**TABLE OF CONTENTS**

**Page**

**Title Page.......................................................................................................................... i**

**Certification and Approval Sheet .................................................................................. ii**

**Acknowledgements......................................................................................................... iii**

**Certification of Originality............................................................................................... iv**

**Abstract............................................................................................................................ v**

**Table of Contents............................................................................................................ vi**

**List of Tables................................................................................................................... vii**

**List of Figures ................................................................................................................. viii**

**Chapter 1. INTRODUCTION.............................................................................................. 1**

Project Context....................................................................................................... 1

Purpose and Description........................................................................................ 2

Objectives............................................................................................................... 4

Scope and Limitation.............................................................................................. 5

**Chapter 2. REVIEW OF THE RELATED LITERATURE AND STUDIES…………………. 7**

Foreign Literature.................................................................................................... 7

Local Literature....................................................................................................... 10

Foreign Studies....................................................................................................... 11

Local Studies........................................................................................................... 12

**Chapter 3. TECHNICAL BACKGROUND.......................................................................... 14**

Research Approach................................................................................................. 14

Methodology............................................................................................................ 15

Requirement Analysis............................................................................................. 16

Systems Requirements........................................................................................... 18

**Chapter 4. RESULTS AND DISCUSSION......................................................................... 23**

Requirements Documentation................................................................................. 23

Design of the Software............................................................................................ 28

Development and Testing........................................................................................ 29

Implementation Plan................................................................................................ 32

Implementation Results........................................................................................... 34

**Chapter 5. CONCLUSIONS AND RECOMMENDATIONS............................................... 36**

Conclusion.............................................................................................................. 36

Recommendations.................................................................................................. 37

**REFERENCES**

**APPENDICES**

1. ENGLISH CRITIC’S CERTIFICATION
2. STATISTICIAN’S CERTIFICATION
3. PROJECT PROPOSAL
4. COMMUNICATION/LETTERS
5. INTERVIEW GUIDE QUESTIONS
6. SURVEY QUESTIONNAIRES
7. FORMS AND DOCUMENTS
8. SYSTEMS EVALUATION
9. SOFTWARE DEVELOPMENT COST
10. COST BENEFIT ANALYSIS
11. DATA FLOW DIAGRAM
12. SYSTEM FLOWCHART
13. DATA DICTIONARY
14. ENTITY RELATIONSHIP DIAGRAM
15. VISUAL TABLE OF CONTENTS
16. INPUT – PROCESS – OUTPUT
17. PROGRAM FLOWCHART
18. USER’S GUIDE
19. PROGRAM OUTPUT SNAPSHOTS
20. PROGRAM CODE LISTING
21. PHOTO DOCUMENTATION
22. CURRICULUM VITAE

APPENDICES

**APPENDIX A.**

**ENGLISH CRITIC’S CERTIFICATION**

**APPENDIX T.**

**PROGRAM CODE LISTING**

//Elections Controller

//

<?php

namespace App\Http\Controllers\Root;

/\*\*

\* Application

\*/

use App\Services\Notify;

use App\Election;

/\*\*

\* Laravel

\*/

use Illuminate\Http\Request;

use App\Http\Controllers\Controller;

class ElectionsController extends Controller

{

/\*\*

\* Show index page.

\* @return \Illuminate\View\View

\*/

public function index()

{

$elections = Election::all();

return view('root.elections.index', compact('elections'));

}

/\*\*

\* Show resource creation page.

\* @return \Illuminate\View\View

\*/

public function create()

{

return view('root.elections.create');

}

/\*\*

\* Store resource.

\* @param \Illuminate\Http\Request

\* @return \Illuminate\Routing\Redirector

\*/

public function store(Request $request)

{

$request->validate([

'name' => 'required|unique:elections,name,NULL,id,deleted\_at,NULL',

'start\_date' => 'required|date|after\_or\_equal:today|before\_or\_equal:end\_date',

'end\_date' => 'required|date|after\_or\_equal:start\_date'

]);

$election = new Election;

$election->fill($request->all());

unset($election->files);

if ($election->save()) {

Notify::success('Election created.', 'Success!');

return redirect()->route('root.elections.dashboard', $election);

}

Notify::warning('Cannot create the election.', 'Warning!');

return redirect()->route('root.elections.index');

}

/\*\*

\* Show resource edit page.

