

Answer To the Question of assignment-24

1. Event Management Project using Laravel

Project Description:

You are tasked with creating an event management web application using the Laravel framework. Users should be able to create, view, edit, and delete events. The application should have authentication in place to ensure that only logged-in users can perform these actions. Events should have a title, description, date, time, and location.

Requirements:

Setup and Authentication:

Set up a new Laravel project.

Implement user authentication.

Create routes for user registration, login, and logout.

Database:

Set up a database with tables for users and events.

Define the necessary columns for the events table (e.g., title, description, date, time, location).

Create relationships between users and events (one-to-many).

Event CRUD:

Create a dashboard page for logged-in users.

Implement a page to list all events with basic details.

Implement a page to view the details of a specific event.

Create a page to add a new event.

Implement editing and updating of existing events. Add a feature to delete events.

Validation and Error Handling: Implement server-side validation for event creation and editing. Display appropriate error messages to users in case of validation failures.

Authentication Middleware

Use middleware to protect routes that require authentication.

Allow only authenticated users to create, edit, and delete events.

Frontend:

Design and implement responsive frontend views using Blade templates and CSS.

Display events in an organized and user-friendly format.

Enhance user experience with proper form layouts and interactions.

Allow users to RSVP for events.

Implement event categories and filtering.

Add pagination to the events listing page.

Implement a search functionality for events. for student. every controller ,table migration,blade,resource,model every step to fullfil the project also every view .

Creating an event management web application using Laravel involves several steps. Here's a detailed breakdown of each step you need to follow:

1. Setup and Authentication:

Install Laravel using Composer:

```
composer create-project laravel/laravel emwa
```

Implement user authentication:

```
php artisan make:auth
```

Create routes for user registration, login, and logout:

Define routes in the routes/web.php file:

2. Database: Name emwa

Set up a database with tables for users and events:

Configure the database connection in the .env file.

```
composer require laravel/breeze --dev
```

```
php artisan breeze:install
```

php artisan migrate

npm install

npm run dev

Create migration files for users and events tables:

php artisan make:migration create_users_table

php artisan make:migration create_events_table

Migration files are for creating tables:

By this command php artisan make:migration create_table_name_table

Below :---

2014_10_12_000000_create_users_table.php

2014_10_12_100000_create_password_reset_tokens_table.php

2019_08_19_000000_create_failed_jobs_table.php

2019_12_14_000001_create_personal_access_tokens_table.php

2023_07_19_075701_create_event_tag_table.php

2023_08_15_093105_create_countries_table.php

2023_08_15_093155_create_cities_table.php

2023_08_15_093250_create_comments_table.php

2023_08_15_093347_create_events_table.php

2023_08_15_093432_create_galleries_table.php

2023_08_15_093525_create_attendings_table.php

2023_08_15_093725_create_likes_table.php

2023_08_15_093833_create_saved_events_table.php

2023_08_15_093903_create_tags_table.php

Define necessary columns for the events table (title, description, date, time, location):

Edit the migration file to include the required columns.

Create relationships between users and events (one-to-many):

Define the relationships in the User and Event models:

php

Copy code

```
// User.php  
public function events() {  
    return $this->hasMany(Event::class);  
}
```

```
// Event.php  
public function user() {  
    return $this->belongsTo(User::class);  
}
```

Run migrations to create the database tables:

Copy code

php artisan migrate

3. Event CRUD:

Create a dashboard page for logged-in users:

Create a dashboard view.

Implement a page to list all events with basic details:

Create a route, controller method, and view to display a list of events.

Implement a page to view the details of a specific event:

Create a route, controller method, and view to show event details.

Create a page to add a new event:

Create a route, controller method, and view for adding events.

Implement editing and updating of existing events:

Create a route, controller method, and view for editing events.

Add a feature to delete events:

Implement a route, controller method, and functionality to delete events.

4. Validation and Error Handling:

Implement server-side validation for event creation and editing:

Use Laravel's validation rules in the controller methods.

Display appropriate error messages to users in case of validation failures:

Display errors in the Blade views using @error directive.

5. Authentication Middleware:

Use middleware to protect routes that require authentication:

Apply the auth middleware to routes that need authentication in the routes/web.php file.

Allow only authenticated users to create, edit, and delete events:

Check user authentication in the controller methods.

6. Frontend:

Design and implement responsive frontend views using Blade templates and CSS:

Create Blade templates in the resources/views directory and style them using CSS.

Display events in an organized and user-friendly format:

Design the events listing view to display event details.

