

LAMPIRAN

```

<?php
namespace App\Exports;
use
Maatwebsite\Excel\Concerns\FromCollection;
use
    Maatwebsite\Excel\Concerns\WithHeadings;
use
    Maatwebsite\Excel\Concerns\ShouldAutoSize;
use
    Maatwebsite\Excel\Concerns\WithStyles;
use
    PhpOffice\PhpSpreadsheet\Worksheet\Worksheet;
use App\Models\Student;
use Carbon\Carbon;

class DashboardExport implements
    FromCollection, WithHeadings,
    ShouldAutoSize, WithStyles
{
    *                               @return
    \Illuminate\Support\Collection
    */
    public function collection()
    {
        $student = Student::latest()->get();
        return $student->map(function ($item) {
            return [
                'registration_number' =>
                    $item->registration_number,
                'status' => $item->status ?
                    ($item->status == 1 ? 'Lolos' :
                    'Tidak Lolos') : 'Ditinjau',
                'type' => $item->type == 1
                    ? 'Zonasi' : 'Prestasi',
                'fullname' => $item->fullname,
                'nickname' => $item->nickname,
                'gender' => $item->gender
            ];
        });
    }
}

```

```

== 1 ? 'Laki-laki' : 'Perempuan',
        'pob' => $item->pob,
        'dob' => $item->dob,
        'religion' => $item->religion,
        'citizenship' => $item->citizenship,
        'birth_order' => $item->birth_order,
        'total_sibling' => $item->total_sibling,
        'total_step_sibling' => $item->total_step_sibling,
        'total_foster_sibling' => $item->total_foster_sibling,
        'language' => $item->language,
        'address' => $item->address,
        'phone' => $item->phone,
        'residence' => $item->residence == 1 ? 'Orang tua' :
        ($item->residence == 2 ? 'Menumpang pada orang lain' :
        'Asrama'),
        'distance_in_meters' => $item->distance_in_meters,
        'transportation' => $item->transportation == 1 ?
        'Kendaraan Umum' : ($item->transportation == 2 ?
        'Kendaraan Pribadi' : 'Jalan Kaki'),
        'distance_in_minutes' => $item->distance_in_minutes,
        'weight_in_kg' => $item->weight_in_kg,
        'height_in_cm' => $item->height_in_cm,
        'blood_type' => $item->blood_type,
        'tanggal_pendaftaran' => Carbon::parse($item->created_at)->locale('id')->format('Y-m-d H:i:s'),
    ];
}

```

```

    })->values();
}
public function headings(): array
{
    return [
        'Nomor Pndaftaran',
        'Status',
        'Jalur',
        'Nama Lengkap',
        'Nama Panggilan',
        'Jenis Kelamin',
        'Tempat Lahir',
        'Tanggal Lahir',
        'Agama',
        'Kewarganegaraan',
        'Anak Keberapa',
        'Jumlah Saudara Kandung',
        'Jumlah Saudara Tiri',
        'Jumlah Saudara Angkat',
        'Bahasa Sehari-hari',
        'Alamat',
        'No. Telepon',
        'Tempat Tinggal',
        'Jarak Tempat Tinggal
(meter)',
        'Kendaraan',
        'Jarak Tempuh (menit)',
        'Berat Badan (kg)',
        'Tinggi Badan (cm)',
        'Golongan Darah',
        'Tanggal Pendaftaran'
    ];
}
public function styles(Worksheet
$sheet)
{
    return [
        // Style the first row as bold
        text.
        1 => ['font' => ['bold' =>
true]],
    ];
}
}
<?php

```

```

// use App\Models\Category;

use App\Models\Setting;
use App\Models\Student;
use Illuminate\Support\Carbon;
use Illuminate\Support\Facades\File;
use
    Illuminate\Support\Facades\Stor
age;
use
    Intervention\Image\Facades\Ima
ge;
function resize_image($file, $width,
$height, $path)
{
    $img = Image::make($file-
>path());
    $img->resize($width, $height,
function($constraint) {
        $constraint->aspectRatio();
    })->save($path);

    return $img;
}
function delete_image($image)
{
    if (Storage::exists($image)) {
        Storage::delete($image);
    }
    return true;
}
function make_directory($dir)
{
    if (!File::isDirectory($dir)) {
        File::makeDirectory($dir);
    }
    return true;
}
function delete_directory($dir)
{
    if (File::isDirectory($dir)) {
        File::deleteDirectory($dir);
    }
    return true;
}
function dateToIndo($date)

```

```

{
    $time = Carbon::parse($date)-
        >locale('id')->isoFormat('dddd,
            D MMMM Y');
    return $time;
}
function getSetting()
{
    return Setting::find(1);
}
function getStudentCount()
{
    $data =
        Student::selectRaw("IF(type = 1,
            'Zonasi', 'Prestasi') as type,
            count(type) as total")-
            >groupBy('type')->get();
    return collect([
        'Zonasi' => $data->where('type',
            'Zonasi')->value('total') ?? 0,
        'Prestasi' => $data-
            >where('type', 'Prestasi')-
            >value('total') ?? 0,
    ]);
}
function getLineChartData()
{
    $days = [];
    $total_zonasi = [];
    $total_prestasi = [];
    for ($i=0; $i < 7; $i++) {
        $day = today()->subDays($i)-
            >format('Y-m-d');
        $zonasi =
            Student::whereDate('created_at',
                $day)->where('type', 1)-
                >count();
        $prestasi =
            Student::whereDate('created_at',
                $day)->where('type', 2)-
                >count();
        // $days[] = $day;
        $days[] = today()->subDays($i)-
            >locale('id')->isoFormat('D
            MMM');
        $total_zonasi[] = $zonasi;
        $total_prestasi[] = $prestasi;
    }
    $days = array_reverse($days);
    $total_zonasi =
        array_reverse($total_zonasi);
    $total_prestasi =
        array_reverse($total_prestasi);
    return collect([
        'data' => [
            collect([
                'name' => 'Zonasi',
                'data' => $total_zonasi
            ]),
            collect([
                'name' => 'Prestasi',
                'data' => $total_prestasi
            ])
        ],
        'categories' => $days
    ]);
}
function getBarChartData()
{
    $days = [];
    $total_zonasi = [];
    $total_prestasi = [];
    for ($i=0; $i < 7; $i++) {
        $day = today()->subDays($i)-
            >format('Y-m-d');
        $zonasi =
            Student::whereDate('created_at',
                $day)->where('type', 1)-
                >count();
        $prestasi =
            Student::whereDate('created_at',
                $day)->where('type', 2)-
                >count();
        // $days[] = $day;
        $total_zonasi[] = ['x' => today()-
            >subDays($i)->locale('id')-
            >isoFormat('D MMM'), 'y' =>
            $zonasi];
        $total_prestasi[] = ['x' =>
            today()->subDays($i)-
            >locale('id')->isoFormat('D
            MMM'), 'y' => $prestasi];
    }
}

```

```

    }
    $data_zonasi = ['name' => 'Zonasi',
        'data' =>
        array_reverse($total_zonasi)];
    $data_prestasi = ['name' =>
        'Prestasi', 'data' =>
        array_reverse($total_prestasi)];
    return collect([$data_zonasi,
        $data_prestasi]);
}
?php
namespace
    App\Http\Controllers\Backend;
use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\ActivityLog;
class ActivityLogController extends
    Controller
{
    public function json(Request
        $request) {
        $data = ActivityLog::query();
        $limit = ($request-
            >iDisplayLength) ? $request-
            >iDisplayLength : $data-
            >count();
        $start = ($request-
            >iDisplayStart) ? $request-
            >iDisplayStart : 0;
        if(isset($start) && $limit != '-1')
        {
            $data = $data->skip($start)-
                >take($limit);
        }
        return datatables()->of($data)
            ->editColumn('created_at',
            function ($row) {
                return $row->created_at-
                    >format('d F Y - H:i:s');
            })
            ->editColumn('causer',
            function ($row) {
                return $row->causer-
                    >name ?? null;
            })
            ->rawColumns(['action'])
            ->addIndexColumn()
            ->make();
    }
    public function index()
    {
        $title = "Activity Log";
        $sub_title = '
            <div class="breadcrumb-
            item">List Activity Log</div>
        ';
        return
            view('backend.activity.index',
                compact('title', 'sub_title'));
    }
}
<?php
namespace
    App\Http\Controllers\Backend;
use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use
    Illuminate\Support\Facades\Aut
    h;
use
    Illuminate\Support\Facades\Has
    h;
use
    Illuminate\Support\Facades\Vali
    dator;
use App\Models\User;
class AuthController extends
    Controller
{
    public function __construct() {
        $this->middleware('guest')-
            >except('logout');
    }
    public function showLogin()
    {
        return
            view('backend.auth.login');
    }
    public function doLogin(Request
        $request)
    {
        $validator =

