Method untuk mereturn view index login

Dan tambahkan method untuk mengolah login user

jika user gagal login return redirect ke halaman login

Buat juga Method untuk mereturn dashboard

beserta middleware access menggunakan session dan cookies dari laravel token

Folder Controllers - AdminAuthController

```
 public function statistic(){
        if(!Auth::user() || Auth::user()->user_roles_id != 1){
            return redirect('/404');
        } else {
            return view('tools.stastic');
        }

    }


    public function setting(){
        if(!Auth::user() || Auth::user()->user_roles_id != 1){
            return redirect('/404');
        } else {
            $user = User::where('user_roles_id',1)->with('article')-
>get();      return view('tools.adminsetting', compact('user'));
        }

    }
```

Membuat Method Untuk Mereturn view tools.stastic dan adminsetting jangan lupa menggunakan middleware untuk mencegah

akses dari url selain admin

Folder Controllers - AdminAuthController

```
public function createnewadmin(Request $request){

  $request->validate([
  "user" => "required",
  "Email" => "required | unique:users,email,",
  "profilepic" => "image|mimes:jpg,jpeg,png,gif|max:3048",
  "password" => "required"

  ]);

  if($request->hasFile('profilepic')){
      $images = $request->file('profilepic');
      $filename =$images->getClientOriginalName();
      $images->storeAs('userprofile',$filename);

     }

   User::create([
      "name" => $request->user,
      "user_profile" => $filename,
      "slug" => Str::slug($request->user),
      "user_roles_id" =>  1,
      "email" => $request->Email,
      "password" => bcrypt($request->password),
   ]);

   Alert::success('Admin Baru Terbuat');


   return redirect('dashboard/admin')->withError('["email" => "Email Are
Used"]');

  }
```

Membuat method createnewadmin , mengambil value yang diketikan user dengan menggunakan Request $request pada parameter method , dan mengakses method lalu menggunakan function create untuk membuat admin jika sudah di submit

munculkan alert berhasil jika berhasil , return erorr ketika gagal dan return redirect ke dashboard ketika semuanya berhasil

Folder Controllers - AdminAuthController

```
public function admindetails(Request $request,$slug){
 if(!Auth::user() || Auth::user()->user_roles_id != 1){
     return redirect('/404');
 } else {
     $user = User::with('userroles')->where('slug',$slug)-
>firstOrFail();
     $roles = UserRole::all();
     if($request->ajax()){
         if($request->has('valueselect')){
             $user->update([
                "user_roles_id" => $request->valueselect
             ]);
         }
     }
     return view('tools.admindetail',compact('user','userroles'));
 }

}
```

mereturn view sesuai dengan slug yang dikirimkan

mengambil request dari ajax untuk mengubah roles

Folder Controllers - AdminAuthController

```php
public function imageuploaderview(){
     $images = Image::all();
     return view('tools.imageuploader',compact('images'));
  }

  public function imageuploader(Request $request){


  $validated = $request->validate([
    'title' => 'required',
    'images' =>
'required|image|mimes:jpg,jpeg,png,gif|max:3048',
  ]);

   if($request->hasFile('images')){
    $images = $request->file('images');
    $filename =$images->getClientOriginalName();
    $images->storeAs('image',$filename);
    $imageSize = $images->getSize();
    $fil = number_format($imageSize / 1048576,2);
   }

    $title = $request->title;

    Image::create([
        "title" => $request->title,
        "url" => $filename,
        "size" => $fil,
    ]);

    return redirect()->back();

    Alert::success('Sukses Menambahkan Gambar');

}
```

Untuk Mengupload Gambar dengan menerima file dari Request $request kemudian di validasi jenis file yang dapat di upload kemudian file tsb di storeAs pada storage lalu path nya diambil menggunakan getOriginalClientName() dan disimpan pada property table url jika telah berhasil akan mereturn redirect back

Folder Controllers - AdminAuthController

```php
public function addcategoryindex(){
    $categories = Category::all();
    return view('tools.addcategory',compact('categories'));
}

public function addcategory(Request $request){
   $title = $request->titlecategory;
   $urlicons = $request->urlicon;

   if($request->hasFile('icon')){
       $icons = $request->file('icon');
       $iconname = $icons->getClientOriginalName();
       $icons->storeAs('iconcategory',$iconname);
       Category::create([
         "name" => $title,
         "slug" => Str::slug($title,'-'),
         "icon" => "<img
src='/storage/iconcategory/$iconname'>"
       ]);
        return redirect()->back();
        Alert::success("Sukses Menambahkan Gambar");
   } else {
     Category::create([
        "name" => $title,
        "slug" => Str::slug($title,'-'),
        "icon" => $urlicons,
       ]);

       return redirect()->back();
       Alert::success("Sukses Menambahkan Gambar");
   }

}
```

addcategoryindex() berfungsi untuk mengambil semua kategori yang telah dibuat dan di return ke view ,

addcategory() mengambil request dari field input untuk di masukan kedalam table category pada database, terdapat kondisi untuk sumber icon dari kategori bisa menggunakan url gambar atau mengupload langsung dari komputer

Folder Controllers - AdminAuthController

```
  public function usersettingindex(){
     $defaultuser = User::where('user_roles_id',2)->orWhere('user_roles_id',3)-
>get();
     return view('tools.usersetting',compact('defaultuser'));
   }

   public function userdetails(Request $request,$slug){
    $defuser = User::with('userroles')->where('slug',$slug)->firstOrFail();
    $userroles = UserRole::all();
    if($request->ajax()){
        if($request->has('valueselect')){
             $defuser->update([
                "user_roles_id" => $request->valueselect
             ]);
        }
   }
    return view('tools.userdetails',compact('defuser','userroles'));
   }
```

usersettingindex() berfungsi untuk mengembalikan view usersetting

userdetails mengambil request dari input field dan mengembalikan view dari slug , function ini menggunakan ajax untuk mengirim request nya jadi tidak perlu menggunakan submit dan akan langsung terupdate ke database