Enhance user experience with proper form layouts and interactions:

Implement forms for event creation, editing, and RSVPs.

7. Additional Features:

Allow users to RSVP for events:

Add functionality to allow users to indicate their attendance.

Implement event categories and filtering:

Add categories to events and allow users to filter events by category.

Add pagination to the events listing page:

Implement pagination for a better user experience.

Implement a search functionality for events:

Create a search feature to allow users to search for events by title, location, or other criteria.

This breakdown covers the major steps required to create an event management web application using Laravel. Each step involves creating routes, controllers, models, views, and possibly other components to achieve the desired functionality. Remember to refer to the Laravel documentation for detailed information on syntax, methods, and best practices.

All route are In web.php

```
<?php
```

```
use App\Http\Controllers\AttendingEventController;  
use App\Http\Controllers\AttendingSystemController;  
use App\Http\Controllers>DeleteCommentController;  
use App\Http\Controllers\EventController;  
use App\Http\Controllers\EventIndexController;  
use App\Http\Controllers\EventShowController;  
use App\Http\Controllers\GalleryController;
```

```
use App\Http\Controllers\GalleryIndexController;
use App\Http\Controllers\LikedEventController;
use App\Http\Controllers\LikeSystemController;
use App\Http\Controllers\ProfileController;
use App\Http\Controllers\SavedEventController;
use App\Http\Controllers\SavedEventSystemController;
use App\Http\Controllers\StoreCommentController;
use App\Http\Controllers\TestEventController;
use App\Http\Controllers>WelcomeController;
use App\Models\Country;
use Illuminate\Support\Facades\Route;
```

```
Route::get('/', WelcomeController::class)->name('welcome');
Route::get('/e', EventIndexController::class)->name('eventIndex');
Route::get('/e/{id}', EventShowController::class)->name('eventShow');
Route::get('/gallery', GalleryIndexController::class)->name('galleryIndex');
```

```
Route::get('/dashboard', function () {
    return view('dashboard');
})->middleware(['auth', 'verified'])->name('dashboard');
```

```
Route::middleware('auth')->group(function () {
    Route::get('/profile', [ProfileController::class, 'edit'])->name('profile.edit');
    Route::patch('/profile', [ProfileController::class, 'update'])->name('profile.update');
    Route::delete('/profile', [ProfileController::class, 'destroy'])->name('profile.destroy');
```

```
Route::resource('/events', EventController::class);
Route::resource('/galleries', GalleryController::class);
```

```

Route::get('/liked-events', LikedEventController::class)->name('likedEvents');
Route::get('/saved-events', SavedEventController::class)->name('savedEvents');
Route::get('/attendind-events', AttendingEventController::class)->name('attendingEvents');
Route::post(
    '/events-like/{id}',
    LikeSystemController::class
)->name('events.like');
Route::post(
    '/events-saved/{id}',
    SavedEventSystemController::class
)->name('events.saved');
Route::post('/events-attending/{id}', AttentingSystemController::class)->name('events.attending');

Route::post('/events/{id}/comments', StoreCommentController::class)->name('events.comments');
Route::delete('/events/{id}/comments/{comment}', DeleteCommentController::class)-
>name('events.comments.destroy');
Route::get('/countries/{country}', function (Country $country) {
    return response()->json($country->cities);
});
});

require __DIR__ . '/auth.php';

```

Create Model :

Attending,

City,

Comment,

Country,
Event,
Gallery,
Like,
SavedEvent,
Tag,
User.

Attending.php.....

<?php

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
use Illuminate\Database\Eloquent\Relations\BelongsTo;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
```

```
class Attending extends Model
```

```
{
```

```
    use HasFactory;
```

```
    protected $fillable = [
```

```
        'user_id',
```

```
        'event_id',
```

```
        'num_tickets'
```

```
    ];
```

```
    public function user(): BelongsTo
```

```
{
```

```
        return $this->belongsTo(User::class);
    }
}
```

```
public function event(): BelongsTo
{
    return $this->belongsTo(Event::class);
}
}
```

City.php.....

<?php

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\HasMany;
use Illuminate\Database\Eloquent\Relations\BelongsTo;
use Illuminate\Database\Eloquent\Factories\HasFactory;
```

```
class City extends Model
```

```
{
    use HasFactory;
```

```
    protected $fillable = [
        'country_id',
        'name'
    ];
```

```
    public function country(): BelongsTo
    {
```

```
        return $this->belongsTo(Country::class);
    }
}
```

```
public function events(): HasMany
{
    return $this->hasMany(Event::class);
}
}
```

Comment.php.....