```

```

Validator::make($request->all(),
[
    "username" => "required",
    "password" => "required"
]);
if ($validator->fails()) {
    return back()->with('error',
$validator->messages()-
>all()[0])->withInput();
}
$user = User::where('username',
$request->username)->first();
if ($user) {
    if (Hash::check($request-
>password, $user->password)) {
        Auth::attempt(['username'
=> $request->username,
'password' => $request-
>password], $request-
>remember ? true : false);
        activity('login')-
>withProperties(['user_agent' =>
$request->header('User-
Agent')])->log('user login');
        return redirect()-
>route('be.index');
    }
    else {
        return back()-
>with("error", "Password
Doesn't Match!")->withInput();
    }
} else {
    return back()->with("error",
"User Not Found!")-
>withInput();
}
}
public function logout(Request
$request)
{
    activity('logout')-
>withProperties(['user_agent' =>
$request->header('User-
Agent')])->log('user logout');
    Auth::logout();
}

```

```

        return redirect()-
>route('be.auth.login');
    }
}
<?php
namespace
    App\Http\Controllers\Backend;
use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\Student;
use App\Models\User;
use Barryvdh\DomPDF\Facade\Pdf;
use
    Maatwebsite\Excel\Facades\Exc
el;
use App\Exports\DashboardExport;
use App\Imports\DashboardImport;

class DashboardController extends
Controller
{
    public function index()
    {
        $title = "Dashboard";
        $sub_title = '
        <div class="breadcrumb-
item">Dashboard</div>
        ';
        $student =
Student::whereYear('created_at',
substr(getSetting()-
>school_year, 0, 4))->get();
        $zonasi = $student-
>where('type', 1)->count();
        $prestasi = $student-
>where('type', 2)->count();
        $operator =
User::role('operator')->count()
        return
view('backend.dashboard.index',
compact('title', 'sub_title',
'zonasi', 'prestasi', 'operator'));
    }
    public function exportPdf()
    {
        $data = Student::latest()->get();
    }
}

```

```

        if ($data->isEmpty()) {
            alert()->error('Error', 'Data
            tidak ditemukan');
            return back();
        }
        view()->share('data',$data);
        $pdf = Pdf::loadView('pdf.dashboard',
        compact('data'))->setPaper('a4',
        'landscape');
        // return view('pdf.dashboard',
        $data);
        return $pdf->download(time().'pdf');
    }
    public function exportExcel()
    {
        return Excel::download(new
        DashboardExport(),
        time().'xlsx');
    }
    public function importExcel(Request $request)
    {
        try {
            Excel::import(new
            DashboardImport, $request->file('file'));
            return response()->json([
                'status' => true,
                'message' => 'Import
                Success',
            ]);
        } catch (Exception $e) {
            return response()->json([
                'status' => false,
                'message' => $e->getMessage(),
            ]);
        }
    }
}
<?php
namespace
    App\Http\Controllers\Backend;
use App\Http\Controllers\Controller;

```

```

use Illuminate\Http\Request;
use App\Models\Student;
use Barryvdh\DomPDF\Facade\Pdf;
use ZipArchive;
use Storage;

class RegistrationController extends
    Controller
{
    public function json(Request
    $request)
    {
        $data = Student::query();

        if ($request->has('gender')) {
            if ($request->input('gender')
            == "1" || $request->input('gender') == "2") {
                $data->where('gender',
                $request->input('gender'));
            } else {
                $data->latest();
            }
        }
        if ($request->has('religion')) {
            $validReligions = ["islam",
            "kristen", "katolik", "hindu",
            "buddha"];
            // Memeriksa apakah nilai
            agama adalah salah satu dari
            beberapa nilai yang diizinkan
            if (in_array($request->input('religion'),
            $validReligions)) {
                // Jika iya, lakukan filter
                berdasarkan agama
                $data->where('religion',
                $request->input('religion'));
            } else {
                // Jika tidak valid, urutkan
                data berdasarkan yang terbaru
                $data->latest();
            }
        }
        $data = $data->get();
        return datatables()->of($data)
    }
}

```

```

->addColumn(
    'action',
    function ($row) {
        $btn = '
            <a href=""
route('be.ppdb.show', $row->id) .
            class="btn btn-info btn-
edit">Detail</a>
        ';
        return $btn;
    }
)
->editColumn('status',
function ($row) {
    switch ($row->status) {
        case '1':
            return <span
class="badge badge-
success">Lulus</span>;
            break;
        case '2':
            return <span
class="badge badge-
danger">Tidak Lolos</span>;
            break;

        default:
            return <span
class="badge badge-
warning">Ditinjau</span>;
            break;
    }
})
->editColumn('type', function
($row) {
    return $row->type == 1 ?
'Zonasi' : 'Prestasi';
})
->editColumn('gender',
function ($row) {
    return $row->gender == 1 ?
'Laki-laki' : 'Perempuan';
})
->editColumn('dob', function
($row) {
    return $row->pob . ' , ' .

```

```

$row->dob->format('d F Y');
    })
->editColumn('religion',
function ($row) {
    return $row->religion;
})
->editColumn('created_at',
function ($row) {
    return $row->created_at-
>format('d F Y - H:i:s');
    })
->rawColumns(['status',
'action'])
->addIndexColumn()
->make();
}
public function index()
{
    $title = "Calon Peserta Didik
Baru";
    $sub_title = '
        <div class="breadcrumb-
item">List Calon Peserta Didik
Baru</div>
    ';
    return
view('backend.ppdb.index',
compact('title', 'sub_title'));
}
public function show($id)
{
    $title = "Calon Peserta Didik
Baru";
    $sub_title = '
        <div class="breadcrumb-
item"><a href=""
route('be.ppdb.index') . "">List
Calon Peserta Didik
Baru</a></div>
        <div class="breadcrumb-
item">Detail</div>
    ';
    $student =
Student::findOrFail($id);
    return
view('backend.ppdb.show',

```

```

compact('title',      'sub_title',
'student'));
}
public function update(Request
$request)
{
    $request->validate([
        'status' => 'required',
    ]);
    $student =
Student::find($request->id);
    if ($request->status == 1) {
        $badge = '<span class="badge
badge-success">Lolos</span>';
        $count_student =
Student::where('type', $student-
>type)->where('status', 1)-
>whereYear('created_at',
substr(getSetting()-
>school_year, 0, 4))->count();
        switch ($student->type) {
            case '1':
                $quota = getSetting()-
>zonasi_quota;
                break;

            default:
                $quota = getSetting()-
>prestasi_quota;
                break;
        }
        if ($count_student >= $quota)
        {
            return response()->json([
                'status' => false,
                'message' => 'Kuota ' .
($student->type == 1 ? 'Zonasi' :
'Prestasi') . ' Penuh',
            ], 422);
        }
    } else {
        $badge = '<span class="badge
badge-danger">Tidak
Lolos</span>';
    }
    $update = $student-
>update(['status' => $request-
>status]);
    if ($update) {
        activity('update')-
>performedOn($student)-
>withProperties(['user_agent' =>
$request->header('User-
Agent')])->log('update status
siswa ke ' . ($request->status ==
1 ? 'lolos' : 'tidak lolos'));
        return response()->json([
            'status' => true,
            'message' => 'Data
Updated',
            'badge' => $badge,
        ]);
    } else {
        return response()->json([
            'status' => false,
            'message' => 'Failed
Update',
        ], 422);
    }
}
public function makeZip(Request
$request)
{
    $request->validate([
        'id' => 'required',
    ]);
    $id = $request->id;
    $student = Student::find($id);
    $zip = new ZipArchive();
    $dir =
public_path('storage/documents/'
. $id);
    make_directory($dir);
    $zip_name = $student-
>registration_number . '.zip';

    if (Storage::disk('public')-
>missing('documents/'
. $zip_name)) {
        view()->share('student',
$student);
        $pdf =