Folder Controllers - AdminAuthController

```
    public function adminUpdatePicture(Request $request,$slug)
{    $user = User::where("slug",$slug);
     if($request->hasFile('imagepicture')){
      $image = $request->imagepicture;
      $getName = $image->getClientOriginalName();
      $image->storeAs('userprofile',$getName);
      $user->update([
        "user_profile" => $getName
      ]);
    }
    if($request->has('updatename')){
     $user->update([
         "name" => $request->updatename,

     ]);
    }
    return redirect()->back();
}
```

adminUpdatePicture adalah function untuk menghandle foto profil admin , mengambil request dari input file kemudian mengupdate ke database sesuai dengan admin pada slug,

serta dapat untuk mengupdate admin username

jika sudah maka akan me return redirect kembali ke halaman sebelumnya

Folder Controllers - AdminAuthController

```
public function imagedestroy($id){
    $imgtodelete = Image::findOrFail(decrypt($id));
    $imgpath = $imgtodelete->url;
    if(!unlink(storage_path('app/public/image/'.$imgpath)))
{       $imgtodelete->delete();
    } else {
        $imgtodelete->delete();
        Storage::delete($imgpath);
    }
    return redirect()->back();
  }
```

imagedestroy adalah method untuk menghapus gambar sesuai dengan $id yang dikirimkan

dan juga akan men-unlink storage_path pada gambar agar gambar dapat hilang dari server

dengan menggunakan Storage::delete

Folder Controllers - AdminAuthController

```
    public function categorydetails(Request $request , $slug){
    $categories = Category::with('articles')->where('slug',$slug)->get();
    return view('tools.categorydetails',compact('categories'));
    }
```

categorydetails berfungsi untuk mereturn view sesuai slug yang dikirimkan

Folder Controllers - AdminAuthController

```
public function signOut(){
        Session::flush();
        Auth::logout();
        return redirect('/login');
    }
```

method signOut() berfungsi untuk melakukan sign out pada user atau admin , dengan cara menggunakan Session::flush(); dan Auth::logout();
kemudian di redirect return ke ‘/login’

Folder Controllers - IndexController

IndexController adalah controller yang berfungsi untuk menghandle View Home dan Artikel Details

serta menhandle search query

```
use App\Models\Category;
use App\Models\ArticleCategory;
use App\Models\Article;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Str;
use Hashids\Hashids;
use Illuminate\Http\Request;
```

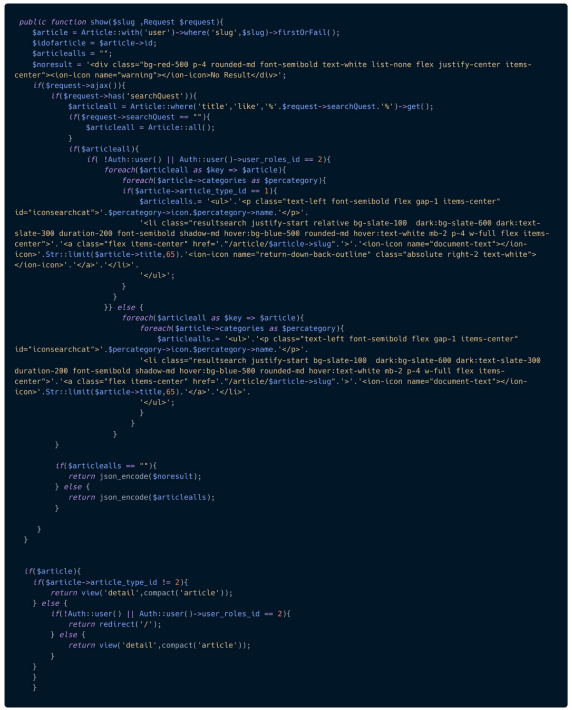
Use juga beberapa utility dan models . seperti Str, Request, Auth dan model (Category , ArticleCategory , Article)

Folder Controllers - IndexController

```
public function index(Request $request){
        $articlealls = "";
        $noresult = '<div class="bg-red-500 flex items-center justify-center p-4 rounded-md font-semibold text-white
        list-none"><ion-icon name="warning"></ion-icon>No Result For "'.$request->searchQuest.'"</div>';
        if($request->ajax()){
           if($request->has('searchQuest')){
            $articleall = Article::where('title','like','%'.$request->searchQuest.'%')->get();
            if($request->searchQuest == ""){
                $articleall = Article::all();
              }
              if($articleall){
                if( !Auth::user() || Auth::user()->user_roles_id == 2){
                    foreach($articleall as $key => $article){
                        foreach($article->categories as $percategory){
                        if($article->article_type_id == 1){
                            $articlealls.= '<ul>'.'<p class="text-left font-semibold flex gap-1 items-center"
                            id="iconsearchcat">'.$percategory->icon.$percategory->name.'</p>'.
                            '<li class="resultsearch justify-start relati`ve bg-slate-100  dark:bg-slate-600
                            dark:text-slate-300 duration-200 font-semibold shadow-md hover:bg-blue-500 rounded-md
                            hover:text-white mb-2 p-4 w-full flex items-center">'.'<a class="flex items-center"
                            href='."/article/$article->slug".'>'.'<ion-icon name="document-text"></ion-
                            icon>'.Str::limit($article->title,65).'<ion-icon name="return-down-back-outline"
                            class="absolute right-2 text-white"></ion-icon>'.'</a>'.'</li>'.
                            '</ul>';
                        }
                    }
                }else {
                    foreach($articleall as $key => $article){
                        foreach($article->categories as $percategory){
                            $articlealls.= '<ul>'.'<p class="text-left font-semibold flex gap-1 items-center"
                            id="iconsearchcat">'.$percategory->icon.$percategory->name.'</p>'.
                            '<li class="resultsearch justify-start bg-slate-100  dark:bg-slate-600 dark:text-
                            slate-300 duration-200 font-semibold shadow-md hover:bg-blue-500 rounded-md
                            hover:text-white mb-2 p-4 w-full flex items-center">'.'<a class="flex items-center"
                            href='."/article/$article->slug".'>'.'<ion-icon name="document-text"></ion-
                            icon>'.Str::limit($article->title,65).'</a>'.'</li>'.
                        '</ul>';
                        }
                    }
                }
            }

        if($articlealls == ""){
            return json_encode($noresult);
        } else {
            return json_encode($articlealls);
        }

    }}
    else {
    $articleall = Article::latest()->paginate(9);
    }

    return view('home',compact('articleall'));


    }
```

di dalam function index terdapat return view untuk halaman index , dan terdapat live search dengan cara mengirimkan value menggunakan ajax ke dalam request controller , terdapat juga kondisi untuk menyatakan apabila ajax kosong maka tampilkan semua