<?php

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\BelongsTo;
use Illuminate\Database\Eloquent\Factories\HasFactory;
```

```
class Comment extends Model
```

```
{
    use HasFactory;
```

```
    protected $fillable = [
        'user_id',
        'event_id',
        'content'
    ];
```

```
    public function user(): BelongsTo
    {
```

```
        return $this->belongsTo(User::class);
    }
}
```

```
public function event(): BelongsTo
{
    return $this->belongsTo(Event::class);
}
}
```

Country.php.....

<?php

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\HasMany;
use Illuminate\Database\Eloquent\Factories\HasFactory;
```

```
class Country extends Model
```

```
{
    use HasFactory;
```

```
    protected $fillable = ['name'];
```

```
    public function cities(): HasMany
    {
        return $this->hasMany(City::class);
    }
```

```
    public function events(): HasMany
```

```

    {
        return $this->hasMany(Event::class);
    }
}

```

Event.php.....

```
<?php
```

```
namespace App\Models;
```

```

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\BelongsTo;
use Illuminate\Database\Eloquent\Relations\BelongsToMany;
use Illuminate\Database\Eloquent\Relations\HasMany;

```

```
class Event extends Model
```

```

{
    use HasFactory;

```

```
    protected $fillable = [
```

```

        'title',
        'slug',
        'description',
        'start_date',
        'end_date',
        'start_time',
        'image',
        'address',
        'user_id',

```

```
        'country_id',  
        'city_id',  
        'num_tickets'  
    ];
```

```
    protected $casts = [  
        'start_date' => 'date:m/d/Y',  
        'end_date' => 'date:m/d/Y',  
    ];
```

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

```
    public function country(): BelongsTo  
    {  
        return $this->belongsTo(Country::class);  
    }
```

```
    public function city(): BelongsTo  
    {  
        return $this->belongsTo(City::class);  
    }
```

```
    public function comments(): HasMany  
    {  
        return $this->hasMany(Comment::class);  
    }
```

```
public function likes(): HasMany
{
    return $this->hasMany(Like::class);
}
```

```
public function savedEvents(): HasMany
{
    return $this->hasMany(SavedEvent::class);
}
```

```
public function attendings(): HasMany
{
    return $this->hasMany(Attending::class);
}
```

```
public function tags(): BelongsToMany
{
    return $this->belongsToMany(Tag::class);
}
```

```
public function hasTag($tag)
{
    return $this->tags->contains($tag);
}
}
```

Gallery.php.....

<?php

namespace App\Models;

```
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\BelongsTo;
use Illuminate\Database\Eloquent\Factories\HasFactory;
```

```
class Gallery extends Model
```

```
{
    use HasFactory;
```

```
    protected $fillable = [
```

```
        'user_id',
```

```
        'image',
```

```
        'caption'
```

```
    ];
```

```
    public function user(): BelongsTo
```

```
    {
```

```
        return $this->belongsTo(User::class);
```

```
    }
```

```
}
```

```
Like.php.....
```

```
<?php
```

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
use Illuminate\Database\Eloquent\Relations\BelongsTo;
```



```

class Like extends Model
{
    use HasFactory;

    protected $fillable = [
        'user_id',
        'event_id'
    ];

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

    public function event(): BelongsTo
    {
        return $this->belongsTo(Event::class);
    }
}

SavedEvent.php.....
<?php

```

```

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\BelongsTo;

class SavedEvent extends Model
{

```

```
use HasFactory;
```

```
protected $fillable = [
```

```
    'user_id',
```

```
    'event_id'
```

```
];
```

```
public function user(): BelongsTo
```

```
{
```

```
    return $this->belongsTo(User::class);
```

```
}
```

```
public function event(): BelongsTo
```

```
{
```

```
    return $this->belongsTo(Event::class);
```

```
}
```

```
}
```

```
Tag.php.....
```

```
<?php
```

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
use Illuminate\Database\Eloquent\Relations\BelongsToMany;
```

```
class Tag extends Model
```

```
{
```

```
    use HasFactory;
```

```

protected $fillable = [
    'name',
    'slug'
];

public function events(): BelongsToMany
{
    return $this->belongsTo(Event::class);
}
}

```

User.php.....

```
<?php
```

```
namespace App\Models;
```

```

// use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Relations\BelongsToMany;
use Illuminate\Database\Eloquent\Relations\HasMany;
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',
    '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',
    'password' => 'hashed',
];