```

```

Pdf::loadView('pdf.student');
$pdf_name = 'FORM-' .
$student->registration_number .
'.pdf';
$pdf->save($dir . '/' .
$pdf_name);
if ($zip->open(public_path('storage/docu
ments/') . $zip_name,
ZipArchive::CREATE |
ZipArchive::OVERWRITE) ===
true) {
    $path =
storage_path('app/public/docum
ents/' . $id);
    $files = new
RecursiveIteratorIterator(new
RecursiveDirectoryIterator($pat
h));
    foreach ($files as $name =>
$file) {
        // We're skipping all
subfolders
        if (!$file->isDir()) {
            $filePath = $file-
>getRealPath();
            // extracting filename
with substr/strlen
            $relativePath =
'documents/' . substr($filePath,
strlen($path) + 1);
            $zip-
>addFile($filePath,
$relativePath);
        }
    }
}
$zip->close();
}
return response()->json([
    'status' => true,
    'message' => 'Download
Success',
    'download_url' =>
route('be.ppdb.download',
$student->registration_number),
]);
}
public function download($file)
{
    return Storage::disk('public')-
>download('documents/' . $file .
'.zip');
}
<?php
namespace
App\Http\Controllers\Backend;
use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use
Illuminate\Support\Facades\Vali
dator;
use App\Models\Setting;
class SettingController extends
Controller
{
    public function index()
    {
        $title = "Pengaturan";
        $sub_title = '
        <div class="breadcrumb-
item">Pengaturan</div>
        ';
        $setting = Setting::find(1);
        return
view('backend.setting.index',
compact('title', 'sub_title',
'setting'));
    }

    public function update(Request
$request)
    {
        $rules = [
            'school_name' => 'required',
            'school_year' => 'required',
            'school_address' => 'required',
            'zonasi_quota' =>
'required|integer',
            'prestasi_quota' =>
'required|integer',

```

```

        'start_date' => 'required|date',
        'end_date' => 'required|date',
    ];
    $validator =
Validator::make($request->all(),
$rules);

    $input =
array_fill_keys(array_keys($rule
s), null);
    $errors = array_merge($input,
$validator->errors()->toArray());
    if ($validator->fails()) {
        return response()->json([
            'status' => false,
            'errors' => $errors,
        ]);
    }
    $setting =
Setting::updateOrCreate([
        'id' => 1
    ], [
        'school_name' => $request-
>school_name,
        'school_year' => $request-
>school_year,
        'school_address' => $request-
>school_address,
        'zonasi_quota' => $request-
>zonasi_quota,
        'prestasi_quota' => $request-
>prestasi_quota,
        'start_date' => $request-
>start_date,
        'end_date' => $request-
>end_date,
    ]);
    activity('update')-
>performedOn($setting)-
>withProperties(['user_agent' =>
$request->header('User-
Agent')])->log('update setting');
    return response()->json([
        'status' => true,
        'message' => 'Data Updated',
        'data' => $setting,
    ]);

```

```

    ]);
}
Public function
updateInstruction(Request
$request)
{
    $setting =
Setting::updateOrCreate([
        'id' => 1
    ], [
        'instruction' => $request-
>instruction,
    ]);
    activity('update')-
>performedOn($setting)-
>withProperties(['user_agent' =>
$request->header('User-
Agent')])->log('update setting');
    return response()->json([
        'status' => true,
        'message' => 'Data Updated',
        'data' => $setting,
    ]);
}
}
<?php
namespace
App\Http\Controllers\Backend;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use
Illuminate\Support\Facades\Vali
dator;
use Storage;
use App\Models\User;

class UserController extends
Controller
{
    public function
credentials($request, $password,
$image)
    {
        return [
            'username' => $request-

```

```

>username,
  'name' => $request->name,
  'password' => $password,
  'image' => $image,
];
}
public function json() {
  $data = User::role('operator')-
>get();
  return datatables()->of($data)
    ->addColumn(
      'action',
      function ($row) {
        $btn = '
          <button class="btn
            btn-warning btn-edit" data-
            id=".'.$row->id.'" title="Edit"
            data-toggle="modal" data-
            target="#editOperator"><i
              class="fa fa-edit"></i></button>
          <button class="btn
            btn-danger btn-delete" data-
            id=".'.$row->id.'" data-
            name=".'.$row->name.'"
            title="Delete"><i class="fa fa-
            trash"></i></button>
        ';
        return $btn;
      }
    )
    ->editColumn('image',
  function ($row) {
    $image =
    asset('assets/img/avatar/avatar-
    1.png');
    if ($row->image) {
      $image =
      asset('storage/user/images/'.$row
      ->image);
    }
    return '
    ';
  })
  ->editColumn('created_at',
  function ($row) {
    return $row->created_at-
    >format('d F Y - H:i:s');
  })
  ->rawColumns(['image',
  'action'])
  ->addIndexColumn()
  ->make();
}
public function index()
{
  $title = "Operator";
  $sub_title = '
    <div class="breadcrumb-
    item">List Operator</div>
  ';
  return
  view('backend.user.index',
  compact('title', 'sub_title'));
}
public function show($id)
{
  $user = User::findOrFail($id);

  return response()->json([
    'status' => true,
    'message' => 'Data Found',
    'data' => $user,
  ]);
}
public function store(Request
  $request)
{
  $rules = [
    'name' => 'required',
    'username' =>
    'required|unique:users,username',
    'password' =>
    'required|confirmed|min:6',
    'password_confirmation' =>
    'required|same:password|min:6',
    'image' => 'image|max:2048',
  ];
  $validator =
  Validator::make($request->all(),
  $rules);
  $input =

```

```

array_fill_keys(array_keys($rules), null);
$errors = array_merge($input,
$validator->errors()->toArray());
if ($validator->fails()) {
    return response()->json([
        'status' => false,
        'errors' => $errors,
    ]);
}

$password = bcrypt($request->password);
$image = null;
if ($request->file('image')) {
    $file = $request->file('image');
    $image = time().".$file->extension();
    $file->storeAs('public/user/images/',
$image);
}
$user = User::create($this->credentials($request,
$password, $image));

if ($user) {
    $user->assignRole('operator');
    return response()->json([
        'status' => true,
        'message' => 'Data Saved',
        'data' => $user,
    ]);
} else {
    Storage::disk('public')->delete('user/images/' . $user->image);
    return response()->json([
        'status' => false,
        'errors' => ['Oops, something bad happened!'],
    ], 500);
}
}

public function update(Request $request)
{
    $id = $request->id;
    $rules = [
        'name' => 'required',
        'username' => 'required|unique:users,username,' . $id,
        'password' => 'nullable|confirmed|min:6',
        'password_confirmation' => 'nullable|required_with:password|same:password|min:6',
        'image' => 'nullable|image|max:2048',
    ];
    $validator = Validator::make($request->all(), $rules);

    $input = array_fill_keys(array_keys($rules), null);
    $errors = array_merge($input, $validator->errors()->toArray());

    if ($validator->fails()) {
        return response()->json([
            'status' => false,
            'errors' => $errors,
        ]);
    }
    $user = User::findOrFail($id);
    $image = $user->image;
    $password = $user->password;

    if ($request->password) {
        $password = bcrypt($request->password);
    }
    if ($request->file('image')) {
        $file = $request->file('image');
        $image = time().".$file->extension();
    }
}

```

```

    $file-
>storeAs('public/user/images/',
$image);
    Storage::disk('public')-
>delete('user/images/' . $user-
>image);
}
$update = $user->update($this-
>credentials($request,
$password, $image));

if ($update) {
    return response()->json([
        'status' => true,
        'message' => 'Data
Updated',
        'data' => $user,
    ]);
} else {
    return response()->json([
        'status' => false,
        'errors' => ['Oops,
something bad happened!'],
    ], 500);
}
}
public function delete($id)
{
    $user = User::findOrFail($id);

    if ($user->delete()) {
        Storage::disk('public')-
>delete('user/images/' . $user-
>image);
        return response()->json([
            'status' => true,
            'message' => 'Data Deleted',
        ]);
    } else {
        return response()->json([
            'status' => false,
            'message' => 'Oops,
something bad happened!',
        ]);
    }
}
}