Folder Controllers - IndexController (Show)



```
public function show($slug ,Request $request){
    $article = Article::with('user')->where('slug',$slug)->firstOrFail();
    $idofarticle = $article->id;
    $articlealls = "";
    $noresult = '<div class="bg-red-500 p-4 rounded-md font-semibold text-white list-none flex justify-center items-
center"><lon-icon name="warning"></lon-icon>No Result</div>';
    if($request->ajax()){
        if($request->has('searchQuest')){
            $articleall = Article::where('title','like','%'.$request->searchQuest.'%')->get();
            if($request->searchQuest == ""){
                $articleall = Article::all();
            }
            if($articleall){
                if( !Auth::user() || Auth::user()->user_roles_id == 2){
                foreach($articleall as $key => $article){
                    foreach($article->categories as $percategory){
                    if($article->article_type_id == 1){
                            $articlealls.= '<ul>'.'<p class="text-left font-semibold flex gap-1 items-center"
id="iconsearchcat">'.$percategory->icon.$percategory->name.'</p>'.
                                '<li class="resultsearch justify-start relative bg-slate-100  dark:bg-slate-600 dark:text-
slate-300 duration-200 font-semibold shadow-md hover:bg-blue-500 rounded-md hover:text-white mb-2 p-4 w-full flex items-
center">'.'<a class="flex items-center" href="./article/$article->slug".'">'.'<ion-icon name="document-text"></lon-
icon>'.Str::limit($article->title,65).'<ion-icon name="return-down-back-outline' class="absolute right-2 text-white">
</lon-icon>'.'</a>'.'</li>'.'</ul>';
                    }
                }
                }} else {
                    foreach($articleall as $key => $article){
                        foreach($article->categories as $percategory){
                            $articlealls.= '<ul>'.'<p class="text-left font-semibold flex gap-1 items-center"
id="iconsearchcat">'.$percategory->icon.$percategory->name.'</p>'.
                                '<li class="resultsearch justify-start bg-slate-100  dark:bg-slate-600 dark:text-slate-300
duration-200 font-semibold shadow-md hover:bg-blue-500 rounded-md hover:text-white mb-2 p-4 w-full flex items-
center">'.'<a class="flex items-center" href="./article/$article->slug".'">'.'<ion-icon name="document-text"></lon-
icon>'.Str::limit($article->title,65).'</a>'.'</li>'.
                                '</ul>';
                        }
                    }
                }
        }

        if($articlealls == ""){
            return json_encode($noresult);
        } else {
            return json_encode($articlealls);
        }

    }
}


if($article){
    if($article->article_type_id != 2){
        return view('detail',compact('article'));
    } else {
        if(!Auth::user() || Auth::user()->user_roles_id == 2){
            return redirect('/');
        } else {
            return view('detail',compact('article'));
        }
    }
}
```

Function Show berfungsi untuk menghandle article sesuai slug dan mengembalikan View

juga terdapat script untuk live searching ajax untuk mencari suatu artikel

Folders Controller - ArticleController

Article controller digunakan untuk menghandle CRUD dari Article seperti create article , delete article, update article dll

```
use App\Models\Article;
use App\Models\ArticleType;
use App\Models\ArticleCategory;
use Illuminate\Support\Str;
use Illuminate\Support\Facades\Storage;
use Illuminate\Support\Facades\Auth;
use Illuminate\Http\Request;
use App\Models\Category;
use
RealRashid\SweetAlert\Facades\Alert;
```

Use terlebih dahulu beberapa utility seperti Str, Storage,Auth,Request

Folders Controller - ArticleController (Index)

Function Untuk Mereturn view dari tambah artikel

```
public function index(){
        if(!Auth::user() || Auth::user()->user_roles_id != 1){
            return redirect('/404');

        } else {
            $category = Category::all();
            $articletype = ArticleType::all();

   return view('tools.addarticle',compact('category','articletype'));

        }
    }
```

ditambahkan juga middleware untuk mencegah user biasa masuk melalui url

Folders Controller - ArticleController (Create)

Function create berfungsi untuk menhandle create article , melalui $request value yang di isi oleh user akan ditangkap lalu dimasukan ke variable dan mengakses model untuk create artikel baru jika semuanya telah berhasil maka akan me-redirect kembali ke page sebelumnya

```php
public function create(Request $request)
    {
          $request->validate([
            'image' => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:8024',
          ]);

          if($request->hasFile('image')){
              $images = $request->file('image');
              $filename = $images->getClientOriginalName();
              $request->image->storeAs('thumbnail',$filename);

              $categoryall = [];
              $category = $request->category;


          $title = $request->title;
          $userid = Auth::user()->id;



        $article = Article::create([
            "user_id" => $userid,
            "title" =>  $title,
            "article_type_id" => $request->typepost,
            "images" => $filename,
            "slug" => Str::slug($title,'-'),
            "description" => $request->editor,
        ]);
           $article->categories()->attach($request->category);

        Alert::success('Artikel Terbuat');
    }

        return redirect('/dashboard');

    }
```

Folders Controller - ArticleController (Edit)

```
  public function edit(Article $article , $id)
    {
      $articletype = ArticleType::all();
      $category = Category::all();
      $editarticle = Article::findOrFail(decrypt($id));


        return view('tools.editarticle',compact('editarticle','category' ,
'articletype'));

    }
```

Function edit berfungsi untuk menghandle view dan return value atau PUT dari $id yang dikirim ke controller melalui url , value ini akan ditaruh ke input sebagai old value

Folders Controller - ArticleController (Update)

function update berfungsi untuk menghandle request dari view edit berdasarkan dari $id yang dikirimkan ke controller lalu melakukan query pada model menggunakan function findOrFail lalu mengupdate jika sukses tampilkan alert sukses dan redirect kembali ke dashboard

```php
public function update(Request $request,$id)
    {
        $decrypted_id = decrypt($id);
        $title = $request->title;
        $typeposts = $request->typepost;
        $description = $request->editor;

         Article::findOrFail($decrypted_id)->update([
                "title" => $title,
                "article_type_id" => $typeposts,
                "slug" => str::slug($title,'-'),
                "description" => $description,

        ]);

        Alert::success('Artikel Terupdate');

        return redirect('/dashboard');
    }
```

Folder Controller - ArticleController (Destroy)

Function Destroy berfungsi untuk menghapus artikel sesuai dengan $id yang dikirim ke controller kemudian akan men-Unlink gambar supaya gambar juga hilang pada storage

jika sudah berhasil akan mereturn redirect kembali ke halaman sebelumnya

```
public function destroy(Article $article, $id)
  {
         $articleToDelete = Article::findOrFail(decrypt($id));
          $articleImagePath = $articleToDelete->images;
          if(!unlink(storage_path('app/public/thumbnail/'.$articleImagePath))){
              $articleToDelete->delete();
          } else {
              $articleToDelete->delete();
              Storage::delete($articleToDelete->images);
          }

        return redirect()->back();

  }
```

Folders Controller - ArticleController (Category)