\* @param \Illuminate\Http\Request

\* @param \App\Election

\* @return \Illuminate\View\View

\*/

public function edit(Request $request, Election $election)

{

return view('root.elections.edit', compact('election'));

}

/\*\*

\* Update resource.

\* @param \Illuminate\Http\Request

\* @param \App\Election

\*/

public function update(Request $request, Election $election)

{

$request->validate([

'name' => "required|unique:elections,name,{$election->id},id,deleted\_at,NULL",

'start\_date' => 'required|date|after\_or\_equal:today|before\_or\_equal:end\_date',

'end\_date' => 'required|date|after\_or\_equal:start\_date'

]);

$election->fill($request->all());

unset($election->files);

if ($election->update()) {

Notify::success('Election event updated.', 'Success!');

return redirect()->route('root.elections.index');

}

Notify::warning('Cannot update election event.', 'Failed');

return redirect()->route('root.elections.index');

}

}

//Election Tally Controller

<?php

namespace App\Http\Controllers\Root;

/\*\*

\* Application

\*/

use App\Exports\{ElectionTallyExport};

use App\Repositories\ElectionRepository;

use App\Services\{Notify};

use App\{User, Election, Position};

/\*\*

\* Package Services

\*/

use PDF;

use Maatwebsite\Excel\Facades\Excel;

/\*\*

\* Laravel

\*/

use Illuminate\Support\Facades\DB;

use Illuminate\Http\Request;

use App\Http\Controllers\Controller;

class ElectionTallyController extends Controller

{

/\*\*

\* Show Tally page.

\* @param \Illuminate\Http\Request

\* @param \App\Election

\* @return \Illuminate\View\View

\*/

public function showTallyPage(Request $request, Election $election)

{

$archives = (new ElectionRepository($election))->getTally(

$request->get('position')

);

return view('root.elections.election.tally', compact(

['election', 'archives']

));

}

}

//Election Ties Controller

<?php

namespace App\Http\Controllers\Root;

/\*\*

\* Application

\*/

use App\Repositories\ElectionRepository;

use App\{Election, Candidate};

/\*\*

\* Laravel

\*/

use Illuminate\Support\Facades\DB;

use Illuminate\Http\Request;

use App\Http\Controllers\Controller;

class ElectionTiesController extends Controller

{

/\*\*

\* Fetch election tie-breakers.

\* @param \Illuminate\Http\Request

\* @param \App\Election

\* @return string

\*/

public function fetch(Request $request, Election $election)

{

$tally = (new ElectionRepository($election))->getTally();

return (new ElectionRepository($election))->getTieBreakers($tally);

}

/\*\*

\* @param \Illuminate\Http\Request

\* @param \App\Election

\* @return string

\*/

public function randomize(Request $request, Election $election)

{

$tally = (new ElectionRepository($election))->getTally();

$tieBreakers = (new ElectionRepository($election))->getTieBreakers($tally);

$candidates = $tieBreakers[$request->post('index')];

// random index.

$randomIndex = mt\_rand(0, count($candidates) - 1);

// candidate (winner).

$candidate = $candidates[$randomIndex]->candidate;

// insert the winner.

DB::table('election\_winners')->insert([

'election\_id' => $election->id,

'candidate\_id' => $candidate->id

]);

// we will just return the candidate

return $candidate;

}

}

//Election Winners

<?php

namespace App\Http\Controllers\Root;

/\*\*

\* Application

\*/

use App\Services\Notify;

use App\Repositories\ElectionRepository;

use App\Election;

/\*\*

\* Laravel

\*/

use Illuminate\Support\Facades\DB;

use Illuminate\Http\Request;

use App\Http\Controllers\Controller;

class ElectionWinnersController extends Controller

{

/\*\*

\* Fetch election winners.

\* @param \Illuminate\Http\Request

\* @param \App\Election

\* @return string

\*/

public function fetch(Request $request, Election $election)

{

return [];

}

/\*\*

\* @param \Illuminate\Http\Request

\* @param \App\Election

\* @return \Illuminate\Routing\Redirector

\*/

public function declare(Request $request, Election $election)

{

/\*\*

\* This data is a bit of biased (for tie breakers).

\* @var Illuminate\Support\Collection

\*/

$leaders = (new ElectionRepository($election))->getWinners();

/\*\*

\* These are early declared winners (randomized).