```

```

public function indexProfile()
{
    $title = "Profile";
    $sub_title = '
        <div class="breadcrumb-
item">Profile</div>
    ';
    $user = auth()->user();
    return
view('backend.profile.index',
compact('title', 'sub_title',
'user'));
}
public function updateProfile($id,
Request $request)
{
    $validator =
Validator::make($request->all(),
[
        'name' => 'required',
        'username' =>
'required|unique:users,username,'
.$id,
        'password' => 'confirmed',
        'image' => 'image|max:2048',
    ]);
    if ($validator->fails()) {
        alert()->error('Error',
$validator->messages()-
>all()[0]);
        return back()->withInput();
    }
    $user = User::findOrFail($id);
    $image = $user->image;
    $password = $user->password;

    if ($request->password) {
        $password = bcrypt($request-
>password);
    }
    if ($request->file('image')) {
        $file = $request-
>file('image');
        $image = time()." ".$file-
>extension();
        $file-

```

```

        >storeAs('public/user/images/',
        $image);
        Storage::disk('public')-
        >delete('user/images/' . $user-
        >image);
    }
    $user->update($this-
    >credentials($request,
    $password, $image));
    activity('update')-
    >withProperties(['user_agent' =>
    $request->header('User-
    Agent')])->log('update profile');
    alert()->success('Success', 'Data
    Updated');
    return back();
    }
}
<?php
namespace App\Http\Controllers;
use
    Illuminate\Foundation\Auth\Acc
    ess\AuthorizesRequests;
use
    Illuminate\Foundation\Bus\Disp
    atchesJobs;
use
    Illuminate\Foundation\Validatio
    n\ValidatesRequests;
use Illuminate\Routing\Controller as
    BaseController;

class Controller extends
    BaseController
{
    use AuthorizesRequests,
        DispatchesJobs,
        ValidatesRequests;
}
<?php
namespace App\Http\Middleware;
use
    Illuminate\Auth\Middleware\Aut
    henticate as Middleware;

class Authenticate extends

```

```

    Middleware
    {
        /**
         * Get the path the user should be
         * redirected to when they are not
         * authenticated.
         *
         * @param \Illuminate\Http\Request
         * $request
         * @return string|null
         */
        protected function
            redirectTo($request)
        {
            if (! $request->expectsJson()) {
                return route('be.auth.login');
            }
        }
    }
}
<?php
namespace App\Http\Middleware;
use
    Illuminate\Cookie\Middleware\E
    ncryptCookies as Middleware;

class EncryptCookies extends
    Middleware
{
    * The names of the cookies that
    * should not be encrypted.
    *
    * @var array<int, string>
    */
    protected $except = [
        //
    ];
}
<?php
namespace App\Http\Middleware;
use
    Illuminate\Foundation\Http\Mid
    dleware\PreventRequestsDuring
    Maintenance as Middleware;

class PreventRequestsDuringMaintena
    nce extends Middleware
{

```

```

    * The URIs that should be
    * reachable while maintenance
    * mode is enabled.
    *
    * @var array<int, string>
    */
    protected $except = [
    ];
}
namespace App\Http\Middleware;
use
    App\Providers\RouteServicePro
    vider;
use Closure;
use Illuminate\Http\Request;
use
    Illuminate\Support\Facades\Aut
    h;
use
    Illuminate\Support\Facades\Coo
    kie;
use
    Illuminate\Support\Facades\Sess
    ion;

class RedirectIfAuthenticated
{
    * Handle an incoming request.
    *
    * @param
    * \Illuminate\Http\Request
    * $request
    * @param
    * \Closure(\Illuminate\Http\Reque
    * st):
    * (\Illuminate\Http\Response|\Illu
    * minate\Http\RedirectResponse)
    * $next
    * @param string|null ...$guards
    * @return
    * \Illuminate\Http\Response|\Illu
    * minate\Http\RedirectResponse
    */
    public function handle(Request
    $request, Closure $next,
    ...$guards)
    {
        $guards = empty($guards) ?
        [null] : $guards;

        foreach ($guards as $guard) {
            if (Auth::guard($guard)-
            >check()) {
                // Session::flush();
                // $remember_cookie =

?php
namespace App\Console;
use
    Illuminate\Console\Scheduling\S
    chedule;
use
    Illuminate\Foundation\Console\
    Kernel as ConsoleKernel;

class Kernel extends ConsoleKernel
{
    * Define the application's
    * command schedule.
    *
    * @param
    * \Illuminate\Console\Scheduling\
    * Schedule $schedule
    * @return void
    */
    protected function
    schedule(Schedule $schedule)
    {
        // $schedule-
        >command('inspire')->hourly();
    }
    * Register the commands for the
    * application.
    *
    * @return void
    */
    protected function commands()
    {
        $this-
        >load(__DIR__.'/Commands');
        require
        base_path('routes/console.php');
    }
}

```

```

}
<?php
namespace App\Exceptions;
use
    Illuminate\Foundation\Exception
        s\Handler as ExceptionHandler;
use Throwable;

class Handler extends
    ExceptionHandler
{
    * A list of the exception types that
    are not reported.
    *
    * @var array<int, class-
    string<Throwable>>
    */
    protected $dontReport = [
        //
    ];
    * A list of the inputs that are never
    flashed for validation exceptions.
    *
    * @var array<int, string>
    */
    protected $dontFlash = [
        'current_password',
        'password',
        'password_confirmation',
    ];
    * Register the exception handling
    callbacks for the application.
    *
    * @return void
    */
    public function register()
    {
        $this->reportable(function
        (Throwable $e) {
            //
        });
    }
}
<?php
namespace
    App\Http\Requests\Registration;

```

```

use
    Illuminate\Foundation\Http\For
    mRequest;
use
    Illuminate\Contracts\Validation\
    Validator;
use
    Illuminate\Http\Exceptions\Http
    ResponseException;

class DocumentRequest extends
    FormRequest
{
    /**
     * Determine if the user is
     authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }
    * Get the validation rules that
    apply to the request.
    *
    * @return array<string, mixed>
    */
    public function rules()
    {
        return [
            'birth_certificate' =>
            'required|mimes:jpeg,bmp,png,gif,svg,pdf|max:2048',
            'family_certificate' =>
            'required|mimes:jpeg,bmp,png,gif,svg,pdf|max:2048',
            'kindergarten_certificate' =>
            'nullable|mimes:jpeg,bmp,png,gif,svg,pdf|max:2048',
            'kia' =>
            'required|mimes:jpeg,bmp,png,gif,svg,pdf|max:2048',
            'kip' =>
            'nullable|mimes:jpeg,bmp,png,gif,svg,pdf|max:2048',

```

```

        'kps' =>
        'nullable|mimes:jpeg,bmp,png,gif,svg,pdf|max:2048',
        'bpjs' =>
        'nullable|mimes:jpeg,bmp,png,gif,svg,pdf|max:2048',
        'father_id' =>
        'required|mimes:jpeg,bmp,png,gif,svg,pdf|max:2048',
        'mother_id' =>
        'required|mimes:jpeg,bmp,png,gif,svg,pdf|max:2048',
        'student_image' =>
        'required|mimes:jpeg,bmp,png,gif,svg,pdf|max:2048',
        'certificate_name' =>
        'nullable|array',
        'certificate_file' => 'array',
        'certificate_file.*' =>
        'nullable|mimes:jpeg,bmp,png,gif,svg,pdf|max:2048',
    ];
}
* Get custom attributes for
validator errors.
*
* @return array<string, string>
*/
public function attributes(): array
{
    return [
        'birth_certificate' => 'Akta
        Kelahiran',
        'family_certificate' =>
        'Kartu Keluarga',
        'kindergarten_certificate' =>
        'Ijazah TK/PAUD',
        'kia' => 'Kartu
        Identitas Anak',
        'kip' => 'Kartu
        Indonesia Pintar',
        'kps' => 'Kartu
        Perlindungan Sosial',
        'bpjs' => 'BPJS',
        'father_id' => 'KTP
        Ayah',
        'mother_id' => 'KTP
        Ibu',
        'student_image' => 'Pas
        Foto Siswa',
        'certificate_name.*' =>
        'Nama Sertifikat',
        'certificate_file.*' => 'File
        Sertifikat',
    ];
}
/**
 * Prepare the data for validation.
 */
protected function
prepareForValidation(): void
{
}
/**
 * Get the error messages for the
defined validation rules.*
 * @return array
 */
protected function
failedValidation(Validator
$validator)
{
    $input =
    array_fill_keys(array_keys($this
->rules()), null);
    $errors = array_merge($input,
$validator->errors()->toArray());

    throw new
    HttpResponseException(respons
e()->json([
        'status' => false,
        'step' => 4,
        'message' => 'Oops,
        Something bad happened.',
        'errors' => $errors,
    ]));
}
}
<?php
namespace
App\Http\Requests\Registration;