Function category berfungsi untuk menhandle category detail dari setiap $id yang dikirimkan melalui href link pada sidebar dan dikirimkan ke controller lalu ditangkap querynya menggunakan WHERE() function , terdapat juga live search menggunakan ajax pada function tersebut

```
    public function category($slug, Request $request){
        $articlealls = "";
        $noresult = '<div class="bg-red-500 flex items-center justify-center p-4
                    rounded-md font-semibold text-white list-none"><ion-icon
                    name="warning"></ion-icon>No Result For "'.$request-
          >searchQuest.'"</div>';
        if($request->ajax()){
           if($request->has('searchQuest')){
             $articleall = Article::where('title','like','%'.$request-
          >searchQuest.'%')->get();
              if($request->searchQuest == ""){
                $articleall = Article::all();
              }
              if($articleall){
                 if( !Auth::user() || Auth::user()->user_roles_id == 2){
                    foreach($articleall as $key => $article){
                        foreach($article->categories as $percategory){
                        if($article->article_type_id == 1){
                            $articlealls.= '<ul>'.'<p class="text-left font-semibold
flex gap-1 items-center" id="iconsearchcat">'.$percategory->icon.$percategory-
>name.'</p>'.
                                '<li class="resultsearch justify-start relative bg-slate-
100  dark:bg-slate-600 dark:text-slate-300 duration-200 font-semibold shadow-md
hover:bg-blue-500 rounded-md hover:text-white m0-2 p-4 w-full flex items-center">'.'<a
class="flex items-center" href="./article/$article->slug".">'.'<ion-icon
name="document-text"></ion-icon>'.Str::limit($article->title,65).'<ion-icon
name="return-down-back-outline" class="absolute right-2 text-white"></ion-
icon>'.'</a>'.'</li>'.
                                '</ul>';
                        }
                    }
                 }else {
                    foreach($articleall as $key => $article){
                        foreach($article->categories as $percategory){
                            $articlealls.= '<ul>'.'<p class="text-left font-
semibold flex gap-1 items-center" id="iconsearchcat">'.$percategory-
>icon.$percategory->name.'</p>'.
                                '<li class="resultsearch justify-start bg-slate-100
dark:bg-slate-600 dark:text-slate-300 duration-200 font-semibold shadow-md hover:bg-
blue-500 rounded-md hover:text-white m0-2 p-4 w-full flex items-center">'.'<a
class="flex items-center" href="./article/$article->slug".">'.'<ion-icon
name="document-text"></ion-icon>'.Str::limit($article->title,65).'</a>'.'</li>'.
                                '</ul>';
                        }
                    }
                 }
              }
        }

       if($articlealls == ""){
          return json_encode($noresult);
       } else {
          return json_encode($articlealls);
       }

    })
    else {

            $categories = Category::with('articles')->where("slug",$slug)->get();
    }
            return view("category",compact("categories"));
        }
```

Folders Controller - IotController (Index)

Function index() berfungsi untuk menghandle view dari Iot Index ,

untuk fungsi fungsi dari iot tidak dihandle oleh controller melainkan oleh view itu sendiri karena controller tidak bisa menghandle javascript

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class IotController extends Controller
{
      public function index(){


         return view('tools.Iot.iotindex');

      }

}
```

Folders Controller - UserAuthController

UserAuthController berfungsi untuk menghandle User Biasa untuk login dan register

use dulu beberapa utility dan model seperti Auth,Str, dan Model User

```
namespace App\Http\Controllers;
use Illuminate\Support\Facades\Auth;
use Illuminate\Http\Request;
use Illuminate\Support\Str;
use App\Models\User;
```

Folders Controller - UserAuthController

function index untuk menhandle view dari login user page ,

function auth mengambil request dari input pada page user lalu melakukan validasi data dan membuat kondisi attemptjika berhasil akan direturn redirect ke /

```php
public function index(){
        return view('Auth.userlogin');
    }

    public function auth(Request $request){
        $request->validate([
            'email' => 'required',
            'password' => 'required',
        ]);

        $credentials = $request->only('email','password');
        if(Auth::attempt($credentials)) {
            return redirect()->intended('/')->with('success','Berhasil Login');
        } else {
            return redirect('/login')->with('error','Email Atau Password Salah');
        }
    }
```

Folders Controller - UserAuthController

function registerindex mereturn view dari page register user

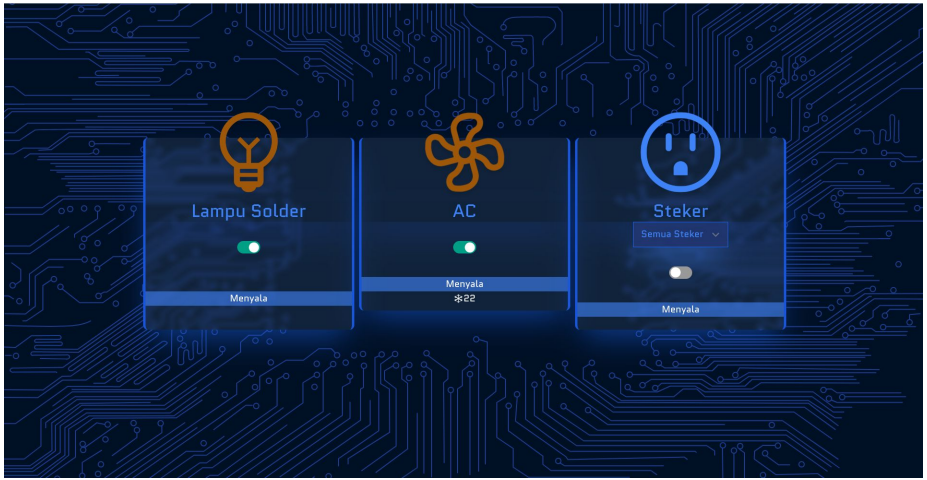
function register mengambil request value dari user input pada page dan melakukan create user setelah itu akan otomatis meloginkan user yang baru saja didaftarkan

jika berhasil return redirect ke home

```php
public function registerindex(){
   return view('Auth.userregister');
 }

 public function register(Request $request){
   $users = User::create([
        "name" => $request->username,
        "slug" => Str::slug($request->username),
        "user_roles_id" => 2,
        "email" => $request->email,
        "password" => bcrypt($request->password),

   ]);

   auth()->login($users);

  return redirect('/');
 }
```

Iot Tools Pages

Iot Tools Page terdapat pada dashboard agar dapat mengaksesnya admin harus login terlebih dahulu , dibuat menggunakan Javascript Dan MQTT



Iot Tools Code

Lakukan Connect terlebih dulu

dengan menuliskan clientId,host,port

lalu dikoneksikan menggunakan function Client

pada class Paho.MQTT.Client

masukan pada 1 variable global

Iot Tools Code

Pada function onConnect

tentukan dulu topic yang akan di subscribe