```

```
public function events(): HasMany
{
    return $this->hasMany(Event::class);
}
```

```
public function galleries(): HasMany
{
    return $this->hasMany(Gallery::class);
}
```

```
public function comments(): HasMany
{
    return $this->hasMany(Comment::class);
}
```

```
public function likes(): HasMany
{
    return $this->hasMany(Like::class);
}
```

```
public function attendings(): HasMany
{
    return $this->hasMany(Attending::class);
}
}
```

Controllers are create as below:--

AttendingEventController:--

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Models\Event;
```

```
class AttendingEventController extends Controller
```

```
{
```

```
    /**
```

```
     * Handle the incoming request.
```

```
    */
```

```
    public function __invoke()
```

```
    {
```

```
        $events = Event::with('attendings')->whereHas('attendings', function ($q) {
```

```
            $q->where('user_id', auth()->id());
```

```
        }->get();
```

```
        return view('events.attendingEvents', compact('events'));
    }
```

```
}
```

```
AttendingSystemController:--
```

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Models\Event;
```

```
class AttendingEventController extends Controller
```

```
{
```

```
    /**
```

```
     * Handle the incoming request.
```

```

*/
public function __invoke()
{
    $events = Event::with('attendings')->whereHas('attendings', function ($q) {
        $q->where('user_id', auth()->id());
    })->get();

    return view('events.attendingEvents', compact('events'));
}
}

```

DeleteCommentController:--

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Models\Comment;
```

```
use Illuminate\Http\Request;
```

```
class DeleteCommentController extends Controller
```

```

{
    /**
     * Handle the incoming request.
     */
    public function __invoke($id, Comment $comment)
    {
        $this->authorize('delete', $comment);

        $comment->delete();

        return back();
    }
}

```

```

    }
}
EventController:--
<?php

namespace App\Http\Controllers;

use App\Models\Comment;
use Illuminate\Http\Request;

class DeleteCommentController extends Controller
{
    /**
     * Handle the incoming request.
     */
    public function __invoke($id, Comment $comment)
    {
        $this->authorize('delete', $comment);

        $comment->delete();

        return back();
    }
}

```

```

EventController:--
<?php

namespace App\Http\Controllers;

```



```
use App\Http\Requests\CreateEventRequest;
use App\Http\Requests\UpdateEventRequest;
use App\Models\Country;
use App\Models\Event;
use App\Models\Tag;
use Illuminate\Http\RedirectResponse;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Storage;
use Illuminate\Support\Str;
use Illuminate\View\View;
```

```
class EventController extends Controller
```

```
{
    /**
     * Display a listing of the resource.
     */
    public function index(): View
    {
        $events = Event::with('country')->get();
        return view('events.index', compact('events'));
    }
}
```

```
/**
 * Show the form for creating a new resource.
 */
public function create(): View
{
    $countries = Country::all();
    $tags = Tag::all();
}
```

```

        return view('events.create', compact('countries', 'tags'));
    }

    /**
     * Store a newly created resource in storage.
     */
    public function store(CreateEventRequest $request): RedirectResponse
    {
        if ($request->hasFile('image')) {

            $data = $request->validated();
            $data['image'] = Storage::putFile('events', $request->file('image'));
            $data['user_id'] = auth()->id();
            $data['slug'] = Str::slug($request->title);

            $event = Event::create($data);
            $event->tags()->attach($request->tags);
            return to_route('events.index');
        } else {
            return back();
        }
    }

    /**
     * Show the form for editing the specified resource.
     */
    public function edit(Event $event): View
    {

```

```

    $countries = Country::all();

    $tags = Tag::all();

    return view('events.edit', compact('countries', 'tags', 'event'));
}

/**
 * Update the specified resource in storage.
 */
public function update(UpdateEventRequest $request, Event $event): RedirectResponse
{
    $data = $request->validated();

    if ($request->hasFile('image')) {
        Storage::delete($event->image);

        $data['image'] = Storage::putFile('events', $request->file('image'));
    }

    $data['slug'] = Str::slug($request->title);

    $event->update($data);

    $event->tags()->sync($request->tags);

    return to_route('events.index');
}

/**
 * Remove the specified resource from storage.
 */
public function destroy(Event $event): RedirectResponse
{
    Storage::delete($event->image);

    $event->tags()->detach();
}

```

```
        $event->delete();  
        return to_route('events.index');  
    }  
}
```

EventIndexController:--

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Models\Event;
```

```
class EventIndexController extends Controller
```

```
{  
    /**  
     * Handle the incoming request.  
     */  
    public function __invoke()  
    {  
        $events = Event::with('country', 'tags')->orderBy('created_at', 'desc')->paginate(12);  
        return view('eventIndex', compact('events'));  
    }  
}
```