```

```

use
    Illuminate\Foundation\Http\FormRequest;
use
    Illuminate\Contracts\Validation\Validator;
use
    Illuminate\Http\Exceptions\HttpResponseException;

class GuardianRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }
    /**
     * Get the validation rules that apply to the request.
     *
     * @return array<string, mixed>
     */
    public function rules()
    {
        return [
            'name_3' => 'nullable',
            'pob_3' => 'nullable',
            'dob_3' => 'nullable',
            'religion_3' => 'nullable',
            'last_education_3' =>
                'nullable',
            'job_3' => 'nullable',
            'monthly_income_3' =>
                'nullable',
            'citizenship_3' => 'nullable',
            'phone_3' =>
                'nullable|min:8',
            'address_3' => 'nullable',
            'relationship_3' => 'nullable',
        ];
    }
    /**
     * Get custom attributes for validator errors.
     *
     * @return array<string, string>
     */
    public function attributes(): array
    {
        return [
            'name_3' => 'Nama Ayah',
            'pob_3' => 'Tempat Lahir',
            'dob_3' => 'Tanggal Lahir',
            'religion_3' => 'Agama',
            'last_education_3' => 'Ijazah Tertinggi',
            'job_3' => 'Pekerjaan',
            'monthly_income_3' =>
                'Penghasilan per Bulan',
            'citizenship_3' =>
                'Kewarganegaraan',
            'phone_3' => 'No. Telepon (HP)',
            'address_3' => 'Alamat',
            'relationship_3' =>
                'Hubungan Keluarga',
        ];
    }
    /**
     * Prepare the data for validation.
     */
    protected function prepareForValidation(): void
    {
        $this->merge([
            'monthly_income_3' =>
                str_replace('.', '', $this->monthly_income_3),
        ]);
    }
    /**

```

```

* Get the error messages for the
  defined validation rules.*
* @return array
*/
protected function
  failedValidation(Validator
    $validator)
{
    $input =
    array_fill_keys(array_keys($this
      ->rules()), null);
    $errors = array_merge($input,
      $validator->errors()->toArray());

    throw new
      HttpResponseException(respons
        e()->json([
          'status' => false,
          'step' => 3,
          'message' => 'Oops,
            Something bad happened.',
          'errors' => $errors,
        ]));
}
}
<?php
namespace
  App\Http\Requests\Registration;
use
  Illuminate\Foundation\Http\For
    mRequest;
use
  Illuminate\Contracts\Validation\
    Validator;
use
  Illuminate\Http\Exceptions\Http
    ResponseException;

class ParentRequest extends
  FormRequest
{
    /**
     * Determine if the user is
     * authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }
    /**
     * Get the validation rules that
     * apply to the request.
     *
     * @return array<string, mixed>
     */
    public function rules()
    {
        return [
            'start_date' => 'required|date|after:' .
              $this->getSetting('start_date'),
            'end_date' => 'required|date|after:' .
              $this->getSetting('end_date'),
            'mt-4' => 'required|integer|between:1,10',
            'route(fe.instruction)' => 'required|url',
            'btn btn-outline-white btn-
              lg btn-icon icon-left' => 'required|alpha-
                numeric',
            'fas fa-circle-info' => 'required|alpha-
              numeric',
            'Lihat Petunjuk Pendaftaran' => 'required|alpha-
              numeric',
            'button class="btn btn-
              info btn-lg btn-icon icon-left ml-
                2 btn-search-student"' => 'required|alpha-
                  numeric',
            'fas fa-circle-
              question' => 'required|alpha-
                numeric',
            'Sudah daftar?
              Lihat Status' => 'required|alpha-
                numeric'
        ];
    }
}
</div>
</div>
</div>
<div class="col-12 col-md-6 col-
  lg-6">
  <div class="card card-dark">
    <div class="card-header">
      <h4>Jalur Zonasi</h4>
      <div class="card-header-
        action">
        <a href="{ {
          route('fe.registration.zonasi') } }"
          class="btn btn-primary { {
            ((date('Y-m-d') >= getSetting()-
              >start_date->format('Y-m-d'))
            && (date('Y-m-d') <=

```

```

    getSetting()->end_date-
    >format('Y-m-d')) ? : 'disabled'
  }}">
      Daftar
    </a>
  </div>
</div>
<div class="card-body">
  <p>Pendaftaran jalur <b
class="text-
primary">zonasi</b>
diperuntukkan bagi calon siswa
yang memiliki domisili desa zona
terdekat dengan sekolah.</p>
</div>
</div>
</div>
<div class="col-12 col-md-6 col-
lg-6">
  <div class="card card-dark">
    <div class="card-header">
      <h4>Jalur Prestasi</h4>
      <div class="card-header-
action">
        <a href="{{
route('fe.registration.prestasi')
}}" class="btn btn-primary {{
((date('Y-m-d') >= getSetting()-
>start_date->format('Y-m-d'))
&& (date('Y-m-d') <=
getSetting()->end_date-
>format('Y-m-d')) ? : 'disabled'
}}">
          Daftar
        </a>
      </div>
    </div>
    <div class="card-body">
      <p>Pendaftaran jalur <b
class="text-
primary">prestasi</b>
diperuntukkan bagi para calon
siswa yang memiliki prestasi dan
berada di luar zonasi sekolah
sepanjang memenuhi

```

```

    persyaratan.</p>
  </div>
</div>
</div>
@push('scripts')
<script type="text/javascript">
  @break

  @default
  <span
class="badge badge-
warning">Ditinjau</span>
  @endswitch
</h4>
</div>
</div>
<div class="card-body">
  <div class="row">
    <div class="col-sm-6
col-md-12 col-lg-12 table-
responsive">
      <h6 class="section-
title" style="margin-top:
0;">Keterangan Pribadi</h6>
      <table class="table
table-md table-striped">
        <tr>
          <td width="20%"
rowspan="6">
            
          </td>
          <td
width="20%">Nama
Lengkap</td>
          <td
width="25%">: {{ $student-
>fullname }}</td>
          <td
width="20%">Kewarganegaraan
</td>
          <td>: {{
$student->citizenship }}</td>
        </tr>

```

```

        <tr>
            <td>Nama
Panggilan</td>
            <td>:      {{
$student->nickname }}</td>
            <td>Anak
Keberapa</td>
            <td>:      {{
$student->birth_order }}</td>
        </tr>
        <tr>
            <td>Jenis
Kelamin</td>
            <td>:      {{
$student->gender == 1 ? 'Laki-
laki' : 'Perempuan' }}</td>
            <td>Jumlah
Saudara Kandung</td>
            <td>:      {{
$student->total_sibling }}</td>
        </tr>
        <tr>
            <td>Tempat,
Tanggal Lahir</td>
            <td>:      {{
$student->pob }}, {{ $student-
>dob->format('d-m-Y') }}</td>
            <td>Jumlah
Saudara Tiri</td>
            <td>:      {{
$student->total_step_sibling
}}</td>
        </tr>
        <tr>
            <td>Agama</td>
            <td>:      {{
$student->religion }}</td>
            <td>Jumlah
Saudara Angkat</td>
            <td>:      {{
$student->total_foster_sibling
}}</td>
        </tr>
        <tr>
        </tr>
        </tr>
    </table>

```

```

    </div>
    <div class="col-sm-12
table-responsive">
        <table class="table
table-md table-striped">
            <tr>
                <td
width="20%">Golongan
Darah</td>
                <td
width="30%">:  {{ $student-
>blood_type }}</td>
                <td
width="20%">Jarak Tempat
Tinggal ke Sekolah</td>
                <td
width="25%">:  {{ $student-
>distance_in_meters }}
meter</td>
            </tr>
            <tr>
                <td>Alamat</td>
                <td>:      {{
$student->address }}</td>
                <td>Bahasa
Sehari-hari</td>
                <td>:      {{
$student->language }}</td>
            </tr>
            <tr>
                <td>No. Telepon
(HP)</td>
                <td>:      +62{{
$student->phone }}</td>
                <td>Berat Badan
/ Tinggi Badan</td>
                <td>:      {{
!$student->weight_in_kg ? '-' :
$student->weight_in_kg }} kg /
{{ !$student->height_in_cm ? '-' :
$student->height_in_cm }}
cm</td>
            </tr>
            <tr>
                <td>Bertempat
Tinggal</td>