```
function onConnect2() {
    topic = "/Cikunir";
    client2.subscribe("lt2/suhu2/sharp")
    client2.subscribe("cmnd/Main_4")
    client2.subscribe("lt2/stts/suhu2/sharp")
    client2.subscribe("Cikunir/lt2/stts2/sharp")
    client2.subscribe("lt2/suhu2")

}

function onConnect() {
    topic = "/tele";
    client1.subscribe("cmnd/lampu_solder/power")

 client1.subscribe("stat/lampu_solder/RESULT")
}
```

Iot Tools Code

Ketika Kondisi checked jangan lupa untuk set localStorage item agar checked value sebelumnya dapat tersimpan dan tidak hilang ketika di reload

```
ckbox.addEventListener('change', () => {
    if (ckbox.checked) {
        message = new Paho.MQTT.Message("1");
        message.destinationName = "cmnd/lampu_solder/power";
        client1.send(message);
        console.log(message.payloadString)
        console.log("checked")
        $('#lampusoldericon').removeClass('text-blue-500')
        $('#lampusoldericon').addClass('text-yellow-600')
        $('#lampusoldericon').addClass('animate-pulse')
        window.localStorage.setItem('lampu_solder',
'menyala');
    } else {
        message = new Paho.MQTT.Message("0");
        message.destinationName = "cmnd/lampu_solder/power";
        client1.send(message);
        console.log("not checked")
        $('#lampusoldericon').removeClass('text-yellow-600')
        $('#lampusoldericon').addClass('text-blue-500')
        $('#lampusoldericon').removeClass('animate-pulse')
        window.localStorage.setItem('lampu_solder', 'mati');
    }
})


ckboxac.addEventListener('change', () => {
    if (ckboxac.checked) {
        message = new Paho.MQTT.Message("1");
        message.destinationName = "Cikunir/lt2/suhu2/sharp";
        client2.send(message)
        window.localStorage.setItem('ac', 'menyala');
        $('#acicon').addClass('animate-spin')
        $('#acicon').removeClass('text-blue-500')
        $('#acicon').addClass('text-yellow-600')
    } else {
        message = new Paho.MQTT.Message("2");
        message.destinationName = "Cikunir/lt2/suhu2/sharp";
        client2.send(message)
        window.localStorage.setItem('ac', 'mati');
        $('#acicon').removeClass('animate-spin')
        $('#acicon').removeClass('text-yellow-600')
        $('#acicon').addClass('text-blue-500')
    }
})
```

Iot Tools Code

function onMessageArrived berfungsi untuk menerima setiap message yang masuk ke mqtt

-tambahkan kondisi ketika destination topic sesuai yang kita inginkan lalu kita set LocalStorage untuk meng-check atau meng-unchek checkbox agar checkbox value tersimpan dan tidak hilang ketika url di reload

```
function onMessageArrived(message) {
    console.log("onMessageArrived: " + message.payloadString);
    console.log("topic: " + message.destinationName);

    if(message.destinationName == "stat/lampu_solder/RESULT"){
        let data = JSON.parse(message.payloadString)
        if(data.POWER == "ON"){
            window.localStorage.setItem('lampu_solder', 'menyala');
            ckbox.checked = true
            $('#lampusoldericon').removeClass('text-blue-500')
            $('#lampusoldericon').addClass('text-yellow-500')
            $('#lampusoldericon').addClass('animate-pulse')
            $('#status_lampu').text("Menyala" );
        } else {
            window.localStorage.setItem('lampu_solder', 'mati');
            ckbox.checked = false
            $('#lampusoldericon').removeClass('text-yellow-500')
            $('#lampusoldericon').addClass('text-blue-500')
            $('#lampusoldericon').removeClass('animate-pulse')
            $('#status_lampu').text("Mati" );
        }

    } else if(message.destinationName == "Cikunir/lt2/stts2/sharp")
{       console.log(message.payloadString)
        let data = JSON.parse(message.payloadString)
        localStorage.setItem("suhuac",data);
        $("#suhuac").text(data)
        console.log("suhu masuk")
    }

}
```

Iot Tools Code

Validasi LocalStorage dengan cara mendapatkan getItem tambahkan kondisi jika sesuai , chekbox akan diberi boolean true or false dan juga akan mentoggle class untuk stylingnya

```
if (localStorage.getItem("suhuac")){
    $("#suhuac").text(localStorage.getItem("suhuac"))
}

if (localStorage.getItem('lampu_solder') == "menyala")
{   ckbox.checked = true
    $('#lampusoldericon').removeClass('text-blue-500')
    $('#lampusoldericon').addClass('text-yellow-600')
    $('#lampusoldericon').addClass('animate-pulse')
    $('#status_lampu').text("Menyala" );
} else {
    ckbox.checked = false
    $('#lampusoldericon').removeClass('text-yellow-
600'$('#lampusoldericon').addClass('text-blue-500')
    $('#lampusoldericon').removeClass('animate-pulse')
    $('#status_lampu').text("Mati" );
}

if (localStorage.getItem('ac') == "menyala") {
    ckboxac.checked = true
    $('#acicon').removeClass('text-blue-500')
    $('#acicon').addClass('text-yellow-600')
    $('#acicon').addClass('animate-spin')

} else {
    ckboxac.checked = false
    $('#acicon').removeClass('text-yellow-600')
    $('#acicon').addClass('text-blue-500')
    $('#acicon').removeClass('animate-spin')
}
```

Iot Tools Code

jika ada perubahan pada element dengan id pilihsteker lalu diberi kondisi jika mengandung value tertentu lalu didalamnya diberi kondisi lagi jika tercheck kirimkan message pada topic cmnd

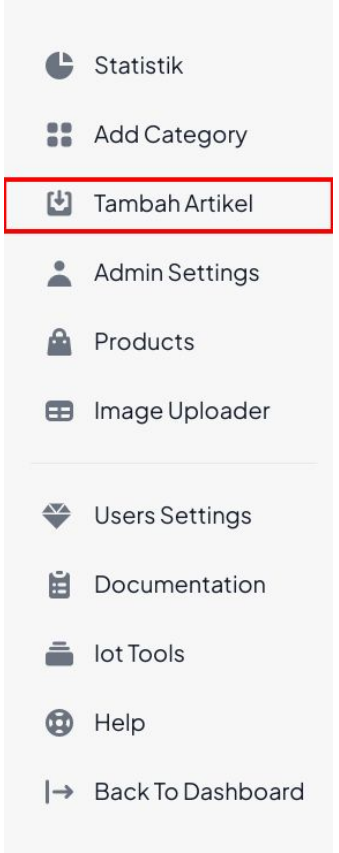
0 untuk mematikan , 1 untuk menyalakan