\* @var array

\*/

$tbWinners = DB::table('election\_winners as w')

->where('w.election\_id', $election->id)

->leftJoin('candidates as c', 'c.user\_id', '=', 'w.candidate\_id')

->where('c.election\_id', $election->id)

->select(['c.user\_id as candidate\_id', 'c.position\_id'])

->get()

->toArray();

/\*\*

\* @var array

\*/

$winners = collect([]);

collect($leaders)->each(function($leader) use ($winners, $tbWinners) {

$positions = array\_column($tbWinners, 'position\_id');

if (($index = array\_search($leader->position\_id, $positions)) === false) {

$winners->push($leader->candidate\_id);

} else {

$winners->push($tbWinners[$index]->candidate\_id);

}

});

// delete tie break winners.

DB::table('election\_winners')

->where('election\_id', $election->id)

->delete();

// store winners all at once in db.

foreach ($winners as $winner) {

DB::table('election\_winners')->insert([

'election\_id' => $election->id,

'candidate\_id' => $winner

]);

}

$election->status = 'closed';

if ($election->save()) {

Notify::success('Election winners declared.');

}

return back();

}

}

//Create Election – View

@extends('root.templates.master')

@section('content')

@component('root.components.breadcrumbs')

@slot('page\_title')

Create an election

@endslot

<li class="breadcrumb-item">

<a href="{{ route('root.elections.index') }}">Elections</a>

</li>

<li class="breadcrumb-item active">

Create

</li>

@endcomponent

<div class="row">

<div class="col-12">

<div class="card">

<div class="card-body">

<h4 class="card-title">Please fill up the form</h4>

<h6 class="card-subtitle">

Fields with

<span class="text-danger">\*</span> are required

</h6>

<form

method="POST"

action="{{ route('root.elections.store') }}"

class="form-material m-t-40"

submit-once

>

@csrf

<!-- Name -->

<div class="form-group">

<label for="Name">

Name <span class="text-danger">\*</span>

</label>

<input

type="text"

name="name"

id="name"

class="form-control form-control-line"

value="{{ old('name') }}"

placeholder="E.g. Election {{ date('Y') }}"

>

@if ($errors->has('name'))

<span class="text-danger">

{{ $errors->first('name') }}

</span>

@endif

</div>

<!--/. Name -->

<!-- Date of Start -->

<div class="form-group">

<label for="start\_date">

Start Date <span class="text-danger">\*</span>

</label>

<input

type="text"

name="start\_date"

id="start\_date"

class="form-control form-control-line"

value="{{ old('start\_date') }}"

placeholder="Start of Election"

>

<span class="help-block text-muted">

<small>

Must be the date after or equal to <code>today</code>.

</small>

</span>

<br />

@if ($errors->has('start\_date'))

<span class="text-danger">

{{ $errors->first('start\_date') }}

</span>

@endif

</div>

<!--/. Date of Start -->

<!-- Date of End -->

<div class="form-group">

<label for="end\_date">

End Date <span class="text-danger">\*</span>

</label>

<input

type="text"

name="end\_date"

id="end\_date"

class="form-control form-control-line"

value="{{ old('end\_date') }}"

placeholder="End of Election"

>

<span class="help-block text-muted">

<small>

Must be the date equal or after the <code>start date</code>.

</small>

</span>

<br />

@if ($errors->has('end\_date'))

<span class="text-danger">

{{ $errors->first('end\_date') }}

</span>

@endif

</div>

<!--/. Date of End -->

<!-- Description -->

<div class="form-group">

<label for="description">Description</label>

<textarea

name="description"

id="description"

class="form-control form-control-line summernote"

>

{{ old('description') }}

</textarea>

@if ($errors->has('description'))

<span class="text-danger">

{{ $errors->first('description') }}

</span>

@endif

</div>

<!--/. Description -->

<!-- Submit -->

<div class="form-group">

<button type="submit" class="btn btn-info btn-loading">

<i class="fas fa-plus"></i> Create

</button>

<a href="{{ route('root.elections.index') }}" class="btn btn-secondary">

Cancel

</a>

</div>

<!--/. Submit -->

</form>

</div>

</div>

</div>

</div>

@endsection

@section('styles')

<link href="/assets/plugins/bootstrap-material-datetimepicker/css/bootstrap-material-datetimepicker.css" rel="stylesheet">

<link href="/assets/plugins/summernote/dist/summernote-bs4.css" rel="stylesheet" />

@endsection

@section('scripts')