EventShowController:--

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Models\Event;
```

```
use Illuminate\Http\Request;
```

```
class EventShowController extends Controller
```

```
{
```

```
    /**
```

```
     * Handle the incoming request.
```

```
    */
```

```
    public function __invoke($id)
```

```
    {
```

```
        $event = Event::findOrFail($id);
```

```
        $like = $event->likes()->where('user_id', auth()->id())->first();
```

```
        $savedEvent = $event->savedEvents()->where('user_id', auth()->id())->first();
```

```
        $attending = $event->attendings()->where('user_id', auth()->id())->first();
```

```
        return view('eventsShow', compact('event', 'like', 'savedEvent', 'attending'));
    }
```

```
}
```

```
GalleryController:--
```

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Models\Gallery;
```

```
use Illuminate\Http\Request;
```

```
use Illuminate\Support\Facades\Storage;
```

```

class GalleryController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        $galleries = auth()->user()->galleries;
        return view('galleries.index', compact('galleries'));
    }

    /**
     * Show the form for creating a new resource.
     */
    public function create()
    {
        return view('galleries.create');
    }

    /**
     * Store a newly created resource in storage.
     */
    public function store(Request $request)
    {
        $this->validate($request, [
            'caption' => 'required',
            'image' => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
        ]);
    }

```

```

if ($request->hasFile('image')) {
    auth()->user()->galleries()->create([
        // Gallery::create([
        // 'user_id' => auth()->id(),
        'caption' => $request->input('caption'),
        'image' => $request->file('image')->store('galleries', 'public'),
    ]);

    return to_route('galleries.index');
}

return back();
}

```

```

/**
 * Show the form for editing the specified resource.
 */
public function edit(Gallery $gallery)
{
    return view('galleries.edit', compact('gallery'));
}

```

```

/**
 * Update the specified resource in storage.
 */
public function update(Request $request, Gallery $gallery)
{
    $path = $gallery->image;

```

```

$this->validate($request, [
    'caption' => 'required',
    'image' => 'nullable|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
]);

if ($request->hasFile('image')) {
    Storage::delete($gallery->image);
    $path = $request->file('image')->store('galleries', 'public');
}
$gallery->update([
    'caption' => $request->input('caption'),
    'image' => $path,
]);
return to_route('galleries.index');
}

/**
 * Remove the specified resource from storage.
 */
public function destroy(Gallery $gallery)
{
    Storage::delete($gallery->image);
    $gallery->delete();
    return back();
}
}

GalleryIndexController:--
<?php

```



```
namespace App\Http\Controllers;
```

```
use App\Models\Gallery;
```

```
class GalleryIndexController extends Controller
```

```
{
```

```
    /**
```

```
     * Handle the incoming request.
```

```
    */
```

```
    public function __invoke()
```

```
    {
```

```
        $galleries = Gallery::orderBy('created_at', 'desc')->paginate(12);
```

```
        return view('galleryIndex', compact('galleries'));
```

```
    }
```

```
}
```

```
LikedEventController:--
```

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Models\Event;
```

```
class LikedEventController extends Controller
```

```
{
```

```
    /**
```

```
     * Handle the incoming request.
```

```
    */
```

```
    public function __invoke()
```

```
    {
```

```

        $events = Event::with('likes')->whereHas('likes', function ($q) {
            $q->where('user_id', auth()->id());
        })->get();

        return view('events.likedEvents', compact('events'));
    }
}

```

LikeSystemController:--

<?php

```

namespace App\Http\Controllers;

```

```

use App\Models\Event;

```

```

class LikeSystemController extends Controller

```

```

{
    /**
     * Handle the incoming request.
     */
    public function __invoke($id)
    {
        $event = Event::findOrFail($id);
        $like = $event->likes()->where('user_id', auth()->id())->first();
        if (!is_null($like)) {
            $like->delete();
            return null;
        } else {
            $like = $event->likes()->create([
                'user_id' => auth()->id()
            ]);
        }
    }
}

```

```

        });
        return $like;
    }
}
}

```

SavedEventController:--

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Models\Event;
```

```
class SavedEventController extends Controller
```

```

{
    /**
     * Handle the incoming request.
     */
    public function __invoke()
    {
        $events = Event::with('savedEvents')->whereHas('savedEvents', function ($q) {
            $q->where('user_id', auth()->id());
        })->get();

        return view('events.savedEvents', compact('events'));
    }
}