```

```

<td>:
@switch($student->residence)
    @case(1)

<span>Orang Tua</span>
    @break

    @case(2)

<span>Menumpang pada orang
lain</span>
    @break

    @case(3)

<span>Asrama</span>
    @break

    @default

<span></span>
    @endswitch
</td>
<td>Riwayat
Penyakit</td>
    <td>:      {{
str_replace(',', ' ', $student-
>disease_history) }}</td>
    </tr>
</table>
<h6 class="section-
title">Pendidikan
Sebelumnya</h6>
<table class="table
table-md table-striped">
    <tr>
        <td
width="15%">Nama TK</td>
        <td>:      {{
$student->kindergarten_name
}}</td>
        <td
width="20%">Tanggal      dan
Nomor Ijazah</td>
        <td>:      {{
$student->certificate_number
}}</td>
    </tr>
</table>
    <td>Alamat
TK</td>
    <td>:      {{
$student->kindergarten_address
}}</td>
    <td>Lama
Belajar</td>
    <td>:      {{
$student->long_study }}</td>
</tr>
</table>
<h6 class="section-
title">Data Orang Tua</h6>
<table class="table
table-md table-striped">
    @php
        $ayah = $student-
>parents->where('type', 1)-
>first();
        $ibu = $student-
>parents->where('type', 2)-
>first();
        $wali = $student-
>parents->where('type', 3)-
>first();
    @endphp
    <tr>
        <td
colspan="2"><h6 class="text-
primary">Data Ayah</h6></td>
        <td
colspan="2"><h6 class="text-
primary">Data Ibu</h6></td>
    </tr>
    <tr>
        <td>Nama</td>
        <td>: {{ $ayah-
>name ?? '-' }}</td>
        <td>Nama</td>
        <td>: {{ $ibu-
>name ?? '-' }}</td>
    </tr>
</table>

```

```

        <td>Pendidikan
Tertinggi</td>
        <td>: {{ $ayah-
>last_education ?? '-' }}</td>
        <td>Pendidikan
Tertinggi</td>
        <td>: {{ $ibu-
>last_education ?? '-' }}</td>
    </tr>
    <tr>
        <td>Pekerjaan</td>
        <td>: {{ $ayah-
>job ?? '-' }}</td>

<td>Pekerjaan</td>
        <td>: {{ $ibu-
>job ?? '-' }}</td>
    </tr>
    <tr>
        <td>Masih
Hidup</td>
        <td>: {{ !$ayah ?
'Tidak' : 'Ya' }}</td>
        <td>Masih
Hidup</td>
        <td>: {{ !$ibu ?
'Tidak' : 'Ya' }}</td>
    </tr>
</table>
@if($wali)
    <h6 class="section-
title">Data Wali</h6>
    <table class="table
table-md table-striped">
        <tr>
            <td>Nama</td>
            <td>: {{ $wali-
>name }}</td>
            <td>Pendidikan
Tertinggi</td>
            <td>: {{ $wali-
>last_education }}</td>
        </tr>
        <tr>
            <td>Pekerjaan</td>
            <td>: {{ $wali-

```

```

>job }}</td>
        <td>Hubungan
Keluarga</td>
        <td>: {{ $wali-
>relationship }}</td>
    </tr>
</table>
@endif
@if($student-
>document)
    <h6 class="section-
title">Dokumen</h6>
    <table class="table
table-md table-striped">
        <tr>
            <td>Akta
Kelahiran</td>
            <td>
                @if($student-
>document->birth_certificate)
                    <button
type="button" class="btn btn-
info btn-sm" data-
toggle="modal" data-
target="#modal-pdf" data-
title="Akta Kelahiran" data-
file="{{
asset('storage/documents/'.$stude
nt->id.'/'.$student->document-
>birth_certificate) }}" data-
type="{{ explode('.', $student-
>document-
>birth_certificate)[1] }}">Lihat
Dokumen</button>
                @else
                    <button
class="btn btn-secondary btn-
sm" disabled>Tidak Ada
Dokumen</button>
                @endif
            </td>
            <td>Kartu
Keluarga</td>
            <td>
                @if($student-
>document->family_certificate)

```

```

        <button
type="button" class="btn btn-
info btn-sm" data-
toggle="modal" data-
target="#modal-pdf" data-
title="Kartu Keluarga" data-
file="{ {
asset('storage/documents/'.$stude
nt->id.'/'.$student->document-
>family_certificate) } }" data-
type="{ { explode('.', $student-
>document-
>family_certificate)[1]
} }">Lihat Dokumen</button>
        @else
        <button
class="btn btn-secondary btn-
sm" disabled>Tidak Ada
Dokumen</button>
        @endif
    </td>
    <td>Ijazah
TK/PAUD</td>
    <td>
        @if($student-
>document-
>kindergarten_certificate)
        <button
type="button" class="btn btn-
info btn-sm" data-
toggle="modal" data-
target="#modal-pdf" data-
title="Ijazah TK/PAUD" data-
file="{ {
asset('storage/documents/'.$stude
nt->id.'/'.$student->document-
>kindergarten_certificate) } }"
data-type="{ { explode('.',
$student->document-
>kindergarten_certificate)[1]
} }">Lihat Dokumen</button>
        @else
        <button
class="btn btn-secondary btn-
sm" disabled>Tidak Ada
Dokumen</button>

```

```

        @endif
    </td>
</tr>
<tr>
    <td>KTP
Ayah</td>
    <td>
        @if($student-
>document->father_id)
        <button
type="button" class="btn btn-
info btn-sm" data-
toggle="modal" data-
target="#modal-pdf" data-
title="KTP Ayah" data-file="{ {
asset('storage/documents/'.$stude
nt->id.'/'.$student->document-
>father_id) } }" data-type="{ {
explode('.', $student->document-
>father_id)[1]
} }">Lihat
Dokumen</button>
        @else
        <button
class="btn btn-secondary btn-
sm" disabled>Tidak Ada
Dokumen</button>
        @endif
    </td>
    <td>KTP
Ibu</td>
    <td>
        @if($student-
>document->mother_id)
        <button
type="button" class="btn btn-
info btn-sm" data-
toggle="modal" data-
target="#modal-pdf" data-
title="KTP Ibu" data-file="{ {
asset('storage/documents/'.$stude
nt->id.'/'.$student->document-
>mother_id) } }" data-type="{ {
explode('.', $student->document-
>mother_id)[1]
} }">Lihat
Dokumen</button>
        @else

```

```

        <button
class="btn btn-secondary btn-
sm" disabled>Tidak Ada
Dokumen</button>
    @endif
</td>
</tr>
</table>
@endif
<h6 class="section-
title">Sertifikat Prestasi</h6>
<table class="table
table-md table-striped">
<tr>
<td>Nama
Sertifikat</td>
<td>Dokumen
Sertifikat</td>
</tr>
@forelse($student-
>otherDocuments as $doc)
<tr>
<td>{{ $doc-
>name ?? '-' }}</td>
<td>
<button
type="button" class="btn btn-
info btn-sm" data-
toggle="modal" data-
target="#modal-pdf" data-
title="{{ $doc->name }}" data-
file="{{
asset('storage/documents/'.$stude
nt->id.'/'.$doc->file) }}" data-
type="{{ explode('.', $doc-
>file)[1] }}">Lihat
Dokumen</button>
</td>
</tr>
@empty
<tr>
<td>-</td>
<td>-</td>
</tr>
@endforelse
</table>

```

```

</div>
</div>
</div>
</div>
</div>
@push('scripts')
<div class="modal fade" id="modal-
pdf" style="display: none;" aria-
hidden="true">
<div class="modal-dialog modal-
lg">
<div class="modal-content">
<div class="modal-header">
<h4 class="modal-
title">Modal</h4>
<button type="button"
class="close" data-
dismiss="modal" aria-
label="Close">
<span aria-
hidden="true">×</span>
</button>
</div>
<div class="modal-body"
style="padding: 0;"></div>
</div>
</div>
<script type="text/javascript">
$(document).ready(function() {
    $('#modal-
pdf').on("show.bs.modal", (e) =>
    {
        var file =
$(e.relatedTarget).data('file')
        if
$(e.relatedTarget).data('type')
== 'pdf') {
            $('<div class="modal-
body">').html('<iframe class="file"
src="'+file+'" width="100%"
style="min-height: calc(100vh -
140px);"></iframe>')
        } else {