<script src="/assets/plugins/moment/moment.js"></script>

<script src="/assets/plugins/bootstrap-material-datetimepicker/js/bootstrap-material-datetimepicker.js"></script>

<script src="/assets/plugins/summernote/dist/summernote-bs4.min.js"></script>

<script>

$('#start\_date').bootstrapMaterialDatePicker({time: false});

$('#end\_date').bootstrapMaterialDatePicker({time: false});

$('.summernote').summernote({

height: 350,

minHeight: null,

maxHeight: null,

focus: false

});

</script>

@endsection

//Edit Election – View

@extends('root.templates.master')

@section('content')

@component('root.components.breadcrumbs')

@slot('page\_title')

Edit an election

@endslot

<li class="breadcrumb-item active">

Edit

</li>

@endcomponent

<div class="row">

<div class="col-12">

<div class="card">

<div class="card-body">

<h4 class="card-title">Please fill up the form</h4>

<h6 class="card-subtitle">

Fields with

<span class="text-danger">\*</span> are required

</h6>

<form

method="POST"

action="{{ route('root.elections.update', $election) }}"

class="form-material m-t-40"

submit-once

>

@csrf

@method('PATCH')

<!-- Name -->

<div class="form-group">

<label for="Name">

Name <span class="text-danger">\*</span>

</label>

<input

type="text"

name="name"

id="name"

class="form-control form-control-line"

value="{{ old('name') ?? $election->name }}"

placeholder="E.g. Election {{ date('Y') }}"

>

@if ($errors->has('name'))

<span class="text-danger">

{{ $errors->first('name') }}

</span>

@endif

</div>

<!--/. Name -->

<!-- Date of Start -->

<div class="form-group">

<label for="Name">

Start Date <span class="text-danger">\*</span>

</label>

<input

type="text"

name="start\_date"

id="start\_date"

class="form-control form-control-line"

value="{{ old('start\_date') ?? $election->start\_date }}"

placeholder="Start of Election"

>

<span class="help-block text-muted">

<small>

Must be the date after or equal to <code>today</code>.

</small>

</span>

<br />

@if ($errors->has('start\_date'))

<span class="text-danger">

{{ $errors->first('start\_date') }}

</span>

@endif

</div>

<!--/. Date of Start -->

<!-- Date of End -->

<div class="form-group">

<label for="Name">

End Date <span class="text-danger">\*</span>

</label>

<input

type="text"

name="end\_date"

id="end\_date"

class="form-control form-control-line"

value="{{ old('end\_date') ?? $election->end\_date }}"

placeholder="End of Election"

>

<span class="help-block text-muted">

<small>

Must be the date equal or after the <code>start date</code>.

</small>

</span>

<br />

@if ($errors->has('end\_date'))

<span class="text-danger">

{{ $errors->first('end\_date') }}

</span>

@endif

</div>

<!--/. Date of End -->

<!-- Description -->

<div class="form-group">

<label for="description">Description</label>

<textarea

name="description"

id="description"

class="form-control form-control-line summernote"

>

{{ old('description') ?? $election->description }}

</textarea>

@if ($errors->has('description'))

<span class="text-danger">

{{ $errors->first('description') }}

</span>

@endif

</div>

<!--/. Description -->

<!-- Submit -->

<div class="form-group">

<button type="submit" class="btn btn-info btn-loading">

<i class="fas fa-edit"></i> Update

</button>

<a href="{{ route('root.elections.index') }}" class="btn btn-secondary">

Cancel

</a>

</div>

<!--/. Submit -->

</form>

</div>

</div>

</div>

</div>

@endsection

@section('styles')

<link href="/assets/plugins/bootstrap-material-datetimepicker/css/bootstrap-material-datetimepicker.css" rel="stylesheet">

<link href="/assets/plugins/summernote/dist/summernote-bs4.css" rel="stylesheet" />

@endsection

@section('scripts')

<script src="/assets/plugins/moment/moment.js"></script>

<script src="/assets/plugins/bootstrap-material-datetimepicker/js/bootstrap-material-datetimepicker.js"></script>

<script src="/assets/plugins/summernote/dist/summernote-bs4.min.js"></script>

<script>

$('#start\_date').bootstrapMaterialDatePicker({time: false});

$('#end\_date').bootstrapMaterialDatePicker({time: false});

$('.summernote').summernote({

height: 350,

minHeight: null,

maxHeight: null,

focus: false

});