```
let cksteker = document.querySelector("#stekerslider")

$("#pilihsteker").on("change", function() {
    cksteker.addEventListener('change', () => {
        if (cksteker.checked) {
            if ($("#pilihsteker").val() == "power-4") {
                message = new Paho.MQTT.Message("1");
                message.destinationName =
"cmnd/Main_4/POWER4";mqt2.send(message)
            } else if ($("#pilihsteker").val() == "power-2") {
                message = new Paho.MQTT.Message("1");
                message.destinationName =
"cmnd/Main_4/POWER2";mqt2.send(message)
            } else if ($("#pilihsteker").val() == "all"){
                message = new Paho.MQTT.Message("1");
                message.destinationName =
"cmnd/Main_4/POWER4";mqt2.send(message1)
                message.destinationName =
"cmnd/Main_4/POWER2";mqt2.send(message2)
            }
        } else {
            if ($("#pilihsteker").val() == "power-4") {
                message = new Paho.MQTT.Message("0");
                message.destinationName =
"cmnd/Main_4/POWER4";mqt2.send(message)
            } else if ($("#pilihsteker").val() == "power-2") {
                message = new Paho.MQTT.Message("0");
                message.destinationName =
"cmnd/Main_4/POWER2";mqt2.send(message)
            } else if ($("#pilihsteker").val() == "all"){
                message = new Paho.MQTT.Message("0");
                message.destinationName =
"cmnd/Main_4/POWER4";mqt2.send(message1)
                message.destinationName =
"cmnd/Main_4/POWER2";mqt2.send(message2)

            }
        }
    })