```

```

        $('modal-
body').html('</img>')
    }
    $('modal-
title').text($(e.relatedTarget).data
('title'))
    })
})
</script>
@endpush
@endsection
<script                src="{ {
    asset('assets/library/jquery/dist/j
query.min.js') } }"></script>
<script                src="{ {
    asset('assets/library/popper.js/dis
t/umd/popper.js') } }"></script>
<script                src="{ {
    asset('assets/library/tooltip.js/dist
/umd/tooltip.js') } }"></script>
<script                src="{ {
    asset('assets/library/jquery-ui-
dist/jquery-ui.min.js')
    } }"></script>
<script                src="{ {
    asset('assets/library/bootstrap/dis
t/js/bootstrap.min.js')
    } }"></script>
<script                src="{ {
    asset('assets/library/jquery.nices
croll/dist/jquery.nicescroll.min.js
') } }"></script>
<script                src="{ {
    asset('assets/library/moment/min
/moment.min.js') } }"></script>
<script                src="{ {
    asset('assets/js/stisla.js')
    } }"></script>

<!-- JS Libraies -->
<script                src="{ {
    asset('assets/library/jquery-
sparkline/jquery.sparkline.min.js
') } }"></script>

```

```

<script                src="{ {
    asset('assets/library/chart.js/dist/
Chart.min.js') } }"></script>
<script                src="{ {
    asset('assets/library/owl.carousel
/dist/owl.carousel.min.js')
    } }"></script>
<script                src="{ {
    asset('assets/library/summernote/
dist/summernote-bs4.js')
    } }"></script>
<script                src="{ {
    asset('assets/library/chocolat/dist
/js/jquery.chocolat.min.js')
    } }"></script>
<script                src="{ {
    asset('assets/library/izitoast/dist/j
s/iziToast.min.js') } }"></script>
<script                src="{ {
    asset('assets/library/sweetalert/di
st/sweetalert.min.js')
    } }"></script>
<!-- bs-custom-file-input -->
<script                src="{ {
    asset('assets/library/bs-custom-
file-input/bs-custom-file-
input.min.js') } }"></script>
<!-- Template JS File -->
<script
    src="https://cdn.datatables.net/1.
12.1/js/jquery.dataTables.min.js"
></script>
<script
    src="https://cdn.datatables.net/1.
12.1/js/dataTables.bootstrap4.mi
n.js"></script>
<script                src="{ {
    asset('assets/library/select2/dist/j
s/select2.full.min.js')
    } }"></script>
<script                src="{ {
    asset('assets/library/bootstrap-
daterangepicker/daterangepicker
.js') } }"></script>
<script                src="{ {
    asset('assets/library/bootstrap-

```

```

timepicker/js/bootstrap-
timepicker.min.js') }}"></script>
<script src="{ {
  asset('assets/library/selectric/pub
lic/jquery.selectric.min.js')
} }"></script>
<script src="{ {
  asset('assets/library/paper-
bootstrap-
wizard/js/jquery.validate.min.js')
} }"
type="text/javascript"></script>
<script src="{ {
  asset('assets/library/cleave.js/dist
/cleave.min.js') } }"></script>
<script src="{ {
  asset('assets/library/jquery.pwstr
ength/jquery.pwstrength.min.js')
} }"></script>

@stack('scripts')

<!-- Template JS File -->
<script src="{ {
  asset('assets/js/scripts.js')
} }"></script>
<script src="{ {
  asset('assets/js/custom.js')
} }"></script>

<!-- Page Specific JS File -->
{ {-- <script src="{ {
  asset('assets/assets/js/page/index.
js') } }"></script> --} }
<script src="{ {
  asset('assets/js/page/modules-
toastr.js') } }"></script>
<script>
$(document).ready(function() {
  bsCustomFileInput.init();
  $('#dataTable').dataTable()
  $('#show_hide_password
a').on('click', function (event) {
    event.preventDefault();
    if ($('#show_hide_password
input').attr('type') == "text") {
      $('#show_hide_password
input').attr('type', 'password');
      $('#show_hide_password
i').addClass("fa-eye-slash");
      $('#show_hide_password
i').removeClass("fa-eye");
    } else if
      ($('#show_hide_password
input').attr('type') ==
"password") {
      $('#show_hide_password
input').attr('type', 'text');
      $('#show_hide_password
i').removeClass("fa-eye-slash");
      $('#show_hide_password
i').addClass("fa-eye");
    }
  })
<!-- CSS Libraries -->
<link rel="stylesheet" href="{ {
  asset('assets/library/bootstrap-
social/bootstrap-social.css') } }">
<link rel="stylesheet" href="{ {
  asset('assets/library/jqvmap/dist/
jqvmap.min.css') } }">
<link rel="stylesheet" href="{ {
  asset('assets/library/summernote/
dist/summernote-bs4.css') } }">
<link rel="stylesheet" href="{ {
  asset('assets/library/owl.carousel
/dist/assets/owl.carousel.min.css'
) } }">
<link rel="stylesheet" href="{ {
  asset('assets/library/owl.carousel
/dist/assets/owl.theme.default.mi
n.css') } }">
<link rel="stylesheet" href="{ {
  asset('assets/library/iziToast/dist/
css/iziToast.min.css') } }">
<link rel="stylesheet" href="{ {
  asset('assets/library/datatables/m
edia/css/jquery.dataTables.min.c
ss') } }">
<link rel="stylesheet"
href="https://cdn.datatables.net/

```

```

1.12.1/css/dataTables.bootstrap4
.min.css">
{{-- <link rel="stylesheet" href="{{
asset('assets/library/datatables.ne
t-select-
bs4/css/select.bootstrap4.min.css
')}}"> --}}
<link rel="stylesheet" href="{{
asset('assets/library/select2/dist/c
ss/select2.min.css')}}">
<link rel="stylesheet" href="{{
asset('assets/library/bootstrap-
daterangepicker/daterangepicker
.css')}}">
<link rel="stylesheet" href="{{
asset('assets/library/bootstrap-
timepicker/css/bootstrap-
timepicker.min.css')}}">
<link rel="stylesheet" href="{{
asset('assets/library/chocolat/dist
/css/chocolat.css')}}">
@stack('styles')

<!-- Template CSS -->
<link rel="stylesheet" href="{{
asset('assets/css/style.css')}}">
<link rel="stylesheet" href="{{
asset('assets/css/components.css'
)}}">
<style type="text/css">
.card .card-header .btn:not(.note-
btn) {
border-radius: .25rem
!important;
}

.text-italic {
font-style: italic;
}
.text-bold {
font-weight: bold;
}
</style>
<!doctype html>
<html lang="en">
<head>

```

```

<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport"
content="width=device-width,
initial-scale=1">

<!-- Bootstrap CSS -->
<link
href="https://cdn.jsdelivrivr.net/np
m/bootstrap@5.1.3/dist/css/boot
strap.min.css" rel="stylesheet"
integrity="sha384-
1BmE4kWBq78iYhFldvKuhfT
AU6auU8tT94WrHftjDbrCEXS
U1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">

<title>Cetak Rekap Data</title>
<style type="text/css">
table {
border-collapse: collapse;
}
table tr {
border-bottom: 1px solid;
vertical-align: bottom;
}
table td {
padding: 5px;
}
.page-break {
page-break-before: always;
}
</style>
</head>
<body>
<div class="row">
<div class="col-sm-12"
align="center" style="border-
bottom: 5px double;">
<h2>Data Pendaftaran
Peserta Didik Baru</h2>
<h4>Tahun Ajaran : {{
getSetting()->school_year
}}</h4>
</div>
<div class="col-sm-12">