</script>

@endsection

//Election Module Index Page – View

@extends('root.templates.master')

@section('content')

@component('root.components.breadcrumbs')

@slot('page\_title')

List of Elections

@endslot

<li class="breadcrumb-item active">

Elections

</li>

@slot('action')

<form method="GET" action="{{ route('root.elections.create') }}">

<button type="submit" class="btn btn-info float-right">

<i class="fas fa-plus"></i> Create

</button>

</form>

@endslot

@endcomponent

<div class="row">

<div class="col-12">

<div class="card">

<div class="card-body">

<h4 class="card-title">Elections</h4>

<h6 class="card-subtitle">

This is the list of elections.

</h6>

<div class="table-responsive m-t-40">

<div>

<table id="table-elections" class="display nowrap table table-hover table-striped table-bordered" cellspacing="0" width="100%">

<thead>

<tr>

<th>Actions</th>

<th width="20%">Name</th>

<th width="15%">Start Date</th>

<th width="15%">End Date</th>

<th>Status</th>

<th>Positions</th>

<th>Candidates</th>

</tr>

</thead>

<tbody>

@foreach($elections as $election)

<tr>

<td class="text-nowrap">

<a

href="{{ route('root.elections.edit', $election) }}"

class="link-edit-election"

data-toggle="tooltip"

data-original-title="Edit"

>

<i class="fas fa-pencil-alt m-r-10"></i>

</a>

<a

href="{{ route('root.elections.dashboard', $election) }}"

class="link-manage-election"

data-toggle="tooltip"

data-original-title="Manage"

>

<i class="fas fa-box-open m-r-10"></i>

</a>

</td>

<td>{{ $election->name }}</td>

<td>{{ $election->start\_date }}</td>

<td>{{ $election->end\_date }}</td>

<td>

<span class="label label-rounded label-{{ $election->status\_class }}">

{{ ucfirst($election->status) }}

</span>

</td>

<td>{{ optional($election->positions)->count() ?? 0 }}</td>

<td>{{ optional($election->candidates)->count() ?? 0 }}</td>

</tr>

@endforeach

</tbody>

</table>

</div>

</div>

</div>

</div>

</div>

</div>

<form method="POST" action="" id="form-destroy-election" style="display: none;">

@csrf

@method('DELETE')

</form>

@endsection

@section('modals')

@component('root.components.modals.confirmation')

<p class="text-center">

You are deleting a resource.

<br />

Warning: you cannot undo this action!

</p>

@endcomponent

@endsection

@section('scripts')

<script src="/assets/plugins/datatables/datatables.min.js"></script>

<script>

$('#table-elections').DataTable({

"bLengthChange" : false,

});

$('.link-destroy-admin').on('click', function(event) {

var action = $(this).data('action');

var form = $('#form-destroy-election');

var modal = $("#modal-confirmation");

form.attr({action: action});

modal.modal().on('click', '#btn-modal-confirm', function() {

form.submit();

});

});

</script>

@endsection

//Election Model

<?php

namespace App;

class Election extends Model

{

public function positions()

{

return $this->belongsToMany(Position::class);

}

public function candidates()

{

return $this->hasMany(Candidate::class);

}

/\*\*

\* @return string

\*/

public function getStatusClassAttribute()

{

switch ($this->status) {

case 'upcoming':

return 'warning';

break;

case 'active':

return 'success';

break;

case 'ended':

return 'danger';

break;

case 'closed':

return 'info';

break;

}

}

}

//Migration – Create Elections Table

<?php

use Illuminate\Support\Facades\Schema;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Database\Migrations\Migration;

class CreateElectionsTable extends Migration

{

/\*\*

\* Run the migrations.

\*

\* @return void

\*/

public function up()

{

Schema::create('elections', function (Blueprint $table) {

$table->increments('id');

$table->string('name');

$table->text('description')->nullable();

$table->string('start\_date');

$table->string('end\_date');

$table->enum('status', ['upcoming', 'active', 'ended', 'closed', ''])->default('upcoming');

$table->boolean('active')->default(1);

$table->integer('created\_by')->nullable();

$table->integer('updated\_by')->nullable();

$table->integer('deleted\_by')->nullable();

$table->timestamps();

$table->softDeletes();

});

}

/\*\*

\* Reverse the migrations.

\*

\* @return void

\*/

public function down()

{

Schema::dropIfExists('elections');

}

}