});
```

Penggunaan Tools Pada Dashboard

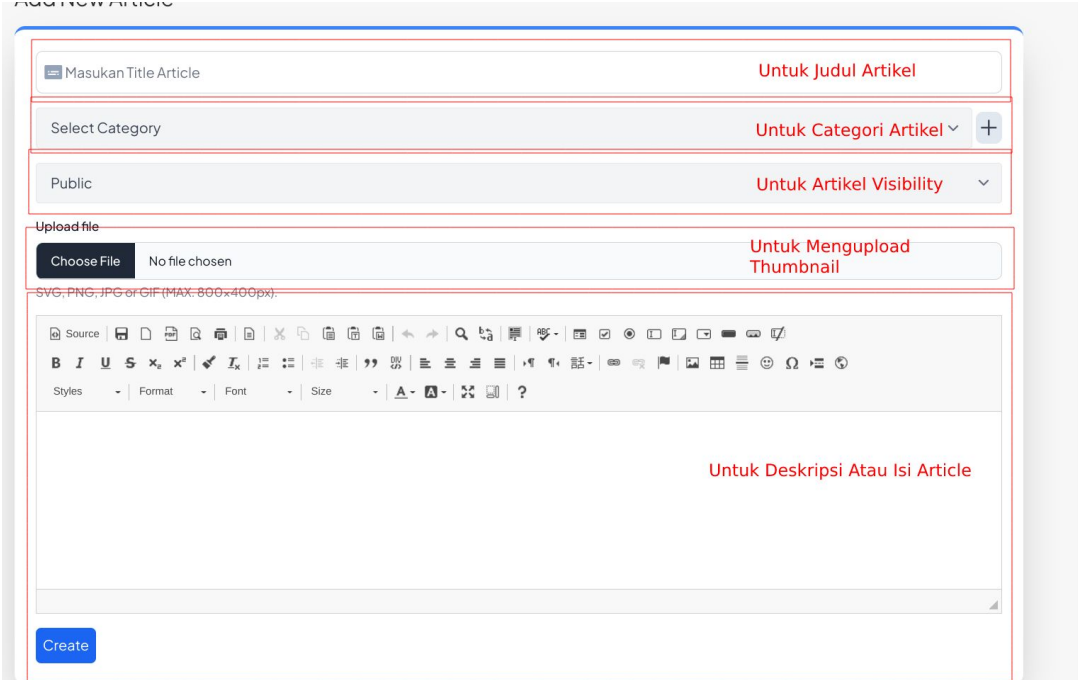
Cara Menambahkan Article

-klik tambahartikel pada sidebar



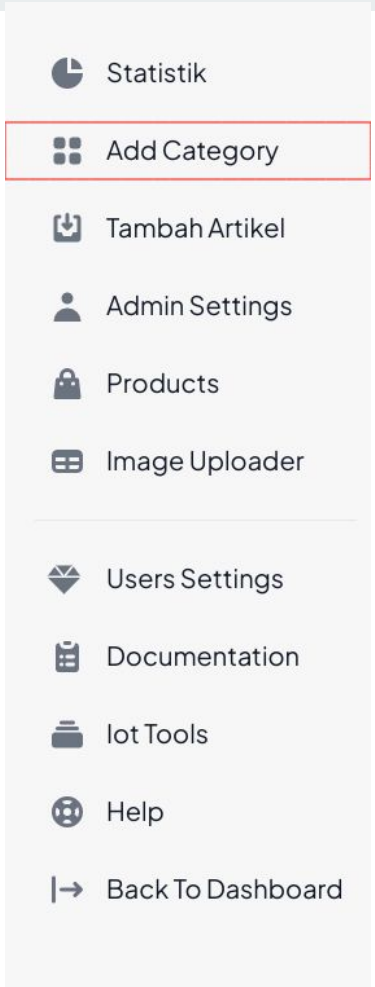
Cara Menambahkan Artikel

Jika Sudah Selesai Klik Create



Cara Menambahkan Category

Klik pada addCategory



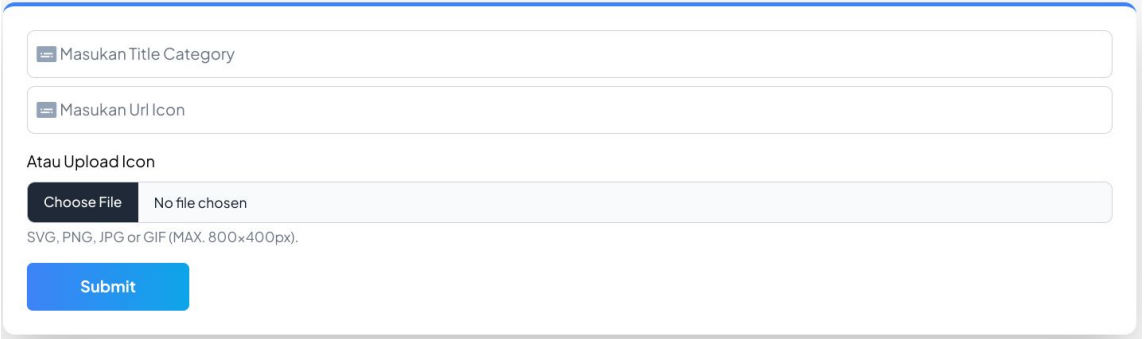
Cara Menambahkan Category

Masukan Title Category

Pilih Salah satu metode upload icon

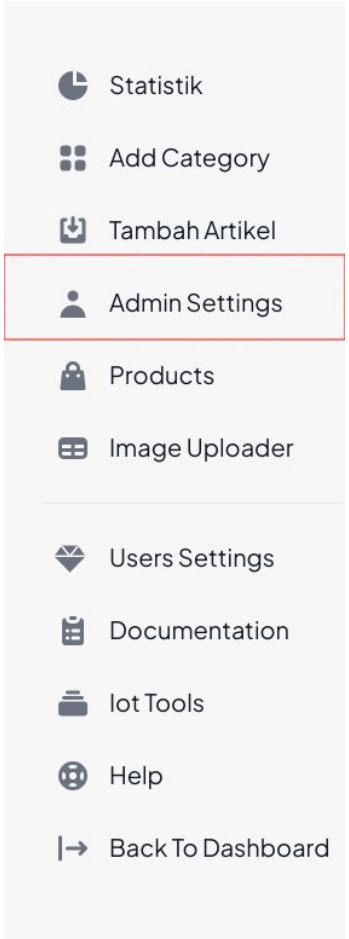
misal dari local file, atau url gambar

jangan pilih keduanya sekaligus



Cara Menambahkan Admin Baru

Klik Admin Settings pada sidebar dashboard



Cara menambahkan admin baru

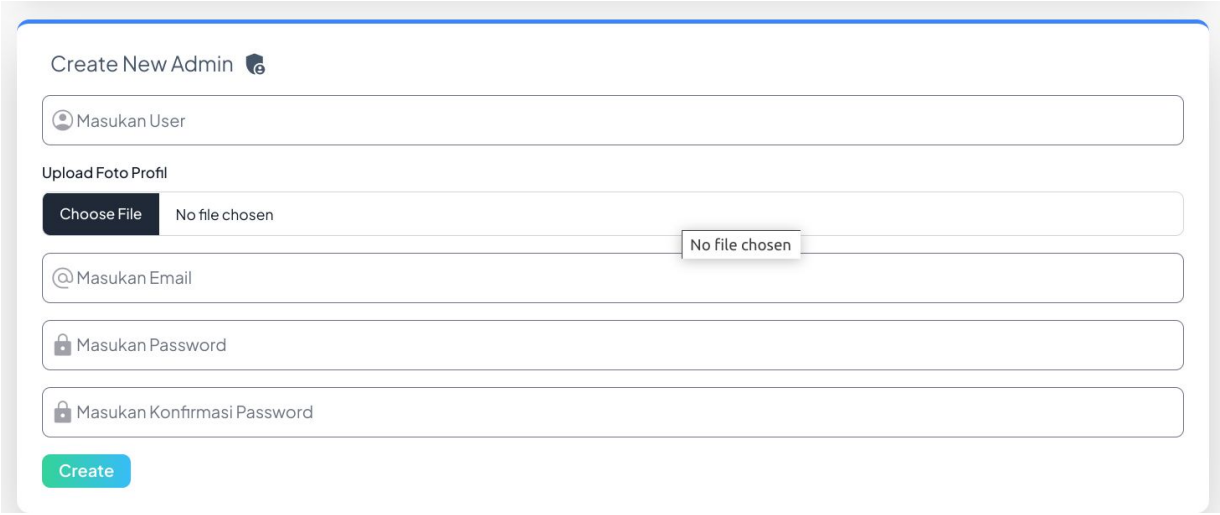
Jika sudah selesai Klik Create

-foto profil harus diupload

agar dapat disubmit

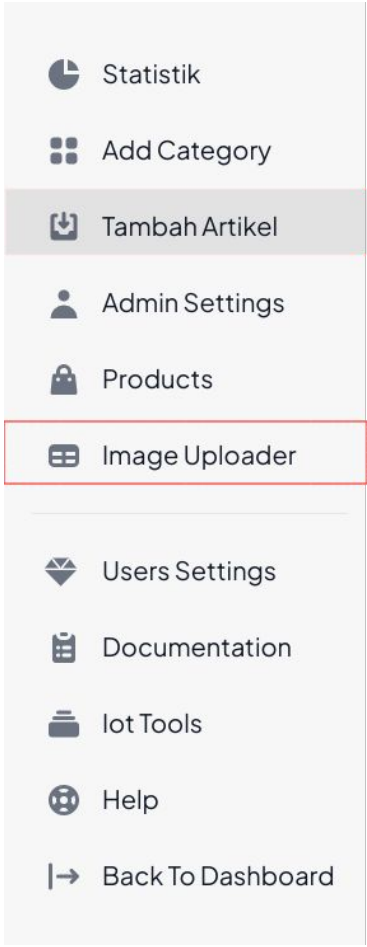
-setelah itu dapat login dengan

admin tersebut yang baru dibuat