```

```

<table class="table">
  <thead>
    <th>No.
Pendaftaran</th>
    <th>Jalur</th>
    <th>Nama
Lengkap</th>
    <th>Jenis
Kelamin</th>
    <th>Tempat, Tanggal
Lahir</th>
    <th>Status</th>
    <th>Tanggal
Pendaftaran</th>
  </thead>
  <tbody>
    @foreach($data as
$row)
      <tr>
        <td>{{ $row-
>registration_number }}</td>
        <td>{{ $row->type
== 1 ? 'Zonasi' : 'Prestasi' }}</td>
        <td>{{ $row-
>fullname }}</td>
        <td>{{ $row-
>gender == 1 ? 'Laki-laki' :
'Perempuan' }}</td>
        <td>{{ $row-
>pob.',
'.Carbon\Carbon::parse($row-
>dob)->locale('id')-
>isoFormat('DD MMMM Y')
}}</td>
        <td>{{ $row-
>status ? ($row->status == 1 ?
'Lolos' : 'Tidak Lolos') : 'Ditinjau'
}}</td>
        <td>{{
Carbon\Carbon::parse($row-
>created_at)->locale('id')-
>isoFormat('DD MMMM Y')
}}</td>
      </tr>
    @endforeach
  </tbody>

```

```

</table>
</div>
</div>
<script
src="https://cdn.jsdelivr.net/npm
/bootstrap@5.1.3/dist/js/bootstra
p.bundle.min.js"
integrity="sha384-
ka7Sk0Gln4gmtz2MlQnikT1wX
gYsOg+OMhuP+IRH9sENBOO
LRn5q+8nbTov4+1p"
crossorigin="anonymous"></scr
ipt>
</body>
</html>
<!doctype html>
<html lang="en">
<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport"
content="width=device-width,
initial-scale=1">
  <!-- Bootstrap CSS -->
  <link
href="https://cdn.jsdelivr.net/np
m/bootstrap@5.1.3/dist/css/boot
strap.min.css" rel="stylesheet"
integrity="sha384-
1BmE4kWBq78iYhFldvKuhfT
AU6auU8tT94WrHftjDbrCEXS
U1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">

  <title>{{ $student-
>registration_number }}</title>
  <style type="text/css">
    table {
      border-collapse: collapse;
    }
    table tr {
      border-bottom: 1px solid;
      vertical-align: bottom;
    }
    table td {
      padding: 5px;

```

```

    }
    .page-break {
        page-break-before: always;
    }
</style>
</head>
<body>
    <div class="row">
        <div class="col-sm-12"
            align="center" style="border-
            bottom: 5px double;">
            <h2>Formulir Pendaftaran
            Peserta Didik Baru</h2>
            <h4>Tahun Ajaran : {{
            getSetting()->school_year
            }}</h4>
        </div>
        <div class="col-sm-6 col-md-
        12 col-lg-12" style="margin-top:
        10px;">
            <h4 style="text-decoration:
            underline;">Nomor Pendaftaran:
            {{
                $student-
                >registration_number }}</h4>
            <h6 class="section-
            title">Keterangan Pribadi</h6>
            <table class="table">
                <tr>
                    <td
                    width="40%">Nama
                    Lengkap</td>
                    <td>: {{ $student-
                    >fullname }}</td>
                </tr>
                <tr>
                    <td>Nama
                    Panggilan</td>
                    <td>: {{ $student-
                    >nickname }}</td>
                </tr>
                <tr>
                    <td>Jenis
                    Kelamin</td>
                    <td>: {{ $student-
                    >gender == 1 ? 'Laki-laki' :
                    'Perempuan' }}</td>
                </tr>
            </table>
        </div>
    </div>

```

```

        </tr>
        <tr>
            <td>Tempat, Tanggal
            Lahir</td>
            <td>: {{ $student-
            >pob }}, {{ $student->dob-
            >format('d-m-Y') }}</td>
        </tr>
        <tr>
            <td>Agama</td>
            <td>: {{ $student-
            >religion }}</td>
        </tr>
        <tr>
            <td>Kewarganegaraan</td>
            <td>: {{ $student-
            >citizenship }}</td>
        </tr>
        <tr>
            <td>Anak
            Keberapa</td>
            <td>: {{ $student-
            >birth_order }}</td>
        </tr>
        <tr>
            <td>Jumlah Saudara
            Kandung</td>
            <td>: {{ $student-
            >total_sibling }}</td>
        </tr>
        <tr>
            <td>Jumlah Saudara
            Tiri</td>
            <td>: {{ $student-
            >total_step_sibling }}</td>
        </tr>
        <tr>
            <td>Jumlah Saudara
            Angkat</td>
            <td>: {{ $student-
            >total_foster_sibling }}</td>
        </tr>
        <tr>
            <td>Bahasa Sehari-
            hari</td>

```

```

        <td>: {{ $student-
>language }}</td>
    </tr>
    <tr>
        <td>Alamat</td>
        <td>: {{ $student-
>address }}</td>
    </tr>
    <tr>
        <td>No.        Telepon
(HP)</td>
        <td>: +62{{ $student-
>phone }}</td>
    </tr>
    <tr>:
        <td>Bertempat
Tinggal</td>
        <td>:
            @switch($student-
>residence)
                @case(1)
                    <span>Orang
Tua</span>
                @break
                @case(2)
                    <span>Menumpang
pada orang lain</span>
                @break
                @case(3)
                    <span>Asrama</span>
                @break
            @default
        <span></span>
        @endswitch
    </td>
</tr>
<tr>
        <td>Jarak        Tempat
Tinggal ke Sekolah</td>
        <td>: {{ $student-
>distance_in_meters        }}
meter</td>
    </tr>

```

```

    <tr>
        <td>Masih
Hidup</td>
        <td>: {{ !$ayah ?
'Tidak': 'Ya' }}</td>
    </tr>
    <tr>
        <td colspan="2"><h6
class="text-primary">Data
Ibu</h6></td>
    </tr>
    <tr>
        <td>Nama</td>
        <td>: {{ $ibu->name
?? '-' }}</td>
    </tr>
    <tr>
        <td>Pendidikan
Tertinggi</td>
        <td>: {{ $ibu-
>last_education ?? '-' }}</td>
    </tr>
    <tr>
        <td>Pekerjaan</td>
        <td>: {{ $ibu->job ??
 '-' }}</td>
    </tr>
    <tr>
        <td>Masih
Hidup</td>
        <td>: {{ !$ibu ?
'Tidak': 'Ya' }}</td>
    </tr>
</table>
@if($wali)
    <div class="page-
break"></div>
    <h6 class="section-
title">Data Wali</h6>
    <table class="table">
        <tr>
            <td
width="40%">Nama</td>
            <td>: {{ $wali->name
}}</td>
        </tr>

```

```

        <tr>
            <td>Hubungan
Keluarga</td>
            <td>: {{ $wali-
>relationship }}</td>
        </tr>
        <tr>
            <td>Pendidikan
Tertinggi</td>
            <td>: {{ $wali-
>last_education }}</td>
        </tr>
        <tr>
            <td>Pekerjaan</td>
            <td>: {{ $wali->job
}}</td>
        </tr>

```

```

    </table>
    @endif
</div>
</div>
<script
src="https://cdn.jsdelivr.net/npm
/bootstrap@5.1.3/dist/js/bootstra
p.bundle.min.js"
integrity="sha384-
ka7Sk0Gln4gmtz2MlQnikT1wX
gYsOg+OMhuP+IlRH9sENBO0
LRn5q+8nbTov4+1p"
crossorigin="anonymous"></scr
ipt>
</body>
</html>

```



KARTU MONITORING BIMBINGAN
MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH PAREPARE

SKRIPSI

Mahasiswa : Sartika	Pembimbing I : Ade Hastuty, ST., S.Kom., MT
NIM : 218280197	Pembimbing II : Mugaffir Yunus, ST., MT.
Judul Skripsi : Aplikasi PPDB Berbasis Web ^{PADA SD} Berbasis Web Negeri 49 PAREPARE	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 uji white box uji black box buat angket untuk penggunaan aplikasi. Dokumen minimal 2mb		Konsultasi 1 Perbaiki fitur daftarnya	
Konsultasi 2 09/04/2024 Acc Skrip 5.		Konsultasi 2 Gambar dipertelas Pengujian jalur zonasi dilampirkan	
Konsultasi 3 Daftar pustaka kajian teori		Konsultasi 3 Sambungkan jalur zonasi dengan map Api	
Konsultasi 4 kerangka pikir Flowchart Abstrak		Konsultasi 4 Acc	
Konsultasi 5		Konsultasi 5	

Lanjut ke halaman sebelah...

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton berwarna hijau muda dan dicetak timbal balik