Setup and Authentication:

At first we need to create the laravel project by running the following command in the CLI:

```
composer create-project --prefer-dist laravel/laravel event_management
```

To implement user authentication using Laravel Breeze we need to run the following commands in the terminal:

```
composer require laravel/breeze --dev

php artisan breeze:install

npm install

npm run dev
```

Database Setup:

Now we need to set up out database from the '.env' file of the project.

Then we need to create migration for the events by running following command in the terminal:

```
php artisan make:migration create_events_table
```

Now we have to define the columns in the 'up()' method from the migration file that we created from the directory 'database/migrations/' as follow:

```
public function up()

{

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

      $table->id();

      $table->foreignId('user_id')->constrained();

      $table->foreignId('category_id')->nullable()->constrained();

      $table->string('title');

      $table->text('description');

      $table->date('date');

      $table->time('time');
```

```
        $table->string('location');
        $table->timestamps();
    });
}
```

Now we have to run the following command in the terminal for the migration:

```
php artisan migrate
```

Models and Relationships:

Now we have to create ‘Event’ and ‘Category’ model by running following command in the terminal:

```
php artisan make:model Event
php artisan make:model Category
```

This will create us new model files named ‘Event.php’ and ‘Category.php’ in the directory ‘app/Models/’.

Now we have to append the following functions in the ‘Event.php’ file:

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

And have to append the following functions in the ‘Category.php’ file:

```
public function events()

{

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

}
```

Event CRUD:

Now we have to generate a controller in the directory ‘app/Http/Controllers/’ named ‘EventController’ by running following artisan command:

```
php artisan make:controller EventController –resource
```

Now we have to use EventController.php for CRUD operations as following:

```
/**

    * Display a listing of the resource.

    */

   public function index()

   {

      $events = Event::with('category')->paginate(10);

      return view('events.index', compact('events'));

   }

   /**

    * Show the form for creating a new resource.

    */

   public function create()

   {

      $categories = Category::all();

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

   /**

    * Store a newly created resource in storage.

    */

   public function store(Request $request)

   {

      // Validate the request

      $validatedData = $request->validate([

          'title' => 'required|max:255',

          'description' => 'required',

          'date' => 'required|date',

          'time' => 'required|date_format:H:i',

          'location' => 'required|max:255',

          'category_id' => 'nullable|exists:categories,id'  // ensure category exists if provided

      ]);

      // Add the current user's ID to the data

      $validatedData['user_id'] = auth()->id();

      // Create the event

      Event::create($validatedData);

      // Redirect to events index with success message

      return redirect()->route('events.index')->with('status', 'Event created successfully!');

   }

   /**

    * Display the specified resource.

    */
```

```
    public function show(string $id)
    {
        // Retrieve the event by its ID along with its related category
        $event = Event::with('category')->findOrFail($id);

        // Return the show view with the event data
        return view('events.show', compact('event'));
    }
    /**
     * Show the form for editing the specified resource.
     */
    public function edit(string $id)
    {
        // Retrieve the event by its ID
        $event = Event::findOrFail($id);
        // Only the owner of the event or an admin should be able to edit it
        // This is a simple ownership check. You may require more complex logic depending on your
application's needs.
        if ($event->user_id != auth()->id()) {
            return redirect()->back()->with('error', 'You do not have permission to edit this event.');
        }
        // Retrieve all categories for the dropdown in the edit view
        $categories = Category::all();
        // Return the edit view with the event data and categories
        return view('events.edit', compact('event', 'categories'));
    }
    /**
     * Update the specified resource in storage.
     */
    public function update(Request $request, $id)
    {
        // Validate the request
        $validatedData = $request->validate([
            'title' => 'required|max:255',
            'description' => 'required',
            'date' => 'required|date',
            'time' => 'required|date_format:H:i',
            'location' => 'required|max:255',
            'category_id' => 'nullable|exists:categories,id'  // ensure category exists if provided
        ]);
        // Find the event by its ID
        $event = Event::findOrFail($id);
        // Only the owner of the event or an admin should be able to update it
        // This is a simple ownership check. Depending on your application's needs, you might require more
advanced logic.
        if ($event->user_id != auth()->id()) {
            return redirect()->back()->with('error', 'You do not have permission to update this event.');
        }
        // Update the event's attributes
        $event->update($validatedData);
        // Redirect to event's detail page with success message
        return redirect()->route('events.show', $event)->with('status', 'Event updated successfully!');
}
```

Now we have to create the blade files named ‘index.blade.php’, ‘create.blade.php’, ‘edit.blade.php’, and ‘show.blade.php’ in the directory ‘resources/views/events/’ and ‘app.blade.php’ in the directory ‘resources/views/layouts/’.