```

SavedEventSystemController:--

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Models\Event;
```

```
class SavedEventSystemController extends Controller
```

```
{
```

```
    /**
```

```
     * Handle the incoming request.
```

```
    */
```

```
    public function __invoke($id)
```

```
    {
```

```
        $event = Event::findOrFail($id);
```

```
        $savedEvent = $event->savedEvents()->where('user_id', auth()->id())->first();
```

```
        if (!is_null($savedEvent)) {
```

```
            $savedEvent->delete();
```

```
            return null;
```

```
        } else {
```

```
            $savedEvent = $event->savedEvents()->create([
```

```
                'user_id' => auth()->id()
```

```
            ]);
```

```
            return $savedEvent;
```

```
        }
```

```
    }
```

```
}
```

```
StoreCommentController:--
```

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Models\Event;
use Illuminate\Http\Request;
```

```
class StoreCommentController extends Controller
```

```
{
    /**
     * Handle the incoming request.
     */
    public function __invoke(Request $request, $id)
    {
        $event = Event::findOrFail($id);
        $event->comments()->create([
            'content' => $request->content,
            'user_id' => auth()->id()
        ]);

        return back();
    }
}
```

```
WelcomeController:--
```

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Models\Event;
```

```
class WelcomeController extends Controller
```

```
{
    /**
```

```

    * Handle the incoming request.
    */
    public function __invoke()
    {
        $events = Event::with('country', 'tags')->where('start_date', '>=', today())->orderBy('created_at',
'desc')->get();

        return view('welcome', compact('events'));
    }
}

```

Create Requests:---

1. CreateEventRequest:--

```

<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class CreateEventRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     */
    public function authorize(): bool
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array<string, \Illuminate\Contracts\Validation\ValidationRule|array|string>
     */
    public function rules(): array
    {
        return [

```

```

        'title' => 'required|max:155|min:2',
        'address' => 'required|max:155|min:2',
        'image' => 'image|required',
        'start_date' => 'required',
        'end_date' => 'required',
        'start_time' => 'required',
        'country_id' => 'required',
        'city_id' => 'required',
        'description' => 'required',
        'num_tickets' => 'required',
        'tags.*' => 'required|exists:tags,id',
    ];
}
}

```

2. ProfileUpdateRequest:--

```

<?php

namespace App\Http\Requests;

use App\Models\User;
use Illuminate\Foundation\Http\FormRequest;
use Illuminate\Validation\Rule;

class ProfileUpdateRequest extends FormRequest
{
    /**
     * Get the validation rules that apply to the request.
     *
     * @return array<string, \Illuminate\Contracts\Validation\Rule|array|string>
     */
    public function rules(): array
    {
        return [
            'name' => ['string', 'max:255'],
            'email' => ['email', 'max:255', Rule::unique(User::class)->ignore($this->user()->id)],
        ];
    }
}

```

3. UpdateEventRequest:--

```

<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

```

```

class UpdateEventRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     */
    public function authorize(): bool
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array<string, \Illuminate\Contracts\Validation\ValidationRule|array|string>
     */
    public function rules(): array
    {
        return [
            'title' => 'required|max:155|min:2',
            'address' => 'required|max:155|min:2',
            'image' => 'image|nullable',
            'start_date' => 'required',
            'end_date' => 'required',
            'start_time' => 'required',
            'country_id' => 'required',
            'city_id' => 'required',
            'description' => 'required',
            'num_tickets' => 'required',
            'tags.*' => 'required|exists:tags,id',
        ];
    }
}

```

Create file in view

View->dashboard.blade.php

->eventIndex.blade.php

->eventsShow.blade.php

->galleryIndex.blade.php

->welcome.blade.php

-> auth

-> components->

application-logo.blade.php

auth-session-status.blade.php

auth-session-status.blade.php

dropdown.blade.php

dropdown-link.blade.php

input-error.blade.php

input-label.blade.php

input-label.blade.php

nav-link.blade.php

primary-button.blade.php

responsive-nav-link.blade.php

secondary-button.blade.php

text-input.blade.php

-> events->

attendingEvents.blade.php

create.blade.php

edit.blade.php

index.blade.php

likedEvents.blade.php

savedEvents.blade.php

-> galleries->

create.blade.php

edit.blade.php

index.blade.php

-> layouts->

app.blade.php

guest.blade.php

main.blade.php

main-navigation.blade.php

navigation.blade.php