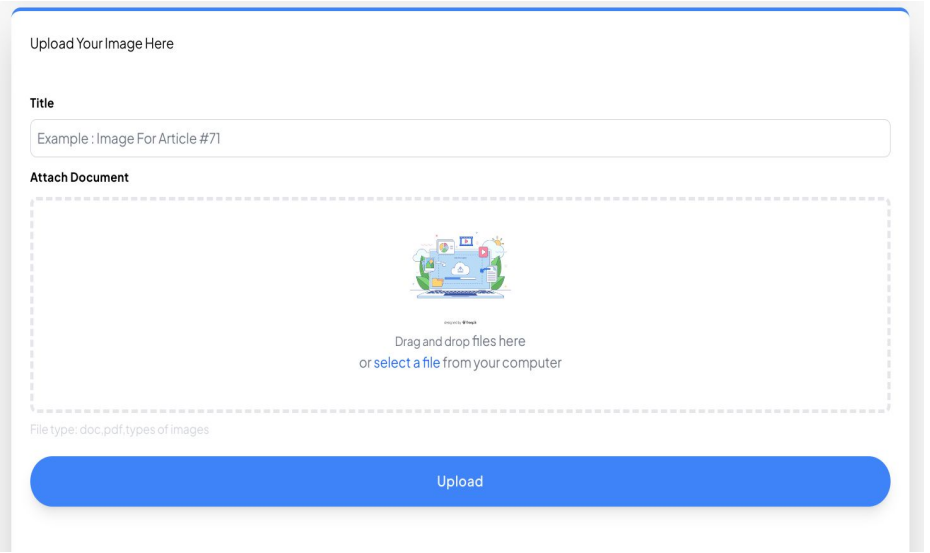
Cara Menguploud gambar ke storage website

Klik ImageUploader pada sidebar dashboard



Cara Menguploadgambar ke storage website

Tambahkan title dan bisa mendrag and drop gambar yang akan di upload , bisa juga dengan mengklik select a file lalu klik tombol upload



Cara Copy Url Dan Menghapus Gambar

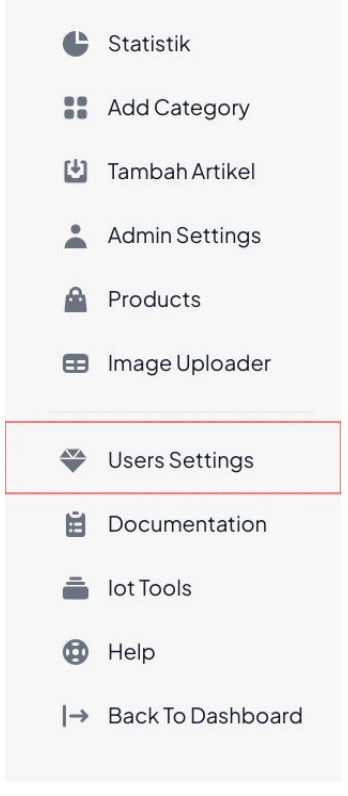
Id	Title	Url	Created_at	Size
1	img wibuh luh	http://127.0.0.1:8000/stora	2022-11-28 16:42:54	0.06 Mb

untuk menyalin url gambar

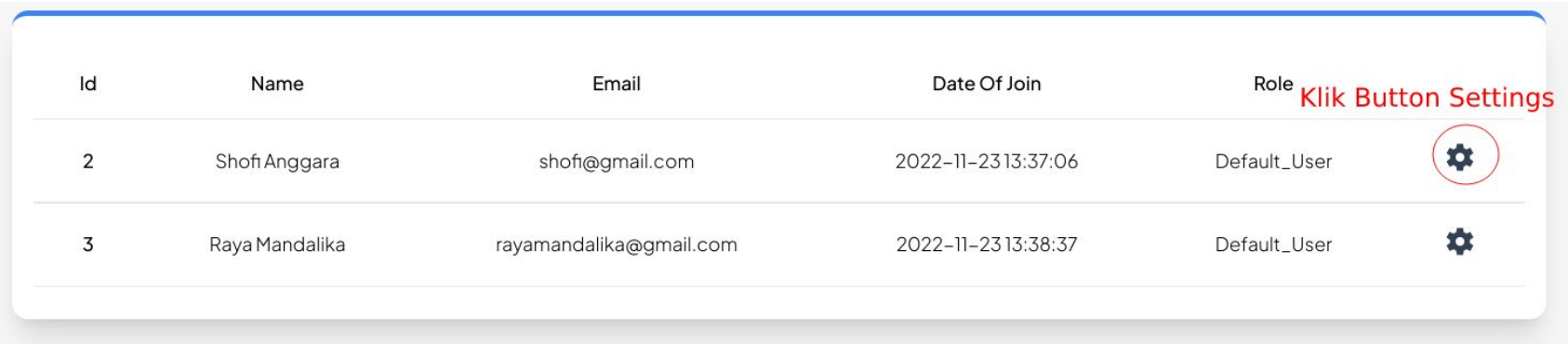
Untuk Menghapus gambar

Membuat User Dapat Membaca Postingan Private

Klik ‘Users Settings’ pada sidebar



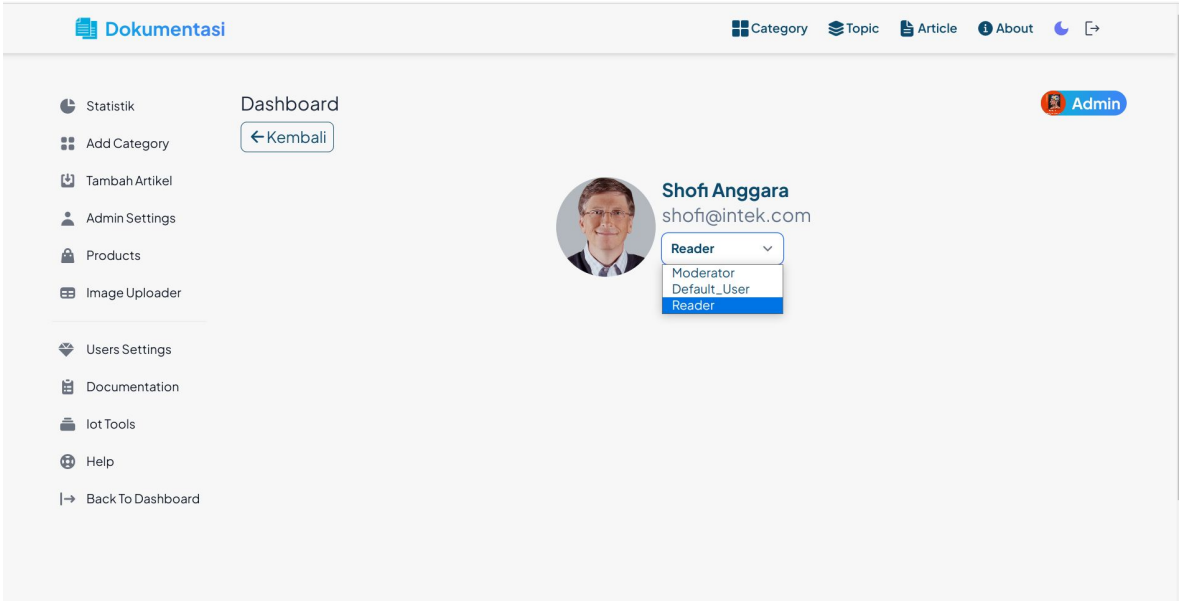
Membuat User Dapat Membaca Postingan Private



Membuat User Dapat Membaca Postingan Private

Klik Pada Option Select

Jadikan User Sebagai Reader



Selesai

03 Oktober 2022 - 1 Desember 2022