Create Event (create.blade.php):

```
@extends('layouts.app')

@section('content')

<div class="container">

  <h2>Create Event</h2>

  <form action="{{ route('events.store') }}" method="post">

    @csrf

    <div class="form-group">

      <label for="title">Title</label>

      <input type="text" class="form-control" id="title" name="title" required>

    </div>

    <div class="form-group">

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

      <textarea class="form-control" id="description" name="description" required></textarea>

    </div>

    <div class="form-group">

      <label for="date">Date</label>

      <input type="date" class="form-control" id="date" name="date" required>

    </div>

    <div class="form-group">

      <label for="time">Time</label>
```

```
            <input type="time" class="form-control" id="time" name="time" required>
        </div>
        <div class="form-group">
            <label for="location">Location</label>
            <input type="text" class="form-control" id="location" name="location" required>
        </div>
        <div class="form-group">
            <label for="category">Category</label>
            <select class="form-control" id="category" name="category_id">
                @foreach($categories as $category)
                    <option value="{{ $category->id }}">{{ $category->name }}</option>
                @endforeach
            </select>
        </div>
        <button type="submit" class="btn btn-primary">Submit</button>
    </form>
</div>
@endsection
```

Edit Event (edit.blade.php):

```
@extends('layouts.app')
@section('content')
<div class="container">
    <h2>Edit Event</h2>
    <form action="{{ route('events.update', $event->id) }}" method="post">
```

```
        @csrf
        @method('PUT')
        <!-- Similar input fields to create.blade.php, just pre-fill values using $event object -->
        <button type="submit" class="btn btn-primary">Update</button>
    </form>
</div>
@endsection
```

List All Events (index.blade.php):

```
@extends('layouts.app')
@section('content')
<div class="container">
    <h2>Events</h2>
    <a href="{{ route('events.create') }}" class="btn btn-primary mb-3">Create New Event</a>

    <table class="table table-striped">
        <thead>
            <tr>
                <th>Title</th>
                <th>Date</th>
                <th>Time</th>
                <th>Location</th>
                <th>Actions</th>
            </tr>
        </thead>
        <tbody>
            @foreach($events as $event)
            <tr>
                <td>{{ $event->title }}</td>
                <td>{{ $event->date }}</td>
                <td>{{ $event->time }}</td>
                <td>{{ $event->location }}</td>
                <td>
                    <a href="{{ route('events.edit', $event->id) }}" class="btn btn-sm btn-warning">Edit</a>
                    <a href="{{ route('events.show', $event->id) }}" class="btn btn-sm btn-info">Details</a>
                    <form action="{{ route('events.destroy', $event->id) }}" method="POST" class="d-inline-
block">
                        @csrf
                        @method('DELETE')
                        <button type="submit" class="btn btn-sm btn-danger">Delete</button>
                    </form>
                </td>
            </tr>
            @endforeach
        </tbody>
    </table>
    {{ $events->links() }}
</div>
@endsection
```

Show Event Details (show.blade.php):

```
@extends('layouts.app')

@section('content')

<div class="container">

    <h2>{{ $event->title }}</h2>

    <p><strong>Description:</strong> {{ $event->description }}</p>

    <p><strong>Date:</strong> {{ $event->date }}</p>

    <p><strong>Time:</strong> {{ $event->time }}</p>

    <p><strong>Location:</strong> {{ $event->location }}</p>

    <p><strong>Category:</strong> {{ $event->category->name }}</p>

    <a href="{{ route('events.edit', $event->id) }}" class="btn btn-warning">Edit</a>

</div>

@endsection
```

Layout page(app.blade.php):

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <meta http-equiv="X-UA-Compatible" content="ie=edge">

    <title>Event Management</title>



    <!-- Bootstrap CSS -->

    <link href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">
```

```
    <!-- Additional styles if needed -->

    <link href="{{ asset('css/custom.css') }}" rel="stylesheet">

</head>


<body>

    <nav class="navbar navbar-expand-lg navbar-light bg-light">

        <a class="navbar-brand" href="{{ url('/') }}">Event Management</a>

        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav"

            aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

            <span class="navbar-toggler-icon"></span>

        </button>

        <div class="collapse navbar-collapse" id="navbarNav">

            <ul class="navbar-nav ml-auto">

                @auth

                    <li class="nav-item">

                        <a class="nav-link" href="{{ route('events.index') }}">Events</a>

                    </li>

                    <li class="nav-item">

                        <a class="nav-link" href="{{ route('logout') }}"

                           onclick="event.preventDefault(); document.getElementById('logout-form').submit();">

                            Logout

                        </a>

                        <form id="logout-form" action="{{ route('logout') }}" method="POST" style="display:
none;">

                            @csrf
                        </form>

                    </li>

                @else

                    <li class="nav-item">

                        <a class="nav-link" href="{{ route('login') }}">Login</a>

                    </li>

                    <li class="nav-item">

                        <a class="nav-link" href="{{ route('register') }}">Register</a>

                    </li>

                @endauth

            </ul>

        </div>

    </nav>


    <div class="container mt-4">

        @yield('content')  <!-- This is where the content of child views will be rendered -->

    </div>


    <!-- Bootstrap JS and dependencies -->

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</body>


</html>
```

Middleware for Authentication:

Now we have to use the auth middleware in the the ‘routes/web.php’ file as follow:

```
use App\Http\Controllers\EventController;


Route::middleware(['auth'])->group(function () {
    Route::resource('events', EventController::class);
});
```